Assessment and Optimization of Methods for Tracking People in Riot Control Scenarios

Yvonne Hübner, Jürgen Metzler, Bernhard Dürr, Uwe Jäger and Dieter Willersinn
Fraunhofer Institute for Information and Data Processing (IITB)
Fraunhoferstrasse 1, 76131 Karlsruhe, Germany

ABSTRACT

If for a given application, candidate tracking methods for humans need to be selected and optimized, then relevant sensor and truth data as well as appropriate assessment criteria are required. In the work reported in this contribution we used data recently collected in a riot control scenario. We then processed the sensor data using a set of tracking methods from literature. Tracking results and truth data allowed us to deduce metrics that reflect the usefulness of a tracking method for the selected scenario. The software implementation of the assessment criteria, together with sensor and truth data, forms a benchmark for tracking algorithms in a riot control scenario. It can be used by developers to optimize their tracking systems and to demonstrate their usefulness for application in a riot control scenario. The performance and robustness of optimized tracking methods can considerably improve situational awareness in a riot control scenario.

Keywords: riot control scenario, people tracking, tracker performance assessment, tracker optimization

1. INTRODUCTION

In these days political demonstrations belong to civil liberty and take place all over the world. Unfortunately peaceful demonstrations sometimes turn over to become violent. In order to prevent outbreaks of disorder security personnel has to be trained and educated accordingly. Therefore, authentic scenarios so-called crowd and riot control (CRC) scenarios are imitated by training supervisors and trainees. In reality security guards have to withstand high pressure: Individuals may act differently and not foreseeable as in trained scenarios. In these situation a surveillance systems can assist security guards to establish order.

In this work we define a CRC assistance system which supports the tracking and seizure of an offender in a crowd. For the acquisition of the supervised areas we install cameras on elevated observation points or on unmanned aerial vehicles. For people tracking we use single object trackers such as covariance tracker from Porikli et al. [11] which produces point hypotheses. Our main focus lies on the tracking module and its optimization using meaningful performance evaluation metrics.

In recent years there were a number of publications in the field of performance evaluation of object detection and object tracking systems which was advanced by the international workshop of PETS (Performance Evaluation of Tracking and Surveillance). In order to get unbiased objective results of performance evaluation there is a need to define standard metrics [15]. In evaluation of object detection algorithms there was an achievement of consistent metrics. By contrast for tracking algorithms there is no general agreement for assessment metrics. Basic metrics such as true positive are defined differently by authors: In [5] Ellis defines true positives as “the number of observations confirmed by the ground truth” whereas Bashir and Porikli define it as the “number of frames where both ground truth and system results agree on the presence of one or more objects…”[1]. In literature authors define metrics that assess every single frame (Bashir and Porikli), or detected objects (Bashir and Porikli [1] / Lazarevic-McManus et al. [6]) or overall trajectories (Perera et al. [10]). The decision what metrics are appropriate for performance evaluation depend on the task definition.

In this paper, we propose suitable metrics for quantitative assessment and optimization of CRC trackers and we evaluate the performance of three different trackers that are demonstrated as appropriate for riot control scenarios. A qualitative assessment of several trackers resulted that these trackers are the best ones for CRC.

In Section 2 we define the CRC assistance system and describe its individual modules. Then we present the data set recently collected in a CRC scenario in Section 3 and in Section 4 we give a brief summary about the tracking methods.

In Section 5 we present suitable metrics for CRC tracker. The results of the performance evaluation are shown in Section 6 before we conclude our contribution in Section 7.

2. CRC ASSISTANCE SYSTEM

We define a CRC assistance system which subsequently tracks an individual in a crowd. In the flow chart (Fig. 1) we show one loop of the system with the four modules: data acquisition, manual detection, tracking and situation analysis. The assistance system starts with the data acquisition by using optical sensors which outputs a single image. In the first loop of the system the object is initialized by an operator sitting in front of a display. The operator selects a suspicious individual in the image by clicking on her or his head. The described process is called manual detection. It delivers the object position to the tracking module. After a successful object initialization the process of manual detection can be skipped in subsequent loops. In this case the object position of the last loop is used as the input of the tracking module. During the tracking module the detected object position of image t-1 is retrieved in image t within a region of interest. As a result the predicted object position is used in the situation analysis module. In the situation analysis expertise is used to detect possible dangerous situations and suspicious behavior. During the runtime of the assistance system the operator can interact with the system and readjust the tracking window if necessary.

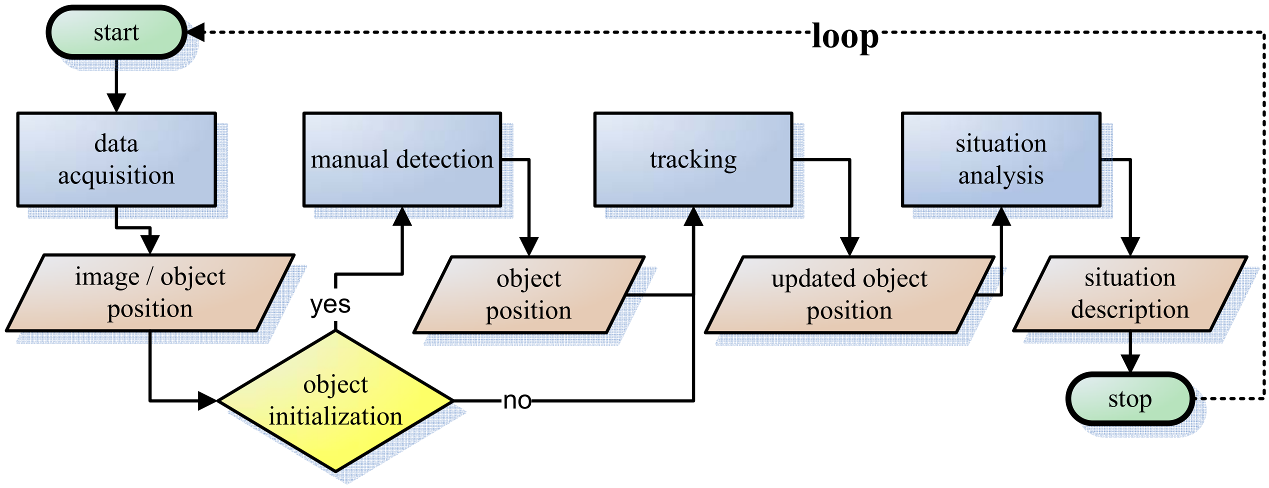

Fig. 1. CRC assistance system

3. DATA SET

For our work we used data which was collected during a military CRC training. The camera was installed 25 meter high on a crane platform and recorded data from nadir view (see Fig. 2). The CRC scenarios consisted of three different levels of escalation. First the crowd started with a peaceful demonstration. Later there were violent protests and an escalation of the riot where offenders bumped into the chain of guards.

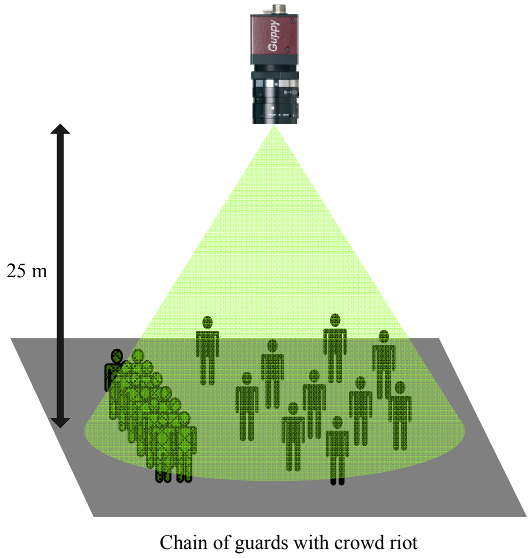

Fig. 2. Technical data acquisition details

In data acquisition we used a single-chip CMOS camera with a 20Hz sampling rate and a resolution of 752 x 480 pixels. The resolution of an individual is about 16 x 16 pixels (compare Fig. 3). Each pixel of the image sensor chip captures only 8 bit: one of the three basic colors red, green or blue. The 24 bit color information for each pixel is computed in an embedded software module.

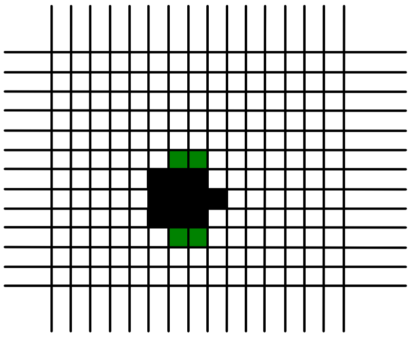

Fig. 3. Sample resolution of an individual

As a result of the nadir view there are only few object occlusions. Naturally, pictures from unmanned aerial vehicles contain a crowd consisting of small individuals which is a challenge for our tracking module (compare Fig. 4). Additionally, in our real data sets we have got to deal with variations of contrast of the individuals to the background. There are individuals with weak (see Fig. 5 circle 1) and significant (see Fig. 5 circle 2) contrast to the background. Another challenge is the similarity of objects (see Fig. 5 circle 3). As a consequence of fast changing lightening conditions there are local and global variations of luminance linked with strong shadowing effects (see right image in Fig. 4).

Fig. 4. Snapshots from the CRC data set

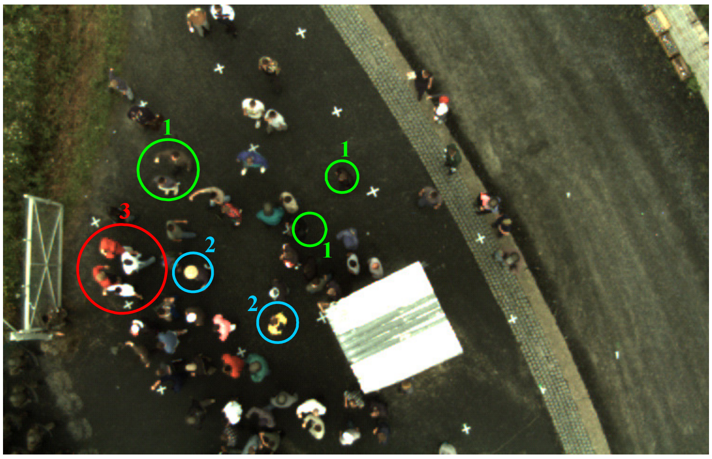

Fig. 5. Data set challenges

4. TRACKING METHODS

We qualitatively assessed a set of original tracking methods from literature and adapted methods to CRC scenarios. The qualitative assessment allowed us to compile a first ranking of the tracking methods. In this section we give a brief summary of the best three trackers.

4.1 Color Based KLT

We extended the pyramidal implementation of the KLT (Kanade-Lucas-Tomasi Feature Tracker) [4] by color weighted features. Detailed information about the original KLT can be found in [7, 12 and 13]. Here we give a short overview of the color based pyramidal KLT. First, weights of colors are determined within a tracking window which contains the target object. A color weight is obtained by subtracting the number of a color in a specified tube around the tracking window from the number of this color within the window. Additionally, the weights are normalized with respect to the total number of colors. Thus, high positive values for weights indicate colors that only exist on the target. Then local image structures that exhibit strong gradients are extracted within the tracking window and are weighted by the color weights: The weights of each pixel within the window are added and divided by the total number of pixels inside. After that, the optical flow vectors for the image structures are calculated by the pyramidal KLT and the mean endpoint of all vectors which points to the new location of the tracking window are determined. Thereby, the vectors are weighted by the weights of the image structures and the sum of squared differences between the image structures of the starting and receiving points.

4.2 Color Histogram Tracker

Histogram trackers are widely used and demonstrated as reliably [8, 9]. As color is an important feature for tracking in CRC scenarios we adapted a color histogram tracker to these scenarios. We give a brief summary of the adapted histogram tracker. The first step is equivalent to the color weighting process of the color based KLT. Then a color histogram is calculated and weighted by the color weights. In the next image this histogram is compared with histograms extracted from the region near the target object. The best match is determined with a figure of merit and returned by the algorithm.

4.3 Covariance Tracker

The main idea of the covariance tracking method of Porikli et al. is to describe image regions by covariance matrices [11]. For each region of interest pixel a number of features such as position, color and gradients of pixels is measured and coded in a covariance matrix. This so-called covariance descriptor contains information about linear correlations between the features. After the detection of the region of interest the covariance descriptor is computed and stored in an initial model. In the next step the initial model is searched in a target image. The algorithm returns the descriptor with the smallest difference to the initial model and then it updates the initial model with this one.

5. PERFORMANCE EVALUATION

The basis of performance evaluation is to define appropriate metrics that fulfill tracking requirements. According to Bernardin et al. there is no common agreement for suitable metrics for performance evaluation of tracking algorithms among image processing experts [2].

For a CRC tracker we claim:

1. The correct recognition of the object from frame to frame.
2. The precise estimation of the object position.
3. The consistent tracking of an object during the image sequence.

Performance evaluation is necessary to assess the robustness of tracking algorithms and to compare them with results from literature [2]. Therefore a number of metrics were presented in the past. In [1] the authors distinguish metrics in frame-based and object-based metrics whereas Perera et al. define frame-based and track-based metrics [10]. Frame-based metrics evaluate only one frame at a time unlike object-based or track-based metrics which assess trajectories. Trajectories are created by the interpolation of discrete hypothesis points obtained from a tracking algorithm.

In literature there is an agreement of basic metrics from signal detection theory being both relevant for performance evaluation of detection and tracking algorithms. These are: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). For every image containing ground truth or not the tracking algorithm returns hypotheses as centers of gravity or not. In [5] Ellis defines the following four combinations:

	Ground truth existing	Ground truth not existing
Algorithm returns hypothesis	TP (detection)	FP (false alarm)
Algorithm does not return hypothesis	FN	TN

Table 1. Metrics from signal detection theory

In spite of the agreement of these basic metrics in literature there is a difference of their interpretation. For instance, the sense of TP differs from the number of hits counted for one ground truth object [5], so multiple detections are counted. Lazarevic-McManus et al. define TP as the number of detected ground truth objects per sequence [6]. Bashir and Porikli define it as the number of images having at least one object detection per image [1]. According to Yin et al. using all four basic metrics is useful for performance evaluation of detection algorithms [14]. In our case, the metric TN isn't significant because no background images were recorded. For the work reported here we used exact ground truth polygons and tracking hypotheses as centers of gravity. To determine the basic metrics we ran a point-in-polygon test.

We define the basic metrics as follows:

N_{TP} The number of objects with at least one detection per ground truth.

N_{FP} The number of detected objects not confirmed by ground truth.

N_{FN} The number of ground truth where no hypothesis was assigned.

[1] and [6] define normalized rates to obtain quantitative evidence. These are: True Positive Rate t_p, False Negative Rate f_n, Positive Prediction p_p and False Alarm Rate f_a.

The True Positive Rate is the relation of the correct detected objects to the number of ground truth objects and is defined as follows:

$$t_p = \frac{N_{TP}}{N_{TP} + N_{FN}} = 1 - f_n.$$

The False Negative Rate is the relation between the numbers of not detected objects and the number of ground truth objects and is defined as follows:

$$f_n = \frac{N_{FN}}{N_{TP} + N_{FN}} = 1 - t_p.$$

The Positive Prediction, likewise a ratio for the correctness of the algorithm, is computed by the relation of the number of detections and the number of hypotheses:

$$p_p = \frac{N_{TP}}{N_{TP} + N_{FP}} = 1 - f_a.$$

The False Alarm Rate defines the relation of the number of false alarms and the number of hypotheses. It gives evidence to the error rate of the tracker:

$$f_a = \frac{N_{FP}}{N_{TP} + N_{FP}} = 1 - p_p.$$

The presented rates fulfill the first requirement. The correct recognition of objects from frame to frame can be measured by the True Positive Rate and Positive Prediction. If tracking algorithms return exactly one object hypothesis per image containing exactly one ground truth, the number of false negatives and false positives is always identical because the number of ground truth objects is equal to the number of tracking hypotheses.

To accomplish the second requirement: the precise estimation of the object position, we use a known metric from literature. The Object Tracking Error (OTE) defined by the authors Black et al. [3] measures the Euclidean Distance of the ground truth center of gravity and the hypothesis center of gravity for one ground truth object of an image. It is defined as follows:

$$OTE = \frac{1}{N_{rg}} \sum_{i \in g(t_i) \wedge r(t_i)} \sqrt{\left(x_i^g - x_i^r\right)^2 + \left(y_i^g - y_i^r\right)^2},$$

Whereas N_{rg} is the number of images containing both ground truths and tracking results. x_i^g and y_i^g define the x- and y-coordinate of the ground truth center of gravity in the i^{th} image respectively x_i^r and y_i^r are the coordinates of the hypothesis center of gravity in the i^{th} image.

For the sake of completeness, we define the dispersion measure of the OTE which is the variance VAR_{OTE}. It is given through

$$VAR_{OTE} = \frac{1}{N_{rg}} \sum_{i=1}^{N_{rg}} E((OTE_i - OTE)^2).$$

Additionally, we computed the minimal OTE (min_{OTE}) and maximal OTE (max_{OTE}):

$$min_{OTE} = \min(OTE_i) \qquad and$$
$$max_{OTE} = \max(OTE_i) \qquad , i = 1 \ldots N_{rg}.$$

The last requirement is the consistent track of an object during an image sequence. Therefore, we introduce the metrics hold-in duration, hold-in loss and track fragmentation.

The hold-in duration specifies how many images were tracked subsequently without a tracking loss, in Fig. 6 e.g. 8 images in track 1 and 5 images in track 2. The hold-in loss specifies the number of images which were not detected subsequently, e.g. 1 image (see Fig. 6). The track fragmentation specifies the number of track fragments, e.g. 1 in Fig. 6 below.

To enhance the significance of the hold-in duration, hold-in loss we defined additionally the mean, the minimum and maximum value.

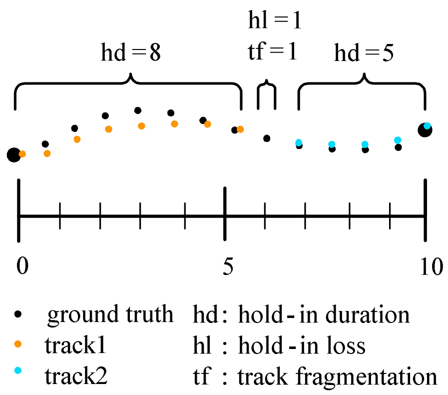

Fig. 6. Definition of hold-in duration, hold-in loss and track fragmentation

6. RESULTS

For performance evaluation we considered several individuals within a sequence of 1000 images. During the whole image sequence the individuals were visible. The ground truth was generated semi-automatic with the in-house annotation tool CART [8, 9]. The assessed tracking methods were Color Based KLT (KLT), Color Histogram Tracker (ColHist) and Covariance Tracker (Covariance) described in Section 4.

The results are shown in Table 2 and Fig. 7 to 11. The results are averaged over the individuals. We plotted the True Positive Rate and False Negative Rate, the hold-in duration, the hold-in loss, the OTE and the track fragmentation. As stated previously the results of True Positive Rate and Positive Prediction respectively False Negative Rate and False Alarm Rate are in each case equal because the trackers returned exactly the same number of hypotheses per image as it contained ground truths.

Metrics	KLT	Covariance	ColHist
Total ground truth	1000	1000	1000
Number of TP	918,2	958,8	538,9
Number of FP	81,8	41,2	461,1
Number of FN	81,8	41,2	461,1
True Positive Rate	92,56%	96,26%	53,73%
False Negative Rate	7,44%	3,74%	46,27%
Positive Prediction	92,56%	96,26%	53,73%
False Alarm Rate	7,44%	3,74%	46,27%
OTE	11,0	9,8	31,0
min_{OTE}	0,9	1,1	1,2
max_{OTE}	33,1	28,1	105,6
Hold-in duration µ	739,3	655,8	309,6
Hold-in duration min	672,7	569,8	268,1
Hold-in duration max	845,2	864,9	429,6
Hold-in loss µ	38,1	3,7	168,7
Hold-in loss min	29,3	0,6	105,9
Hold-in loss max	69,5	27,8	324,8
Track Fragmentation	1,9	4,6	6,0

Table 2. Performance evaluation overview

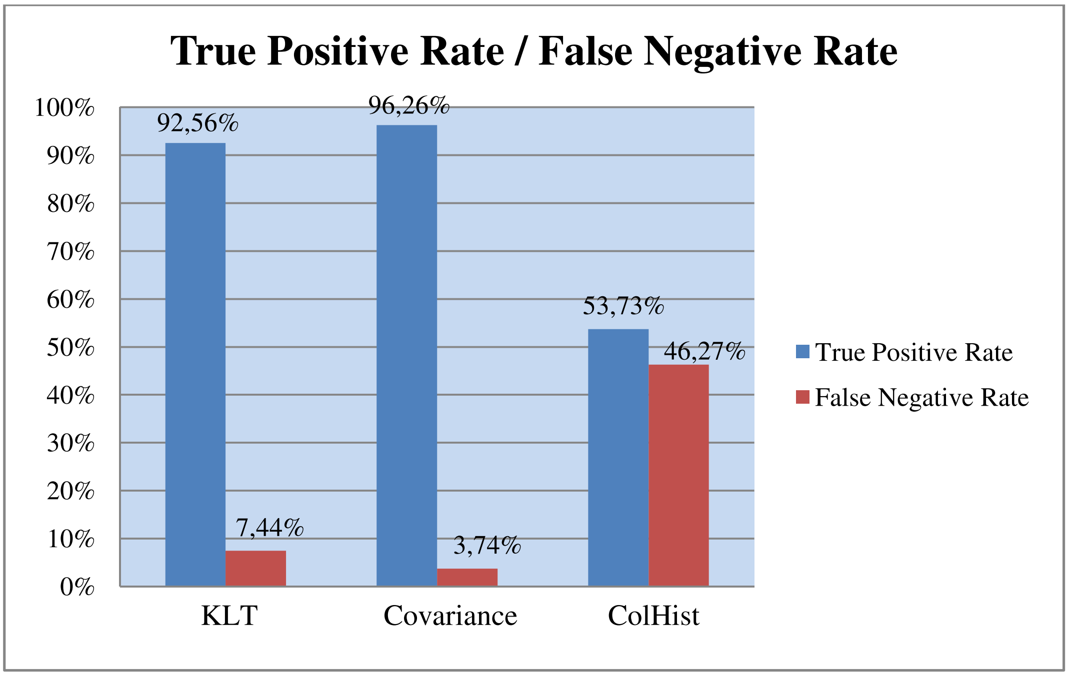

Fig. 7. True Positive Rate / False Negative Rate

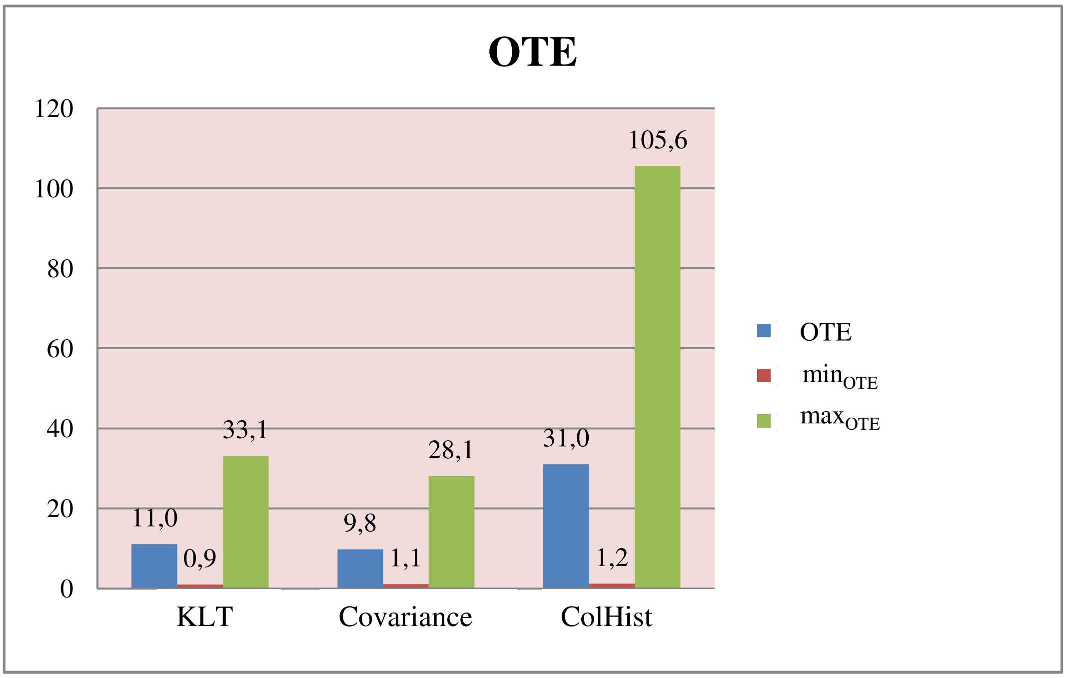

Fig. 8. OTE

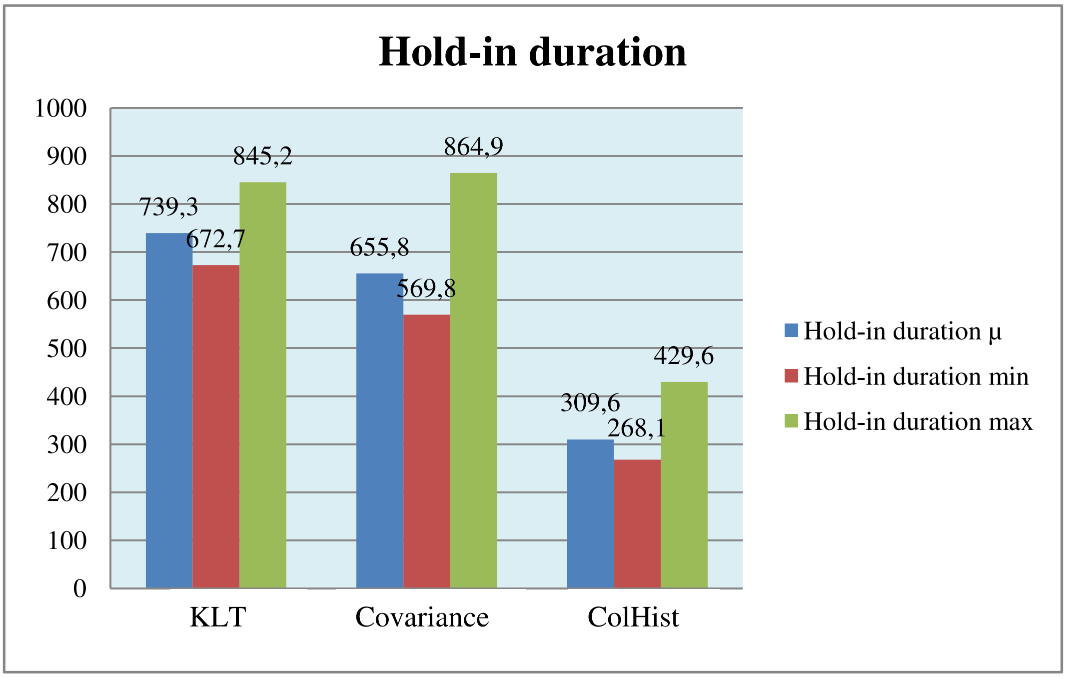

Fig. 9. Hold-in duration

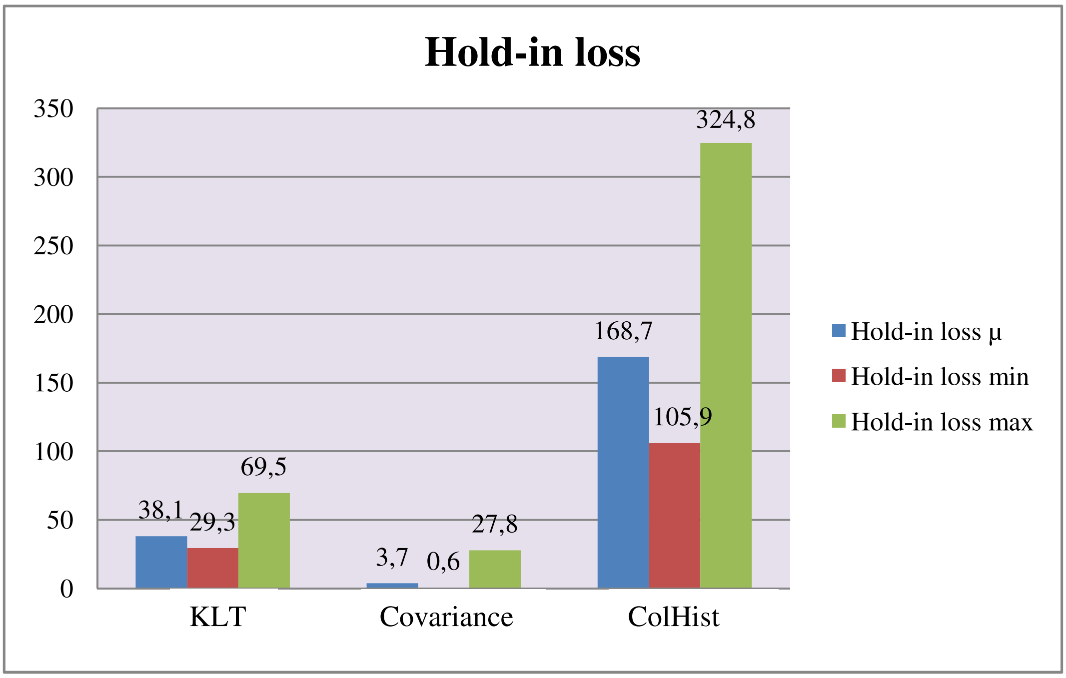

Fig. 10. Hold-in loss

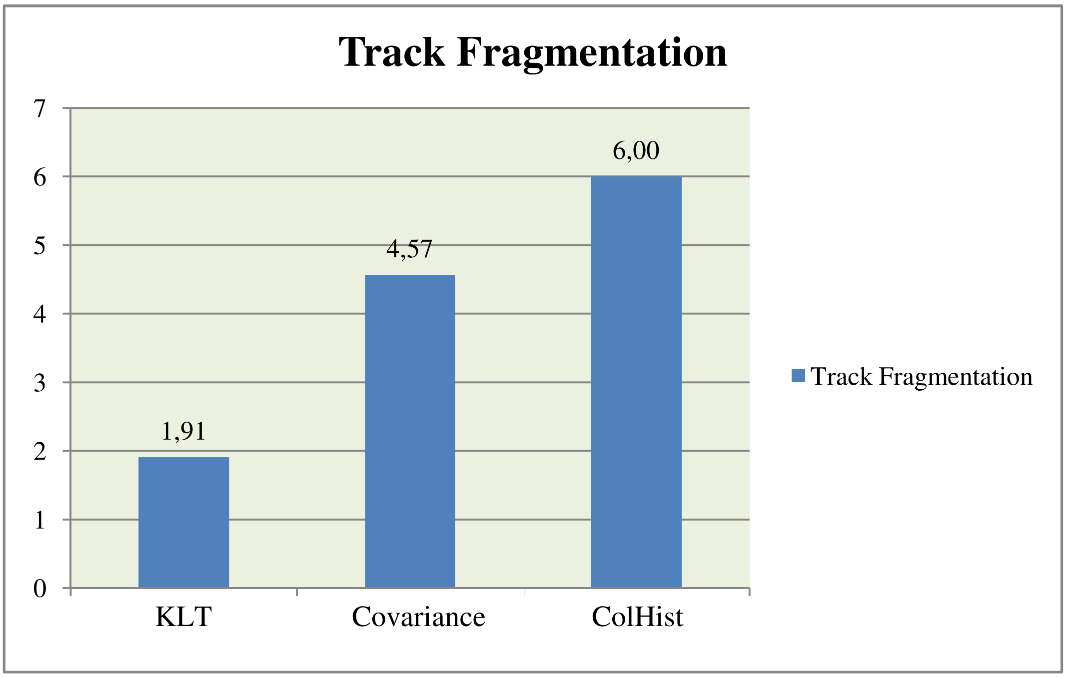

Fig. 11. Track Fragmentation

7. SUMMARY

We defined a CRC assistance system that supports the control of crowds and the breakup of riots. Since in riot control scenarios it is important to track individual people reliably a performance evaluation of tracking methods is necessary. We proposed metrics for quantitative performance evaluation and optimization of tracking methods for riot control scenarios. The metrics, sensor data recently collected in a riot control scenario and the ground truths form a benchmark for CRC trackers. We used it to optimize trackers for CRC scenarios proposed in this contribution and for the performance evaluation of these tracking algorithms. The benchmark is a beneficial support for the developing of methods for tracking people in riot control scenarios.

REFERENCES

[1] Bashir, F. and Porikli, F., "Performance Evaluation of Object Detection and Tracking Systems", Proc. of 9th IEEE International Workshop on PETS, New York, 7-14 (2006).

[2] Bernardin, K., Elbs, A. and Stiefelhagen, R., "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment", Proc. of 6th IEEE International Workshop on Visual Surveillance, Graz (2006).

[3] Black, J., Ellis, T. and Rosin, P., "A novel method for video tracking performance evaluation", IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), Nice, 125–132, (2003).

[4] Bouguet, J.-Y., "Pyramidal Implementation of the Lucas Kanade Feature Tracker - Description of the algorithm", Intel Research Laboratory, Technical Report (1999).

[5] Ellis, T., "Performance Metrics and Methods for Tracking in Surveillance", Proc. of 3rd IEEE International Workshop on PETS, Copenhagen, 26-32 (2002).

[6] Lazarevic-McManus, N., Renno, J.R., Makris, D. and Jones, G.A., "Designing Evaluation Methodologies: the case of motion detection", Proc. of 9th IEEE International Workshop on PETS, New York, 23-30 (2006).

[7] Lucas, B. D. and Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision", International Joint Conference on Artificial Intelligence, 674-679 (1981).

[8] Müller, T. and Müller M., "Computer-aided camouflage assessment in real-time", Proc. of the SPIE 6543: Infrared Imaging Systems: Design, Analysis, Modeling and Testing, SPIE Defense and Security Symposium, Orlando (2007).

[9] Müller, T. and Müller M., "CACAMO – Computer Aided Camouflage Assessment of Moving Objects", Proc. of the SPIE: Infrared Imaging Systems: Design, Analysis, Modeling and Testing, SPIE Defense and Security Symposium, Orlando (2008).

[10] Perera, A., Hoogs, A., Srinivas, C., Brooksby, G. et al., "Evaluation of Algorithms for Tracking Multiple Objects in Video", Proc. of 35th Applied Imagery and Pattern Recognition Workshop, Washington D.C., 35 (2006).

[11] Porikli, F., Tuzel, O. and Meer, P., "Covariance Tracking using Model Update Based on Means on Riemannian Manifolds", Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, New York (2005).

[12] Tomasi, C. and Kanade, T., "Detection and Tracking of Point Features", Carnegie Mellon University Technical Report CMU-CS-91-132 (1991).

[13] Tomasi, C. and Shi, J., "Good Features to Track", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 593-600 (1994).

[14] Yin, F., Makris, D. and Velastin, S.A., "Performance Evaluation of Object Tracking Algorithms", Proc. of 10th IEEE International Workshop on PETS, Rio de Janeiro, 17-24 (2007).

[15] Young, D. and Ferryman, J., "PETS Metrics: On-Line Performance Evaluation Service", Proc. 2nd Joint IEEE International Workshop on VS-PETS, 317-324 (2005).