Pedestrian Tracking in Video Sequences using Particle Filters

Philipp Mondorf
KTH Royal Institute of Technology
Stockholm, Sweden
mondorf@kth.se
Author

Thomas Labourdette-Liaresq
KTH Royal Institute of Technology
Stockholm, Sweden
tbll@kth.se

Devrat Singh
KTH Royal Institute of Technology
Stockholm, Sweden
devrat@kth.se

***Abstract*—In this work, we evaluate and compare three different methods for pedestrian tracking in video sequences using particle filters. Pedestrian tracking in video data is an important field of research with a variety of applications, ranging from surveillance systems to self-driving cars. Particle filters have become popular tools in solving visual tracking tasks as they are capable of handling complex non-linear motions and non-Gaussian distributions. For this work, two particle filter approaches are implemented that use different image features to compare the target pedestrian with respective state estimates. While the first technique is based on HSV color histograms, the second method makes use of moment invariants. All tracking approaches use a linear Support Vector Machine (SVM) based on a Histogram of Oriented Gradients (HOG) to detect the target pedestrian. Furthermore, the parameters of the PF are adaptively adjusted based on the quality of the detection. By combining the two particle filter approaches, we are able to create a third tracking system that benefits from the advantages of both former techniques. We evaluate the introduced methods on a highly challenging dataset, showing that the combined tracker is robust against various challenges in visual object tracking and able to outperform the other two approaches.**

I. Introduction

Tracking moving objects appears to be a natural task to us humans. It is crucial to safely interact with our environment, for example when navigating through crowded scenes without running into other pedestrians. However, this extraordinary visual capability of humans is the result of thousands of years of evolution. When we track pedestrians, our brain performs highly complex calculations, processing an incredible amount of information, e.g. about the pedestrian's velocity, occurring motion patterns [3], cultural norms or the pedestrian's body language [11].

Despite its complexity, pedestrian tracking has attracted the interest of researchers for decades. It plays an important role in many computer vision systems such as surveillance, video annotation, sports reporting or traffic management. With the development of autonomous moving platforms such as self-driving cars or social robots that need to share their physical environment with humans, pedestrian tracking has become even more valuable [21]. Even so, tracking pedestrians in video sequences can be particularly challenging due to the following reasons [38]:

- **Variations in illumination**. The recorded video data can be subject to variations in illumination. In this manner, the tracked pedestrian might appear in different colors or lightning within a video sequence.
- **Loss of depth information**. Most cameras, in particular mono-cameras, do not incorporate depth information.
- **Occlusions**. Tracked pedestrians can be occluded by other static or moving objects.
- **Unexpected motion**. It might occur that the tracked pedestrian suddenly stops or abruptly changes direction.
- **Similar Distractors**. One or more pedestrians might enter the scene who look very similar to the actual tracked pedestrian.
- **Relative motions**. Despite the motion of the tracked pedestrian, it is possible that the camera or the reference background is moving. This makes it particularly difficult to identify the target.

To tackle the above mentioned problems, various techniques have been developed to track pedestrians even under difficult conditions [29, 35, 36]. In general, the tracking task can be subdivided into three main parts:

1) Detect target and determine its state vector (location, velocity, etc.).
2) Track detected target by identifying how its state vector changes in consecutive frames.
3) Analyze tracking data (e.g. path prediction, determination of motion behavior, etc.)

In this work, we implement three different particle filters in order to track pedestrians in video data. In particular, color histograms and moment invariants are used for the observation model of particle filters to compare the estimation with the actual target. While a similar work has already been done by Junxiang et al. to track faces in video sequences [20], we improve and extend this approach to track pedestrians in more complex scenarios, including non-rigid targets, similar distractors, full occlusions and a moving camera. In order to detect a target pedestrian and obtain its initial state, we use a linear Support Vector Machine (SVM) that is based on a Histogram of Oriented Gradients (HOG) [7]. Furthermore, we adaptively adjust the parameters of the particle filters based on the quality of the detection. This approach is inspired by the

work done by Weng et al. who introduce an adaptive kalman filter that adjusts its estimation parameters, respectively [36]. Our research goal is to compare the performances of the different particle filters and evaluate their suitability for the respective tracking task. In general, it is important that a tracking algorithm satisifes the following three criteria: *simplicity*, *adaptivity* and *robustness*. *Simplicity* implies that the algorithm is easy to implement and computationally efficient, while *adaptivity* requires the tracking system to be able to adapt to various changes regarding the target or the environment. *Robustness* demands that the algorithm can successfully track the target pedestrian even under complex settings [2, 37]. In our study, we will therefore particularly analyze and compare the methods implemented with regards to these criteria.

We begin this work by providing a brief overview of existing object detection and tracking algorithms that can be used for pedestrian tracking in video sequences. For this, we discuss the advantages and disadvantages of different techniques in section II. We continue by explaining the basic concepts of the particle filter algorithm and how it can be used for pedestrian tracking in section III. In this section, we also discuss the details of the different observation models and their implementation. In section IV, we present and discuss the results of our experiments. Finally, we discuss the key findings of our work in section V.

II. Related Work

Tracking pedestrians in video data is a rich field of research with remarkable contributions from all over the world. With the development of autonomous moving platforms that frequently need to interact with humans, pedestrian tracking has become increasingly popular in recent years [9]. Hence, many works can be found on this topic. In the following, we give an overview of important concepts and techniques that have been introduced to the subject. Furthermore, we explain how these concepts relate to the work done in this study.

A. Object Detection Techniques

As mentioned in section I, every tracking method consists of an object detection part to distinguish the target object from its surroundings, and a tracking technique to pursue the development of the target's state. According to Yilmaz et al. [38], object detection techniques can be subdivided into four categories:

- **Point Detectors.** Point detectors aim to find interest points in the processed image/frame. These points have distinguishable characteristics in their localities (e.g. edges, corners, blobs) and are preferably invariant to changes in illumination and camera viewpoint. Commonly used point detectors are the *Moravec's detector* [27], *Harris interest point detector* [12], *KLT detector* [19] and *SIFT detector* [23]. An extensive comparison between the different point detectors is done by Mikolajczyk and Schmid [25].
- **Segmentation.** Image segmentation techniques aim to divide the processed image/frame into semantically meaningful regions. Prominent methods are *Mean-shift* clustering [4, 5], *Graph-Cuts* [18] and *Active Countours* [31]. While *Mean-shift* clustering segments the image into groups of pixels of similar characteristics by finding local maxima in a high-dimensional feature space, *Graph-Cuts* partitions the image into foreground and background by minimizing an energy-based graph model. In an *Active Contours* framework, the segmentation is found by evolving a contour towards the targeted object boundary.
- **Background Subtraction.** Moving objects in an image can be detected by building a model for the background and defining deviations from it as foreground. The foreground, i.e. the target is obtained by subtracting the background model from the current image. However, background subtraction techniques assume a stable background image that is not always given [26, 28, 32].
- **Learning Methods.** Object detection can be achieved by learning characteristic features of the target by processing a sufficient amount of data. In particular, a function is learned from data that maps a certain image input to the desired output. In recent years, this approach has become increasingly popular and happens to outperform other approaches [9]. Examples are *Support Vector Machines* (SVM) [1] or *Adaptive Boosting* [10]. While SVMs cluster data into two categories by maximizing the distance between data points and a constructed hyperplane, *Adaptive Boosting* combines many base classifiers (so-called *weak* classifiers) into one accurate *strong* classifier.

B. Object Tracking Techniques

Once the respective target has been identified, its state needs to be tracked. As mentioned earlier, object tracking is a popular field of research that has been studied extensively. Various attempts have been made to classify different object tracking systems. Fiaz et al. classify tracking algorithms as trackers that utilize correlation filters (CTFs) and non-correlation trackers (NCFTs) [9]. Zhou et al. categorize visual tracking systems into deterministic algorithms that usually solve an optimization problem and stochastic algorithms that often reduce to an estimation problem [39]. Other works classify visual tracking systems as generative vs. discriminative, online vs. offline learning, or context-aware vs. non-aware techniques [37]. An extensive comparison between different tracking algorithms can be found e.g. in [9, 39, 38] .

In this study, we will restrict ourselves to Sequential Monte Carlo (SCM) based methods [8]. Particle filter methods have become popular tools in solving visual object tracking problems due in parts to the introduction of the CONDENSATION algorithm [15]. Compared to kalman filter approaches that have been used in early works [16, 17, 36], SCM based methods can handle complex non-linear motion behavior and non-Gaussian posterior distributions. In this manner, SCM based pedestrian tracking techniques have been able to tackle various difficulties of the tracking task such as moving cameras, occlusions or

unexpected and complex motion behaviors. Although the basic particle filter framework usually remains the same among different approaches, different features are used to track the target. Several works rely on the target's shape for tracking [13, 15, 24]. Taekyu and Sukbum [33] make use of seven moment invariants of the target image in order to define a corresponding feature vector. As color-based image features are more robust against out-of-plane rotations, Owczarek et al. [29] compute a HSV histogram for the target frame in order to track a pedestrian. Junxiang et al. [20] combine these two approaches and introduce an Integration of Color-based and Moment-based tracker (ICM). By doing so, they are able to successfully track faces in video sequences.

In this work, we follow the approach of Junxiang et al. [20] by introducing an ICM tracking system that tracks pedestrians in a highly complex environment. In order to detect the target pedestrian, we use a HOG-based Support Vector Machine [7]. Similar to [39, 36], we achieve further robustness by adaptively adjusting the parameters of the particle filter based on the results of the target detection.

III. Particle Filter for Pedestrian Tracking

In this section, we describe the details of our particle filtering approach for pedestrian tracking in video sequences. Particle filters belong to the class of nonparametric filters that approximate posteriors by a finite number of values. This nonparametric approach allows for representing a broad space of distributions. Furthermore, particle filters are able to model nonlinear transformations [34]. As most tracking problems are non-Gaussian and often nonlinear, particle filters are highly suitable for this task [15, 29, 33]. For a deeper understanding of the concepts of particle filters, we would like to refer to the work done by Doucet et al. [8]. Our research goal is to compare different techniques of particle filtering for pedestrian tracking. For this, we follow the idea of Junxiang et al. [20], using two different observation models to track the target. In this manner, we are able to compare three different particle filter approaches. While the first method uses a color-based likelihood model, the second technique compares the estimated state and target based on moment invariants. The third method combines these two approaches.

Figure 1 illustrates a systematic overview of our method. As it can be seen, particle filtering is an iterative technique, where the target's state distribution is consecutively estimated and corrected, incorporating measurements from the detection method. First, a set of particles are randomly initialized according to a uniform distribution over the state space. Then, new particle states are predicted, given the transition equation of a motion model. The weights of the particles are subsequently updated, using a likelihood model either based on a color-histogram or invariant moments. As the target pedestrian might change appearance in color and shape during the video sequence, e.g. due to illumination changes or out-of-plane rotations, the target is updated if the results of the detection methods are of sufficient quality. This ensures additional robustness. Finally, the estimated states of both

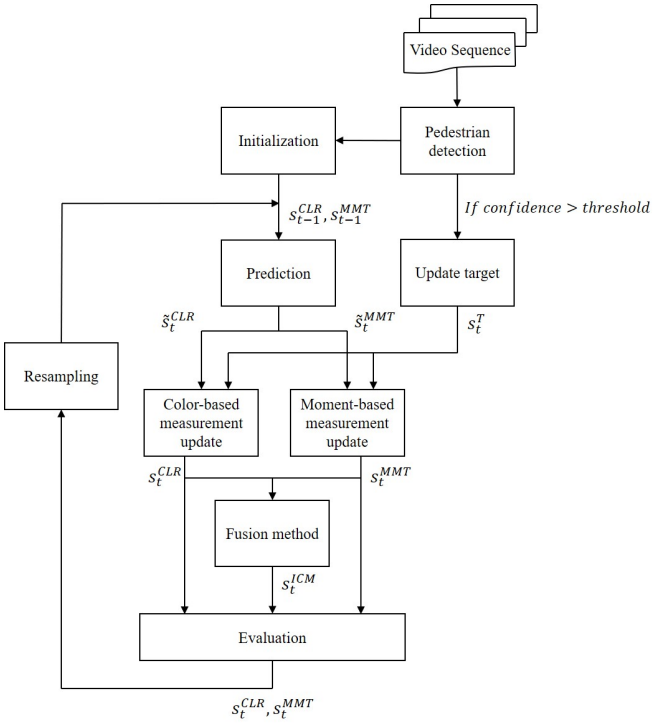


Fig. 1: A systematic overview of our method

approaches are combined to obtain an estimate of the ICM tracker. All estimated states are evaluated. Subsequently, the particles are resampled and the estimation process is repeated. In the following, we will discuss each step in more detail.

A. Initialization

Given the first frame of a video sequence, a target has to be defined that is tracked in the subsequent frames. Linear Support Vector Machines (SVM) have proven to be successful classifiers when given the right feature sets [1]. In this work, we use a pre-trained SVM that uses well-normalized local histograms of image gradient orientations to detect pedestrians in a frame. Gradients are particularly useful to identify edges and corners and convey a lot of information about an object's shape. Dalal et al. [7] have shown that this techniques outperforms other existing edge and gradient based human descriptors and works even under difficult conditions. Once the target pedestrian has been identified, the detection method returns the location and size of a rectangle, i.e. a bounding box that marks the pedestrian in the current frame. Subsequently the state vector of the target is defined:

$$\begin{aligned} \boldsymbol{s}^T &= [x, y, v_x, v_y, H_x, H_y, k]^T \\ \boldsymbol{s}_0^T &= [x, y, 0, 0, H_x, H_y, 1]^T \end{aligned} \tag{1}$$

where (x, y) are the coordinates of the center of the representation rectangle and $(v_x, v_y) = (\frac{dx}{dt}, \frac{dy}{dt})$ denote the center point's velocity. H_x and H_y represent the rectangle's width and height, while k defines the scale of the rectangle with respect to its initial size. After the target state has been

defined, a set of particles are initialized. Each particle is defined by a state vector s that is randomly drawn from a uniform distribution over the state space and a weight π. The state vector is of the same form as the target's state and the weights are assigned equal values:

$$\begin{aligned} \boldsymbol{s}_0^{(n)} &= [x, y, v_x, v_y, H_x, H_y, k]^T \quad \text{for } n = 1, ..., N \\ \pi_0^{(n)} &= \frac{1}{N} \qquad\qquad\qquad\qquad\quad \text{for } n = 1, ..., N \end{aligned} \tag{2}$$

where N is the number of particles. Note that as we use two different observation models to update the particles' weights, two respective sets of particles are initialized, i.e. $\boldsymbol{S}^{CLR} = (\boldsymbol{s}^{CLR}, \boldsymbol{\pi}^{CLR})$ and $\boldsymbol{S}^{MMT} = (\boldsymbol{s}^{MMT}, \boldsymbol{\pi}^{MMT})$, where the superscript CLR denotes particles that are updated by the color-based likelihood model, while MMT represents the set that is updated by the moment-based model.

B. Prediction

As the target pedestrian changes position during a video sequence, new particle states have to be predicted for every new frame. Due to relative motions of the camera and out-of-plane rotations in the given video data, this is a rather difficult task. In order to find a suitable motion model that describes the respective state transitions well, we follow existing approaches [20, 29] that use a second order linear difference equation based on prior observations:

$$\boldsymbol{s}_t^{(n)} = \boldsymbol{A}_t \, \boldsymbol{s}_{t-1}^{(n)} + \boldsymbol{w}_{t-1}^{(n)} \tag{3}$$

where $\boldsymbol{w}_{t-1}^{(n)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t)$ is a vector of random values drawn from a multivariate normal distribution that is governed by the covariance matrix of the motion model $\boldsymbol{Q}_t$. The transition matrix $\boldsymbol{A}$ is defined as follows:

$$\boldsymbol{A}_t = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{t-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{t-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Note that, given the above process equation 3, the width and height of the bounding box are scaled by the scaling factor k of the previous time step, while its center changes position based on the pedestrian's velocity.

C. Observation Model

Once the particles have been processed, each state represents a hypothesis as to what the real state may be at time t. To weight this set of samples, each hypothetical target representation is compared to the latest target state $\boldsymbol{s}_t^T$. As mentioned earlier, the target pedestrian is represented by a 7×1 state vector that describes a rectangular bounding box. In order to quantify the similarity between an estimated particle state and target state, the regions of the image enclosed by the respective bounding boxes are compared. This is done using two different approaches: a comparison based on color distributions and a comparison based on computed moment invariants.

D. Color-based Measurement Model

Color-based image features are scale invariant and robust against out-of-plane rotations. To compare the color distribution of the target's bounding box with the distribution of an estimated state, the respective regions of the image are first converted to HSV (Hue, Saturation, Value) color space. Then, a $8 \times 8 \times 8$ bins HSV-histogram is computed, flattened and normalized, obtaining a 1×512 distribution vector for each enclosed area. Using a HSV color space makes our method less sensitive to illumination variations [22]. Once the histograms are computed, we calculate the *Hellinger distance* as similarity measure between target distribution vector $\vec{q}$ and estimated distribution vector $\vec{p}^{\,(n)}$:

$$d_{CLR}(\vec{q}, \vec{p}^{\,(n)}) = \sqrt{1 - \sum_{i=1}^{512} \sqrt{q_i \, p_i^{(n)}}} \tag{5}$$

where $d_{CLR} \in [0, 1]$. For two identical normalized HSV-histograms, we obtain $d_{CLR} = 0$, representing a perfect estimation of the target state.

E. Moment-based Measurement Model

As color-based tracking systems are sensitive to changes in lightning, robustness can be improved by introducing *Hu moments* that contain information about the shape boundary of an object. These image moments are independent of the chromatic content of the image and invariant to translations, rotations and scaling [14].

Similar as for the color-based measurement model, regions of the frame that are enclosed by estimated bounding boxes are compared to the target patch. To compute moment invariants for these regions, each patch is first transformed to grayscale. Then, a threshold is applied such that a binary image $f(x, y)$ is obtained that highlights the characteristic shape of the target pedestrian. An example result is shown in figure 2. Once the binary image is obtained, a moment of order $(a + b)$ can be computed using the following equation:

$$m_{ab} = \sum_x \sum_y x^a \, y^b \, f(x, y) \quad \text{for } a, b = 0, 1, 2, ... \tag{6}$$

To normalize for translations in the image plane, central moments are defined:

$$\begin{aligned} \mu_{ab} &= \sum_x \sum_y (x - \overline{x})^a \, (y - \overline{y})^b \, f(x, y) \\ \text{where} \quad \overline{x} &= \frac{m_{10}}{m_{00}}, \quad \overline{y} = \frac{m_{01}}{m_{00}} \end{aligned} \tag{7}$$

For the sake of scale invariance, the obtained moments are further normalized using the following equation:

$$\eta_{ab} = \frac{\mu_{ab}}{\mu_{00}^{\gamma}} \quad \text{with } \gamma = \frac{a + b}{2} + 1 \tag{8}$$

Fig. 2: A threshold is applied to the frame patch on the left to obtain the binary image on the right that highlights the pedestrian's characteristic shape

Based on the obtained central moments, it is possible to compute seven *Hu moments* that are invariant to object scale, position and orientation changes:

$$
\begin{aligned}
\phi 1 &= \eta_{20} + \eta_{02} \\
\phi 2 &= (\eta_{20} - \eta_{02})^2 + 4\,\eta_{11}^2 \\
\phi 3 &= (\eta_{30} - 3\,\eta_{12})^2 + (3\,\eta_{21} - \eta_{03})^2 \\
\phi 4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi 5 &= (\eta_{30} - 3\,\eta_{12})\,(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 - 3\,(\eta_{21} + \eta_{03})^2)] \\
&\quad + (3\,\eta_{21} - \eta_{03})\,(\eta_{21} + \eta_{03})\,[3\,(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi 6 &= (\eta_{20} - \eta_{02})\,[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\,\eta_{11}\,(\eta_{30} + \eta_{12})\,(\eta_{21} + \eta_{03}) \\
\phi 7 &= (3\,\eta_{21} - \eta_{03})\,(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 - 3\,(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} - 3\,\eta_{12})\,(\eta_{21} + \eta_{03})\,[3\,(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

Given these moment invariants, we can define a feature vector $\mathbf{\Phi}_T = [\phi_1^T, \phi_2^T, \ldots, \phi_7^T]$ for the target and for every estimated state $\mathbf{\Phi}_C^{(n)} = [\phi_1^C, \phi_2^C, \ldots, \phi_7^C]$. Looking at the values of the computed moment invariants, it can be seen that the moments differ highly in scale. A direct computation of the distance between $\mathbf{\Phi}_T$ and $\mathbf{\Phi}_C^{(n)}$ (e.g. using a euclidean or mahalanobis distance measurement) would, therefore, lead to a dominance of either very big or very small moments [30]. Hence, we need to standardize the values of the feature vectors when comparing the distance between target and candidate. Furthermore, we shift all values to the positive domain, such that $\phi_i^T,\ \phi_i^C \geq 0$ for $i = 1, 2, ..., 7$:

$$
\begin{aligned}
\tilde{\phi}_i^T &= \phi_i^T + \min(\min \mathbf{\Phi}_T, \min \mathbf{\Phi}_C^{(n)}) \\
\tilde{\phi}_i^C &= \phi_i^C + \min(\min \mathbf{\Phi}_T, \min \mathbf{\Phi}_C^{(n)}) \\
r_i &= abs\left(\frac{\tilde{\phi}_i^T - \tilde{\phi}_i^C}{\tilde{\phi}_i^T + \tilde{\phi}_i^C}\right)
\end{aligned}
\tag{10}
$$

The distance that measures the similarity between target and estimation is finally given by the following equation:

$$
d_{MMT} = \frac{1}{7}\sum_{i=1}^{7} r_i \tag{11}
$$

where a smaller value of d_{MMT} indicates bigger similarities between target and estimate. Respectively, a value of $d_{MMT} = 0$ indicates a perfect match.

F. Weight update

The computed distances of color- and moment-based model are used to calculate the likelihood of each particle of representing the correct target state:

$$
p(\boldsymbol{z}_t|\boldsymbol{s}_t^{(n)}) = \frac{1}{\sqrt{2\pi}\,\sigma_{obs}} \exp\left(-\frac{(d_t^{(n)})^2}{2\,\sigma_{obs}^2}\right) \tag{12}
$$

where σ_{obs} denotes the standard deviation that represents the observation noise and $d_t^{(n)}$ describes the respective distance measure at time t. The weights are updated accordingly. Furthermore, we adaptively increase the standard deviation σ_{obs} when the confidence measurement of our detection method is below a certain threshold or more than one pedestrian is detected and we are not able to assign the correct state to the actual target.

G. Final State Estimation

Given the estimated states and updated weights of each particle, the final state estimation can be computed. Note that the set of samples represent a discrete approximation of a continuous state distribution. There exist several density extraction techniques to compute a continuous density function from the given particle set [34]. However, due to the high-dimensional state space at hand, these techniques are mostly infeasible or computationally too expensive. Therefore, we estimate the posterior as a weighted average of the particles' states:

$$
\boldsymbol{s}_t^{CLR/MMT} = \sum_{i=1}^{N} \boldsymbol{s}_t^{(n)} \cdot \pi_t^{(n)} \tag{13}
$$

While this technique avoids computationally expensive calculations, we need to mention that it could sometimes lead to misleading results, e.g. when the particle filter tries to represent a multimodal distribution or during the initialization phase. However, as the results in section IV show, despite these rare occasions, the approach is able to successfully track pedestrians in complex video sequences.

After the two final state estimates $\boldsymbol{s}_t^{CLR}$ and $\boldsymbol{s}_t^{MMT}$ are obtained using a color-based and moment-based observation model, a third state estimate is computed that represents a combination of the two former ones. The key idea of such an Integration of Color-based and Moment-based tracking system (ICM) is that it gains additional robustness by benefiting from the strengths of both methods. Therefore,

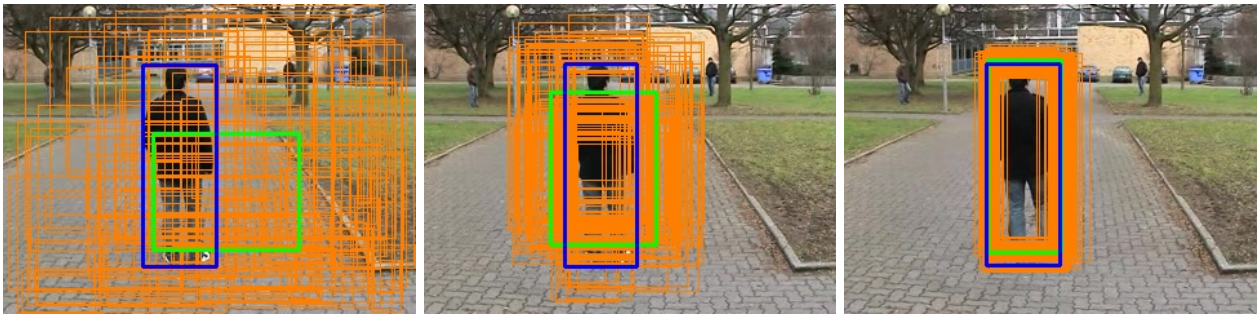


Fig. 3: Frames from the given video sequence showing the evolution of particles of the color-based tracking system. The target is represented by a blue bounding box, while the system state given by equation 13 is marked as green. The states of particles are represented as orange bounding boxes. In the image on the left, a set of particles with equal weights is randomly initialized according to a uniform distribution. As the video proceeds, particles are resampled according to their weights. After some iterations, it can be seen that most particles converged to the true state

scene-adaptive weights are computed that emphasize the more suitable method for the respective frame:

$$W_t^{CLR} = \frac{\exp(-\beta\, D_t^{CLR})}{\exp(-\beta\, D_t^{CLR}) + \exp(-\beta\, D_t^{MMT})}$$
$$W_t^{MMT} = \frac{\exp(-\beta\, D_t^{MMT})}{\exp(-\beta\, D_t^{CLR}) + \exp(-\beta\, D_t^{MMT})} \tag{14}$$

where D_t^{CLR} is the euclidean distance between the center points of the color-based estimated state's and target's state bounding box representation. D_t^{MMT} is the euclidean distance with respect to the moment-based estimated state, respectively. The value of β represents an attenuation constant. The fused state is finally given by:

$$\boldsymbol{s}_t^{ICM} = W_t^{CLR}\, \boldsymbol{s}_t^{CLR} + W_t^{MMT}\, \boldsymbol{s}_t^{MMT} \tag{15}$$

H. Resampling

The higher a particle's weight, the more likely it represents the true state of the target. After the weight update, particles that represent false states end up with negligible weights. The resampling step has the important function to force these particles back to the true posterior. It updates the set of samples by replacing particles with negligible weights by new particles in the proximity of particles with higher weights. Figure 3 illustrates this behavior. In the beginning, a set of particles with equal weights are randomly initialized. After the weights are updated, samples that represent the target state well are assigned higher weights than particles that fail in representing the target. Subsequently, the particles are resampled. As new frames are processed, particles with low weights are discarded, while more and more particles are generated in the proximity of the true state.

In the literature, there exists a variety of different resampling techniques [34]. For this work, we use a systematic resampling approach because of its favourable simplicity.

I. Target Update

During the video sequence, the appearance of the target pedestrian may change due to variations in illumination and out-of-plane rotations. To ensure robustness of our visual tracking system, the target is repeatedly updated. The detection method described in III-A returns a confidence level cf besides the target state, indicating how confident the pedestrian was detected. We use this confidence level to decide whether to update the target state or not. In particular, we only update the target state, if the confidence level exceeds a certain threshold td and the target pedestrian can be uniquely identified:

$$\boldsymbol{s}_t^T = \begin{cases} [x_t, y_t, v_x^t, v_y^t, H_x^t, H_y^t, k_t]^T, & \text{if } cf > td. \\ \boldsymbol{s}_{t-1}^T, & \text{otherwise.} \end{cases} \tag{16}$$

with

$$v_x^t = \frac{\Delta x}{dt} = \frac{x_t - x_{t-1}}{dt}$$
$$v_y^t = \frac{\Delta y}{dt} = \frac{y_t - y_{t-1}}{dt}$$
$$k_t = \frac{H_x^t\, H_y^t}{H_x^{t-1}\, H_y^{t-1}}$$

where (x_t, y_t) are the coordinates of the center of the target patch detected at time t and (v_x^t, v_y^t) denote the center point's velocity. H_x^t and H_y^t represent the rectangle's width and height and k_t the scale of the rectangle at time t with respect to the rectangle of the previous time instance $t-1$.

We further improve robustness by incorporating the estimated posterior state into the calculation of the target's feature vector when the target is not updated, i.e. when the confidence level is equal or less than the given threshold:

$$\boldsymbol{P}_t^T = \begin{cases} (1-\alpha)\, \boldsymbol{P}_t^T + \alpha\, \boldsymbol{P}_{t-1}^C, & \text{if } cf \leq td. \\ \boldsymbol{P}_t^T & \text{otherwise.} \end{cases} \tag{17}$$

where α defines the emphasis on the previous estimate, $\boldsymbol{P}_t^T$ denotes the feature vector based on the target state at time t and

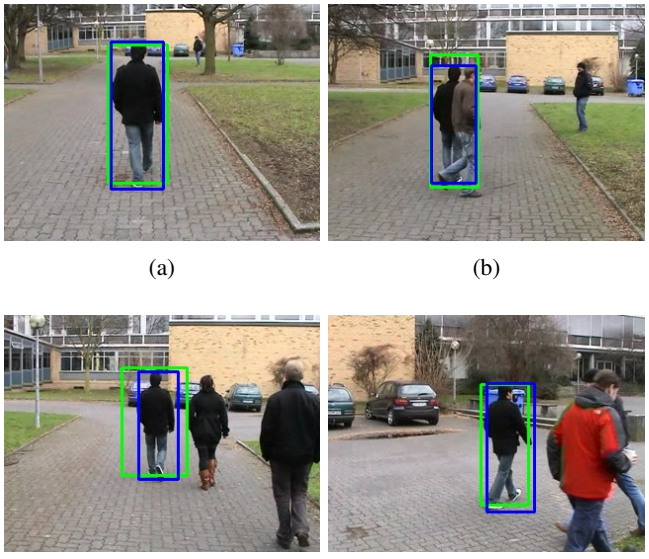
$\boldsymbol{P}_{t-1}^{C}$ represents the feature vector based on the estimated state at time $t-1$. The feature vectors are calculated as described in section III-D and III-E.

IV. Experiments

The implemented PF tracking systems have been tested on different video sequences from public datasets. In the following, we will present the results obtained for one particularly challenging dataset from the BoBoT benchmark on tracking datasets [6]. The implementation to reproduce the presented results and the dataset can be found at: https://github.com/PMMon/AppliedProject.

A. Dataset

The research goal of this study is to compare different particle filtering approaches for pedestrian tracking. In particular, we want to analyze the *simplicity*, *adaptivity* and *robustness* of these methods. Therefore, we evaluate the implemented techniques on a sufficiently challenging dataset that allows such an analysis. Table I provides an overview of the dataset's configurations. As illustrated in figure 4, the video data shows a pedestrian walking along a pavement who is followed by a moving camera. During the video sequence, the target pedestrian is frequently crossed by other pedestrians, resulting in full occlusions of the target (cf. figure 4b). Furthermore, similar distractors, i.e. other pedestrians who look similar to the target pedestrian enter the scene and temporary walk besides the target (cf. figure 4c). At the end of the video sequence, the tracked pedestrian moves to his right and therefore is displayed from his side, resulting in a changed appearance (cf. figure 4d).



(a) (b)

(c) (d)

Fig. 4: Tracking results of the ICM tracking system. The blue bounding box denotes the ground truth, while the green patch represents the estimated state. Different challenging scenarios for pedestrian tracking are shown, including occlusions, similar distractors and out-of-plane rotations.

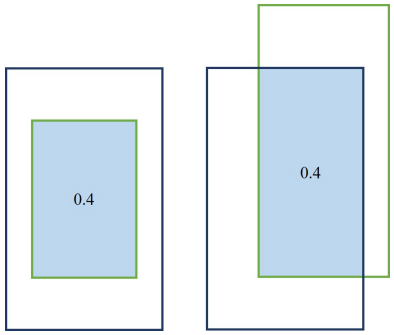


Fig. 5: Two different configurations resulting in an overlap of $E^{OE} = 0.4$. The dark blue rectangle represents the ground truth, while the green bounding box denotes the estimated patch. The light blue shaded area defines the intersection of both boxes.

The described characteristics of the dataset introduce various challenges in visual tracking. Therefore, the dataset at hand is suitable for our comparison.

B. Evaluation Metrics

In order to evaluate and compare the introduced tracking systems on the given dataset, we need to define expressive evaluation metrics. Therefore, we compute the following two error measures for every frame of the video sequence:

1) Euclidean Error: The first metric that comes to mind is the euclidean distance between the center point of the bounding box denoting ground truth and the center point of the estimated patch. The euclidean error is calculated as follows:

$$E_t^{L2} = ||X_t^{GT} - X_t^{EST}||_2 \tag{18}$$

where $X_t^{GT} = (x_t^{GT}, y_t^{GT})$ is the center point of the bounding box representing ground truth and $X_t^{EST} = (x_t^{EST}, y_t^{EST})$ is center point of the estimated patch at time t.

2) Overlap Rate: The euclidean error does not account for the size of the estimated bounding boxes. For example, a small rectangle can have the same center point as a large rectangle, resulting in the same euclidean error for both shapes. Therefore, we need to introduce an additional evaluation metric. The overlap accounts for the area of estimation and ground truth and is defined as follows:

$$E_t^{OE} = \frac{area(R_t^{EST} \cap R_t^{GT})}{area(R_t^{EST} \cup R_t^{GT})} \tag{19}$$

TABLE I: Properties of the dataset

Parameter	Setting
Format	320 × 240 at 25 fps
Size	1017 frames
Characteristics	Moving and non-rigid target, moving camera, out-of-plane rotations, full occlusions, similar distractors
Link to dataset	/dataset

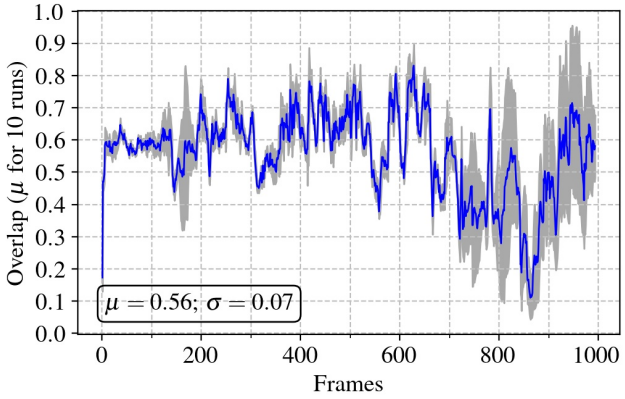
where R_t^{GT} is the bounding box representing ground truth and R_t^{EST} denotes the estimated patch at time t. Note that $E_t^{OE} \in [0, 1]$, where a perfect match would be represented by $E_t^{OE} = 1$. A complete mismatch is given by $E_t^{OE} = 0$. An example of the overlap rate can be found in figure 5.

C. Results

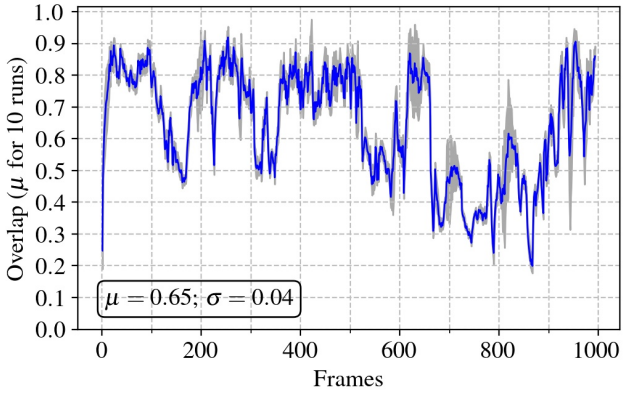
For the evaluation of our tracking systems, we run each particle filter approach ten times on the given dataset to account for the stochastic nature of the Sequential Monte Carlo method. Figure 6 illustrates the overlap rate for each frame of the video data. All tracking systems have been implemented using $N = 100$ particles. The mean value for every frame is displayed as a blue curve, while the standard deviation is indicated by the grey regions. As it can be seen, the color-based particle filter approach and the ICM tracker clearly outperform the moment-based technique. While color-based and ICM tracker achieve an overall overlap rate of 64% and 65%, respectively, the moment-based tracking system accomplishes only 56%. All techniques have difficulties in tracking the target pedestrian during the sequence starting from frame 700 to 900. During this part of the video, the target pedestrian is first fully occluded, then followed by two similar distractors (cf. 4c) and finally moves to his right, resulting in an out-of-plane rotation (cf. 4d). When looking at this sequence, we note that especially the performance of the moment-based tracking system is subject to strong deviations. This is due to the fact that for some runs, the tracking system loses track of the target and has difficulties in relocating itself to the true state. Hence, the moment-based approach is less robust than the other two approaches against the imposed challenges. Despite the fact that the color-based and ICM tracker perform worse compared to the rest of the given video data, they are still able to achieve an overlap rate that is bigger than 20% for every frame of the video sequence. At the beginning of the video data, both methods oscillate around an overlap rate of 80%, reaching up to 95% in the proximity of frame 600. Table II shows the overall mean overlap rates for different parts of the video sequence. During the first third of the video, the target is crossed by two similar distractors and one other pedestrian, while during the second third of the video the target pedestrian is crossed by one similar distractor and two other pedestrians. The last third of the video is the most challenging sequence. The target pedestrian moves to his right and, therefore, changes appearance. Furthermore, he is crossed by seven pedestrians and is temporarily followed by two similar distractors. These challenges are reflected in the results shown in table II. All methods perform considerably better during the first two thirds of the video data. While the color-based method performs slightly better than the ICM tracker during the first third of the video, the ICM tracker

TABLE II: Overlap rates with standard deviation for different sequences of the video data. The best results for the respective sequence are marked as bold values

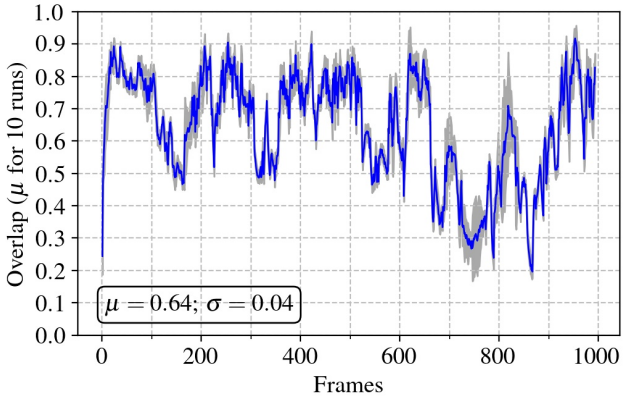
	Overlap 1st third	**Overlap 2nd third**	**Overlap 3rd third**
MMT	0.59 ± 0.04	0.64 ± 0.04	0.45 ± 0.14
CLR	**0.73 ± 0.03**	**0.70 ± 0.04**	0.51 ± 0.04
ICM	0.71 ± 0.04	**0.70 ± 0.04**	**0.52 ± 0.05**



(a) Moment-based particle filter approach



(b) Color-based particle filter approach



(c) ICM tracker

Fig. 6: Overlap rates of different particle filter approaches for pedestrian tracking based on the given video sequence. The grey region marks the standard deviation. Each tracking system was evaluated 10 times on the given dataset. The overall mean and standard deviation is displayed in the box on the bottom left of each graph

TABLE III: Euclidean errors with standard deviation for different sequences of the video data. The best results for the respective sequence are marked as bold values

	Euclidean Error 1st third	**Euclidean Error 2nd third**	**Euclidean Error 3rd third**	**Overall Euclidean Error**
MMT	9.74 ± 3.10	7.21 ± 2.01	22.10 ± 7.54	13.02 ± 4.22
CLR	6.73 ± 1.43	7.41 ± 1.71	14.40 ± 2.04	9.51 ± 1.73
ICM	**6.08 ± 1.27**	**5.90 ± 1.16**	**14.15 ± 2.76**	**8.71 ± 1.73**

marginally outperforms the color-based technique during the last third of the sequence. Both methods perform equivalent during the second third of the video data.

Table III illustrates the performance of each method with regards to the euclidean error. It can be seen that the ICM tracking system clearly outperforms both other particle approaches during all parts of the given video sequence. As opposed to the performance with regards to the overlap rate, the ICM tracker clearly benefits from the advantages of both color-based and moment-based tracking systems when it comes to the euclidean distance between the center points of the bounding boxes. This can be explained by looking at equation 14. The scene-adaptive weights that are used to calculate the fused state of the ICM tracker are computed based on the respective euclidean distances. Hence, the estimate that minimizes the euclidean error is emphasized when calculating the fused state. Furthermore, we note that the moment-based tracking system performs considerably better with regards to the euclidean error than with regards to the overlap rate. We note that during the second third of the video data, the moment-based particle filter approach even outperforms the color-based tracking technique. This difference from the performance with regards to the overlap rate can be explained by the fact that the size of the estimated patch does not significantly influence the invariant moments as long as the patch encloses the target pedestrian's shape. Hence, a mismatch between the bounding boxes' center points is penalized much stronger than a mismatch in area.

When looking at the simplicity of each method, color-based and moment-based tracking approaches only differ in the observation model of the Sequential Monte Carlo method, as described in section III-C. While the color-based tracking system utilizes HSV histograms, the moment-based approach computes seven moment invariants for each region of interest. An implementation for both types of feature vectors can be found in the OpenCV library. However, when implementing these methods ourselves, we find the color-based particle filter approach to be easier to implement. As the ICM tracker combines these two approaches, it can be seen as the most complex approach.

V. Conclusion

In this study, we have implemented and compared three different approaches to track pedestrians in complex video sequences using particle filters. We have related our work to existing research done in the field of visual object tracking and have explained the theoretical background of our particle filter approaches. The objective of this study was to compare the presented techniques with respect to their *simplicity*, *adaptivity* and *robustness*. For a rich and meaningful analysis, we evaluated each method on a challenging dataset from the BoBoT benchmark, including difficulties such as full occlusions, similar distractors, a moving camera and out-of-plane rotations. We have shown that the color-based and ICM tracking system perform almost equivalent with respect to the overlap rate, while both methods clearly outperform the moment-based technique. When looking at the euclidean error, the ICM tracker is able to benefit from both techniques and, therefore, outperforms the color-based and moment-based approach. Hence, we conclude that the ICM tracking system is the most robust and adaptive technique of the methods presented. While the moment-based and color-based particle filter approache are almost equivalently complex to implement, the combination of both in order to obtain the ICM tracking system is of course more challenging. Furthermore, the ICM tracker is computationally the most expensive technique. However, a strong advantage of Sequential Monte Carlo method is that it can be accelerated using parallel processing architectures. In the end, a tradeoff has to be made between accuracy and computational efficiency. Such a decision always has to be made based on the given objective.

Bibliography

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401.

[2] Y. Cai, N. Freitas, and J. Little. "Robust Visual Tracking for Multiple Targets". In: May 2006, pp. 107–118. ISBN: 978-3-540-33838-3. DOI: 10.1007/11744085_9.

[3] C.-J. Chang and M. Jazayeri. "Integration of speed and time for estimating time to contact". In: *Proceedings of the National Academy of Sciences* 115.12 (2018), E2879–E2887. ISSN: 0027-8424. DOI: 10.1073/pnas.1713316115.

[4] D. Comaniciu and P. Meer. "Mean shift analysis and applications". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1197–1203 vol.2. DOI: 10.1109/ICCV.1999.790416.

[5] D. Comaniciu and P. Meer. "Meer, P.: Mean shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603-619". In: *Pattern Analysis and*

Machine Intelligence, IEEE Transactions on 24 (June 2002), pp. 603–619. DOI: 10.1109/34.1000236.

[6] D. A. Klein. "Bobot - bonn benchmark on tracking". In: *Technical Report* (2010).

[7] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[8] A. Doucet, N. Freitas, and N. J. Gordon. *Sequential Monte-Carlo Methods in Practice*. Vol. 1. Jan. 2001. ISBN: 978-1-4419-2887-0. DOI: 10.1007/978-1-4757-3437-9.

[9] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung. *Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends*. 2019. arXiv: 1812.07368 [cs.CV].

[10] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504.

[11] N. Fridman, A. Zilka, and G. A. Kaminka. *The impact of cultural differences on crowd dynamics in pedestrian and evacuation domains*. Tech. rep. MAVERICK 2011/01. Bar Ilan University, Computer Science Department, 2011, pp. 18–23.

[12] C. Harris and M. Stephens. "A combined corner and edge detector". In: *In Proc. of Fourth Alvey Vision Conference*. 1988, pp. 147–151.

[13] T. Heap and D. Hogg. "Wormholes in shape space: tracking through discontinuous changes in shape". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 344–349. DOI: 10.1109/ICCV.1998.710741.

[14] M.-K. Hu. "Visual pattern recognition by moment invariants". In: *IRE Transactions on Information Theory* 8.2 (1962), pp. 179–187. DOI: 10.1109/TIT.1962.1057692.

[15] M. Isard and A. Blake. "Contour tracking by stochastic propagation of conditional density". In: *Computer Vision — ECCV '96*. Ed. by B. Buxton and R. Cipolla. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 343–356.

[16] D.-S. Jang and H.-I. Choi. "Active models for tracking moving objects". In: *Pattern Recognition* 33 (July 2000), pp. 1135–1146. DOI: 10.1016/S0031-3203(99)00100-4.

[17] D.-S. Jang, S.-W. Jang, and H.-I. Choi. "2D human body tracking with Structural Kalman filter". In: *Pattern Recognition* 35 (Oct. 2002), pp. 2041–2049. DOI: 10.1016/S0031-3203(01)00201-1.

[18] Jianbo Shi and J. Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. DOI: 10.1109/34.868688.

[19] Jianbo Shi and Tomasi. "Good features to track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[20] G. Junxiang, Z. Tong, and L. Yong. "Face tracking using color histograms and moment invariants". In: *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*. 2009, pp. 519–523. DOI: 10.1109/ICBNMT.2009.5347867.

[21] J. S. Kulchandani and K. J. Dangarwala. "Moving object detection: Review of recent research trends". In: *2015 International Conference on Pervasive Computing (ICPC)*. 2015, pp. 1–5. DOI: 10.1109/PERVASIVE.2015.7087138.

[22] Z. Liu, W. Chen, Y. Zou, and C. Hu. "Regions of interest extraction based on HSV color space". In: *IEEE 10th International Conference on Industrial Informatics*. 2012, pp. 481–485. DOI: 10.1109/INDIN.2012.6301214.

[23] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94.

[24] J. MacCormick and A. Blake. "A probabilistic exclusion principle for tracking multiple objects". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 1. 1999, 572–578 vol.1. DOI: 10.1109/ICCV.1999.791275.

[25] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10 (2005), pp. 1615–1630. DOI: 10.1109/TPAMI.2005.188.

[26] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. "Background modeling and subtraction of dynamic scenes". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 1305–1312 vol.2. DOI: 10.1109/ICCV.2003.1238641.

[27] H. P. Moravec. "Visual Mapping by a Robot Rover". In: *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI'79. Tokyo, Japan: Morgan Kaufmann Publishers Inc., 1979, pp. 598–600. ISBN: 0934613478.

[28] N. M. Oliver, B. Rosario, and A. P. Pentland. "A Bayesian computer vision system for modeling human interactions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 831–843. DOI: 10.1109/34.868684.

[29] M. Owczarek, P. Barański, and P. Strumiłło. "Pedestrian tracking in video sequences: A particle filtering approach". In: *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2015, pp. 875–881. DOI: 10.15439/2015F158.

[30] M. T. Razali and B. J. Adznan. "Detection and Classification of Moving Object for Smart Vision Sensor". In: *2006 2nd International Conference on Information Communication Technologies*. Vol. 1. 2006, pp. 733–737. DOI: 10.1109/ICTTA.2006.1684463.

[31] G. Sapiro, R. Kimmel, and V. Caselles. “Geodesic active contours”. In: *Computer Vision, IEEE International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, June 1995, p. 694. DOI: 10.1109/ICCV.1995.466871.

[32] C. Stauffer and W. Grimson. “Learning Patterns of Activity Using Real-Time Tracking”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000), pp. 747–757.

[33] Y. Taekyu and K. Sukbum. “Tracking for moving object using invariant moment and particle filter”. In: *2008 27th Chinese Control Conference*. 2008, pp. 351–354. DOI: 10.1109/CHICC.2008.4605325.

[34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics.* Intelligent robotics and autonomous agents. MIT Press, 2005, pp. 85–116. ISBN: 978-0-262-20162-9.

[35] Viola, Jones, and Snow. “Detecting pedestrians using patterns of motion and appearance”. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 734–741 vol.2. DOI: 10.1109/ICCV.2003.1238422.

[36] S.-K. Weng, C.-M. Kuo, and S.-K. Tu. “Video object tracking using adaptive Kalman filter”. In: *Journal of Visual Communication and Image Representation* 17.6 (2006), pp. 1190–1208. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2006.03.004.

[37] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. “Recent advances and trends in visual tracking: A review”. In: *Neurocomputing* 74.18 (2011), pp. 3823–3831. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2011.07.024.

[38] A. Yilmaz, O. Javed, and M. Shah. “Object Tracking: A Survey”. In: *ACM Comput. Surv.* 38.4 (Dec. 2006), 13–es. ISSN: 0360-0300. DOI: 10.1145/1177352.1177355.

[39] S. K. Zhou, R. Chellappa, and B. Moghaddam. “Visual tracking and recognition using appearance-adaptive models in particle filters”. In: *IEEE Transactions on Image Processing* 13.11 (2004), pp. 1491–1506. DOI: 10.1109/TIP.2004.836152.