electronics

MDPI

Article

Adaptive Method for Modeling of Temporal Dependencies between Fields of Vision in Multi-Camera Surveillance Systems

Karol Lisowski * and Andrzej Czyżewski *

Faculty of Electronics, Telecommunications and Informatics, Multimedia Systems Department, Gdańsk University of Technology, ul. Narutowicza 11/12, 80-233 Gdańsk, Poland
* Correspondence: lisowski@multimed.org (K.L.); andcz@multimed.org (A.C.)

Abstract: A method of modeling the time of object transition between given pairs of cameras based on the Gaussian Mixture Model (GMM) is proposed in this article. Temporal dependencies modeling is a part of object re-identification based on the multi-camera experimental framework. The previously utilized Expectation-Maximization (EM) approach, requiring setting the number of mixtures arbitrarily as an input parameter, was extended with the algorithm that automatically adapts the model to statistical data. The probabilistic model was obtained by matching to the histogram of transition times between a particular pair of cameras. The proposed matching procedure uses a modified particle swarm optimization (mPSO). A way of using models of transition time in object re-identification is also presented. Experiments with the proposed method of modeling the transition time were carried out, and a comparison between previous and novel approach results are also presented, revealing that added swarms approximate normalized histograms very effectively. Moreover, the proposed swarm-based algorithm allows for modelling the same statistical data with a lower number of summands in GMM.

Keywords: re-identification; particle swarm optimization; multi-camera surveillance systems

check for updates

Citation: Lisowski, K.; Czyzewski, A. Adaptive Method for Modeling of Temporal Dependencies between Fields of Vision in Multi-Camera Surveillance Systems. *Electronics* **2021**, *10*, 1303. https://dx.doi.org/10.3390/electronics10111303

Academic Editor: Jun Liu

Received: 26 April 2021
Accepted: 27 May 2021
Published: 30 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Currently, the number of video cameras in public places is huge and is still increasing. Operators of surveillance systems cannot concentrate on many fields of view (FOVs) efficiently for long periods. Browsing through hundreds of hours of video data is time-consuming and arduous. In general, these are the main reasons that imply the need for automated tools for video data analysis to facilitate operating with multi-camera surveillance systems. One such useful tool is a re-identification method whose task is to connect many observations of one real object from multiple cameras. The issues mentioned above became the basis and the motivation for the authors' research to develop re-identification methods.

The re-identification is equal to tracking objects between non-overlapping FOVs. In this case, the location of an object that has disappeared from a particular camera can be described with a probability measure. Furthermore, the pair of linked observations from different cameras are also related to a certain probability. This probability is determined on the basis of so-called premises such as an appearance (visual features) of the object in a certain pair of observations, a physical possibility of transition between the pair of cameras related to these observations, a time of this transition, and statistical patterns of objects movement on the observed area (called a behavior model). The scope of this paper considers mainly the issues related to obtaining the model of transition time based on statistical data. The authors formulate the conception of the modification of particle swarm optimization that use multiple swarms in order to match the Gaussian Mixture Model to statistical data with an adaptation of the number of mixtures in the model. Moreover, the comparison to the non-adaptive Expectation-Maximization method is also provided.

The paper is organized in the following way: Section 2 contains a short description of the research field and the authors' previous works. It also provides more details about

the experimental framework that performs the re-identification method, and explains the importance of time dependency modeling. In Section 3.5, the concept of transition time modeling is presented more precisely; Performed experiments and results are included in Section 5; the paper is concluded with Section 6 in which the two used methods are compared and the adaptability of the mPSO algorithm is discussed.

2. Related Works

Significant efforts were put into automated video data analysis. Such processing is performed as a compilation of many algorithms and methods that depend on each other. In general, video data processing consists of the following phases: background subtraction, object tracking, event detection and re-identification. The first step refers to distinguishing a moving object and a still background (also called background subtraction) [1–3]. Thus, the output of this step is the objects detected in a single FOV. The next phase, while the detected object is moving, is related to obtaining a connection between the same object detected in consecutive video frames. Results of this phase of video processing are trackers of the objects [4]. When object tracking has been performed, the directions of objects' movement and object trajectories (routes) can be determined within a single FOV. Having such metadata event detection methods can be applied. They are mostly based on rules related to the location or movement direction of an object tracker within a given FOV. Moreover, regions, so-called hot areas, are defined within a camera FOV, and they can be used as part of rules in the event detection methods [5]. At the output of the previous steps of video processing, the re-identification (inter-camera object tracking) methods can be performed. Events of an object entering or an object leaving, given hot areas, are essential for the re-identification algorithm because they are transformed into observations of objects, which provide the input meta-data for re-identification methods [6,7].

Many efforts were made to research the possibilities of tracking objects in non-overlapping video surveillance systems. Starting from a comparison of color histograms of objects [8], the methods of re-identification were developed for more sophisticated mechanisms. The main challenge posed with regards to these methods were related to changes in an object appearing in different cameras, which can be caused by unstable and changing lighting of the observed scene, as well as various settings of cameras (like the white balance). In order to cope with these problems, methods for the compensation of color or making visual descriptors independent to these factors were developed [9–14]. In general, these approaches to re-identification assume, on the one hand, utilizing additional data such as spatio-temporal dependencies or a kind of behavior model (particle filters [9], Bayesian networks [10], Markov chains [12], probability dispersion function [13] etc.), but on the other hand, some methods use the visual feature descriptors resistant to color changes [6,15]. The issue of discovering the topology of the camera network is considered in the literature related to tracking objects in multi-camera surveillance systems with non-overlapping cameras' FOVs [7,16]. Previous authors' works were focused on the methods for modeling the behavior of objects in the whole observed area, with their results being published earlier. This method is based on Pawlak's flow graphs and rough set theory [17,18]. Further works took into account the situation when the spatial dependencies or behavior of objects are changing. Thus, the adaptation method is proposed [19]. Moreover, the authors considered issues related to computational and memory complexity [20]. Therefore, a better adaptation of the Gaussian mixture model to statistical data related to the time of transition between adjacent cameras is essential in terms of saving processor cycles and memory. In this article, the authors focus on the issue of temporal dependencies between cameras. The proposed method is a modification of the particle swarm optimization algorithm [21]. The modification of this algorithm is related to the utilization of a multiple swarm approach in which the particular swarms compete with each other in order to find the best match of the Gaussian mixture model to the statistical data. Moreover, the mPSO algorithm also optimizes the number of summands in the Gaussian mixture model. As a reference, the EM (Expectation-Maximization) methodology is used [22]. It

MOST WIEDZY Downloaded from mostwiedzy.pl

was the previous approach to temporal dependencies modeling, which requires the a priori assumption of the number of Gaussian Mixture Model summands.

3. Methodology

The proposed approach is a part of a system for re-identification; therefore, a whole context is delineated. The forms of input and output, as well as data structures, are also described.

3.1. Re-Identification

As mentioned above, the re-identification algorithm is based on the observations. A single observation of an object contains elements that are used to obtain the necessary premises, such as a visual feature descriptor, location of the observation (the camera and the hot area identifiers) and timestamp. Essential for this method is the determination of a probability measure corresponding to the situation that a given pair of observations is related to the same object (also called the identity of the object). It is realized through computing the weighted sum of probabilities measuring object identity depending on the above-mentioned premises as is formulated in Equation (1):

$$\begin{aligned} P_{id}(O_{in}, O_{out}) = w_v \cdot P_v(O_{in}, O_{out}) + w_t \cdot P_t(O_{in}, O_{out}) \\ + w_b \cdot P_b(O_{in}, O_{out}), \end{aligned} \tag{1}$$

where O_{in} and O_{out} is a pair of observations related to the event of entrance into the FOV and the event of an exit from another FOV, respectively; $P_v(.)$,$P_t(.)$ and $P_b(.)$ are probabilities of object identity based on visual, temporal and behavioral premises, respectively; similarly, weights are expressed as w_v, w_t and w_b for visual, temporal and behavioral premises, respectively.

3.2. Spatio-Temporal Dependencies

At the beginning of the re-identification process, an important part of the algorithm is a limitation of the number of pairs of observations. It is obtained due to knowledge of the camera network topology, which is described with a graph. In such a graph, vertices refer to particular cameras and edges describe physically possible transitions between cameras. Having the topology graph, it is possible to determine which cameras are adjacent to the given one. Therefore, during the assessment of observation pairs, only pairs from adjacent cameras are considered.

The topology graph also contains temporal dependencies (description of times of transitions between adjacent pairs of cameras) described with weights on the edges. In order to determine temporal dependencies, statistical data related to transition times between particular pairs of cameras have to be gained. The creation of the transition time model is presented more precisely in the following part of the paper (Section 3.5).

3.3. Visual Features

Because of using video input data, visual object features cannot be omitted while comparing a pair of observations. Therefore, a region containing a silhouette of a detected object in a camera's FOV (also called a blob) have to be extracted for both observations. In order to assess a visual similarity of a certain pair of observations, visual feature descriptors (mostly color and texture) are calculated. Next, the descriptors are used for comparing the pair of observations and for determining their similarity, which describes the mentioned probability of the object identity (Equation (1)) from the point of view of the visual features.

Input data obtained directly from the multi-camera surveillance system have the form of a vector which contains times of transitions between the given pair of the cameras. This vector is the input for the EM algorithm, however the input for the modified particle swarm optimization (mPSO) algorithm is a histogram of transition time. The outcome of

MOST WIEDZY Downloaded from mostwiedzy.pl

the processing of both algorithms is the Gaussian Mixture Model, which determines the probability density function of transition time between a given pair of FOVs.

3.4. Input Data Description

The input for the EM-based algorithm is the vector mentioned above, which is described by the following formula:

$$\mathbf{s} = [s_1, \ldots, s_i \ldots, s_n], \tag{2}$$

where s_i is the value of i-th transition time in the vector $\mathbf{s}$ and n is the size of the vector $\mathbf{s}$.

The input for the (mPSO) algorithm can be described by two vectors $\mathbf{t}$ and $\mathbf{h}$ presented in Equations (3) and (5). The vector $\mathbf{t}$ determines the time resolution of the histogram as well as a minimum and maximum transition time:

$$\mathbf{t} = [t_1, \ldots, t_i \ldots, t_{max}], \tag{3}$$

where the difference between two consecutive elements describes the time resolution, $i = \{1, 2, \ldots, max\}$, t_1 and t_{max} are minimal and maximal transition time, respectively. Moreover, the value max is a number of intervals used to create the histogram. The vector $\mathbf{n}$ contains the numbers of transitions referring to particular times (elements) from the vector $\mathbf{t}$. Thus, the vector $\mathbf{n}$ is formulated as in Equation (4).

$$\mathbf{n} = [n_1, \ldots, n_i \ldots, n_{max-1}], \tag{4}$$

where n_i is a number of transitions that lasted from t_i to t_{i+1}. Next, the histogram $\mathbf{n}$ is normalized and in the following calculations this normalized histogram $\mathbf{h}$ is taken into consideration:

$$\mathbf{h} = [h_1, \ldots, h_i \ldots, h_{max-1}], \tag{5}$$

where $h_i = {}^{n_i}/_{\sum_{k=1}^{k=max-1} n_i}$.

3.5. Output Description—Transition Time Model

The transition time for each edge of the topology graph is described with the Gaussian Mixture Model. It is determined according to the following formula:

$$p_e(t) = \sum_{i=1}^{i=M} w_{ei} \cdot N(t|\mu_{ei}, \sigma_{ei}), \tag{6}$$

where $\sum_{i=1}^{i=M} w_{ei} = 1$; $p_e(t)$ express the probability of a given time of a transition t through the edge e; $N(t|\mu_{ei}, \sigma_{ei})$ determines the value of a normal distribution for the given time of transition; the parameters of the distribution are described with the mean value μ_{ei} and standard deviation σ_{ei}; w_{ei} is the weight assigned to a particular gaussian; M is the number of summands in the Gaussian Mixture Model.

4. Transition time Modeling

4.1. Expectation-Maximization Algorithm

The creation of the transition time model is an optimization task related to finding a Gaussian Mixture Model that is best fitted to the statistical data (in this case, in the form of time transition samples). Thus, having the samples of transition time (see Equation (2)), the parameters of the Gaussian Mixture Model have to be determined. This task can be realized with the Expectation-Maximization [22]algorithm, which is an iterative algorithm and consists of a few steps:

1. Initialization: initial values of parameters of the Gaussian Mixture Model are set:
   - M: number of summands in GMM ($j = 1, \ldots, M$)

MOST WIEDZY Downloaded from mostwiedzy.pl

- $w_j^{(0)}$: initial values of weights:

$$w_j^{(0)} = 1/M \tag{7}$$

- parameters of the Gaussians in the mixture μ_j, σ_j:

$$\sigma_j^{(0)} = \frac{1}{2 \cdot M} \cdot (t_{max} - t_1) \tag{8}$$

$$\mu_j^{(0)} = t_1 + (2 \cdot j - 1) \cdot \sigma_j^{(0)}. \tag{9}$$

Moreover, the initial value of log-likelihood is obtained:

$$L^{(0)} = \frac{1}{n} \sum_{i=1}^{i=n} log \left[\sum_{j=1}^{j=M} w_j^{(0)} N\left(s_i | \mu_j^{(0)}, \sigma_j^{(0)}\right) \right], \tag{10}$$

where n is the number of samples in the vector $\mathbf{s}$ and s_i determines the i-th value from samples vector $\mathbf{s}$ (see Equation (2)).

2. Expectation-step: so-called responsibilities $\rho_{ij}^{(k)}$ are computed for all samples from $\mathbf{s}$ (see Equation (2)) and for all summands of GMM:

$$\rho_{ij}^{(k)} = \frac{w_j^{(k)} N\left(s_i | \mu_j^{(k)}, \sigma_j^{(k)}\right)}{\sum_{l=1}^{l=M} w_l^{(k)} N\left(s_i | \mu_l^{(k)}, \sigma_l^{(k)}\right)}, \tag{11}$$

where (k) is the index of iteration of the algorithm; $i = \{1, \ldots, n\}$; $j = \{1, \ldots, M\}$; n is the number of samples in the input vector; M is the number of summands in the GMM and $\rho_{ij}^{(k)}$ measures how much each Gaussian in the mixture is responsible for each sample from the input. Next, the responsibilities of each summand are also computed:

$$resp_j^{(k)} = \sum_{i=1}^{i=n} \rho_{ij}^{(k)}, \tag{12}$$

where $j = \{1, \ldots, M\}$ and $resp_j^{(k)}$ describe the degree of j-th summand responsibility of values contained in input vector $\mathbf{s}$ (See Equation (2)).

3. Maximization-step: new values of Gaussian Mixture Model parameters are calculated for each of the GMM summands:

$$w_j^{(k+1)} = \frac{resp_j^{(k)}}{n} \tag{13}$$

$$\mu_j^{(k+1)} = \frac{1}{resp_j^{(k)}} \sum_{i=1}^{i=n} \rho_{ij}^{(k)} s_i \tag{14}$$

$$\sigma_j^{(k+1)} = \sqrt{\frac{1}{resp_j^{(k)}} \sum_{i=1}^{i=n} \rho_{ij}^{(k)} \left(s_i - \mu_j^{(k+1)}\right)^2}. \tag{15}$$

In Equations (13)–(15) $j = \{1, \ldots, M\}$ and $k+1$ is an index of the consecutive iteration of the algorithm.

MOST WIEDZY Downloaded from mostwiedzy.pl

4. Assessment of the actual Gaussian Mixture Model match: in order to determine how well the obtained GMM fits the histogram, the new log-likelihood has to be computed:

$$L^{(k+1)} = \frac{1}{n}\sum_{i=1}^{i=n} log\left[\sum_{j=1}^{j=M} w_j^{(k+1)} N(.)\right], \tag{16}$$

where $N(.) = N\left(s_i|\mu_j^{(k+1)}, \sigma_j^{(k+1)}\right)$. The ending condition of the algorithm is described with the following formula:

$$\left|L^{(k+1)} - L^{(k)}\right| > \delta, \tag{17}$$

where the value of δ determines the threshold, which refers to the precision of the GMM matching the distribution of values contained in the vector **s**. After the ending condition is checked, there are two possibilities:

(a) when the condition (17) is **true**, the algorithm must **return to step 2**;
(b) otherwise, the algorithm ends.

Thus, the input of this algorithm is the vector of transition times **s** and fitness threshold δ and the output is the parameters of the Gaussian Mixture Model. In Section 5, the experiments and results of using this algorithm are presented.

4.2. Modified Particle Swarm Optimization Algorithm

The second approach to the raised optimization task is the utilization of the swarm algorithm with some modifications. In PSO methods [23,24], each particle describes one probable solution of the optimization problem. A set of particles is distributed in the N-dimensional space and should be considered a swarm, which can realize a kind of inside communication [25,26]. Therefore, in each iteration of the algorithm, a new position of each particle (in the solution space) is calculated. In the classical approach, three vector components determine the direction and distance of the particle movement. These components are:

- **inertia component:** it contains information about the actual direction and speed of a particle movement;
- **global component:** is it is related to the best solution known in the whole set of particles; when one of particle finds the best solution the rest of them tend toward this solution;
- **cognitive component:** it refers to the experience form the past of a particle; this component is directed to the best solution known by the particular particle.

The steps of the Particle Swarm Optimization algorithm are presented below:

1. **Initialization:** an initial position $\mathbf{w}_i^{(0)}$ and a velocity $\mathbf{v}_i^{(0)}$ of i-th particle is set randomly, where $i = \{1, \ldots, P\}$ and P is the number of particles in the swarm.
2. **Assessment of the particles:** the assessment function $f\left(\mathbf{w}_i^{(k)}\right) : \mathbb{R}^n \to \mathbb{R}$, as arguments, takes parameters of a possible solution represented by a given particle, the result of this function is the estimation of the solution correctness in the form of a real value.
3. **Update the global and local best solutions:** updating of the best solution known by the i-th particle $\mathbf{l}_i^{(k)}$ (needed while calculating the cognitive component of the i-th particle) and updating the global best solution $\mathbf{g}^{(k)}$ (which is used to compute the global component) is performed as follows:
   - for each i-th particle where $i = \{1, \ldots, P\}$,
     - the following condition has to be considered: if $\left(f\left(\mathbf{l}_i^{(k-1)}\right) < f\left(\mathbf{w}_i^{(k)}\right)\right)$, then a new local best solution for the i-th particle is assigned: $\mathbf{l}_i^{(k)} = \mathbf{w}_i^{(k)}$,

MOST WIEDZY Downloaded from mostwiedzy.pl

otherwise it is not changed: $\mathbf{l}_i^{(k)} = \mathbf{l}_i^{(k-1)}$. In the case of a just initialized algorithm the actual position of the i-th particle is assigned as the best local solution: $\mathbf{l}_i^{(0)} = \mathbf{w}_i^{(0)}$;

- next, in order to determine the best global solution $\mathbf{g}^{(k)}$, the following conditions have to be considered: if $\left(f\left(\mathbf{g}^{(k)}\right) < f\left(\mathbf{l}_i^{(k)}\right)\right)$, then the new best global solution is assigned: $\mathbf{g}^{(k)} = \mathbf{l}_i^{(k)}$, otherwise $\mathbf{g}^{(k)} = \mathbf{g}^{(k-1)}$. The solution $\mathbf{g}^{(0)}$ is determined as the position of the best-assessed particle after initialization.

4. **Obtaining particles movements:** the new translation vectors of particles in the solution space are obtained as it is formulated in Equations (18):

$$\begin{aligned} \mathbf{w}_i^{(k+1)} &= \mathbf{w}_i^{(k)} + \mathbf{v}_i^{(k+1)} \\ \mathbf{v}_i^{(k+1)} &= \mathbf{v}_i^{(k)} + c_1 r_1^{(k)} \cdot \left(\mathbf{g}^{(k)} - \mathbf{w}_i^{(k)}\right) \\ &\quad + c_2 r_2^{(k)} \cdot \left(\mathbf{l}_i^{(k)} - \mathbf{w}_i^{(k)}\right), \end{aligned} \tag{18}$$

where $\mathbf{w}_i^{(k+1)}$ is a new position of the i-th particle in the solution space, $\mathbf{w}_i^{(k)}$ is a position of the i-th particle in the previous iteration of the algorithm, $\mathbf{v}_i^{(k)}$ is the part of the actual inertia component of the i-th particle movement (can be considered as the vector of the speed of the particle), $\mathbf{g}^{(k)}$ is the position (in the solution space) of the best-assessed particle is the whole swarm, $\mathbf{l}_i^{(k)}$ is the best-assessed solution known by the $i-th$ particle, values $\{c_1, c_2\}$ are certain constants and values $\{r_1^{(k)}, r_2^{(k)}\}$ are random variables obtained for the actual iteration of the algorithm.

5. **Termination criterion check:** two types of termination criteria are used. The first is the number of iterations of the algorithm, and the second is a difference between solutions calculated in the consecutive iterations. This difference can be described with Formula (19).

$$d_{rms}^{(k)} = \sqrt{\frac{\sum_{i=1}^{1=n}\left(w_i^{(k)} - w_i^{(k-1)}\right)^2}{n}}. \tag{19}$$

Thus, the two threshold values must be determined as termination criteria that are: the threshold for difference between consecutive iterations of algorithm δ_{rms} and the limit of iteration number δ_{iter}. Furthermore, the following condition has to be considered: if $\left(d_{rms}^{(k)} < \delta_{rms} \vee k < \delta_{iter}\right)$, then **terminate** the algorithm, otherwise **return to step 2**.

In order to facilitate understanding the idea of the PSO algorithm, a method for calculating movement of a single particle is suggestively presented in Figure 1 (in this case a 2-dimensional solution space was considered).

MOST WIEDZY Downloaded from mostwiedzy.pl

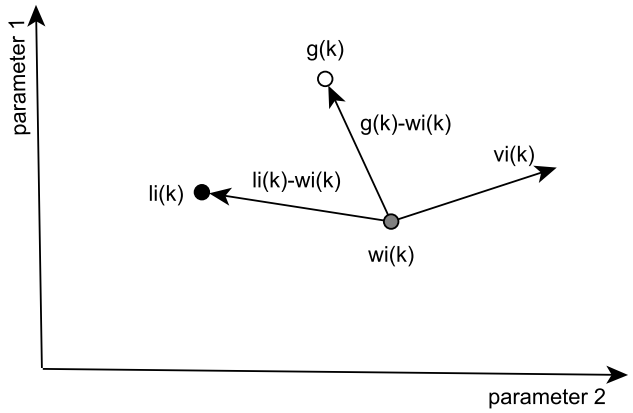

Figure 1. Example components of i-th particle movement used to obtain a location in the consecutive iteration of the algorithm in two-dimensional solution space.

In the case of transition time modeling, the solution space is one-dimensional, but some modifications must be introduced in the PSO algorithm. Each particle represents a certain transition time as its position $w_i^{(k)}$. In general, the proposed method assumes usage of the few particle swarms which will compete with each other in order to obtain parameters of the Gaussian Mixture Model. Each swarm is used to calculate the parameters of a single GMM summand. Therefore, there is a set of swarms $\mathbf{S} = S_1, \ldots, S_L$ and L is the number of swarms. The competition of them refers to the two mechanisms:

- the particles from the given swarm avoid being close to the particles of the other swarms, in order to find different mean values μ of model summands,
- and within a single swarm, the particles also avoid being too close to each other which allows the obtaining of a standard deviation value for a given summand of a GMM.

The next change is the introduction of the topology (within the swarm), which determines the adjacency of each particle. For the one-dimensional solution space, as in the case of the time transition modeling, the topology has the form of a chain, in which a particle has two neighbors, as is schematically presented in Figure 2. Such an approach modifies the method of calculation of the global component (in which the two closest particles are taken into consideration) $\mathbf{g}^{(k)}$ when the particles are moving (See Equation (18)).

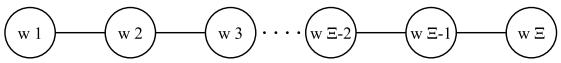

Figure 2. Swarm topology proposed for the one-dimensional solution space.

The vector **t** (containing transition times) also has to be modified in the way presented in Equation (20).

$$\mathbf{u} = [u_1, \ldots, u_i, \ldots u_{max}]$$
$$u_i = \frac{t_i - t_1}{t_{max} - t_1} \tag{20}$$
$$i = \{1, \ldots, max\},$$

where **u** is a vector of normalized transition time. Moreover, the parameter called particle satisfaction ϕ is introduced. This parameter is determined as Equation (21) shows.

$$\phi_i^{(k)} = \hat{g}\left(w_i^{(k)}\right) + \alpha d_{\mathbf{a}}(w_i^{(k)}) + \beta d_o(w_i^{(k)}), \tag{21}$$

MOST WIEDZY Downloaded from mostwiedzy.pl

where $\hat{g}\left(w_i^{(k)}\right)$ determines how high values of the histogram $\mathbf{h}$ are near the i-th particle, the value of $d_{\mathbf{a}}$ describes the average distance between the i-th particle and adjacent particles from the same swarm, d_o describes the average distance to the other swarms, and α, β constant coefficients.

Because the normalized transition times contained in $\mathbf{u}$ and the normalized histogram $\mathbf{h}$ contain discrete values, the interpolation needs to be applied. When the location of the i-th particle is determined as w_i, the four bins from the histogram $\mathbf{h}$ are chosen (the two are related to the closest transition times below w_i denoted as h_{low2}, h_{low1} and the other two refer to transition times above w_i denoted as h_{up1}, h_{up2}) as values of the function to interpolate. The arguments of function to interpolate are taken from the vector $\mathbf{u}$. Assuming that the four chosen values are $\hat{h}_0 = h_{low2}$, $\hat{h}_1 = h_{low1}$, $\hat{h}_2 = h_{up1}$ and $\hat{h}_3 = h_{up2}$, then values from the vector $\mathbf{u}$ are $\hat{u}_0 = u_{low2}$, $\hat{u}_1 = u_{low1}$, $\hat{u}_2 = u_{up1}$ and $\hat{u}_3 = u_{up2}$, respectively. It is worth mentioning that the value of w_i fulfills the condition $u_{low1} \leq w_i < u_{up1}$. In the proposed mPSO algorithm the Newton interpolation is used, which consists of the following steps:

1. Determine the points as the input for the interpolation algorithm:

$$\begin{aligned} g(\hat{u}_0) &= \hat{h}_0 \\ g(\hat{u}_1) &= \hat{h}_1 \\ g(\hat{u}_2) &= \hat{h}_2 \\ g(\hat{u}_3) &= \hat{h}_3, \end{aligned} \tag{22}$$

   where $g(.)$ is the function to interpolate
2. Estimate the $\hat{g}(\hat{u} = w_i)$ using the Newton interpolation formula (See Equation (24)).

$$g[\hat{u}_j, \hat{u}_{j+1}, \ldots, \hat{u}_{k-1}, \hat{u}_k] = \frac{g[\hat{u}_{j+1}, \ldots, \hat{u}_k] - g[\hat{u}_j, \ldots, \hat{u}_{k-1}]}{\hat{u}_k - \hat{u}_j} \tag{23}$$

$$\begin{aligned} \hat{g}(\hat{u}) = g[\hat{u}_0] + g[\hat{u}_0, \hat{u}_1](\hat{u} - \hat{u}_0) + g[\hat{u}_0, \hat{u}_1, \hat{u}_2](\hat{u} - \hat{u}_0)(\hat{u} - \hat{u}_1) + \\ + g[\hat{u}_0, \hat{u}_1, \hat{u}_2, \hat{u}_3](\hat{u} - \hat{u}_0)(\hat{u} - \hat{u}_1)(\hat{u} - \hat{u}_2) \end{aligned} \tag{24}$$

The second part of the particle satisfaction $\phi_i^{(k+1)}$ (see Equation (21)) is related to its avoidance of being close to its adjacency (called adjacency distance $d_{\mathbf{a}}$). Thus, $d_{\mathbf{a}}$ can be formulated as Equation (25) shows.

$$d_{\mathbf{a}}(w_i^{(k)}) = \sqrt{\frac{\sum_{\substack{\mathbf{j=1} \\ j \neq i}}^{j=\Xi_{\mathbf{a}}} \left(w_i^{(k)} - w_j^{(k)}\right)^2}{\Xi_{\mathbf{a}}}}, \tag{25}$$

where $\Xi_{\mathbf{a}}$ is the number of particles adjacent to the i-th particle, $\mathbf{a}$ describes the adjacency of the i-th particle (it is a vector containing the indices of adjacent particles).

The last part of the particle satisfaction $\phi_i^{(k+1)}$ (see Equation (21)) denoted as $d_o\left(w_i^{(k)}\right)$ describes a particle avoidance of the particles from other swarms.

$$d_o\left(w_i^{(k)}\right) = \sqrt{\frac{\sum_{\substack{\mathbf{j=1} \\ j \neq i}}^{j=L} \left(al_i^{(k)} - al_j^{(k)}\right)^2}{L}}, \tag{26}$$

where L is the number of swarms and $al_i^{(k)}$, $al_j^{(k)}$ are calculated in accordance with Equation (30).

The obtained value of $\phi_i^{(k+1)}$ is directly used to assess i-th particle in the algorithm (see the step "assessment of the particle") as a value of the assessment function $f(w_i^{(k)})$.

MOST WIEDZY Downloaded from mostwiedzy.pl

Additionally, a swarm satisfaction Φ is determined as Equation (27) presents:

$$\Phi_l^{(k)} = \frac{1}{\Xi} \sum_{\xi=1}^{\xi=\Xi} \phi_\xi \left(w_i^{(k)} \right), \tag{27}$$

where Ξ is the number of particles in the l-th swarm, ξ is the particle index, k is the index of algorithm iteration, ϕ_ξ determines the ξ-th particle satisfaction and l is the index of the swarm.

The next modification is adding two components that have an influence on the movement of the particles that are:

- **inner reluctance component:** is introduced in order to realize the avoidance between particles within a single swarm (denoted as $ir_i^{(k)}$ for i-th particle in k-th iteration);
- **external avoidance component:** is responsible for the repulsion between whole swarms (denoted as $ea_i^{(k)}$ for i-th particle in k-th iteration).

Thus, the step of the algorithm called "Obtaining particles movements" (See Equation (18)) have to be changed in the following way:

$$\begin{aligned} w_i^{(k+1)} &= w_i^{(k)} + v_i^{(k+1)} \\ v_i^{(k+1)} &= v_i^{(k)} + c_1 r_1^{(k)} \cdot \left(g^{(k)} - w_i^{(k)} \right) \\ &+ c_2 r_2^{(k)} \cdot \left(l_i^{(k)} - w_i^{(k)} \right) - ir_i^{(k)} - ea_i^{(k)}. \end{aligned} \tag{28}$$

The value of $ir_i^{(k)}$ is calculated according to the following formula:

$$ir_i^{(k)} = \frac{1}{A} \sum_{\substack{a=1 \\ a \neq i}}^{a=A} \frac{\phi_i^{(k)} - \phi_a^{(k)}}{\left| w_i^{(k)} - w_a^{(k)} \right|}, \tag{29}$$

where A is the number of particles adjacent to i-th particle, a is the index of adjacent particles, $\phi_i^{(k)}$ is the satisfaction of i-th particle, $\phi_a^{(k)}$ is the satisfaction of an adjacent particle and $\left| w_i^{(k)} - w_a^{(k)} \right|$ determined a distance between i-th particle and the adjacent one. In order to calculate the value of $ea_i^{(k)}$, the average location of the particular swarm $al_l^{(k)}$ has to be described:

$$al_l^{(k)} = \frac{1}{\Xi} \sum_{\xi=1}^{\xi=\Xi} w_\xi^{(k)}, \tag{30}$$

where Ξ is the number of particles in the l-th swarm, ξ is the particle index, k is the index of algorithm iteration, $w_\xi^{(k)}$ determines the ξ-th particle location and l is the index of the swarm. Therefore, the value of external avoidance component $ea_i^{(k)}$ is calculated in the way shown in Equation (31):

$$ea_i^{(k)} = \sum_{\substack{l=1 \\ l \neq i}}^{l=L} \frac{\left(al_i^{(k)} - al_l^{(k)} \right) \cdot \Phi_l^{(k)}}{\sum_{j=1}^{j=L} \Phi_j^{(k)}}, \tag{31}$$

where L is the number of swarms, $al_i^{(k)}$ is the average location of the swarm, which the i-th particle belongs to.

Generally, the modified Particle Swarm Optimization algorithm is performed as it is presented in Figures 3 and 4.

The algorithm is described below in detail:

MOST WIEDZY Downloaded from mostwiedzy.pl

1. Initialize vectors **h**, **u** (according to Equations (5) and (20)) and set the number of swarms L.
2. While the index of the swarm l is lower than L:
   (a) initialize a new swarm S_l;
   (b) perform the modified Particle Swarm Optimization algorithm for the swarm S_l (the rest of previously added swarms remains unchanged);
   (c) increase the index l
3. Obtain the parameters of Gaussian Mixture Model basing on the distributions of particles in particular swarms (see Equation (33)).

The last step will be described below. The particular swarm is a distribution of particles on the axis of normalized transition time (See Equation (20)). The input vector **u** was normalized (see Equation (20)), therefore, inverse operation (See Equation (32)) needs to be performed before the whole algorithm output.

$$T[u] = u(t_{max} - t_1) + t1. \tag{32}$$

Thus, the parameters of a given GMM summand can be determined as follows:

$$\begin{aligned} \mu_l &= T[al_l] \\ \sigma_l &= \eta \cdot \{T[max(S_l)] - T[min(S_l)]\} \\ w_l &= \frac{\Phi_l}{\sum_{j=1}^{j=L} \Phi_j}, \end{aligned} \tag{33}$$

where $min(S_l)$, $max(S_l)$ are minimum and maximum a particle location within the whole S_l swarm, respectively, and L is a number of swarms.

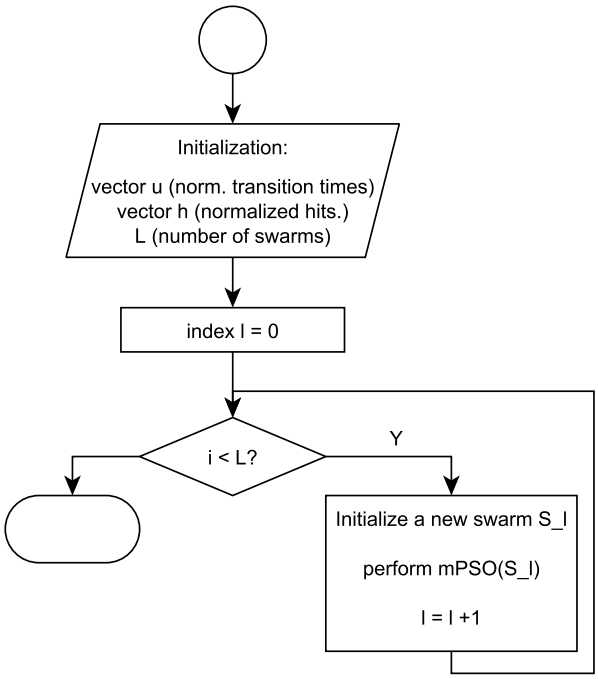

Figure 3. Flowchart of the whole modified Particle Swarm Optimization algoritm for all swarms. Operation '$mPSO(S_l)$' executes the algorithm presented in Figure 4.

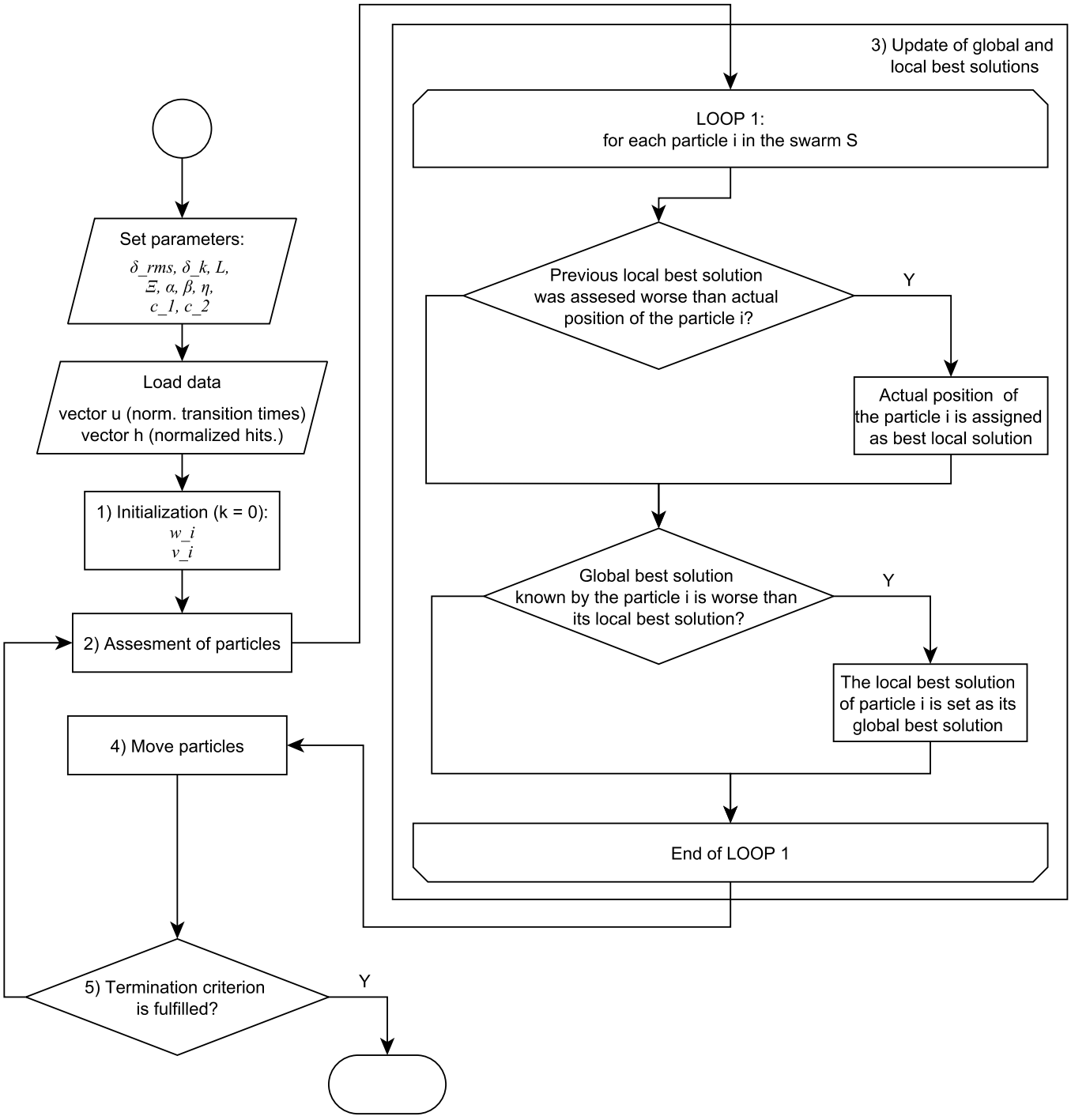

Figure 4. The flowchart presenting the algorithm of modified Particle Swarm Optimization for a single swarm.

5. Experiments and Results

Video data were recorded with the cameras (AXIS P1377) which were observing the road and the parking lot. The described optimization algorithms are implemented as modules of a video processing framework in C++ and run on a workstation computer. In reference to the mentioned re-identification method, the temporal dependency modeling has to be carried out as an initialization of the re-identification method. Thus, transition time models in the form of the Gaussian Mixture Model are used to determine the probability of object identity based on temporal premise $P_t(O_{in}, O_{out})$ (see Equation (1)). In the case of the EM algorithm, the number of summands in GMM has to be set a priori and

the results of Equations (13)–(15) from the last step of the algorithm are used to create the transition time model, whereas the mPSO algorithm can adapt the number of mixtures to the histogram data. Each swarm added by the modified Particle Swarm Optimization algorithm determines the parameters of a single Gaussian from the output GMM model (see Equations (33)).

In order to test both proposed algorithms, the two adjacent cameras from the video surveillance systems were chosen. Their FOVs are presented in Figures 5 and 6.

Figure 5. Field of vision of Camera A used in the experiments.

Figure 6. Field of vision of Camera b used in the experiments.

The transition time for each observation was determined as the time between the object's disappearance in the field of vision of the first camera and the moment of the object reappearance in the area observed by the second camera (see Figure 7). Next, 1028 samples of transition times **s** were obtained manually from 13.4 h of video data. These transition times were the input data for the presented algorithms (see Equation (2)). The histogram **n** and transition times vector **t** were also created (see Figure 8). Based on **n** and **t** vectors containing normalized values are obtained too (see Figure 9). Both histograms are shown in Figures 15 and 18.

MOST WIEDZY Downloaded from mostwiedzy.pl

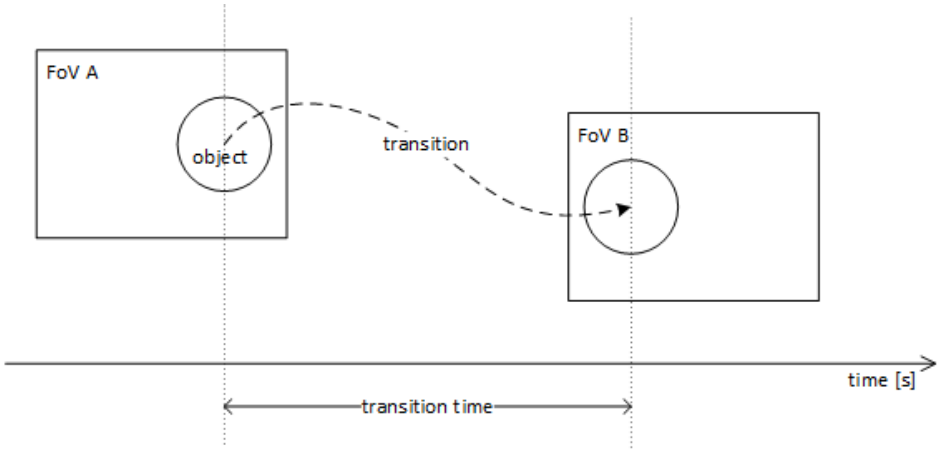

Figure 7. The transition time measurement.

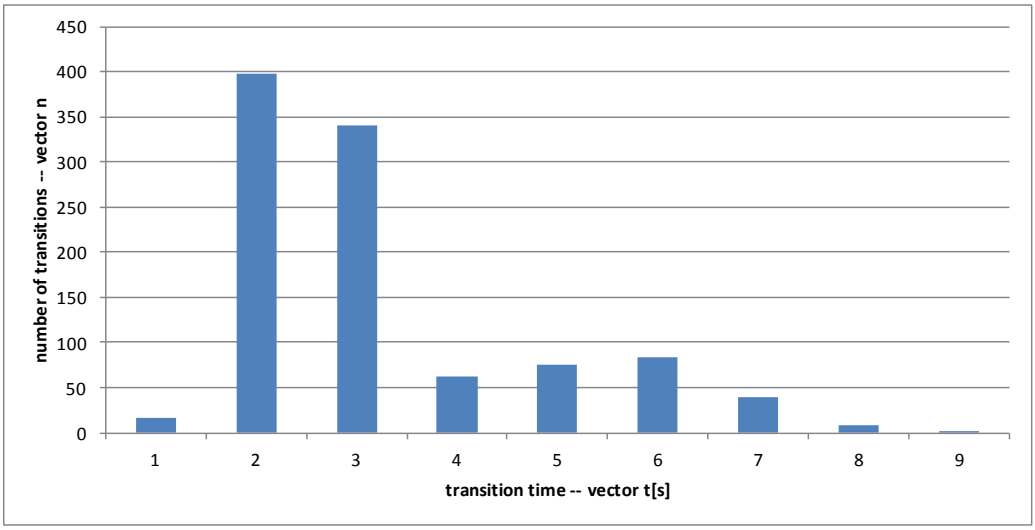

Figure 8. Histogram **n** of transition time.

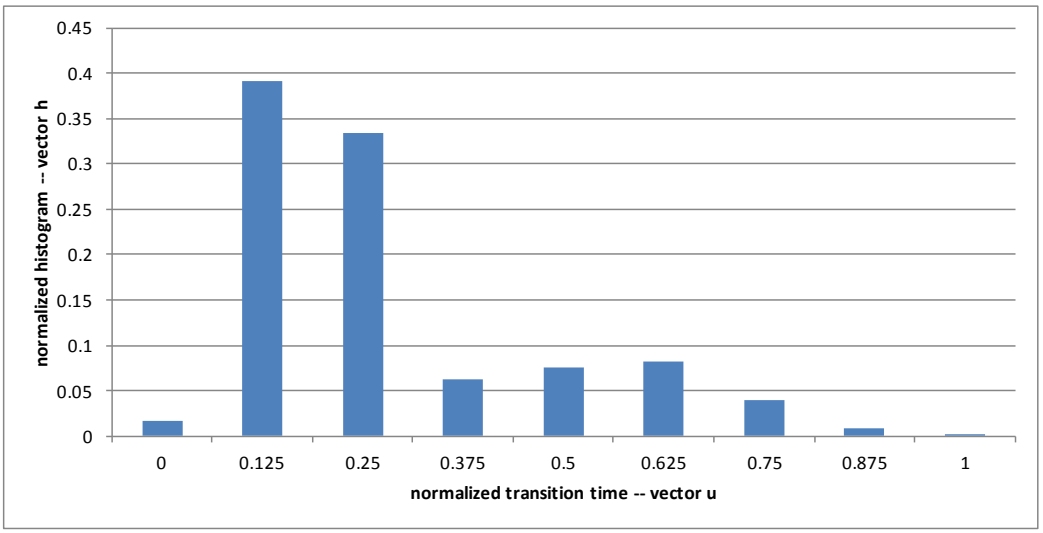

Figure 9. Normalized histogram **h** of transition time.

In order to use the normalized histogram in the modified particle swarm optimization, its values should be interpolated (see Figure 10). Interpolation operation is described

MOST WIEDZY Downloaded from mostwiedzy.pl

with the Equations (22)–(24). The interpolation error is determined as RMSE and is equal to 4.75%.

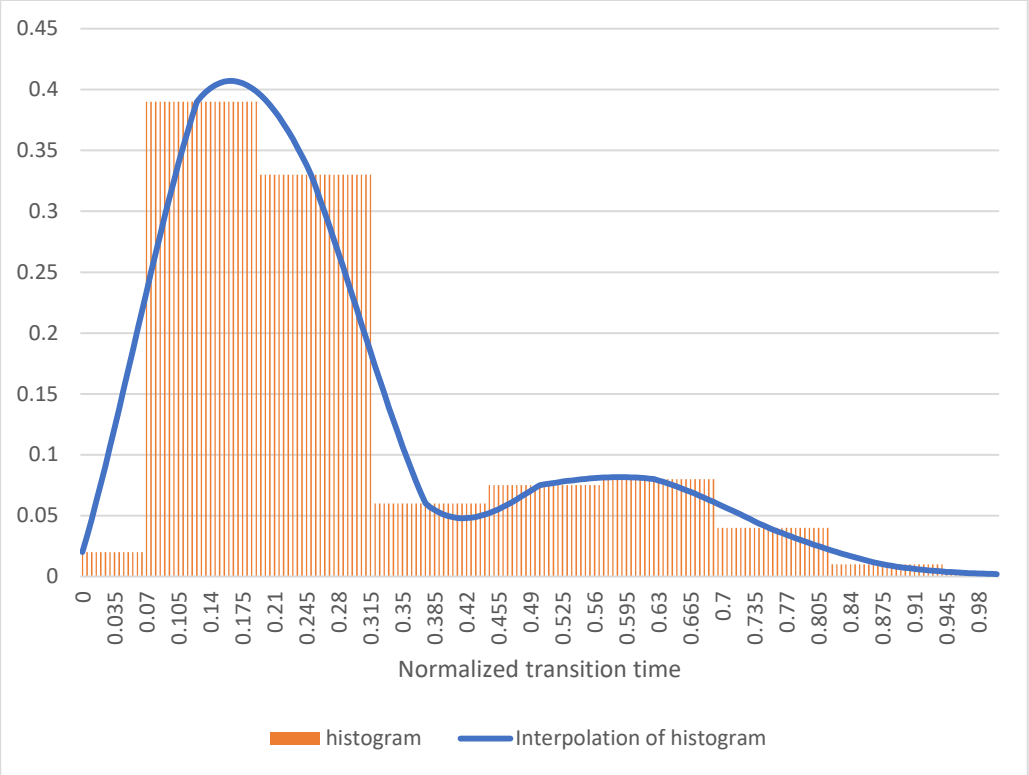

Figure 10. Interpolated values of normalized histogram **h**.

The EM method was tested with the following parameters: $M = 3, \delta = 0.0001$. Results provided by the EM algorithm in the consecutive steps is presented in Figures 11–15.

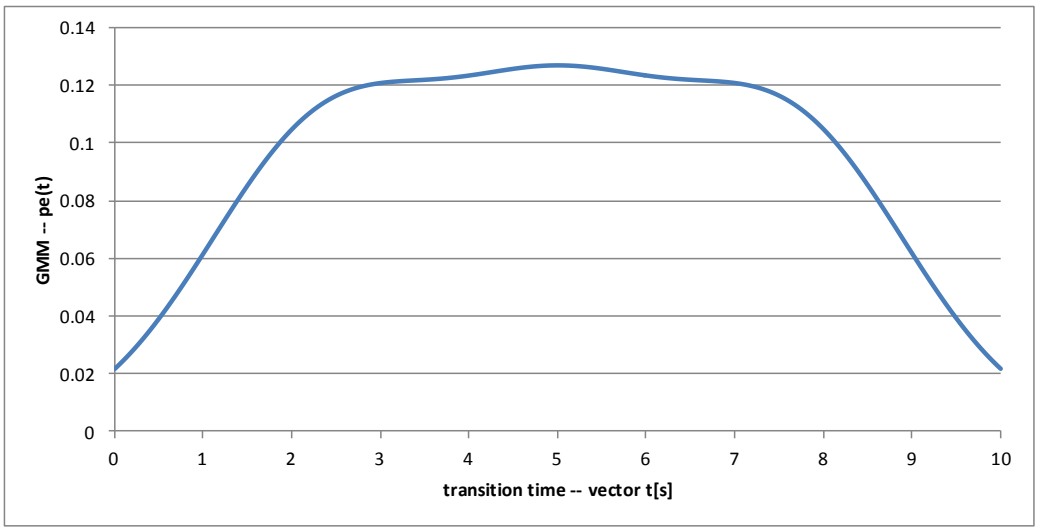

Figure 11. Just initialized GMM for the EM algorithm (iteration $k = 0$) of the EM algorithm.

MOST WIEDZY Downloaded from mostwiedzy.pl

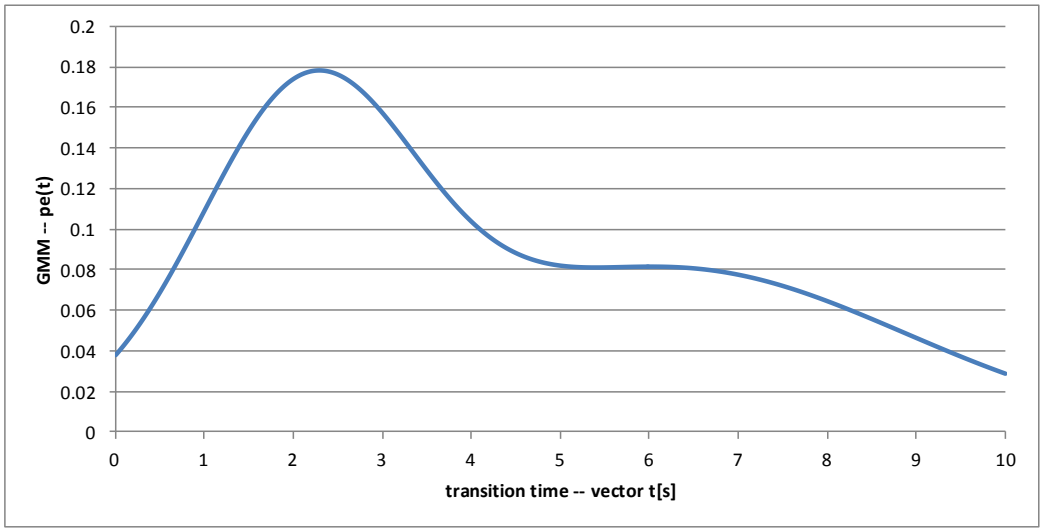

Figure 12. GMM after iteration $k = 5$ of the EM algorithm.

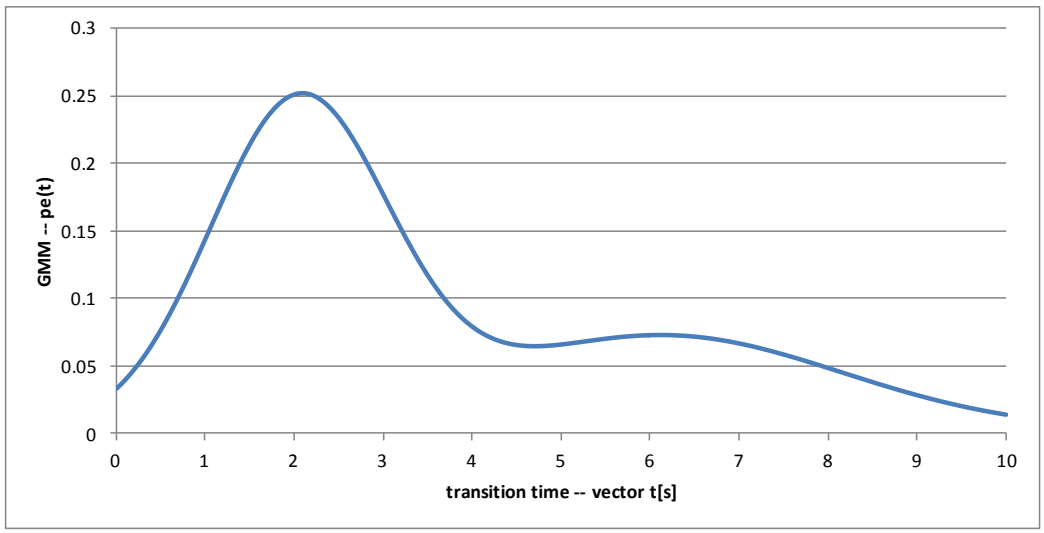

Figure 13. GMM after iteration $k = 10$ of the EM algorithm.

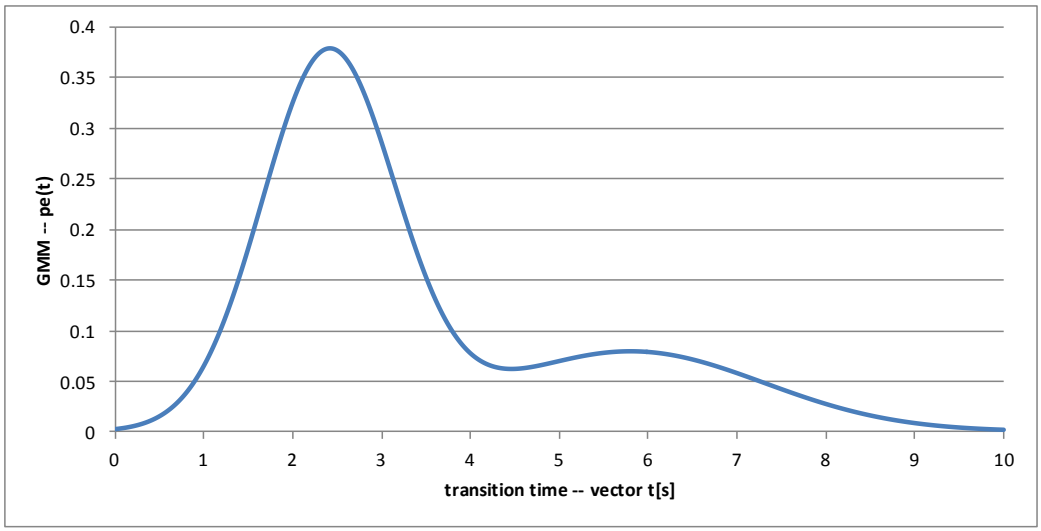

Figure 14. GMM after iteration $k = 15$ of the EM algorithm.

MOST WIEDZY Downloaded from mostwiedzy.pl

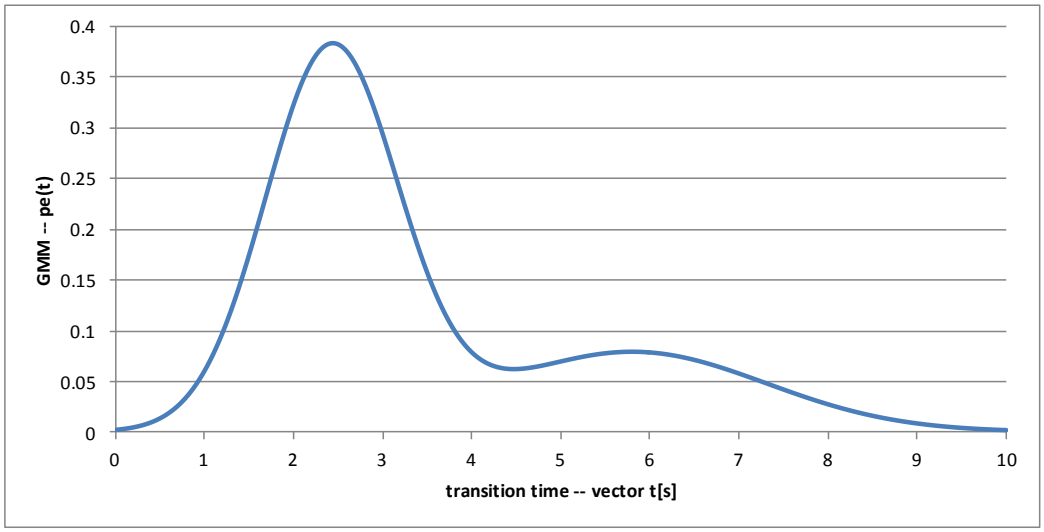

Figure 15. GMM after iteration $k = 17$ the EM algorithm ends because of termination criterion.

The proposed mPSO is performed with the following parameters: $\delta_{rms} = 0.0005$, $\delta_k = 1000$, $L = 3$, $\Xi = 150$, $\alpha = 0.015$, $\beta = 0.25$, $\eta = 0.7$, $c_1 = 0.15$, $c_2 = 0.15$. These results are also shown (See Figures 16–18). The output GMM parameters are tabulated in Table 1.

Table 1. Output GMM from EM and mPSO methods.

	EM			mPSO		
j	w_j	μ_j	σ_j	w_j	μ_j	σ_j
1	0.7012	2.4231	0.7429	0.7176	2.1304	0.76100
2	0.0022	5.0671	5.8210	0.1759	6.4303	1.2378
3	0.2966	5.8014	1.5019	0.1065	5.0610	5.9013

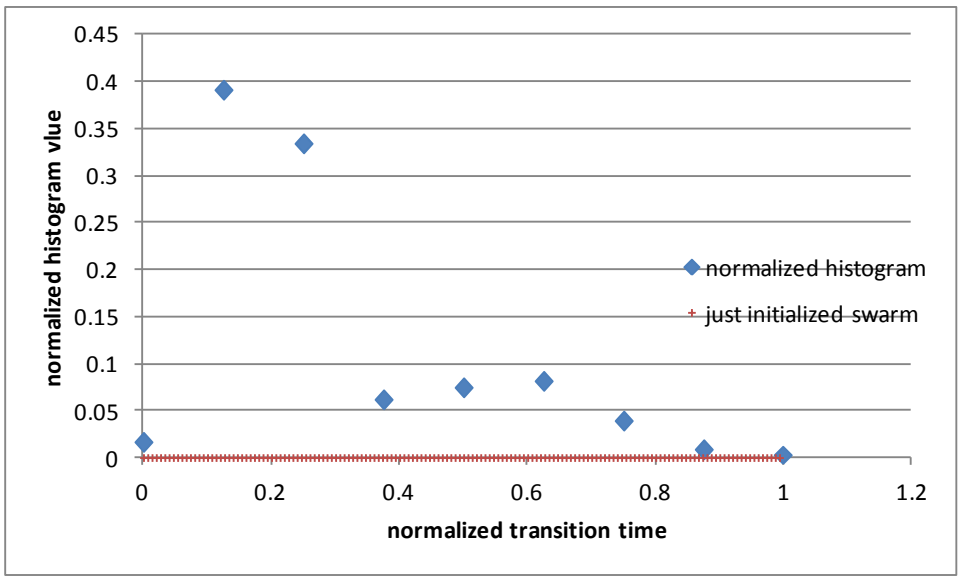

Figure 16. Example of just initialized swarm is distributed evenly.

MOST WIEDZY Downloaded from mostwiedzy.pl

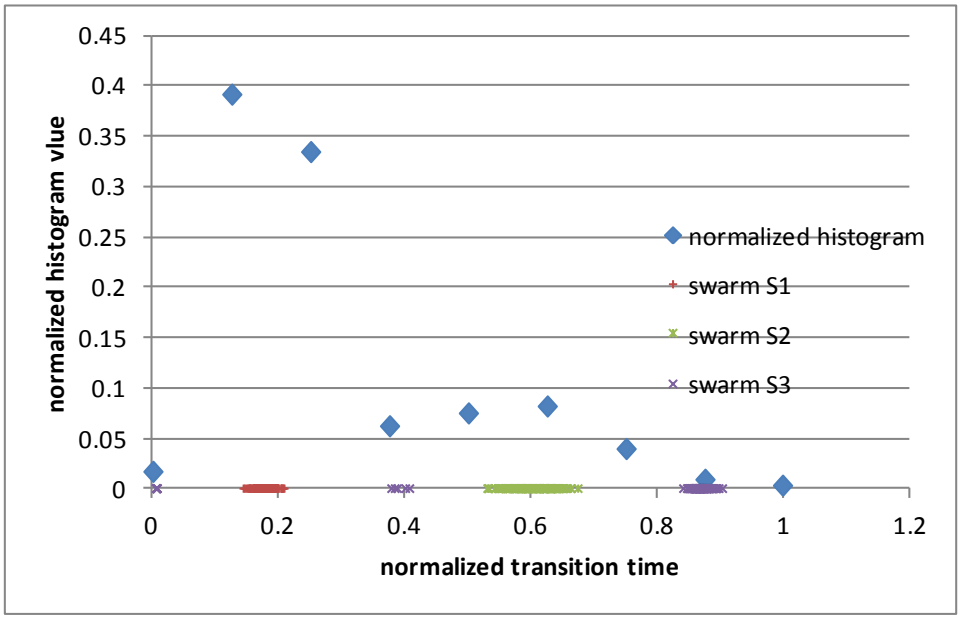

Figure 17. Normalized histogram and distribution of three different swarms when termination criterion for the last swarm S_3 is fulfilled.

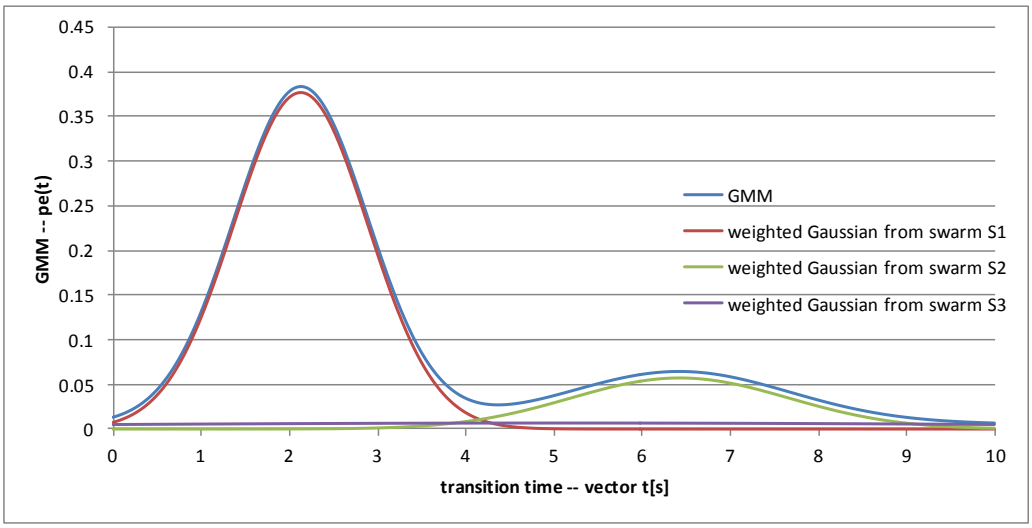

Figure 18. GMM obtained on the basis of swarms distributions.

6. Conclusions

Two approaches to object transition time modeling in multi-camera surveillance systems were developed and tested. The EM algorithm is based on iterative searching for the best fitness of the Gaussian Mixture Model to the transition time histogram. Both methods allow for conversion from statistically abundant data into a form of a Gaussian Mixture Model that is a quite compact representation of the probability density function. Moreover, an approximation histogram with the Gaussian Mixture Model enables the determination of probability for the continuous variable as an argument. However, in the EM method, the number of Gaussian Mixture Model summands has to be determined at the start of the algorithm. On the one hand, more mixtures should result in a better match to histogram data. On the other hand, additional mixtures cause an increment in computation consumption. This increment is related to the creation of the model as well as the usage of it. In the case of the modified Particle Swarm Optimization algorithm, the consecutive swarm can be added but this swarm will not change the Gaussian Mixture Model significantly, because previously added swarms took the most satisfying places of

MOST WIEDZY Downloaded from mostwiedzy.pl

the normalized histogram. Therefore, the swarm S_3 was insignificant in the experiment since it adds very little to the Gaussian Mixture Model. The proposed modified Particle Swarm Optimization algorithm allows not only for the match of Gaussian Mixture Model parameters to the histogram but at the same time it adjusts the number of mixtures in the Gaussian Mixture Model.

The future development of the proposed methods is related to testing the modified Particle Swarm Optimization in multi-dimensional solution space and checking other more complex topologies of the swarms. Moreover, computation complexity analysis of the modified Particle Swarm Optimization algorithm is also an interesting issue to pursue.

Author Contributions: Conceptualization, K.L. and A.C.; methodology, K.L.; software, K.L.; investigation, K.L.; writing—original draft preparation, A.C.; writing—review and editing, K.L.; visualization, K.L.; supervision, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the Polish National Science Centre within the grant belonging to the program "Preludium" No. 2014/15/N/ST6/04905 entitled: "Methods for design of the camera network topology aimed to re-identification and tracking objects on the basis of behavior modelling with the flow graph".

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GMM	Gaussian Mixture Model
EM	Expectation-Maximization
mPSO	modified particle swarm optimization
FOV	fields of view
List of Symbols	
P_i	probability of object identity
O_{in}	observation event related to entrance into area observed by camera
O_{out}	observation event related to disappearance of camera field of vision
P_v, P_t, P_b	component of the probability of object identity based on visual, temporal, and behavioral premises, respectively
w_v, w_t, w_b	weights used to determine the importance of given type of premises (visual, temporal, and behavioral, respectively)
$\mathbf{s}$	vector containing raw input data, that are transition times for all observed objects
s_i	time of the i-th observed transition
$\mathbf{t}$	vector of bin boundaries for the histogram of transition times
t_i	i-th bin boundary for the histogram of transition times
$\mathbf{n}$	vector containing number of transitions that are assigned to particular bins of the histogram of transition times
n_i	number of transitions that are assigned to i-th bin of the histogram of transition times
$\mathbf{h}$	vector containing number of transitions that are assigned to particular bins of the normalized histogram of transition times
h_i	number of transitions that are assigned to i-th bin of the normalized histogram of transition times
$p_e(t)$	probability density function (expressed as GMM) that estimate how probable is the object identity based on the time of transition between given pair of adjacent cameras e
w_{ei}	weight related to the i-th mixture and given pair of adjacent cameras e
$N()$	normal distribution

MOST WIEDZY Downloaded from mostwiedzy.pl

μ_i	mean value related to the i-th GMM mixture
σ_i	standard deviation related to the i-th GMM mixture
M	number of summands in the Gaussian Mixture Model
$.^{(k)}$	value of variable in k-th iteration of the algorithm ('.' can be substituted by any variable)
$resp_j$	degree of the j-th GMM summand responsibility of values contained in input vector $\mathbf{s}$
L	log-likelihood used in EM algorithm
δ	fitness threshold used in EM algorithm
$\mathbf{w}_i$	position of the i-th particle in the solution space
$\mathbf{w}_i$	velocity of the i-th particle in the solution space
P	is the number of the particles in the swarm
$f(\mathbf{w})$	the assessment function for the position in the solution space
$\mathbf{l}_i$	the best solution known by the i-th particle
$\mathbf{g}$	the best global solution
$\mathbf{u}$	vector of normalized transition time
ϕ	particle satisfaction
$\hat{g}(w_i)$	interpolated value of the histogram $\mathbf{h}$ for the position w of the i-th particle
$d_{\mathbf{a}(w_i)}$	average distance between the i-th particle and adjacent particles from the same swarm
d_o	average distance to the other swarms for the i-th particle
Φ	swarm satisfaction
ir	inner reluctance
al	average swarm location
ea	external avoidance

References

1. Czyżewski, A.; Szwoch, G.; Dalka, P.; Szczuko, P.; Ciarkowski, A.; Ellwart, D.; Merta, T.; Lopatka, K.; Kulasek, L.; Wolski, J. Multi-Stage Video Analysis Framework. In *Video Surveillance*; Weiyao, L., Ed.; Intech: Rijeka, Croatia, 2011; pp. 145–171. doi:10.5772/16088.
2. Szwoch, G. Performance Evaluation of the Parallel Codebook Algorithm for Background Subtraction in Video Stream. In *Multimedia Communications, Services and Security*; Dziech, A.; Czyżewski, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 149, pp. 149–157. doi:10.1007/978-3-642-21512-4_18.
3. Sobral, A.; Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput. Vis. Image Underst.* **2014**, *122*, 4–21, doi:10.1016/j.cviu.2013.12.005.
4. Czyżewski, A.; Dalka, P. Moving Object Detection and Tracking for the Purpose of Multimodal Surveillance System in Urban Areas. In *New Directions in Intelligent Interactive Multimedia*; Tsihrintzis, G., Virvou, M., Howlett, R., Jain, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 142, pp. 75–84. doi:10.1007/978-3-540-68127-4_8.
5. Dalka, P.; Szwoch, G.; Ciarkowski, A. Distributed Framework for Visual Event Detection in Parking Lot Area. In *Proceedings of the Multimedia Communications, Services and Security: 4th International Conference, MCSS 2011, Krakow, Poland, 2–3 June 2011*; Czyżewski, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 37–45. doi:10.1007/978-3-642-21512-4_5.
6. Dalka, P.; Ellwart, D.; Szwoch, G.; Lisowski, K.; Szczuko, P.; Czyżewski, A. Selection of Visual Descriptors for the Purpose of Multi-camera Object Re-Identification. In *Feature Selection for Data and Pattern Recognition*; Stańczyk, U., Jain, C.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 263–303. doi:10.1007/978-3-662-45620-0_12.
7. Radke, R.J. A Survey of Distributed Computer Vision Algorithms. In *Handbook of Ambient Intelligence and Smart Environments*; Nakashima, H., Aghajan, H., Augusto, J.C., Eds.; Springer: Boston, MA, USA, 2010; pp. 35–55. doi:10.1007/978-0-387-93808-0_2.
8. Cai, Y.; Chen, W.; Huang, K.; Tan, T. Continuously Tracking Objects Across Multiple Widely Separated Cameras. In *Proceedings of the Computer Vision—ACCV 2007, Tokyo, Japan, 18–22 November* 2007; Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 843–852.
9. Lev-Tov, A.; Moses, Y. Path Recovery of a Disappearing Target in a Large Network of Cameras. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, Atlanta, GA, USA, 31 August–4 September 2010*; Association for Computing Machinery: New York, NY, USA, 2010; pp. 57–64. doi:10.1145/1865987.1865997.
10. Javed, O.; Shafique, K.; Shah, M. Appearance modeling for tracking in multiple non-overlapping cameras. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June2005; Volume 2, pp. 26–33. doi:10.1109/CVPR.2005.71.
11. Cheng, Y.; Huang, C.; Fu, L. Multiple People Visual Tracking in a Multi-Camera System for Cluttered Environments. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 675–680. doi:10.1109/IROS.2006.282554.

MOST WIEDZY Downloaded from mostwiedzy.pl

12. Kim, H.; Romberg, J.; Wolf, W. Multi-camera tracking on a graph using Markov chain Monte Carlo. In Proceedings of the 2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Como, Italy, 30 August–2 September 2009; pp. 1–8. doi:10.1109/ICDSC.2009.5289352.
13. Colombo, A.; Orwell, J.; Velastin, S. Colour Constancy Techniques for Re-Recognition of Pedestrians from Multiple Surveillance Cameras. 2008. Available online: https://hal.inria.fr/inria-00326744/document (accessed on 29.05.2021).
14. Hamdoun, O.; Moutarde, F.; Stanciulescu, B.; Steux, B. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In Proceedings of the 2008 Second ACM/IEEE International Conference on Distributed Smart Cameras, Palo Alto, CA, USA, 7–11 September 2008; pp. 1–6.
15. Gilbert, A.; Bowden, R. Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity. In *Computer Vision—ECCV 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 125–136.
16. Loy, C.C.; Xiang, T.; Gong, S. Multi-camera activity correlation analysis. In Proceedings of the Computer Vision and Pattern Recognition, CVPR 2009, Miami, FL, USA, 20–21 June 2009; pp. 1988–1995. doi:10.1109/CVPR.2009.5206827.
17. Pawlak, Z. Decision algorithms and flow graphs: A rough set approach. *J. Telecommun. Inf. Technol.* **2003**, 3,98–101.
18. Czyżewski, A.; Lisowski, K. Adaptive Method of Adjusting Flowgraph for Route Reconstruction in Video Surveillance Systems. *Fundam. Inform.* **2013**, *127*, 1–16.
19. Czyżewski, A.; Lisowski, K. Employing flowgraphs for forward route reconstruction in video surveillance system. *J. Intell. Inf. Syst.* **2014**, *43*, 521–535.
20. Lisowski, K.; Czyzewski, A. Complexity Analysis of the Pawlak's Flowgraph Extension for Re-Identification in Multi-Camera Surveillance System. *Multimed. Tools Appl.* **2016**, *75*, 10495–10511, doi:10.1007/s11042-015-2652-z.
21. Poli, R. Analysis of the Publications on the Applications of Particle Swarm Optimisation. *J. Artif. Evol. App.* **2008**, *2008*, 4:1–4:10, doi:10.1155/2008/685175.
22. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–38.
23. Wang, X.; Yang, J.; Teng, X.; Xia, W.; Jensen, R. Feature selection based on rough sets and particle swarm optimization. *Pattern Recognit. Lett.* **2007**, *28*, 459–471, doi:10.1016/j.patrec.2006.09.003.
24. Omran, M.G.H.; Engelbrecht, A.P.; Salman, A. Particle Swarm Optimization for Pattern Recognition and Image Processing. In *Swarm Intelligence in Data Mining*; Abraham, A., Grosan, C., Ramos, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 125–151. doi:10.1007/978-3-540-34956-3_6.
25. Zemmal, N.; Azizi, N.; Sellami, M.; Cheriguene, S.; Ziani, A.; AlDwairi, M.; Dendani, N. Particle Swarm Optimization Based Swarm Intelligence for Active Learning Improvement: Application on Medical Data Classification. *Cogn. Comput.* **2020**, *12*, 991–1010, doi:10.1007/s12559-020-09739-z.
26. Banitalebi, A.; Aziz, M.I.A.; Bahar, A.; Aziz, Z.A. Enhanced Compact Artificial Bee Colony. *Inf. Sci.* **2015**, *298*, 491–511, doi:10.1016/j.ins.2014.12.015.

MOST WIEDZY Downloaded from mostwiedzy.pl