

Pawlak's flow graph extensions for video surveillance systems

Karol Lisowski*, Andrzej Czyżewski*

*Electronics, Telecommunications, and Informatics Faculty

Gdansk University of Technology, Narutowicza 11/12, PL80-233 Gdansk, Poland

Abstract—The idea of the Pawlak's flow graphs is applicable to many problems in various fields related to decision algorithms or data mining. The flow graphs can be used also in the video surveillance systems. Especially in distributed multi-camera systems which are problematic to be handled by human operators because of their limited perception. In such systems automated video analysis needs to be implemented. Important part of this analysis is tracking object within a single camera and between cameras' fields of vision. One of element needed to re-identify the single real object besides object's visual features and spatio-temporal dependencies between cameras is a behaviour model. The flow graph after some modifications, is a suitable data structure, which concept is based on the rough set theory, to contained as a behaviour model in it. Additionally, the flow graph can be used to predict the future movement of given object. In this paper a survey of authors research works related to employing flowgraphs in video surveillance systems is contained. The flow graph creation based on the paths of objects inside supervised area will presented. Moreover, a method of building a probability tree on the basis of the flow graph and a method for adapting the flowgraph to the changing topology of the camera network are also discussed.

I. INTRODUCTION

THE video surveillance systems have become common in public places and provided new possibilities (as well as challenges) in fields like security, crime prevention and automated video data processing. One of the main problems related to increasing number of cameras is that cameras' Fields of Vision (FOVs) do not overlap. In other words, there are locations which are not observed by any of the cameras. Thus the method for tracking object in a such adverse environment is needed. Therefore, this issue formed the basis of the authors' research work presented in this paper.

Tracking objects in a single camera is based on visual features of a moving object which differ from a background of a video image [1], [2]. Unfortunately such an approach is not suitable for the posed problem of re-identification of the same object in two different cameras. Therefore, some additional information related to statistical data analysis need to be obtained. There are two more types of premises (except of a comparison of visual features) which can be used to track objects more efficiently, that are:

- time of transition between given pair of the cameras
- probability that object will pass between a given pair of the cameras.

The first type of premise can be presented in the form of the weighted directed graph called topology graph. Edges of the

topology graph determine physical possibility of transitions between cameras and describe time of these transitions. The description of transition time can be in form of:

- a single value (like average time of transition)
- a probability density function (e.g. Gaussian)
- a model (approximation) of time transition time (e.g. with Gaussian Mixture Model)

The second type of the premise (that is behaviour model) can be also described with a weighted directed graph but in this case it is also acyclic graph called flow graph. The idea of flow graphs was introduced by Pawlak and is based on the rough set theory [3], [4]. This paper is focused on a presentation of methods related to behaviour modelling with the Pawlak's flow graphs based on the data from video surveillance system. Utilization of the presented modified flow graph corresponds to tracking objects between cameras with non-overlapping fields of vision (FOVs). A general aim of this paper is presentation of the survey of the authors' works on the implementation of the Pawlak's flowgraph (as behaviour model) in the video surveillance systems. Certain paths of objects through observed area are more frequent than others and some transitions are more probable. The behaviour model can be considered as a container for knowledge about these patterns. In order to perform tracking object between cameras, three types of premises can be used to re-identify a single real object: visual features, time of transition and probability of choosing particular transition. The last premise is contained in the behaviour model (in the flow graph). The idea of Pawlak's flow graph is consistent with the rough set theory and Bayes' theorem. Description and definition of the flow graph can be found in the literature [5], [6], [7]. The flowgraph can be utilized in context of data mining and decision tree building [4]. Additionally, the flowgraph idea can be employed in processing of musical meta data [8]. Extensions and modifications of the flowgraph were also introduced [9], [10], [11].

The paper begins with a short presentation of the idea of flowgraphs with accordance to the rough set theory (see Sec. II). Next, in Sec. III modifications and extensions are introduced that were needed to apply, in order to use them with metadata obtained from the analysis of video data from surveillance system. Sec. IV presents an application of the extended flow graph in surveillance system and describes consecutive steps of authors' research

work. The paper ends with summary and conclusion in Sec. V

II. PAWLAK'S FLOW GRAPHS AND ROUGH SET THEORY

In order to start a reflection on the flow graphs, some terms related to the rough set theory have to be presented according to literature [3]. Thus, a data set used in rough set theory is called an information system. The set of attributes denoted as A must be considered. Each attribute $a \in A$ may have values from a certain set V_a (called the domain). If two disjoint subsets of attributes (called conditions C and decisions D) are distinguished in the information system, then such a system becomes a decision system is denoted as:

$$S = (U, C, D), C \sqcup D \quad (1)$$

where U is called the universe, C is a set of condition attributes and D is a set of decision attributes.

Based on the Pawlak publication [12] definition of flow-graph and their properties will be presented below. The flowgraphs are actually a kind of data structure suitable for containing a distribution of information flow and to present statistical features of objects from the mentioned universe U . Such an approach enables a new possibility of statistical data analysis belonging to intelligent methods.

A flow graph can be considered as a directed acyclic graph:

$$G = (N, E, \varphi) \quad (2)$$

where N is a set of nodes, E is a set of edges ($E \subseteq N \times N$) and $\varphi : E \rightarrow (R^+ \cup \{0\})$ is a flow function. Moreover, the idea of flow graph assumes the following notations and terminology:

- (x, y) determines an edge with a node x as an input and a node y as an output, the edge (x, y) must be contained in the set E ;
- $I(x)$ is the set of all inputs of node x and $O(x)$ is the set of all outputs of node x , while $x \in N$;
- also output and input of the whole flowgraph G can be denoted as $I(G) = \{x \in N : I(x) = \emptyset\}$ and $O(G) = \{x \in N : O(x) = \emptyset\}$;
- input and output nodes are called external nodes and the rest of nodes are internal;
- $\varphi(x, y)$ is called throughflow from x to y which fulfils condition $\varphi(x, y) \neq 0$ for each edge (x, y) in the set E .

Thus, for each node x of a flow graph G , the inflow can be determined as:

$$\varphi_+(x) = \sum_{y \in I(x)} \varphi(y, x) \quad (3)$$

and the outflow can be defined as:

$$\varphi_-(x) = \sum_{y \in O(x)} \varphi(x, y) \quad (4)$$

In a similar way input and output of the whole flow graph can be formulated as:

$$\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x) \quad (5)$$

$$\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x) \quad (6)$$

Each internal node x fulfil the condition:

$$\varphi_+(x) = \varphi_-(x) = \varphi(x) \quad (7)$$

where $\varphi(x)$ is called a throughflow of node x . For the whole flow graph G the following formula is true:

$$\varphi_+(G) = \varphi_-(G) = \varphi(G) \quad (8)$$

where $\varphi(G)$ is a throughflow of the whole flow graph G . Hence, considering these assumptions all flows in the graph G can be normalized with the value of $\varphi(G)$ as is presented in the formulae:

$$\sigma(x, y) = \frac{\varphi(x, y)}{\varphi(G)}, \quad 0 \leq \sigma(x, y) \leq 1 \quad (9)$$

$$\sigma(x) = \frac{\varphi(x)}{\varphi(G)}, \quad 0 \leq \sigma(x) \leq 1 \quad (10)$$

The value of $\sigma(x, y)$ is called the strength of edge (x, y) and the value of $\sigma(x)$ is called the strength of node x .

Above defined normalized flows in the flowgraph allow for obtaining relative parameters assigned to the edges which are called certainty factor:

$$cer(x, y) = \frac{\sigma(x, y)}{\sigma(x)}, \quad \sigma(x) \neq 0 \quad (11)$$

and coverage factor:

$$cov(x, y) = \frac{\sigma(x, y)}{\sigma(y)}, \quad \sigma(y) \neq 0 \quad (12)$$

III. USING FLOWGRAPH FOR BEHAVIOUR MODELLING

In order to create behaviour model based on the flow graph a video surveillance system will be considered as in the previous paper [13]. Thus, a set of locations related to particular cameras is distributed on a certain area which can be described with the formula:

$$C = \{c_1, \dots, c_N\} \quad (13)$$

where c_i is camera with i index and N is a number of the cameras ($1 \leq i \leq N$). As it was mentioned a network of the cameras can be presented with the topology graph on which spatial dependencies between cameras are described (see Fig. 1)

Object which moved through the observed area creates a path which contains consecutive locations visited by the object. This path can be defined with the following formula:

$$p = \{(c_{id_1}, 1), \dots, (c_{id_L}, L)\} \quad (14)$$

where c_{id_1} and c_{id_L} corresponds to camera which is visited by the object at the entrance to the observed area and at the exit from this area, respectively; numbers from id_1 to id_L define consecutive values of the index i from Eq. 13. Thus, the set of paths can be described as presented by the following formula:

$$P = \{p_1, \dots, p_M\} \quad (15)$$

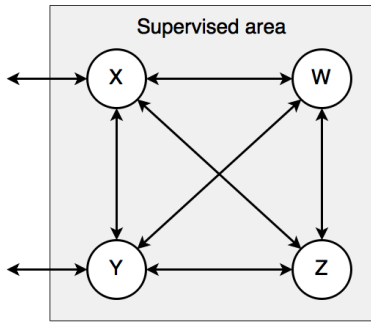


Fig. 1: The graph presenting topology of video surveillance system. According to Eq. 13: $C = \{w, x, y, z\}$

where M is the number of paths in the set. Having the input data prepared in this way, next steps of creating the behaviour model can be carried out. However, in the beginning, the attributes and the domains must be reconsidered. Hence, the attribute used in the flow graph contains two parts:

- the index (number) that describes an order of this element in its path,
- the label of camera in which an object appeared.

For example, the attribute X_1 means that the given object was observed for the first time (on entrance to the observed area) in the camera 'X'. The consecutive domains are determined with the index mentioned above. The example flow graph created on the basis of the set of paths is presented in Fig. 2. Moreover, the certainty and coverage factors are also used in a specific way. In order to realize above, also parameters of the flowgraph need to be redefined as the following formulae show:

$$\begin{aligned} \sigma(x_i, y_{i+1}) &= \frac{\varphi(x_i, y_{i+1})}{\varphi(G)} \\ \sigma(x_i) &= \frac{\varphi(x_i)}{\varphi(G)} \\ cer(x_i, y_{i+1}) &= \frac{\sigma(x_i, y_{i+1})}{\sigma(x_i)} \\ cov(x_{j-1}, y_j) &= \frac{\sigma(x_{j-1}, y_j)}{\sigma(y_j)} \end{aligned} \quad (16)$$

where $\sigma(x_i, y_{i+1})$ defines the rate of objects passing from the camera x in the step i of the path to the camera y in the next step $i + 1$ of the path, $\varphi(x_i, y_{i+1})$ determines the number of paths (in the set of path) which contain a transition from step x_i to step y_{i+1} , the total number of paths taken into consideration while building the flow graph is denoted as $\varphi(G)$, and $\varphi(x_i)$ is the number of paths in the set of paths which contains the step (flow graph's node) x_i . Also the values of the certainty and coverage change their meanings. The certainty (cer) estimates the conditional probability that the object which left the camera x in the step i of its path will appear in the camera y in the consecutive step $i + 1$, whereas the coverage (cov) determines estimation of the conditional probability that an object which appears in the camera y in the step j of the path was seen before, in the camera x in the previous step $j - 1$ of the path. The certainty cer is

used to predict future movements of the object whereas the coverage cov is useful in re-identification method. The cov is utilized during the decision-making related to identification of the single object observed in two different cameras.

IV. FLOWGRAPHS IN SURVEILLANCE SYSTEMS

The essential use case of the flow graph in the surveillance is related to prediction of object movements. The certainty factor estimates the probability of the future location that the object will visit, based on its previous route through the supervised area. Because of data concerning flows of objects through the observed locations, which is contained within the flow graph, probabilities of more than one location ahead can be predicted. As a result a probability tree is obtained. The formula which allows for creation of the probability tree (according to [14]) is shown in Eq. 17:

$$cer[x_{root}, \dots, z_{end}] = \prod_{i=root}^{i=end} cer(x_i, x_{i+1}) \quad (17)$$

where the root of probability tree is denoted as x_{root} , the probability of the path from the vertex x_{root} to z_{end} is calculated as a product of probabilities of subsequent steps in the given path. An example probability tree created on the basis of the flow graph presented in Fig. 2 is shown in Fig. 3. The probability tree is created on the basis of the particular instance of the flow graph. It presents possible future transitions of the object observed in the certain camera in determined step of the object path.

Another important issue is the changing environment which is observed with cameras. It causes changes in the topology of the camera network. Some transitions may become physically impossible or new transitions appear. In such fluctuating conditions the created behaviour model may quickly become out of date and it will contain incorrect transition probability estimates. The flow graph is a quite slowly updating structure so a dedicated method for speed up adaptation to new conditions is needed. In order to solve this problem, an adaptation method employing some additional modifications, according to [15] is necessary:

- each path (obtained from video surveillance system) before adding to the set of paths and being used to build a recent behaviour model are weighted by so called importance factor;
- the importance factor is based on probability of the occurrence of the same path in the past;
- two instances of flowgraph are created: the first called *core_* and the second called *recent_*
- measures of distance between two flow graphs also have to be introduced;
- values of two thresholds need to be determined: the first one called *learningThreshold_* is created in order to enable making decision that flow graph *recent_* is a proper model of unchanging object behaviour, and the second one called *adaptiveThreshold_* is used to determine a moment in which adaptation method must be

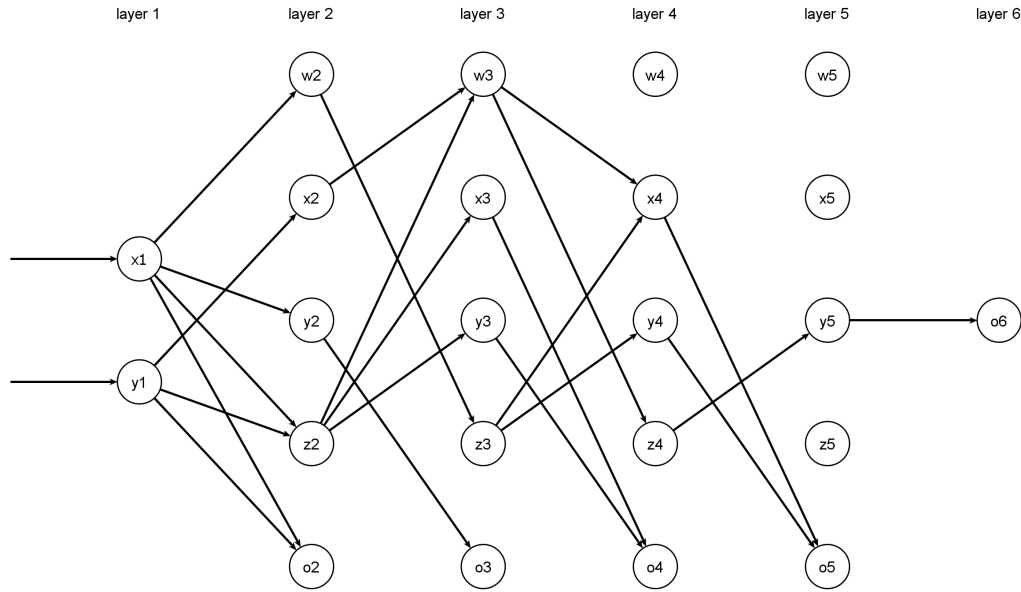


Fig. 2: The example flow graph obtained from the set of paths

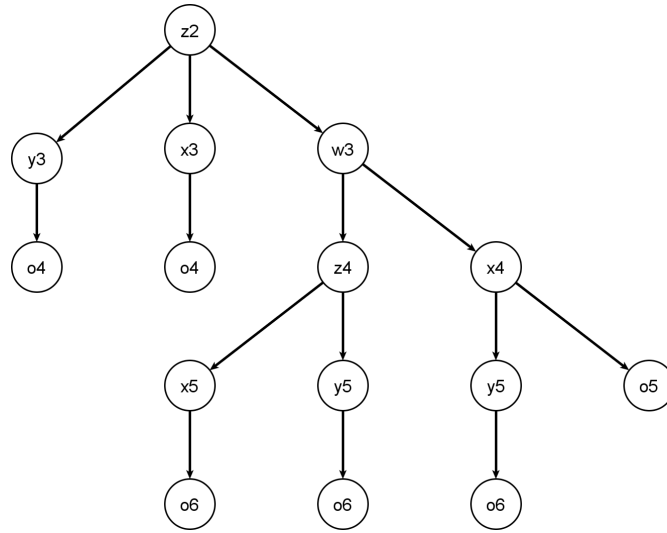


Fig. 3: Probability tree which begins in node Z_2 (what means that $x_{root} = Z_2$)

performed (when distance between *core_* and *recent_* is too large).

The adaptation method requires that some paths will be considered as more important ones than single paths. Moreover, a measure of importance of the path *imp* needs to be introduced. In case of normal adding path to the flow graph, the importance factor *imp* is equal to 1, but in the adaptation process *imp* can be greater than 1. Hence, a weighting of each path must be performed and a weighted set of paths $\hat{w}P$ needs to be introduced, as follows:

$$\hat{w}P = \{\hat{w}p_i\} \quad , \quad \hat{w}p_i = \langle \hat{p}_i, imp_i \rangle \quad (18)$$

where $\hat{w}p_i$ is weighted path that contains path defined previously by Eq. 14 and *imp_i* is importance factor of path \hat{p}_i .

The set of path *P* is extended by probability of occurrence of a particular path in the past, upon fulfilling the condition:

$$Pr(\underline{Path} = p_i) = \frac{\|p_i\|}{\|P\|} \quad (19)$$

where $\|p_i\|$ is number of instances of particular path in the set of paths *P* and $\|P\|$ in number of all paths in set *P*.

The difference between two flowgraphs needs to be determined with distance metrics. The first metric is based on the average absolute deviation in certainty factor assigned to all edges of the flowgraph. This metric is called also conformity. In case that we have two flowgraphs denoted *A* and *B*, it

holds:

$$D = \frac{\sum_{Edges_A} |cer_B(x, y) - cer_A(x, y)|}{\|Edges_A\|} \quad (20)$$

where $Edges_A$ is a set of edges in flowgraph A , cer_A and cer_B are certainty factors from flowgraph A and flowgraph B , respectively, $\|Edges\|$ is the number of edges in the flowgraph A .

The second metric utilizes probability distributions defined by Eq. 19. The metric is based on a coincidence index. Changes in the flowgraph are also related to changes in the probability distribution of paths $Pr(Path)$. One of probability distributions must be determined as a reference (in this case it is Pr_A), whereas the second probability distribution is the modified one (Pr_B). The probability distribution Pr_B comes from the flowgraph B that used more paths as input than the flowgraph A . The mentioned modification is implied as a possibility of appearance paths which were not present in the input set of the flowgraph A . In order to solve this problem a modification must be made of Pr_B . All instances of paths that are in probability distribution Pr_A do not appear in probability distribution Pr_B must be removed from the probability distribution Pr_B . Next, a renormalization of the probability distribution Pr_B is made. The modified probability distribution \hat{Pr}_B prepared in such a way can be used in the formula 21 as follows:

$$CI = \frac{\sum_{Path_A} [Pr_A(Path_A = p_i) \cdot \hat{Pr}_B(Path_B = p_i)]}{\|Path_A\|} \quad (21)$$

where $\|Path_A\|$ determines how many different instances of path is in the set of paths P defined in Eq. 14.

The adaptation method operates in two phases. The first is creation of two new flow graphs ($core_$ and $recent_$) and adding paths to $recent_$ in groups (of i.e. one hundred paths). After adding a group of paths, the distance between $core_$ and $recent_$ is calculated (using formula Eq. 20). If the distance is larger than $learningThershold_$, then $recent_$ is copied to the $core_$ flowgraph and a next group of paths is added to the $recent_$, otherwise the $core_$ is considered as a proper behaviour model and the adaptation algorithm passes to the second phase. In this phase paths are still adding to the $recent_$ in groups, but there is no copying of the $recent_$ flowgraph to the $core_$. After adding the group of paths the distance between the flow graph $core_$ and the flow graph $recent_$ is calculated (see Eq. 20). If this distance is lower than $adaptiveThreshold_$, the next group of paths is added to the $recent_$ flowgraph, otherwise the importance weights (see Eq. 21) is calculated for the last group of paths and these paths are added to the $core_$ flowgraph with appropriate weights. Next, consecutive groups of paths are added to the $recent_$ flowgraph. The flowchart of this algorithm is presented in the Fig. 4.

In order to prove this concept, simulations were performed. The results of the simulations are presented in Fig. 5,

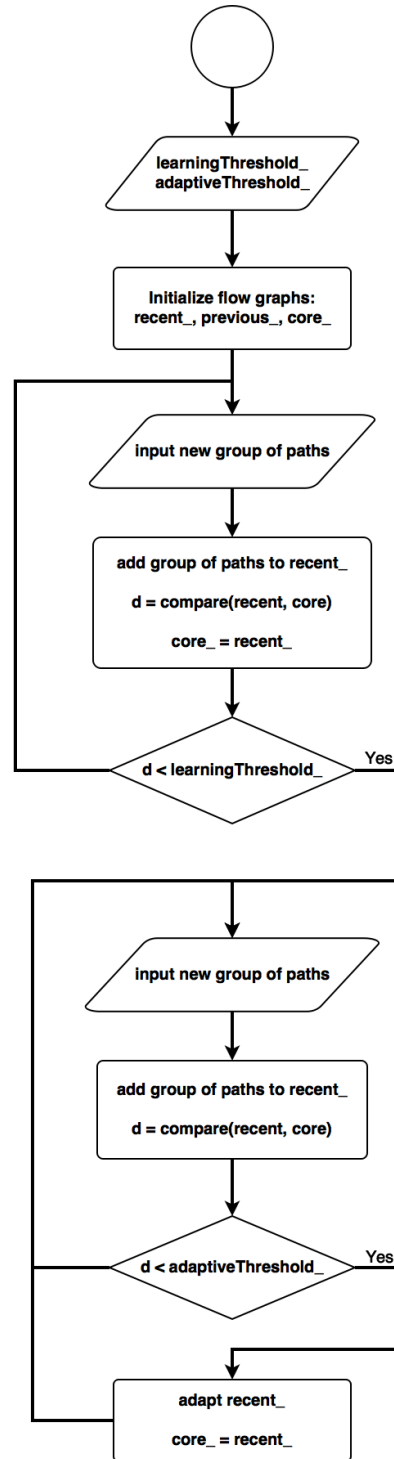
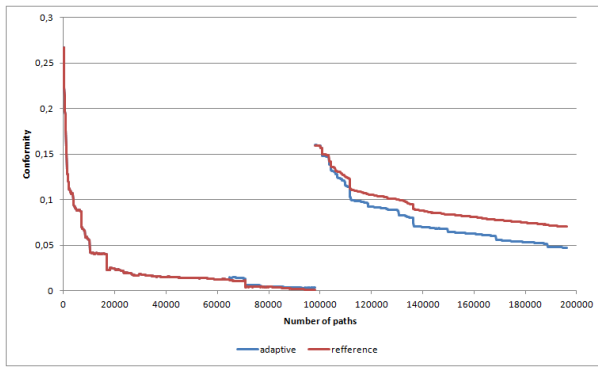


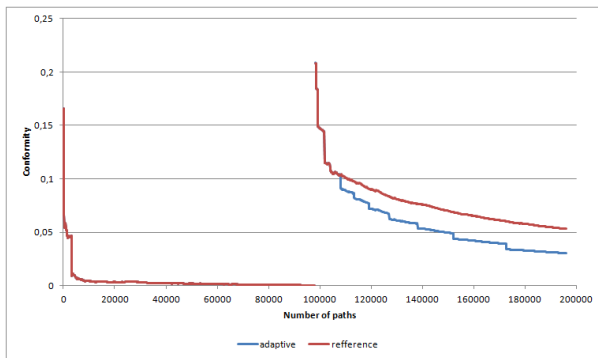
Fig. 4: Flowchart of the used adaptation algorithm

The simulation was carried out in the following way:

- 1) on the basis of real set of paths, which was quite small (about 1000 paths), a large set of path was generated (about 100 thousands paths),
- 2) the *real_small* flowgraph is built on the basis of this



(a) Topology I, learningThreshold_ = 0.0001, adaptiveThreshold_ = 0.005;



(b) Topology II, learningThreshold_ = 0.0001, adaptiveThreshold_ = 0.005;

Fig. 5: Result of preformed simulations for different topologies of video surveillance systems

small set of path,

- 3) building a flow graph enforced with adaptation method is performed with the large (generated) set of paths – adding paths in groups of 100,
- 4) the small set of path is modified in a way which simulates changing the topology of the cameras network,
- 5) the new version of *real_small_flowgraph* is built on the basis of this small modified set of path,
- 6) the second large set of paths is generated on the basis of this modified small set of paths,
- 7) adding the paths form the large set to the recent flow-graph occurs.

The charts from Fig. 5 present the distance (see Eq. 20) between the *real_small_flowgraph* and the *core_flowgraph* (when adaptation method is in use). In order to show the difference, the *reference_flowgraph* is added. This flow graph is built without any adaptation method. The drastic change in the center of the chart is related to the modification of the small set of the path (simulating change in the topology graph). In order to obtain real input data (a small set of object paths) the setup of 6 cameras, denoted as *Topology I*, was used. The *Topology II* was a group of 11 cameras. The *real_small_flowgraph* for both sets of camera was created basing on the analysis of 1,5 hour of video material.

V. CONCLUSION AND FUTURE WORK

The flow graph is a suitable data structure to contain a behaviour model. It is prone to extensions, modifications and adaptation to various types of problems because of its transparency and simplicity. In case of video surveillance systems the flow graph is a container for knowledge concerning object behaviour which is easily to obtain and fast in use. The certainty and coverage factors can be clearly explained and easily applied to problems of object movement prediction (in correspondence to *cer*) or object re-identification (in correspondence to *cov*). Additionally, problem of changing conditions in video surveillance systems also can be also managed using the adaptation method presented above. This adaptation method allows for obtaining better conformity of the flow graph in case of modification of the topology of camera network. Moreover, the flow graph adapts faster to new conditions. The future works related to the flow graphs are concentrated on the application of them in tracking object in one camera in case when the object is obscured by another object or some other obstacles.

ACKNOWLEDGMENT

This work has been partially funded by the Artemis JU and by the Polish National Centre for Research and Development (NCBR) as part of the COPCAMS project (<http://copcams.eu>) under GA number 332913.

REFERENCES

- [1] A. Czyżewski, G. Szwoch, P. Dalka, S. P., C. A., E. D., M. T., L. K., K. L., and W. J., “Multi-stage video analysis framework,” in *Video Surveillance*, L. Weiyao, Ed. Intech, 2011, ch. 9, pp. 145–171. [Online]. Available: <http://dx.doi.org/10.5772/16088>
- [2] A. Czyżewski and P. Dalka, “Moving object detection and tracking for the purpose of multimodal surveillance system in urban areas,” in *New Directions in Intelligent Interactive Multimedia*, ser. Studies in Computational Intelligence, G. Tsihrintzis, M. Virvou, R. Howlett, and L. Jain, Eds. Springer Berlin Heidelberg, 2008, vol. 142, pp. 75–84. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68127-4_8
- [3] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Norwell, MA, USA: Kluwer Academic Publishers, 1992. [Online]. Available: [http://dx.doi.org/10.1016/s0967-0661\(96\)90021-0](http://dx.doi.org/10.1016/s0967-0661(96)90021-0)
- [4] —, “Transactions on rough sets iii,” J. F. Peters and A. Skowron, Eds. Berlin, Heidelberg: Springer-Verlag, 2005, ch. Flow Graphs and Data Mining, pp. 1–36. [Online]. Available: http://dx.doi.org/10.1007/11427834_1
- [5] —, “Decision algorithms, bayes theorem and flow graphs,” in *Neural Networks and Soft Computing*, ser. Advances in Soft Computing, L. Rutkowski and J. Kacprzyk, Eds. Physica-Verlag HD, 2003, vol. 19, pp. 18–24. [Online]. Available: http://dx.doi.org/10.1007/978-3-7908-1902-1_3
- [6] —, “Decision algorithms and flow graphs: A rough set approach.”
- [7] —, “Rough sets and flow graphs,” in *RSFDGrC (1)*, ser. Lecture Notes in Computer Science, D. Slezak, G. Wang, M. S. Szczuka, I. Dntsch, and Y. Yao, Eds., vol. 3641. Springer, 2005, pp. 1–11. [Online]. Available: http://dx.doi.org/10.1007/11548669_1
- [8] B. Kostek and A. Czyżewski, “Processing of musical metadata employing pawlaks flow graphs,” in *Transactions on Rough Sets I*, ser. Lecture Notes in Computer Science, J. Peters, A. Skowron, J. Grzymala-Busse, B. Kostek, R. Swiniarski, and M. Szczuka, Eds. Springer Berlin Heidelberg, 2004, vol. 3100, pp. 279–298.

- [9] P. Pattaraintakorn, "Entropy measures of flow graphs with applications to decision trees," in *Rough Sets and Knowledge Technology*, ser. Lecture Notes in Computer Science, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Springer Berlin Heidelberg, 2009, vol. 5589, pp. 618–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02962-2_78
- [10] J. Peters, D. Chitcharoen, and S. Ramanna, "Reasoning with near set-based digital image flow graphs," in *Multi-disciplinary Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, S. Ramanna, P. Lingras, C. Sombatheera, and A. Krishna, Eds. Springer Berlin Heidelberg, 2013, vol. 8271, pp. 199–210. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-44949-9_19
- [11] Z. Suraj and K. Pancierz, "Flow graphs as a tool for mining prediction rules of changes of components in temporal information systems," in *Rough Sets and Knowledge Technology*, ser. Lecture Notes in Computer Science, J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. Cercone, and D. Izak, Eds. Springer Berlin Heidelberg, 2007, vol. 4481, pp. 468–475. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72458-2_58
- [12] Z. Pawlak, "Rough sets, decision algorithms and bayes theorem," *European Journal of Operational Research*, vol. 136, no. 1, pp. 181–189, 2002. [Online]. Available: [http://dx.doi.org/10.1016/s0377-2217\(01\)00029-7](http://dx.doi.org/10.1016/s0377-2217(01)00029-7)
- [13] A. Czyżewski and K. Lisowski, "Employing flowgraphs for forward route reconstruction in video surveillance system," *Journal of Intelligent Information Systems*, vol. 43, no. 3, pp. 521–535, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10844-013-0253-8>
- [14] Z. Pawlak, "Rough sets and flow graphs," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, ser. Lecture Notes in Computer Science, D. Slezak, G. Wang, M. Szczuka, I. Duntsch, and Y. Yao, Eds. Springer Berlin Heidelberg, 2005, vol. 3641, pp. 1–11. [Online]. Available: http://dx.doi.org/10.1007/11548669_1
- [15] A. Czyżewski and K. Lisowski, "Adaptive method of adjusting flowgraph for route reconstruction in video surveillance systems," *Fundam. Inf.*, vol. 127, no. 1-4, pp. 561–576, Jan. 2013. [Online]. Available: <http://dx.doi.org/10.3233/FI-2013-927>