Topology Estimation of a Camera Network Using Sparse Methods

Isabel Queirós da Silva
Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
isabel.silva@tecnico.ulisboa.pt

***Abstract* — This work studies the topological representation of multi-camera, non-overlapping, vigilance systems, through graphs, using data from pedestrian's trajectories. The topological information of a system is very useful for the re-identification problems. By knowing the possible paths in an environment, when a pedestrian disappears from the image of a camera, the number of possible places where he could reappear decreases, making the problem easier to solve. The nodes of the graphs that we want to obtain will be the entry and exit zones of the fields of view of the cameras and the links represent the possible paths in the considered environment. The thesis proposes an automatic estimation method of sparse graphs. For that, we use the data of the verified paths in the environment. These data may contain errors and, consequently, not all the links verified in it will be true. The proposed method uses these data and assigns weights to all the links verified in it, based on the number of occurrences for each path and the similarity of feature vectors associated with each pedestrian, at the moments of entry and exit of the fields of view of the cameras. The proposed method was tested using data generated by a simulator. We performed tests in the presence of a fixed noise in the feature vectors and for several values of the percentage of errors of missed detections and of false positives.**

Keywords - Multi-Camera Network; Non-Overlapping Fields of View; Topology; Sparse Graphs

I. INTRODUCTION

Video surveillance has a wide range of potential applications [1]. To monitor a big environment it is necessary to use multiple cameras. To a set of cameras monitoring the same environment with a particular purpose, we call *camera network* [2]. The manual monitoring of a camera network is inefficient and unreliable. Hence, several automatic tracker systems have been developed. The task of tracking objects through a surveilled environment can be divided into three problems [3]. The first one is tracking objects within the field of view of a single camera. The second one is combining the tracks of the same object that is being simultaneously observed by more than one camera. The third one is tracking objects when they transit from an observed region of the environment to an unobserved one, reappearing after a while. This requires re-identification of the targets. In this work, we are going to focus on the third mentioned problem.

The re-identification problem is defined as the process of establishing correspondences between images of the same object, acquired from cameras without overlapping fields of view, in different instances of time [4]. This problem represents a great challenge in computer vision mainly because of the many possible variations of acquisition's conditions of the different cameras. These could be, for instance, differences in illumination, position or color. All combined, they generate great changes in the appearance of the detected objects, making this a difficult problem to solve.

If we have no information about the network, it is considered possible, for an object that disappears from one camera, to reappear in any other region of the network, at any possible time. But if we have some knowledge about the spatial relations between cameras, we decrease the number of possibilities and, consequently, decrease the ambiguity of the problem. Thus, the topological information has an important role in the re-identification task and automatic topology estimation methods are needed.

The goal of our work is to develop a topology estimation method for a non-overlapping camera network. The topology will be represented by a graph, where the nodes correspond to regions of the environment and the links represent the paths where the pedestrians can walk through. The graph can also have information about the transition probabilities and the time distributions of the links.

Our method will use data from verified paths in a certain camera network. These data may contain errors and the goal is that the topology estimation algorithm can be able to obtain a perfect graph, that is, only with links that correspond to real paths.

We will work with a manually labeled dataset. This provides us error-free data, that we can use to obtain the ideal graphs and to model the network. We will then simulate it, for different values of error. Using the proposed topology estimation method, we can use the simulated data to obtain graphs. We will compare them with the ideal ones to evaluate the performance of the algorithm in different error conditions.

This paper is organized as follows. Section II presents the problem formulation and the proposed topology estimation method. Section III describes how we have extracted data from a labeled dataset and from the simulator. Section IV presents the characteristics of the considered camera network, explains how are we going to evaluate the results and the tests that we are going to perform. Section V presents the results and the discussion. Finally, we conclude the paper in section VI.

II. GRAPH ESTIMATION METHODS

A. Problem formulation

Given an environment observed by a non-overlapping camera network, consider a trajectory made by a pedestrian, like the one represented in Fig. 1a. The pedestrian can pass through visible and invisible regions. Thus, the trajectory could be divided into segments: the visible paths (is the case of the segments A-B, C-D and E-F) and the invisible paths (the segments B-C and D-E). What divide the trajectory are the entries and exits of the pedestrians in the fields of view of the cameras. We consider that exists a limited set of regions where it is possible to enter and exit a field of view. We call them entry and exit zones.

We want to represent the paths made by the pedestrians using graphs: the graph of visible paths and the graph of the invisible paths. The nodes will represent the entry and exit zones and the links between them represent the possible visible and invisible paths, respectively, in the considered environment. In Fig. 1b and Fig. 1c we represent the obtained graphs for the trajectory represented in Fig. 1a.

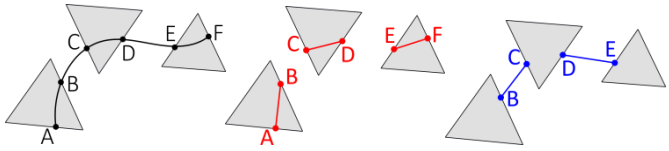

Figure 1. Example of a (a) trajectory and the respective (b) graph of the visible paths and (c) graph of invisible paths. The grey areas represent the fiels of view of the cameras.

We consider we have access to data about the visible and invisible paths, verified in a certain observed environment. For each path, we have the identification number of the pedestrian, the entry node, a feature vector associated with the entry, the exit node, a feature vector associated with the exit and the duration of the path, that we call transition time. The feature vector has representative values about the pedestrian that could be, for example, color characteristics. The data could contain errors and we want to use these data to obtain a perfect graph, where all the links correspond to real paths.

B. Proposed topology estimation method

The proposed topology estimation method is in [5]. We are going to present the algorithm and explain how to use it in our work. With this method, we can learn sparse graphs by comparing a set of features of the objects of interest. Each node of the obtained graph represents an object. The feature matrix D has dimensions $n \times m$, where n is the number of nodes and m is the number of features used to compare them. By solving the convex optimization problem presented in (1), we obtain the matrix W that has the weights of all possible links of the graph. W has dimensions $n \times n$. If the features of two objects are very similar, then the corresponding link will have a higher weight, representing a stronger connection.

In (1) the only free parameter is β. A larger value of β encourages sparser graphs, that is, smaller weight values. The solution for the convex optimization problem in (1) can be found using the CVX package.

$$\max_{\tilde{\Delta}>0, W, \sigma^2} \log|\tilde{\Delta}| - \operatorname{trace}\left(\tilde{\Delta}\frac{1}{m}DD^T\right) - \frac{\beta}{m}\|W\|_1$$

s.t.

$$\begin{aligned} &\tilde{\Delta} = \operatorname{diag}\left(\textstyle\sum_j w_{ij}\right) - W + \frac{I}{\sigma^2} \\ &w_{ii} = 0, i = 1, \dots, n \\ &w_{ij} \geq 0, i = 1, \dots, n; j = 1, \dots, n \\ &\sigma^2 > 0 \end{aligned} \tag{1}$$

In our work, we are going to use the data from the feature vectors to obtain the weights. The considered feature vectors are simultaneously associated with a node and a link. So, it is not possible to construct a feature matrix to compare all the existing nodes. Instead, we compute a matrix D for each link verified in the considered data. For each link, we will solve the optimization problem in (1) and obtain the corresponding weight w. In our case, the value of the weights is also as high as the number of occurrences verified for that path.

We will consider the weights of all links with no occurrences as being zero and they will not be represented in the obtained graph. For all the other links we can choose, based on their weight values w, which ones we want to accept as being correct. We can do this by setting a threshold value T and choose all the links with $w > T$.

C. Transition probabilities

The transition probability $P_{i\to j}$ represents the probability of a pedestrian, being at node i, travel to node j. We can obtain it using (2).

$$P_{i\to j} = \frac{1+\#\, i \to j}{\#possibilities + \#\, i \to \forall} \tag{2}$$

Where $\#\, i \to j$ is the number of occurrences of the path that begins at node i and ends at node j; $\#possibilities$ is the number of links in the graph that are connected to node i and $\#\, i \to \forall$ is the number of occurrences of all paths that begin at node i.

D. Time distributions

The time distributions give us information about the probability of a pedestrian taking a certain amount of time to make a given path. We represent them with probability distribution functions and we will obtain one for each represented link on the graph. We are going to use the time information of each verified path to obtain the corresponding distribution.

We choose to model the probability distribution functions with a mixture of two densities: a Gamma and a Rayleigh. We are going to estimate the mixtures with the Expectation-Maximization algorithm [6].

III. GENERATION OF VIDEO EVENTS

To obtain the camera events it is necessary to make a direct data processing of the video. That is not the goal of our work, so we are going to generate events by two ways: with a manually labeled dataset and with a simulator.

A. Labelled dataset

We are going to work with a fully manually labeled dataset, composed of the image sequences of the considered camera network. We have access to the position of all pedestrians at every frame, with the corresponding identification. With this information it is possible to make some data processing and obtain all the entry and exit zones of the considered cameras. We can also obtain all the verified visible and invisible paths.

As the dataset was manually labeled, all the data about the verified paths and the entry and exit zones are considered correct. So, we can use this information to obtain the graphs of the visible and the invisible paths, without any errors. We will call them *ideal graphs*. We can also use the data to obtain all the transition probabilities and the time distributions for all links of the ideal graphs.

B. Simulator

We will use the transition probabilities and the time distributions obtained from the information of the dataset to simulate the considered camera network. The simulator will also receive as a parameter the total simulation time and the probability of entering a new pedestrian in the camera system, per second. It will generate entries and exits of pedestrians in the considered nodes, providing us with visible paths.

Each simulated pedestrian will be associated with some generated feature vector $v_F = (x_F, y_F)$. Each value of the vector will be randomly generated with a uniform distribution and can have values between -0.5 and 0.5.

We will also simulate errors in the generated paths. The simulated errors are missed detections, false positives and noise in the feature vectors.

The missed detections error will only affect paths with less than three frames of duration. We will receive as parameter the probability P_{MD} of a one frame path not being detected. Given this value, the probability of a two frame path not being detected is given by P_{MD}^2 and for a three frame path the probability is P_{MD}^3.

The false positives error will generate false visible paths. The generated paths must begin and end at the same node. We receive as a parameter the probability P_{FP} of being detected a false positive in the images of the camera network, per second. To each false positive we will attribute a simulated feature vector. By comparing those features with the ones from simulated paths with no error, we can associate the false positive to an existing pedestrian or to a new one.

The simulator also generates noise in the feature vectors, with a magnitude r_F, received as a parameter. The noise will affect the features associated with each visible path, simulated with no errors. The features associated with the entry will stay equal to the features associated with the exit, for the same visible path, even after the noise is generated.

At the end of the simulation, we can infer the invisible paths from the information about the visible paths.

IV. Experiences

A. Characteristics of the considered camera network

We are going to work with a set of images acquired by 13 cameras from the *HDA dataset* [7][8]. These images were manually labeled and provide us the position and identity of all visible pedestrians. The cameras are spread over two floors of a building, in a typical office environment. Some have overlapping fields of view. In these cases, we consider the overlapping cameras as being one, what results in 8 considered cameras with no overlapping. From the verified detections, we obtained 33 entry and exit zones, that will constitute the nodes of the desired graphs. The data results in 38 invisible links and 60 visible ones. The ideal graph of invisible paths is presented in Fig. 2 and the ideal graph of the visible paths is presented in Fig. 3. The graphs are plotted over a scheme of the two floors where the cameras are located. The cameras position and their fields of view are represented in the scheme, in a light red.

The percentage of possible links between the existing nodes that were in fact verified is 3.49% for the invisible links and 5.51% for the visible ones. From the data, we were able to obtain the transitions probabilities and the time distributions for the verified paths.

B. Validation

Using the proposed topology estimation method, we obtain a set of weights *w*, each corresponding to a path verified in the used data. It is possible that not all verified paths correspond to real ones and so we will use the weight information to choose the links we want to consider as correct. We can select a threshold *T* and accept all the links that verified $w > T$. We will evaluate the used method comparing the *selected links* to the real ones, which we know from the ideal graphs and call *relevant links*. To the selected links that are represented in the ideal graphs we call *true positives* and to the others we call *false positives*. There can also exist some correct links, represented in the ideal graphs, which were not selected. To them we call *false negatives*.

For each estimated graph and selected threshold *T* we can compute the *precision* and the *recall*. The precision is given by the ratio between the number of *true positives* and the number of *selected links*. The recall is given by the ratio between the number of *true positives* and the number of *relevant links*. We can use these values to compute the *F-measure*, given by

$$Fmeasure = \frac{2\,precision \cdot recall}{precision + recall} \quad (3)$$

For a set of weights, we can vary the value of the *threshold* and compute the *precision*, *recall* and *F-measure*. We obtain the *PR* curve, which is given by the *recall* vs. the *precision* values, and the *F* curve, which is given by the *F-measure* vs. the *threshold* values. By analysing the *F* curves, we can see which value of the threshold maximizes the *F-measure*. This represents the best value for the threshold, which gives us the graph with less possible error, given the considered data.

C. Tests

We will use the transition probabilities and the time distributions for the verified paths in the ideal graphs to simulate the considered camera system, resulting in generated

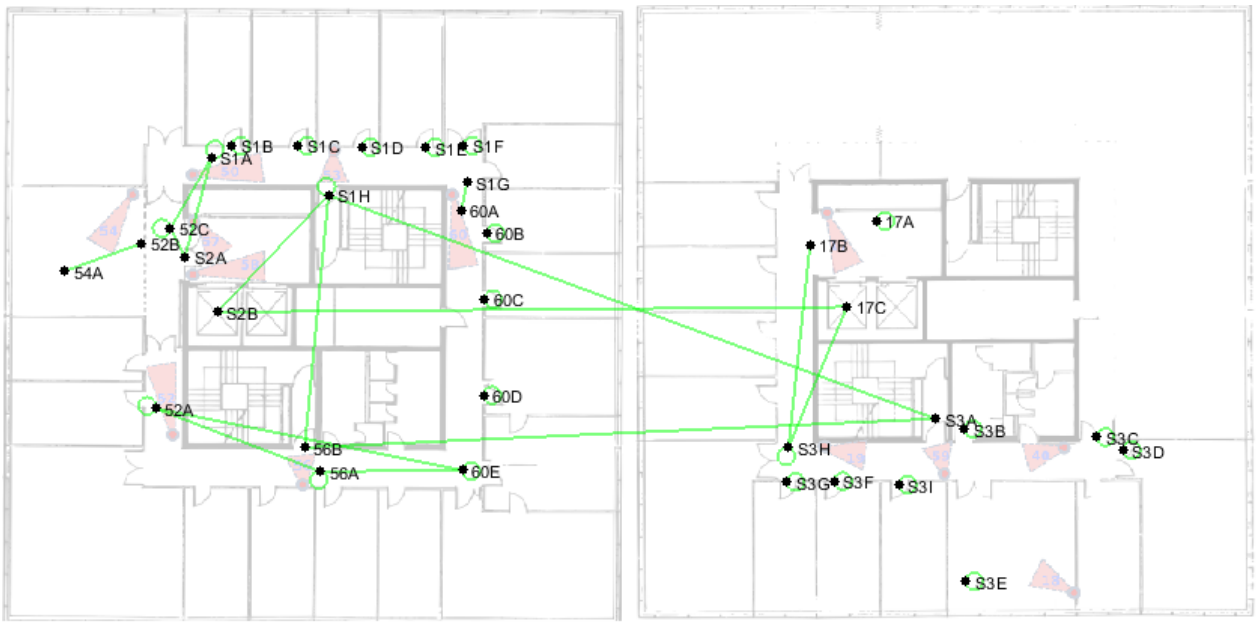

Figure 2. Ideal graph of the invisible paths.

Figure 3. Ideal graph of the visible paths.

path's information. We are going to generate errors in these data and use them to obtain graphs, with the proposed topology estimation method. Comparing the obtained graphs with the ideal ones, we can evaluate the performance of the proposed algorithm for different values of errors.

We want to study the impact of missed detection and false positive errors in the obtained graph using the proposed topology estimation method. For this, we are going to simulate the considered camera system for different values of these errors, by varying the probability P_{MD} of a one frame path not being detected and the probability P_{FP} of being detected a false positive in the images of the camera network, per second. The tested values are $P_{MD} = \{0, 0.3, 0.065, 0.8\}$ and $P_{FP} = \{0, 0.025, 0.05, 0.08\}$. We made simulations for every combination of these values, which results in 16 cases. For each case we made 10 simulations.

All the simulations will be made for a total simulation time of *172800* seconds and for a probability $P_{new} = 0.05$ of entering a new pedestrian in the camera system, per second. We selected a fixed value for the noise in the feature vectors of $r_F = 0.1$. For the visible paths, we set the free parameter $\beta = 60$ and, for the invisible paths, $\beta = 20$.

The *PR* and *F* curves that we are going to present are the mean of the curves obtained for each of the 10 simulations made for the considered case. Also, the graphs result of the

median of the 10 graphs obtained from the simulations in the considered conditions.

V. RESULTS

The simulated case for $P_{MD} = 0$ and $P_{FP} = 0$, even with the presence of noise in the feature vectors, only provides us data about correct paths. So, for small values of the threshold, the obtained graphs will be equal to the ideal ones. This happens because the tested noise does not create any occurrences or ignore data. Comparing this results with the results from the simulated case without any errors, the only verified change was a generalized decrease in the weights of the invisible links. This happens because, in the presence of noise, the feature vectors associated with the entries and exits of the same path aren't necessarily equal. For the visible links these values are equal.

First, we are going to compare this case to the ones with missed detection errors, then with the ones with false positive detections. After, we are going to present the results with both errors. By the end, we will make some generalized comments about the results.

A. Cases with missed detection errors

Since the missed detection errors do not generate new paths, the data will not verify new occurrences of visible links. So, for a small value of the threshold, the obtained graph of visible paths will remain equal to the ideal one. The only difference is a small decrease in the weights of some links, which is expected since the tested error ignores some visible paths. However, this generates new occurrences of invisible paths, which is also expected. For example, considering Fig. 1, if we ignore the visible path C-D, we will infer the invisible path B-E instead of the represented ones. The mean PR and F curves for the invisible paths are represented in Fig. 4 and Fig. 5., respectively.

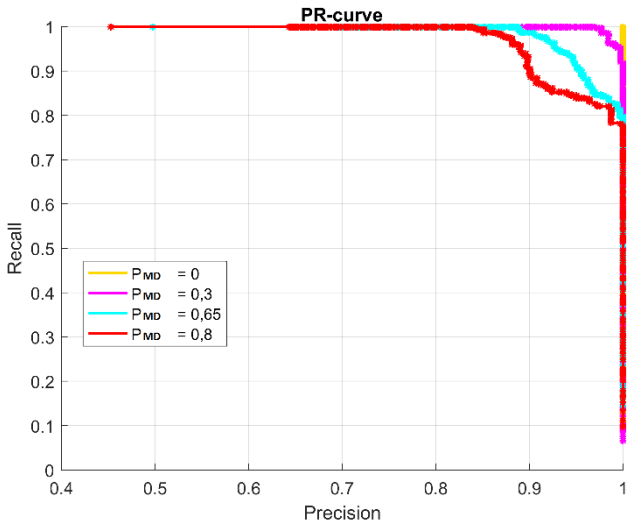

Figure 4. Mean PR curves obtained for the invisible paths, for the values $P_{MD} = \{0, 0.3, 0.065, 0.8\}$ and $P_{FP} = 0$.

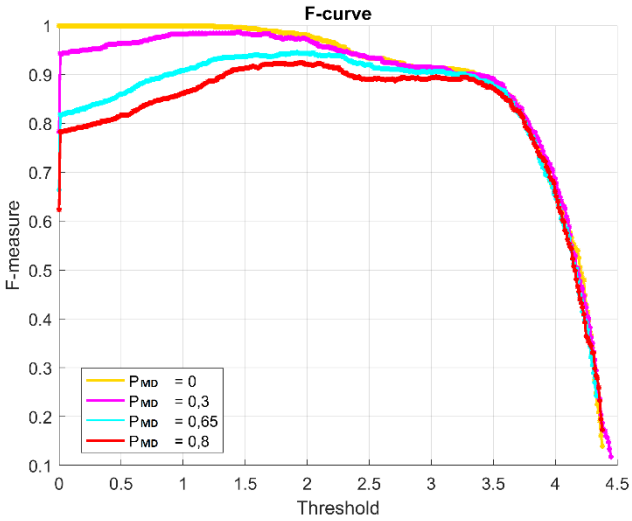

Figure 5. Mean F curves obtained for the invisible paths, for the values $P_{MD} = \{0, 0.3, 0.065, 0.8\}$ and $P_{FP} = 0$.

The errors in the invisible paths affect the final results, as we can see in the F curve (Fig. 5). With no missed detection errors, the maximum possible value for the F-measure is 1. This indicates that it is possible to obtain a graph equal to the ideal. For the other tested cases, this is not verified. This happens because, as we increase the error, more false paths will be generated and the respective weights will become higher than the weights associated with some correct links.

B. Cases with false positives errors

The false positives errors generate new occurrences of visible paths. All the false positives correspond to links that begin and end at the same node. We verified that approximately half of the generated false positives are attributed to pedestrians that had already appear in the network. In these cases, new invisible paths may be generated. The mean PR and F curves for the invisible paths are presented in Fig. 6 and Fig. 7, respectively. The PR and F curves for the visible paths are presented in Fig. 8 and Fig. 9. In both cases, as the error increases, the maximum possible value of the F-measure decreases.

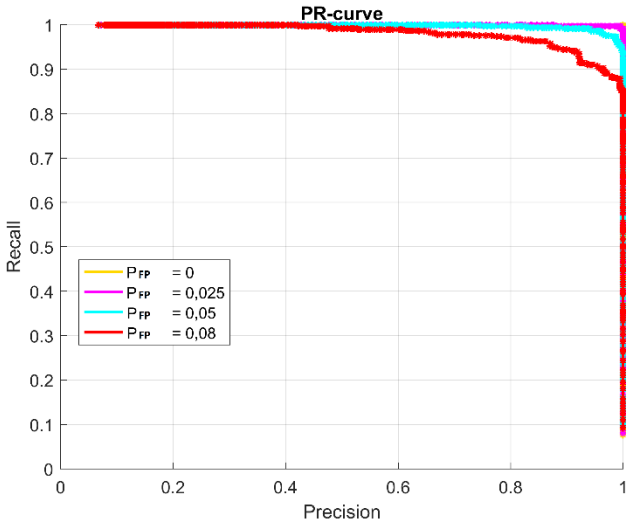

Figure 6. Mean PR curves obtained for the invisible paths, for the values PFP = {0, 0.025, 0.05, 0.08} and PMD = 0.

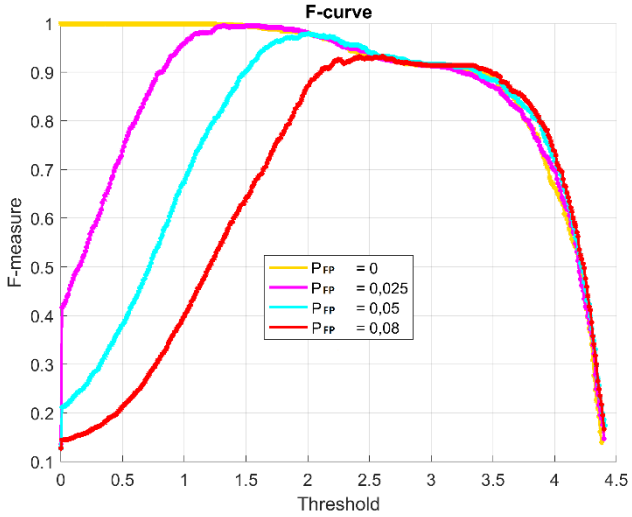

Figure 7. Mean F curves obtained for the invisible paths, for the values PFP = {0, 0.025, 0.05, 0.08} and PMD = 0.

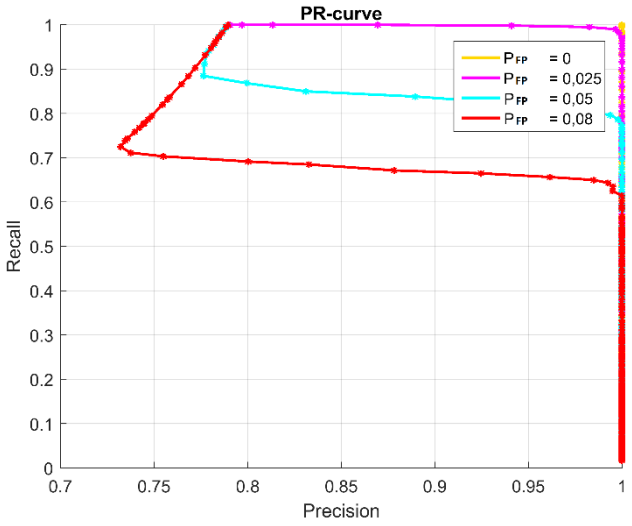

Figure 8. Mean PR curves obtained for the visible paths, for the values PFP = {0, 0.025, 0.05, 0.08} and PMD = 0

For the visible paths it is possible to choose a threshold value ($T = 5$) that maximizes the F curve for all the tested errors. This occurs because, even if we continue to increase the error, is it not possible to generate more wrong links with false positives, since it only generates paths that begin and end at the same node. We can also see that for the higher error cases, the precision value for the visible paths initially decreases. This happens because all first rejected links correspond to correct paths.

For the invisible paths we obtain better results, since the F curve reaches higher values. We can also notice that the threshold value that maximizes the curve is not the same for all the tested errors.

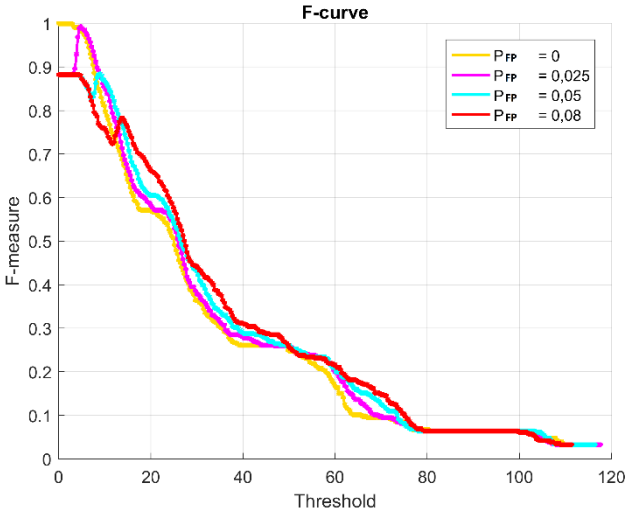

Figure 9. Mean F curves obtained for the visible paths, for the values P_{FP} = *{0, 0.025, 0.05, 0.08} and* P_{MD} = *0.*

C. *Cases with missed detections and false positive errors*

We are now presenting the results for the simulated cases with missed detections and false positives errors. The tested cases were all combinations of the values P_{MD} = *{0, 0.3, 0.065, 0.8}* and P_{FP} = *{0, 0.025, 0.05, 0.08}*. The mean PR and F curves obtained for the invisible paths are presented in Fig. 10 and Fig. 11 and the curves obtained for the visible paths are presented in Fig. 12 and Fig. 13.

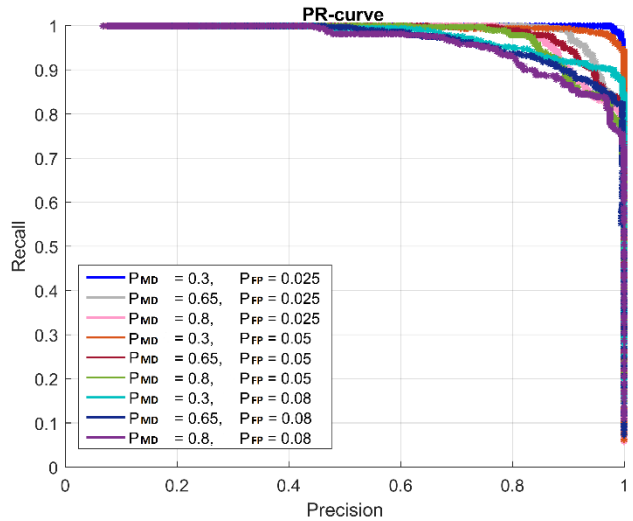

Figure 10. Mean PR curves obtained for the invisible paths, for the values P_{MD} = *{0.3, 0.065, 0.8}* and P_{FP} = *{0.025, 0.05, 0.08}*.

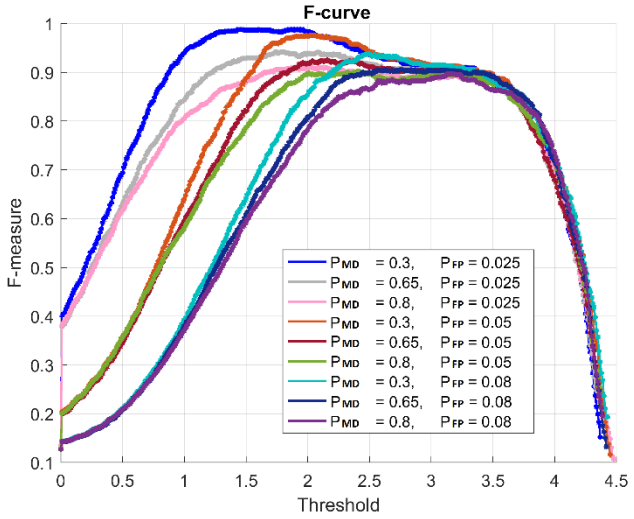

Figure 11. Mean F curves obtained for the invisible paths, for the values P_{MD} = *{0.3, 0.065, 0.8}* and P_{FP} = *{0.025, 0.05, 0.08}*.

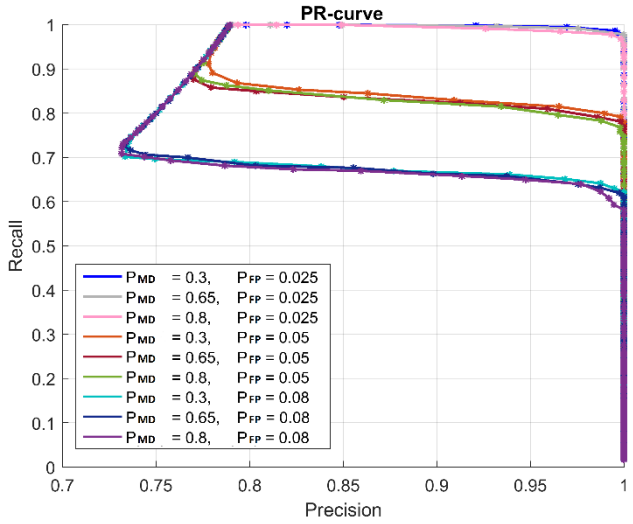

Figure 12. Mean PR curves obtained for the visible paths, for the values P_{MD} = *{0.3, 0.065, 0.8}* and P_{FP} = *{0.025, 0.05, 0.08}*.

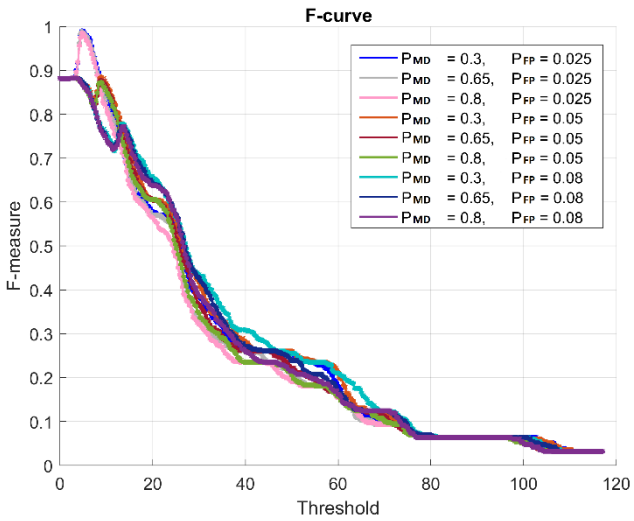

Figure 13. Mean F curves obtained for the visible paths, for the values P_{MD} = *{0.3, 0.065, 0.8}* and P_{FP} = *{0.025, 0.05, 0.08}*.

For the invisible paths, the maximum possible value of the F curve and the corresponding threshold value depends of the tested error.

The mean F curve obtained for the visible paths depends essentially from the value of false positive errors. We can choose a value for the threshold that maximizes the F curve for all the tested cases presented in Fig. 13. That value is around T = 4.5.

We are going to represent two obtained graphs for the visible paths. In both cases we choose the threshold value that maximizes the corresponding F curve, that is $T = 4.5$. We choose to represent the results obtained for the tested conditions with the smallest and the biggest values of error. But, as we can see by the similarity of the F curves, all the other tested cases have very similar resulting graphs. In Fig. 14 we represent the graph of the visible paths obtained for the case with values $P_{MD} = 0.3$ and $P_{FP} = 0.025$. In Fig. 15 it is represented the resulting graph of the visible paths obtained for the values $P_{MD} = 0.8$ and $P_{FP} = 0.08$.

Figure 14. Graph of the visible paths for the case with values $P_{MD} = 0.3$, $P_{FP} = 0.025$ and $T = 4.5$. The green links represent the true positives.

Figure 15. Graph of the visible paths for the case with values $P_{MD} = 0.8$, $P_{FP} = 0.08$ and $T = 4.5$. The green links represent the true positives and in red are the false positives.

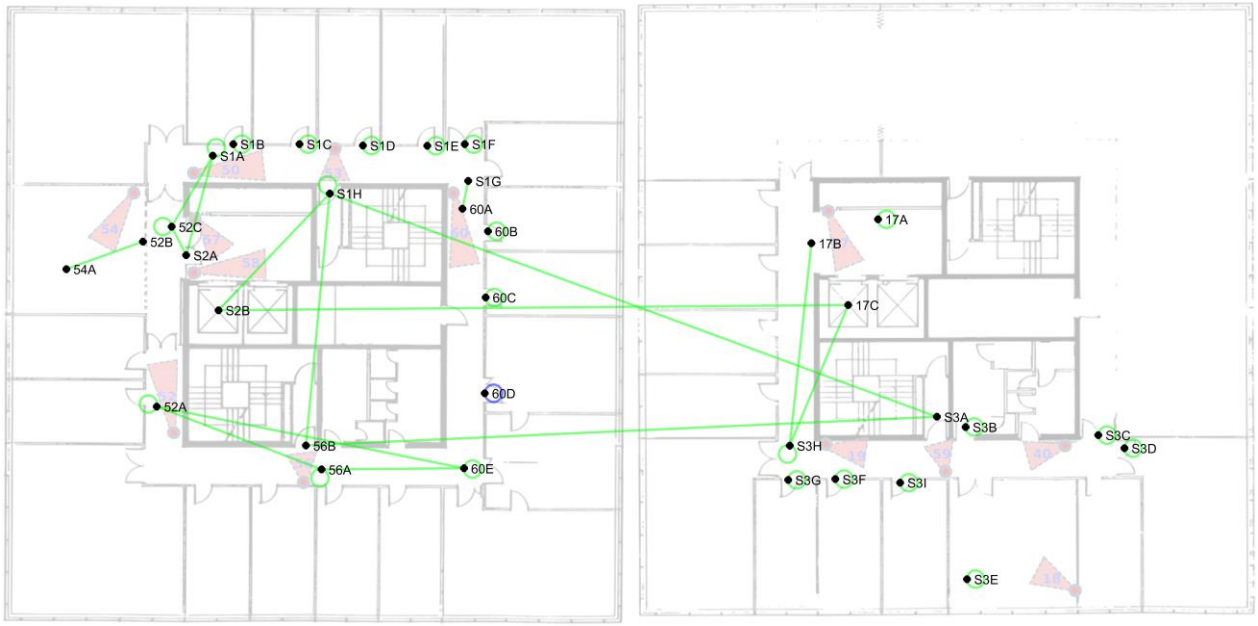

Figure 16. Graph of the invisible paths for the case with values $P_{MD} = 0.3$, $P_{FP} = 0.025$ and $T = 2$. The green links represent the true positives and the blue ones represent the false negatives.

We also represent the graph of invisible paths for the same cases, but with different values for the threshold. The selected threshold maximizes the corresponding curves. In Fig. 16 we represent the graph for the case $P_{MD} = 0.3$ and $P_{FP} = 0.025$, with $T = 2$. In Fig. 17 we represent the resulting graph for the case $P_{MD} = 0.8$ and $P_{FP} = 0.08$, with $T = 3$.

D. Discussion

For the tested cases, it is possible to choose a threshold value that maximizes the value of the F-measure, for the visible links. This allows us to obtain always the best possible graph of the visible paths.

For the invisible paths the same is not true. However, it is not necessary to know exactly the error values. If we have a general knowledge about the error of the tracker that we are working with, we can choose an adequate threshold value. According with the tested cases, if we have a small error we should choose a threshold around 2 and if we have a higher error we should choose a threshold around 2.5 and 3.

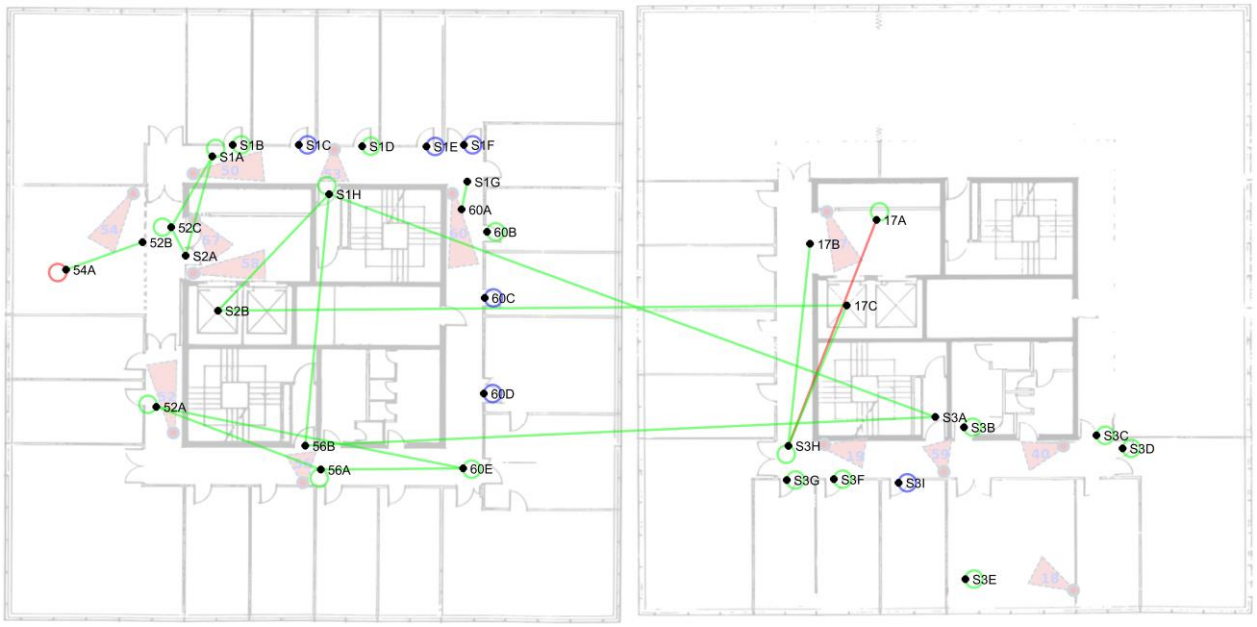

Figure 17. Graph of the invisible paths for the case with values $P_{MD} = 0.8$, $P_{FP} = 0.08$ and $T = 3$. The green links represent the true positives, the red ones represent the false positives and the blue ones the false negatives.

We expect the general behavior of PR and F curves to be the same, for all systems: that is, for higher threshold values, we will select less links. If the rejected links correspond to correct paths, then the F curve decreases. If, on the other hand, the link was generated by errors, the F curve will increase.

Besides this, the threshold values for which the F curve increases or decreases will be different. Even for the same system, these values will change with the acquisition time. For the same system, a shorter acquisition will generate fewer paths and that will correspond to links with smaller weights. So, the F curves will have the same behavior for different threshold values.

For other systems, even if we consider the same acquisition time, the paths will have a distinct behavior and we cannot predict the threshold values that maximizes the F curve.

The obtained resulting graphs using the proposed topology estimation method depend on the considered values of missed detection and false positive errors.

VI. Conclusions

In this work we proposed a topology estimation method. We obtained results for simulated data and performed tests for several error values of missed detections and false positives. We evaluated the obtained results by comparing them with ideal graphs, constructed from error-free data. For each tested case we computed the PR and F curves. With the F curves we can choose a value of threshold that allow us to obtain the best graphs possible, that is, with less errors.

We have concluded that, for the considered system and for the chosen acquisition time, the threshold value that maximizes the F curve is approximately the same for all the tested cases. For the graph of the invisible path the same is not valid. We can, however, choose an approximated threshold value, even if we do not have knowledge about the error values.

The most basic algorithm for the topology estimation is to consider all the links verified in the graphs. That corresponds to choosing the threshold value $T = 0$, for all the considered cases. So, with our method, by choosing a threshold value different from zero, even if we don't choose the best one, we will obtain better results.

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.*, vol. 34, no. 3, pp. 334–352, 2004.

[2] D. Kamenetsky, "Camera Network Topology Discovery Literature Review," 2011.

[3] D. Makris, Tim Ellis, and J. Black, "Bridging the gaps between cameras," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 205–210, 2004.

[4] A. Bedagkar-Gala and S. K.Shah, "A survey of approaches and trends in person re-identification," *Image Vis. Comput.*, vol. 32, no. 4, pp. 270–286, 2014.

[5] B. M. Lake and J. B. Tenenbaum, "Discovering Structure by Learning Sparse Graphs," 2010.

[6] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," 1998.

[7] D. Figueira, M. Taiana, A. Nambiar, J. C. Nascimento, and A. Bernardino, "The HDA+ data set for research on fully automated re-identification systems," in *Workshop, European Conference on Computer Vision (ECCV)*, 2014.

[8] A. Nambiar, M. Taiana, D. Figueira, J. C. Nascimento, and A. Bernardino, "A Multi-camera video data set for research on High-Definition surveillance," in *Int. J. of Machine Intelligence and Sensory Signal Processing*, 2014.