# Multiple target tracking in a network of cameras

MCT

Table of Contents

Abstract

The paper investigates the learning of a model of relationship between cameras using a large set of observations. This model of relationship then can be used to perform multiple target tracking across a network of multiple cameras. The relationship model between cameras can differ on whether two cameras have an overlapping field of view. In the case on which two different cameras are overlapped, fused information from multiple cameras into a 3D coordinate system can be used to perform object tracking. In the case on which two cameras do not overlap, a link relationship model between the sensor gap of the two cameras can be used to perform object tracking. In contrast to most existing work, we only rely on high-level tracklet information from real-time single camera tracking software of video streams used for surveillance purpose.

## Necessity

In the field of surveillance, the task of assigning the same identifier to all instances of a particular individual captured in a series of images or videos is one of the most important and desirable yet very challenging especially when there are a significant number of targets involved. Solving this tracking issue automatically using algorithms instead of using human effort is highly desirable considering the huge amount of data from video streams. Due to the nature of surveillance programs being time-critical, it is important to solve this video in real-time without sacrificing performance compared to the state-of-the-art offline multi-object tracking approach.

## Related Works

A common approach to multiple people tracking is to perform the task in a hierarchical approach. First, an object detector is used to detect object in a single frame of a video. This step is then repeated for the set of all frames in a video to produce a set of detections. This set of detections are then linked together using tracking algorithms to create a set of tracklets. This task of creating tracklets from a series of images from a single camera is called single camera tracking (SCT). The next task is to perform object tracking across two cameras or also called inter-camera tracking (ICT). This task is concerned on maintaining the identities of individuals even when they move from the field of view of one camera to another. This is done by linking the tracklets in one camera to another by using a data association framework. In a system where there are multiple, more than two cameras, the task of maintaining identities of individuals across all cameras is called multi camera tracking (MCT). The goal of this work is to perform multi camera tracking. Multiple object tracking in SCT, ICT and MCT becomes a data association problem where first, at a low-level pre-processing step, detections need to be linked to form

tracklets, then high-level tracklets need to be linked to form target trajectories. Most recent tracking algorithms address the association problem differently yet most of them rely heavily on building strong appearance models by performing low level computations at the pixel level.

The goal of this work is to introduce a novel framework of using knowledge derived from information of high-level tracklets produced by single camera tracking software to track identities of individuals in multiple scenes. Knowledge can be discovered from identifying patterns from a large data set of tracklets produced by multiple cameras that can be used to perform object tracking.

## Multiple target tracking in a network of cameras

We propose to solve the data association problem by first performing supervised and unsupervised learning of a relationship model between cameras from a large training data set produced by single camera tracking software of multiple cameras. The relationship between cameras can differ whether two cameras have an overlapping field of view. This research will make an assumption that the organization of the network of cameras have been learned beforehand.

### Overlapping Field of View

We propose a learning algorithm to establish the correspondence between the multiple cameras from a large set of observations. This would enable the projection of multiple scenes into a 3D coordinate system which then could be used to achieve the matching by using projected position information. This 3D coordinate system can be used to robustly estimate the location of objects in a scene. This could be done by first creating a mapping table.

A mapping table records all the instances where a detection at one camera simultaneously occurs with another detection at another camera. A camera view is first divided into 100 equal sized grids indexed 0 from the top left to 99 at the bottom right. A mapping table comprises of association between a single grid of a camera to another grid of another camera. An association has a counter that records the frequency of a single instance. The mapping table could be interpreted later to identify patterns that could be used to create a 3D coordinate system.

**Mapping table.**

A simultaneous observation in a camera with another observation from another camera are associated as an entity called mapping. A mapping table is a record of all mappings of observation between multiple cameras. As mentioned above, a camera view is first divided into 100 equal sized grids indexed 0 from the top left to 99 at the bottom right. The associated grids are the aforementioned mappings. A mapping has four attributes: grid_a, grid_b, frequency, support. During the learning phase, observations are first stored in the mapping table for later analysis. A repeating observation is recorded as an increment of the frequency attribute. The frequency attribute can be later used to calculate the support for every association. Table 1 is an example of a mapping table. The frequency is used to calculate the support. The support of a mapping is the ratio between its own frequency and the number of mapping observations of grid_a. The support of a mapping is later used to compare the similarity of two tracklets. [Table 1] shows an example of a mapping table. The support value is the division of a mapping's frequency and the value of all the frequency of similar mappings.

| GRID_A | GRID_B | FREQUENCY | SUPPORT |
|---|---|---|---|
| 32 | 45 | 24 | 0.41379 |
| 32 | 35 | 12 | 0.20690 |
| 32 | 54 | 8 | 0.13793 |

| | | | |
|---|---|---|---|
| 32 | 53 | 7 | 0.12069 |
| 32 | 44 | 7 | 0.12069 |

*Table 1 Mapping table example of all the mappings of Grid 32 of Camera A*

**Similarity metric.**

Similarity metric is used to measure the probability of two tracklets to be associated as a trajectory with an identical id. Similarity metric is calculated by comparing every time-intersecting detections between two tracklets. The association between every two intersecting detections is then fetched from the mapping table. When the association is high enough, then the two intersecting detections are considered as a strong association. A similarity metric is the degree of association between detections of two tracklets.

In the example shown in [Figure 1], the two tracklets have 6 intersecting detections. The support for every intersecting is then fetched from the mapping table. If the support value exceeds a certain threshold, then we could know that the association is strong. The ratio of strong detection associations to the number of detections is the similarity metric. The closer is the similarity metric to 1, the higher the probability that the two tracklets are the same object trajectory.

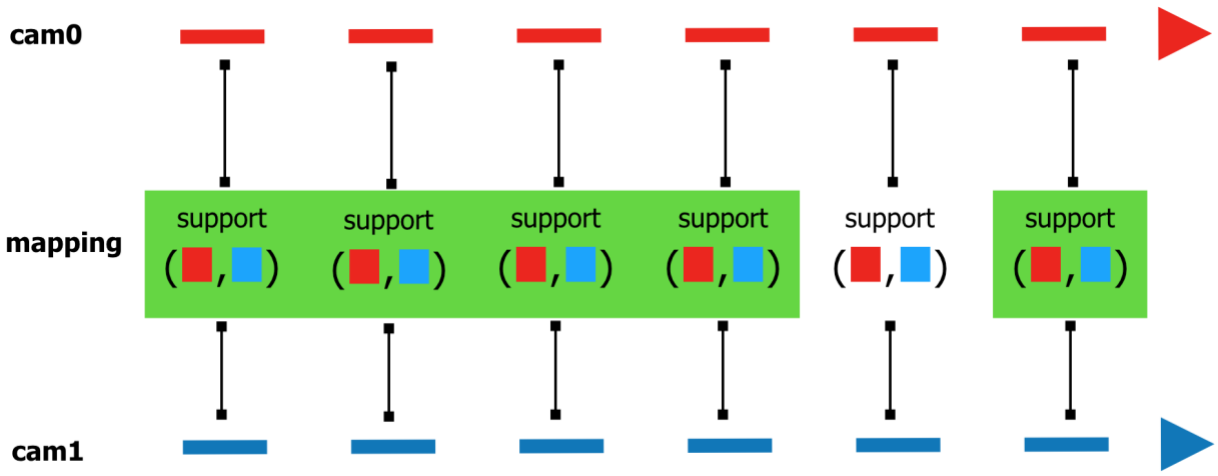


*Figure 1 Similarity metric between two tracklets example. Mappings with green background indicate mappings with a support score that exceed the minimum threshold value*

## Experiment

### *Experiment with small dataset.*

An experiment is conducted with a small dataset to determine the viability of this methodology. Three objects walk in a straight line between three columns, with two cameras filming. [Figure 2] shows the top bird's view of the scene.

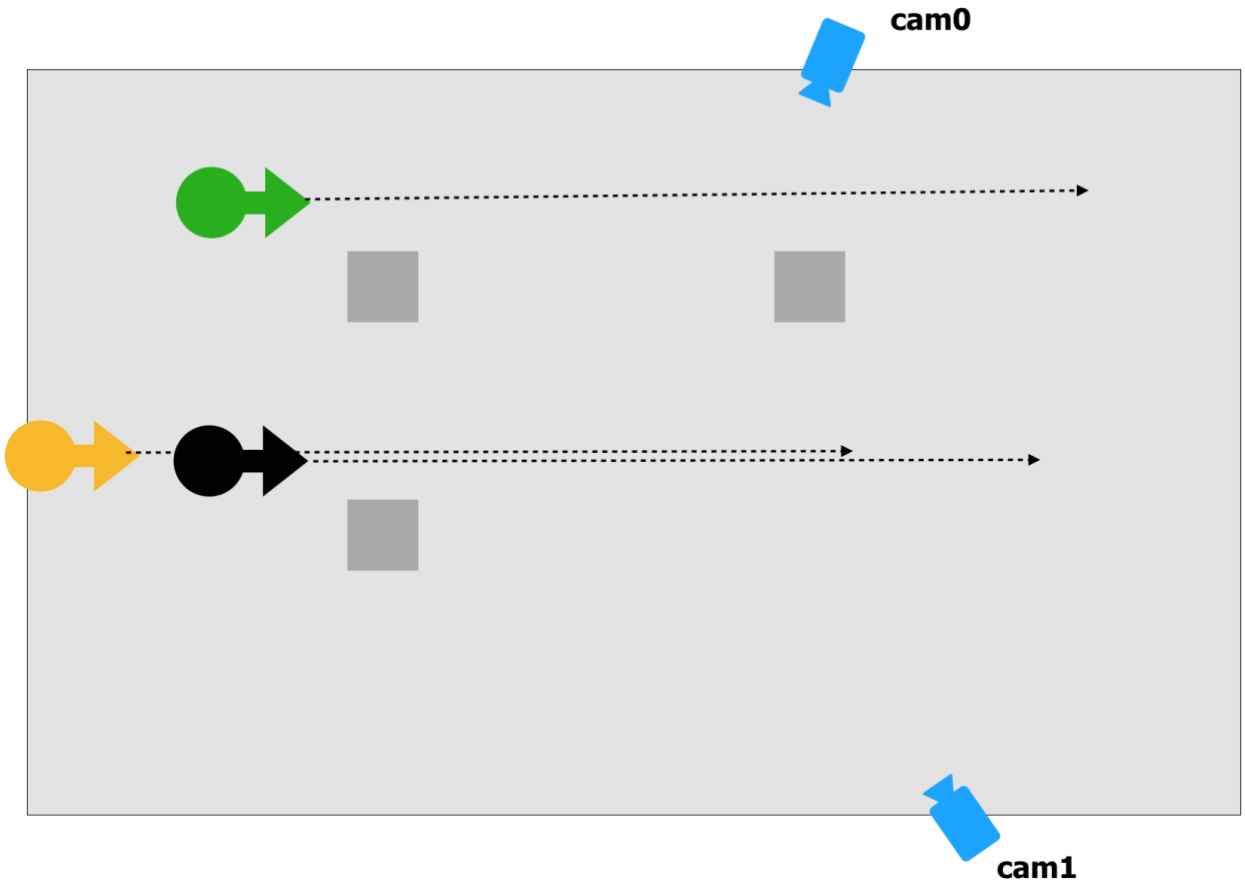


*Figure 2 Top bird's view scene of challenging dataset experiment*

Cam0 produced 7 tracklets that are fragmented while cam1 produced three perfect tracklets. The tracklets are then arranged as a weighted bipartite graph such that the tracklets are the nodes while their weights are the similarity metric between two tracklets.

The similarity metric between all tracklets are then calculated and shown as in the Table 2. From the weights of the graph we could easily assign tracklets with high association as the same object trajectory and compare with the ground truth. Comparison with the ground truth shows 100% accuracy. The first row and column of [Table 2] shows the tracklet id for cam1 and cam0 respectively.

| | **1** | **2** | **3** |
|---|---|---|---|
| **1** | 0.5581 | 0.0833 | 0.4167 |
| **2** | 0.3771 | 0.0 | 0.7288 |
| **3** | 0.0 | 0.6226 | 0.0 |
| **4** | 0.8667 | 0.0 | 0.8667 |
| **5** | 0.6538 | 0.0769 | 0.5384 |
| **6** | 0.0 | 0.0 | 0.0 |
| **7** | 0.0 | 1.0 | 0.0 |

*Table 2 Assignment results of experiment with small dataset*

***Experiments with challenging dataset.*** Another set of experiments is conducted with a more challenging dataset. Nine objects walk in a straight line with one big column placed in the middle. [Figure 3] shows the top bird's view of the scene.

The scene dependent nature of the mapping table methodology signifies that the location of the camera of which the scene was captured is important. Due to this fact, eight cameras are installed at the scene indexed from cam0 to cam7 as shown in [Figure 3].

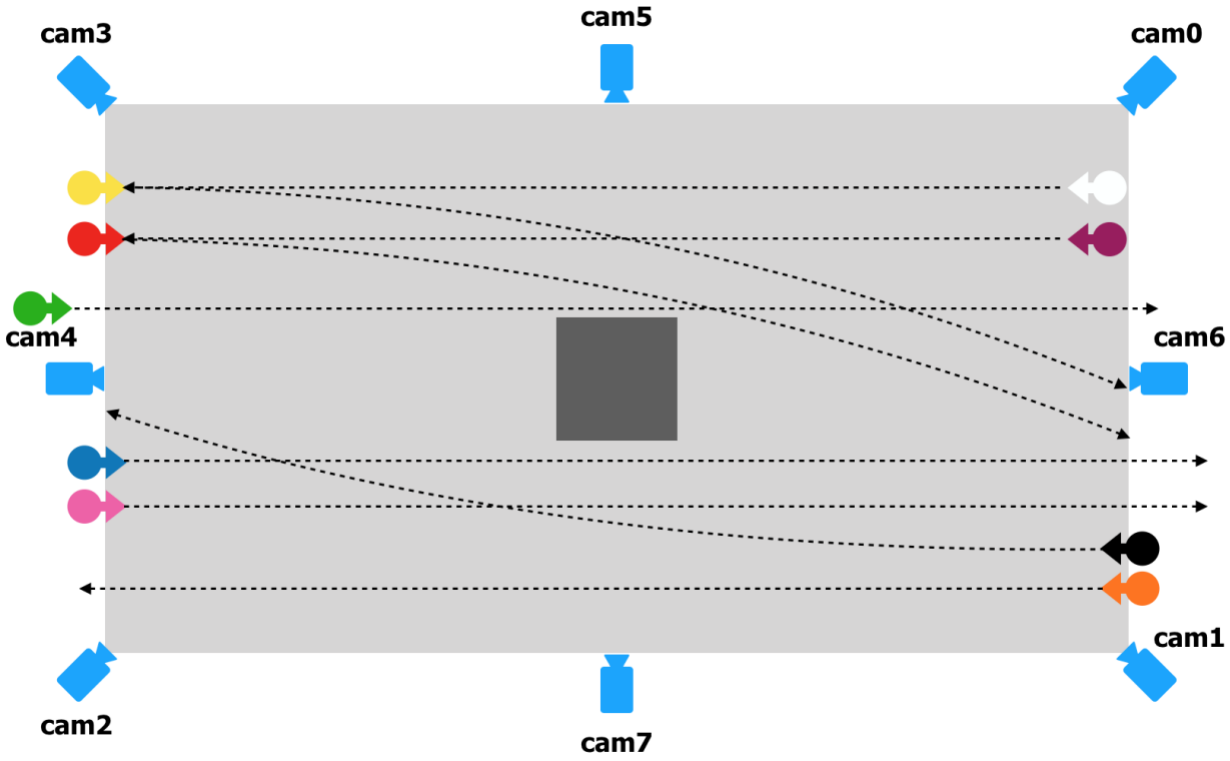


*Figure 3 Top bird's view scene of challenging dataset experiment*

Experiments are then conducted for every element of the subset of all combinations of two cameras. The assignment results by the algorithm are then compared with the ground truth. [Table 3] shows the accuracy results from the comparisons with the ground truth.

| CAM_A_ID | CAM_B_ID | ACCURACY |
|---|---|---|
| 0 | 1 | 50.00% |
| 0 | 2 | 66.67% |
| 0 | 3 | 83.33% |
| 0 | 4 | 75.00% |
| 0 | 5 | 66.67% |
| 0 | 6 | 50.00% |
| 0 | 7 | 41.67% |

*Table 3 Result of experiment with challenging dataset*

Results from [Table 3] shows that the combination of cam0 and cam3 achieved the highest accuracy. More experiments could be done using two different cameras at another different locations until the experiment produces a desirable result.

**Non-overlapping Field of View**

We also proposed another concept in order to solve the object association problem using multiple cameras, but which do not share the field of views. We called this, non-overlapping field module. For this module, we divided the view of the camera into certain number of grids, here we divided the view into 144 grids (12 x 12).

As we mentioned, we are using the log data from the video parser, called IntelliVix, which is provided by the database lab.

We already knew the relationship between the installed cameras which one is next to which.

**Temporal Model**

In order to establish the algorithm for the non-overlapping module, the temporal model is one of the features that we are using. So, the temporal model is using the time gap between the two adjacent cameras which are not sharing the field of view. First, at the beginning of the learning algorithm, we just used any time gap. After the significant amount of time, we can see the trends of the time gap for each grid. Then, we are using this trend for associating the object. Since we already knew the relationship between the cameras, we will use this model to identify the objects like this [Figure 4]. If the model comes out to the adjacent camera within the time gap, we will consider it as one of the possible candidates.

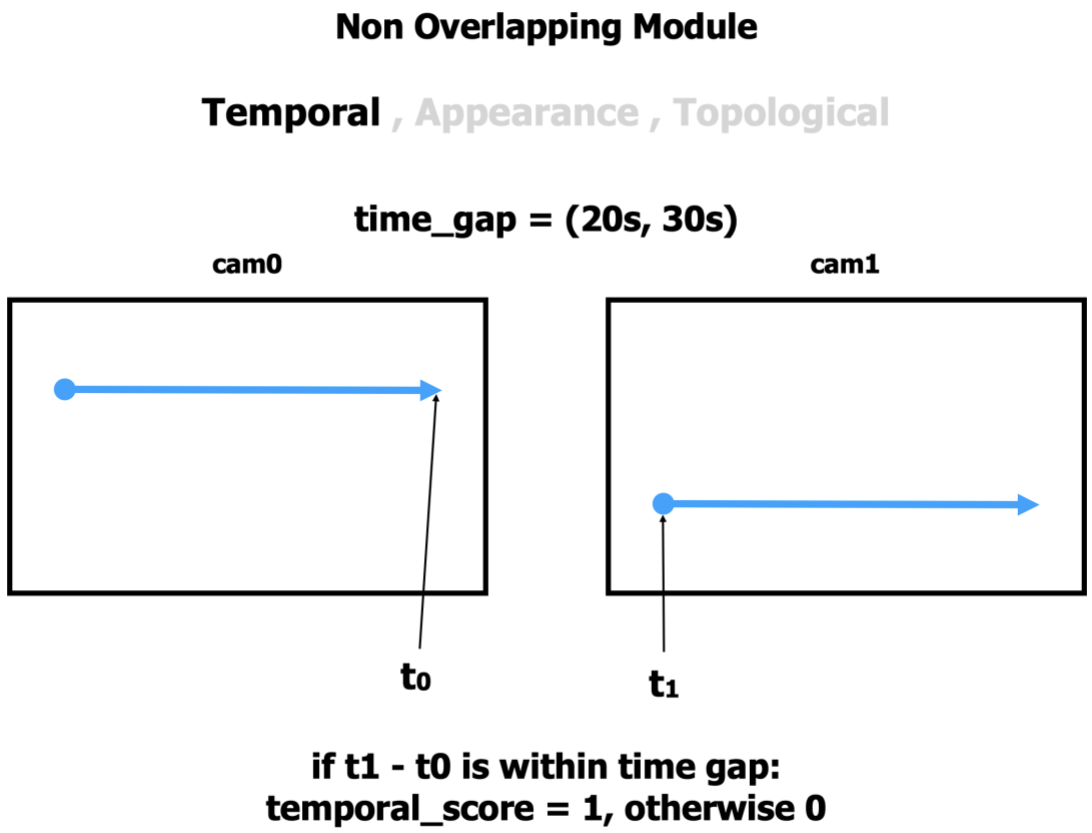


*Figure 4 Temporal model*

**Appearance Model**

The second model that we are using for this module is the appearance model. With the appearance model, we are retrieving the size variation and the speed variation of the objects detected by the video parser. In the learning phase, we will update this for every inserting log datum. The usage for the appearance model is same as given [Figure 5]. During the learning phase, we expect that the average variation rate of the size and the speed of the objects between two camera views can be retrieved. Then, using this retrieved variation rate, we will calculate the score for the confidence. The formulas are followed by [Figure 5].

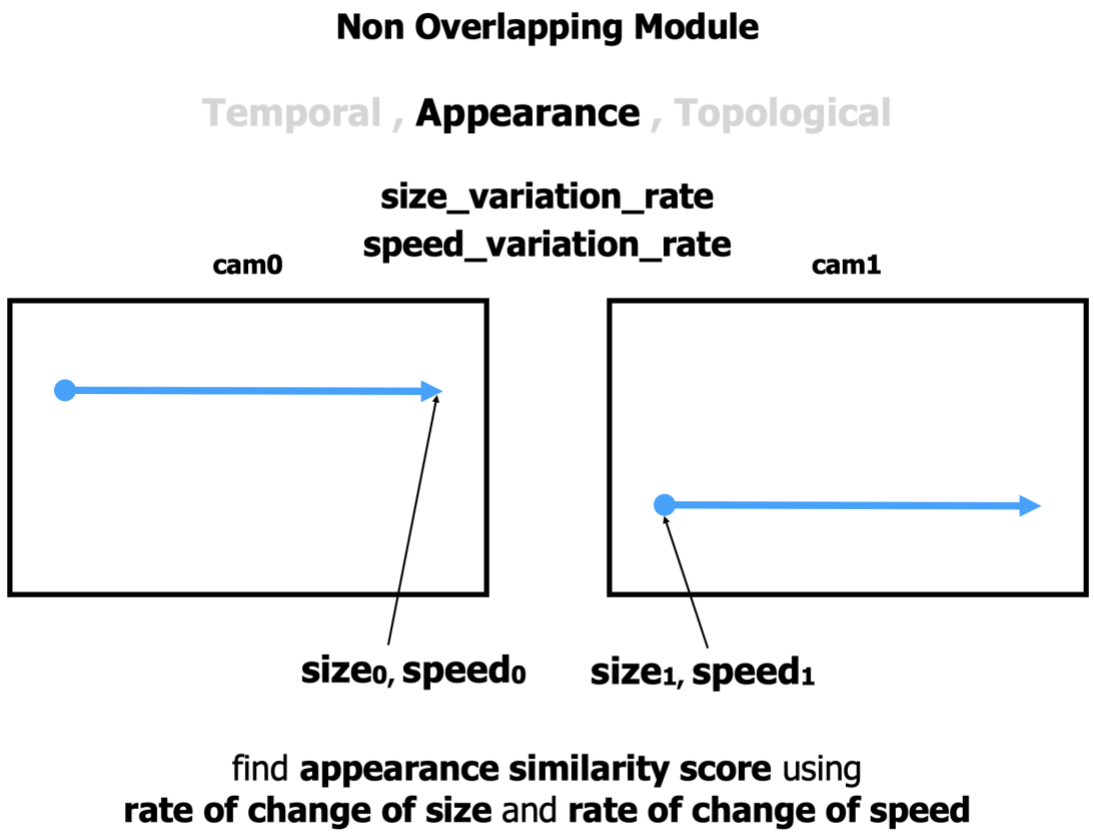


*Figure 5 Appearance Model*

$$estimated\_size_b = size\_variation\_rate \times real\_size_a$$

$$size\_similarity = 1 - \frac{estimated\_size_b - real\_size_b}{real\_size_b}$$

$$estimated\_speed_b = speed\_variation\_rate \times real\_speed_a$$

$$speed\_similarity = 1 - \frac{estimated\_speed_b - real\_speed_b}{real\_speed_b}$$

*Equation 1 Score formula using the speed and the size variation*

**Grid topological model**

The last model that we proposed for this module is the grid topological model. The grid topological model is using the topological properties of the grids. It's obvious that every object will use the different route. So, we can be sure that the objects will pass the different grid when they walk through the different cameras. However, even though they are using the different path, if the data are collected, we can see the trends of it. The following [Figure 6] is how we are using the topological model in here. One grid for cam0 will have many different connectable grids for cam1, this is the fact retrieved from the log data. But it has a trend. The stated score will be calculated by $\frac{the\ number\ of\ occurences\ to\ grid\ cam1}{the\ number\ of\ occurences\ from\ grid\ cam0}$.

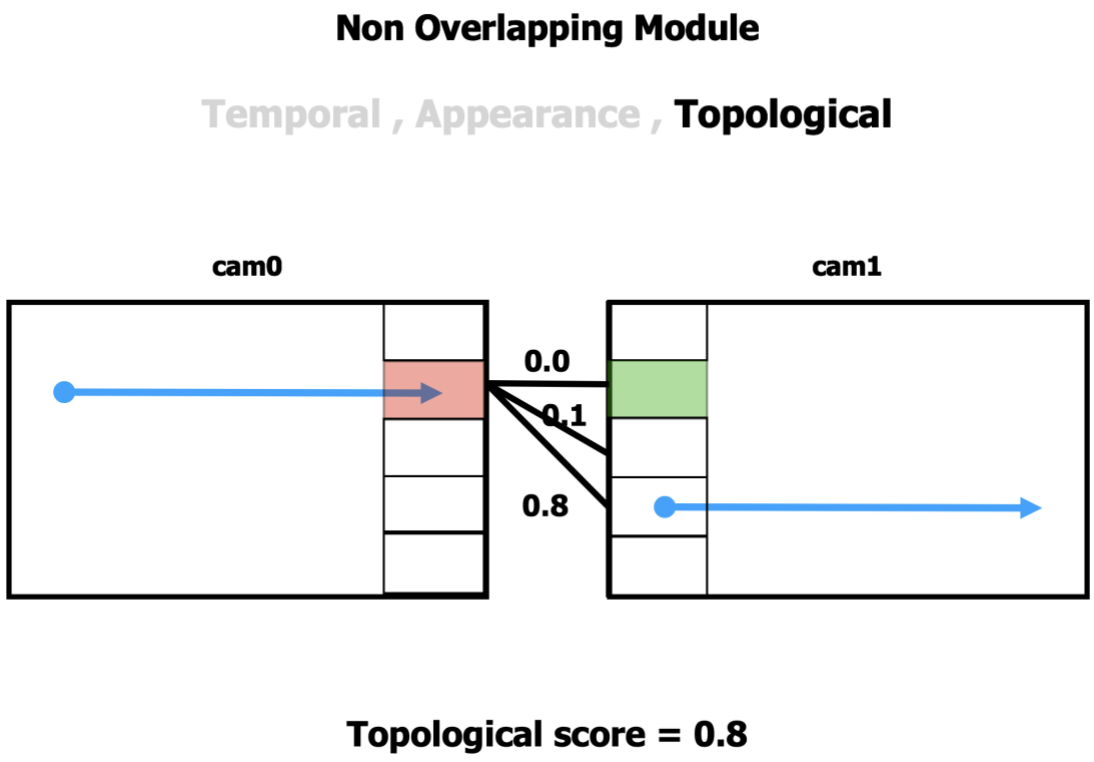


*Figure 6 Topological Model*

### Total Score

We will calculate the total score using the 3 different models by multiplying every scores. There are many ways to calculate the total score, but we decide to multiply all of these. It's because if one of the scores from each model comes out to be zero, which means it does not meet this condition, then it should not be connected.

## Experiment

We have used 40 minutes of video for the learning and 2 minutes for the monitoring. From this video, we retrieved 1760 rows of data from two cameras. And from that data we made the mapping table which is like following [Table 4]. The result of this module is followed by [Table 4].

| cctv_a | g_a | cctv_b | g_b | t_l | t_u | sp_v_r | sz_v_r | time_avg | time_std | count |
|---|---|---|---|---|---|---|---|---|---|---|
| 204 | 1103 | 205 | 1206 | 3628 | 5484 | 0.645 | 1.414 | 4555.556 | 463.980 | 9 |
| 204 | 1203 | 205 | 905 | 2679 | 4988 | 4.341 | 2.171 | 3833.333 | 577.350 | 3 |
| 204 | 905 | 205 | 1204 | 3043 | 4457 | 0.222 | 0.398 | 3750.000 | 353.553 | 2 |
| 204 | 1202 | 205 | 1103 | 3500 | 5500 | 2.709 | 1.344 | 4500.000 | 500.000 | 1 |
| 204 | 903 | 205 | 1105 | 1586 | 4414 | 0.611 | 1.412 | 3000.000 | 707.107 | 2 |
| 204 | 1002 | 205 | 1205 | 3000 | 5000 | 0.780 | 1.439 | 4000.000 | 500.000 | 1 |
| 204 | 1003 | 205 | 1104 | 1000 | 3000 | 1.214 | 1.237 | 2000.000 | 500.000 | 1 |
| 204 | 1205 | 205 | 1103 | 3500 | 5500 | 0.828 | 0.542 | 4500.000 | 500.000 | 1 |
| 204 | 1203 | 205 | 1105 | 4089 | 5244 | 3.262 | 2.303 | 4666.667 | 288.675 | 3 |
| 204 | 103 | 205 | 205 | 2500 | 4500 | 1.893 | 2.160 | 3500.000 | 500.000 | 1 |
| 204 | 1104 | 205 | 1205 | 4000 | 6000 | 0.868 | 0.808 | 5000.000 | 500.000 | 2 |
| 204 | 1103 | 205 | 1205 | 3401 | 5456 | 0.730 | 1.165 | 4428.571 | 513.553 | 14 |

*Table 4 mapping table for non-overlapping*

| t_a | c_a | g_a | tr_a | sp_a | sz_a | t_b | c_b | g_b | tr_b | sp_b | sz_b | trjtry | total_sc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4005 | 204 | 1203 | 1 | 16 | 15018 | 4050 | 205 | 1104 | 1 | 311 | 26720 | 1 | 0.064 |
| 4225 | 204 | 804 | 3 | 526 | 32472 | 4260 | 205 | 1204 | 3 | 133 | 6916 | 2 | 0.198 |
| 4400 | 204 | 1204 | 5 | 91 | 12740 | 4425 | 205 | 1005 | 5 | 347 | 23430 | 3 | 0.275 |
| 4575 | 204 | 1104 | 7 | 106 | 24143 | 4625 | 205 | 1205 | 7 | 36 | 13440 | 4 | 0.274 |
| 4755 | 204 | 903 | 9 | 460 | 22916 | 4785 | 205 | 905 | 9 | 629 | 22318 | 5 | 0.127 |
| 4935 | 204 | 1103 | 11 | 28 | 16046 | 4985 | 205 | 1106 | 11 | 241 | 33704 | 6 | 0.042 |

*Table 5 result of the non-overlapping module*

**Limitation**

Since we divide the view of the camera into 144 girds, it was very important to set the camera with the same position. However, unfortunately, we overlooked to fix the position of the camera. To be precisely, we fixed our camera, but it was tilted a bit. We still could use the data even though it has some noise. That was our one of the limitations.

Another limitation was lack of the data. It was really difficult to get the data for the learning, especially. We recorded 40 minutes of video, but in fact, it took 4 over hours to take because of the battery of the camera. We should charge our camera after each shot. And also, the place to record was another problem. We need the place where nobody was for the learning video. Because for the learning video only one man should be there. However, even though we record the video on the weekend, people were everywhere, so we had to cut the video and start it again for many times.

## Conclusion

We conducted this study in order to solve the object tracking problem not in the Computer Vision point of view but in the data mining point of view. And we constructed two different modules, the overlapping module and the non-overlapping module. Consequently, it was not really perfect, but we can see the prospect that it’s not only for the computer vision problem, it is a data mining

problem. From time to time, the importance of the data is getting important. If we can solve the limitation, we are sure that we can solve this problem more perfectly.