

École polytechnique de Louvain

Metric definition and optimization for the appearance information in Multi-Object Tracking

Author: **Henry GRAFÉ**

Supervisors: **Christophe DE VLEESCHOUWER, Vladimir SOMERS**

Readers: **Benoit MACQ, Christophe DE VLEESCHOUWER, Vladimir SOMERS**

Academic year 2022–2023

Master [120] in Electrical Engineering

Abstract

Multiple Object Tracking, or MOT, is a computer vision tasks, which consists in locating detections of multiple objects in video, obtaining information about these objects and associating them into trajectories. This task has many applications, including autonomous driving, pedestrian and crowd monitoring, or player localization in a sport setting. one of the important steps in MOT is feature extraction, where appearance information are obtained on the targets in order to help determine their trajectories. Currently, there are no ways to evaluate the capacity of a given set of extracted appearance information to enable good tracking performances, other than performing tracking directly. In this thesis, we proposes a new metric which evaluates such a thing. Our new metric evaluates a ranking of the set of all possible pairs of detections, ordered by distance between the appearance information distance between each elements of the pairs.

In addition, we propose a method for modifying the rankings of pairs of detections to improve the output of our new metric. For this, we used the concept of side information, which is the information outputted by a feature extraction model besides the appearance information and which can indicate us for example the certainty that we have that we can trust the appearance information. We tested our new metric on two datasets, and tried to improve these result using our ranking modification.

Acknowledgments

First, I would like to thank my supervisor, Professor Christophe De Vleeschouwer, for the support he provided during the development of this thesis. He gave me much advice and guidance and helped me refocus my thinking many times.

I would also like to thank Vladimir Somers for his continuous support during this thesis.

I would also like to thank my parents, as well as my brother, for their love and presence during all my studies.

Contents

0.1	General Idea	2
0.2	Thesis outline	3
1	Multiple Object Tracking and Person Re-Identification	5
1.1	Multiple Object Tracking	5
1.1.1	Detection stage	5
1.1.2	Feature extraction stage	6
1.1.3	Association stage	6
1.1.4	The MOT metrics	8
1.2	Person Re-Identification	8
1.2.1	Feature extraction	9
1.2.2	The Person Re-ID metrics	9
2	The pure appearance graph	15
2.1	Motivations for introducing the simplification	15
2.2	The pure appearance graph	17
2.3	Reformulation as a pair ranking	18
3	Comparing the feature vector's use in Person Re-ID entification and multi-objects tracking	20
3.1	The use of feature vectors in Person Re-ID and MOT	21
3.2	Toy situation	22
3.3	Comparison with pair Person Re-ID entification	25
3.4	Summary	27
4	Introducing the new metric	28
4.1	The concept of confidence	28
4.1.1	Examples in the literature	28
4.1.2	Confidence in the pure appearance graph	30
4.2	The general principle of the new metric	31
4.3	Definition of the metric	32
4.4	Practical example	33
4.5	Interpretation of the metric	36

5	Improving the new metric curve using side information	37
5.1	Comparing two new metric curves	37
5.2	Method for improving the ranking using side information	38
5.2.1	The concept of center of the ranking	39
5.2.2	Mathematical description of the method	40
6	Experiments	43
6.1	Market-1501 and VSNet	43
6.1.1	Market-1501	43
6.1.2	VSNet	45
6.1.3	Making a pair tracking dataset from Market-1501	46
6.2	Artificially generated dataset	47
7	Results	50
7.1	Results on the artificial data	50
7.1.1	Analysis of the new metric	50
7.1.2	Results of the ranking improvement	51
7.2	Results on Market-1501 and VSNet	52
7.2.1	Analysis of the new metric	52
7.2.2	Results of the ranking improvement	52
8	Conclusion	55
8.1	Summary of the work	55
8.2	Perspective for future works	56

Introduction

0.1 General Idea

[3] Multiple Object Tracking (MOT) is a mid-level computer vision task consisting in locating multiple objects, maintaining their identities and yielding their individual trajectories. This task has many applications, including autonomous driving, pedestrians and crowd monitoring, or player localization in a sport setting. In this master thesis, we will focus on pedestrian tracking.

MOT is very complicated to perform due to, among others, frequent occlusions due to the crowd, tracks initialization and termination, similar appearance among objects, size of object which might be very small, and interaction of objects between each others.

There exists many formulations and approaches for doing MOT, and the literature on the subject tries a wide variety of methods. The standard approach is called “tracking-by-detection”[4]. It consist in extracting detections from the video frame and trying to associate them using motion and appearance information.

In order to better visualize what this approach is composed of, we will inspire ourselves of [4] to divide MOT tracking-by-detection approach into 3 steps: the detection stage, the feature extraction stage, and the association stage. In reality, MOT algorithms do not follow these steps sequentially, and often iterate over them or skip some of them. It is however still a good way to visualize what an algorithm is doing in MOT :

1. Detection: At the detection stage, an object detection algorithm analyzes the video frame by frame to identify objects belonging to the targets we want to track, and delimit them using “bounding boxes”, a set of coordinate that locate where the target is and its size in the image. A detected object, associated with the frame the detection was made at and the location of the bounding box is called a “detection”.
2. Feature extraction: At the feature extraction stage, a feature extraction algorithm analyzes the object that was detected to extract appearance information. Motion information can also be extracted using detections over several frames of the video.
3. Association : In the association stage, the appearance and motion information that was extracted in the last step, as well as the time and location of the

detections, are used to determine the trajectories of the objects by assigning the same label to detections that are of the same target.

This work focuses on the use of the appearance information produced at the feature extraction stage for the association stage. More precisely, it is interested in using the appearance information that has been produced by feature extraction algorithms that were not designed originally for MOT, such as the ones that were designed for person Re-Identification.

The first goal of this work is to design a new metric that evaluates the quality of the appearance information for doing association on MOT. Secondly, we would like to improve the output of this new metric we designed using what we call side information, which is the information outputted by some feature extraction algorithm that complements the extracted appearance information. By determining what we call a certainty coefficient using this side information, we would like to improve the ordering of all the possible pairs of detection by distance between their feature vectors to improve the output of our new metric.

0.2 Thesis outline

We will first describe in details the tasks of Multiple Object Tracking and Person Re-Identification. Then we will introduce an assumption on the form of the appearance information when it is used in the association step of MOT. Using this assumption, we will compare the use of the feature vectors in person Re-ID and association in MOT.

This will outline that the two tasks are not the same, and that a specific metric needs to be designed to specifically characterize the capacity of a given set of appearance features to lead to good association in MOT.

In order to design this new metric, we will first analyze what are the principal needs of an association algorithm when it comes to appearance information. Then, we will introduce our new metric that tries to evaluate the appearance information based on those needs. We will also illustrate it with a simple case.

Then, we will introduce a method to try and improve the output of our new metric, by modifying the ranking of pairs, based on what we call the side information, from which we derive what we call certainty coefficients.

After this, we will try our new metric and our custom method for improving it on two different sources of data : first an artificially generated data set including side information, and second a pseudo-tracking dataset made with the features extracted by VSNet ([2]) from the images of the test set of the Market-1501 ([19]) Person Re-ID

dataset, which also includes side information generated by VSNet that we use to improve the ranking.

Finally, we will finish with a conclusion which summarizes the thesis and offer perspectives for future works.

Chapter 1

Multiple Object Tracking and Person Re-Identification

Introduction

In the introduction chapter, we introduced the tasks of Multiple Object Tracking. In this chapter, we will explain Multiple Object Tracking in more details, and describe the task of Person Re-ID in details. We will list the different steps for both tasks, as well as the metrics used to evaluate them.

1.1 Multiple Object Tracking

We defined Multiple Object Tracking (MOT) in the introduction. As we also said, we inspired ourselves of [4] to divide Multiple Object Tracking into 3 steps: the detection stage, the feature extraction stage, and the association stage. In this section, we will detail all these steps. Finally, we will describe the metrics used to evaluate MOT.

1.1.1 Detection stage

At the detection stage, each frame of the video is analyzed to find the targets on it. As we said, a detected object, associated with the frame the detection was made at and the location of the bounding box is called a “detection”. Figure ?? illustrate person detection in a video.

Currently, the best performances in object detections are achieved using deep learning algorithm, the current state of the art being yolov7 [12].

Ensuring good detections is an efficient way of improving tracking performances, but represents a separate stage from the feature extraction and association stage that come next. Because the quality of the detector can strongly influence the tracking performances, many MOT datasets already contain the pre-made detections to be used for feature extraction and association, and this stage is skipped.

1.1.2 Feature extraction stage

The feature extraction step is when we compute motion and appearance information from the detections, for further use in association. Appearance and motion information extraction are an important step in MOT. The quality of both information is what leads to good association, and thus to good tracking.

Motion information is usually extracted from simple information about the detection, like the frame the detection was made at and the location of the frame in the image. These information, over several frames, allow us to compute the speed of the detection [6]. The way the motion information is used for association is described in more details in chapter 2.

The state of the art in appearance feature extraction is achieved by deep learning techniques, and more precisely convolutional neural networks (CNN). Examples include [14], [15], [16], and [13], which use various CNN architectures. Of these 4 works, all of them trained the feature extraction models on Person Re-Identification datasets. It is to be noted that, in all of these papers, the feature extraction algorithm was part of a larger MOT framework, including also an association method and sometimes a detection method. In MOT literature, appearance feature extraction is not the only focus, as there is also the association stage.

1.1.3 Association stage

The association step is when we determine which detection belongs to which trajectory. The literature explores many different methods and algorithms for the association stage. Before presenting some of them, we will introduce two commonly used terms in association ([3]):

- Trajectory : a trajectory is the final output of a MOT system. One trajectory corresponds to one target, and is composed of multiple detections in an image sequence
- Tracklet : a tracklet is an intermediate between a detection response and a trajectory. It is composed of several detection responses that are believed to be from the same target. In fact, a detection response can be viewed as a tracklet composed of only one detection response.

Association methods are usually divided into two main categories: offline (batch) and online[4]. In the offline setting, the determination of trajectories is done with the detections at all the time frames at hands, while in the online setting the algorithm

doesn't have access to future frames to determine trajectories. Since they have access to more information, the offline methods usually yield better results than the online methods.

Online association methods include Kalman filters to predict motion and formulate the association of the prediction with the current frame's detection as a assignment problem, solved using the Hungarian algorithm [14]. Multi-Hypotheses Tracking, or MHT, can also be used instead of the Hungarian algorithm. It consist in delaying the decision of assigning a detection to a trajectory by propagating multiple hypotheses to the next frames [17].

Offline association methods are usually graphed-based solutions. This means the problem is represented as a graph, where the nodes correspond to detections (or in some case like [6] to tracklets), and each edge gets a weight that correspond to the distance/similarity between two detections, in term of spatio-temporal likelihood the two detections are of the same object, or in terms of appearance similarity, i.e. the similarity between the appearance information extracted in the last stage. Then, the association algorithm tackles the association by processing the graph. We can also note that some graph-based solutions propose an online variant of their method, like in [6] and [11].

This formulation of the association problem as a graph varies through the works. Label propagation on complementary graphs [11] for example, create several separate graphs: one for each main appearance feature, one for spatio-temporal proximity incorporating the motion information, and an exclusion graph which ensure detections that have been made at the same frame will not be associated to the same trajectory. The appearance and spatio-temporal graph's weights indicate the likelihood that two detections belong to the same object. The optimization is seen as a label assignment problem trying to minimize the "labeling energy" for the spatio-temporal and appearance graph, and maximize it for the exclusion graph.

Iterative Hypothesis Testing [6] create only one graph, where the edge weights represent spatio-temporal proximity, and the appearance similarity information is contained in the nodes. The graph is modified at each iteration of the algorithm, as nodes get aggregated together into larger and larger tracklets.

In GMCP-tracking [18], a graph is created, where the appearance weight between detections at the same frame are zero, and the edge weights between detections at different frames represents their appearance similarity only. The problem is solved as a minimization problem over a few short frames, trying to minimize a cost function depending on appearance and motion information, where the motion information is included not in the graph but in the cost function.

1.1.4 The MOT metrics

To evaluate the performances of tracking, the most commonly used methods are the CLEAR MOT metrics [8]. There are two main metrics : the Multiple Object Tracking Precision (MOTP), and the Multiple Object Tracking Accuracy (MOTA). MOTP evaluates the quality of the detection of the objects, regardless of yielding their correct trajectories, and MOTA evaluate the quality of the determined trajectories. Since our focus is not on the detection stage, we will only describe MOTA.

When evaluating the result of a tracking algorithm, we compare the computed trajectories with the ground truth, which are the given set of bounding boxes at each frames that are considered to be correct. To know if a bounding box of a computed trajectory matches the ground truth box, we use the intersection over union (IoU).

There are 4 types of errors that can be done in tracking :

- Misses: When an object that has a ground truth box is not assigned to a trajectory.
- False positive: When a bounding box that is assigned to a trajectory does not correspond to any ground truth box
- Mismatch: When a ground truth of a same object that was assigned a label the frame $t - 1$ gets assigned a different label at frame t .

MOTA is defined as :

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \in] - \infty, 1] \quad (1.1)$$

where m_t , fp_t , mme_t , and g_t are the number misses, false positive, mismatches and total ground truth box at frame t , respectively. Since there can be more errors than ground truth boxes, the MOTA can be under 0.

1.2 Person Re-Identification

Person re-identification is a computer vision task, which consists in matching the image of a person of interest, called the query, with a large collection of images of persons, called the gallery ([1]). The objective is to find the images in the gallery that are of the same person as the query image. An example of a query and a gallery set is presented in figure 1.1 (a).

Person re-identification is considered a very challenging task, because of the many situations that can arise : inaccurate bounding boxes, varying viewpoint of a person,

pose variation of a person, luminosity change of the person, occlusions by object or other persons of the person, and poor quality (noise) of the person’s image. With all these artifacts, recognizing the same person in two different images from two different cameras at two different times and with the person in a different position,, and under different illumination as well as potentially occluded can be difficult ([2]).

Currently, state of the art approaches to solve this problem rely heavily on deep learning to extract feature vectors from the images to represent the person and compare it to other images ([1]).

In this section, we will detail the feature extraction process, the concept of query-gallery ranking, as well as present the metrics used to evaluate Person Re-ID.

1.2.1 Feature extraction

Person Re-ID is generally represented as a representation learning task : each image is processed by a given model to extract a feature vector describing the image, and then the similarity/distance between two images is computed as the similarity/distance between the two image’s feature vectors ([2]). This is illustrated in figure 1.1 (b).

Currently, state of the art performance is achieved using deep learning solutions, and more precisely Convolutionnal Neural Networks [9]. Researchers are trying to create models that are able to detect appearance information in the images of persons using specific neural network architecture. For example, [10] develops a CNN architecture that can detect cues at different scales of the image. Some works like [2] try to handle occlusions in the image by extracting feature vectors for each body part, and only use the feature vectors of the body parts that are visible. Other works try to experiments on the types of loss functions for the training of the deep neural network to obtain the best results [9].

1.2.2 The Person Re-ID metrics

The Person Re-identification task is based on a query-gallery setting. The task is formulated as ranking all the gallery images by similarity to a query image. The task is considered to be performed perfectly if the gallery images that are of the same person as the query are on the top of the ranking. An example of such a ranking is shown in figure 1.2.

To order the gallery images by similarity to the query, we take feature vectors that have been extracted from each image in the gallery, as well as from the query image, and the compute the similarity between the query and each gallery image. The gallery images are then ranked by similarity.

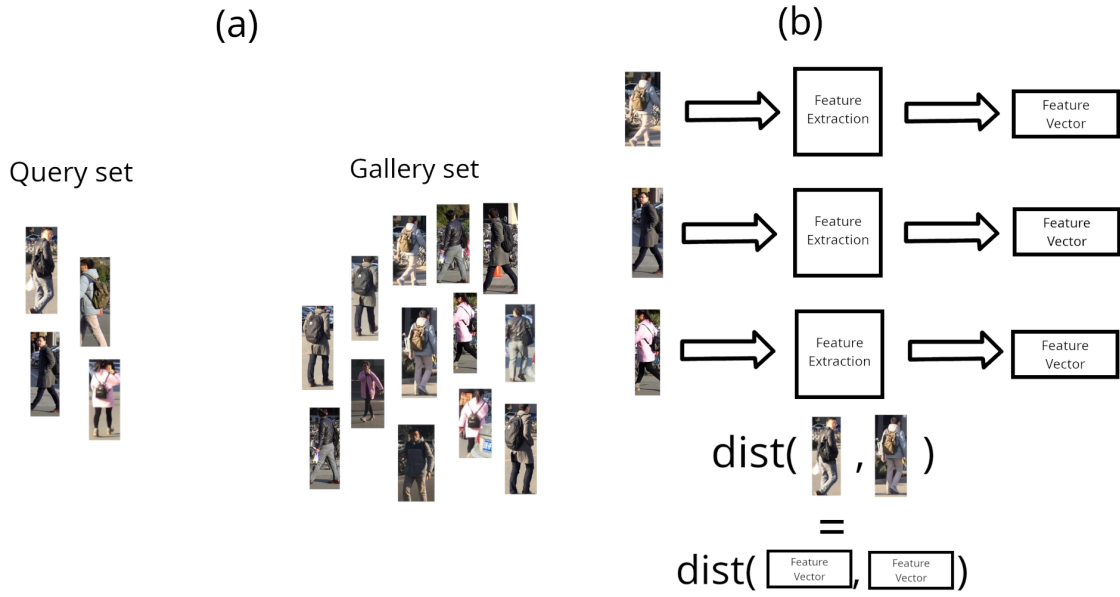


Figure 1.1: (a) An example of a query and gallery set. The task of Person Re-Identification consists in finding for each query image the images in the gallery set that are of the same person. (b) Illustration of the feature extraction step in Person Re-ID: images go through a feature extraction model, which outputs a feature vector. This feature vector is then used to compute the similarity between two images by computing the similarity between the two feature vectors.

The evaluation of the task of Person Re-Identification is based on the creation of query-gallery rankings for every query image. Thus during the testing of Person Re-ID performances, we make as many rankings as there are query images.



Figure 1.2: Example of an ideal query-gallery ranking. Every gallery image is ranked by distance to the query image, which is indicated under each gallery image. Green cadres indicate that the gallery image is of the same person as the query image. Red cadres indicate the gallery image is of a different person than the query image. As we can see, all the gallery images that are the same as the query are on top of the ranking in this ideal case.

Two main metrics exist to evaluate how close all the query rankings are to the ideal ranking, and thus evaluate the performance of a given feature extraction model used to produce the rankings. These metrics are the cumulative matching characteristic (CMC) and the mean average precision (mAP).

Cumulative Matching Characteristic

The rank- k Cumulative Matching Characteristic (CMC) of a ranking is a metric allowing to evaluate how well a feature extraction model can at least put one image of the gallery that is of the same person as the query at the top of the ranking. This metric is computed like this:

$$cmc_k = \begin{cases} 1 & \text{if the firsts } k \text{ gallery images contain the query person} \\ 0 & \text{otherwise} \end{cases} \quad (1.2)$$

The rank- k CMC is the average of this over all of the rankings. The cases where this metric reaches 100% do not mean that the rankings are perfect. Only one correctly identified image needs to be in the top- k images of the ranking in order for the metric to be equal to 1. An example of a ranking that is not ideal but whose rank-1 CMC is equal to 1 is shown in figure 1.3. Usually the rank-1, rank-5 and rank-10 CMC are computed and displayed.



Figure 1.3: Example of a ranking with a $cmc_k = 1$ for any k , despite having two of the three gallery images with the person from the query at the very bottom of the ranking and thus being far from the ideal ranking.

Mean Average Precision

The average precision (AP) of a given ranking is computed with the following formula:

$$AP = \frac{1}{N_{TP}} \sum_{k=1}^{N_G} \frac{TP_k}{k}$$

with

$$TP_k = \begin{cases} \# \text{ of images of query ID in the first } k \text{ images} & \text{if image } k \in \text{query ID} \\ 0 & \text{otherwise} \end{cases} \quad (1.3)$$

The mean average precision (mAP) is the mean of the AP over all the query images.

This formula is not obvious, so we shall explain where it comes from. In order to do that, we introduce some vocabulary. Let us consider a ranking, where we are attempting to determine which images in this ranking are of the same identity as the query image. To do that, a classifier will label a given share of the image with the lowest distance as “positive”, meaning classified as being of the same identity as the query, and the rest of the images as “negative” meaning classified as a different identity as the query. We introduce the following quantities:

- *TP*: True Positive: the number of gallery images that have been classified “positive” and are indeed of the same identity as the query image.
- *FP*: False Positive: the number of gallery images that have been classified “positive” and are actually **not** of the same identity as the query image.

- *TN*: True Negative: the number of gallery images that have been classified “negative” and are indeed of a different identity than the query image.
- *FN*: False Negative: the number of gallery images that have been classified “negative” and are actually **not** of a different identity than the query image.

Now, with these values, we can compute the performance of this given classifier. The common metrics used here are called the precision and recall. Precision is the proportion of images classified as of the same person as the query that are actually of the same person as the query:

$$\text{precision} = \frac{TP}{TP + FP} \quad (1.4)$$

Recall is the proportion of images that have been classified as of the same person as the query over the total number of images that are of the same person as the query:

$$\text{recall} = \frac{TP}{TP + FN} \quad (1.5)$$

Precision and recall metrics characterize the performance of this classification. Figure 1.4 gives an example of the computation of those metrics for a threshold classifying the first 3 images of the ranking as “positive”.

To do the classification, the classifier has at its disposal the ranking by distance of the gallery images. The classifier can choose to classify a certain share of the images with the lowest error rate as positive, and the rest as negative. We move a decision threshold over the ranking that modulate this decision, and we get a series of precision and recall value as we move through the ranking. These value can be plotted on a 2d space, where the x axis is the recall and the y axis the precision. We can see a curve, called the precision-recall curve, is formed. An example of such a curve for an example ranking is available at figure 1.4, as well as the computation of the precision and recall for one point of the curve.

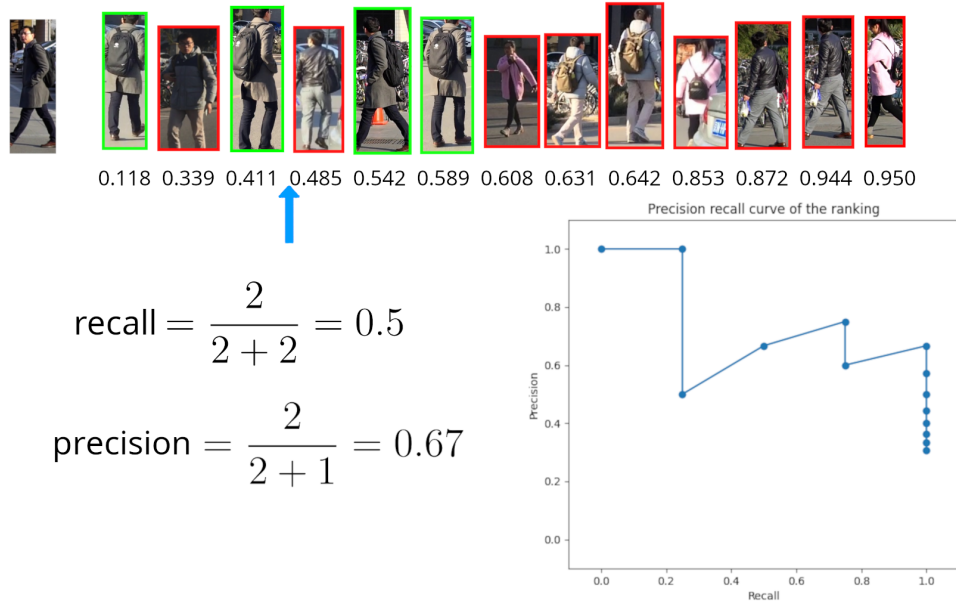


Figure 1.4: Explicit computation of the precision and recall for a threshold (the blue arrow) classifying the first 4 images of an example ranking as “positive”. The precision-recall curve of this ranking is also shown. The vertical series of points at the end of the ranking correspond to when the threshold reaches the end of the ranking, where there are no more true positives, so the recall stays at one but the precision falls down.

The average precision (AP) is the area under the precision-recall curve. The equation 1.3 above encapsulates the computation of this area based on the position of the images in the ranking.

Chapter 2

The pure appearance graph

Introduction

In this work we will be discussing how the appearance feature vectors are used in the association step of MOT.

In this chapter, we will introduce the pure appearance graph association setting, which we will use as our simplified baseline for how the appearance information is encoded for the association step in MOT for the rest of this thesis. We will describe what motivates us to introduce this simplification, and we will justify its introduction by some of the association methods in the literature we described in chapter 1.

2.1 Motivations for introducing the simplification

In the last chapter, we saw that the association step was composed of two things: the formulation of the problem, usually in the form of some sort of graph, and the association algorithm using the information contained in the formulation to determine the trajectories. As we saw, there exists many different formulations of the association problem. However, we saw that all of them used two types of information:

- The motion information, in the the form of the time frame a detection was made at, and the location of the detection in the image. This information is used to compute the likelihood that two detections are of the same target by producing a motion prediction from a tracklet, and seeing if motion prediction match a detection at the next frame. For example, Iterative Hypothesis Testing [6] assigns a weight of a directed edge between two nodes (which are tracklets) using this formula:

$$w_{uv} = \begin{cases} [1 + \gamma(t_v^{(s)} - t_u^{(e)} - 1)]g_{sp}(u, v) & \text{if } 0 < t_v^{(s)} - t_u^{(e)} \leq \tau_{max} \\ \infty & \text{otherwise} \end{cases} \quad (2.1)$$

$t_v^{(s)}$ and $t_u^{(e)}$ are respectively the times of the first detection of the tracklet v and of the first detection of the tracklet u . $\gamma > 0$ is a penalty parameter for missed

detections, and $g_{sp}(u, v)$ is the distance between v and the predicted position of the object corresponding to node u :

$$g_{sp}(u, v) := \|\mathbf{y}_v^{(s)} - \mathbf{y}_u^{(e)} - \dot{\mathbf{y}}_u^{(s)}(t_v^{(s)} - t_u^{(e)})\|_2$$

$\mathbf{y}_v^{(s)}$, $\mathbf{y}_u^{(e)}$, $\dot{\mathbf{y}}_u^{(s)}$ are respectively the position of the first detection in tracklet v , position of the last detection of tracklet u and speed at the end of tracklet u , computed using the last two detections of the tracklet or 0 if the tracklet is made of only one detection.

We can see that w_{uv} reflects the likelihood that target u , if it continued its course, would yield a target that correspond to the first detection of target v . This one example of how motion information is included in a graph.

- The appearance information, in the form of the distance between feature vectors corresponding to two detections or tracklet. For example in Generalized Minimum Clique Problem-Tracking [18], the appearance information are color histograms and is encoded in the graph in the form:

$$w(v_m^i, v_n^j) = \sum_{l=1}^8 k(\phi_{ml}^i, \phi_{nl}^j)$$

Where v_m^i is the m th detection made at frame i , v_n^j the n th detection made at frame j , and ϕ_{ml}^i and ϕ_{nl}^j are the histograms of the l th body part for the detections v_m^i and v_n^j , respectively. k represents the histogram intersection kernel (similarity between histograms).

We can see that $w(v_m^i, v_n^j)$ reflects the appearance similarity between two nodes in the graph using their color histogram.

As we can see, both motion information and appearance information are included in the graph formulation of association. This means that the subsequent association algorithms developed in the papers we presented have at their disposal both motion and appearance information, in one form or another, to find the trajectories. This means that when trying to solve the association problem, the association algorithm will give more or less importance to the motion or the appearance information depending on the situation.

For example, in Generalized Minimum Clique Problem-Tracking [18], the determination of an intermediary tracklet using the graph is done by optimizing a cost function which is:

$$\hat{\mathbf{V}}_s = \underset{\mathbf{V}_s}{\operatorname{argmin}}(\gamma_{\text{appearance}}(\mathbf{V}_s) + \alpha\gamma_{\text{motion}}(\mathbf{V}_s))$$

where $\mathbf{V}_s$ is a subset of detections of the graph aiming at identifying one target. $\gamma_{\text{appearance}}(\mathbf{V}_s)$ and $\gamma_{\text{motion}}(\mathbf{V}_s)$ are the appearance and motion cost of selecting this tracklet. α is a variable that modulates the importance of the motion information.

Another example is label propagation on complementary graph [11], which tries to optimize a label assignment problem, and which tries to minimize a “labeling energy” function which depends both on the spatio-temporal (i.e. motion) and appearance information.

In summary, there are many different ways the appearance information is encoded in the graph formulation of the association problem, often along the motion information.

To be able to discuss the use of the appearance information in association, we will introduce a simplified formulation of how the appearance information is encoded before undergoing association, that separates it from the motion information, so that we can discuss the appearance information present in this simplified formulation.

2.2 The pure appearance graph

We will now describe the simplified formulation of the the appearance information for association we decided to adopt. We will represent the association problem as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ where

- $\mathcal{V}$ is the set of nodes, with each node corresponding to a detection
- $\mathcal{E}$ is the set of edges, containing all possible undirected edges linking two different nodes.
- $\mathbf{W}$ is the set of weights, that assign to each edge a weight corresponding to the similarity between the appearance information of the two detections.

$$w_{ij} = \text{distance}(\mathbf{f}_i, \mathbf{f}_j)$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the appearance feature information extracted from the detection corresponding to node i and j , respectively, and $\text{distance}()$ is the distance function used to compute the distance between two extracted appearance information.

An example of such a pure appearance graph of 4 detections is available in the next chapter in figure 2.1 (a).

This pure appearance graph formulation is inspired by label propagation on complementary graph [11], where, as we described before, the situation is represented with separate graphs, some of them being appearance only. But on the contrary of this paper, where only the nodes in a certain neighborhood are linked with an edge and assigned a weight, every node are linked together with their appearance distance.

2.3 Reformulation as a pair ranking

The information contained in the pure appearance graph are the weights of its edges. Since the graph is undirected and fully connected, there is a weight assigned to each pair of nodes, and thus to each pair of detection.

We can thus reformulate the information present in the graph as the set of all possible pairs of detections and their associated weights. An example of the reformulation of the pure appearance graph as a set of pairs is available in figure 2.1 (b).

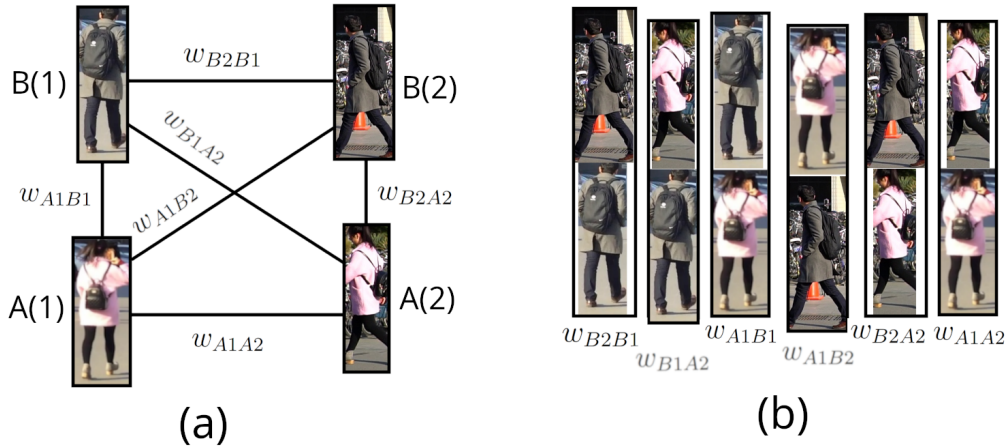


Figure 2.1: (a) An example of the pure appearance graph. The graph is fully connected, with the edge weight indicating the distance between the feature vectors of the two nodes. The graph does not contain any motion information. The letter (A or B) indicates the person's identity. (b) the formulation of the information contained in the graph in the form of a set of pair. Since there are 4 detections, there are 6 possible pairs of detections, with their associated weights.

We can easily compute the number of pairs a set of pairs corresponding to a pure graph contains:

$$N_{pair} = \frac{N_{node}(N_{node} - 1)}{2} \quad (2.2)$$

Where N_{node} is the number of nodes in the pure appearance graph, corresponding to the number of detections, and N_{pair} is the number of pairs of detections in the set.

Furthermore, since each pair has a weight, we can order this set of pairs according to them, which means ordering the pairs by distance. What we obtain is a pair ranking. For the rest of this work, we will assume in the pair ranking that the pairs that have the smallest distance are on the left, and the biggest distances on the right.

Without loss of information, we reformulated the information present in the pure appearance graph as a pair ranking.

Summary

In this chapter, we introduced the notion of the pure appearance graph, which is an simplified formulation of the form of the appearance information when it is used for association in MOT. The pure appearance graph is a graph where the nodes are detections, which is fully connected, where and the edges have a weight corresponding to the distance between the appearance features of the two nodes. We have seen that we can reformulate the information contained in the pure appearance graph as a ranking of pairs ordered by distance.

Chapter 3

Comparing the feature vector's use in Person Re-ID entification and multi-objects tracking

Introduction

Fundamentally, the appearance feature extraction stage in Person Re-ID and MOT serve the same purpose: from a detection, extract a feature vector describing the person, such that the feature vector is independent of different artifacts like luminosity, orientation, occlusion, etc.

In chapter 1, we saw how the feature vectors were used in Person Re-ID to produce rankings. In chapter 2 we introduced a simplification on the form that the appearance information takes when doing association in MOT, in the form of a pure appearance graph and thus of a pair ranking.

This chapter aims at demonstrating that this difference in the use of the feature vectors leads to a difference in their expected properties. Consequently we cannot assume feature vectors leading to good performances in Person Re-ID also leads to good performances in the association step in MOT.

We will first describe how the feature vectors are used in the two settings, Person Re-ID and rankings, and association in MOT and the pure appearance graph. Then we will show via a toy example that a given set of feature vectors can give perfect results in Person Re-ID but not perfect results in the pair ranking corresponding to the pure appearance graph. We will then make a link with the task of pair Person Re-ID entification, developed in [9].

3.1 The use of feature vectors in Person Re-ID and MOT

Appearance feature vectors in Person Re-ID

As we saw before, Person Re-ID operates on a query-gallery basis. It takes one query image, and rank all the gallery images by distance from it. The distance is computed using the appearance feature vectors. This means that, in the context of a ranking, the distance between feature vectors between the query and the gallery must be such that the gallery images that are of the same person as the query image have the smallest distance. The task operates over multiple such rankings.

Appearance feature vectors in MOT

Assuming we restrain ourselves with the simplification of the pure appearance graph, we can see that during the association step, the association algorithm has access to the distances between each possible pair of detections. Except that in tracking, it is all the distances between all the detections at once, and not by rankings.

We will not yet assume how a given association algorithm uses the pure appearance graph. However, we know that The role of the pure appearance graph is to help the association by giving information on whether two detections are from the same target or not, based on the weights of the graph, i.e. the distances.

We explained in the last chapter that the information inside the pure appearance graph can be represented in the form of a ranking of the pairs by distance. Intuitively, we can see that the ideal case would be that in such a ranking of pairs all the pairs of detections that are from the same target be on the left of the ranking, with the smallest distance, and that all the pairs of detections that are from different targets be on the right of the ranking, with the biggest distance. This would mean that if we found the correct threshold, we could determine with perfect accuracy which pairs of detections are from the same target and which are from different targets. An example of such an ideal ranking, for the toy example we will develop in the next section, is shown in figure3.2 (b).

Having such information even dispenses us of doing the association, since the trajectories can be deduced from the graph directly. Any ranking of the pairs by similarity that is not this ideal ranking would mean that the association is not guaranteed to succeed, because the association algorithm would not have perfect information fed to it.

3.2 Toy situation

We will now present the simplified toy case that will allow us to show that the two uses of the feature vectors, in Person Re-ID and in MOT, are not equivalent.

Consider there are only 2 different persons in a video. Consider that for each person, 2 detections have been made.

To create a Person Re-ID situation, we consider that 1 detection of each person belong to the query set, and that 1 detection of each person belong to the gallery set. The query and gallery are represented in figure 3.1 (a).



Figure 3.1: (a) The query and gallery set of detections in the simplified situation for Person Re-ID . The letter gives the person's identity (A or B). (b) The set of detections that will be used to do the association in the simplified situation for tracking. The detections are the same as for Person Re-ID .

We will take the same 4 detections and consider them the detections from a MOT dataset, and construct a pure appearance graph out of them. We of course cannot do real association with these detections, since we have no information about their time and their spatial location, and thus have not motion information. However, as long as we only analyze the pure appearance graph, any set of images of objects or persons to track can serve as a mock MOT dataset.

To create an artificial MOT situation (where detections are considered pre-made), we will consider all of the 4 images as detections that were made during the video. The set of all the 4 detections (that are the same detections as the Person Re-ID dataset) is represented in figure 3.1 (b). We can create a pure appearance graph with these detections, and its corresponding set of pairs.

Ideal results

Let us consider what would be the ideal results for Person Re-ID and the pure appearance graph :

- For Person Re-ID , the ideal situation would be that for all the query-gallery ranking, images that are the same person as the query be at the top of the ranking. This is shown in figure 3.2 (a).
- As we explained, the ideal situation for the pure appearance graph would be to have a pair ranking by distance where all the pairs of detections that are of the same target be one side of the ranking, and the rest on the other side of the ranking. This is shown in figure 3.2 (b). In this ideal setting, the association algorithm would just have to determine that similarity threshold, and it would determine the trajectories with 100% accuracy.

Results for a specific set of feature vectors

Consider the simplified toy setting we described. Consider also that a given feature extractor model has outputted a feature vector for each detection, and the feature vectors are 1-D vectors. The 4 feature vectors of the 4 people are as shown in the figure 3.3. The distance metric for computing distance between feature vectors is euclidean.

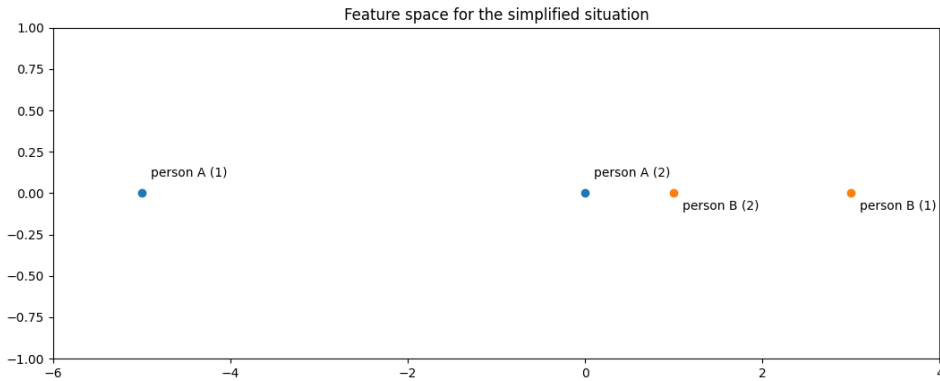


Figure 3.3: representation of the feature vectors in the feature space for the simple example. The feature vector of each detections are respectively $(-5,)$ for A(1), $(0,)$ for A(2), $(1,)$ for B(2) and $(3,)$ for B(1).

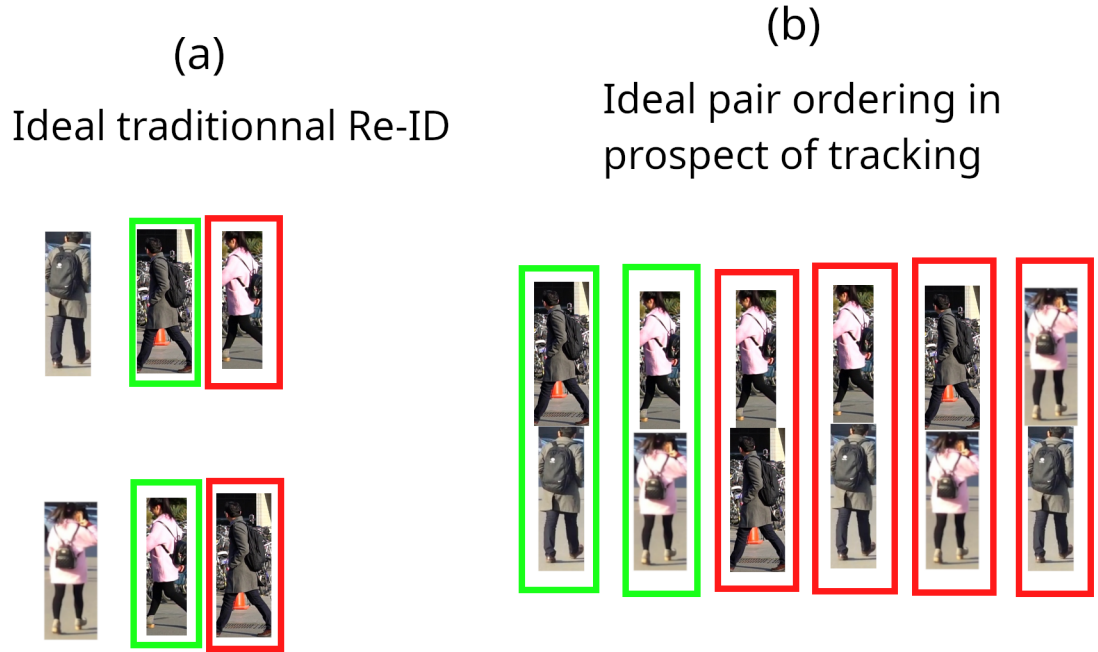


Figure 3.2: (a) The ideal query-gallery rankings of the detections. Green means the gallery image is of the same identity as the query, and red means it is a different identity. As we can see the detection that is the same identity as the query is at the top of the ranking for both query images. (b) The ideal ordering of all the possible pairs of detections by distance. Green means a pair is of two same identities, red means the pair is of different identities. As we can see, all the pairs that are of the same target have the lowest distance and are at the top of the ranking. A set of feature vectors that would order all the possible pairs of identities that way will give the MOT association algorithm a maximal accuracy at determining the trajectories.

Using these feature vectors and the distances between them, let us do the two ranking by distance of the gallery images compared to each of the two query images. The rankings are represented in figure 3.4 (a). As we can see, the two rankings are ideal and perform Person Re-ID perfectly. This means that these feature vectors allow us to perform Person Re-ID perfectly.

Using the same feature vectors, let us now rank every pair of identity by distance (the information contained in the pure appearance graph) and visualize this ranking. The ranking is shown in figure 3.4 (b). As we can see, the ranking is far from ideal. As we said above, this means that the association algorithm does not have the ability to determine whether two detections are of the same or different identities on the basis of pair distance with this set of feature vectors, and that could hurt tracking performances if an algorithm was given the pure appearance graph corresponding to this pair ranking.

This toy example shows us that a set of feature vectors allow to perform the task of Person Re-ID perfectly but does not furnish the information necessary (not a perfect ordering of pairs) to perform tracking perfectly.

Since we saw that feature vectors that perform Person Re-ID perfectly does not perform tracking perfectly, we cannot take feature extraction models optimized for Person Re-ID and use them as such to do the feature extraction step in multi-objects tracking for subsequent use in the association stage, and expect results proportional to the output of the metrics of Person Re-ID .

3.3 Comparison with pair Person Re-ID entification

The results of our toy example, that a set of feature vectors that perform Person Re-ID perfectly does not produce a perfect pure appearance graph, is confirmed by [9]. Our formulation that the ideal pure appearance graph, when ranking all the pairs by similarity would have all the pairs that are of the same person on one side of the ranking is linked to the task of pair Person Re-Identification, or pair Re-ID. The definition of the task of pair Re-ID is:

Definition (Pair Re-Identification): *Is this pair of images from the same identity ?*

[9] establishes that pair Person Re-ID is a different task than Person Re-ID . It also tries to train deep learning models to improve the performances in pair Person Re-ID .

(a)
Traditionnal Re-ID



(b)
Pair ordering in
prospect of tracking



Figure 3.4: (a) query-gallery ranking using the distance between feature vectors of figure 3.3, as we can see they are ideal. Green means it is the same person as the query and red that it is a different person. The numbers under the image indicate the distance between the feature vector of the query image and the feature vector of the gallery image. (b) Pair ranking by distance using the same feature vectors from figure 3.3. Green means that it is a pair of the same person and red that it is of different persons, and the number under the pair indicate the distance between the two feature vectors of the two images. As we can see, it is not ideal. This means that distance between two feature vectors does not lead to a confident decision on whether the two images of a pair are of the same person or not, and could lead to bad tracking performances.

The difference between Pair Person Re-ID and association

Even though there are a lot of commonalities with what we described, pair Person Re-ID is a different task than evaluating if the information contained in a given pure appearance graph is suited for association. The information contained in the pure appearance graph is indeed a set of pairs, and the objective is also to decide whether a pair of detections are of the same target. In both tasks, the idea situation would be that the pair distance allows to classify perfectly the pairs.

However, as opposed to pair re-id, the pure appearance graph is given to an association algorithm to determine the trajectories. This means other properties are expected of the pure appearance graph than what is required when performing pair Person Re-ID. The properties that we want to evaluate in the pure appearance graph (and thus in the corresponding pair ranking) will be described in the next chapter.

3.4 Summary

In this chapter, we have shown that feature vectors leading to good performances in Person Re-ID do not lead to the construction of a good pure appearance graph in term of pair ranking. We used a toy example to show a concrete simple example. We then made the link between this results and [9], which present the task of pair Re-ID, which is much closer to what we are trying to achieve with the pure appearance graph where the information is in the form of pairs. We however noted that pair Person Re-ID and association in MOT using the pure appearance graph are not the same thing, as the pure appearance graph will be used by an association algorithm.

Chapter 4

Introducing the new metric

Introduction

In the last chapter, we saw that the fact that a set of appearance feature vectors allows good performances in Person Re-ID do not equate with it allowing to construct a good pure appearance graph, and thus allow good association and good tracking. This means we cannot evaluate the performances of a given appearance feature extraction model at producing feature vectors that allow good association on the basis of this same model's performances on Person Re-ID.

This brings us to the need of designing a new metric to evaluate the ability of a set of appearance feature vectors to yield a pure appearance graph that will allow good tracking performances.

In this chapter, we will introduce this new metric that we designed to evaluate the quality of the appearance information of a set of detections to lead to good association performances. First, we will detail the concept of confidence in association, which is the main concept we want our new metric to encapsulate. Then, we will make an explanation of the general principle of the metric. After that, we will define the metric exactly and show an example for a simple case.

4.1 The concept of confidence

The way the appearance information is exploited during the association is on a confidence basis. Basically, the association algorithm will give more confidence to feature vectors that are closer to one another. If a pair of detections has a very small distance, the association algorithm will consider it is very likely they are of the same target, and if a pair of detections has a very large distance, it will consider it is very likely they are of different target. The association algorithm will associate in priority the detections with the smallest distances, and exclude in priority the detections with the biggest distances.

4.1.1 Examples in the literature

This behavior is encoded both implicitly and explicitly in the literature.

For example, in the Iterative Hypothesis Testing ([6]) paper, the association algorithm will, at the start, only associate detections into trajectories when the appearance and motion information indicates that the association is better than other possible association, in part in terms of distances between the appearance features. This constraint to associate only when the alternative associations are less attractive in terms of appearance and motion is relaxed as the algorithm progresses. An illustration of this is shown in figure 4.1

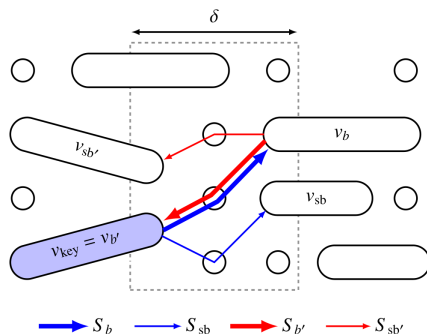


Figure 4.1: Illustration of confidence from the Iterative Hypothesis testing paper [6]. The association of two tracklet with the lowest cost in appearance and motion is represented by S_b , and is compared with the second best possibility, represented by S_{sb} . S_b must be sufficiently better than S_{sb} . The algorithm does this procedure both in forward time and in backward time with $S_{b'}$ and $S_{sb'}$. The condition that S_b be sufficiently better than $S_{sb'}$ is relaxed as the algorithm advances. The measure of the cost each possible association is partly based on the distance between the appearance feature vectors. (A. Kumar K.C. et-al, [6])

Another example where we can see the concept of confidence is in [11]. The appearance information is encoded in an appearance graph, and the algorithm tries solve a label assignment problem by minimizing the "labeling energy". In order to minimize the labeling energy, the algorithm is incited to give the same label to two detection the closer the distance between their appearance features are, in order to minimize the labeling energy.

In [18], the first part of the association is done by finding the tracklets that have the minimum appearance cost. This appearance cost depends on motion information and appearance. Detections that are closer in terms of appearance information have more chances to be a part of same tracklet, as associating them means a lower appearance cost.

4.1.2 Confidence in the pure appearance graph

This view of the literature allows us to get an insight on how the information present in the pure appearance graph will be used in an association algorithm. If we reformulate the information present in the pure appearance graph as the set of all possible pairs with their corresponding distances, the pairs that have the smallest distances will be assumed by the algorithm to be the most likely to be of the same target, and the pairs with the biggest distance will be the ones assumed by the algorithm to be the most likely to be of different targets. This has one major consequence: An error in the assignment of distance of a pair will not hurt the tracking process the same, depending on the distance. If a pair of detections that are not of the same target have a distance smaller than the distance of every pairs that are of the same target, it will hurt the tracking far more than if the same pair of different identities had a distance smaller than the distance of some pairs that are of the same target.

To help us better represent this, we will reformulate this by ranking all the pairs of the list by distance. In such a ranking, the pairs with the smallest distances, which are on the far left of the ranking, will be considered by the association algorithm to be the most likely to be the of the same target, and vice versa for the pairs with the biggest distance.

A reformulation of the concept of confidence is that, in a given ranking of pairs by distance, a pair of detections that are from different target has a distance so small that it places it at the top of the ranking, where all the pairs that are of the same targets are, the error will lead to more degradation in tracking performance than if it was not at its place in the ranking, but not at the bottom end. This example is shown in figure 4.2.

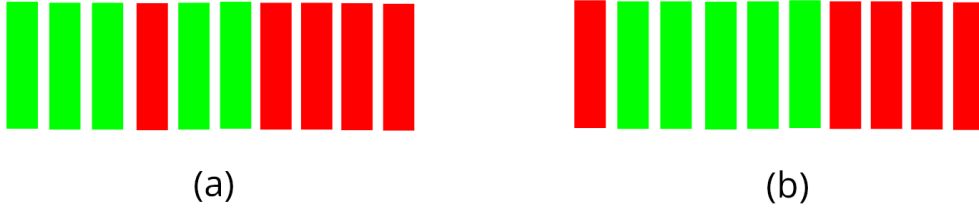


Figure 4.2: Example of the concept of confidence applied to a pair ranking. Red boxes are pairs of different targets and green boxes are pairs of the same target, and they are assumed to be ordered by distance. The misplacement of a pair of different targets on the left of the ranking in (a) is less hurtful to association than the misplacement of a pair of different targets in (b).

The importance of both pairs of different targets and same target

It is to be noted that the concept of confidence concerns both the pairs that are of the same targets and the ones that are of different target.

In terms of the ranking of pairs, the misplacement of a pair of different targets at the far left of the ranking, is as hurtful to tracking as the misplacement of a pair of same targets at the far right of the ranking.

4.2 The general principle of the new metric

We decided to design a metric that will take the form of a 2D curve. We also decided that the metric would evaluate the information contained in the pure appearance graph in the form of a ranking of pairs. The metric is given a ranking of pairs corresponding to a pure appearance graph, where the pairs have been ranked by their weights in the pure appearance graph, which are the similarity/distance between the appearance feature vectors of the detections that compose the pair.

The main principle of the metric is that it will look at the pairs at both extremes of the ranking, that is to say the ones that the association algorithm will put the most trust in believing they are either of the same target or different targets. Looking at this limited set of extreme pairs, it will compute an error rate. The metric will allow us to see how this error rate evolves as we select more and more pairs starting from the extremes of the ranking.

The metric will progressively select pairs, starting from both extreme ends of the ranking, where the pairs with the smallest and biggest distances are. The metric

will select the $r * X\%$ pairs with the smallest distance, that is to say the pairs the association algorithm will be the most confident they are of the same target. In the same selection, it will select the $X\%$ pairs with the biggest distance, that is to say the pairs that the association algorithm will be the most confident they are of different targets. r is the ratio of pairs that are of the same target over the ones that are of different targets in the ranking. We select a different quantity of each types of pairs because our objective is to take, at each selection, a certain share of the total number of pairs for each type.

For each selection of pairs, the metric will count how many errors there are in the selection. That is to say, from the pairs selected at the extreme of the ranking where the distances are the lowest, how many pairs that are made of different target are present, and vice versa for the pairs selected at the extreme of the ranking with the biggest distances. The errors on both sides are then summed, and an overall error rate is computed.

A ratio is also made of the quantity of pairs that are correct for each of the two selections over the total number of pairs. So for each selection of pairs, there is an error rate, and a ratio of the quantity of correct pairs over the total number of pairs. We increase the selection of pairs and compute the two values for each selection. We stop when all the pairs have been selected.

4.3 Definition of the metric

Given a ranking of pairs, the metric will select the N last pairs of the ranking and label them as of different identities, and select the rN first pairs of the ranking and label them as of same identities, and not label the rest of the pairs of the ranking. r is the ratio of pairs of same identities over pairs of different identities in the pair ranking, and rN is ceiling rounded to the closest integer. In this situation, we define the following numbers:

- *TPS* (True Positive Same) The number of pairs that have been correctly labeled as being of the same target.
- *FPS* (False Positive Same) The number of pairs that have been incorrectly labeled as being of the same targets, meaning they are actually of different targets.
- *TPD* (True Positive Different) The number of pairs that have been correctly labeled as being of different identities.

- *FPD* (False Positive Different) The number of pairs that have been incorrectly labeled as being of different targets, meaning that they are actually of same target.
- *N* (Neutral) The number of pairs that have not been selected in this iteration.

Then it will compute two things:

1. First, *Quantity*, the number of pairs that have been correctly labeled as of same or different targets, expressed as a share of the total number of pairs:

$$Quantity = \frac{TPS + TPD}{TPS + TPD + FPS + FPD + N} \quad (4.1)$$

2. Second, *Error*, the error rate in of the pairs labeling, ignoring the *N* pairs that have not been labeled:

$$Error = 1 - \frac{TPS + TPD}{TPS + TPD + FPS + FPD} \quad (4.2)$$

We do this computation again for an increasing value of *N*, until the value of *N* where all the pairs have been labeled as either of same or different identities. Thus we obtain a series of points that each correspond to a specific regime, corresponding to a selection of pairs with a given quantity of information available at a given error rate.

the variable *r* in the selection of pairs of same identities is introduced to guarantee a ratio in each selection of pairs that are labeled that is close to the ratio in the complete set of pairs. The assumption is that if the ranking was perfect, selecting the pairs in such a way would give us a selection of pairs with the exact same ratio as the complete set of pairs. Since the ranking is not perfect, this is not the case in our metric, but we observe empirically that is is sufficiently close.

4.4 Practical example

We will now show a practical example of our new metric, computed on a simple ranking shown in figure 4.3. This is the same ranking that was produced by the toy example of chapter 3. The ratio of pairs of same targets over different targets is $\frac{1}{3}$.



Figure 4.3: The ranking that we will evaluate with our new metric, the number under the pairs represent distances. Pairs with a green cadre are of the same target, and pairs with a red cadre are of different target.

As we described, the metric will progressively select pairs on both sides of the ranking and compute the quantity and error variables. Figure 4.4 show the computation of quantity and error for $N = 2$.

Figure 4.5 represents the metric of the ranking. It is 2D, and must be read as follow: selecting the *quantity* most confident pairs gives an *error* error rate.

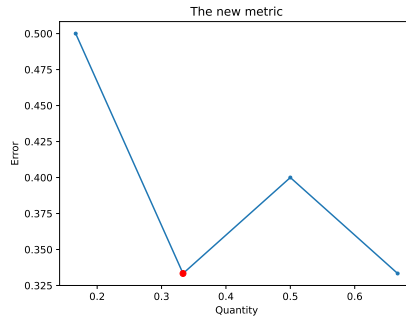


Figure 4.5: The new metric for our example ranking. The x axis is the quantity variable, and the y axis is the error variable. The red point represent the values we computed in figure 4.4. As we can see, the ranking is bad, as selecting the pairs we are most confident (low quantity) do not give us a low error rate.

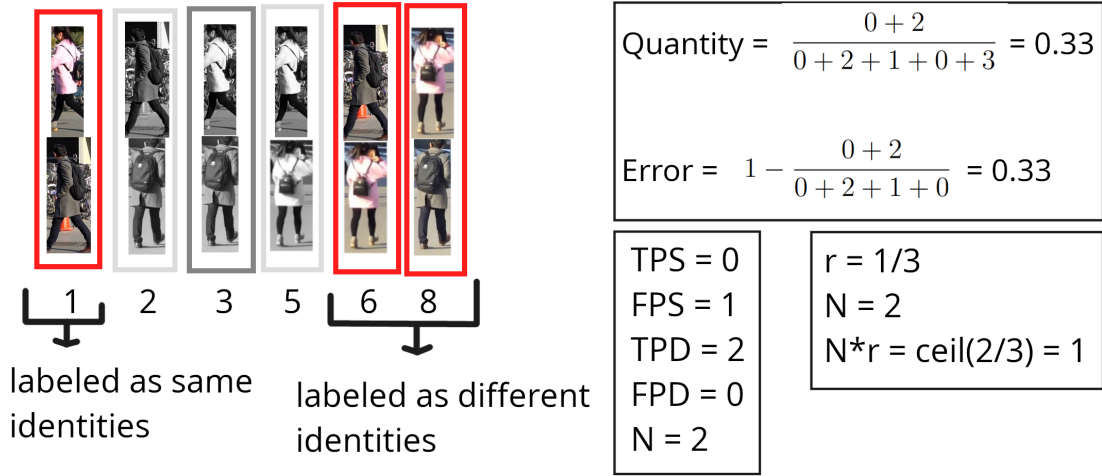


Figure 4.4: Illustration of the computation of *Quantity* and *Error* for $N = 2$. Red indicate a pair in the ranking is of different targets, and green means the pair is of same target. The pairs that have been grayed were not selected. In this example, since the first pair of the ranking was not of same target due to a poor ranking, no pairs of the same target were selected, so there is no green. As explained, $N = 2$ pairs were selected at the end of the ranking and labeled as different target, and $\text{ceil}(Nr) = 1$ once ceiling rounded. The ratio of the pairs of same identities over pairs of different identities in the whole ranking is $\frac{1}{3}$

4.5 Interpretation of the metric

This metric is 2 Dimensional, and represent a series of "operation points", which indicates, for a given selection of pairs, what share of correctly classified information there is, and at which error rate. It reflect, for a given selection of the pairs in the ranking we are the most confident they are either of the same target or different targets, at what point the selected pairs are correct (using the error rate). The metric thus describes a compromise between the quantity and the reliability of the information.

We designed the metric to be 2-D, because what this metric evaluate is the quality of the information contained in the pure appearance graph in order to perform association in general. Depending on the association algorithm, we might prefer to have more or less pairs with more or less error.

Chapter 5

Improving the new metric curve using side information

Introduction

In the last chapter, we defined our new metric to evaluate the performance of a given set of feature vectors for association through the evaluation of a ranking of pairs. In this chapter, we will see methods to improve the results with this new metric using the side information.

Since our metric is a 2 Dimensional curve, we will first detail the meaning of improving the new metric. After that, we will explain the concept of side information. Then we will discuss the possibility of using the side information that some feature extraction models produce to improve the new metric.

5.1 Comparing two new metric curves

The new metric that we designed is a 2D curve because, depending on the association algorithm, we might need more or less appearance information with more or less error rate. But as we want to compare two curves of the metric without specifying which association algorithm will be used, every point in the curve of the metric is important, as they all represent valid operation points.

To perform a complete comparison, the best way is to plot the two metric curve side by side and compare them at every point. A given metric curve can be better than another only on certain points. Depending on which association algorithm you use, a metric curve that has a lower error rate at the beginning but a higher error rate at the end compared to another curve might be considered an improvement.

If we do not want to perform a visual examination of the curves, we can compute the area under the curve (AUC) for each curve and compare them ; This gives us an indication if the a curve is globally better than the other, all possible operation points in the curve being equal. With this method however, the details of the difference between the two curves will be invisible. AUC allows us to see if a curve is globally better than the other, but not where it is better. Two curves might have the same

AUC but have different characteristic that we want to capture. An example of two imaginary hand-drawn curves of our metric which would have the same AUC but different characteristic that we want to capture is shown in figure 5.1. This is why, to determine if a given metric curve has been improved, it is better to look at the whole curves.

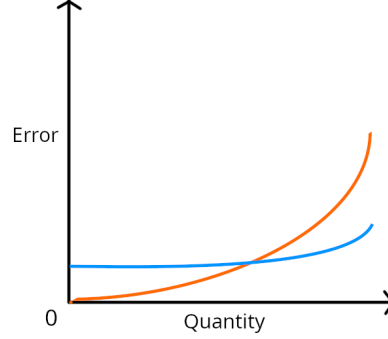


Figure 5.1: A hand drawn example of two imaginary metric curves that have the same area under the curve (AUC), but important special characteristics. The error is far higher for the blue curve at the beginning, but gets smaller than the one of the orange curve at the end. Comparing these two curves using the AUC would lead to the conclusion they are the same.

5.2 Method for improving the ranking using side information

Our new metric computes the curve by evaluating a ranking of all the pairs of detections. Consequently, trying to improve the metric consists in determining a new ordering of the pairs that will lead to a better curve.

Because of how we designed the new metric, the more a pairs of detections is misplaced at the wrong extreme of the ranking, the more it is penalizing for the new metric, since it will influence the error rate from the beginning of the curve.

The option we propose here is to improve the pair ranking and thus the new metric is to take the pairs whose distance we are not sure of and which are at the extreme of the pair ranking, and displace them in a more central region of the ranking, where they will be selected last when evaluating the new metric.

To establish which pairs we are more or less confident about their distances, we will base ourselves on the side information. The side information is an information that is outputted by a feature extraction model, but which is not the feature vectors. It is a type of information that is not, or only partially used when computing the distance between two appearance feature vectors. Even after computing the distance between the feature vectors of two detections, this side information still contains information that is not encoded in the distance.

We want to use the side information available to improve the ranking of pairs. In this section, we will propose a simple naive algorithm to do so.

We will assume that, for each pair of detections in the ranking, the side information of the two detections of the pairs have been processed into a variable we call the "certainty coefficient". The certainty coefficient is a number meant to indicate how confident we are that we can trust the distance corresponding to the pair as an indicator of if it is a pair of detections that are from the same target or not.

5.2.1 The concept of center of the ranking

Consider a pair ranking of length $L = N_{same} + N_{diff}$, with N_{same} pairs that are of the same target and N_{diff} pairs that are of different targets. and assuming the pair ranking is sorted so that the pairs with the smallest distances are on the left of the ranking. $r = \frac{N_{same}}{N_{diff}}$ is the ratio of the pairs of both types.

As we described in chapter 4, to compute each point in the metric, we make a selection of the pairs with the most extreme distance of the ranking. We defined our metric so that the N last pairs of the ranking were selected, as well as the rN first pairs of the ranking.

Knowing this, we can compute the value of N at which the selection of N pairs from the end of the ranking and rN pairs from the start of the ranking will sum to make the whole ranking, i.e. when all pairs will be selected. This value of N that we call N_{center} is:

$$N_{center} = \frac{L}{1+r} = N_{diff} \quad (5.1)$$

N_{center} is the number of pairs, starting at the right of the ranking, at which the last pairs that will be selected before the two selections select the whole ranking. By complementarity, since $N_{center} = N_{diff} = L - N_{same}$, N_{same} is the number of pairs starting from the left of the ranking at which the last pair will be selected. This is illustrated in figure.

This place in the ranking is important, because the pairs close to this point are the ones that will least influence the error in the metric, since they will be present in the selections last.

To improve the ranking, we propose the following method: we will take a certain share of the pairs with the most extreme distance, and check their certainty coefficient. If it is under a certain value, we will displace the pair to the "center" of the ranking that we defined with N_{center} .

5.2.2 Mathematical description of the method

The ranking is represented by an ordered list P of L pairs that have been ranked by distance: $P = (p_1, p_2, \dots, p_L)$ s.t. $\text{dist}(p_i) \geq \text{dist}(p_j) \forall i > j$, where $\text{dist}(p_i)$ is the distance between the appearance features of the detections of pair i .

There are N_{same} pairs that are of the same target and N_{diff} pairs that are of different targets in the ranking, such that $L = N_{same} + N_{diff}$. The pairs in the ranking are in a ratio of $r = \frac{N_{same}}{N_{diff}}$.

We will divide the pair ranking at the level of its center. As we saw, starting from the left of the ranking, the ranking center is N_{same} pairs away from the start of the ranking. We divide P into two ordered lists P_1 and P_2 at this center:

$$P_1 = (p_1, \dots, p_{N_{same}}), P_2 = (p_{N_{same}+1}, \dots, p_L) \quad (5.2)$$

Now we will divide P_1 into P_1^C and P_1^F (C stands for "Change" and F for "Fixed"). P_1^C is the list of pairs from which the pairs that have a certainty coefficient lower than a certain threshold will be moved further from the extreme of the ranking. P_1^F is the list of pairs that will not be moved if they have a certainty coefficient lower than the threshold because they will already be considered to be at the extremes of the ranking.

To make this division of P_1 , we define $N_{RS} = \sigma N_{same}$ (RS stands for "Restrained Same"), with $\sigma \in]0, 1]$ a parameter of our method. N_{RS} will select a portion σ of the pairs of P_1 with the smallest distance.

Thus we define P_1^C and P_1^F as:

$$P_1^C = (p_1, \dots, p_{N_{RS}}), P_1^F = (p_{N_{RS}+1}, \dots, p_L) \quad (5.3)$$

We will divide a third and last time P_1^C into P_1^{CC} (C stands for "Certainty"), the set of pairs from P_1^C that have a certainty coefficient higher than the threshold and that we will not move, and P_1^{CU} (U stands for "Uncertainty"), the set of pairs from

P_1^C that have a certainty coefficient lower than the threshold and that we will moved farther in the ranking because of their uncertainty. P_1^{CC} is defined as

$$P_1^{CC} = (p_i | p_i \in P_1^C \text{ and } \text{cert}(p_i) \geq \lambda_c) \quad (5.4)$$

Where $\text{cert}(p_i)$ compute the certainty coefficient of the pair p_i , and λ_c is the certainty coefficient threshold over which a pair will not be moved. P_1^{CC} is still an ordered list of pairs where the pairs are ordered by distance. In the same fashion, we define P_1^{CU} as:

$$P_1^{CU} = (p_i | p_i \in P_1^C \text{ and } \text{cert}(p_i) < \lambda_c) \quad (5.5)$$

The same way, P_1^{CU} is still an ordered list of pairs where the pairs are ordered by distance.

We then concatenate the three lists, P_1^{CC} , P_1^F and P_1^{CU} :

$$P'_1 = \text{concat}(P_1^{CC}, P_1^F, P_1^{CU}) \quad (5.6)$$

P'_1 is thus not a list of pairs ordered by distance anymore. P'_1 represents the new ordering of pairs on one side of the ranking.

We will now do a similar operation with P_2 . First we divide P_2 into P_2^C and P_2^F (again C stands for "Change" and "F for "Fixed"). The same way as before, the pairs of P_2^C , if they have a certainty coefficient lower than the threshold, will be moved further to the center of the ranking.

We define $N_{RD} = \sigma N_{diff}$ (RD stands for "Restrained Different"). N_{RD} will select a portion σ of the pairs of P_2 with the biggest distance. Note that σ has the same value for the definition of N_{RS} and N_{RD} .

Thus we define P_2^C and P_2^F :

$$P_2^C = (p_{L-N_{RD}}, p_{L-N_{RD}+1}, \dots, p_L), P_2^F = (p_{N_{same}+1}, \dots, p_{L-N_{RD}-1}) \quad (5.7)$$

The same way as for P_1^C , we will divide P_2 into P_2^{CC} (C for "Certainty") and P_2^{CU} (U for "Uncertainty"), depending on the certainty coefficient of the pair and which will decide which pairs are moved away from the extreme of the ranking. P_2^{CC} is defined as

$$P_2^{CC} = (p_i | p_i \in P_2^C \text{ and } \text{cert}(p_i) \geq \lambda_c) \quad (5.8)$$

Again P_2^{CC} is still an ordered list of pairs where the pairs are ordered by distance. We define P_2^{CU} as:

$$P_2^{CU} = (p_i | p_i \in P_2^C \text{ and } \text{cert}(p_i) < \lambda_c) \quad (5.9)$$

Again we will concatenate P_2^{CU} , P_2^F and P_2^{CC} to obtain the new ordering of the pairs P'_2 :

$$P'_2 = \text{concat}(P_2^{CU}, P_2^F, P_2^{CC}) \quad (5.10)$$

Finally, we concatenate P'_1 and P'_2 into the modified ordering of the pair:

$$P' = \text{concat}(P'_1, P'_2) \quad (5.11)$$

This new ranking of the pairs can be evaluated by the metric as any other pair ordering. The method only has as parameters σ and λ_c , the share of the extreme pairs that can be moved and the certainty coefficient threshold under which the distance of a pair is considered unreliable.

Summary

In this chapter, we introduced attempt at designing a method to improve the pair rankings measured by our new metric.

We first described what improving the metric meant, either by comparing two curves of the metric or comparing their area under curve.

We introduced the concept of side information, which is the information outputted by some models beside their normal appearance information, and which allows us to generate what we call a "certainty coefficient" for each pair of detections to determine how confident we can be that the distance associated with the pair is indicative of its nature.

Then we gave the general idea of the method for improving the rankings, which is using the side information, for replacing the pairs that are on the extreme of the ranking, where if they were errors they would hurt more the new metric, and displacing them more to the center of the ranking. Then we gave a precise description of our method.

In the experiments in chapter 6 and 7, we will try and apply this method to see if it improves the new metric curve.

Chapter 6

Experiments

Introduction

In the next chapter, we will be experimenting with the new metric we designed and the ranking improvement method we introduced. We want to analyze and discuss the output of our new metric for some cases, as well as try to see if the ranking improvement method we designed in the last chapter leads to improvement of the metric curve in those same cases. In order to do this, we will present here two sources of data that we designed to evaluate our metric on, which both include a part of side information that will be used to apply the ranking : the feature vectors outputted by VSNet for the Market-1501 dataset, which we used to create an set of pairs that replicates a tracking situation, and a simple artificially generated dataset of pairs which also comprises a certainty coefficient.

6.1 Market-1501 and VSNet

In this section, we describe the first source of data for our experiments, which are the appearance features extracted from the images of the testing set of the dataset Market-1501 [19] using the VSNet model [2].

We will first present the Market-1501 dataset, more precisely the testing set. Then we will explain how we composed an artificial tracking dataset from the images of Market-1501. After that, we will describe VSNet, the features and the side information it outputs. We will also describe how we use this side information to compute the certainty coefficients.

6.1.1 Market-1501

Market-1501 [19] is a dataset that has been specially constructed for Person Re-ID and which is publicly available. It is composed of 32 668 images of 1501 different persons. Every image in the dataset is labeled with the identity of the person, as well as the camera where it was taken. The dataset is composed of:

- A training set composed of 12936 images of 751 identities (i.e. 751 different persons).
- A query set containing 3368 images of 751 identities, used for testing.
- A gallery set containing 19732 images of 750 identities, as well as "junk" and "distractor" images (described after), used for testing.

bounding boxes

To increase the realism of the dataset, the bounding box around the image of person is not hand drawn but obtained with a detector called the Deformable-Part-Model (DPM) and they are manually annotated with their identity.

To determine if the bounding box generated by the DPM for a person is valid, the generated bounding box is compared with a hand-drawn bounding box that represent the ground truth. If the area ratio of the two bounding boxes exceeds 50%, then the image is marked as "good". If it is under 20%, then it is marked as "distractor". Otherwise, it is marked as "junk". Only the "good" bounding boxes are annotated with the identity of the person. It is to be noted that often the "junk" and "distractor" boxes contain parts of the person or a person from far away. Some contain also an entire person in the same way as one that can figure in the gallery set, but which has not been labeled. Examples of such bounding boxes are shown in figure 6.1.

The query images bounding boxes are hand-drawn and do not come from DPM.



Figure 6.1: An example of "junk" bounding boxes. From left to right: parts of a person, a person from far away, background, and a person that has not been labeled.([19])

The testing set of Market-1501

In this thesis, we will only use the testing set of Market-1501. As we said, it is composed of 3368 query images and 19732 gallery images. However, in the framework

that we used for this work (torchreid, [20]), It outputted for testing the features 3368 query image but only 15913 gallery images, including 2798 "junk" and "distractor images". These 15913 images contain all the 750 identities, with at least 4 images per identity. For the experiments, we used this limited test set.

6.1.2 VSNet

For our experiments on Market-1501, we will be using the feature vectors and side information produced by an instance of VSNet ([2]) that has been trained on Market-1501, and thus has been optimized for Person Re-ID. VSNet has the advantage of producing side information that will allow us to experiment on improving the pair ranking.

VSNet is a body part-based deep learning model for Person Re-ID. It has been trained to recognize the body parts of a person, and produce embedding (feature vector) for each body part. We will not detail here the inner working of this model, as it is not the subject of the thesis. Instead we will discuss its outputs.

This model outputs 9 feature vectors $(f_f, f_1, \dots, f_8)$, with f_f being the foreground feature vector, which describe the whole person, and f_i , $i \in \{1, \dots, 8\}$ being the body part feature vectors. The model also outputs a set of 2D body part maps $M_1, \dots, M_8 \in R^{H \times W}$, which locates the body part of the detected person in the image. These map locate 8 main body parts, namely the head, torso, left/right arm, left/right leg and left/right foot, as well as the foreground of the person in a feature map M_f . These body part maps are used to compute the feature vector of each specific body parts $\{f_1, \dots, f_8\}$, as well as a feature vector for the whole person f_f . Then, the body part maps are used to assess if the said body part was not occluded in the image, to determine whether the corresponding feature vector should be included in the computation of distance. it does this by determining a visibility coefficient ν_i :

$$\nu_i = \begin{cases} 1 & \text{if } \max_{h,w} (M_i(h, w)) > \lambda_\nu \\ 0 & \text{otherwise} \end{cases} \quad (6.1)$$

With λ_ν the threshold for a body part to be visible. The foreground visibility coefficient is computed as $\nu_f = \max_{i \in \{1, \dots, 8\}} (\nu_i)$. The distance between two detections is computed using the feature vectors and the visibility coefficients:

$$\text{dist}_{\text{total}}^{\text{qg}} = \frac{\sum_{i=\{f,1,\dots,8\}} (\nu_i^q \nu_i^g \text{dist}_{\text{cos}}(f_i^q, f_i^g))}{\sum_{i=\{f,1,\dots,8\}} (\nu_i^q \nu_i^g)} \quad (6.2)$$

Where $\text{dist}_{\text{cos}}(f_i^q, f_i^g)$ is the cosine distance between the two feature vectors of images q and g .

Computing the certainty coefficients

The objective of the experiments is to compute our new metric over a pair ranking, then improve this ranking using the side information. We assumed when designing our method for improving the ranking in chapter 5 that the side information had been processed to compute a single certainty coefficient per pair, which indicated how much we could rely on the distance associated with the pair.

We will now use the side information outputted by VSNet, namely the visibility coefficients, to compute a certainty coefficient for each pair of images. Our assumption is as follow: the least body parts are comparable between two images, the more unreliable the total. distance between two images becomes. We will also assume that each of the 8 body parts produced by the model have equal importance in the computation of a reliable distance.

We will define the certainty coefficient c_{ij} for a pair of images i and j as:

$$c_{ij} = \frac{\sum_{i=\{1,\dots,8\}} \nu_i \nu_j}{8} \quad (6.3)$$

This coefficient thus computes the share of body parts that are visible in both images.

6.1.3 Making a pair tracking dataset from Market-1501

Market-1501 is a Person Re-ID dataset, which means that the testing works on a query-gallery basis. Our new metric is based on the pure appearance graph we introduced, where all the possible pairs of detections are considered. To evaluate our new metric on the market-1501 dataset, we will select a subset of the images from the query and gallery set, and assume they form the set of detections in an hypothetical tracking situation. The new metric will be evaluated on the pure appearance graph corresponding to this set of images, that is to say on the list of all possible pair of images in the subset.

We decided to generate a subset composed of 7000 images, both from the query and gallery set. We excluded the "junk" and "distractor" images. This is because these images contain multiple instances of unlabeled persons, with sometimes the same person in multiple junk images. This means that if we selected them, we would get pairs of images that are actually of the same person but will be marked as different persons because they are both junk images. This is not something that we want when evaluating our new metric.

To select the 7000 images, we each time selected a person identity at random, and add all the instances of this person, both from the query and gallery set, to the subset of images. We do this until the number of added images reaches 7000 and then stop.

When evaluating the new metric, we compute all the possible pair of images. When evaluating Market-1501 for Person Re-ID, the pairs of images that are of the same person and that come from the same camera are eliminated because they are deemed too simple. This is not the case in our artificial dataset, as we want to create a set of pair that correspond to the pure appearance graph, where all the pairs, even the ones that are of the same target and taken from the same camera, are present.

7000 images give us 24 496 500 different pairs of images. In the dataset we generated, 90 735 of these pairs are of the same identity, and 24 405 765 are of different identities. This gives us a ratio of pairs of around 0.00372.

For each of these pairs, we compute the distance between the images using the feature vectors of each image produced by VSNet using equation 6.2, and we compute the associated certainty coefficients using the visibility coefficients of each image with equation 6.3. Thus we obtain a list of pairs with their associated distances and certainty coefficients.

6.2 Artificially generated dataset

In this section, we will describe our second source of data for the experiments, which is an artificially generated dataset specially designed to evaluate our metric on. The dataset try to replicates the information contained in a pure appearance graph. For that, it generates a set of pairs, whose distances are sampled from a normal distributions. We also include a side information mechanism , in the form of a randomly selected certainty coefficient which determines an offset that is added to the pair distance. The higher the certainty coefficient is, the potentially higher the offset will be. This way, we artificially create a case where the certainty coefficient defines the degree to which we can trust the distance associated with a pair.

The data is generated as follow. For each pair p_i that are of the same target:

1. We sample $\text{dist}(p_i)' \leftarrow \mathcal{N}(\mu_{\text{same}}, \sigma_{\text{same}}^2)$ where μ_{same} and σ_{same} are parameters to choose. $\text{dist}(p_i)'$ is the distance of the pair before adding the offset.
2. We sample $c_i \leftarrow U(0, 1)$ where $U(0, 1)$ is the uniform continuous probability distribution from 0 to 1. c_i is the certainty coefficient of pair p_i
3. We sample $\text{offset}(p_i) \leftarrow \mathcal{N}(0, (\sigma_{\text{max same}}(1 - c_i))^2)$, which is the offset that depends in the certainty coefficient. The lower the certainty coefficient, the higher the offset. $\sigma_{\text{max same}}$ is a parameter to choose.
4. We define $\text{dist}(p_i) = \text{dist}(p_i)' + \text{offset}(p_i)$, the final distance for the pair

We thus obtain a pair p_i with the associated information $(\text{dist}(p_i), c_i)$.

For each pair that are of different targets, we do the same procedure, but by replacing μ_{same} , σ_{same} and $\sigma_{\text{max same}}$ by μ_{diff} , σ_{diff} and $\sigma_{\text{max diff}}$, respectively, and which are parameters to choose.

With this method, we compute N_{same} pairs of the same target and N_{diff} pairs of different targets. In the experiments, we decided to use a ratio of N_{same} over N_{diff} similar to the tracking dataset we made with Market-1501, so we choose 0.003.

For the experiments, we choose to set the parameters as $\mu_{\text{same}} = 5$, $\mu_{\text{diff}} = 8$, and $\sigma_{\text{same}} = \sigma_{\text{diff}} = \sigma_{\text{max same}} = \sigma_{\text{max diff}} = 1$. We choose to have $N_{\text{same}} = 10\,000$ pairs of the same target and $N_{\text{diff}} = 3\,333\,333$ pairs of different targets. Figure 6.2 show the distribution of the distances of the pairs for the parameters we set.

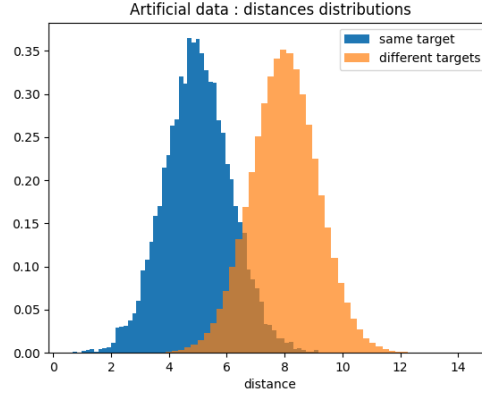


Figure 6.2: Pair distance distribution for the artificial dataset

Summary

In this chapter, we presented the two sources of data that we will use to evaluate our new metric and try to improve the pair ranking in regard to the new metric. The first source is Market-1501, where we took a subset of the images from the query and gallery testing set, and computed all the possible pairs of images between them. We compute the distance between image for each pair using the feature vectors and visibility coefficients produced by VSNet. We then computed a certainty coefficient for each pair, using the visibility coefficients outputted by VSNet. The second source of data is an artificially generated dataset, which replicates the data contained in a pure appearance graph. For each pairs, it generate and certainty coefficient at random, and generate a distance designed to be more uncertain the more the certainty coefficient is high. The ratio of pairs of same target and different targets reproduces the one of the first source of data.

Chapter 7

Results

Introduction

In this chapter, we will discuss the results we obtained with our new metric and the pair ranking improvement method on the two data sources we described in the last chapter. First, we will show the new metric applied on a pair ranking from the artificial dataset. Then, we will try to use the ranking improvement method and see if we can improve the original curve. After that, we do the same on the Market-1501 and VSNet data.

7.1 Results on the artificial data

7.1.1 Analysis of the new metric

Figure 7.1 shows us the new metric applied on the pair ranking of the artificial dataset.

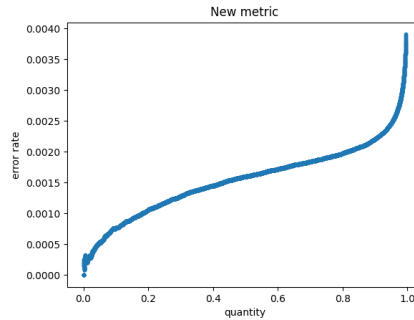


Figure 7.1: The new metric on the artificial dataset

We can see that the error in the curve increases as the the quantity of pairs selected increase. This is what we expect, since the artificial data is made of two Gaussian distribution, the extreme distances, and thus the extremity of the pair ranking, are dominated by the pairs of the same target to the left of the ranking and

dominated by the pairs of different targets to the right of the ranking. As we select more pairs from deeper inside the ranking, the error starts to increase.

7.1.2 Results of the ranking improvement

We applied the ranking improvement method on the ranking, and the results we obtained with the parameters (see section 5.2.2) $\sigma = 0.3$ and $\lambda_c = 0.1, 0.2, 0.3$ are shown in figure 7.2.

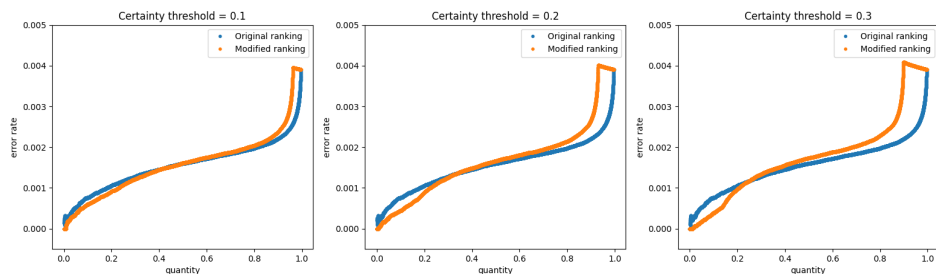


Figure 7.2: Results of the ranking improvement method on the artificial data. $\sigma = 0.3$, from left to right $\lambda_c = 0.1, 0.2, 0.3$

The results of our ranking improvement method are mitigated. On one hand, it reduces the error at the beginning of the curve, but on the other hand it increases the error at the end of the curve, and create an angular shape at the end. The diminution of the error at the start is conform with the fact that low confidence pairs had a more uncertain distance and had more chances to be present at the wrong extreme of the ranking, are moved by our method away from the extremes, thus diminishing the error. We can see that this effect of error reduction measured by our metric is constrained between around 0 and 0.3 on the quantity axis, which is coherent with the parameter $\sigma = 0.3$ that we set, which made so that we moved only the 30% most extreme pairs from both sides of the ranking. This is confirmed by figure 7.3 which apply the ranking improvement method with $\sigma = 0.5$.

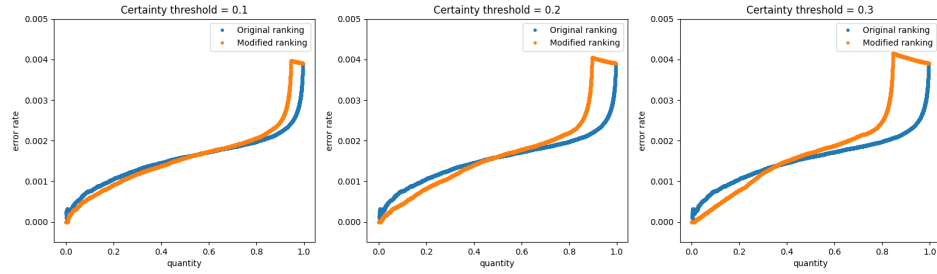


Figure 7.3: Results of the ranking improvement method on the artificial data, $\sigma = 0.5$, from left to right $\lambda_c = 0.1, 0.2, 0.3$. We can see the zone where the error is diminished is larger than with $\sigma = 0.3$

We hypothesize that the the angular shape at the end of the curves is due to the pairs with low certainty that have been moved by our method and which were actually correct. They get selected last by the metric, and slightly decrease the error at the end. The fact that these correct pairs have been moved also seem to cause the error after the diminished-error zone at the start to increase.

7.2 Results on Market-1501 and VSNet

7.2.1 Analysis of the new metric

Figure 7.4 shows the new metric evaluated on the pair ranking. As for the artificial dataset, we see that the error increases as the quantity increases. This means the VSNet model produced feature vectors that put two images closer if it was more certain they were of the same identities, and farther apart if it was more confident they were of different identities.

We can see at the start of the curve that the error starts a little bit higher than 0. This means that there are misplaced pairs at the very extremities of the ranking, but in low numbers.

7.2.2 Results of the ranking improvement

We will apply the ranking improvement method on the ranking. The results with $\sigma = 0.3$ and $\lambda_c = 0.376, 0.51, 0.626$ are shown in figure 7.5. The only improvement from the original curve is the suppression of the small increase in error at the start of the curve. When we increase the certainty threshold to $\lambda_c = 0.51$, we see that the

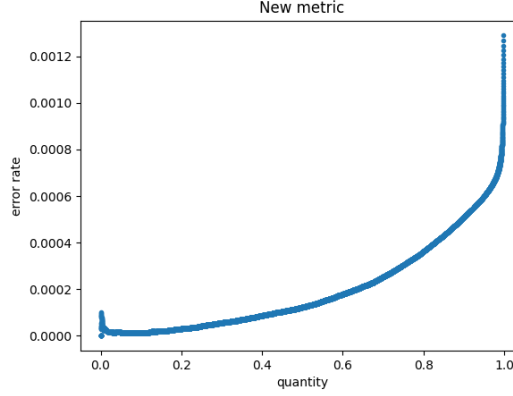


Figure 7.4: The new metric for the data from Market-1501 and VSNet.

angular shape appears, which seems to indicate that it is mostly correct pairs that are displaced from the extreme of the ranking. This result might be explained by the fact that, on the contrary of our artificial dataset, we do not know if the certainty coefficients we computed for each pair are indicative of how much we can trust its corresponding distance. In our experience, the certainty coefficient still demonstrate some limited capacity to indicate how much we can trust the distance, since the pairs that generated the small error peak at the beginning of the curve are displaced.

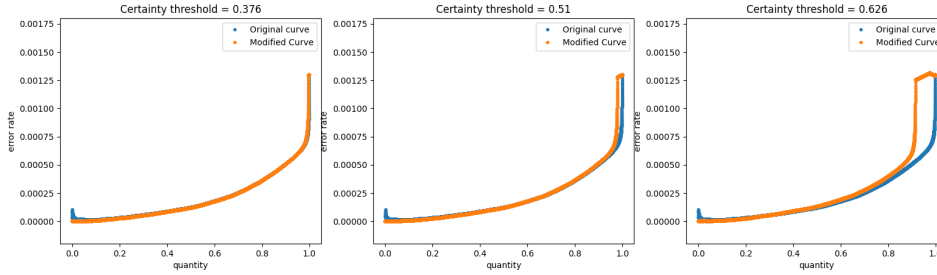


Figure 7.5: Results of the ranking improvement method on the Market-1501 data. $\sigma = 0.3$, from left to right $\lambda_c = 0.376, 0.51, 0.626$

Summary

In this chapter, we analyzed the results of our new metric for the artificial dataset and the data from Market-1501 with VSNet. For both datasets, we computed the

new metric and discussed it. Next, we applied our ranking improvement method on the rankings. We observe that on the artificial dataset, the ranking improvement method yields a lower error rate at the start, but an increased error rate at the end. We managed to change the width of the zone in the curve where the error rate is reduced by changing the parameter σ of our method. We observed an angular shape at the end of the curve, which we hypothesized is caused by the pairs that were moved at the end of the ranking by our method, but which were correct, i.e. which were on the correct side of the ranking at the extreme to begin with. After that, we tried our ranking improvement method on the Market-1501 with VSNet data, and we obtained a small decrease of the error at the beginning of the curve, but no significant improvement anywhere else and an increase of the error rate with an angular shape as for the artificial dataset.

Chapter 8

Conclusion

8.1 Summary of the work

In this work, we outlined the need to introduce a new metric for evaluating the appearance information outputted by a feature extraction model for performing association in Multiple Object Tracking.

In chapter 1, we described the tasks of Person Re-Identification (Person Re-ID) and Multiple Object Tracking (MOT). We outlined the different steps the two tasks were composed of, namely feature extraction and query-gallery ranking for Person Re-ID, as well as the metrics to evaluate the two tasks. We outlined that the feature extraction step was shared by the two tasks and proposed to use the feature vectors produced by feature extraction models specialized for Person Re-ID for the association step in tracking.

Before going any further, we made the assumption in chapter 2 that the appearance information that is used for the association step in tracking comes in the form of the distance between appearance information of the detections, which we encapsulate in what we call the pure appearance graph. We assumed for the rest of this work that the information given to a tracking association algorithm comes in this form.

After this, in chapter 3 we detailed why the idea of using feature extraction models that were optimized for Person Re-ID for doing the association in MOT using the pure appearance graph was not guaranteed to yield good performance. In order to show this, we constructed a simple toy example where we show that a set of feature vectors might be optimized for Person Re-ID but not give a pure appearance graph that will give optimal association performances. We also made the comparison between having good quality of information in the pure appearance graph and the task of Pair Person Re-ID, and outlined the difference between the two. This brought us to the need to devise a new metric that will evaluate the performance for association of a set of appearance information.

In chapter 4, in order to design our new metric that evaluates the performances of a set of appearance information for association, we had a closer look at the properties of the information present in the pure appearance graph that leads to good association. We developed the concept of confidence that we justified by taking some examples in the literature. Still in chapter 4, we introduced the new metric that tries to

encapsulate the concept of confidence. We devised a 2-dimensional metric, which associates an error rate with a certain quantity of the available information selected. We assumed that the selection always take the two type of pairs (of pairs the same target or of different targets) in a certain ratio.

Chapter 5 Tries to develop a method that uses what we call the side information outputted by some feature extraction models to improve the curve of our new metric. It works by reordering the pairs of detections in a ranking of the pairs of the pure appearance graph if the side information indicates that the distance corresponding to the pair cannot be trusted with a certain degree of certainty, that we encapsulate in a certainty coefficient.

Finally, in chapter 6 and 7, We devised two datasets, one artificial and one constructed from the feature vectors outputted by VSNet ([2]) extracted from the images of the dataset Market-1501 ([19]). We used them to test our new metric as well as the ranking improvement method, and found that the ranking improvement method improved parts of the new metric curve, but not the curve overall.

8.2 Perspective for future works

In this work, we designed a metric that aims at evaluating the performances of a set of appearance features at enabling good association in Multi-Object Tracking. However, we did not apply our new metric to a real MOT dataset, but only on an artificial dataset and a dataset based on Market-1501, which is not a MOT dataset. A future work could evaluate our new metric on a real MOT dataset, and perform tracking association using the same feature vectors that we used for evaluating our metric using different association method and evaluate the results using traditional MOT metrics. Then it could compare the results for the various association method to our metric.

Another possible work would be to develop special feature extraction models that focus on improving our new metric curve

In our work, the ranking improvement method we designed had mitigated effects on the curve. In the artificial dataset, it decreased the error at the beginning but increased it at the end, while in the Market-1501 and VSNet dataset, it didn't decrease the error at all and only increased it at the end. Determining a new and finer ranking improvement method might a good way to improve our metric curve and thus the association performances.

On the Market-1501 and VSNet dataset, the fact that we did not manage to improve the new metric curve using the ranking improvement method with the

certainty coefficient we generated, might also indicate that the certainty coefficients were not correct, i.e. did not indicate how much we could trust the distance associated with a pair. Determining a way of generating certainty coefficient that are indicative of how much we can trust the distance of a pair, using the side information outputted by VSNet (the body part masks), might lead to an improvement of the new metric curve by using the same ranking improvement method.

Bibliography

- [1] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao and Steven C.H. Hoi, "Deep Learning for Person Re-identification: A Survey and Outlook", 01 2020
- [2] V. Somers, A. Alahi, and C. D. Vleeschouwer, "Body Part-Based Representation Learning for Occluded Person Re-Identification" 11 2022
- [3] Wenhan Luo, Xiaowei Zhao and Tae-Kyun Kim, "Multiple Object Tracking: A Review", 09 2014
- [4] G. Ciaparrone, F. Luque Sanchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera, "Deep Learning in Video Multi-Object Tracking: A Survey", *Neurocomputing* (2019), doi: <https://doi.org/10.1016/j.neucom.2019.11.023>
- [5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, Dec. 2006, doi: 10.1145/1177352.1177355.
- [6] A. Kumar K.C., D. Delannay and C. De Vleeschouwer, "Iterative hypothesis testing for multi-object tracking in presence of features with variable reliability", 2010
- [7] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static bodypart detection", *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, p. 951–958, 2006
- [8] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008, doi: 10.1155/2008/246309.
- [9] N. Bastien, "A comparative analysis of deep Re-Identification models for matching pairs of identities", 2020, UCLouvain, Master's Thesis
- [10] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-Scale Feature Learning for Person Re-Identification." *arXiv*, Dec. 18, 2019. Available: <http://arxiv.org/abs/1905.00953>
- [11] A. Kumar K.C., L. Jacques, and C. De Vleeschouwer, "Discriminative and Efficient Label Propagation on Complementary Graphs for Multi-Object Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 61–74, Jan. 2017, doi: 10.1109/TPAMI.2016.2533391

- [12] C.-Y. Wang, A. Bochkovski, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.” arXiv, Jul. 06, 2022
- [13] C. Kim, F. Li, and J. M. Rehg, “Multi-object Tracking with Neural Gating Using Bilinear LSTM,” in *Computer Vision – ECCV 2018*, vol. 11212, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 208–224. doi: 10.1007/978-3-030-01237-3_13
- [14] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric.” arXiv, Mar. 21, 2017
- [15] K. Fang, Y. Xiang, X. Li, and S. Savarese, “Recurrent Autoregressive Networks for Online Multi-Object Tracking.” arXiv, Mar. 03, 2018.
- [16] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, “POI: Multiple Object Tracking with High Performance Detection and Appearance Feature,” in *Computer Vision – ECCV 2016 Workshops*, vol. 9914, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 36–42. doi: 10.1007/978-3-319-48881-3_3.
- [17] [1] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004, doi: 10.1109/MAES.2004.1263228.
- [18] A. Roshan Zamir, A. Dehghan, and M. Shah, “GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs,” in *Computer Vision – ECCV 2012*, vol. 7573, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 343–356. doi: 10.1007/978-3-642-33709-3_25
- [19] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable Person Re-identification: A Benchmark,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1116–1124. doi: 10.1109/ICCV.2015.133.
- [20] K. Zhou et-al, deep-person-reid, (2019), GitHub repository, <https://github.com/KaiyangZhou/deep-person-reid>

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl