FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Inter-Camera Multi-Object Tracking Using Transfer Learning from Synthetic Dataset
Student:	Bc. Erik Hulmák
Supervisor:	Ing. Filip Naiser
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 25 January 2023 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Instructions

Inter-Camera Multi-Object Tracking Using Transfer Learning from Synthetic Dataset

Managers of complex buildings such as shopping malls and office spaces face various challenges, including optimizing lighting and heating based on occupancy, determining rent amounts, and placing and targeting advertisements. To better understand their customers, it is vital to have systems for pedestrian detection, gender and age prediction, and inter-camera identity preservation. At iC Systems.ai, s.r.o., we are working on developing such systems. The current solution for preserving identity among multiple cameras in complex buildings is described in [1].

This thesis aims to extend this system for preserving identity among multiple cameras in complex buildings. To do this, the student will first conduct a literature review on the topic. They will then design a solution for generating synthetic datasets that will be used to animate and simulate human behaviour. Using both artificial and real data, the student will design and train a neural network that encodes image crops into global descriptor vectors. These vectors will be used to measure the similarity between pedestrian image crops, taking into account the orientation of objects relative to the camera. The student will also implement spatio-temporal inter-camera identity matching using these descriptor vectors. Finally, they will evaluate the overall system.

[1] Erik, Hulmák. Re-identifikace osob v systému kamer. BS thesis. České vysoké učení technické v Praze. Vypočetní a informační centrum., 2021.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 25 January 2023 in Prague.

Master's thesis

INTER-CAMERA MULTI-OBJECT TRACKING USING TRANSFER LEARNING FROM SYNTHETIC DATASET

Bc. Erik Hulmák

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Ing. Filip Naiser
June 29, 2023

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Hulmák Erik. *Inter-Camera Multi-Object Tracking Using Transfer Learning from Synthetic Dataset.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

List of Figures

List of Tables

Firstly, my deepest gratitude goes to my supervisor, Ing. Filip Naiser, whose persistent guidance, unwavering faith, and infectious enthusiasm have been invaluable. Our enlightening discussions on theory, engineering, and the broader landscape of work ethics were truly instrumental. My partner deserves special mention and deep appreciation. Her infinite care, support, and, not to forget, her culinary talents played a substantial role in the successful execution of this project. I also wish to extend my heartfelt thanks to my friends and family. Their nurturing support during my studies has been a beacon of warmth and strength. And finally, a note of affection for my dog, Yllo. His boundless optimism, encapsulated in his fluffy demeanor, brought a sense of lightheartedness and perspective to my journey.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. I further declare that I have concluded an agreement with the Czech Technical University in Prague, on the basis of which the Czech Technical University in Prague has waived the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant to Section 60(1) of the Copyright Act. This fact does not affect the provisions of Section 47b of the Act No. 111/1998 Coll., on Higher Education Act, as amended.

In Prague on June 29, 2023 . . . .

Abstract

Multi-Target Multi-Camera Tracking (MTMCT) can provide invaluable insights into human behavior and movement patterns. An effective tracking infrastructure can significantly influence service refinement, public safety, and resource management. Unfortunately, developing such intricate systems is costly due to the challenges of obtaining large datasets of sufficient quality. To mitigate this, we created and incorporated synthetic data into the training process. This project has both scientific and practical objectives. The scientific objectives include using synthetic datasets and orientation estimation to create a robust visual feature extractor. The practical goal is implementing a powerful MTMCT solution that utilizes available resources to achieve optimal performance. The results of this work will be applied in over ten large complex buildings across Europe. We have constructed four large datasets to address both objectives, each serving different purposes. We utilized multi-task curriculum learning to develop robust models to build and solve the assignment problem. The efficiency of the proposed methodology has been demonstrated in a simulated environment with convincing results. We have shown the benefit of synthetic data use, particularly for smaller datasets.

Keywords computer vision, multi-target multi-camera tracking, synthetic data, multi-task learning, person re-identification, image retrieval, assignment problem

Abstrakt

Multi-Target Multi-Camera Tracking (MTMCT) je systém, který přináší cenné informace o pohybu a chování lidí a dokáže spolehlivě re-identifikovat návštěvníky bez narušení jejich soukromí. Výstupy jsou následně využity pro zkvalitňování služeb a zajišťování veřejné bezpečnosti. Vývoj takovýchto systémů je nákladný kvůli vysokým nárokům na kvalitu a čistotu trénovacích dat. Za účelem jejich snížení jsme se rozhodli využít generovaných dat. Účel práce je primárně akademický, ale předpokládáme, že dosažené výsledky budou brzy nasazeny v praxi, a to do více než deseti komplexních budov po celé Evropě. Hlavním přínosem práce je využití víceúlohového tréninku na čtyřech námi vytvořených datových sadách, které velikostí přesahují 0.5M obrázků. Metodologie vede k lepší generalizaci modelu a pomocí postupného zvyšování náročnosti tréninku jsme se výrazně posunuli oproti předchozím verzím. Výsledky demonstrují užitečnost generovaných dat a navrhovaných metod, a to zejména v případě, kdy je reálných dat nedostatek.

Klíčová slova počítačové vidění, mutli-object multi-camera tracking, generovaná data, hluboké učení, více-úlohové učení, re-identifikace osob

Introduction

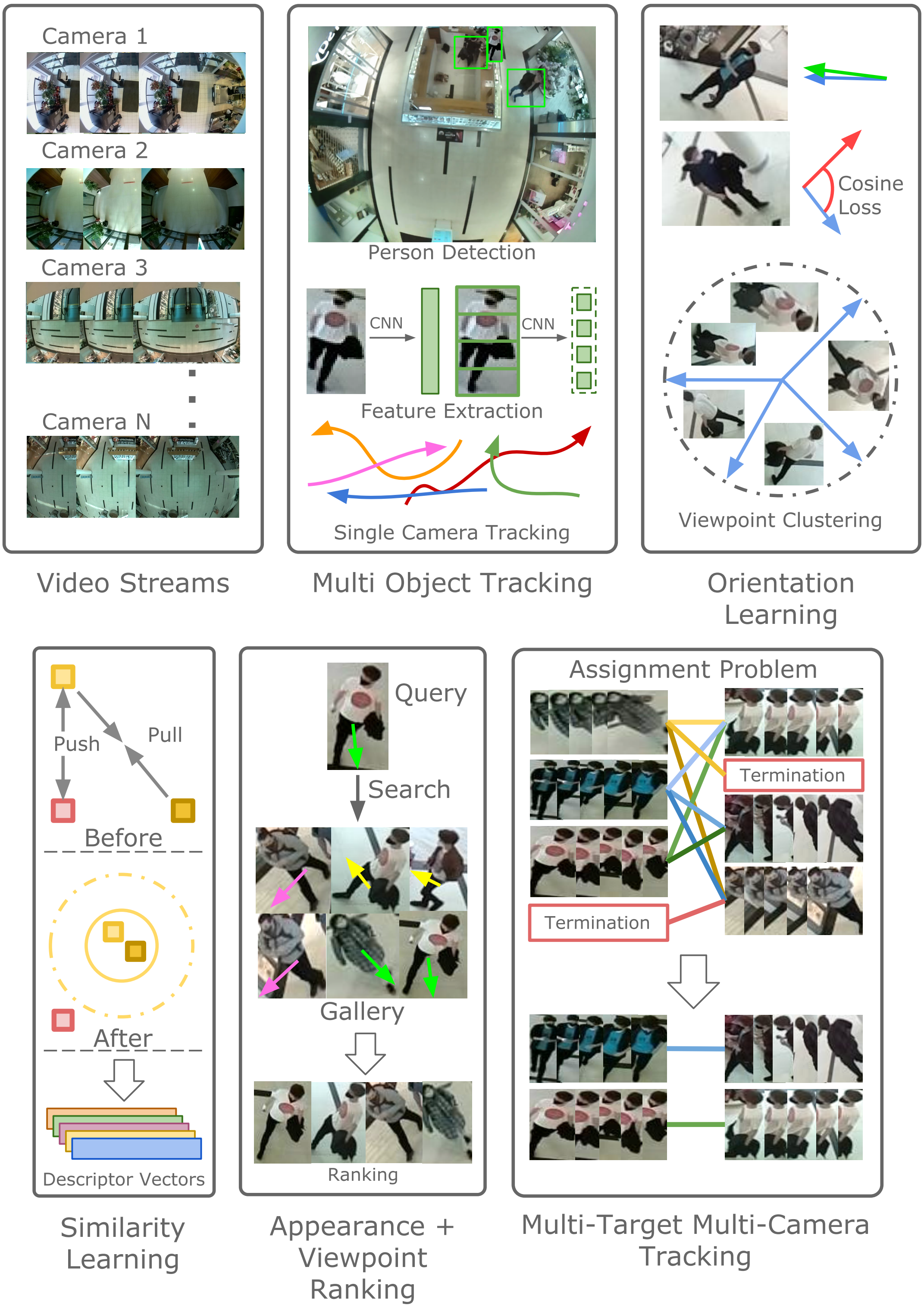
The rise of Artificial Intelligence (AI) and Machine Learning (ML) technologies has led to transformative advancements across many industries, optimizing tasks and processes. One essential application of these innovations is people counting, an increasingly vital tool for managing complex buildings. By gaining insights into customer profiles and behavior, building managers can enhance services, optimize energy consumption, and effectively control people flow within their facilities. Similar systems are also often used for security and anomaly detection because each sensor can recognize suspicious behavior and alert the security worker.

At iC Systems.ai, we are developing a people-counting solution strategically employing low-resolution cameras mounted on ceilings. Each sensor performs the tracking on edge, meaning that all computations are happening on the sensor without an external server. This makes the system cost-effective and secure, as transmitting or storing video footage is unnecessary. The trade-off is lower resolution, weaker computational power, and performance. Whenever we use the term ”re-identification,” we only mean it from a local perspective. That way, matching customers across multiple visits is nearly impossible. Our method is based on visual and textural image features without facial recognition. If, for example, someone takes their jacket off, we cannot match him to his previous self. Please note that each person published in this thesis is a contracted actor.

My primary role at the company involves Multi-Target Multi-Camera Tracking (MTMCT) and pedestrian re-identification. MTMCT is a very challenging problem since installed cameras are often inexpensive with various lens distortions and low-resolution output. Cameras are placed far apart to reduce costs, and their field of view does not always intersect. Visual features of tracked objects are weak and often occluded. We must face different viewpoints, color and illumination changes, various backgrounds, unreliable detection sequences, and bounding boxes. In addition, the number of people is typically not known in advance, and the amount of data to process is enormous. We need a high-quality dataset for training identity embeddings to overcome these issues.

In our prior work, we faced challenges in creating such a dataset. The task requires hard sample mining that exploits the dataset's slightest impurity. One potential solution to this issue involves generating a synthetic dataset using 3D graphics. With perfect control over the digital space, we can simulate various scenarios and extract metadata that would otherwise be very expensive to annotate and almost impossible to obtain.

The research question is: How can we generate a synthetic dataset for re-identification? How can we leverage the data for training a high-performant model? And how to utilize model capabilities to perform multi-camera multi-target tracking. To address this, we will train a Siamese network for re-identification on multiple tasks and aim to develop a robust solution that functions efficiently under all possible conditions. Hopefully, our work will contribute to the growing field of AI and ML and help managers of complex buildings improve their services and optimize their operations.



■ **Figure 1** The diagram outlines key components of our work. Multi-Object Tracking is performed simultaneously on multiple video streams via an end-to-end tracker [1] or a proposal-based system [2]. Then, a multi-task model estimates target orientation and calculates identity embeddings. Orientation, alongside visual descriptors, are involved in viewpoint-based retrieval. This technique and spatiotemporal constraints narrow down candidates for the correspondence matrix. The Hungarian algorithm solves the assignment problem, allowing the reconstruction of the target's original trajectory across cameras.

Chapter 1
Background

1.1 Single-Camera Tracking

Single-camera tracking, also known as Multiple-object tracking (MOT) [3], tracks an object or multiple objects within a single camera view. The traditional approach is tracking by detection, although end-to-end algorithms also exist. The first step in single-camera tracking is object detection [4], which involves identifying and classifying [5] the object of interest in each frame of the video. Once an object has been detected in a frame, its location is recorded, a bounding box is drawn around it, and its visual features are extracted as a local descriptor for later re-identification [6].

A track is a temporal sequence of object detections in consecutive frames. The tracking algorithm aims to maintain the object's identity over long durations.

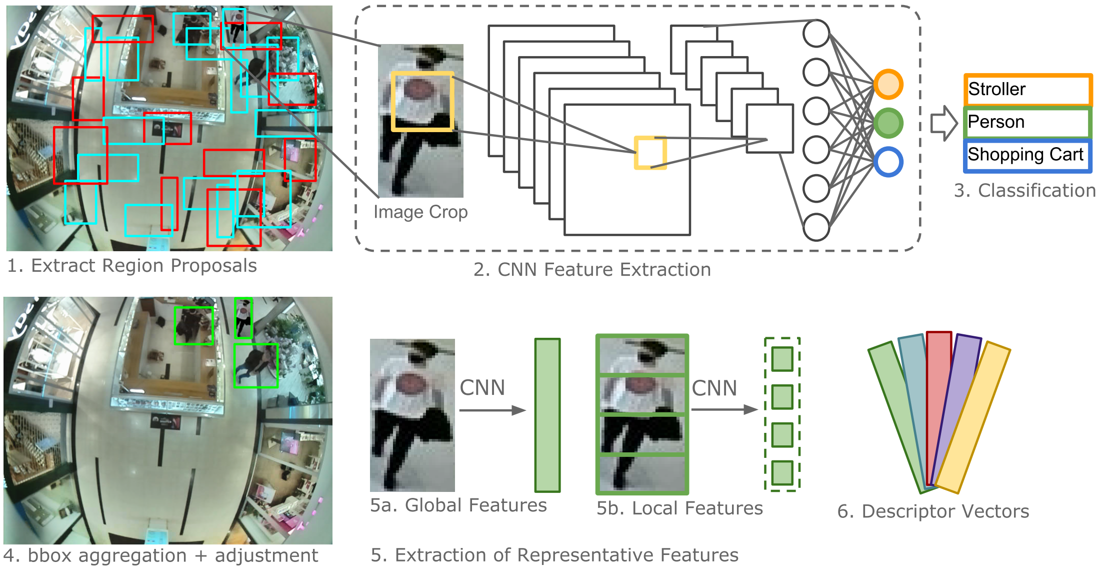
Object Detection

Object detection [2, 7, 8, 9] is one of the fundamental fields of computer vision. In our setting with limited computational power, we can't afford to use state-of-the-art detectors like YOLOv7 [10, 11] (Although hardware and software development progresses quickly). Instead, we use pipelines similar to the ones used for autonomous driving [12] or traditional computer vision [13] that are known for their lightweight and reliable process.

One popular approach to object detection is the sliding window technique, which involves scanning a window of fixed size over the entire image and classifying each window as either containing an object of interest or not. While this method can be effective, it can also be computationally expensive [14], particularly when applied to high-resolution images.

To address the speed/accuracy tradeoff [15], a more efficient approach to object detection is to use a pipeline that combines region proposals, classification, and bounding box regression [13]. This approach involves first generating a set of candidate regions within the image likely to contain an object of interest, then using classification and bounding box regression to refine the regions and output the final object detection.

With the rise of Transformers, many interesting methods were proposed. An end-to-end detection network DETR [16] and Deformable DETR [1] achieved state-of-the-art performance at the time.


1. Extract Region Proposals
Image Crop
2. CNN Feature Extraction
Stroller
Person
Shopping Cart
3. Classification
4. bbox aggregation + adjustment
CNN
5a. Global Features
CNN
5b. Local Features
6. Descriptor Vectors
5. Extraction of Representative Features

■ **Figure 1.1** An example of a proposal-based object detection pipeline. The system first extracts the region proposals. Each image crop is then classified with a score value. Post-processing steps include bbox refinement and aggregation using the non-maxima suppression algorithm. Representative features can be global in the form of identity embedding or local in various forms (Bag of words, key points, etc.)

Region Proposals

To extract region proposals, we move a sliding window that changes in size and shape depending on the image region, examining each image region for potential objects. A region proposal algorithm is utilized to reduce the number of bounding boxes that need to be evaluated thus reducing the computational load immensely.

One popular method for generating region proposals is the Selective Search algorithm [17], which combines adjacent segments in the image based on similarity in color, illumination, texture, size, and shape. Another popular method is Region Proposal Networks (RPNs) [15], neural networks designed to generate region proposals directly from an input image.

Classification

Once the region proposals are generated, they are passed through a deep learning-based classifier, such as a Convolutional neural network (CNN) [18, 19], to determine the presence and class of objects within each proposal. The CNN extracts features from the region proposals, and a fully connected layer classifies the proposals into object classes or backgrounds. Features can also be extracted in a more traditional manner using Histogram of Oriented Gradients (HOG) [13] or with Haar-like features [20, 21]. For classification Support vector machine (SVM) [22] can be used. This step filters out false positives and identifies the most probable objects within the image.

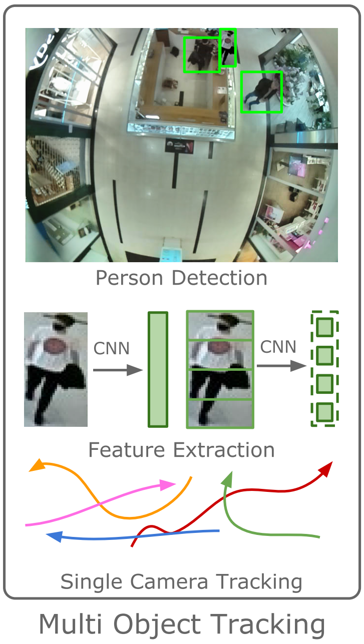
We use Bounding Box Aggregation to combine or cluster multiple overlapped bounding boxes into one final detection. [4] The Non-Maxima Suppression (NMS) [21] algorithm selects the bounding box with the highest classification score, discarding any neighboring boxes with significant overlap (based on a predefined Intersection over Union (IoU) threshold). The proposed regions only sometimes capture the object well. Therefore, we often adjust the regions using a bounding box regression.

Identity Embedding

Once the objects have been detected and localized within the image, an identity embedding function extracts descriptors for re-identification. This function processes the cropped images of the detected objects and generates a feature vector (or descriptor) for each object. These feature vectors serve as compact representations of the object's appearance. They can be compared using a distance metric (e.g., Euclidean distance) to determine if two objects from different images or camera views are the same. The fact that simple descriptors like Color Names [23] or LOMO [24] work well demonstrates the importance of visual features. Traditional keypoint descriptors like LAF [25], LBP [26], and ORB [27] didn't prove to be useful. On the other hand, SIFT [28], alternatively RootSIFT [29], used previously in relevant works [30], could serve as a simple baseline. Finally, the deep learned descriptors [31] outperform any handcrafted alternative by a significant margin. Deep learned descriptors are further covered in the ReID section (sec: 1.2.1).

Multiple-Object Tracking

The current state-of-the-art multi-object tracker [32] is based on the Hungarian algorithm [33] that connects detections obtained via YOLO [10]. Trackers often solve a global optimization problem by continuous energy minimization [34, 35], or attention-based Transformer pipeline [36, 37] that originated as DETR [16, 1]. The biggest disadvantage of the previously mentioned methods is their complexity since they solve a global optimization problem. Accurate object detections allow for a much simpler IoU-based tracking algorithm [38] that can be extended to a Visual-IoU tracker [39].



■ **Figure 1.2** Single-camera multi-object tracking starts with object detection on each frame. Then runs a tracking algorithm that connects detections into sequences of the same identity. The algorithm must allow new tracks to begin and finished tracks to terminate.

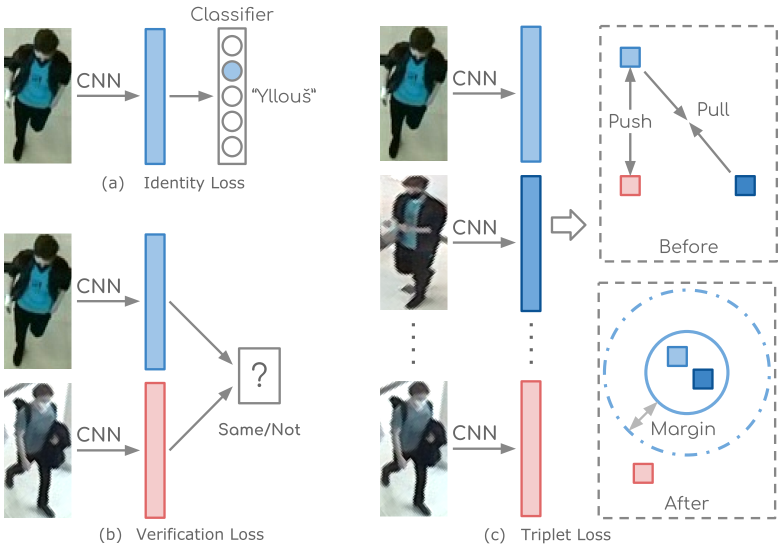
1.2 Multi-Target Multi-Camera Tracking

MTMCT aims to group occurrences of every person at all times from video streams taken by multiple cameras. The resulting identity ensembles are helpful for visual surveillance, anomaly detection, crowd behavior analysis, and many more applications.

MTMCT and Person Re-Identification (Re-ID) are closely related, but they differ in objective. Re-ID ranks identity distances to a query, while MTMCT ultimately solves an assignment problem. The performance of MTMCT is evaluated as a classification error rate [40], while that of Re-ID is as a ranking performance. However, training with a loss of the MTMCT type is very expensive [41]. Therefore we divide the task into two parts. The first part is building a strong Re-ID system. The second part incorporates Re-ID as a tool for solving a global assignment problem.

1.2.1 Deep Metric Learning and Curriculum Learning

We must extract a global feature vector for each person's image to capture discriminative cues. Similarity learning is a subfield of machine learning that focuses on learning and measuring the



(a) Identity Loss

(b) Verification Loss

(c) Triplet Loss

■ **Figure 1.3** Overview of various identity learning methods. Methods often intersect and can be used jointly. (a) Identity Loss - classification of pedestrians, suitable for closed world domains. (b) Verification loss - allows us to decide whether two embeddings represent the same person. (c) Triplet loss - tends to minimize interclass distance while maximizing margin to others.

similarity or distance between two objects. It aims to determine how alike or dissimilar two samples are based on their feature vectors. In many applications, the objective is to compare samples and identify those belonging to the same class or category [42, 43].

These paradigms can be described through a relevant loss function. There are multiple widely studied loss functions with their variants for a person Re-ID, including the identity loss [44], verification loss [45, 46], and triplet loss [47].

Identity loss - The training process is derived from the image classification problem, where each identity is a distinct class. The prediction is usually encoded with the softmax function. Consider input image x_i with identity label y_i. Probability of x_i being recognized as y_i is represented by $p\left(y_i \mid x_i\right)$. Then the identity loss is computed by the cross-entropy loss.

$$\mathcal{L}_{id} = -\frac{1}{n} \sum_{i=1}^{n} \log\left(p\left(y_i \mid x_i\right)\right) \tag{1.1}$$

However, identity loss is applicable only if a limited number of identities exist. Any new identity in the training set is necessarily out-of-distribution. Using the identity loss on top of other losses is usually beneficial but impractical in the open-world scenario.

Verification loss - Pedestrian re-identification can also be approached as a validation problem. Unlike the classification loss, a network trained using verification loss necessitates two input images. Comparing the feature information from both images establishes whether the two input images depict the same pedestrian. It is possible to use either binary verification loss or a contrastive loss [45, 46]. The contrastive loss optimizes a pairwise distance of two feature vectors x_i and x_j, enabling validation and ranking. The function is parametrized by a distance threshold ρ.

$$\mathcal{L}_c = \delta_{ij} \left\|x_i - x_j\right\|_2^2 + \left(1 - \delta_{ij}\right) \left[\rho - \left\|x_i - x_j\right\|_2^2\right]_+ \tag{1.2}$$

Where δ_{ij} is a binary label indicator ($\delta_{ij} = 1$ when x_i and x_j are co-identical and $\delta_{ij} = 0$, otherwise).

Triplet loss - The objective is to ensure that the distance between the positive pair is smaller than the distance between the negative pair by a predefined margin ρ. A triplet consists of a co-identical anchor x_a, a positive x_p sample pair, and one negative sample x_n with a different identity.

$$\mathcal{L}_{trip} = \left[\|x_a - x_p\|_2^2 - \|x_a - x_n\|_2^2 + \rho \right]_+ \tag{1.3}$$

The combination of triplet/contrastive loss and identity loss is mutually beneficial and used in many Re-ID papers [6].

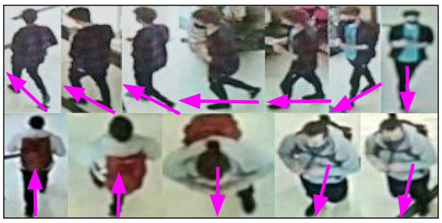
Authors of the quadruplet loss [48] propose to form quadruplets by adding another negative sample. The two negative samples are of different identities. The quadruplet loss leads to the model output with a larger inter-class variation and a smaller intra-class variation than the triplet loss. It is because of a constraint that demands a margin between unrelated image pairs. In our previous work [49], we achieved the best performance using this method.

The previously described methods share a common challenge: tuples that quickly satisfy the loss equations do not contribute significantly to the training process but merely pass through the network. This necessitates hard sample mining, a technique focusing on training with more difficult examples that activate the loss and contribute to training. Hard sample mining is described in more detail in Section 3.1.4. However, if the batch of training samples becomes too difficult, it can increase the risk of gradient explosion, a problem where gradient values become excessively large, as explained in [50]. A technique to balance these factors and improve model performance while mitigating the risk of gradient explosion is curriculum learning [51]. Curriculum learning is a methodology that organizes the training samples in a meaningful order, typically from easy to difficult. This strategy has been successfully applied in a wide range of tasks. However, determining how to rank the samples in terms of difficulty and setting the appropriate pacing for training often presents challenges. Fortunately, we have a simple way to measure the difficulty of data samples. For more information on how we implement these techniques, please refer to Sections 3.1.4 and 3.1.1.

Local Feature Learning

The field of pedestrian re-identification largely categorizes learning-based methods into two groups: global feature learning and local feature learning. Global feature learning techniques extract a single comprehensive feature from a pedestrian image. However, these methods often struggle to capture the granular details necessary for accurate Re-ID.

In contrast, local feature-based learning approaches emphasize distinct image regions that contain critical information. These regions can be suggested via various methods, including manual annotation, pose estimation, hardcoded horizontal division, or even neural networks. Local-based approaches potentially alleviate challenges associated with occlusion, errors in boundary detection, and variations in view and pose. Local approaches can suffer from misalignment. Consequently, researchers often propose a hybrid approach, combining global and local features for more accurate and reliable re-identification.



■ **Figure 1.4** Examples of orientation vectors. The bottom row shows the complexity of spherical space, where categorization is inadequate. Determining a person's view category is uncertain as they walk beneath the camera.

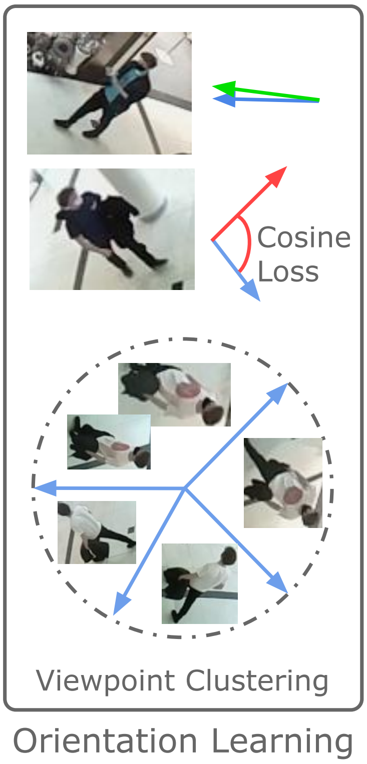
Viewpoint Bias

Significant variations in visual and textural features can be observed when viewing a subject from different angles, as shown in fig: 1.4. Recognizing a pedestrian from different angles is often impossible, even for a trained human eye. A considerable body of research is geared towards creating viewpoint invariant Re-ID frameworks [52]. For instance, the authors in [53] proposed a method that divides an image equally into six horizontal stripes, with a single histogram for each stripe. This local feature approach has been widely adopted in viewpoint invariant person representation. Alternatively, others have disregarded the view information entirely, instead building a shared space where the view-specific bias can be mitigated [54]. Recently, researchers in [55] suggested a novel approach combining view-specific and view-invariant features to achieve state-of-the-art performance. Our previous work [49] neglected the view bias during the learning phase. Surprisingly, even with low-resolution constraints, this proved to be a reasonable baseline.

Existing techniques, such as those previously discussed, operate under the assumption of a front-view dataset, where the viewing angle is subject to a circular geometric pattern, as shown in figure 1.5. Often, authors will quantize orientation into four (or sometimes eight) categories: front, back, left, and right [55, 56]. However, these categories prove inadequate in a top-down-view setup, given that the geometry becomes spherical and the object of interest can move directly beneath the camera.

Our previous experiments encountered challenges with pedestrian orientation categorization. Also, quantifying a continuous space is confusing for annotators. We suggest treating the viewpoint as a regression problem rather than a classification.

We define the orientation as a unit vector in image space that aligns with the direction the person is facing (fig: 1.4). The orientation Loss is then the Cosine Distance Loss 1.4. This method allows us to capture the orientation in a manner that overcomes the limitations associated with strict categorization, potentially leading to more accurate person re-identification models.



■ **Figure 1.5** The orientation vector represents the direction a person is facing. With knowledge of position, we can estimate from which angle we see the target. Crops with similar views usually share visual features.

$$\mathcal{L}_{ori} = 1 - \frac{x_1 \cdot x_2}{\max\left(\|x_1\|_2 \cdot \|x_2\|_2, \epsilon\right)} \tag{1.4}$$

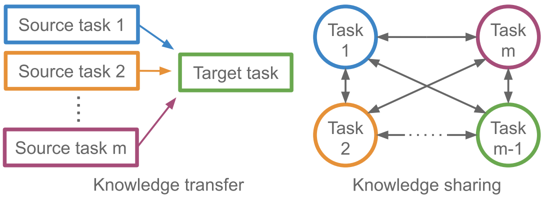
Where x_1, x_2 are orientation vectors, and ϵ is a value to avoid division by zero.

In Chapter 2, we explain a method for acquiring a suitable dataset. Finally, we can measure the two samples' viewing distance, estimating the visual distance relevance. Finally, we can employ the view angle distance in the hard sampling process during the descriptor training.

1.2.2 Multi Task Learning and Transfer Learning

The conventional approach to machine learning presumes that the domains and distributions of the training and testing data are identical. Nevertheless, this is often not the case in many real-world applications. Often, acquiring training data is expensive, difficult, or near impossible.

Therefore, utilizing available data from various domains to create high-performing learners is useful. This methodology is known as transfer learning.



■ **Figure 1.6** A difference between transfer learning and multi-task learning. Transfer learning utilizes knowledge from different domains to achieve higher performance in the target domain. While multi-task learning trains multiple tasks jointly leveraging shared knowledge. The two methodologies intersect, and often one is used to enhance the other.

Multi-task Learning (MTL) [57], a distinct machine learning paradigm, seeks to learn multiple related tasks simultaneously. MTL enhances the overall performance by allowing knowledge gained from one task to be applied to others. Previously, MTL was primarily used to address data sparsity issues, where each task has a limited amount of labeled data. Studies have demonstrated that deep MTL models outperform models focused on a single task [8]. The Multi-Task Learning (MTL) is closely related to transfer learning but differs in fundamentals highlighted in Figure 1.6.

However, it should be noted that knowledge transfer [58] or multi-task learning does not always yield positive results. If domains share few similarities, the effectiveness of knowledge transfer may be limited. For instance, learning to ride a bicycle does not expedite learning to play the piano - a phenomenon referred to as negative transfer. The negative transfer may depend on several factors, such as the domain gap, the learner's ability to transfer knowledge across domains, or misleading domain similarities. These factors exist similarly within MTL.

Both techniques are currently widely used. For instance, a multi-task loss function is used for object detection and semantic segmentation [59, 60]. Deep metric learning combines Identity Loss and Contrastive/Triplet Loss [45, 6, 61]. Using synthetic data for weight initialization in learning keypoint descriptors [62], or using virtual humans rendered under different illumination conditions [63].

1.2.3 Person Re-ID

Person Re-Identification has gained substantial attention in recent years, leading to an increasing number of research papers published in top conferences [64]. The main goal of ReID is to identify the appearances of a person of interest at different times and places. However, a significant gap exists between research scenarios and real-world applications [65]. This project aspires to overcome this gap and aims to deliver a robust solution for practical use.

The open-world application [65, 66, 67] implies that the person in the query may not occur in the gallery set. Under these circumstances, the task can be perceived as person verification. In contrast, in closed-world settings [68, 69], ReID is treated more like a personal identification or classification problem. Like in MOT, various attention-based algorithms have emerged [70, 71], achieving state-of-the-art results.

Viepoint Based Ranking

As discussed in Section 1.2.1, a person's appearance features change based on the viewpoint angle. To mitigate this view bias impact on the ranking, we initially planned to use a Long short-term memory (LSTM) model [72] to accumulate all detections and produce a view-invariant identity embedding. However, such methods often require substantial data and computational resources, which could be challenging under our constraints. As an alternative, we propose a track representation similar in function to the LSTM-based identity embedding. In Figure 3.2, notice a large CNN backbone with two heads: one for the descriptor vector and the other for the orientation vector. The descriptor is trained using similarity learning (sec: 1.2.1), and the orientation is trained using the method proposed in Section 1.2.1.

A track may contain numerous detections, leading to potential redundancy in information. We extract a representative set of features with distinct viewpoints to address this and significantly reduce the track size using k-Means clustering based on the viewpoint angle. We select one representative from each cluster closest to the center, although using the centroid is also an option [61]. Consequently, the track representation consists of k descriptor vectors with corresponding orientation information. The distance between two tracks is then calculated as the mean distance between chosen representatives that are sufficiently close in view. This method normalizes the track length, discards irrelevant information, and substantially decreases the ranking computation time.

Forcing the model to ensemble latent representations into tight clusters is beneficial if the targets share visual features. Otherwise, it may introduce confusion, especially with hard sample mining. Therefore positive pairs should share the viewpoint angle. Once the orientation head is trained (or during its training), we can enable view-sensitive hard sample mining (Section 1.2.1). The difficulty of samples forming pairs (contrastive loss [46]), triplets (triplet loss [47]), or quadruplets (quadruplet loss [48]) can be weighted by the viewpoint distance. It is also possible to weight the loss proportionally to the viewpoint distance. When tuned, only the most relevant and challenging pairs enter the training loop.

The ranking in its original form indeed operates with single images. Since our objective is MTMCT, we prefer robust distance metrics for tracks rather than image crops. Appearance ranking concerning viewpoint for image crops can be done using uncertainty. For each retrieved image, it is possible to return a number that would estimate the similarity uncertainty. The number is proportional to the view angle distance. Then it is a question of weighing the distance, which can result in re-ranking.

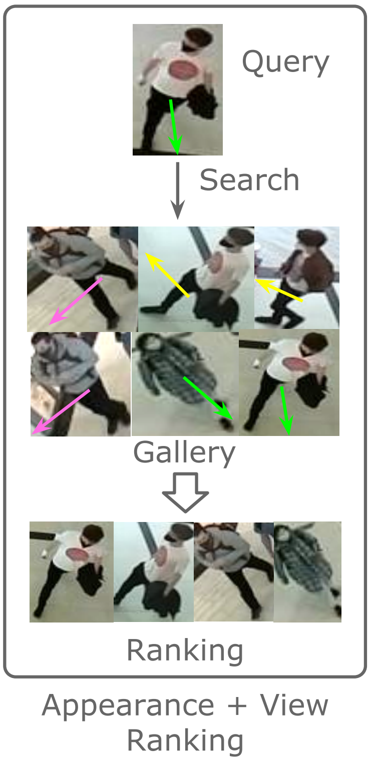


Figure 1.7 It is possible to retrieve similar-looking pictures from a pool of images based on visual features. The viewpoint angle can help to determine how relevant the similarity is.

1.2.4 The Assignment Problem

A recognized approach to achieving classification involves formulating the task as an assignment problem [73, 74] and employing a deterministic algorithm, such as the Hungarian method [33]. Libraries like NetworkX [75], or SciPy [76] offer implementations of augmenting algorithms for balanced [77] or unbalanced [78] linear assignment problems. Authors in [79] utilize the LOMO [24] appearance feature extraction and the Total Least Squares algorithm [80, 81] to estimate an assignment cost matrix. They frame the issue as a Mixed-Binary Integer Programming problem.

Alternately, multi-target tracking data association can be learned through recurrent neural networks [82] or by backpropagating through a network-flow solution [83]. Employing Transformers is also possible [71].

In our prior work [49], we addressed the linear assignment problem using the method proposed by [73, 74]. Originally designed for two cameras, the solution also works for multiple cameras. Construction of the assignment problem correspondence matrix is further described in the method section (sec: 3.2).

1.2.4.1 The Hungarian Method (Kuhn-Munkers algorithm)

While various methods exist for solving the assignment problem, one of the earliest polynomial-time algorithms for a balanced assignment was the Hungarian algorithm (also known as the Kuhn-Munkres algorithm [33]), with a complexity of $\mathcal{O}(V^3)$, where V is a number of nodes. By using the Fibonacci heap, the complexity can be improved to $\mathcal{O}(mn+n^2 \log n)$, where m denotes the number of edges [84]. The Hungarian method can be implemented using a correspondence matrix or a graph. Lecture notes [85] inspired the following description. A curious reader can find all additional proofs and theorems there. A deeper analysis of the algorithm is unnecessary for this work.

▶ **Definition 1.1** (Matching)**.** *A matching is a subset $M \subseteq E$ such that $\forall v \in V$ at most one edge in M is incident upon v. A Perfect Matching is an $M \subseteq E$ in which every vertex is adjacent to some edge in M.*

▶ **Definition 1.2** (Vertex labeling)**.** *A function $\ell : V \to R$. Labeling is feasible if $\ell(x) + \ell(y) \geq w(x,y), \forall x \in X, y \in Y$.*

▶ **Definition 1.3** (Equality Graph)**.** *Graph $G_\ell = (V, E_\ell)$ where $E_\ell = \{(x,y) : \ell(x) + \ell(y) = w(x,y)\}$*

▶ **Definition 1.4** (Neighborhood N_ℓ of $u \in V$ and set $S \subseteq V$)**.** $N_\ell(u) = \{v : (u,v) \in E_\ell\}, N_\ell(S) = \cup_{u\ inS} N_\ell(u)$

▶ **Theorem 1.5** (Kuhn-Munkers)**.** *If ℓ is feasible and M is a perfect matching in E_ℓ, then M is a max-weight matching.*

Algorithm 1: The Hungarian Algorithm

```
Data: Complete, bipartite weighted graph $G = (V, E)$ with partitions $X, Y$
Construct Equality Graph: $G_\ell = (V, E_\ell)$ with initial labelling:
  $\forall x \in X, y \in Y \mid \ell(y) = 0, \ell(x) = \max_{y \in Y}(w(x,y))$
$M :=$ some matching in $E_\ell$
while $M$ is not perfect do
    Pick a free node $u \in X$
    $S := \{u\}, T := \emptyset$
    while True do
        if $N_\ell(S) = T$ then
            update labels (forcing $N_\ell(S) \neq T$) in a following way:
            $\alpha_\ell := \min_{s \in S, y \notin T} \{\ell(x) + \ell(y) - w(x,y)\}$
            if $v \in S$ then $\ell(v) := \ell(v) - \alpha_\ell$
            if $v \in T$ then $\ell(v) := \ell(v) + \alpha_\ell$
        else
            pick $y \in N_\ell(S) - T$
            if $y$ is free then
                Augment $M$ ($u$—$y$ is an augmenting path)
                break
            else
                Extend the alterning tree: $S := S \cup \{z\}, T := T \cup \{y\}$, where $z$ is the node
                  $y$ is matched to
Result: A perfect matching $M \subseteq E_\ell$, where $\ell$ is feasible.
```

1.2.5 Evaluation Metrics

The ranking component of a person Re-ID in multi-target multi-camera tracking necessitates a robust evaluation score.

For each query, an algorithm sorts all gallery samples. The **rank-k** recognition rate signifies the fraction of top-k ranked gallery samples that contain the query identity. The Cumulative Matching Characteristics (CMC) curve [86] is a step function that represents multiple rank-k evaluations for a sufficiently large k.

Although the CMC is a common evaluation metric for person re-identification, it is only accurate when a single ground truth exists for each query. It is because the metric considers only the first match. The always-present ground truth in the gallery also implies a closed-world scenario. Large-scale datasets do not always meet this condition. Another commonly used metric, Mean Average Precision (mAP) [68], measures the average retrieval performance when multiple ground truths are present.

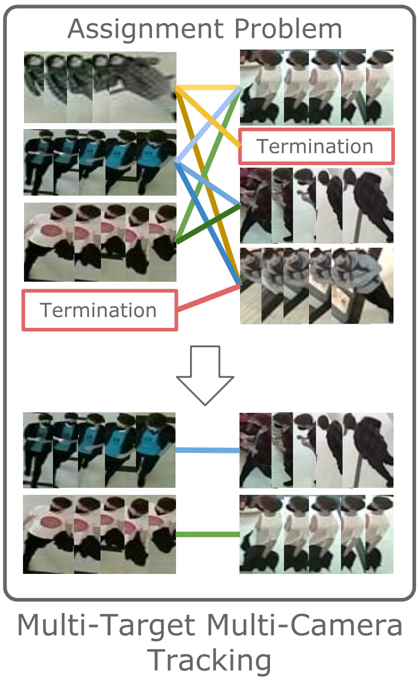
In evaluating MTMCT, we use the counts of false negatives (IDFN), false positives (IDFP), and true positives (IDTP) to compute the Identification Precision (IDP), Identification Recall (IDR), and the corresponding F1 score (IDF1).

IDTP - Correct non-terminal assignments.

IDFP - Incorrect non-terminal assignments.

IDTN - Tracks correctly predicted as terminal.

IDFN - Tracks incorrectly as terminal.



■ **Figure 1.8** The process of solving the assignment problem involves constructing a correspondence matrix by proposing potential connections based on spatiotemporal constraints. By utilizing descriptor vectors, we can calculate assignment costs. The MTMCT is then resolved using the Hungarian algorithm.

IDF1 is a variant of the F1 score, a frequently employed metric in machine learning that combines precision (IDP) and recall (IDR) into a single measure. Within the MTMCT context, precision is the proportion of relevant instances among retrieved samples, while recall is the fraction of relevant samples to the total number of positive samples.

$$IDP = \frac{IDTP}{IDTP + IDFP} \qquad IDR = \frac{IDTP}{IDTP + IDFN} \tag{1.5}$$

(1.6)

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{1.7}$$

Chapter 2
Dataset

To our knowledge, there is currently no sufficient public top-view dataset for person Re-ID. Widely-used datasets such as Market-1501 [68] and VIPeR [87] predominantly focus on non-top-view perspectives and are set in outdoor environments, leading to a significant illumination domain gap. Consequently, we have created a dataset tailored to our specific needs. Re-ID datasets are highly sensitive to noise, and hard sample mining can exacerbate this issue, thereby elevating the cost of data collection. Given these constraints, acquiring a high-quality dataset at scale would be challenging. As a result, we propose using a synthetic dataset and multi-task learning to address these limitations.

2.1 Geometry

Pinhole Camera

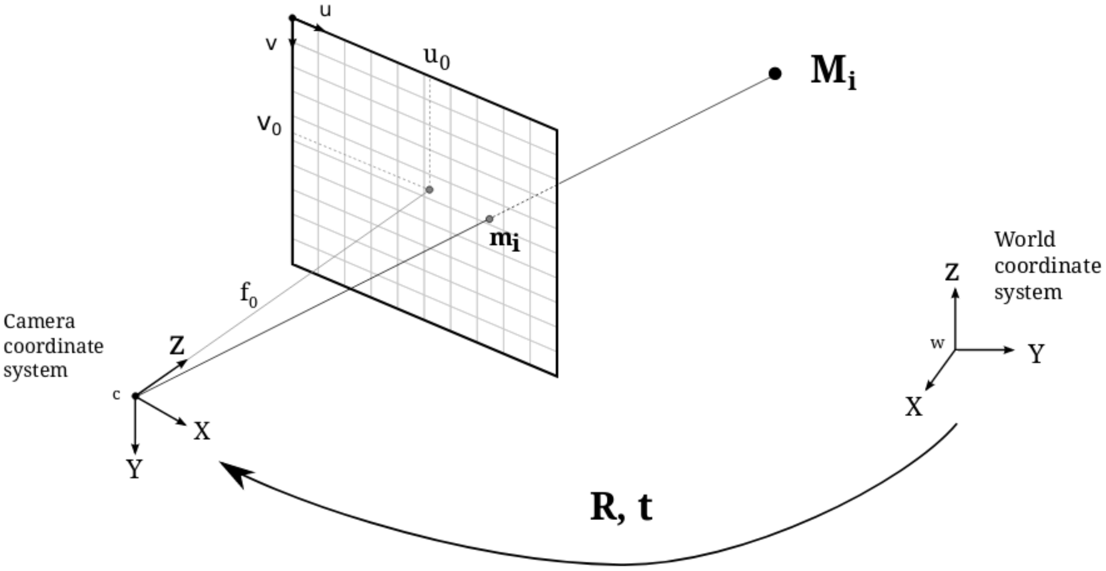
A pinhole camera is the simplest camera model in computer vision that maps between the 3D world and a 2D image. The model uses projective geometry that can be described with the following equation $sm' = K[R \mid t]M'$ [88].

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1}$$

M' is a point in 3D Euclidean space known as the world coordinate system. m' is the projection of the 3D point M' onto the image plane with coordinates $[u, v]^T$ expressed in pixel units. K is the camera calibration matrix, called the intrinsic camera matrix. C is the principal point offset with coordinates $[u_0, v_0]^T$ at the origin in the image plane. f_x, f_y are the focal lengths expressed in pixel units. Camera rotation and translation are expressed with joint rotation-translation matrix $[R \mid t]$ that transforms the coordinates of a 3D point from the world coordinate system to the camera coordinate system.

Camera Calibration

We already understand the relationship between the 3D world and the 2D camera plane, but typically, a third coordinate system is involved. We refer to this space as the Image coordinate space, which is a product of lens distortion and serves as the actual output of a video. Lens distortion is a form of optical aberration where straight lines in the scene do not remain straight



■ **Figure 2.1** Pinhole Camera - The simplest camera model in computer vision that maps between the 3D world and a 2D image [88].

in the image. Examples of lens distortions include barrel distortion (fig: 2.2), fisheye distortion, and pincushion distortion. They are usually a combination of more basic distortions like radial, tangential, thin prism, and tilt distortions.

The equation below [88], describes the relationship between distorted (u_d, v_d) 2D coordinated and undistorted (u, v) coordinates. It covers parameters for radial $(k_1, k_2, k_3, k_4, k_5, k_6)$, tangential (p_1, p_2), and thin prism (s_1, s_2, s_3, s_4) distortion. For simplicity $r = u^2 + v^2$.

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 2p_1 uv + p_2 \left(r^2 + 2u^2\right) + s_1 r^2 + s_2 r^4 \\ 2p_2 uv + p_1 \left(r^2 + 2v^2\right) + s_3 r^2 + s_4 r^4 \end{bmatrix} \tag{2.2}$$

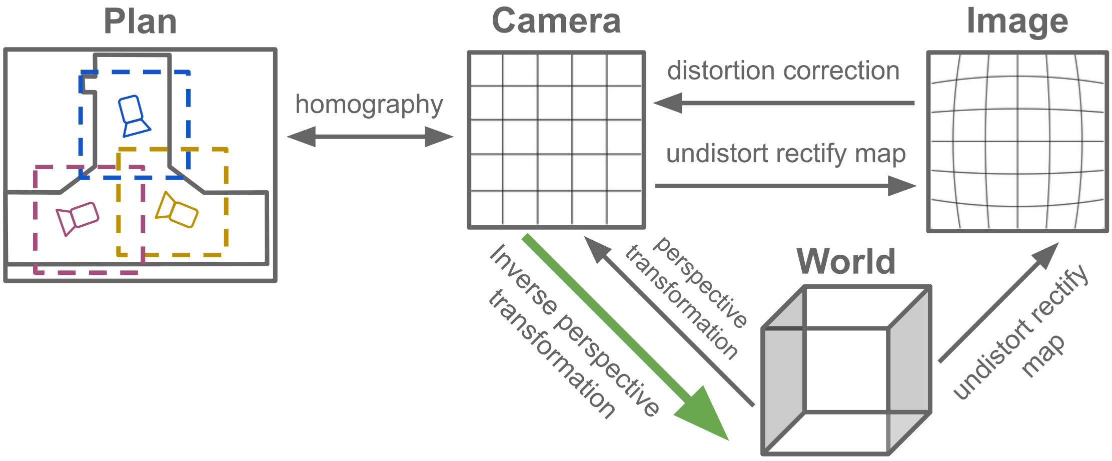
We need to inverse the previously described model to undistort coordinates that leave the physical pinhole camera. To undistort an image, libraries like OpenCV [89] usually distort each pixel in the destination image to match the input image.

Coordinate Transformer

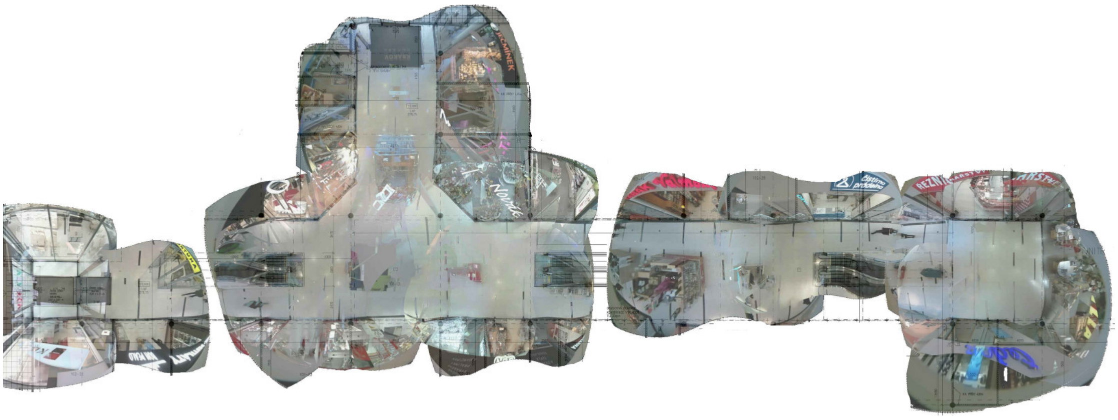
To fully utilize our setup's capabilities, we require a method for transforming between coordinate spaces. We already explained the process of calibration. However, other necessary transitions need to be covered. In later stages, we need a method to project Camera/Image coordinates into the World system. Given that our camera at a given height is orthogonal to the floor, we can calculate an inverse perspective transformation at the corresponding depth and project camera coordinates onto a 3D plane.

2.2 Real Dataset

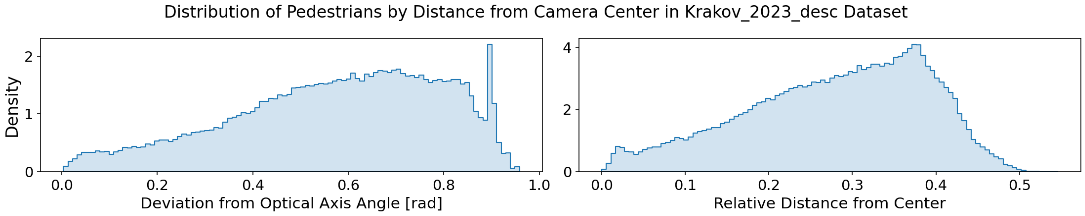
We mounted cameras with overlapping fields of view in the Krakov shopping center. This allowed us to employ our existing single-camera tracking capabilities to identify tracks overlapping in time and space. Firstly, we registered the cameras into a common plan using a perspective



■ **Figure 2.2** Figure depicts the transformation between coordinate systems, with each arrow representing one method in our CoordinateTransformer. The Image-space corresponds to video stream images. With camera calibration, it is possible to correct the distortion and transition to the Camera frame. The inverse distortion for the Camera and World is done using a reference rectify map. A perspective transformation describes the transition between cameras. An inverse perspective transformation is feasible with the knowledge of scene depth, which is crucial for generating synthetic data in shared world space. Each camera's homography was estimated using the Perspective-n-Point (PnP) method [90] (as depicted in fig: 2.3).



■ **Figure 2.3** A projection of more than 20 undistorted camera images on a common floor plan. We manually estimated the homography using the PnP method [90]. Wide Baseline Stereo homography estimation, based on matching local key points [28, 25, 13] and applying the RANSAC [91], proved problematic due to imperfect distortion correction.



■ **Figure 2.4** Unbalanced density of optical axis angle indicates that scenarios, when the pedestrian is captured directly under the camera are rare. This observation supports the distribution of distances from the image center. If the camera lens is not distorted, the optical axis intersects the center of the image. The x-axis in the second plot corresponds to a relative distance if the image size is normalized to be (1, 1).

transformation between two planes (fig: 2.3). As all cameras face directly downwards, we assumed all ground points were at zero height. We identified corresponding points between the virtual building plan and the undistorted camera images, computing the homography by solving the PnP problem [90]. The mapping functions estimated in sec: 2.1 were sufficiently accurate to compute a spatiotemporal distance matrix between relevant tracks. Using the Interval Tree data structure sped up searching for tracks with temporal overlap. We represent the spatial cost of two tracks as the mean distance between detections adjacent in time. Track pairs with sufficiently low cost were bonded and sent to the annotation tool (fig: 2.5). Further reading on this topic is available in our previous work [49].

2.2.1 Annotation Job

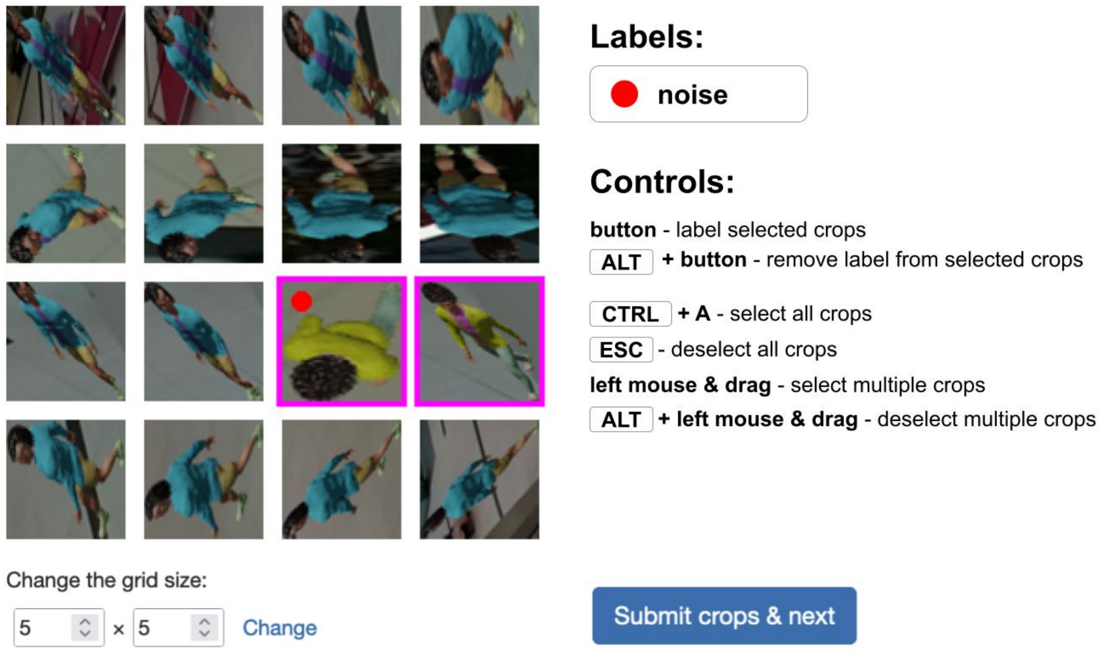
We optimize the single-camera tracking for accurate people counting rather than tracking. The resulting tracks can be noisy, and the identities mismatched, which adversely affects hard sampling during the re-identification learning process. Consequently, we had to annotate and clean the data. In collaboration with Jiří Hulmák, we developed an annotation grid selection tool (fig: 2.5), a web application designed to speed up the annotation process. Every screen contains two tracks that were suggested for bonding. An annotator who encounters a noisy detection must select the corresponding spot and click on a button that assigns a ”noise” label. The types of problems we encountered included: 1. intra-track mismatches, 2. inter-track mismatches, 3. noisy object detections, 4. objects outside the bounding box, and 5. poor quality samples.

We enlisted the help of multiple annotators to label the proposed data. Managing even a small group of individuals to perform a high-precision task accurately proved extremely challenging. To post-process the collected data, we implemented the following steps:

1. Tracks containing fewer than five clean detections were excluded.
2. If all detections from one track were noisy, we excluded both tracks involved in the bond.
3. Some tracks are involved in multiple pairs, appearing multiple times during the annotation process. A detection is conflicting if labeled noisy and clean in different occurrences. All conflicting detections were considered noisy.
4. As track bonds exhibit transitive relations, we searched for connected components, ultimately unifying all identity occurrences into a single identity group.

2.2.2 Orientation Dataset

The distance of embeddings can fluctuate depending on the viewpoint. The viewpoint angle is determined by the object’s rotation and position relative to the camera. The MOT pipeline esti-



Labels:

● noise

Controls:

button - label selected crops

ALT **+ button** - remove label from selected crops

CTRL **+ A** - select all crops

ESC - deselect all crops

left mouse & drag - select multiple crops

ALT **+ left mouse & drag** - deselect multiple crops

Change the grid size:

5 × 5 Change

Submit crops & next

Figure 2.5 The user interface of the grid selection tool for labeling image crops. Crops with a magenta border are selected, and a crop with the red dot is labeled as noise. If necessary, it is possible to define multiple attributes. For better clarity, we filled the grid with synthetically generated images. The real annotation task requires a grid of shape 11x11. Grouping crops by possible identity and displaying them in one batch is desirable.

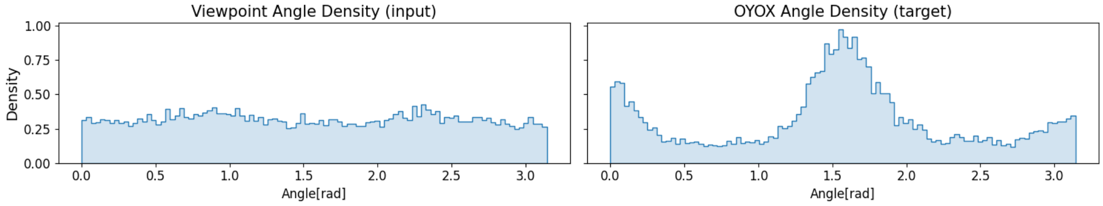


Figure 2.6 By using location coordinates and the orientation vector, we can compute the Viewpoint Angle. It represents the direction from which we observe the target. The distribution is balanced, meaning we have a good representation of how people appear from all angles. The OYOX Angle is the radial distance between the target's orientation vector and the y-axis. A balanced input can project an unbalanced target. The effect is caused by aligning the camera with a corridor so that the resulting image covers as much space as possible. It usually corresponds to the x-axis being parallel to the passage below.

mates the coordinates for each detection, but it is unreliable to derive orientation from location measurements alone. While it is true that a person usually faces in the direction of movement, targets often remain stationary or rotate in place. Such noisy data is not very useful for machine learning applications. However, with the help of reasonable heuristics and filters, we can generate a noisy dataset that humans can review and clean.

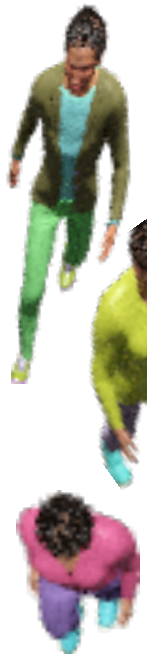
The orientation annotation job was conducted similarly to the initial annotation process (sec: 2.2.1). We began by using strict rules to identify linear, consistent movements. We then applied the Kalman Filter [92] to refine the ground point predictions. Afterward, we generated suggestions for orientation vectors (fig: 1.4). Finally, we used our grid selection tool (fig: 2.5) to filter out noisy suggestions.

2.3 Synthetic Dataset

Synthetic datasets have proven to be useful across various machine-learning domains. They offer an economical means of boosting generalization by pre-training networks and securing robust weight initialization. Synthetic datasets also provide effective regularization capabilities, preventing multi-task learning from divergence. Absolute control over 3D space presents numerous advantages, such as varying perspectives, illumination changes, or color changes. We can obtain semantic segmentation, depth maps, or labels for location, orientation, and pose estimation. Synthetic datasets have been extensively used for action recognition and anomaly detection tasks. More recently, CycleGANs have been used for domain adaptation in autonomous driving, helping researchers to mitigate the negative effects of long-tail distributions.

We can generate synthetic data with a 3D graphics engine [93] or a video game [94, 95]. While these datasets offer high quality, they are predominantly front-view based, which does not meet our needs. To overcome this limitation, we developed our synthetic data generator using Blender [96], in conjunction with the Human Generator v3 plugin [97]. We modeled multi-view scenes, developed a trajectory generator (fig: 2.11, and animated pedestrians to simulate walking.

In terms of volume in this thesis, the synthetic dataset covers only a fraction of the space. However, by the time spent, it was really a challenging engineering journey that took several months of dedicated work. The synthetic dataset is useful, as proven in the experiments Chapter 4. However, in the large scheme of things, this work is only a fraction of what the dataset offers. In IC Systems.ai, s.r.o. (iC), we already employed the synthetic data to multiple different pipelines, and they have proven to be very helpful for various reasons. The synthetic generator is a key that opens many doors.



■ **Figure 2.7** An example of three generated humans. Low render quality is tailored to match the real dataset domain.

2.3.1 Trajectory Generator

The most challenging aspect of creating the synthetic dataset was the engineering process. The second challenge was the problem of generating trajectories that closely resemble authentic human movement patterns.

Our assumptions about accurate human trajectories include the following:

1. Humans tend to walk the shortest path, maintaining a comfortable distance from obstacles.

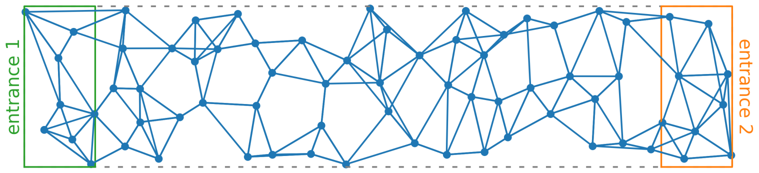
2. People in motion exhibit inertia, meaning they do not typically change direction suddenly.

3. They walk parallel to corridor walls.

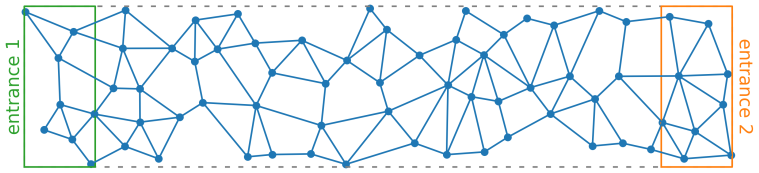
Our objective was to simulate these behaviors while occasionally breaking the rules. For instance, we wanted to incorporate instances where a pedestrian turns around, walks back, or avoids an obstacle by a larger margin than usual.

The central concept behind our trajectory generator involved constructing a network of nodes and paths for pedestrians to follow. We temporarily closed a pathway or node to simulate an obstacle, forcing the person to walk around it. We then randomly augmented the path to achieving reasonable variance while retaining the underlying network structure. Finally, we simulated a continuous, smooth walking curve using a pure path controller (sec: 2.3.1).

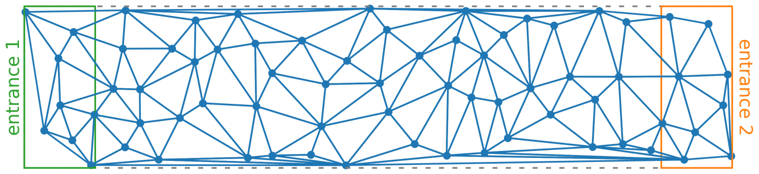
Background Network Generator



■ **Figure 2.8** 5-Nearest Neighbor Graph. Non-planar alternative to graphs below.



■ **Figure 2.9** Gabriel Graph is a subgraph of the Delaunay Triangulation.



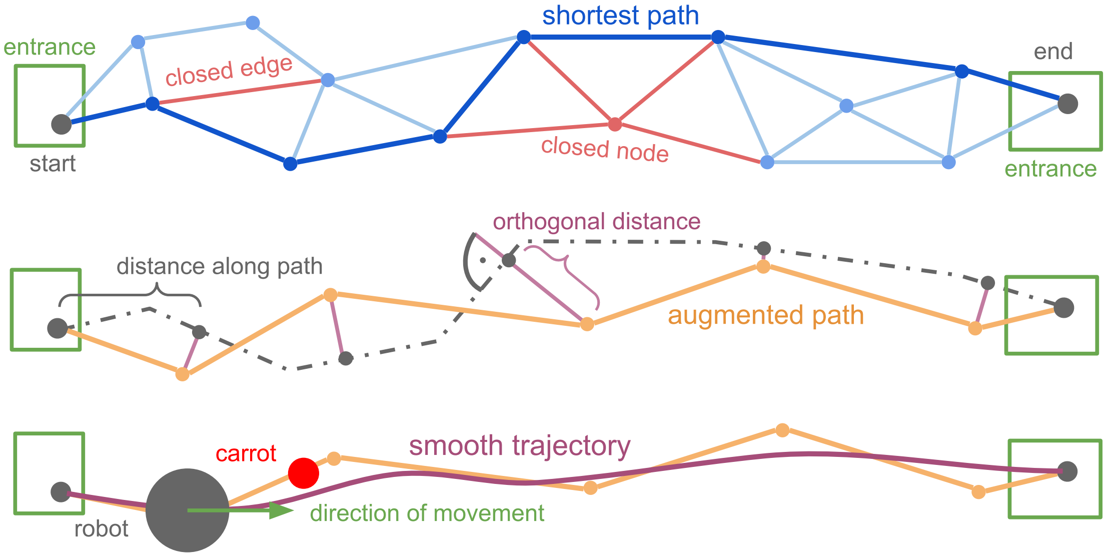
■ **Figure 2.10** Delaunay Triangulation - a suitable structure for trajectory simulation.

■ **Table 2.1** A comparison of three proximity graphs reveals distinct characteristics. The k-nearest neighbor graph (k-NNG) connects each node to its k-nearest neighbors. However, unlike Gabriel and Delaunay graphs, k-NNG is not a planar graph and does not possess a constant vertex degree. Among these proximity graphs, we selected Delaunay triangulation due to its structural advantages, such as density, connectivity, and ease of planting obstacles, making it the ideal choice for our purposes.

The algorithm begins by randomly distributing points within a virtual floor plan while ensuring a minimum distance between nodes. Points too close to one another are removed or slightly repositioned to prevent unwanted clustering. Subsequently, vertices are connected using Delaunay triangulation (Dt) [98] (fig: 2.1). Initially, we experimented with the k-NN graph, but its disadvantages, such as non-constant vertex degree, non-planarity, and over-connectivity of clusters, led us to discard the approach. In contrast, Dt is a planar graph with a constant vertex degree, ensuring the placement of an obstacle always results in a detour. We also considered employing subgraphs of Dt like Gabriel Graphs. They provide properties similar to DT with less computation. The complexity advantage was insignificant, given the reasonable initial sample density and coverage.

Trajectory Augmentation

Initially, we constructed a proximity graph using Delaunay Triangulation. Following this, we generate trajectories in several stages. First, we identify the shortest path between two random entrance locations (fig: 2.11). Subsequently, we sample two sets of distances. The first set, which

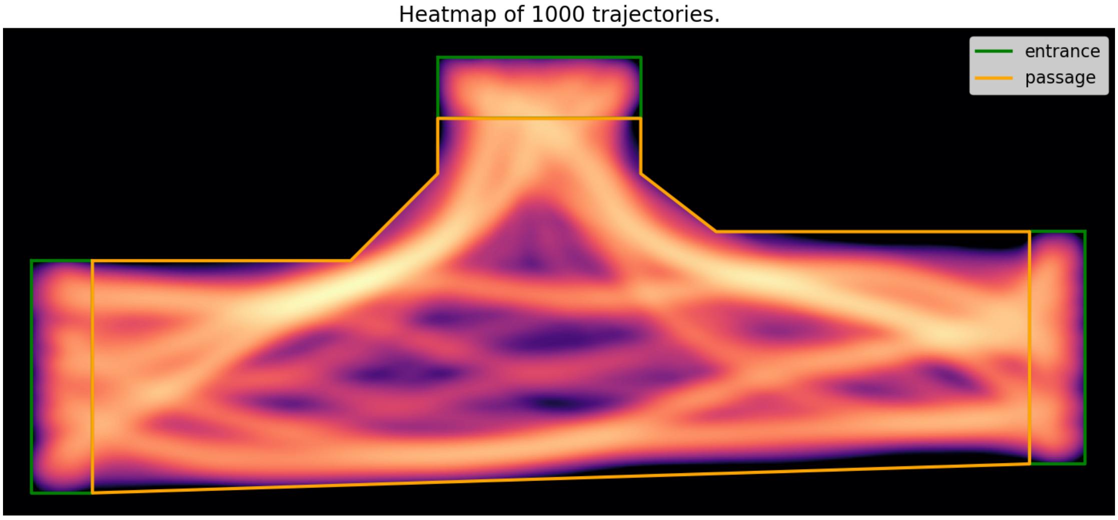


■ **Figure 2.11** Initially, we constructed a proximity graph using Delaunay Triangulation. Following this, we generate trajectories in multiple stages. First, we identify the shortest path between two random entrance locations. After augmenting this path, we create a smooth trajectory utilizing a Pure Pursuit Controller. Finally, after each iteration, we simulate obstacles by closing nodes or edges along the path. Upon completing a predefined number of iterations, we restart the entire process. The algorithm is highly parametrizable, making it an effective tool for simulating all kinds of human locomotion. Please see an animation of this process: `https://youtu.be/l9kU5Lj_hTI`

follows a uniform distribution, represents an offset from the starting point without exceeding the path length. The second set, sampled from the Gaussian distribution, signifies an orthogonal deviation from the path. Both distributions can be parameterized to control the properties of the generated trajectories. A greater distance between points results in a less convoluted path, while a larger orthogonal distance allows for deviation from the structure defined by the background network.

Pure Path Controller

There are various methods to transform a series of points into a curve, such as using the Kalman filter [92] with kinetic energy and small step size, Bézier curves, spline interpolation, polynomial regression, or Bayesian processes like Gaussian process regression. However, one method stands out due to its close resemblance to the actual walking process: the pure pursuit controller. This navigation algorithm, commonly used in robotics to guide a robot along a path, operates on the ”look-ahead” control principle. The controller predicts the robot’s future position along the path and calculates the steering angle required to reach that position. By parameterizing the pure pursuit controller, it becomes straightforward to simulate various walking speeds and path curvatures, making it a flexible and effective tool for simulating human locomotion across diverse scenarios. The algorithm’s high degree of parameterization allows for adjustments to distributions, their properties, distances, and graph density. Factors such as the robot’s motion speed, turning speeds, and look-ahead distance can be altered to influence trajectory smoothness. This adaptability enables the fine-tuning of the algorithm to meet specific requirements, resulting in the creation of diverse and realistic pedestrian trajectories across a wide range of scenarios.



■ **Figure 2.12** In a toy corridor model, we animated 1000 trajectories. They can be visualized at once as a heatmap. The brightness corresponds to the density.

Scene Settings

Blender allows us to set intrinsic camera parameters and establish a projection matrix between camera and world coordinates. Previously, we explained our good knowledge of scene geometry. Therefore, we used known geometry to digitally simulate installations in various shopping malls. However, imitating lens distortion was challenging, as Blender does not support the specifying parameters for barrel distortion. To save time, we opted to generate images without distortion and apply the distortion later if necessary.

A scene consists of multiple cameras and lights, with randomly generated humans featuring random clothing, dimensions, and attributes. Rather than using a 3D background, we employ images from cameras converted into virtual space. Using the described coordinate transformations (sec: 2.1), we project generated trajectories into the shared Blender 3D scene and animate pedestrians walking along the smooth trajectory. Each camera has slightly different lighting, simulating illumination changes. It is also possible to generate a segmentation mask for every object. With a segmentation mask, background alteration during the training process is feasible. Since bounding box information is available, we can crop images and **reduce the dataset size by more than 50 times**.

2.4 Dataset Overview

In Blender, we implemented a pipeline capable of simulating and animating pedestrians. The solution is highly customizable, allowing the parametrization of crowd density, pedestrian motion patterns, and common appearances, including clothing, skin tone, hair color, and facial expression. The generator also allows for changes in both gender and age. The render utilizes a transparent background, which, combined with a bounding box, makes it possible to save only crops, significantly reducing the dataset's size. The transparent background also enables us to change the background to real images during training, thus preventing overfitting, which is common in our data domain. It is because of the static cameras with low background variance. With the additional information provided, it is possible to train pose estimation, orientation, segmentation, and more.

- **Synth** - Using our synthetic data generation, we created **1 803** identities and **6 646** tracks with **234 034** crops.

- **Krakov_2023_desc** - The real dataset, recorded by overlapping cameras. We registered all cameras into the floor plan (2.3). Then with a coordination transformer, we made assignment suggestions for humans to label. Using the annotation tool (sec: 2.2.1), we organized a job for multiple annotators that filtered out most of the noisy smart dataset suggestions. All in all, **20** cameras recorded more than **170** hours of footage with **3 579** identities and **7 385** tracks over **291 602** crops.

- **Krakov_2023_desc_small** - As described in Chapter 4, a smaller subset of the real descriptor dataset was necessary. Therefore we sampled **512** identities of **1 036** tracks, and **38 168** image crops.

- **Krakov_2023_ori** - Real orientation dataset. We developed a heuristic that suggests a direction based on the Kalman Filter prediction for the orientation dataset (sec: 2.2.2). After filtering noisy data using the annotation tool, we were left with **12 753** crops.

- **footfall_background** - We made a fourth dataset to switch the transparent background of the synthetic data. It consists of **20 291** real images. We took the pixel-wise median image to get a real scene without pedestrians. For faster data loading, we randomly extracted multiple sections from each scene.

Chapter 3
Method

This chapter explains interesting implementation details of multi-task learning and the intricacies of building a correspondence matrix for track assignment problems. Finally, we explain the benchmark that we use for the final evaluation.

We chose Python 3.10 as a programming platform, providing us with a broad range of scientific computing libraries. We implemented our models in PyTorch 2.0.1+cu117 [99], which offers robust support for GPU-accelerated computation. Image processing tasks were performed using OpenCV 4.5.5 [89]. For data management, we used the FiftyOne 0.20 package and an instance of MongoDB running in a docker container.

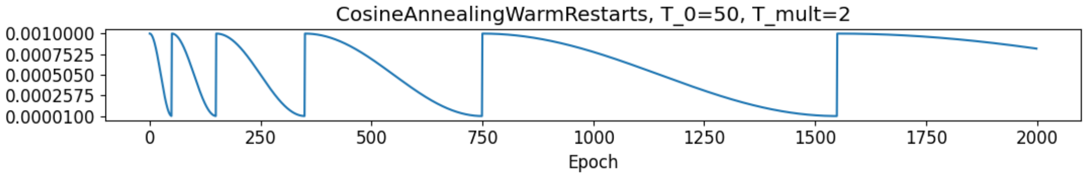
3.1 Multi-Task Learning

The complexity of our dataset allowed for multi-task learning. To balance the benefit-cost ratio, we chose a handful of essential tasks. The training combines semantic segmentation, orientation prediction, and identity embedding on synthetic and real data. The network architecture we used is outlined in Figure 3.2. It is a modified version of the EfficientNet-B0 [100] with 5.3M trainable parameters and 0.39B FLOPS. Its main building block is mobile inverted bottleneck MBConv as described in the MobileNet paper [101]. Authors extended the MBConv layer with the squeeze-and-excitation optimization [102].

In the scheme, there are two inputs. The first one is the image data input that accepts matrices of shape (3, 64, 64). The second input is for side information relevant to the image crop. The metadata is a concatenation of location coordinates, bounding box, camera height, and optical axis angle. This angle is then encoded using a multi-layer perceptron and concatenated with a feature map that leaves the second block.

Data augmentations are simple color, illumination, and transformation changes. We jitter brightness, contrast, hue, and saturation simulating relevant lighting conditions. Since cameras are mounted to the ceiling, orientations of full 360 degrees are possible. We flip the input crops and metadata depending on the image quadrant to ensure a uniform alignment. The angle of pedestrians is then in a 90-degree range. This simple transformation proved to be extremely advantageous.

For our deep learning tasks, we employed the Adam optimizer [103], favored for its efficiency and minimal memory requirement, making it suitable for our large-scale datasets. We set the initial learning rate at 1e-3, a commonly-used value that balances speed and convergence. As a learning rate schedule, we selected the CosineAnnealingWarmRestart (more in the torch library [99]) for its ability to converge effectively during training through periodic learning rate adjustments with a lower bound set to 1e-5. Tensorboard provided real-time monitoring and



■ **Figure 3.1** Cosine annealing with warm restarts - a learning rate schedule. The initial cycle length is 50, and the following cycle is twice as long as the previous one. The values move in the range from 1e-5 to 1e-3.

visualization of the training progress. We recorded various metrics and visualizations for all tasks and validations. We saved model weights every 50 epochs to ensure progress and facilitate potential recovery.

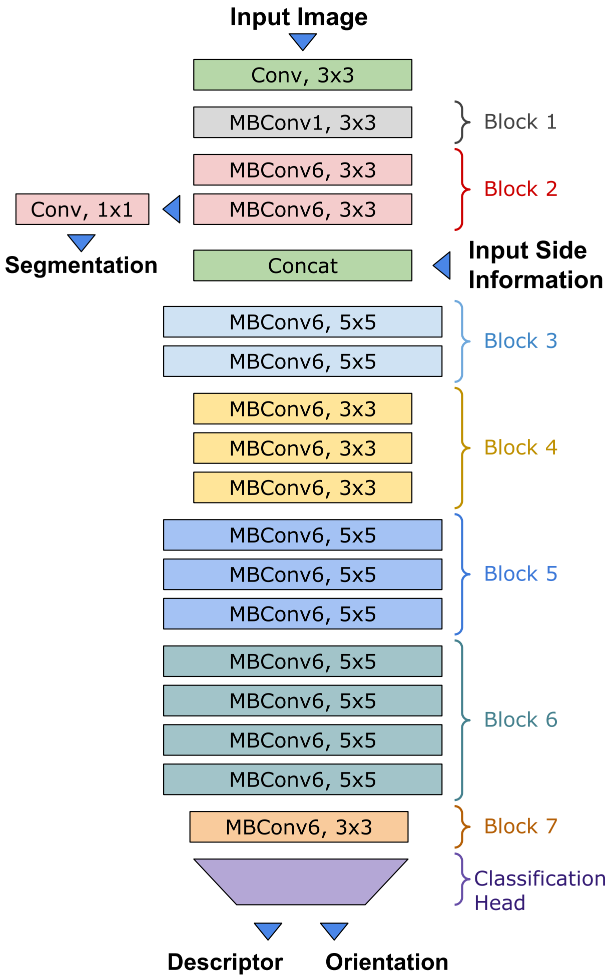
Upon the completion of training, we put the model into inference mode. The model is then fed with image crops and side information to generate predictions. The resulting output is a Python dictionary containing three predictions, one each from the segmentation, descriptor, and orientation branches. We have discovered that saving indices of input data and logit values is also beneficial.

3.1.1 Task Scheduling

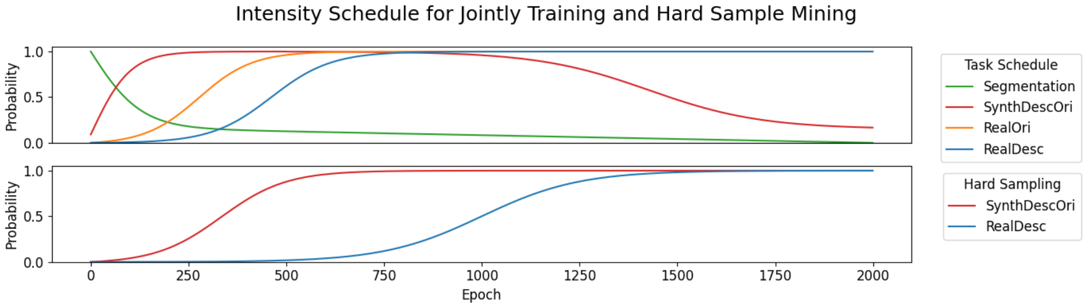
The training of multiple tasks jointly can be approached in numerous ways. These tasks could share datasets, inputs, and outputs and individually manage their resources. It is possible to train several neural networks, share all weights, or deploy a backbone with distinct branches. The loss function can be a weighted combination of several losses, or the tasks can alternatively perform updates based on their internal loss. Achieving a balance and optimizing the training process is challenging. Guided by past experiences, we adopted the method of switching between tasks. Careful scheduling must occur since each task can lead the network differently. As such, we designed a probabilistic trainer to facilitate task-switching by a prescribed schedule (fig: 3.3).

For each task described below, we created an identical task for the corresponding test dataset. We took advantage of the ability to switch between tasks and ran one evaluation step every five training steps. This allowed for more continuous evaluation compared to the more traditional approach where we evaluate all testing data at once every nth epoch. Moreover, we created two additional evaluation tasks when hard sample mining was employed (sec: 3.1.4): One task with a mining schedule identical to the parent task and one with random sampling. This method gave us valuable intel to distinguish between higher loss due to the mining difficulty and actual overfitting.

Figure 3.3 illustrates the training schedule for the first Segmentation task. As previously discussed, the segmentation task provides a robust stream of information for the model, enabling it to perform exceptionally well after only a few epochs. Given this characteristic, we have structured the schedule as follows: The model initially focuses on quickly learning various human shapes through the segmentation task, and then as more tasks are introduced, we maintain the segmentation at a lower percentage. This strategy serves as a form of regularization, preventing overfitting by not allowing the model to rely too heavily on any single feature. Similar reasoning is applied to the combined synthetic orientation and descriptor task (SynthDescOri). We prioritize early learning of accurate human representation and then retain the task at a lower intensity for the same regularization reasons. The hard sampling schedule is aligned closely with the task schedule. Waiting too long before implementing hard sampling could accelerate overfitting, whereas introducing it too early could lead to gradient explosion and a weaker foundational understanding. Consequently, we introduce the real orientation task early in the training process.



■ **Figure 3.2** A scheme of a modified version of the EfficientNet-B0 [100] with 5.3M trainable parameters and 0.39B FLOPS. Its main building block is mobile inverted bottleneck MBConv as described in the MobileNet paper [101]. Authors extended the MBConv layer with the squeeze-and-excitation optimization [102]. The model accepts Image Input of shape (B, 3, 64, 64), where B is the batch size. Then we extract the feature map early as Segmentation of shape (B, 1, 16, 16) with Sigmoid as an activation. The Side Information Input is a method to inject additional information into the network, including location coordinates, bounding box, camera height, and optical axis angle. This metadata vector is then encoded using a multi-layer perceptron to shape (B, 1, 16, 16) and concatenated with a feature map that leaves the second block. Then we follow the original routine. Ultimately, we branch the output into two streams: The Descriptor (B, 32) and the Orientation (B, 2). After long experimentation, we used Identity as an activation for the Descriptor and hyperbolic tangent for the Orientation.



■ **Figure 3.3** Intensity schedule for joint task training (upper) and hard sample mining (below). Rigorous experimentation with schedule tuning would be extremely costly. Therefore we manually set the schedules based on previous experience and results during training. Explanation described in Section 3.1.1

Since orientation is crucial for later stages, we keep its intensity at maximum until the end. Lastly, we initiate training on the real descriptor task. The network is already warmed up from the synthetic descriptor task and has learned real dataset visual patterns from the real orientation task. Throughout this phase, we maintain maximum intensity for both the task and the mining schedule. The curves depicted in the schedule represent parameterized sigmoids. We chose them because they change over a relatively short interval (as described in the curriculum learning survey [51]) while ensuring long tails that keep the schedule at either zero or one.

3.1.2 Semantic Segmentation

A deep neural network usually recognizes basic shapes in the first few layers and then builds up the complexity with depth. Semantic segmentation is a strong pixel-wise stream of information that quickly allows the model to recognize basic human shapes and textures. Past experiments revealed that the network tended to overfit the background in our data domain, which means that instead of concentrating on the designated person, it focused on comparing the similarity of the background. We hoped to accelerate the learning of diverse pedestrian shapes using semantic segmentation. In Blender, it is possible to render scenes with transparent backgrounds. The result was exported as a PNG with four channels (RGBA). The last alpha channel serves as transparency and corresponds to the mask applied for segmentation training.

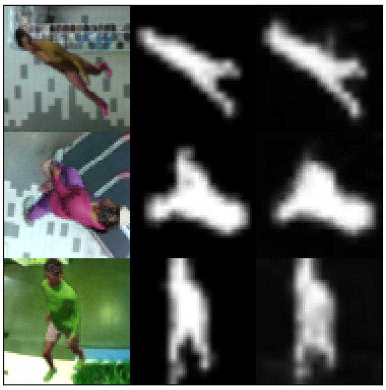


Figure 3.4 The first column is an input image of shape (3, 64, 64) - a synthetic person with a real background. The second column is a target segmentation mask of shape (1, 16, 16). The last column is a predicted segmentation mask of the corresponding size.

Using the mask, we can change the image's background. Initially, we deployed the OIV7 dataset [104], considering the broad variance in real-world scenarios. To personalize the task, we curated a dataset using our footage. The dataset is devoid of humans, reducing potential confusion during training. The segmentation branch incorporates an additional one-by-one convolution to integrate all layers, with Sigmoid as the activation function. The anticipated mask size is 16x16 pixels. We considered IoU loss and Binary Cross-Entropy loss. Experiments demonstrated the superior performance of the Binary Cross-Entropy loss.

$$\mathcal{L}_{bce} = -w\left[y \cdot \log x + (1-y) \cdot \log(1-x)\right] \tag{3.1}$$

Where y is the target segmentation mask, x is the predicted segmentation mask, and w is a manual rescaling weight parameter.

3.1.3 Orientation

We conducted experiments with category learning (front, back, left, right as per [55, 56]) and regression learning. Given that our cameras are ceiling-mounted, category learning presents many challenges. The process of quantizing continuous space often results in ambiguity for border values. As pedestrians can pass directly under the cameras, there is a requirement for an additional (top) category. This could lead to numerous issues, given that the top category borders all other specified categories. Annotators often label the same image differently, demonstrating the inherent difficulty in categorizing in this context, even for humans. Such ambiguity in annotations could lead to a weak and uncertain model. Figure 1.4 presents an example of an image crop that is ambiguous when classified using categorization, while it is straightforward with regression.

In the next iteration of the orientation learning, we formulated the task as regression learning (fig: 1.5). Continuous space better represents the data, and the annotation task (fig: 1.4) was slightly easier. The uncertainty of the prediction (and annotation) is then possible to express using the deviation from the ground truth. The synthetic dataset provides perfect ground truth and is a strong regularization for learning on the real orientation dataset.

The output of the orientation branch is a two-dimensional direction vector with values from the $< -1, 1 >$ range. As an activation, we used the hyperbolic tangent function. The loss function is cosine distance (eq: 1.4).

3.1.4 Identity Embedding

We extensively explored identity embedding learning in our previous work [49]. We compared various loss functions in that study, including triplet, quadruplet, contrastive, and cluster losses. However, considering the objective of this thesis, we prioritized convenience over performance. Consequently, we opted to employ the contrastive loss function (eq: 1.2) in our algorithm. Unlike other methods that sample triplets, quadruplets, or tuples, our approach uses sampling pairs for contrastive loss. This choice was made because contrastive learning is a form of validation loss. Therefore, it is better suited for the next stage, which is MTMCT.

When training models for descriptor learning, the Softmax function is frequently utilized as the activation function. However, we found it to be less effective in our setup. The Softmax function normalizes values increasing the relative ratio's importance. As a result, the descriptor may degenerate into a one-hot encoding akin to a classification problem. Assigning a class to each identity is suitable in a closed-world scenario. However, in our case, this led to suboptimal results. Consequently, we used the Identity function as the activation function, as it does not force the values to compete against each other. Alternatively, a sigmoid or hyperbolic tangent function could ensure a specific range of values. We have discussed combining the loss function of two tasks and training two branches with the same data. However, a challenge arises because the data for orientation learning is only available for a small subset of the real dataset. Therefore we must carefully balance the number of steps, as the orientation task would begin to overfit much sooner than the descriptor task. We can train the synthetic descriptor and orientation jointly, as we have ground truth for both tasks across the entirety of the synthetic dataset. This saved us time and computational power because we only needed to make a single pass through the backbone network instead of two.

Hard Sample Mining and Curriculum Learning

If we took two random people, distinguishing between them would be easy. They often wear clothes of different colors and vary in body shape or other features. The critical situation is when the two persons are visually very similar, making it difficult to decide whether they are the same person. Sampling a difficult batch in this manner is rather unlikely. A technique of hard sample mining is incorporated [47] that searches both for difficult positive and negative samples. It is important to schedule the hard sample mining and ramp up the difficulty of the training progressively. A rapid change in difficulty can cause gradient explosions, although gradient clipping significantly helps [50]. We implemented a hard sample mining schedule to provide the network with progressively harder samples (fig: 3.3). The sampler then picks a random/hard sample based on current probability.

Searching for hard pairs across the whole dataset is infeasible. Therefore we implemented a routine that first uses a forward pass on a structured batch of data and then selects difficult cases. Four variables (of arbitrary names) parametrize the process (B, I, J, K). A batch consists of B different identities. Each identity is a set of I crops, where $I/2$ crops originated in one camera and $I/2$ in the second camera. To each identity, we then attach J negative samples. However, they are negative for the whole batch, meaning that none of the B identities can occur in any negative group. This setup consequently creates a batch of shape $(B, I+J)$. The sampling process is then parametrized by K, the number of samples we make to reach identity. The loss is then calculated on K quadruplets, where a positive is chosen from $I/2$ crops, and negative pair is chosen from $B * K$ crops. At first glance, it is obvious that the potential for very difficult negative samples is large. The hard mining schedule in Figure 3.3 corresponds to the hard samples' proportion. This means that we can finetune the difficulty of each batch and use the potential of curriculum learning [51] without exploding the gradient.

We explained the influence of orientation on the visual resemblance of two images. If we ignored the orientation during the hard sample mining phase, we would probably force embeddings of two differently oriented images to be close even when they do not look the same. This results in suboptimal network performance. We utilized the orientation predictions during the sampling phase to address this issue. Targets with large enough differences can not be sampled. The probability of an image being sampled is proportional to the weighted combination of embedding distance and view angle distance. Using view angle for hard sample mining with contrastive learning offers many more combinations than traditional learning with triplets/quadruplets.

3.2 Multi-Camera Multi-Object Tracking

The MTMCT solvers can be very complicated. For simplicity and stability, we use solvers for minimum weighted matching. First, we define a correspondence matrix $\mathbf{H}$ of size $(2m+2n) \times (2m+2n)$ as follows:

$$\mathbf{H} = \left[\begin{array}{cc|cc} \mathbf{A}_{m\times m} & \mathbf{B}_{m\times n} & \mathbf{E}_{m\times m} & -\infty_{m\times n} \\ \mathbf{C}_{n\times m} & \mathbf{D}_{n\times n} & -\infty_{n\times m} & \mathbf{F}_{n\times n} \\ \hline \mathbf{G}_{m\times m} & -\infty_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} \\ -\infty_{n\times m} & \mathbf{K}_{n\times n} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} \end{array}\right] \tag{3.2}$$

Where the parts of the matrix are as follows:

A	targets leave and return back to Camera a
B	targets leave and return back to Camera b
C	targets leave Camera b and enter Camera a
D	targets leave and return back to Camera b
E	targets terminate in Camera a
F	targets terminate in Camera b
G	new targets initialized in Camera a
K	new targets initialized in Camera b

It is crucial to acknowledge that creating the full-sized matrix described above is unnecessary. We can limit ourselves to the upper part of the matrix, leaving $G_{m\times m}$, $K_{n\times n}$, and the rest out. This simplification calls for a slightly modified algorithm. Instead of the Hungarian Method (alg: 1), we employed a solver incorporated in the NetworkX [75] library. This solver, introduced by Karp in 1980 [105], solves an $m \times n$ assignment problem in $\mathcal{O}(mn \log n)$ steps.

Track Assignment Cost

The metrics for obtaining an assignment cost can be interpreted as a distance function between two tracks. We employed and experimented with various functions that fall into three broad categories.

- Global pairwise group: This method involves the computation of a pairwise distance matrix for each descriptor vector, followed by some form of reduction (e.g., Minimum, Mean, Min×Max, etc.). The most relevant candidate from this group was the minimum value from Min×Max, referred to as **MinMaxCost**.

- Viewpoint-based clustering: This method divides the track into n clusters based on the view angle, then selects a representative closest to the centroid from each cluster. This representation is memory-efficient and normalizes track lengths. To measure the resemblance of the two tracks, we select a suitable counterpart for each sample based on the view angle. Then a suitable reduction is applied. The reduction yielding the best results was the mean distance. In the follwing text: **MeanClusterMatchingCost**.

- Hybrid approach: This method combines the best aspects of the previous two groups. Each track is divided into clusters based on the view angle, with each cluster then reduced to its centroid. A distance matrix is built to compare two tracks, returning the minimum. We refer to this method as the **ClosestCentroidCost**.

Reducing the Number of Possible Links

We anticipated a strong correlation between the number of possible links and the algorithm's performance. A dense problem may contain a negative sample closer (in latent space) to the anchor than the positive sample (or the termination node). Therefore, the goal is to minimize the number of permitted links without excluding the positive track we seek. Since we consider our cameras non-overlapping, we rely solely on visual and temporal cues. Visual features help estimate the cost in the correspondence matrix, while temporal cues act as a filtering mechanism, restricting the number of potential links. (Outside this work, we also apply spatial constraints to reduce the difficulty further. Since we consider our cameras non-overlapping, we cannot use them.)

Tracks are matchable if and only if they occur within a reasonable time interval. We check the constraint efficiently by using an interval tree. Surveillance often involves re-identifying pedestrians over a long time horizon or multiple visits. However, our algorithm is geared towards statistical analysis to operate within a smaller time window. A practical rule of thumb we employed is in the range of minutes. If an individual leaves a camera's view for a period exceeding this threshold, we cannot match subsequent tracks to the individual's previous identity. Using a transition map between sensors and exits can further filter potential matches. Tracks from unconnected cameras then cannot be matched. Ultimately, the performance of the MTMCT algorithm largely depends on the number of possible links in the correspondence matrix.

MTMCT Benchmark

The benchmark we have created is designed to test the performance of the MTMCT algorithm in scenarios involving non-overlapping cameras. Our results demonstrate that the algorithm's success largely depends on its ability to compare the visual similarities between two image crops. The density of the correspondence matrix determines the complexity of the problem. A robust benchmark should simulate pedestrian traffic and adjust the difficulty level based on crowd density.

The benchmark process is straightforward: it first groups the test dataset by tracks and identities then generates a schedule for each track to occur while respecting spatiotemporal constraints. There are two crucial parameters to consider: the simulation and maximum transition lengths. The simulation length dictates the intervals in which the tracks are distributed. The maximum transition time restricts the duration an identity can spend off-camera. While simulating challenging situations could provide additional insights, it would likely defeat the purpose of early-stage testing. The distribution is uniform across the simulation period, and all tracks are included in the test process. Some might argue that excluding tracks tests the algorithm's ability to terminate rather than its ability to avoid incorrect matching. However, each identity sequence starts and ends, meaning that exactly two termination nodes are necessary for the correct assignment. This benchmark, while simple, serves as a powerful tool for constructing various matching problems with specified difficulty levels. It provides a flexible and effective means of evaluating and refining the performance of the MTMCT algorithm.

Chapter 4

Experiments and Results

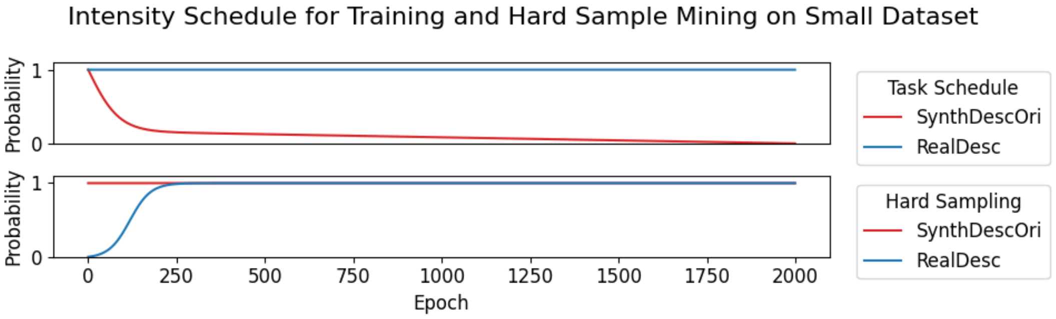
This chapter presents the experimental procedures conducted to validate the proposed methodologies. The experiments were performed on the test set of the **Krakov_2023_desc**, which contains real sequences of pedestrians passing in the Krakov shopping center. The test set consists of **513** unique identities, **1 048** tracks, and **38 151** image crops. The testing set covers real-world scenarios to ensure a broad spectrum of situations. The objective is not merely to demonstrate the effectiveness of our methodologies but to provide a concrete foundation for future research directions in this domain.

The machine for conducting our experiments and data processing is a powerful computing unit with Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz, 32 cores, and 93GB RAM. The machine is complemented by GPU by Nvidia GeForce RTX 3060 Ti, which provides exceptional acceleration for deep learning and computer vision tasks. The system operates on the Ubuntu 20.04 platform running as a docker container.

Selected Models

As a baseline, we trained the model using the entire Krakov_2023_desc dataset. However, this training process was excessively long, with one model requiring over a week for joint training on all datasets. Consequently, conducting dozens of experiments would be nearly unfeasible. To address this, we reduced the dataset to approximately 38,000 images, enabling us to load all the necessary data into working memory and expedite the training process to a matter of hours. We included a model **BaselineLong** that we trained on the full Krakov_2023_desc dataset. It is mainly for comparison; although we believe achieving even greater results would be possible, but that is not the purpose of this work. The **BaselineShort** model is an alternative to the large baseline. As it utilizes only real data, we applied the RealDesc schedule as shown in Figure 4.1. Both baseline models were trained without orientation, assuming all their related predictions to be zero. To evaluate the impact of orientation training, we trained the same model with orientation—denoted as **OriShort**. The third model **WarmupOriShort** on the reduced dataset is pre-trained using synthetic data in the first phase and then trained as the **OriShort** in the second phase. The final model, **OldBest**, is the best-performing model from our previous work [49], trained on a dataset of 436,043 image crops—over eleven times larger than the reduced dataset.

Notably, we trained one extra model, initially using synthetic data, then jointly applying the schedule in Figure 4.1. This model did not perform well on the testing set, presumably due to the synthetic data interfering with the necessary intensification for peak performance during training. However, this effect was not apparent during joint training on the large dataset.



■ **Figure 4.1** The above image illustrates the intensity schedule for task training (top) and hard sample mining (bottom) on the Krakov_2023_desc_small dataset (38k images). Models BaselineShort, OriShort, and WarmupShort were trained using the RealDesc schedule, excluding the secondary SynthDescOri schedule. However, the red curve represents the intention of joint training on the small dataset. Regrettably, we could not train a high-performing model that could compete with the models we tested.

Model	Precision	Recall	ACC	F1
BaselineLong	0.920138	0.894080	0.908240	0.906922
WarmupOriShort	0.784244	0.829040	0.800480	0.806020
OriShort	0.768608	0.805440	0.781480	0.786593
BaselineShort	0.764932	0.797120	0.776080	0.780694
OldBest	0.666667	0.673920	0.668480	0.670274

■ **Table 4.1** Comparison of all models in the verification test. All models trained using the new methodology demonstrate superior performance compared to OldBest. The table indicates that the baseline can be improved by employing orientation. The synthetic pre-training improves the score further. As anticipated, BaselineLong significantly outperforms the other models.

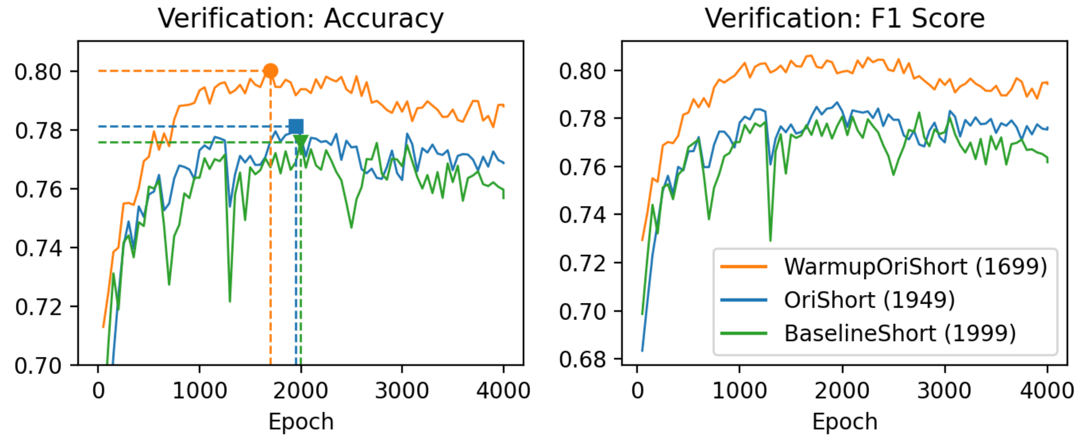
4.1 Identity Verification

Identity verification is a problem where the objective is to decide whether two images feature the same individual. We utilized this task to select the best weights for each model discussed previously (see Figure 4.2).

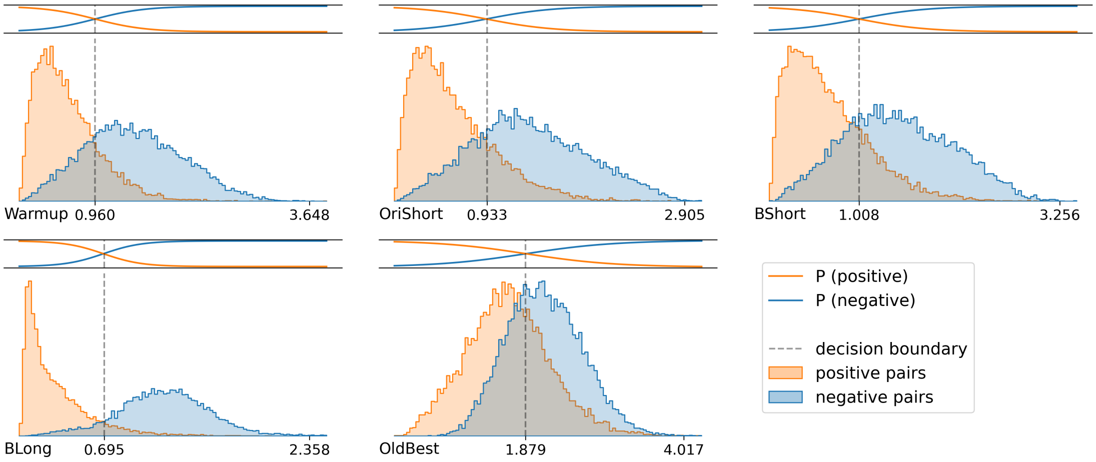
The contrastive loss is parameterized with a margin value corresponding to the optimal decision boundary. We began by measuring the distances between numerous image pairs. Then we sorted the measurements into two classes. The first class contains distances of positive pairs, while the second comprises distances of negative pairs. Finally, we employed logistic regression to estimate the decision boundary based on the measured distances (Figure 4.3). Results indicate that the better the model gets, the smaller the threshold is. This is presumably because the intra-class distances are reduced more than the inter-class distances are expanded. Distinguishing between two visually similar identities is exceptionally challenging when working with crops of size (64, 64) pixels. Given enough samples, a negative pair with a very small distance occurs relatively frequently. Despite this high difficulty level, BaselineLong can reasonably distinguish between positive and negative pairs.

4.2 Ranking and Retrieval

Mean Average Precision at K ($mAP@K$) is a widely adopted evaluation metric in recommender systems and other rank-based tasks. This metric considers an algorithm's precision and recall,



■ **Figure 4.2** Selection of the best model based on Accuracy on the validation dataset. Even tho we trained models for 4000 epochs, performance peaks around the 2000th epoch (peak epoch is specified in the legend). The optimal epoch for WarmupOriShort is 1700, proving that the training time is about 15% faster.



■ **Figure 4.3** The graph represents the density of positive and negative pairs by distance. The upper bar indicates the probabilities of both classes, while the lower image is a distribution plot. Each model has a unique threshold obtained through logistic regression. The x-axis displays the threshold value, with a second value near the end to provide a sense of scale. The label ”Warmup” represents the WarmupOriShort model, while the prefix ”B” denotes Baseline. The intersection of the distributions reflects the descriptor’s effectiveness in separating positive and negative sample pairs.

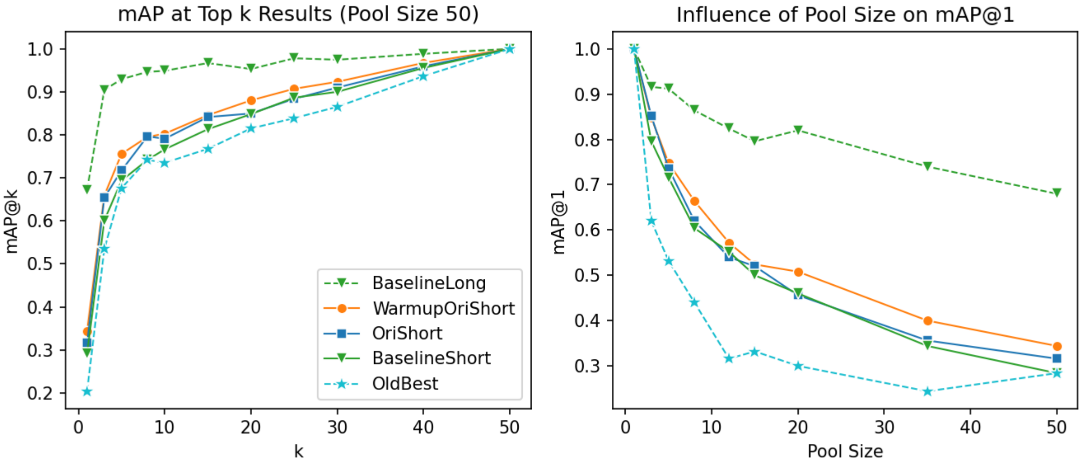


Figure 4.4 Comparison of models for various $mAP@k$ values. Increasing k reduces the effects of error, while expanding the pool size intensifies the difficulty. As anticipated, BaselineLong exhibits the best performance on both graphs. Models trained on the reduced dataset performed as expected. The top performer was the model incorporating synthetic warmup and orientation. The second and third models, trained exclusively on the real subset, demonstrated that OriShort outperforms BaselineShort. OldBest finished last. Interestingly, the OldBest model considerably underperforms at $mAP@1$, while the difference is not as pronounced for other k values.

providing a comprehensive assessment. For a better understanding, it is important to define two fundamental measures. Precision at k (P@k) is the ratio of relevant results within the top k retrieved documents to the number of retrieved documents. The area under the Precision-Recall curve signifies average precision at k ($AP@K$).

$$\begin{aligned} \text{AP@}K &= \frac{1}{r}\sum_{k=1}^{K} \text{P@}k \cdot \text{rel}(k) \\ \text{rel}(k) &= \begin{cases} 1, & \text{if item at } k_{th} \text{ rank is relevant} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4.1}$$

Where r is the total number of relevant documents.

Finally, the mean value of $AP@K$ of all queries ($mAP@K$) is calculated, combining an algorithm's precision and recall into a single robust value.

In this section, the results consider viewpoint predictions during the pool creation. This enables viewpoint-based ranking, giving an advantage to models trained with orientation. Models trained without do not influence the process in any way. As the results suggest, the integration of orientation can enhance the outcomes. Figure 4.4 depicts the ranking results for various $mAP@k$ values. The pool size significantly influences the performance, with a larger gallery size increases the chance of a negative sample infiltrating the top k results. It is important to note that the slope is very steep initially. In the context of MTMCT, matching results can be very accurate with a limited number of possible connections. However, as evidenced by BaselineLong, the steepness seems to diminish with sufficient data.

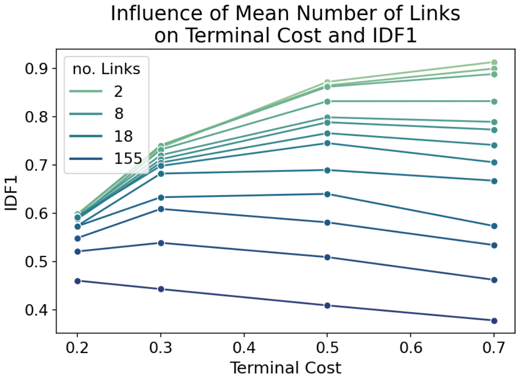
4.3 MTMCT Benchmark

Model	mAP@1	mAP@3	mAP@5	mAP@8	mAP@10	mAP@25
BaselineLong	0.67200	0.90433	0.92952	0.94639	0.94891	0.97803
WarmupOriShort	0.34400	0.65900	0.75586	0.79530	0.80268	0.90707
OriShort	0.31600	0.65400	0.71742	0.79661	0.79089	0.88380
BaselineShort	0.29200	0.60100	0.69390	0.74172	0.76643	0.88634
OldBest	0.20400	0.53467	0.67633	0.74208	0.73496	0.83877

■ **Table 4.2** The ranking results depicted in the table further validate the superiority of synthetic datasets and enhancements brought about by orientation learning. Notably, the OldBest model exhibits a surprisingly close performance compared to the other models, especially considering the verification results.

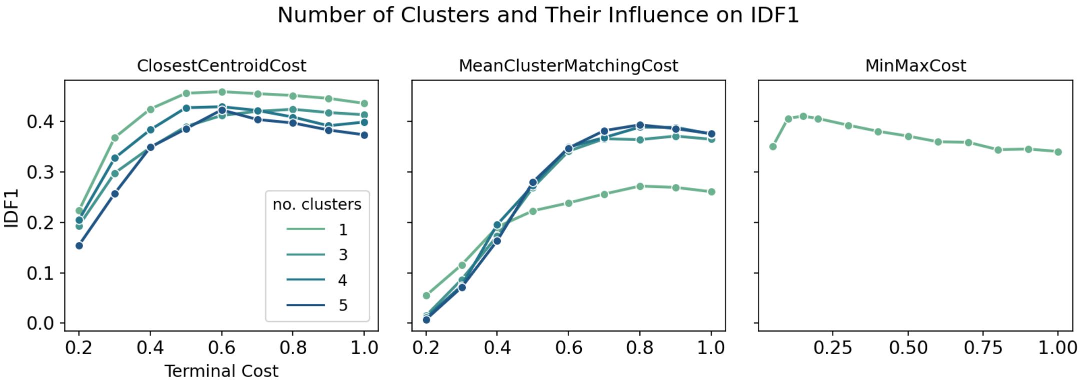
Our chosen metric, IDF1, described in Section 1.2.4, is well-accepted in academia. We applied the functions outlined in Section 3.2 to estimate the best assignment cost. We also experimented with several other functions and their variations but could not yield satisfactory results. This aspect of our work still holds potential for further exploration. Modifying the training process to favor a given cost function could significantly enhance results. Regrettably, we could not conduct these experiments within a reasonable time frame due to their resource-intensive nature.

One surprising insight we gleaned was the correlation between the correspondence matrix's density and the optimal termination value (fig: 4.5). The denser the matrix, the larger the termination cost should be. Decreasing the number of connections is equal to reducing the gallery size during ranking (fig: 4.4). Fewer connections mean we can have more confidence in the accuracy of a match. The termination cost is in some way tied to confidence. In a larger pool, our confidence that the closest pair shares an identity decreases, and the nature of the task encourages more frequent terminations. This effect then reflects in other experiments. We believe that in later stages, this parameter can be learned as part of the model. For now, we tune the termination cost for each model, cost function, and matrix density.



■ **Figure 4.5** The mean number of links is equivalent to the density of the correspondence matrix. A denser matrix necessitates a smaller optimal termination cost. This is because the termination cost is associated with confidence levels. When faced with a larger selection pool, our confidence in the closest pair sharing an identity with the anchor decreases. The intrinsic nature of the MTMCT makes more frequent termination desirable. We obtained the data with the BaselineLong model and ClosestCentroidCost.

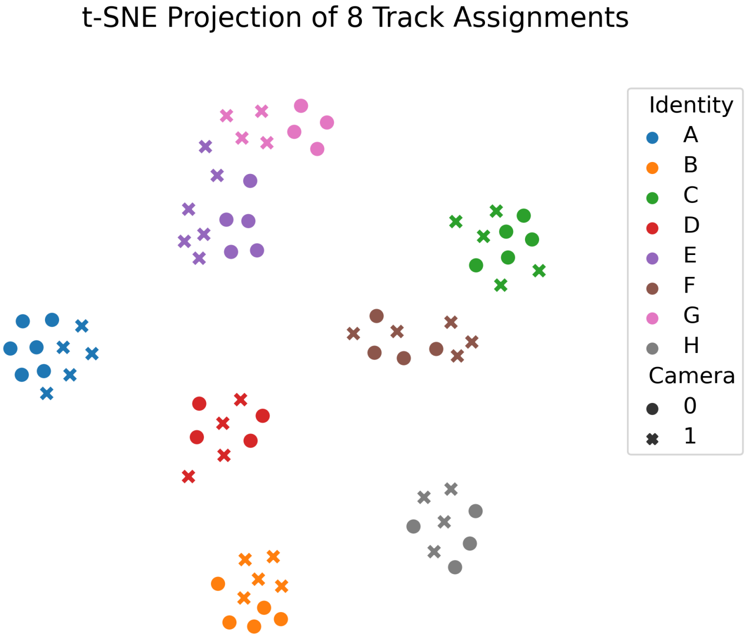
In order to measure the difference between the tracks, we employed the cost functions discussed in Section 3.2. The initial idea was that clustering tracks based on the view angle could potentially improve the assignment. We aimed to design a cost function that reduces the negative impact of outliers and compares only the relevant parts of tracks. Unfortunately, we were unable to validate this hypothesis. Figure 4.6 demonstrates the advantage of using a single centroid to represent tracks. This pattern was consistently observed across all models. For the final results presented in Table 4.3, we used the ClosestCentroidCost with a single cluster for all models. It is important to note that the same cost function also yielded the best results in our previous studies, further confirming the research on the ”Unreasonable Effectiveness of Centroids in Image Retrieval” [61].



■ **Figure 4.6** The expectation was higher for multiple clusters due to their potential to avoid outliers. However, we could not confirm this hypothesis, as the best score consistently peaked with only one cluster, regardless of the training method or terminal cost. Despite this, we can't rule out the possible utility of multiple clusters. With a more advanced hard sample mining approach, we could potentially train a model to identify outliers and thus support our original hypothesis. A positive observation is a stable performance across a wide range of terminal costs. The results were obtained using the WarmupOriShort model, with the correspondence matrix averaging 20 connections.

Model	IDF1	IDP	IDR	IDTP	IDFP	IDTN	IDFN
BaselineLong	0.68503	0.59113	0.81441	373	258	332	85
WarmupOriShort	0.42029	0.33388	0.56704	203	405	285	155
OriShort	0.40396	0.29143	0.65807	204	496	242	106
BaselineShort	0.38680	0.26641	0.70569	211	581	168	88
OldBest	0.22222	0.12853	0.82000	123	834	64	27

■ **Table 4.3** The final evaluation of all models. As expected, the most significant performance difference is due to the size and quality of the training set. Both tested methods (synthetic warmup and orientation learning) consistently outperform the baseline. There is a significant performance gap compared with the OldBest model, even more so than in previous results. In this test, the old model was placed in a scenario that did not favor its intended purpose.



■ **Figure 4.7** t-SNE projection of 160 identity embeddings. Each color represents a different identity. An identity connects multiple tracks (two in this scenario). We like to observe homogeneous clusters of the same identity emerge. Since we reduce a 32-dimensional vector onto a flat surface, we can not be certain if tracks E and G are intertwined or if this is the product of the projection.

It is important to address why the old descriptor underperforms so significantly in Table 4.3. The old dataset was large compared to the current short dataset, consisting of nearly half a million samples. At the time, conditions made it nearly impossible to ensure the high quality and purity of the data. Therefore, we developed an annotation tool to exclude noisy samples efficiently. Hard sample mining elevates this problem further, putting the old model at a big disadvantage. The most effective approach to inter-camera tracking involved heavy use of camera overlaps, with visual cost being a much weaker factor in the global assignment problem. In this thesis, we wanted to focus extensively on training a strong visual feature extractor, and therefore, we assumed the camera views to be non-overlapping. Training with an impact on identity verification enabled better track termination. The metric used in the previous work was weaker, meaning that not terminating tracks were not as heavily penalized. Considering the conditions at the time, the old model performed well in all tasks except the last one.

Chapter 5
Discussion

Our experiments suggest enriching training with extra tasks enhances the system's performance. Incorporating an orientation task into training positively impacted our baseline model. Further, using a synthetic dataset for weight initialization boosted the model's effectiveness. Synthetic datasets, being cost-effective supplements to real data, save training time due to the early identification of human shapes, which is essential with smaller datasets. Integrating synthetic data can increase the model's generalization ability, even with plenty of real data. Joint training following this prevents overfitting and strengthens regularization.

This project is key to iC Systems' goal of developing a privacy-preserving, pedestrian behavior analysis solution in large structures. Until now, our solution relied on isolated cameras for pedestrian tracking within their field of view only. The ability to link tracks across multiple cameras elevates our product to a new level of sophistication. Experience suggests that augmenting previous geometry-focused solutions with our current focus on visual feature extraction can enhance performance further.

Two findings from our study were particularly unexpected yet crucial. Firstly, the correlation between the correspondence matrix's density and termination cost. This implies that for real-world applications, it would be necessary to estimate the optimal termination cost from the matrix density and characteristics of the model. Secondly, we were surprised that centroid distances were the most effective cost function despite our efforts to accommodate training to different cost functions.

Limitations of presented results are tied to the size of the used dataset. With larger data, it is unknown if synthetic data improves performance. The training process can be yet improved significantly to suit specific cost functions for the assignment problem.

In future work, we plan to explore existing end-to-end multi-camera trackers [83, 71]. However, a custom solution tailored to our needs may be more effective than adopting a larger, generalized system. A future research direction could involve connecting the assignment algorithm and deep learning techniques. Additionally, adopting more advanced methods for training the descriptor [61] could be promising.

The creation of multiple large datasets for training various deep learning tasks, and the development of a highly customizable synthetic data generator, opens numerous opportunities. The utility of synthetic data in our field was initially uncertain, given the challenging conditions such as low resolution and changing illumination. Fortunately, we simulated these conditions and constructed a solution that could benefit others in the future.

Chapter 6
Conclusion

Our research presents four large datasets, each having a unique purpose. The Krakov_2023_desc dataset comprises **291,602** images and is designed for identity embedding. Using overlapping camera views and precise geometry, we generated smart suggestions that were labeled using our annotation tool. The second dataset, Krakov_2023_ori, includes **12,753** image crops and is intended for orientation learning. Initial suggestions were obtained using the Kalman filter, and annotations were made similarly to the previous dataset. The third dataset, Synth, contains **234,034** crops and employs 3D graphics to simulate human behavior and movement patterns in an environment that closely mirrors real-world scenarios. We navigated a challenging engineering process that automated the entire generation pipeline—from scene setting, trajectory planning, and animation to rendering and refining the final product. Finally, the footfall_background dataset consists of **20,291** images without pedestrians. This dataset serves as an extension for the synthetic dataset and allows for changing the background to real scene images. This feature significantly improves generalization when training on synthetic data.

As outlined in Chapter 6, the combined count of relevant lines of code is approximate **17,000** lines + estimating 5,000 more of experimental code saved in jupyter notebooks.

We developed a probabilistic multi-task trainer to incorporate semantic segmentation, descriptor, and orientation tasks for both real and synthetic data. Through experimentation, we utilized task scheduling and curriculum learning to gradually increase the training process's difficulty level. We successfully transferred the knowledge acquired from the synthetic domain to the real one, with the model pre-trained on synthetic data demonstrating a **15%** shorter training time while consistently outperforming other models.

Despite the challenges posed by extremely low resolution, illumination changes, lens distortion, and a low frame rate, we successfully trained a robust model. The BaselineLong model significantly outperformed the previous best model, largely due to the higher quality of available data. A remarkable observation was that models trained on only **38 thousand** samples outperformed the prior best model trained with nearly **half a million** samples. This improvement was primarily achieved due to a cleaner dataset and the application of advanced methodologies.

Furthermore, our study identified and addressed weaknesses in the assignment algorithm. By training identity verification, we discovered a method that allows tracks to match and terminate on time. This new approach, when combined with our previous solution that heavily relied on known geometry, will hopefully result in a robust system suitable for deployment.

Looking forward, our goal is to implement our solution in real-world settings, further enhancing the robustness and consistency of the methods presented. A future research direction is exploring and establishing a connection between the assignment algorithm and the training process. This would transform our pipeline into an end-to-end system, tying all components together into a larger, interconnected infrastructure.

Chapter 7
Acronyms

CMC Cumulative Matching Characteristics.

CNN Convolutional neural network.

Dt Delaunay triangulation.

HOG Histogram of Oriented Gradients.

iC iC Systems.ai, s.r.o..

IDP Identification Precision.

IDR Identification Recall.

IoU Intersection over Union.

LSTM Long short-term memory.

mAP mean Average Precision.

MOT Multiple-object tracking.

MTL multi-Task Learning.

MTMCT Multi-Target Multi-Camera Tracking.

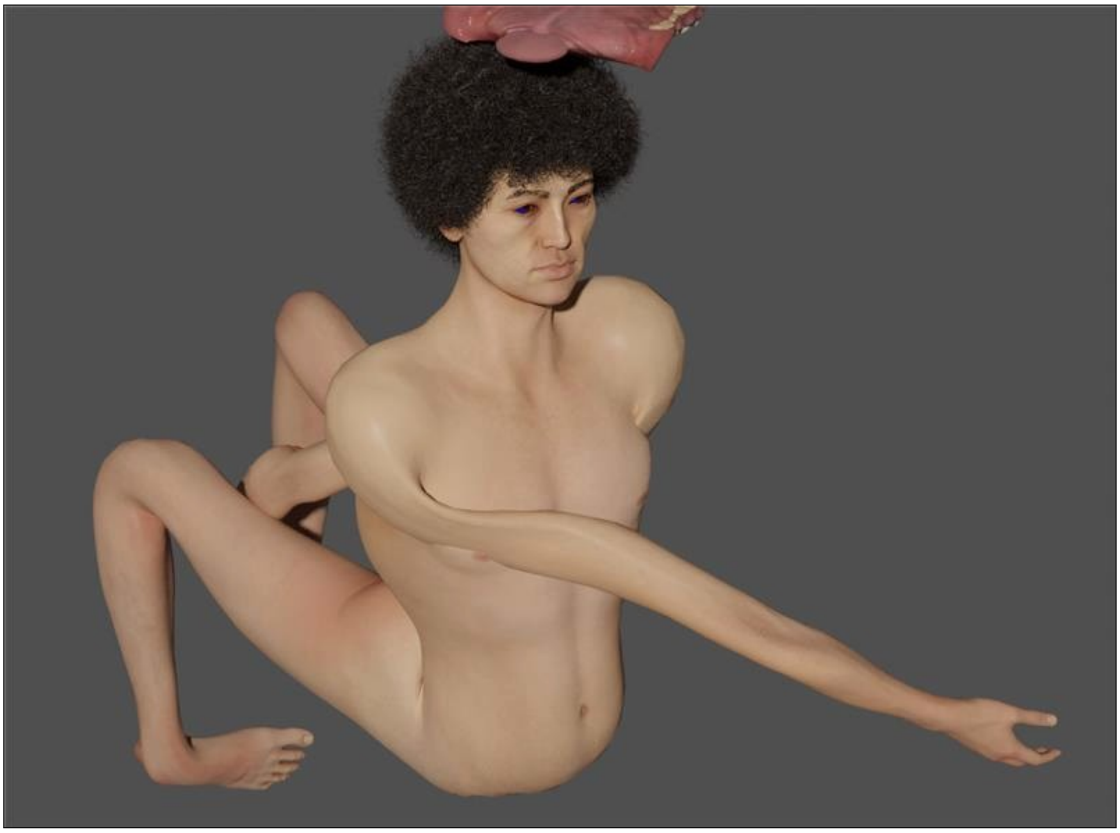
NMS Non-Maxima Suppression.

PnP Perspective-n-Point.

Re-ID Person Re-Identification.

RPNs Region Proposal Networks.

SVM Support vector machine.



■ **Figure 1** The figure shows a humorous example from the early development stage, where we intended to set a random (human) pose. To this date, this result remains a mystery. Notice the unfortunate position of the upper palate.

Bibliography

1. ZHU, Xizhou; SU, Weijie; LU, Lewei; LI, Bin; WANG, Xiaogang; DAI, Jifeng. *Deformable DETR: Deformable Transformers for End-to-End Object Detection.* 2021. Available from DOI: 10.48550/arXiv.2010.04159.
2. GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Los Alamitos, CA, USA: IEEE Computer Society, 2014, pp. 580–587. ISSN 1063-6919. Available from DOI: 10.1109/CVPR.2014.81.
3. LUO, Wenhan; XING, Junliang; MILAN, Anton; ZHANG, Xiaoqin; LIU, Wei; KIM, Tae-Kyun. Multiple object tracking: A literature review. *Artificial Intelligence.* 2021, vol. 293, p. 103448. ISSN 0004-3702. Available from DOI: 10.1016/j.artint.2020.103448.
4. ZOU, Zhengxia; CHEN, Keyan; SHI, Zhenwei; GUO, Yuhong; YE, Jieping. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE.* 2023, vol. 111, no. 3, pp. 257–276. Available from DOI: 10.1109/JPROC.2023.3238524.
5. KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM.* 2017, vol. 60, no. 6, pp. 84–90. ISSN 0001-0782. Available from DOI: 10.1145/3065386.
6. YE, Mang; SHEN, Jianbing; LIN, Gaojie; XIANG, Tao; SHAO, Ling; HOI, Steven C. H. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2022, vol. 44, no. 6, pp. 2872–2893. Available from DOI: 10.1109/TPAMI.2021.3054775.
7. ZHAO, Zhong-Qiu; ZHENG, Peng; XU, Shou-Tao; WU, Xindong. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems.* 2019, vol. 30, no. 11, pp. 3212–3232. Available from DOI: 10.1109/TNNLS.2018.2876865.
8. GIRSHICK, Ross. *Fast R-CNN.* 2015. Available from DOI: 10.48550/arXiv.1504.08083.
9. REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2017, vol. 39, no. 6, pp. 1137–1149. Available from DOI: 10.1109/TPAMI.2016.2577031.
10. REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 779–788. Available from DOI: 10.1109/CVPR.2016.91.
11. WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.* 2022. Available from DOI: 10.48550/arXiv.2207.02696.

12. YURTSEVER, Ekim; LAMBERT, Jacob; CARBALLO, Alexander; TAKEDA, Kazuya. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*. 2020, vol. 8, pp. 58443–58469. Available from DOI: 10.1109/ACCESS.2020.2983149.

13. DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005, vol. 1, 886–893 vol. 1. Available from DOI: 10.1109/CVPR.2005.177.

14. BIANCHINI, Monica; SCARSELLI, Franco. On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. *IEEE Transactions on Neural Networks and Learning Systems*. 2014, vol. 25, no. 8, pp. 1553–1565. Available from DOI: 10.1109/TNNLS.2013.2293637.

15. HUANG, Jonathan; RATHOD, Vivek; SUN, Chen; ZHU, Menglong; KORATTIKARA, Anoop; FATHI, Alireza; FISCHER, Ian; WOJNA, Zbigniew; SONG, Yang; GUADARRAMA, Sergio; MURPHY, Kevin. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. Available from DOI: 10.48550/arXiv.1611.10012.

16. CARION, Nicolas; MASSA, Francisco; SYNNAEVE, Gabriel; USUNIER, Nicolas; KIRILLOV, Alexander; ZAGORUYKO, Sergey. End-to-end object detection with transformers. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229. Available from DOI: 10.1007/978-3-030-58452-8_13.

17. UIJLINGS, Jasper RR; VAN DE SANDE, Koen EA; GEVERS, Theo; SMEULDERS, Arnold WM. Selective search for object recognition. *International journal of computer vision*. 2013, vol. 104, pp. 154–171. Available from DOI: 10.1007/s11263-013-0620-5.

18. LECUN, Yann; BENGIO, Yoshua, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*. 1995, vol. 3361, no. 10, p. 1995. Available also from: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e26cc4a1c717653f323715d751c8dea7461aa105.

19. ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. Available from DOI: 10.1109/ICEngTechnol.2017.8308186.

20. LIENHART, R.; MAYDT, J. An extended set of Haar-like features for rapid object detection. In: *Proceedings. International Conference on Image Processing*. 2002, vol. 1, pp. I–I. Available from DOI: 10.1109/ICIP.2002.1038171.

21. VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, vol. 1, pp. I–I. Available from DOI: 10.1109/CVPR.2001.990517.

22. CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. *Machine learning*. 1995, vol. 20, pp. 273–297. Available from DOI: 10.1007/BF00994018.

23. WEIJER, Joost van de; SCHMID, Cordelia; VERBEEK, Jakob; LARLUS, Diane. Learning Color Names for Real-World Applications. *IEEE Transactions on Image Processing*. 2009, vol. 18, no. 7, pp. 1512–1523. Available from DOI: 10.1109/TIP.2009.2019809.

24. LIAO, S.; HU, Y.; ZHU, Xiangyu; LI, S. Z. Person re-identification by Local Maximal Occurrence representation and metric learning. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2197–2206. ISSN 1063-6919. Available from DOI: 10.1109/CVPR.2015.7298832.

25. LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005, vol. 27, no. 8, pp. 1265–1278. Available from DOI: 10.1109/TPAMI.2005.151.

26. OJALA, Timo; PIETIKÄINEN, Matti; HARWOOD, David. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*. 1996, vol. 29, no. 1, pp. 51–59. ISSN 0031-3203. Available from DOI: 10.1016/0031-3203(95)00067-4.

27. RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. ORB: An efficient alternative to SIFT or SURF. In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. Available from DOI: 10.1109/ICCV.2011.6126544.

28. LOWE, David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004, vol. 60, pp. 91–110. Available from DOI: 10.1023/B:VISI.0000029664.99615.94.

29. ARANDJELOVIĆ, Relja; ZISSERMAN, Andrew. Three things everyone should know to improve object retrieval. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2911–2918. Available from DOI: 10.1109/CVPR.2012.6248018.

30. JÜNGLING, Kai; BODENSTEINER, Christoph; ARENS, Michael. Person re-identification in multi-camera networks. In: *CVPR 2011 WORKSHOPS*. 2011, pp. 55–61. Available from DOI: 10.1109/CVPRW.2011.5981771.

31. KOCH, Gregory; ZEMEL, Richard; SALAKHUTDINOV, Ruslan, et al. Siamese neural networks for one-shot image recognition. In: *ICML deep learning workshop*. Lille, 2015, vol. 2. No. 1. Available also from: www.cs.toronto.edu/~gkoch/files/msc-thesis.pdf.

32. WANG, Yu-Hsiang; HSIEH, Jun-Wei; CHEN, Ping-Yang; CHANG, Ming-Ching. *SMILEtrack: SiMIlarity LEarning for Multiple Object Tracking*. 2022. Available from DOI: 10.48550/arXiv.2211.08824.

33. KUHN, Harold W. The Hungarian method for the assignment problem. *Naval research logistics quarterly*. 1955, vol. 2, no. 1-2, pp. 83–97. Available from DOI: 10.1002/nav.3800020109.

34. ANDRIYENKO, Anton; SCHINDLER, Konrad. Multi-target tracking by continuous energy minimization. In: *CVPR 2011*. 2011, pp. 1265–1272. Available from DOI: 10.1109/CVPR.2011.5995311.

35. ANDRIYENKO, Anton; SCHINDLER, Konrad; ROTH, Stefan. Discrete-continuous optimization for multi-target tracking. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1926–1933. Available from DOI: 10.1109/CVPR.2012.6247893.

36. GALOR, Amit; ORFAIG, Roy; BOBROVSKY, Ben-Zion. *Strong-TransCenter: Improved Multi-Object Tracking based on Transformers with Dense Representations*. 2022. Available from DOI: 10.48550/arXiv.2210.13570.

37. ZENG, Fangao; DONG, Bin; ZHANG, Yuang; WANG, Tiancai; ZHANG, Xiangyu; WEI, Yichen. MOTR: End-to-End Multiple-Object Tracking with Transformer. In: AVIDAN, Shai; BROSTOW, Gabriel; CISSÉ, Moustapha; FARINELLA, Giovanni Maria; HASSNER, Tal (eds.). *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 659–675. ISBN 978-3-031-19812-0.

38. BOCHINSKI, Erik; EISELEIN, Volker; SIKORA, Thomas. High-Speed tracking-by-detection without using image information. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2017, pp. 1–6. Available from DOI: 10.1109/AVSS.2017.8078516.

39. BOCHINSKI, Erik; SENST, Tobias; SIKORA, Thomas. Extending IOU Based Multi-Object Tracking by Visual Information. In: *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2018, pp. 1–6. Available from DOI: 10.1109/AVSS.2018.8639144.

40. RISTANI, Ergys; SOLERA, Francesco; ZOU, Roger; CUCCHIARA, Rita; TOMASI, Carlo. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In: HUA, Gang; JÉGOU, Hervé (eds.). *Computer Vision – ECCV 2016 Workshops*. Cham: Springer International Publishing, 2016, pp. 17–35. ISBN 978-3-319-48881-3.

41. RISTANI, Ergys; TOMASI, Carlo. *Features for Multi-Target Multi-Camera Tracking and Re-Identification*. 2018. Available from DOI: 10.48550/arXiv.1803.10859.

42. ZHONG, Zhun; ZHENG, Liang; CAO, Donglin; LI, Shaozi. *Re-ranking Person Re-identification with k-reciprocal Encoding*. 2017. Available from DOI: 10.48550/arXiv.1701.08398.

43. SUN, Yifan; ZHENG, Liang; YANG, Yi; TIAN, Qi; WANG, Shengjin. *Beyond Part Models: Person Retrieval with Refined Part Pooling (and a Strong Convolutional Baseline)*. 2018. Available from arXiv: 1711.09349 [cs.CV].

44. ZHENG, Liang; ZHANG, Hengheng; SUN, Shaoyan; CHANDRAKER, Manmohan; YANG, Yi; TIAN, Qi. *Person Re-identification in the Wild*. 2017. Available from DOI: 10.48550/arXiv.1604.02531.

45. VARIOR, Rahul Rama; SHUAI, Bing; LU, Jiwen; XU, Dong; WANG, Gang. *A Siamese Long Short-Term Memory Architecture for Human Re-Identification*. 2016. Available from DOI: 10.48550/arXiv.1607.08381.

46. HADSELL, R.; CHOPRA, S.; LECUN, Y. Dimensionality Reduction by Learning an Invariant Mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 2006, vol. 2, pp. 1735–1742. Available from DOI: 10.1109/CVPR.2006.100.

47. SCHROFF, Florian; KALENICHENKO, Dmitry; PHILBIN, James. FaceNet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. Available from DOI: 10.1109/CVPR.2015.7298682.

48. CHEN, Weihua; CHEN, Xiaotang; ZHANG, Jianguo; HUANG, Kaiqi. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1320–1329. Available from DOI: 10.1109/CVPR.2017.145.

49. HULMÁK, Erik; NAISER, Filip. *Re-identifikace osob v systému kamer*. 2021. Available also from: http://hdl.handle.net/10467/92890. B.S. thesis. České vysoké učení technické v Praze. Vypočetní a informační centrum.

50. PASCANU, Razvan; MIKOLOV, Tomas; BENGIO, Yoshua. On the difficulty of training recurrent neural networks. In: DASGUPTA, Sanjoy; MCALLESTER, David (eds.). *Proceedings of the 30th International Conference on Machine Learning*. Atlanta, Georgia, USA: PMLR, 2013, vol. 28, pp. 1310–1318. Proceedings of Machine Learning Research, no. 3. Available also from: https://proceedings.mlr.press/v28/pascanu13.html.

51. WANG, Xin; CHEN, Yudong; ZHU, Wenwu. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022, vol. 44, no. 9, pp. 4555–4576. Available from DOI: 10.1109/TPAMI.2021.3069908.

52. FENG, Zhanxiang; LAI, Jianhuang; XIE, Xiaohua. Learning View-Specific Deep Networks for Person Re-Identification. *IEEE Transactions on Image Processing*. 2018, vol. 27, no. 7, pp. 3472–3483. Available from DOI: 10.1109/TIP.2018.2818438.

53. ZHENG, Wei-Shi; GONG, Shaogang; XIANG, Tao, et al. Person re-identification by support vector ranking. In: [n.d.]. Available also from: http://www.eecs.qmul.ac.uk/~sgg/papers/ProsserEtAl_BMVC2010.pdf.

54. YU, Hong-Xing; WU, Ancong; ZHENG, Wei-Shi. Cross-View Asymmetric Metric Learning for Unsupervised Person Re-Identification. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 994–1002. Available from DOI: 10.1109/ICCV.2017.113.

55. LIU, Fangyi; ZHANG, Lei. View Confusion Feature Learning for Person Re-Identification. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6638–6647. Available from DOI: 10.1109/ICCV.2019.00674.

56. LI, Dangwei; ZHANG, Zhang; CHEN, Xiaotang; LING, Haibin; HUANG, Kaiqi. *A Richly Annotated Dataset for Pedestrian Attribute Recognition*. 2016. Available from DOI: 10.48550/arXiv.1603.07054.

57. ZHANG, Yu; YANG, Qiang. A Survey on Multi-Task Learning. *IEEE Transactions on Knowledge and Data Engineering*. 2022, vol. 34, no. 12, pp. 5586–5609. Available from DOI: 10.1109/TKDE.2021.3070203.

58. ZHUANG, Fuzhen; QI, Zhiyuan; DUAN, Keyu; XI, Dongbo; ZHU, Yongchun; ZHU, Hengshu; XIONG, Hui; HE, Qing. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*. 2021, vol. 109, no. 1, pp. 43–76. Available from DOI: 10.1109/JPROC.2020.3004555.

59. HE, Kaiming; GKIOXARI, Georgia; DOLLÁR, Piotr; GIRSHICK, Ross. *Mask R-CNN*. 2018. Available from DOI: 10.48550/arXiv.1703.06870.

60. CHEN, Kai; PANG, Jiangmiao; WANG, Jiaqi; XIONG, Yu; LI, Xiaoxiao; SUN, Shuyang; FENG, Wansen; LIU, Ziwei; SHI, Jianping; OUYANG, Wanli; LOY, Chen Change; LIN, Dahua. Hybrid Task Cascade for Instance Segmentation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4969–4978. Available from DOI: 10.1109/CVPR.2019.00511.

61. WIECZOREK, Mikolaj; RYCHALSKA, Barbara; DABROWSKI, Jacek. On the Unreasonable Effectiveness of Centroids in Image Retrieval. In: MANTORO, Teddy; LEE, Minho; AYU, Media Anugerah; WONG, Kok Wai; HIDAYANTO, Achmad Nizar (eds.). *Neural Information Processing*. Cham: Springer International Publishing, 2021, pp. 212–223. ISBN 978-3-030-92273-3.

62. DETONE, Daniel; MALISIEWICZ, Tomasz; RABINOVICH, Andrew. SuperPoint: Self-Supervised Interest Point Detection and Description. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 337–33712. Available from DOI: 10.1109/CVPRW.2018.00060.

63. BAK, Slawomir; CARR, Peter; LALONDE, Jean-Francois. *Domain Adaptation through Synthesis for Unsupervised Person Re-identification*. 2018. Available from DOI: 10.48550/arXiv.1804.10094.

64. MING, Zhangqiang; ZHU, Min; WANG, Xiangkun; ZHU, Jiamin; CHENG, Junlong; GAO, Chengrui; YANG, Yong; WEI, Xiaoyong. Deep learning-based person re-identification methods: A survey and outlook of recent works. *Image and Vision Computing*. 2022, vol. 119, p. 104394. ISSN 0262-8856. Available from DOI: 10.1016/j.imavis.2022.104394.

65. LENG, Qingming; YE, Mang; TIAN, Qi. A Survey of Open-World Person Re-Identification. *IEEE Transactions on Circuits and Systems for Video Technology*. 2020, vol. 30, no. 4, pp. 1092–1108. Available from DOI: 10.1109/TCSVT.2019.2898940.

66. ZHENG, Wei-Shi; GONG, Shaogang; XIANG, Tao. Towards Open-World Person Re-Identification by One-Shot Group-Based Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016, vol. 38, no. 3, pp. 591–606. Available from DOI: 10.1109/TPAMI.2015.2453984.

67. WANG, Hanxiao; ZHU, Xiatian; XIANG, Tao; GONG, Shaogang. Towards unsupervised open-set person re-identification. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 769–773. Available from DOI: 10.1109/ICIP.2016.7532461.

68. ZHENG, Liang; SHEN, Liyue; TIAN, Lu; WANG, Shengjin; WANG, Jingdong; TIAN, Qi. Scalable Person Re-identification: A Benchmark. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1116–1124. Available from DOI: 10.1109/ICCV.2015.133.

69. ZHENG, Liang; BIE, Zhi; SUN, Yifan; WANG, Jingdong; SU, Chi; WANG, Shengjin; TIAN, Qi. MARS: A Video Benchmark for Large-Scale Person Re-Identification. In: LEIBE, Bastian; MATAS, Jiri; SEBE, Nicu; WELLING, Max (eds.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 868–884. ISBN 978-3-319-46466-4.

70. HE, Shuting; LUO, Hao; WANG, Pichao; WANG, Fan; LI, Hao; JIANG, Wei. *TransReID: Transformer-based Object Re-Identification*. 2021. Available from DOI: 10.48550/arXiv.2102.04378.

71. MEINHARDT, Tim; KIRILLOV, Alexander; LEAL-TAIXÉ, Laura; FEICHTENHOFER, Christoph. TrackFormer: Multi-Object Tracking with Transformers. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 8834–8844. Available from DOI: 10.1109/CVPR52688.2022.00864.

72. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. *Neural Computation*. 1997, vol. 9, no. 8, pp. 1735–1780. Available from DOI: 10.1162/neco.1997.9.8.1735.

73. KUO, Cheng-Hao; HUANG, Chang; NEVATIA, Ram. Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models. In: DANIILIDIS, Kostas; MARAGOS, Petros; PARAGIOS, Nikos (eds.). *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 383–396. ISBN 978-3-642-15549-9.

74. CAI, Yinghao; MEDIONI, Gérard. Exploring context information for inter-camera multiple target tracking. In: *IEEE Winter Conference on Applications of Computer Vision*. 2014, pp. 761–768. Available from DOI: 10.1109/WACV.2014.6836026.

75. HAGBERG, Aric; SWART, Pieter; S CHULT, Daniel. *Exploring network structure, dynamics, and function using NetworkX*. 2008. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

76. VIRTANEN, Pauli; GOMMERS, Ralf; OLIPHANT, Travis E.; HABERLAND, Matt; REDDY, Tyler; COURNAPEAU, David; BUROVSKI, Evgeni; PETERSON, Pearu; WECKESSER, Warren; BRIGHT, Jonathan; VAN DER WALT, Stéfan J.; BRETT, Matthew; WILSON, Joshua; MILLMAN, K. Jarrod; MAYOROV, Nikolay; NELSON, Andrew R. J.; JONES, Eric; KERN, Robert; LARSON, Eric; CAREY, C J; POLAT, İlhan; FENG, Yu; MOORE, Eric W.; VANDERPLAS, Jake; LAXALDE, Denis; PERKTOLD, Josef; CIMRMAN, Robert; HENRIKSEN, Ian; QUINTERO, E. A.; HARRIS, Charles R.; ARCHIBALD, Anne M.; RIBEIRO, Antônio H.; PEDREGOSA, Fabian; VAN MULBREGT, Paul; SCIPY 1.0 CONTRIBUTORS. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, vol. 17, pp. 261–272. Available from DOI: 10.1038/s41592-019-0686-2.

77. JONKER, Roy; VOLGENANT, Ton. A shortest augmenting path algorithm for dense and sparse linear assignment problems. In: SCHELLHAAS, Helmut; BEEK, Paul van; ISERMANN, Heinz; SCHMIDT, Reinhart; ZIJLSTRA, Mynt (eds.). *DGOR/NSOR*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 622–622. ISBN 978-3-642-73778-7.

78. KARP, R.M. *An Algorithm to Solve the mxn Assignment Problem in Expected Time O (mn log n).* 1978. Tech. rep., UCB/ERL M78/67. EECS Department, University of California, Berkeley. Available from DOI: `10.1002/net.3230100205`.

79. LIU, Wenqian; CAMPS, Octavia; SZNAIER, Mario. *Multi-camera Multi-Object Tracking.* 2017. Available from DOI: `10.48550/arXiv.1709.07065`.

80. PARK, Haesun; ZHANG, Lei; ROSEN, J Ben. Low rank approximation of a Hankel matrix by structured total least norm. *BIT Numerical Mathematics.* 1999, vol. 39, pp. 757–779. Available from DOI: `10.1023/A:1022347425533`.

81. DICLE, C.; CAMPS, O. I.; SZNAIER, M. The Way They Move: Tracking Multiple Targets with Similar Appearance. In: *2013 IEEE International Conference on Computer Vision (ICCV).* IEEE Computer Society, 2013, pp. 2304–2311. ISSN 1550-5499. Available from DOI: `10.1109/ICCV.2013.286`.

82. MILAN, Anton; REZATOFIGHI, S Hamid; DICK, Anthony; REID, Ian; SCHINDLER, Konrad. Online multi-target tracking using recurrent neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* 2017, vol. 31. No. 1. Available from DOI: `10.1609/aaai.v31i1.11194`.

83. SCHULTER, Samuel; VERNAZA, Paul; CHOI, Wongun; CHANDRAKER, Manmohan. *Deep Network Flow for Multi-Object Tracking.* 2017. Available from DOI: `10.48550/arXiv.1706.08482`.

84. FREDMAN, Michael L.; TARJAN, Robert Endre. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. *J. ACM.* 1987, vol. 34, no. 3, pp. 596–615. Available from DOI: `10.1145/28869.28874`.

85. GOLIN, M. *Bipartite Matching and the Hungarian Method.* 2006. Available also from: `www.cse.ust.hk/~golin/COMP572/Notes/Matching.pdf`.

86. WANG, Xiaogang; DORETTO, Gianfranco; SEBASTIAN, Thomas; RITTSCHER, Jens; TU, Peter. Shape and Appearance Context Modeling. In: *2007 IEEE 11th International Conference on Computer Vision.* 2007, pp. 1–8. Available from DOI: `10.1109/ICCV.2007.4409019`.

87. GRAY, Douglas; BRENNAN, Shane; TAO, Hai. Evaluating appearance models for recognition, reacquisition, and tracking. *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill., Oct.* 2007. Available also from: `www.researchgate.net/publication/228345677_Evaluating_appearance_models_for_recognition_reacquisition_and_tracking`.

88. RIBA, E.; MISHKIN, D.; PONSA, D.; RUBLEE, E.; BRADSKI, G. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In: *Winter Conference on Applications of Computer Vision.* 2020. Available also from: `https://arxiv.org/pdf/1910.02190.pdf`.

89. BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools.* 2000. Available also from: `https://opencv.org/`.

90. FISCHLER, Martin A.; BOLLES, Robert C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM.* 1981, vol. 24, no. 6, pp. 381–395. ISSN 0001-0782. Available from DOI: `10.1145/358669.358692`.

91. FISCHLER, Martin A.; BOLLES, Robert C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM.* 1981, vol. 24, no. 6, pp. 381–395. ISSN 0001-0782. Available from DOI: `10.1145/358669.358692`.

92. KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*. 1960, vol. 82, no. 1, pp. 35–45. Available from DOI: `10.1115/1.3662552`.

93. VAROL, Gül; ROMERO, Javier; MARTIN, Xavier; MAHMOOD, Naureen; BLACK, Michael J.; LAPTEV, Ivan; SCHMID, Cordelia. Learning from Synthetic Humans. In: *CVPR*. 2017. Available also from: `www.di.ens.fr/willow/research/surreal/data/`.

94. XIANG, Suncheng; FU, Yuzhuo; YOU, Guanjie; LIU, Ting. Taking A Closer Look at Synthesis: Fine-Grained Attribute Analysis for Person Re-Identification. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 3765–3769. Available from DOI: `10.1109/ICASSP39728.2021.9413757`.

95. XIANG, Suncheng; YOU, Guanjie; GUAN, Mengyuan; CHEN, Hao; YAN, Binjie; LIU, Ting; FU, Yuzhuo. *Less is More: Learning from Synthetic Data with Fine-grained Attributes for Person Re-Identification*. 2021. Available from DOI: `10.48550/arXiv.2109.10498`.

96. COMMUNITY, Blender Online. *Blender - a 3D modelling and rendering package*. Stichting Blender Foundation, Amsterdam: Blender Foundation, 2018. Available also from: `http://www.blender.org`.

97. POST, Oliver J. 2022. Available also from: `www.humgen3d.com`.

98. LEE, Der-Tsai; SCHACHTER, Bruce J. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*. 1980, vol. 9, no. 3, pp. 219–242.

99. PASZKE, Adam; GROSS, Sam; CHINTALA, Soumith; CHANAN, Gregory; YANG, Edward; DEVITO, Zachary; LIN, Zeming; DESMAISON, Alban; ANTIGA, Luca; LERER, Adam. Automatic differentiation in PyTorch. 2017.

100. TAN, Mingxing; LE, Quoc V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR*. 2019, vol. abs/1905.11946. Available also from: `http://arxiv.org/abs/1905.11946`.

101. SANDLER, Mark; HOWARD, Andrew G.; ZHU, Menglong; ZHMOGINOV, Andrey; CHEN, Liang-Chieh. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *CoRR*. 2018, vol. abs/1801.04381. Available also from: `http://arxiv.org/abs/1801.04381`.

102. HU, Jie; SHEN, Li; ALBANIE, Samuel; SUN, Gang; WU, Enhua. *Squeeze-and-Excitation Networks*. 2019. Available from DOI: `10.48550/arXiv.1709.01507`.

103. KINGMA, Diederik P.; BA, Jimmy. *Adam: A Method for Stochastic Optimization*. 2017. Available from arXiv: `1412.6980 [cs.LG]`.

104. KRASIN, Ivan; DUERIG, Tom; ALLDRIN, Neil; FERRARI, Vittorio; ABU-EL-HAIJA, Sami; KUZNETSOVA, Alina; ROM, Hassan; UIJLINGS, Jasper; POPOV, Stefan; KAMALI, Shahab; MALLOCI, Matteo; PONT-TUSET, Jordi; VEIT, Andreas; BELONGIE, Serge; GOMES, Victor; GUPTA, Abhinav; SUN, Chen; CHECHIK, Gal; CAI, David; FENG, Zheyun; NARAYANAN, Dhyanesh; MURPHY, Kevin. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. 2017. Available also from: `https://storage.googleapis.com/openimages/web/index.html`.

105. KARP, R.M. *An Algorithm to Solve the mxn Assignment Problem in Expected Time O (mn log n)*. 1978-09. Tech. rep., UCB/ERL M78/67. EECS Department, University of California, Berkeley. Available also from: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/1978/29160.html`.

Contents of the enclosed media

The project we described in this thesis consists of four repositories worth mentioning. Each dir tree lists modules involved in each project. The number specified in the description corresponds to the number of relevant lines in the module.

The lines were obtained with the following code, ensuring only relevant files are included. Each repository also contains a folder with a considerable amount of jupyter notebooks that we did not consider as production code.

```
#! /bin/bash
for d in */ ; do
    echo -n "$d"
    cd $d
    git ls-files "*.bashrc" "*.Dockerfile" "*.Dockerfile_jupyterlab" "*.py"
    ↪  "*.yaml" | xargs cat | wc -l
    cd ..
done
echo -n "./"
find . -maxdepth 1 -type f -name "*.bashrc" -o -name "*.Dockerfile" -o -name
↪  "*.Dockerfile_jupyterlab" -o -name "*.py" -o -name "*.yaml"  | xargs cat |
↪  wc -l
```

Synthetic Dataset

A repository that is responsible for generating synthetic data. The code involves tools for Blender automation and simulation of multiple pedestrians in scenes with multiple cameras.

- `calibration` .... files with saved camera calibrations (89)
- `scripts` .... scripts to automate the data generation process (779)
- `synthetic_dataset` .... automatizace process (3825)
  - `animation` .... module for walking animations (225)
  - `camera` .... module for controlling the blender camera object(242)
  - `misc` .... miscellaneous (116)
  - `object` .... module for representing various Blender objects (655)
  - `sample` .... tools for random sampling (121)
  - `scene` .... factories that generate static and dynamic scenes (551)
  - `trajectory` .... trajectory generator module (551)

Inter Camera Tracking

Code related to solving the assignment problem. Various evaluation methods are present, including the benchmark. This codebase also provides the source code for creating the real and orientation dataset.

- inter_camera_tracking ... (4672)
  - evaluation ... evaluation of mtmct and image retrieval (167)
  - matching ... code for solving the assignment problem (1387)
  - misc ... miscellaneous (281)
  - orientation_dataset ... files involved during creating the orientation dataset (310)
  - scripts ... scripts for automating various tasks (250)
  - smart_dataset ... sequence of scripts necessary for building the real dataset (1597)

Global Descriptor

Repository for training the global descriptor.

- global_descriptor ... (6137)
  - config ... training configuration files corresponding to experiment runs (265)
  - database ... interface for the FiftyOne database (1001)
  - dataset ... data loading pipelines for training with PyTorch (1443)
  - evaluate ... scripts for generating reports, results, and evaluations (836)
  - loss ... definitions of cost functions (357)
  - model ... modified EfficientNet_b0 (554)
  - train ... code for scheduling and multi-task training (1101)

Camera Geometry Utils

A repository with general camera geometry transformations that we used across mentioned projects.

- test ... unit tests (115)
- camera_geometry_utils ... (1136)
  - camera ... object representing a pinhole camera (479)
  - homography ... source code for obtaining a homography (82)