

**Master Thesis**

# **3D Pedestrian Tracking using Stereo Images**

Gottfried Wilhelm Leibniz Universität  
Institute for Photogrammetry and Geoinformation

**Winter Semester 2022/23**

**Submitted by:**

Aswin Lal, 10027318

**First examiner:**

Prof. Dr.-Ing. habil. Christian Heipke

**Second examiner:**

Dr.-Ing. Max Mehltrittter

April 17, 2023, Hannover



## Statement

I declare that this thesis is the result of independent research conducted by me under the guidance of my supervisor. It does not contain the results of any other scientific research that has been published or written by any other individuals or groups, except for those already cited in the thesis. Furthermore, I state that this work in the same or a similar form has not been submitted to an examination authority.

HANNOVER, 17.04.2023

Place, Date

A handwritten signature in black ink, appearing to be 'J. K.' or similar, written over a horizontal line.

Signature

# Abstract

Pedestrian tracking finds several applications in surveillance, autonomous driving, security and many more. When monocular tracking is a widely discussed and researched area in computer vision with several algorithms and frameworks in existence, combining information from multiple viewing angles and retrieving relevant 3D information of the tracked pedestrians still remains a challenging task. In view of this, the thesis aims at tackling the problem of multi-view pedestrian tracking by tracking pedestrians in 3D using a pair of stereo images. Pedestrians in each frame of a sequence are detected using the popular Mask-RCNN framework which detects the target in a frame and regresses the coordinates of a bounding box which is then fit tightly around it and finally gives a “mask” for the entity using pixel-wise segmentation. Leveraging the possibilities (like the disparity map) of the geometrical constraints inherent in the scene set-up, in addition to the well established data driven approaches using neural networks, allows the shifting of the observed scene from the 2D image plane to a 3D coordinate system. Drawing inspiration from the popular “tracking-by-matching” paradigm, given a set of detections (in the form of masks and 2D bounding boxes) for a frame, under favorable circumstances (like the absence of occlusions), each detection is matched to its instance in the subsequent frame to form a track (and predict the 2D bounding boxes) using the concept of optical flow. The issue of identity switches among the tracked pedestrians is dealt with a re-identification algorithm. The tracking algorithm was tested on the KITTI dataset and its evaluation using some of the most common evaluation metrics (like MOTA, MOTP and IDF1) shows that even when the proposed methodology provides promising results, the tracks obtained still suffer from issues like identity switches and in some cases, the separation of one long track to form two shorter tracks, which raises questions about the reliability and robustness of the re-identification and tracking strategies in complicated scenarios, including partial or complete occlusions of pedestrians and the inability to accurately localize a pedestrian with increase in “depth” from the camera pair. With the limitations aside, the proposed methodology was successful in implementing the tracking of pedestrians in a 3D object space by producing trajectories. The thesis analyzes the different algorithms and results for their achievements and shortcomings and concludes by putting forth ideas for improving the obtained results and their possible outcomes.

# Contents

<b>List of Figures</b>	<b>VII</b>
<b>List of Tables</b>	<b>VIII</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Contributions . . . . .	3
1.3 Outline of the Thesis . . . . .	3
<b>2 Related Works</b>	<b>4</b>
2.1 Detection . . . . .	4
2.2 Tracking . . . . .	4
<b>3 Theoretical Background</b>	<b>7</b>
3.1 Deep Learning . . . . .	7
3.1.1 Artificial Neural Networks (ANNs) . . . . .	7
3.1.2 ResNet . . . . .	9
3.1.3 Mask-RCNN . . . . .	10
3.1.4 Siamese Networks for Similarity Learning . . . . .	14
3.2 Geometry of an Image Pair . . . . .	16
3.2.1 Camera Parameters . . . . .	16
3.2.2 Epipolar Geometry . . . . .	17
3.2.3 Triangulation . . . . .	19
3.3 Dense Stereo Matching . . . . .	20
3.3.1 Semi-Global Matching . . . . .	20
3.4 RANSAC . . . . .	22
3.5 Optical Flow . . . . .	23
3.5.1 Flow Determination . . . . .	23
3.5.2 Feature Selection . . . . .	24
<b>4 Methodology</b>	<b>26</b>
4.1 Detection and 3D Localization . . . . .	26
4.2 Ground Extraction . . . . .	28
4.3 Tracking . . . . .	31
4.4 Re-identification . . . . .	34
<b>5 Experimental Setup</b>	<b>37</b>
5.1 Datasets . . . . .	37
5.2 Training and Hyper-parameter Settings . . . . .	38
5.3 Evaluation Metrics . . . . .	40

<b>6</b>	<b>Results</b>	<b>43</b>
6.1	Detection . . . . .	43
6.2	Dense Stereo Matching and Triangulation . . . . .	45
6.3	Ground Extraction . . . . .	47
6.4	Optical Flow . . . . .	50
6.5	Re-identification . . . . .	52
6.6	Predicted Trajectories . . . . .	54
6.7	Evaluation Metrics . . . . .	59
<b>7</b>	<b>Conclusions and Future Work</b>	<b>63</b>
<b>8</b>	<b>References</b>	<b>XI</b>

## List of Figures

1	Perceptron using a single neuron . . . . .	7
2	2D Convolution . . . . .	9
3	Residual learning: A building block . . . . .	10
4	Faster-RCNN network . . . . .	12
5	Mask-RCNN framework . . . . .	13
6	Mask-RCNN detections . . . . .	14
7	The siamese architecture . . . . .	15
8	Triplet loss interpretation . . . . .	16
9	Epipolar geometry of a stereo image pair . . . . .	18
10	Reference image and ground-truth disparities. . . . .	20
11	Aggregation of costs from all directions . . . . .	22
12	Mask-RCNN detections for a crowded scene. . . . .	27
13	Test environment and disparity map . . . . .	29
14	V and U-disparity maps . . . . .	29
15	3D pedestrian point and its projection on the ground plane . . . . .	30
16	Correction of bounding boxes using back-projection . . . . .	31
17	Workflow of the tracking algorithm in 2D . . . . .	32
18	Workflow of the tracking algorithm in 3D . . . . .	34
19	Examples of results of the re-identification network . . . . .	36
20	Sample images from the Market-1501 dataset. . . . .	37
21	Intersection over Union . . . . .	40
22	Offset between the predicted position and ground-truth . . . . .	42
23	Detection results . . . . .	43
24	Inaccurate detection results . . . . .	44
25	Inaccurate detection results across views . . . . .	44
26	A stereo image pair and the disparity map . . . . .	45
27	Matching results of pedestrians . . . . .	46
28	3D point cloud of a pedestrian . . . . .	46
29	Results for V and U-disparity maps . . . . .	47
30	Ground pixels identified for a reference image . . . . .	48
31	Foot positions of the pedestrians . . . . .	48
32	Bounding box improvement during tracking . . . . .	49
33	Incorrect predictions of bounding boxes during tracking . . . . .	50
34	Reliable points selected for tracking . . . . .	50
35	Reliable points for tracking selected in frame $t_1$ . . . . .	51
36	Predictions of the points in frame $t_2$ . . . . .	51
37	Example of a bounding box prediction and key-frame update . . . . .	52
38	Example of a bounding box prediction and key-frame update . . . . .	52
39	Examples of ground tracks obtained for tracked pedestrians . . . . .	52
40	Example of an identity switch . . . . .	53

41	The identity switch corrected using the re-identification algorithm . . . . .	53
42	The re-identification algorithm failing to resolve a switch in identity. . . . .	54
43	Pedestrian 15 tracked in two different frames . . . . .	55
44	Trajectories for pedestrian 15 in 3D . . . . .	55
45	Trajectories for pedestrian 15 in 2D . . . . .	56
46	Pedestrian 32 tracked in two different frames . . . . .	56
47	Trajectories for pedestrian 32 in 3D . . . . .	57
48	Trajectories for pedestrian 32 in 3D and the extracted ground plane . . . . .	58
49	Trajectories for pedestrian 32 in 3D and the extracted ground plane from a different viewing angle . . . . .	58
50	Trajectories for pedestrian 32 in 2D . . . . .	59
51	Precision x recall curve for the detections in sequence 16. . . . .	60
52	Precision x recall curve for the detections in sequence 17. . . . .	60
53	Wrong annotations given in the ground-truth and the boxes obtained dur- ing detection . . . . .	61



## List of Tables

1	Metrics evaluating the detections for sequence 16 . . . . .	59
2	Metrics evaluating the detections for sequence 17 . . . . .	59
3	MOT metrics for sequence 16 . . . . .	61
4	MOT metrics for sequence 17 . . . . .	61
5	RMSE of the $x$ and $z$ -coordinates obtained for pedestrian 15 . . . . .	62
6	RMSE of the $x$ and $z$ -coordinates obtained for pedestrian 32 . . . . .	62

# 1 Introduction

Multi-object tracking is one of the most actively researched topics in computer vision. Pedestrian tracking is of particular interest since human beings are often the most interesting entities in images especially for applications like understanding human behaviours or social activities (X. Liu 2016), autonomous driving (Galvao et al. 2021) and traffic surveillance (Gawande et al. 2020), to state a few. The challenges that are most prevalent in any tracking algorithm include occlusions within the scene, identity switches and target interactions. The concept of multi-object tracking has found several upgrades over the years and to date is still an active topic of research with several benchmarks like the MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking (L. Leal-Taixé et al. 2015), MOT16: A Benchmark for Multi-Object Tracking (Milan et al. 2016) and the KITTI Vision Benchmark Suite (Geiger et al. 2012).

Tracking algorithms can be classified based on their initialization methods as “Detection-Based Tracking” (DBT) and “Detection-Free Tracking” (DFT) (Luo et al. 2021). DBT strategies rely on object detectors to localize targets in every frame and then link each detection in subsequent frames to obtain tracks, whereas DFT methods require manual initialization the targets followed by their localization in subsequent frames. DBT approaches therefore allow the smooth integration of tracks for new detections to the existing ones but at the cost of a pre-trained detector, while DFT approaches fail to initialize tracks for objects that were not included in the initialization but runs without a pre-trained detector.

Detecting the pedestrians in a given frame can be achieved, for example, by fitting a bounding box around them. 2D (Two Dimensional) bounding boxes obtained from a single view severely limit the potential of the tracking paradigm in 3D scenarios. Obtaining 3D (Three Dimensional) information from a scene is challenging and often an impossible task from single view geometry. Since humans live in a 3D world, the understanding and perception of a scene in 3D is imperative for several real life applications. Stereo cameras provide crucial information that can shift the whole tracking environment from 2D to 3D. Tracking in 3D can be further enhanced by dealing with missed or false detections due to occlusions with multiple viewing angles thus enabling a more robust reconstruction of the scene (Iguernaissi et al. 2019; X. Liu 2016).

Frameworks like the Faster-RCNN (Ren et al. 2017) are used for object detection on the image plane using 2D bounding boxes. Bounding boxes, however, come with some obvious ambiguities and disadvantages like the inclusion of background pixels and partially occluded parts of neighbouring pedestrians or other entities. This can be reduced to a large extent by obtaining segmented masks of the pedestrians from frameworks like the Mask-RCNN (He et al. 2017) thus providing a per-pixel classification of the scene and its background along with bounding boxes. A stereo set-up, formed by a pair of cameras separated by a “baseline”, observing a scene from two different view points producing two images (a reference image and a matching image or equivalently, the “left image” and the “right image”) of the scene can be used for shifting of the scene from the 2D

image plane to a 3D space defined by the parameters of the cameras involved. Tracking in such an environment has huge advantages like the localization of pedestrians in the 3D space and the knowledge of their “depth” from the baseline of the stereo system. This is done by finding correspondences between the targets in the reference image and matching image. A special case of such an image pair simplifies the matching process by reducing the search space for target pixels in the reference image to the corresponding rows in the matching image using a method called “planar rectification”. In the context of tracking pedestrians in 3D, generating the disparity image of each image pair in a given sequence can be used to establish matches between the left and right images, which can then be used together with the camera parameters of the stereo set-up to obtain the 3D positions of the matched pedestrians using the principle of triangulation (Scharstein et al. 2001; McGlone 2013; Förstner et al. 2016).

Tracking using single view geometry has seen extensive research over the years while tracking in 3D from multiple views has not yet acquired its true potential (Luo et al. 2021). With rapid advancements in the areas of autonomous driving and robotics, a three dimensional perception of the world is imperative. When there is the possibility of capturing a scene from different viewing angles, retrieving only 2D information limits the capabilities of the multi-view geometry which can otherwise enhance the scene understanding. These tasks and the challenges related to them are investigated and explored in this work. While monocular tracking tries to achieve tracks within 2D space of the image, this thesis proposes a methodology which results in the formation of tracks of pedestrians on the 2D image plane and in addition, produces tracks in a 3D model coordinate system formed by the stereo camera pair.

## 1.1 Problem Statement

Tracking of pedestrians is of great importance in several applications like autonomous driving and human-robot interaction where humans beings and robot systems function in a shared space. Being able to localize and track the movements of people is essential in such an environment to ensure safety. Improvements in the fields of photogrammetry and computer vision have produced significant improvements not only in tracking, but also in anticipating their moving directions and behaviours. While most of the advancements in the tracking domain has been made on monocular image sequences, the limitations associated with it makes tracking in 3D a relevant research topic.

Taking the limitations of monocular tracking algorithms into consideration, the objective of this thesis is to develop a deep learning based method to track pedestrians in the 3D object space using stereo images of the same scene. The set-up assumes that the cameras are time synchronized and the orientation parameters of the cameras are known and kept constant. Given such a scene and camera set-up, the main tasks of this thesis are defined as follows:

1. A 3D trajectory per pedestrian, where each trajectory is defined with respect to a reference camera.

2. A suitable representation (for example, the foot positions obtained by projecting a bounding box or pixel-wise segmentation mask onto the ground) of the position of each tracked pedestrian per frame.
3. A practical implementation of the developed methodology and its evaluation on publicly available datasets.

## **1.2 Contributions**

The thesis puts forward a methodology for tracking pedestrians on the 2D image plane and in addition, producing trajectories in the 3D object space using stereo image pairs in each frame of a sequence. The thesis also explores the possibilities of using disparity images for reconstructing the ground plane of a scene which can be used to find the foot positions of the tracked pedestrians in the 3D space. These points can be projected back on to the 2D image plane to obtain tracks of the pedestrians on the ground, even in case of partial occlusions. A re-identification strategy, which relies on the visual resemblances of the tracked pedestrians to begin, join, break and end tracks when identity switches are encountered, has been experimented with. Based on the analyzes of the results obtained from the different approaches and evaluation metrics, the thesis tries to draw conclusions on how comprehensive the methodology is, what its advantages and limitations are and based on the inferences, proposes new research directions in the 3D multi-view tracking domain.

## **1.3 Outline of the Thesis**

Chapter 2 reviews the existing literature and research papers published in the areas of object detection and tracking. The fundamental concepts and mathematical backgrounds of the different methods implemented in the thesis are discussed in Chapter 3. The proposed methodology and the steps involved in its implementation are described in Chapter 4. The experimental setup is given in Chapter 5. Chapter 6 does the visualization of the results and their evaluation based on several evaluation metrics. Chapter 7 draws some conclusions in light of the results obtained and proposes scopes for future works in the domain of tracking in 3D.

## 2 Related Works

### 2.1 Detection

Most state of the art tracking algorithms use deep learning based frameworks for detection, like the Faster-RCNN (Ren et al. 2017) which fits bounding boxes around the targets or by masking them out using the Mask R-CNN (He et al. 2017). The Faster-RCNN was introduced as an upgrade to the Fast-RCNN (Girshick 2015) in terms of speed and accuracy which, in turn, was itself an upgrade to the RCNN (Girshick et al. 2014a). Apart from the RCNN family there are other methods in existence like YOLO (Redmon et al. 2016) and SDP (F. Yang et al. 2016) for object detection. (Tian et al. 2015) developed a deep learning framework for addressing occlusions called DeepParts using strategies like parts selection and bounding box shifts. (Cai et al. 2015) deal with false positives and partial occlusions by combining hand crafted-features and fine-tuned deep CNNs using a network called CompACT-Deep. (J. Liu et al. 2016) use yet another deep neural network called multispectral DNN which detects pedestrians by combining complementary information from both color and thermal images.

In addition to fitting 2D bounding boxes, (Mousavian et al. 2017) propose a method relying on the fact that the perspective projection of a 3D bounding box should fit tightly within its 2D detection window by assuming that the training of the 2D detector produces boxes that correspond to the bounding box of the projected 3D box. (Chen et al. 2016) assume a flat ground plane constraint and sample 3D boxes in the physical world with the boxes being scored depending on contextual, shape and categorical features.

### 2.2 Tracking

**Bayesian filters:** Bayesian filters using various types of probabilistic inference models have been used for tracking objects in space (Luo et al. 2021; Fortmann et al. 1983; Kratz et al. 2012). Filters like the Kalman Filters (Kalman 1960); Rodriguez et al. 2011), the Extended Kalman Filters as in (Reich et al. 2021; Mitzel et al. 2011) and Particle Filter implementations by (H. Li et al. 2016; Fen et al. 2010 and Y. Jin et al. 2007) are different Bayesian filter implementations of monocular multi-object and pedestrian tracking, in particular, before the advent of data driven approaches using artificial neural networks. Over the last several years, methods using neural networks have proven to outperform all other methods making them the backbone of all the state of the art tracking algorithms (Krizhevsky et al. 2012 ; Chavdarova et al. 2017).

**Tracking as a graph problem:** Tracking or data association can be represented as a graph problem, where each node of the graph indicates a detection and each edge indicates a possible link (Ess et al. 2008; J. Berclaz et al. 2006), thereby modelling the tracking problem as a bipartite graph matching solved using bipartite assignment algorithms (Shu et al. 2012; Breitenstein et al. 2011) or the famous Hungarian algorithm (Huang et al. 2008; Qin et al. 2012 and Reilly et al. 2010). Other works include the use of linear-programming (Jiang et al. 2007; Jérôme Berclaz et al. 2009), K-shortest paths (Jerome

Berclaz et al. 2011), maximum multi-clique (Dehghan et al. 2015) and conditional random field models (B. Yang et al. 2012; Le et al. 2016) for solving the data association problem using graphical methods.

**Crowd handling:** Identity preservation and crowd handling can be controlled by considering factors like speed and direction of objects and modelling their interactions. (Yu et al. 2016) implement a “mutual force model” in crowded scenes wherein one pedestrian is subject to a so called “force” from other pedestrians and/or entities. This approach is based on the intuition that pedestrians are expected to change their speeds and directions to prevent collisions and when walking across streets they are expected to guide and follow one another. Also, for pedestrians especially, there exist “social-force models” (Helbing et al. 1995) and “crowd motion pattern models” (M. Hu et al. 2008) which model the movements and interactions pedestrians within a crowd. Social-force models consider pedestrians to be dependent on each other and on the environment based on several factors like velocity, acceleration and destination, allowing interactions to be modelled by minimizing an energy objective, for example, by modeling social force as energy terms (Maksai et al. 2019). Crowd motion pattern models were introduced to improve performance of tracking in densely crowded scenes where detections are often inaccurate due to partial occlusions and appearance features are unreliable. Such motion patterns are learned using several methods, for example, by considering scene structures (Ali et al. 2008), using ND tensor voting (Zhao et al. 2012) or Hidden-Markov-Models (Kratz et al. 2012). However, such methods can also suffer from inaccuracies when observed from single viewing angle due to occlusions and lead to the formation of false trajectories, for example, when the neighbours are occluded for a significantly large number of frames and then re-introduced as belonging to fresh tracks.

**Artificial Neural Networks:** Tracking based on artificial neural networks has been used widely in recent years (Pal et al. 2021). (Kim et al. 2015 and Tang et al. 2016) use features learned using CNNs, replacing the conventional hand-crafted features and (Wojke et al. 2017 and Tang et al. 2017) train CNNs on labelled datasets for models like ImageNet (Krizhevsky et al. 2012) and person re-identification datasets like CUHK03 (W. Li et al. 2014) and MARS (L. Zheng et al. 2016). Researchers have also employed attention mechanisms (Zhu et al. 2019) and Long Short-Term Memory (LSTM) neural networks (Kim et al. 2018) for solving the tracking problem. (G. Wang et al. 2018) propose a CNN architecture called “multi-scale TrackletNet” which measures the connectivity of two tracklets by combining temporal and appearance information. (Bergmann et al. 2019) suggest the “Tracktor” algorithm, which performs tracking by using only a detection step and estimating the position of a pedestrian in each frame by regressing its position in the previous frame. Neural network architectures, siamese networks in particular, have also greatly improved the performance of person re-identification strategies (Khamis et al. 2014; S. Ding et al. 2015; Paisitkriangkrai et al. 2015). A popular example of such a framework is FaceNet (Schroff et al. 2015) which is trained using the so called “triplet loss” introduced by (Weinberger et al. 2005). (Hermans et al. 2017) propose a framework

that uses a triplet model to generate embeddings of the input pedestrians which are then clustered using an unsupervised clustering algorithm for trajectory modification. A siamese architecture is implemented by (Shuai et al. 2020) with a network consisting of a backbone that is shared among the detection, tracking and re-identification branches capable of performing both detection and association in a single forward pass. Tracking using neural networks has also been used jointly with other computer vision tasks like human pose estimation (S. Jin et al. 2019; Raaj et al. 2019; Andriluka et al. 2014) and action recognition (Choi et al. 2012). A major limitation of such data driven approaches is that they may make false assumptions when subject to scenes that they were not exposed to while training.

**Optical Flow:** In addition to the popular paradigm of “tracking-by-detection”, optical flow can be used for mapping objects from one frame to another by determining a “flow vector” (Kale et al. 2015). Since the concept of optical flow is related to movement of objects on the image plane, it can be utilized to encode motion information (Walk et al. 2010). (Rodriguez et al. 2009 and Izadinia et al. 2012) link detection responses into short tracklets using optical flow. (Choi 2015) determines the movement pattern of pixels corresponding to the target enclosed by bounding boxes in adjacent frames. (Ali et al. 2008 and Rodriguez et al. 2011) use optical flow for discovering crowd motion patterns in densely crowded scenarios. An obvious disadvantage of local methods for determining optical flow is their heavy dependence on the success of extracting reliable features for tracking. This dependency can lead to major inaccuracies in several cases, depending on the light conditions and the distance of the pedestrian from the camera(s). Apart from local methods, CNNs have been used to create dense flow maps of sequences like the FlowNet (Fischer et al. 2015), DeepFlow (Weinzaepfel et al. 2013 and the work of (Sun et al. 2017).

**Multi-view:** In the multi-view domain, more complex models include multi-camera sequence reconstruction (Laura Leal-Taixé et al. 2012). (Chavdarova et al. 2017) initially train a CNN on monocular pedestrian classification and retain  $d$  of its layers to create an embedding. A concatenation of such embeddings is created, on top of which a binary classifier is trained, which they argue, can discriminate features that are similar across views (like colors) and features that differ across views (for example the lateral inversion of curves resulting from multiple viewing angles). (Peng et al. 2015) employ a Bayesian Network for handling occlusions per view and use ground locations and geometric constraints to combine the networks to form a multi-view network. (U. Nguyen et al. 2019) use the disparity image of a stereo image pair to determine the ground plane of the scene to determine the positions of the foot points of the pedestrians. This strategy has also been implemented in this thesis to obtain the positions of tracked pedestrians on the ground plane. The accuracy of most of the methods discussed above relies heavily on the quality of the detections. This thesis tries to incorporate the additional 3D information to the tracking workflow, combined with optical flow, to improve the bounding boxes predicted by the object detection framework.

## 3 Theoretical Background

### 3.1 Deep Learning

#### 3.1.1 Artificial Neural Networks (ANNs)

As it was previously mentioned, the use of deep neural networks have improved the state-of-the-art in several computer vision and artificial intelligence tasks like object detection, speech recognition and machine translation (Yann LeCun et al. 2015). At the most fundamental level, the design of artificial neural networks are inspired from the interactions of neurons in the nervous system of a human body and human brain. The way these neurons fire and trigger other neurons learning neural pathways for performing a particular task is similar to how artificial neural networks work by learning weights that influence their outputs by showing them labelled data. In 1958, Frank Rosenblatt introduced the perceptron, which was shown to have the ability to learn in accordance with associationism (Rosenblatt 1958). Figure 1 shows an example of one such perceptron that uses only a single neuron with weights  $w_{ji}$ , bias  $b_j$  and an activation function  $f$  producing the output  $y_j$ .

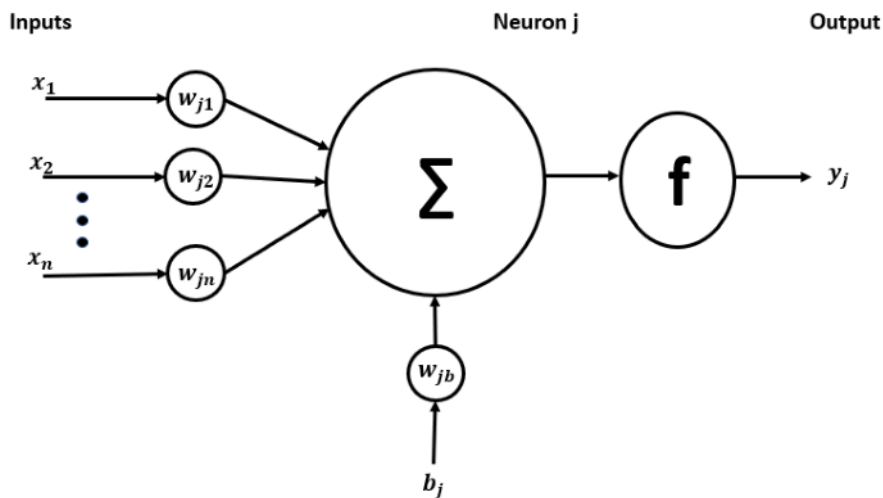


Figure 1: Perceptron using a single neuron

It accepts the inputs  $x_i$  and calculates the weighted sum:

$$z_j = \sum w_{ji} \cdot x_j + b_j = w_j^T \cdot x + w_{jb} \quad (1)$$

This is then used as the input for the activation function to give the output  $y_j$  of the neuron  $\mathbf{j}$ :

$$y_j = f(z_j) = f(w_j^T \cdot x + w_{jb}) \quad (2)$$

Such a perceptron can, for example, be used as a binary classifier creating a linear



decision boundary. Their inability to perform more complex problems were seen as huge limitations leading to a slow fading of the amount of research that was being done on neural networks for a long time. The perceptron was proven to work better by stacking one layer of neurons after another (creating intermediate layers are called “hidden layers”) to form a multi-layer neural network called as Multi-Layer Perceptrons (MLPs) (Kawaguchi 2000). Since then shallow networks have been replaced by “deeper” neural networks with hundreds of thousands (or even millions) of parameters that are learned enabling much higher levels of abstraction.

Training of a neural network involves determining or “learning” the weights by using labelled data. The entire dataset is often split into “train”, “test” and/or “validation” data. During the training phase, the network is shown only the training samples and once the network has been trained, it is evaluated using the test/validation samples. The whole training process is often carried out in epochs with random batches of the input data. Training a neural network is often done using an algorithm called “back-propagation”, originally a method used to determine the gradient of parameters during the implementation of the gradient descent algorithm (Hecht-Nielsen 1989), by minimizing what is called a “loss function”. The loss function measures the error in the output of the network using the labelled training samples. The idea of back-propagation is to measure the error at the output layer and propagate the error term back to the layer where the parameters need to be updated. Update is done using the gradient descent method. The process is also called “stochastic gradient descent” due to the random sampling of training data in each epoch. The gradient of a function shows the direction of its maximum increase and hence, the minimum can be searched for by moving in the negative direction. The rate at which the algorithm updates the parameters is controlled by an important hyper-parameter called the “learning rate” (Bishop et al. 2006; Yann LeCun et al. 2015).

The realm of deep learning found a new wave of interest and research applications with the success of Convolutional Neural Networks (CNNs) (Y. LeCun et al. 1989; Krizhevsky et al. 2012). CNNs typically contain many layers (deep networks) forming blocks each of which contain a combination of convolutional layers, activations, pooling and batch normalization followed by one or a series of fully connected layers producing an output that can be considered to be a high level interpretation of a certain part of the input. Weights are shared among the convolutional layers and can be interpreted as the coefficients of linear filters.

Convolution is a mathematical operation that measures the overlap of one function over another. Convolution of a function  $f$  with another function  $g$  can be written as  $f * g$ , where  $*$  denotes the convolution operator. Convolutions are often calculated using matrices called “kernels”. Convolutions are carried out by sliding the kernel over the input to perform element wise multiplication and summation of the products of overlapping elements. Figure 2 shows an example of 2D convolution, where the leftmost matrix represents an input image and the matrix in the middle represents the kernel. Convolution the two gives the output matrix on the right.

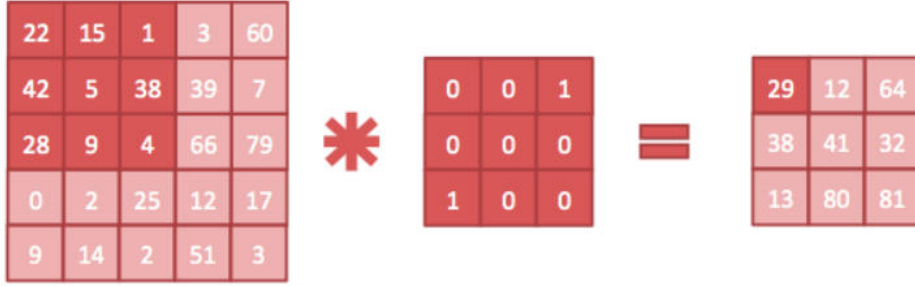


Figure 2: Example of 2D convolution (H. Wang et al. 2017)

Training neural networks often involves the learning of complex decision boundaries and mapping of highly non-linear functions. To this end, activation functions are used to introduce non-linearities into the model. Popular activation functions include *tanh*, sigmoid and the ReLu (B. Ding et al. 2018; Nair et al. 2010). Pooling operations are carried out for sub-sampling allowing the network to be invariant to small transformations and distortions in an input image. It breaks the input representations into smaller and manageable embeddings. Different pooling methods include max-pooling, average-pooling and probabilistic-pooling. Batch normalization is carried out to reduce the effects of noisy gradients by averaging the gradients over all samples in a batch (Lee et al. 2009). Methods to eliminate over-fitting in neural networks include weight decay (Xie et al. 2020), dropout (Srivastava et al. 2014), regularization methods (Szabo et al. 2004) and data augmentation (B. Li et al. 2022). Popular CNN based architectures include AlexNet (Krizhevsky et al. 2012), VGGNet (Simonyan et al. 2014), GoogLeNet (Szegedy et al. 2015) and ResNet (He et al. 2015b). The methodology proposed in this thesis relies on artificial neural networks during different stages of its implementation like object detection (Section 4.1), tracking on the image plane (Section 4.3) and re-identification (Section 4.4).

### 3.1.2 ResNet

With the general trend of designing deep networks, the networks also became increasingly more difficult to train. Increase in depth also lead to the issue of vanishing gradients (Glorot et al. 2010) and the saturation of accuracy followed by degradation when the networks start to converge (He et al. 2015a). (He et al. 2015b) introduced the ResNet architecture in 2015 taking such factors into account and showed better accuracy in image classification tasks like the ILSVRC 2015 (Russakovsky et al. 2015) as compared to other existing networks at the time. ResNet addresses the degradation problem using a deep residual learning framework. The authors show that for any desired mapping denoted as  $H(x)$ , the non-linear layers stacked in ResNet fit another mapping  $F(x) := H(x) - x$  and the original mapping is recast into  $F(x) + x$ . They argue that the optimization of the residual mapping is much easier than the original mapping.

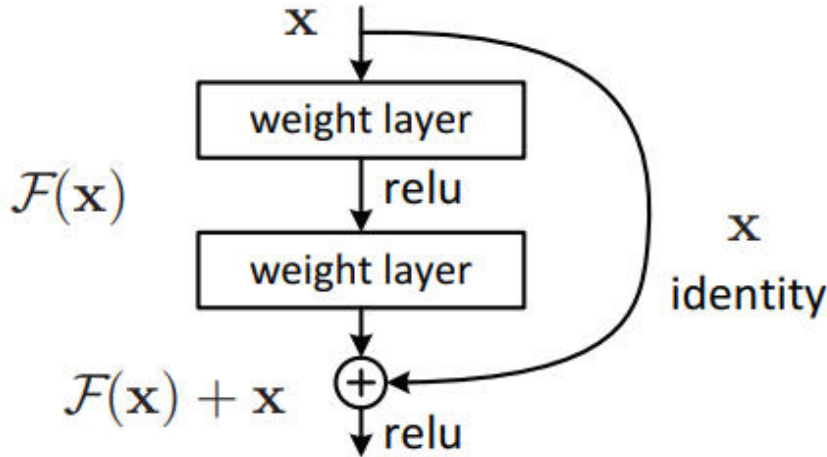


Figure 3: Residual learning: A building block (He et al. 2015b)

Feed forward neural networks are used to realize the aforementioned residual formulation with the so called “shortcut connections” (Ripley 1996). Shortcut connections refer to the skipping of one or more layers. Figure 3 shows one such network used in the ResNet architecture for residual learning using identity mapping and ReLU as the activation. The outputs of the identity mapping are added to the outputs of the stacked layers. Such a model can be trained in the conventional way using stochastic gradient descent with back-propagation.

The plain (without residual learning) network of ResNet is inspired from the VGGNet (Simonyan et al. 2014) having convolutional layers of mostly 3x3 filters. Downsampling is done using convolutional layers of stride 2. Global average pooling is done at the end of the network and contains a 1000-way fully connected layer with softmax activation. The network includes 34 weighted layers. Based on this plain network, the residual network is designed by inserting the identity shortcut connections when the input and the output are of the same dimension.

Using an ensemble of the residuals, the authors were able to achieve superior results on the ImageNet (Krizhevsky et al. 2012) and COCO (Lin et al. 2014) object detection datasets. With the success of the architecture, ResNet has been widely used especially in research and for developing frameworks that uses it as the backbone. It also serves as the backbone for the detection and re-identification networks implemented in this thesis.

### 3.1.3 Mask-RCNN

Mask-RCNN (He et al. 2017) is a popular framework for performing instance segmentation in many computer vision tasks. For an application like pedestrian tracking, exact localization of pedestrians in each frame is crucial for accurate outcomes. Frameworks like Faster-RCNN (Ren et al. 2017) achieve this by confining the targets within rectangular bounding boxes. Mask-RCNN can be seen as an extension to the Faster-RCNN by adding to the classification and regression branches, a “mask branch” which is a Fully Convolutional Network or FCN (Long et al. 2015) applied to each Region of Interest (RoI) for

predicting a segmentation mask. FCNs are commonly used for pixel-wise classification tasks and are capable of operating on input images of any given size and producing outputs of corresponding (possibly re-sampled) spatial dimensions. The Mask-RCNN along with bounding boxes, also generates a segmentation mask for the detected object. This results in a pixel-wise classification of the targets separating them from other entities and the scene background. Such segmentation masks have several advantages over conventional 2D bounding boxes. The bounding boxes, for example, in most cases include several pixels belonging to the background of a scene. This pollutes the inputs given to appearance based models. Such problems also exist in cases of bounding boxes enclosing pixels belonging to neighbouring pedestrians or other entities present in the scene due to occlusions.

The Region based-CNN (Girshick et al. 2014a) implements object detection by producing object proposal regions from an input image which are then given to the classification and regression heads. The classification head checks for the “objectness” of the region by assigning a confidence score to each proposal measuring the probability of the proposal to belong a particular class. The regression head improves the coordinates of the bounding boxes to make them fit tighter around the target. (Girshick 2015) improved this approach by using several layers of convolutions and max-pooling on the input image to form a feature map. A RoI pooling layer is then used to extract a fixed length feature vector from the feature map and feed it into a sequence of fully connected layers which eventually branch into the classification and regression heads. This framework was proven to be much faster and more accurate than its predecessor. Faster-RCNN introduced another stage called the Region Proposal Network (RPN) on top of the features produced by the convolutional layers of the Fast-RCNN by building additional layers of convolutions that simultaneously regress the box proposals and confidence scores. For the purpose of unification of RPNs with the Fast-RCNN object detection networks, the authors suggest a training procedure that alternates between the fine-tuning for the region proposal task followed by the fine tuning for detection, while keeping the proposals fixed. The convolutional layers are shared at test time, thus minimizing the marginal cost of computing proposals. At test time, the proposals created by RPNs might have high overlaps. Therefore, these proposals are subject to non-maximum suppression (Girshick et al. 2014b) based on their confidence scores. Following this, only the top- $N$  ranked proposals are used for detection.

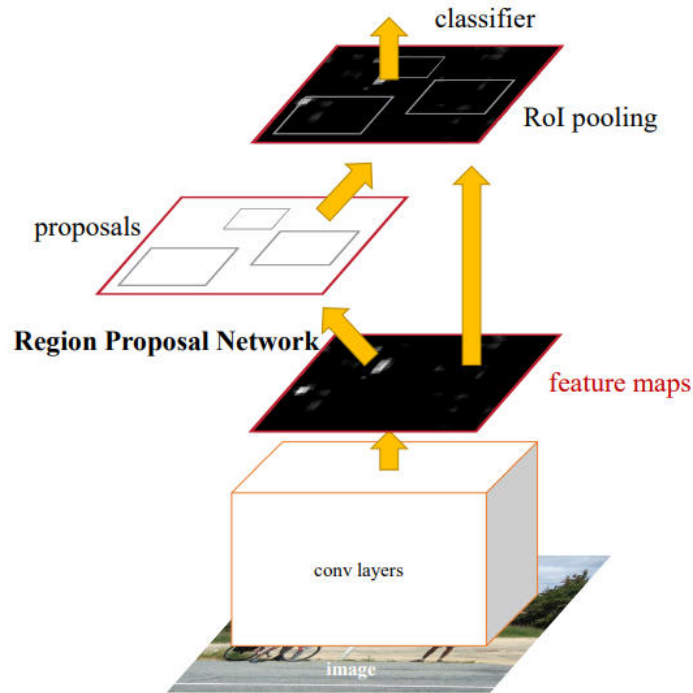


Figure 4: Faster-RCNN network (Ren et al. 2017)

Figure 4 shows the implementation of the Faster-RCNN as given in the original paper which consists of a fully convolutional layer called the Region Proposal Network for producing region proposals as the first module and a Fast-RCNN detector as the second module, thus creating a unified network for object detection.

As already mentioned, the Mask-RCNN was an extension to the Faster-RCNN which was not designed for pixel-wise classification with its inherent pixel-to-pixel misalignment due to the RoI pooling. The “RoIAlign” layer of Mask-RCNN fixes this misalignment issue by preserving the exact spatial locations. Misalignment or loss of information arises due to quantization involved in the RoI pooling operation. RoIAlign deals with this issue by avoiding quantization and in each RoI bin, bi-linear interpolation (Jaderberg et al. 2016) is used to compute the values of the input features at four regularly sampled locations and the results are aggregated. Another aspect of the Mask-RCNN is to let the RoI branch to predict the classifications so as to eliminate competition from other classes leading to a decoupling of the mask and prediction steps as opposed to the per-pixel multi-class categorization using a fully convolutional network. Figure 5 shows the basic framework of the Mask-RCNN.

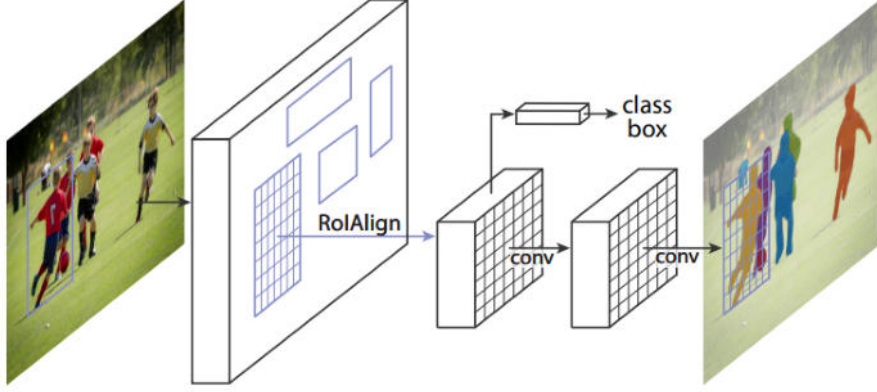


Figure 5: Mask-RCNN framework (He et al. 2018)

A multi-task loss is defined during training on each RoI as:

$$L = L_{cls} + L_{box} + L_{mask} \quad (3)$$

where  $L_{cls}$ ,  $L_{box}$  and  $L_{mask}$  represent the losses for classification, bounding box and segmentation mask respectively.  $L_{cls}$  and  $L_{box}$  are defined similar to how they are for the Fast-RCNN. For each of the  $K$  classes, the mask branch has a  $Km^2$ -dimensional output for each RoI encoding  $K$  binary masks of resolution  $m \times m$ . A per-pixel sigmoid is applied to this, defining  $L_{mask}$  as the average binary cross-entropy loss. Contributions to the loss from other mask outputs are avoided by defining  $L_{mask}$  only on the  $k$ -th mask for an RoI associated with the ground-truth class  $k$ . Similar to the Fast-RCNN, a RoI is accepted only if it has an Intersection over Union (IoU) value larger than a threshold with the ground-truth box and only the positive RoIs are used for defining the mask loss,  $L_{mask}$ . The target mask is obtained as the intersection between RoI and its corresponding ground-truth mask.

At test time, the box prediction branch is run on the proposals obtained from the backbone network followed by non-maximum suppression. Masks are computed only for the top 100 detection boxes and the mask branch predicts  $K$  masks per RoI, but only the  $k$ -th mask is retained,  $k$  being the class predicted by the classification branch. The floating number mask of size  $m \times m$  is then resized to the RoI size and binarized at a threshold of 0.5.

The authors recommend backbones like ResNet and Feature Pyramid Network (Lin et al. 2017) for feature extraction as they provide superior accuracy and speed. Mask-RCNN outperforms other state-of-the-art frameworks in segmentation and was the winner of the COCO segmentation challenges of 2015 and 2016. The Mask-RCNN framework has been used in this thesis for detecting pedestrians on the images (Section 4.1). Figure 6 give some examples of the results obtained using the Mask-RCNN for object detection and segmentation on the COCO dataset based on ResNet-101. Results include masks (given in colors), bounding boxes, category and confidences.

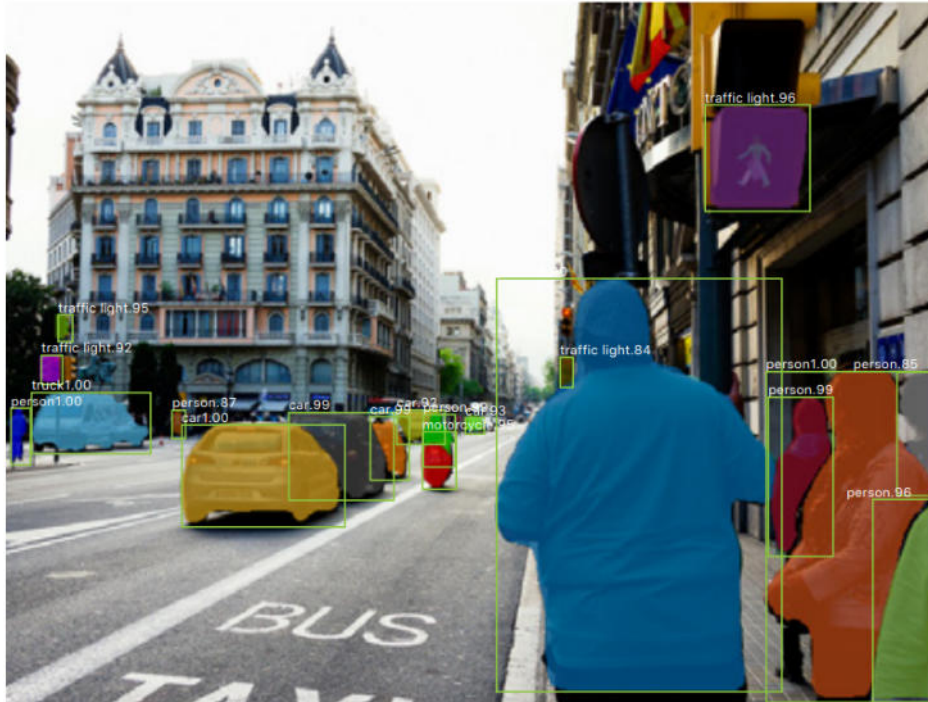


Figure 6: Mask-RCNN detections (He et al. 2018)

### 3.1.4 Siamese Networks for Similarity Learning

A really useful application of deep learning frameworks in the tracking domain is its ability to use distance-based methods which consists of learning a similarity metric from data. Such a similarity metric can be used for comparing samples that the model had not “seen” during training. This is particularly useful in a classification problem where the number of categories is high and/or not all categories are available during the training phase. (Chopra et al. 2005) propose such a framework that tries to map input images to points in a low dimensional space (like the Euclidean space) where the distance between these points is small if the images belong to the same category and large if they do not. This can be expressed as finding a function that maps input images into a target space, such that a simple distance measure like the Euclidean distance can approximate their semantic distance in the input space. To put it more mathematically, a similarity metric defined as:

$$E_W(X_1, X_2) = \|G_W(X_1) - G_W(X_2)\| \quad (4)$$

given a family of functions  $G_W(X)$  parameterized by  $W$  and two categories  $X_1$  and  $X_2$  (for example, a pair of input images), the goal is to find a value for  $W$  such that  $E_W(X_1, X_2)$  is small if  $X_1$  and  $X_2$  are from the same category and large if they are not. An architecture of neural networks that is designed following such an approach is called a “siamese architecture” because both inputs are processed by the same function  $G$  with the same parameter  $W$ . Figure 7 is an example of such a siamese architecture that accepts a pair of input images  $X_1$  and  $X_2$  which are then passed on to a network consisting of two

identical convolutional layers. An important factor of such convolutional layers is that their weights are shared.

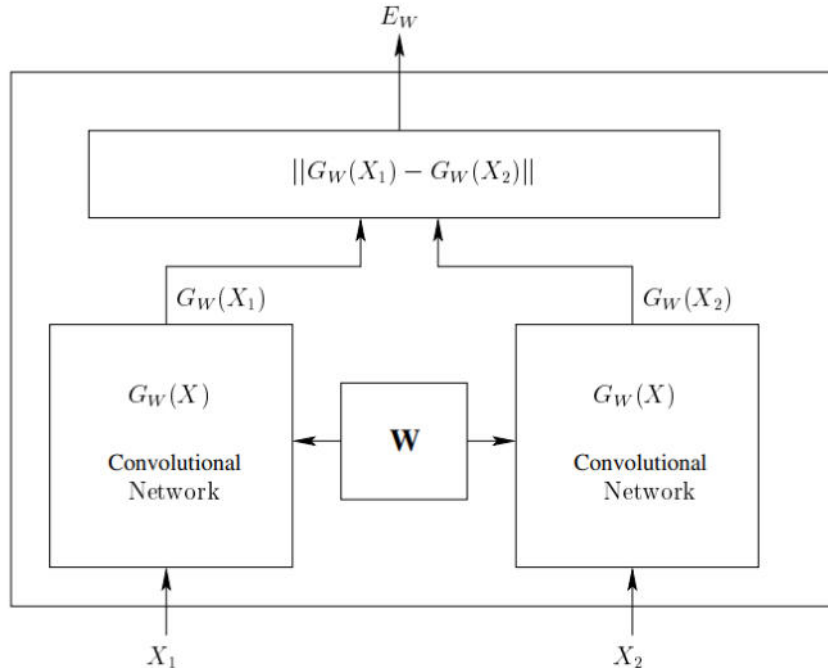


Figure 7: The siamese architecture (Chopra et al. 2005)

During training, the network tries to find  $W$  that minimizes a loss function that is evaluated over a training set. The loss function includes a *contrastive term* to make sure that not only is the energy (distance) for a similar input pair small, but also that the energy for a pair of dissimilar is large. Following this idea, (Schroff et al. 2015) employ what is called a “triplet loss” for better performance.

A triplet consists of a pair of matching entities (called “anchor” and “positive” respectively) and a non-matching entity (called “negative”) and the loss function tries to separate the matching pair from the negative by a predefined margin,  $\alpha$ . An input image,  $x$  is passed through a deep neural network to obtain an *embedding*,  $f(x)$  such that the squared distance between the identical objects is small in the feature space and that of non-identical objects is large with the loss function trying to bring the anchors,  $x_i^a$  of a specific person closer to the positives,  $x_i^p$  than to the negatives,  $x_i^n$ . This is visualized in Figure 8.



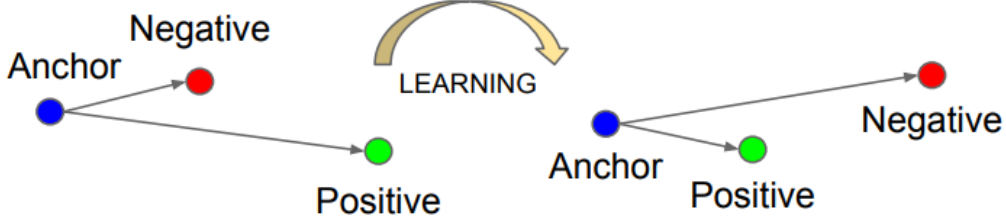


Figure 8: Triplet loss interpretation (Schroff et al. 2015)

It can be expressed mathematically as follows:

$$\|f(x_i^a) - f(x_i^p)\|^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|^2, \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T} \quad (5)$$

where  $\mathcal{T}$  is the set of all possible triplets in the training set and has a cardinality of  $N$ . The loss function can then be defined as:

$$L = \sum_i^N [\|f(x_i^a) - f(x_i^p)\|^2 - \|f(x_i^a) - f(x_i^n)\|^2 + \alpha] \quad (6)$$

Siamese networks can be trained end-to-end using the standard stochastic gradient descent approach with back-propagation. Such a siamese triplet models is used, for example, for face recognition tasks and is implemented in this thesis for pedestrian-re-identification (Section 4.4) and for improving the annotations predicted using optical flow in certain “key-frames” (Section 4.3).

## 3.2 Geometry of an Image Pair

### 3.2.1 Camera Parameters

Given a pair of cameras, separated by a baseline (distance between their projection centers), observing a scene from different viewing angles and points of the target on their image planes, the geometry of such a pair, known as a “stereo image pair”, describes the relation between the scene, the cameras and the image points. The orientation of two individual cameras can be described using their interior and exterior orientation parameters. The exterior orientation or extrinsics include the three coordinates of the projection center ( $X_0$ ) or the translation, given by a translation vector ( $T$ ) describing the translation of the camera from the origin to its position during exposure and a  $3 \times 3$  rotation matrix ( $R$ ) describing the rotation of the camera in the form of rotation angles around the camera axes. The camera axes refer to the three axes ( $X^c$ ,  $Y^c$  and  $Z^c$ ) of a coordinate system centered at the projection center  $X_0$  of a camera. In order to simplify the mapping (object into image space) relations,  $X^c$  and  $Y^c$  are chosen to be parallel to the image plane and  $Z^c$  is chosen to be perpendicular to the image plane. The intrinsic parameters or intrinsics include the focal length ( $c$ ) of the camera which is the distance

of the projection center to the image plane and the coordinates of the principle point  $(x_0, y_0)$ , which is the point on the image plane closest to the projection center or the foot of the perpendicular dropped from the projection, the scale difference,  $m$  and the shear  $s$ . The intrinsics also often include distortion parameters to undo the radial and tangential distortion that might pollute an image (Förstner et al. 2016).

There is an obvious loss in dimension as a 3D point from a scene is converted to a 2D point on a camera’s image plane. The image point  $x$  can be considered as a central projection of the object point  $X$ . It can be described using the projection matrix,  $P$  as:

$$x \sim P \cdot X \tag{7}$$

where  $\sim$  means that the two quantities are identical only upto a scale. It is used to represent the *homogeneity* of a quantity. A point  $x$  can have a homogeneous representation consisting of a *Euclidean part* and a *homogeneous part* as shown:

$$x \sim \begin{bmatrix} u \\ v \\ - \\ w \end{bmatrix} \tag{8}$$

where  $u$  and  $v$  belong to the Euclidean part and  $w$  belongs to the homogeneous part. Such a representation is useful in several contexts of photogrammetry.

$P$  is a  $3 \times 4$  matrix and has 11 degrees of freedom, meaning it depends only on 11 parameters, namely the five parameters of the camera intrinsics and the six parameters of the camera extrinsics (translation and rotation). Therefore, the  $P$  matrix can be determined from the camera extrinsics and intrinsics as given below:

$$P = K \cdot R^T \cdot [Id \mid -X_0] \tag{9}$$

where  $T$  indicates the transpose operation of a matrix,  $Id$  represents a  $3 \times 3$  identity matrix and  $K$  represents the camera calibration matrix given by:

$$K = \begin{bmatrix} -c & -c \cdot s & x_0 \\ 0 & -c \cdot m & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

### 3.2.2 Epipolar Geometry

Reconstruction of an object point from a single image is not possible due to the loss in 3D information as stated earlier. It is, however, possible to find the 3D coordinates of an object point from a stereo pair by intersecting the projection rays to the point from the two images. This needs the location of the image points of the object in both images. Given a pair of cameras separated by a baseline, the coordinates of the image point ( $x'$ )

of an object point  $X$  in one image (say the left image), the concept of epipolar geometry can be used to reduce the search space for finding its corresponding point  $x''$  (or conjugate point) in the other image to a line, as explained below.

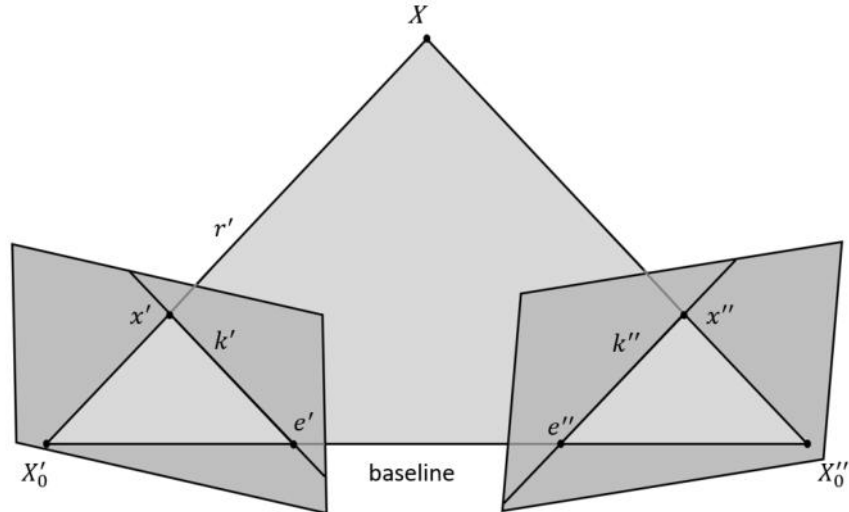


Figure 9: Epipolar geometry of a stereo image pair

The visualization of such a set-up is given in Figure 9. The projection ray from the left camera, given by its projection center  $X'_0$  and directional vector,  $r'$  and the projection center of the right camera ( $X''_0$ ) define the epipolar plane. Epipolar lines  $k'$  and  $k''$  are formed by the intersection of the two image planes with the epipolar planes and the epipolar lines intersect at the epipoles  $e'$  and  $e''$ . If an image point is given for the left image, such a set up reduces the search space for finding the corresponding image point in the right image to its epipolar line. A special case of such a set is called the “normal case” or a pair of “rectified stereo images”, where the viewing directions are parallel and orthogonal to the baseline making the image points, the projection centers and the scene point planar leading to parallel image coordinate systems. In computer vision applications, given such a pair of normalized or rectified stereo images and the image point of an object in the left image, the conjugate point in the right image can be determined as:

$$x'' = x' + d, y'' = y' \quad (11)$$

This means that the conjugate point is shifted only along the  $x$ -axis and the amount of shift is given as the *disparity* ( $d$ ) or *parallax*. Disparity can be considered as the inverse of depth or height of the object from the baseline. Therefore, if the image point of an object point is known in one such stereo pair along with its disparity value, the conjugate point on the other image can be determined (Förstner et al. 2016; Scharstein et al. 2001).

### 3.2.3 Triangulation

Reconstructing an object point in 3D from a pair of stereo images is possible using a method called triangulation. According to (Förstner et al. 2016), the optimal solution of triangulation requires two steps:

- correction of the image rays guaranteeing that they are co-planar.
- intersection of the two rays in 3D

Given a pair of stereo rectified images with an image point  $x'$  of an object  $X$  in the reference image and its conjugate point  $x''$  in the matching image and their projection matrices are  $P'$  and  $P''$  respectively, the following relations can be formulated in accordance with equation 7:

$$x' \sim P' \cdot X \quad (12)$$

$$x'' \sim P'' \cdot X \quad (13)$$

The equations above can be modified by introducing a scale factor,  $\alpha$  as:

$$x' = \alpha_1 \cdot P' \cdot X \quad (14)$$

$$x'' = \alpha_2 \cdot P'' \cdot X \quad (15)$$

On multiplying both sides of the two equations using the axiator,  $S(x)$  they can be modified as:

$$S(x') \cdot x' = \alpha_1 \cdot S(x') \cdot P' \cdot X \quad (16)$$

$$S(x'') \cdot x'' = \alpha_2 \cdot S(x'') \cdot P'' \cdot X \quad (17)$$

where the axiator is a skew-symmetric matrix defined for a point  $x$  with homogeneous coordinates  $u$ ,  $v$  and  $w$  as:

$$S(x) = \begin{bmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{bmatrix} \quad (18)$$

Multiplication of a vector with the axiator is, in effect, the cross product (Khropov et al. 2011). The cross product of a vector with itself is a null vector and therefore, the left hand sides of both equation 16 and 17 are reduced to 0. Since the scale factors  $\alpha_1$  and  $\alpha_2$  cannot be 0s, the equations can be modified as:

$$S(x') \cdot P' \cdot X = 0 \quad (19)$$

$$S(x'') \cdot P'' \cdot X = 0 \quad (20)$$

The equations above can be reduced to a single equation by collecting the products of the axiators and the projection matrices in a  $6 \times 4$  matrix,  $A$  as:

$$\underset{6 \times 4}{A} \cdot X = 0 \quad (21)$$

The optimal point can now be calculated as the singular vector of  $A$ , corresponding to its smallest singular value by applying a singular value decomposition (Förstner et al. 2016). This concept is used for finding the 3D positions of the tracked pedestrians in the object space as explained in Section 4.1.

### 3.3 Dense Stereo Matching

Dense stereo matching is the method of estimating the disparity value for each pixel (of a reference image) in a stereo image pair, hence creating a dense disparity map  $d(x, y)$ . To this end, the concept of a disparity space  $d(x, y, d)$  is introduced. The  $x$  and  $y$  coordinates of the disparity space coincide with the pixel coordinates of the reference image and therefore, for every pixel in the reference image, one obtains along with the  $x$  and  $y$ -coordinates, also the disparity ( $d$ ) of the pixel which can then be used to find the coordinates of the conjugate point in the matching image using equation 11. Stereo matching algorithms use the concept of disparity space image (DSI), which is a function or image defined over the disparity space representing the “cost” for a match given by the disparity map. A surface embedded in the DSI is obtained that follows an optimum criterion based on cost and smoothness constraints (see Section 3.3.1) to obtain a disparity map as the output. Figure 10 shows an example of a reference image in a stereo pair and its ground-truth disparities taken from the famous Middlebury Stereo Dataset (Scharstein et al. 2001).



Figure 10: Left: Reference image; Right: True disparities (Scharstein et al. 2001)

#### 3.3.1 Semi-Global Matching

Semi-global matching uses the idea of matching a pixel in the reference image to its pair using its pixel intensity, subject to the constraint expressed in equation 11 and combining multiple 1D constraints to approximate a global 2D smoothness constraint, assuming normalized stereo images. Algorithms that are based on an implicit assumption about constant disparity inside an area that is considered for matching gets violated at sharp discontinuities leading to poor results. Therefore, such an assumption is discarded. (Hirschmüller 2005) describes a rigorous workflow for determining the matching cost in

his paper, in which calculating the matching cost is based on Mutual Information (Viola et al. 1995) between two images ( $I_1$  and  $I_2$ ) defined as:

$$MI_{I_1, I_2} = H_{I_1} + H_{I_2} - H_{I_1, I_2} \quad (22)$$

where  $H$  gives the entropy of the images (meaning, their information content) and the third term gives their joint entropy calculated from the probability distributions of the intensities of the images. For further simplifications and derivations of the matching cost, the reader can refer to the work of (Hirschmüller 2005).

Pixel-wise cost can lead to errors if wrong matches have equal or lower costs than correct ones, for example, due to texture-less areas or due to noise. To eliminate such errors, in his paper (Hirschmüller 2005) defines an energy function  $E(D)$  that depends on the disparity image  $D$  for a pixel  $p$  in the base image and a “suspected” correspondence at pixel  $q$  in its pair given as:

$$E(D) = \sum_p C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \quad (23)$$

where the first term corresponds to the sum of all matching costs, the second term consists of a penalty  $P_1$  for pixels in the neighbourhood  $N_p$  of  $p$  for which disparity change is small (i.e 1 pixel) for permitting an adaptation to slanted or curved surfaces in the scene, the third term introduces a larger constant penalty  $P_2$  for disparity changes that are larger for preserving discontinuities, which are visible as intensity changes. The  $T[\ ]$  operator defines the probability distributions of matching intensities.

Stereo matching is now possible by finding the disparity image  $D$  that minimizes  $E(D)$  as given in equation 23. One approach for such a solution is using dynamic programming (Baker et al. 1981), however such a solution suffers from streaking artefacts. This is due to the utilization of 1D optimization of image rows being related to each other in a 2D image and in effect, combining strong constraints along the image rows and none along the image columns. Semi-global matching circumvents this problem by aggregating costs from all directions equally. This implies that for any pixel, the aggregated cost is calculated by summing up the costs along all the minimum cost paths that end in that pixel at disparity  $d$  as visualized in Figure 11. Following this method, the disparity  $d$  is selected for every pixel  $p$  in the base image which corresponds to the minimum cost and hence the disparity image is determined.

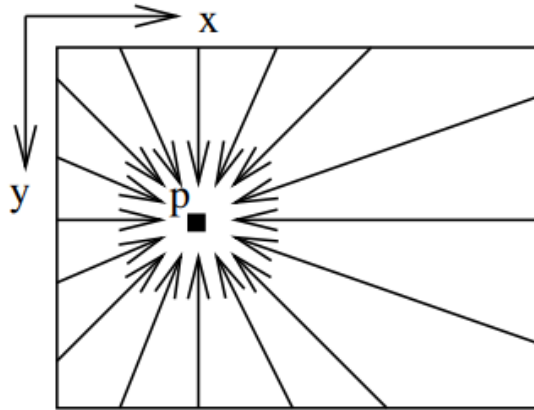


Figure 11: Aggregation of costs from all directions (Hirschmüller 2005)

### 3.4 RANSAC

(Fischler et al. 1987) introduced the paradigm, Random Sample Consensus (RANSAC) as a way to fit a model to experimental data. Methods for model fitting like least squares optimize the fit of a functional model to all of the available data and has no internal mechanism for detecting and removing gross errors. RANSAC on the other hand, can be used to fit a model using data that contain a significant amount of errors. It has been implemented in this thesis (Section 4.2) as part of extracting the ground plane of a scene for tracking the foot positions of pedestrians.

The algorithm of RANSAC randomly samples an initial set from the data and sets it as feasible and enlarges this data consistently when possible. (Fischler et al. 1987) explains the implementation using an example of fitting a circle to a given set of 2D points. Since any circle is defined by three parameters ( $x$  and  $y$ -coordinates of its center and the radius), three points are randomly chosen and the center and radius of a circle is determined. Then, the number of points close enough to the circle is determined so as to see the compatibility of the fit with the rest of the data. This is defined using a predefined threshold (called “error tolerance”). If the result is satisfactory, the “inliers” are used to perform a smoothing technique like least-squares to improve the estimated parameters. If not, the process is iterated. Another important hyper-parameter is the lower bound for the size of an acceptable consensus set. Depending on the problem at hand, this has to be chosen such that the consensus set is large enough to accommodate sufficient number of points for the smoothing procedure and to ensure that the correct model has been found for the data. The iterative process in the algorithm is carried out for a predefined number of trials or iterations. The proposed way of choosing the number of iterations  $N$  for a problem involving  $s$  unknowns,  $w$  percentage of inliers with a desired probability  $p$  of finding the consensus set is given by:

$$N > \frac{\log(1 - p)}{\log(1 - w^s)} \quad (24)$$

### 3.5 Optical Flow

The concept of optical flow has been implemented in this thesis to predict the position of pedestrians on the image plane by estimating a “flow vector” that connects them in adjacent frames. Optical flow can be defined as a motion field of points in an image (Vedula et al. 1999). Methods to determine optical flow has been widely discussed with the works of (Lucas et al. 1981 and Horn et al. 1981) laying the foundations. The strategies proposed in their work has inspired the implementation of determining the flow vectors between image frames in a given sequence for this thesis.

#### 3.5.1 Flow Determination

The determination of optical flow depends on the brightness patterns of pixels in images. Consider a point  $(x, y)$  (on an object) in an image at time  $t$ . The brightness of this point is given by  $E(x, y, t)$ . Suppose the object moves during a small time interval  $dt$  and the movement of the point on the image plane is given by  $(dx, dy)$ . The implicit assumption is that, given a very high frame rate, the brightness,  $E$  has not changed. Following the mathematical derivations as given in the paper by (Horn et al. 1981), this constraint can be formulated as:

$$E(x, y, t) = E(x + dx, y + dy, t + dt) \quad (25)$$

On expanding the right hand side of equation 25 about the point  $(x, y, t)$  (Taylor series expansion) :

$$E(x, y, t) = E(x, y, t) + dx \frac{\partial E}{\partial x} + dy \frac{\partial E}{\partial y} + dt \frac{\partial E}{\partial t} + \epsilon \quad (26)$$

$\epsilon$  contains the higher order differentials of  $dx$ ,  $dy$  and  $dt$ . On subtracting  $E(x, y, t)$  on both sides of equation 26, dividing through by  $dt$  and considering  $\epsilon$  to be negligible, it can be modified as:

$$\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \quad (27)$$

Using  $u$  and  $v$  to denote the unknowns in equation 27,  $\frac{dx}{dt}$  and  $\frac{dy}{dt}$ , the *Optical Flow equation* can be given as:

$$E_x u + E_y v + E_t = 0 \quad (28)$$

The problem now boils down to estimating the unknowns  $u$  and  $v$ . In order to make the optimization feasible, an additional constraint is required. This can be made possible by assuming that for the point  $(x, y)$ , its neighbouring points also show similar movement pattern and hence establishing a constant flow constraint within the local neighbourhood. Thus, for  $n$  neighbouring pixels,  $n$  equations can be formulated and the unknown vector



$[u, v]^T$  ( $T$  indicates the transpose operation) can be determined using the least squares method with the observation vector  $l$ :

$$l = \begin{bmatrix} E_t(x_1, y_1) \\ E_t(x_2, y_2) \\ \cdot \\ \cdot \\ E_t(x_n, y_n) \end{bmatrix} \quad (29)$$

and the design matrix  $A$ :

$$A = \begin{bmatrix} E_x(x_1, y_1) & E_y(x_1, y_1) \\ E_x(x_2, y_2) & E_y(x_2, y_2) \\ \cdot \\ \cdot \\ E_x(x_n, y_n) & E_y(x_n, y_n) \end{bmatrix} \quad (30)$$

The least squares solution for the unknown vector is then:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} \cdot A^T \cdot l \quad (31)$$

### 3.5.2 Feature Selection

A key aspect in the success of tracking using optical flow is to identify reliable features of the targets involved in the scene. (Shi et al. 1994) developed an algorithm to detect “good” features crucial for tracking. This was based on the Harris detector (Harris et al. 1988). Following the approach put forward by (Moravec 1980), consider a local window  $w$  (which is 1 within a specified rectangular region and 0 elsewhere) around a point  $(x, y)$  with intensity  $E$ , shifting it by  $(u, v)$  in various directions could lead to three cases:

- Constant intensities within the patch leads to small changes in all directions.
- Edge leads to large changes when shifted perpendicular to the edge and small change when shifted along the edge.
- Corner leads to large changes in all directions.

This can be expressed mathematically as the local autocorrelation function,  $S$  as given in equation 32

$$S(u, v) = \sum_x \sum_y w(x, y) [E(x + u, y + v) - E(x, y)]^2 \quad (32)$$

On approximating  $[E(x + u, y + v)]$  using a first order Taylor expansion as:

$$[E(x + u, y + v)] \approx E(x, y) + E_u(x, y)u + E_v(x, y)v \quad (33)$$

equation 32 can be modified as:

$$S(u, v) \approx \sum_x \sum_y w(x, y) [E_u(x, y)u + E_v(x, y)v]^2 \quad (34)$$

which can be further simplified as:

$$S(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \cdot M \cdot \begin{bmatrix} u \\ v \end{bmatrix} \quad (35)$$

where  $M$  is the structural tensor defined as:

$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} E_u(x, y)^2 & E_u(x, y)E_v(x, y) \\ E_u(x, y)E_v(x, y) & E_v(x, y)^2 \end{bmatrix} \quad (36)$$

Once  $M$  has been determined its eigenvalues  $\lambda_1$  and  $\lambda_2$  can be analyzed to make conclusions as follows:

- If  $\lambda_1$  and  $\lambda_2 \approx 0$ , then the point has no features of interest.
- If  $\lambda_1 \approx 0$  and  $\lambda_2$  has a large positive value, the point is part of an edge.
- If both  $\lambda_1$  and  $\lambda_2$  have large positive values, the point is part of a corner.

(Shi et al. 1994) suggest selecting a window if the following condition is satisfied:

$$\min(\lambda_1, \lambda_2) > \lambda \quad (37)$$

where  $\lambda$  being a predefined threshold.

(Harris et al. 1988) circumvent the computational expensiveness of the eigen value decomposition of  $M$  by calculating only the determinant ( $det$ ) and trace of  $M$  to evaluate the *corner response*,  $R$  given as:

$$R = det(M) - k \cdot trace^2(M) \quad (38)$$

where  $k$  is a tunable parameter. For flat regions,  $R$  gives a small value. It gives negative values for edges and positive values for corners.

## 4 Methodology

The proposed methodology accepts planar rectified stereo image sequences of a scene, assuming a high frame rate. The projection matrices of the stereo cameras  $P1$  and  $P2$  are used for triangulation in each frame. The expected results include tracks of the detected pedestrians on the image planes of the left and right stereo images and their corresponding trajectories in the 3D object space. Tracks on the image planes are visualized using 2D bounding boxes around the tracked pedestrians with identities assigned to them. Their positions on the ground plane, giving rise to “ground tracks” are also obtained. In the 3D space, both the ground tracks and trajectories following the centers of gravity of the tracked pedestrians with their corresponding identities from the image planes are obtained.

Detecting the pedestrians to be tracked is the first step and the experiments carried out as part of the thesis use the Mask-RCNN as the object detector (Section 4.1) in the first frame to get masks and bounding boxes around the detected pedestrian to initialize their tracks. The disparity values obtained using semi-global matching (Section 3.3.1) are used to match pedestrians in the left and right images and triangulation (Section 3.2.3) is done to determine their positions in the 3D object space. These points are projected on to the ground plane extracted using the process explained in Section 4.2.

The concept of optical flow is used for linking the detections on the image plane from one frame to the next (Section 4.3). The pedestrians are assumed to move only horizontally over the ground and hence are assumed to have displacements only along the  $x$  direction while making predictions using the flow vector. The object detector is re-introduced in certain “key-frames” to check for possible new detections to create fresh tracks, to improve the bounding box coordinates of current ones predicted using optical flow and to end tracks of pedestrians who may have left the scene to suppress false positives. After the tracks over all the frames have been obtained following the aforementioned steps, it may still contain identity switches and incorrectly broken or merged tracks. To improve these results, a dedicated pedestrian re-identification algorithm (Section 4.4) is implemented. This algorithm checks the tracks obtained from the previous step for possible identity switches or wrong assignment of identities to pedestrians and tries to merge broken tracks back together and break tracks that were falsely merged together. The re-identification step implements this using the TriNet architecture (Hermans et al. 2017) followed by a clustering algorithm.

### 4.1 Detection and 3D Localization

The objects of interest on images in the proposed tracking algorithm are pedestrians. The algorithm begins by detecting and localizing pedestrians on the 2D image plane using the Mask-RCNN framework as explained in Section 3.1.3. For each detected pedestrian, the Mask-RCNN provides:

- the coordinates of the 2D bounding boxes used to localize the pedestrians as  $[x_1, y_1,$

$w, h]$ , where  $x_1$  and  $y_1$  represent the  $x$  and  $y$  coordinates of the top-left corner and  $w$  and  $h$  represent the width and height of the boxes

- an instance segmentation mask for each detected pedestrian within the bounding box separating each pedestrian from each other and from the background
- a confidence score,  $\rho$  measuring the probability of the detection as being a correct one

These results are obtained for every frame in a sequence for both the left and right cameras. This results in individual detections that can be used for tracking, stereo matching and triangulation in the subsequent steps. The availability of the segmentation masks in addition to the 2D bounding boxes minimizes the errors in localizing pedestrians partially occluded by other pedestrians or other entities. Even when their bounding boxes may overlap, the masks allows a pixel-wise separation of every detected pedestrian. Figure 12 shows an example of how the Mask-RCNN produces distinct segments in crowded scenarios even when the bounding boxes of neighbouring pedestrians have considerable overlaps.



Figure 12: Mask-RCNN detections for a crowded scene.

The results include bounding boxes, segmentation masks, identified class name and confidence scores for each detection.

The next step is to determine the disparity images of every image pair in the sequence. This is done using the semi global matching algorithm described in Section 3.3.1, taking the left image in every frame as the reference image. Once the disparity images have been obtained, the instance of each detected pedestrian in the left image can be matched to its corresponding instance in the right image using the disparity values. The segmentation mask for each detected pedestrian allows the matching of only those pixels that belong to the pedestrian excluding pixels that may belong to the background or to another partially occluding pedestrian, a typical error possible when using only bounding boxes for localization and using one pixel (for example, the center of the bounding box) to represent the detection.

Once the detected pedestrians have been matched between the left and right image pairs, the projection matrices of both the cameras can be used to find the coordinates of the

matched pixels in the 3D model coordinate system as explained in Section 3.2.3. Since points belonging to the segmentation masks of the pedestrians are used, triangulation results in a 3D point cloud. The  $z$ -coordinates of each point in this 3D point cloud gives the distance or depth of the point to the baseline of the stereo set-up. Since a collection of points is now available in the 3D object space, a 3D point can be determined to represent the pedestrian in the 3D space by finding the center of gravity of the point cloud. This is done by averaging the  $x$ ,  $y$  and  $z$  coordinates of the points in the cloud. Such a 3D point representing the detected pedestrian can be considered to be a position along its 3D trajectory in the object space. This process is repeated for every frame in the sequence.

## 4.2 Ground Extraction

The disparity map obtained using the dense stereo matching algorithm can be used to distinguish between the pixels belonging to the ground and to other obstacles. Extracting the ground from a scene provides a better visualization of the tracks obtained from the tracking algorithm by projecting the center of gravity of the point cloud obtained by triangulation onto the ground plane. This is especially valid in crowded scenarios where bounding boxes overlap highly with those of neighbouring pedestrians and very little can be understood without a visualization that localizes each pedestrian better. The back-projection of a point projected onto the ground plane in the 3D space from a partially occluded pair of segmented masks back on to the image plane gives the position of the foot of the pedestrian. This also improves the dimensions of the bounding boxes given by the detector. The foot positions so obtained can be used to form ground tracks on the image plane and also in the 3D object space.

The extraction of the ground plane within the scene assumes that the ground is horizontal almost everywhere. This is particularly valid for urban scenes where vertical structures could belong to objects like buildings, traffic lights or lamp posts. (Z. Hu et al. 2005 and Zhang et al. 2010) propose ways to use the disparity image ( $D_b$ ) of a stereo pair to generate the so called “U-disparity” and “V-disparity” maps which are projections on column and row directions respectively. For example, the number of rows in the V-disparity map is equal to the number of rows of the original disparity image and the number of columns correspond to the maximum disparity value. Each column in the V-disparity map is a disparity histogram of the corresponding column in  $D_b$ . And conversely, for a U-disparity map, the number of columns equal to the number of columns of the disparity image and number of rows equals the maximum disparity. The ground is projected as a diagonal line in the V-disparity map and obstacles as horizontal lines in the U-disparity map. These projections of the obstacles and ground can be used to find pixels on the image plane corresponding to the two categories. Figure 13 shows a test environment and its disparity image and Figure 14 shows the corresponding V and U-disparity maps.

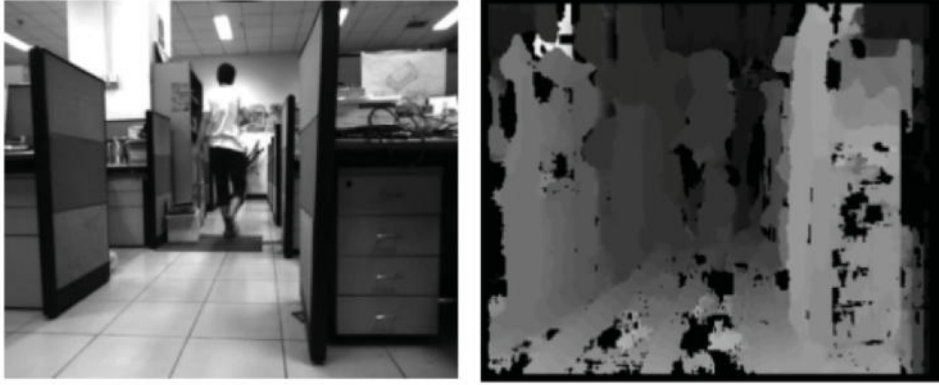


Figure 13: Left: Test environment Right: Disparity map (Zhang et al. 2010)

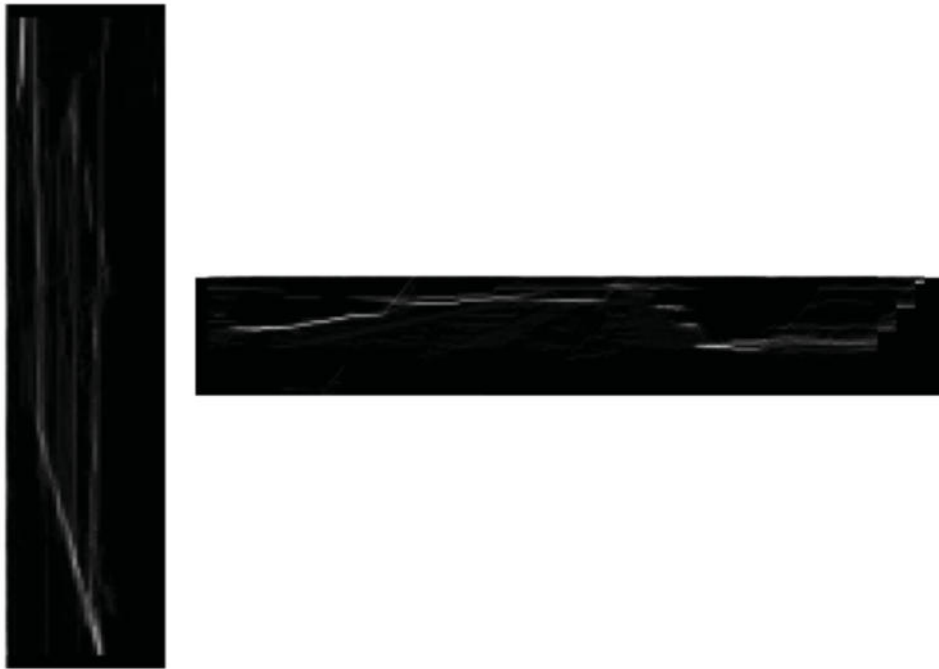


Figure 14: Left: V-disparity map Right: U-disparity map (Zhang et al. 2010)

Following the ideas put forward by (Z. Hu et al. 2005), obstacles in an image are considered to be regions with normal vector approximately parallel to the ground and regions within a disparity image containing obstacles can be seen as regions with same disparity value along the columns. Each pixel  $(u, v)$  in the V-disparity image contains a value corresponding to the number of pixels in column  $v$  of  $D_b$  with disparity  $u$ . A binary mask  $\mathcal{M}_{obstacle}$  can hence be obtained by collecting pixels in  $D_b$  corresponding to pixels in the V-disparity map with values larger than a threshold. These pixels can be considered as pixels belonging to the obstacles in the scene. Small obstacle regions can be joined using morphological closing. Since the pixels belonging to the obstacles have now been collected in the obstacle mask, the remaining pixels in  $D_b$  mostly belong to the ground plane. They can be collected to form a ground mask,  $\mathcal{M}_{ground}$ .

Since the left image was used as the reference image for dense stereo matching, the

pixels in  $\mathcal{M}_{ground}$  correspond to ground pixels in the left image. The disparities of these pixels can be used for finding pixels belonging to the ground in the right image. Once the ground pixels have been matched, they can be subject to triangulation to obtain the 3D coordinates of the ground plane in the object space. Given such a collection of points in the 3D object system, any point  $P(x, y, z)$  that lies on the ground plane has to satisfy the equation of a plane given as:

$$ax + by + cz + d = 0 \quad (39)$$

where  $(a, b, c)$  is the normal vector of the plane and  $d$  is the distance of the plane from the origin. The ground plane can now be extracted using RANSAC with the points in  $\mathcal{M}_{ground}$  subject to the constraint given in equation 39 using the steps described in Section 3.4. This also removes possible outliers due to pixels that may have been incorrectly included in  $\mathcal{M}_{ground}$ , for example, pixels belonging to other horizontal surfaces (like flat roofs of cars, buildings and so on).

With the 3D pedestrian points and the ground plane set up, the 3D foot position of the pedestrian on the ground plane can be obtained. This can be done by determining the projection of the 3D pedestrian point on the ground plane. An example of projecting a 3D pedestrian point onto the ground plane obtained using RANSAC is shown in Figure 15. The origin of such a coordinate system corresponds to the projection center of the left camera in the stereo set-up, as defined by the projection matrices.

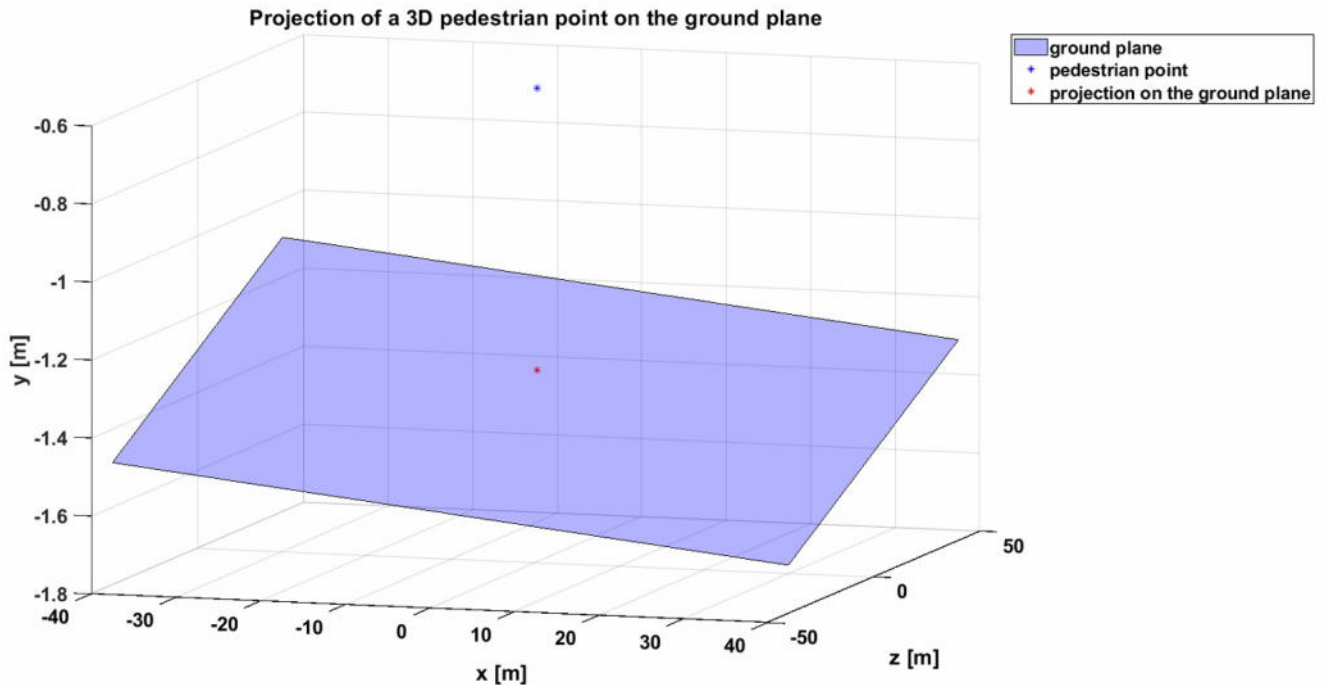


Figure 15: Visualization of a 3D pedestrian point in object space and its projection on the ground plane

These 3D foot positions of pedestrians can be back-projected onto the image plane using the projection matrix (equation 12). This helps in tracking the movement of pedestrians along the ground and for improving the coordinates of the bounding box on the image plane even in cases of partial occlusions as shown in Figure 16.



Figure 16: Left: Bounding boxes around partially occluded targets; Right: Bounding boxes corrected by back-projecting the foot positions in 3D onto the image planes (U. D.-X. Nguyen 2020)

### 4.3 Tracking

Given a set of  $n$  pedestrian detections  $DL^{t_0} = \{DL_1^{t_0}, DL_2^{t_0}, \dots, DL_n^{t_0}\}$  and  $DR^{t_0} = \{DR_1^{t_0}, DR_2^{t_0}, \dots, DR_n^{t_0}\}$  for a frame  $t_0$ , for the left and right images respectively from the Mask-RCNN, dense stereo matching and ground plane extraction are carried out as explained previously. Each pedestrian is cropped out to determine the points that are reliable for tracking based on the approach described in Section 3.5.2. Once the points have been determined, the image for frame  $t_1$  is used to determine the optical flow of the selected points of each detection in  $DL^{t_0}$  and  $DR^{t_0}$  following the steps explained in Section 3.5.1. The magnitude of the flow vector gives the displacement of the selected points and the sign gives the direction (left or right) of the displacement. Given the high frame rate of the KITTI dataset and considering the fact that pedestrians move horizontally on the ground, the displacements of the selected points in the vertical direction is assumed to be negligible and discarded. The displacements in the horizontal direction are averaged over the selected points of each detection to obtain the mean displacement of each pedestrian between the two frames and predict the bounding boxes of the detections in  $t_1$ . The mean displacement can also be used to predict the foot points of the pedestrians in  $t_1$ , given their foot positions in  $t_0$ . Using the segmentation mask of the previous frame and the predicted foot positions in  $t_1$  the height of the bounding box predicted for  $t_1$  can be approximated. The  $y$ -coordinate of the predicted foot is assigned as the  $y$ -coordinate of the bottom-right corner of the bounding box. This allows the box to go no lower than the foot position and the width is calculated by assuming that the height of a box is three times its width. The detections for frame  $t_1$  can hence be obtained for the left and right images forming tracks and can be assigned unique identities. For each detected pedestrian, the quantities passed on to the next frame includes the frame number,  $x$  and  $y$  coordinates of the top left corner of the predicted box,  $x$  coordinate of the bottom right corner,  $y$  coordinate of the foot position and the identity assigned to the track. The



detections predicted for  $t_1$  can be used similarly for predicting the labels of the pedestrians in  $t_2$ , and so on. The identities of same pedestrians detected in the left and right images are always identical and hence can be used to form tracks with the same identities in the 3D object space using triangulation.

One caveat with such an approach relying on several strong assumptions is that a pedestrian who was fully occluded in  $t_0$  or who entered the scene in frame  $t_1$  will remain undetected. It can also happen that a pedestrian occluded partially in the left image in  $t_0$ , for example, was tracked successfully, but the same pedestrian was completely occluded and hence undetected in the right image. This would lead to a missed position in the 3D trajectory, because triangulation cannot be implemented in such a situation. Since the success of determining the flow vector relies on detecting reliable points for tracking, failure in doing so inevitably leads to wrong approximations of the flow vector and the predicted box may fall over the background or a over neighbouring pedestrian leading to false positives or identity switches. The accuracy of the stereo matching algorithm might also start to reduce with increase in depths, leading to poor results for triangulation. These issues can be dealt with by introducing the detections produced by the Mask-RCNN at regular intervals. This allows the algorithm to check for new tracks in case of new detections, end false tracks and improve the bounding box coordinates of the existing detections predicted using optical flow.

Figure 17 shows a flowchart of the steps involved in predicting the labels for the frame  $t_{i+1}$ , given the labels, the left and right images at frame  $t_i$ , which are then used to calculate the disparity map. Once the predictions have been made, the frame  $t_{i+1}$  is checked for a key frame update (not shown in the Figure for the ease of visualization). In case of an update step, the predictions are refined before they are used to make predictions for the upcoming frame.

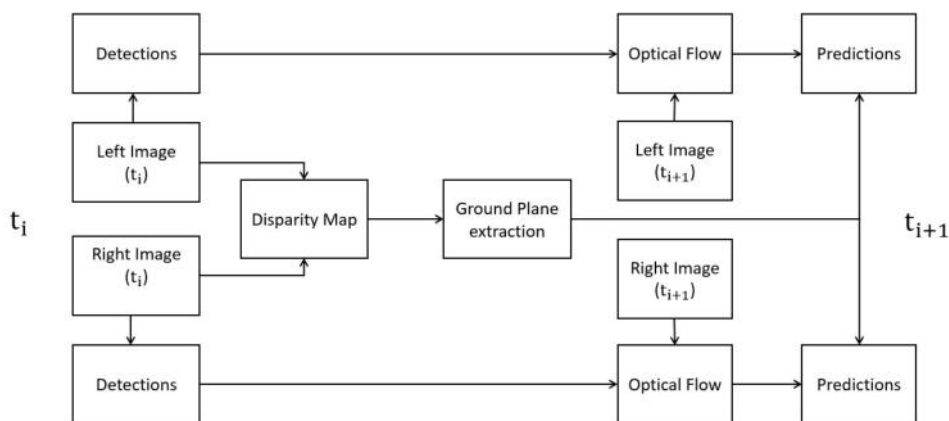


Figure 17: Workflow of the tracking algorithm in 2D

During such a “key-frame update”, the detections given by the Mask-RCNN need to be assigned correctly to each set of predictions from the previous frame. Such an assignment also includes the foot positions of the pedestrians detected. A match between each

detected and predicted pedestrian is attempted for one-to-one assignments. A strategy involving three steps is proposed for this:

- Use a siamese triplet model to predict the similarity of each pedestrian detection to the predictions from the previous frame. The cosine similarity (Unde et al. 2021; Wojke et al. 2018) of the embeddings obtained from the siamese model is determined to give a measure of similarity between two pedestrians. Such a score is often unreliable especially when two pedestrians similar in appearance, leading to similar features, are under consideration or if the boxes in question also includes parts of other entities or neighbouring pedestrians. Such a score can, however, be used to select  $m$  good matches based on appearance by defining a threshold.
- Use the 3D coordinates of the predictions and detections in the object space obtained using triangulation to calculate the Euclidean distance. Assuming that the pair that gives smallest distance to be the correct match could lead to false assignments in case of crowded scenes. But, by using a threshold, this too can be used to filter out improbable matches based on their spatial proximity in the 3D space.
- Take the best matches obtained from the previous steps and use the segmentation mask of each detection to look over overlaps within the area enclosed by the predicted bounding boxes. The bounding box obtained from Mask-RCNN corresponding to the mask with maximum overlap has to correspond to the prediction.

Given a small frequency between such updates, the third step provides an accurate match but at the cost of an exhaustive search. This can be alleviated to an extent using the first two steps. Once the labels have been updated they can be used for making predictions for the subsequent frame using optical flow as mentioned before. In case of no suitable matches, it can be assumed that the detection belongs to pedestrian who had entered the scene since the last update or was occluded until the current update and can therefore be assigned new tracks. Predictions using optical flow can also be used to end tracks if the predicted corners of the bounding boxes exceeds the boundaries of the image plane. Depth of the pedestrians from the baseline of the stereo cameras can also be used to start or end tracks based on the reliability of the matching algorithm and the complexity of the tracking environment.

Another possibility of identity switches and wrong flow estimation while using such a method arises if two pedestrians cross each other or one is being overtaken by another. This could lead to the identity of one pedestrian being carried over to the other and hence, lead to the creation a new track. Such a situation cannot be avoided without additional constraints or dedicated modeling for such scenarios and hence makes pedestrian re-identification a crucial aspect in the success of such a tracking algorithm.

The tracks over all frames obtained from the tracking algorithm given above are subject to the re-identification algorithm described in Section 4.4. Once the 2D tracks on the left and right image sequences have been refined using re-identification, the 3D trajectories can

be determined using the triangulation method as described in Section 3.2.3 and Section 4.1. Each tracked pedestrian now has a unique identity and belongs to a track on the left and right image sequences. Corresponding pedestrians on the left and right images are given the same identities. This allows in creating 3D trajectories in the object space tracking both the centers of gravity and the foot positions of the pedestrians. An overview of these steps is given as a flowchart in Figure 18

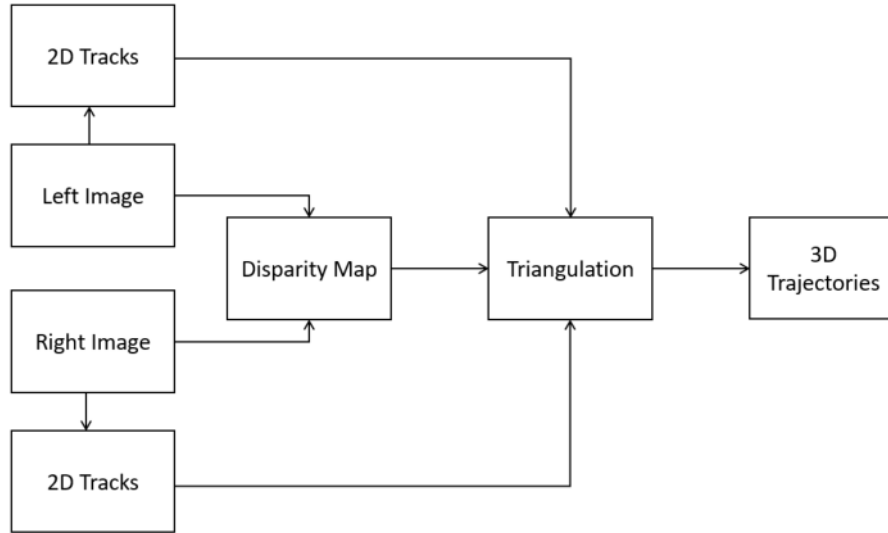


Figure 18: Workflow of the tracking algorithm in 3D

#### 4.4 Re-identification

Once the tracks have been obtained for the left and right image sequences, they are searched for possible identity switches by classifying them as being “stable” or “unstable” tracks based on their number of occurrences of their identities. The identities of stable tracks are used as “references” for tracks that may have been broken away from them. The “loss” of an identity followed by the introduction of a new identity suggests the possibility of a switch or an occlusion accompanied by a new pedestrian entering the scene. But such cases could often be signs of identity switches due to any of the scenarios mentioned in the previous section and hence, the pedestrians belonging to the tracks that were deemed unstable until that point, the reference identities, the pedestrians belonging to the missed identities and the ones belonging to the new one are subject to the re-identification process.

The TriNet architecture using a siamese triplet model implemented in the work of (Hermans et al. 2017) is used for the purpose pedestrian re-identification as it can be used independent of the tracking procedure described in the previous section and can be integrated to the workflow as a post processing step to merge broken tracks together and to split incorrectly joined tracks.

This is done by cropping the pedestrians from the images around their bounding boxes

and feeding them to the TriNet model, which produces embeddings for each input pedestrian. These embeddings are then be subjected to a clustering algorithm. Being a popular clustering mechanism, the K-means clustering algorithm has been used for this purpose as it is highly effective in mining patterns involved in the data with a relatively high convergence speed (Y. Li et al. 2012; Bishop et al. 2006). K-means is an unsupervised method for recognizing patterns and works by clustering the input data based on a criterion of distance between the cluster centers. It is an iterative process that starts with a random cluster center but iteratively updates the clusters based on the distance of similar or dissimilar samples from the cluster centers.

The clustering algorithm is expected to converge and brings together instances of the reference tracks that may have broken away from it. It also separates two falsely joined pedestrian tracks that had formed a single track. A pedestrian who is not identified as belonging to any of the reference identities is left with its current identity and observed in the further frames for possible merges or splits.

Such an algorithm that takes into account only the appearance cues could often result in mismatches due to the pedestrian reappearing in the scene after a prolonged occlusion rendering the clustering algorithm unable to find a suitable match for it. This could be understood by considering a situation where a pedestrian belonging to an identity that was considered to be stable, walking towards the camera (or conversely, away from the camera), who gets detected after a large number of frames due to an occlusion (for example, detected in the 20th frame and occluded until the 100th frame). The pedestrian gets a new identity and might appear different due to an enlargement (or shrinkage) in the dimensions of the bounding box in the recent detection. The cropped version of reference identity that was obtained in the previous detection and the cropped image of the new identity may not necessarily belong to the same cluster during the clustering process due to differences in embeddings produced by the siamese model, even though they belong to the same pedestrian. Differences in appearances due to different lighting conditions between the two observed instances of the same pedestrian in two distant frames could also lead to wrong clustering results (for example, a person detected under sunlight gets occluded for a large number of frames and reappears in shades or shadows giving rise to different emeddings leading to different clusters). Different viewing angles could also lead to mismatches in the identities due to same pedestrians appearing differently when viewed from different viewpoints and in some cases, different pedestrians producing similar embeddings. Figure 19 shows some results of the re-identification procedure. It can be seen how two different identities when viewed from different angles were identified as one due to extremely similar appearances.



Figure 19: The query image is given in the leftmost column. This is followed by three images that were re-identified as matches to the query. The last two columns show the ground-truth matches. Correctly re-identified images are given green borders while wrong matches are given red borders. (Hermans et al. 2017)

It can also happen that due to highly similar embeddings, the clustering algorithm found more than one match for an embedding leading to two or more instances of the same identity for a single frame. This calls for a dedicated subroutine that checks for “imposters”, who should be reassigned to their original identity or given new identities. This can, once again, be done using the TriNet model followed by clustering by taking the duplicates and instances of other existing tracks in the subsequent and previous frames as inputs.

Having to employ such mechanisms only after the initial tracks have been obtained on the image planes for every frame of the left and right sequences, limits the real-time capabilities of the methodology. The sole dependence on visual cues to prevent switches in identities could inevitably lead to errors that can be dealt with by using additional geometrical constraints or by modelling the behaviour of pedestrians in a scene.

## 5 Experimental Setup

### 5.1 Datasets

**COCO:** The Mask-RCNN model for detecting pedestrians was trained on the COCO (Common Objects in Context) dataset (Lin et al. 2014). The COCO dataset has 90 different annotated classes of objects along with the “background” class leading to a total of 91 classes. It is a prominent dataset used for training models for object detection and instance segmentation. The annotated data consists of 2D bounding boxes and instance segmentation masks for detecting and localizing the object categories on images. The dataset consists of 328k images with 2.8 million labelled instances. The advantage of the COCO dataset over other popular datasets like the ImageNet (Krizhevsky et al. 2012) dataset is that even when it has fewer categories, it provides more instances of the categories for training. With nearly 270k instances of people, a model trained on the COCO dataset is a suitable choice for detecting pedestrians in a tracking scenario.

**Market-1501:** The pedestrian re-identification network (TriNet) using the siamese triplet loss was trained on the Market-1501 dataset (L. Zheng et al. 2015). This dataset is an attractive option for training a re-identification model that learns a similarity metric due its use of six cameras recording people in a campus supermarket from six different view points. The creators claim the dataset to be the largest person re-identification dataset at the time of its release in terms of the number of query images and identities of people cropped out using bounding boxes. The dataset consists of 751 identities for training with each person having, on average, 3.6 images at each viewpoint. The dataset also includes samples with visually similar appearance with different identities, which is an ideal choice for training a siamese network to learn similarities and dissimilarities. Figure 20 shows examples of samples included in the Market-1501 dataset.

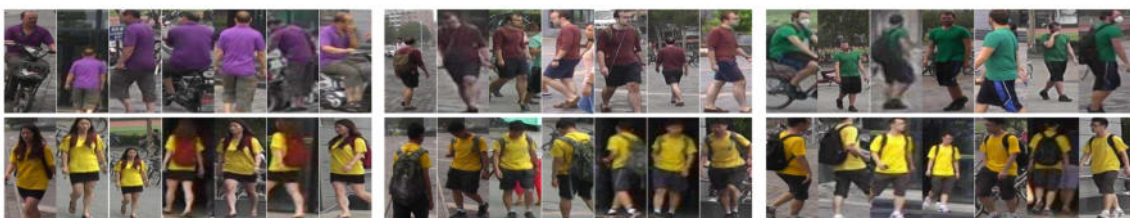


Figure 20: Top: Samples of 3 identities with distinctive appearance. Bottom: Samples of 3 identities with similar appearance. (L. Zheng et al. 2015)

**KITTI:** Training of the siamese triplet network for matching predictions obtained using optical flow and the detections obtained from the Mask-RCNN and testing of the entire tracking methodology has been carried out on the “mulit-object” tracking dataset of the KITTI vision benchmark (Geiger et al. 2012). The KITTI dataset contains street view scenes obtained by a stereo camera mounted on a car. The dataset includes RGB images for the left and right stereo cameras separated by a baseline of 0.54 m, obtained at a frame rate of 10 FPS (frames per second). Since the purpose of the KITTI dataset being

the investigation of autonomous driving applications, it focuses primarily on vehicles and pedestrians. The training set consists of 21 labelled sequences out of which 5 include pedestrians, namely sequences 13, 15, 16, 17 and 19. The ground-truth includes the 2D bounding boxes coordinates and 3D positions of the pedestrians in the object space with respect to the left camera. Since the tracking methodology implemented as part of this thesis includes the extraction of the ground plane, only the sequences (16 and 17) captured by the cameras when the car on which they were mounted was stationary, has been used for testing and evaluating the results of the implementation.

Since the COCO dataset includes a “Person” class and “Bicycle” class, it does not make a distinction between a “Pedestrian” class and “Cyclist” class. Therefore, implementing a detection model trained on the COCO dataset on the KITTI dataset leads also to the detection of cyclists as “persons” and hence will be included as a valid detection for tracking. The evaluation criteria for the KITTI dataset, however, does not consider objects belonging to neighbouring classes of pedestrians like cyclists or people who are seated as false positives.

## 5.2 Training and Hyper-parameter Settings

**Mask-RCNN:** The Mask-RCNN model for detecting pedestrians was built on FPN and ResNet-101 and hyper-parameters were tuned as suggested by (He et al. 2017 and Abdulla 2017). The model was trained with the number of steps per epoch set to 1000 and the number of validation steps set to 50 for optimal training times. Training was done on mini-batches of 16 images. The threshold for non-maximum suppression to filter the RPN proposals was set to 0.7 and 256 anchors were used per image during training. The number of maximum final detections was set to 100 with a detection being accepted only if it has a minimum confidence of 0.7. The non-maximum suppression threshold for a detection was set to 0.3. The model was trained with a learning rate of 0.001 and momentum of 0.9 with a weight decay of 0.0001 for regularization.

**Siamese triplet models:** The siamese triplet model used for matching the detections of the Mask-RCNN with the predictions obtained using optical flow was trained on the KITTI dataset using a network with ResNet-50 as the base (discarding the last layer) followed by three fully connected layers, the first layer consisting of 512 units and the last two containing 256 units each. The fully connected layers have ReLu non-linearities with intermediate layers of batch normalization. The model uses weights pre-trained on the ImageNet dataset for faster convergence. Training samples were generated by cropping out pedestrians based on their labelled bounding box coordinates and identities to form “anchors”, “positives” and “negatives” to learn the similarity metric. The size of the input pedestrians images were scaled to (64, 64) pixels. The batch size was set to 32 and the triplet loss was defined with a margin set to 1 based on the accuracy on the validation set. The model was trained using the Adam optimizer for 60 epochs with a learning rate of 0.0001.

The architecture and hyper-parameters of the TriNet model for pedestrian re-identification

were in accordance with the implementations of (Hermans et al. 2017). The TriNet model also uses the ResNet-50 as the base architecture removing the last layer and adding two fully connected layers consisting of 1024 units and 128 units respectively and the ReLU non-linearity. The input images fed into the network were resized to have a height of 256 pixels and width of 128 pixels. The batch size was limited to 72 due to the size of the network (consisting of 25.74 million parameters). The network was trained using the Adam optimizer with momentum and used a learning rate of 0.0003 and has 20k training iterations. A margin of 0.2 was used in the triplet loss for separating the similar identities and non-similar identities.

An important hyper-parameter to consider while tracking on the image plane using optical flow is the frequency in update of the labels. The frequency of such an update in “key frames” highly depends on the complexity of the tracking problem. For cameras spanning a wide area with pedestrians moving not only across the image, but also towards and away from the cameras would mean gradually shrinking or expanding bounding boxes resulting in incorrect dimensions of the boxes and in turn, wrong predictions for the subsequent frames without timely updates. Detections could be missed if the scene is expected to have new pedestrians entering frequently, as is in the case of surveillance cameras. Pedestrians moving in small clusters with varying speeds, thus going past other groups could also lead to errors unless such movements are also modelled.

Assigning a small frequency for a key-frame update, to the tracking algorithm for a scene where the number of pedestrians to be tracked is not expected to change dramatically or even allowing the predictions for every frame to be followed by a key-frame update, leads to results not much different from those predicted by optical flow and can be avoided considering the computational and time costs of such an update as described in Section 4.3. A small frequency (for example, every 5 frames or less) assumes that the bounding box dimensions of the tracked pedestrians are expected to change significantly within that time frame and should be updated accordingly. Assigning a large frequency (for example, after every 10 frames) given a scene where new pedestrians are expected to enter quite frequently could lead to false negatives or missed detections. A still larger frequency for a scene, where pedestrians are expected to enter and exit within a shorter period of time, in effect, leads to missing out completely on several tracks giving incomplete results for the tracking algorithm. Given the density of the pedestrians in each frame and the changes in bounding box dimensions for the tracked pedestrians between frames and experimenting and evaluating the results with several frequencies, it was decided to update the predicted labels after every 3 frames for the test sequences used in this thesis. As it was mentioned earlier, depth of the pedestrians can, in addition, be used for ending tracks predicted by optical flow. A minimum depth of 2m from the baseline was chosen to end tracks for both the test sequences, in addition to the boundaries of the image plane. This is to prevent inaccurate flow estimation given the fact that objects show larger displacements on the image plane at smaller depths.



### 5.3 Evaluation Metrics

IoU or Intersection over Union is a measure that determines if a detection is correct by comparing the bounding box detected with its ground-truth. It measures the overlap of a detected bounding box with the ground-truth box and divides it by the area of the union between them as shown in Figure 21.

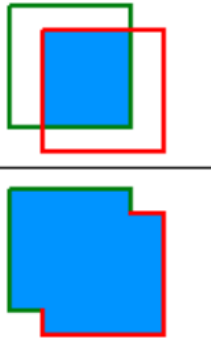
$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Diagram 1}}{\text{Diagram 2}}$$


Figure 21: Determination of the IoU between the detected bounding box and its ground-truth (Padilla et al. 2020)

While evaluating the results of an object detector, a True Positive (TP) is defined as a correct detection of an object given in the ground-truth. A False Positive (FP) is an incorrect detection of an object that does not exist and a False Negative (FN) is defined as a ground-truth box that is undetected. True Negatives (TN) do not apply in the context of object detection because one can imagine infinite number of boxes in a scene that should not be detected. A detection can be considered to be True Positive if the IoU of its bounding box with the ground-truth box is greater than a threshold. The KITTI evaluation criteria specifies a threshold of 0.5 for declaring a detection a TP.

Such a classification of the detections can be used to calculate two evaluation metrics: precision and recall. Precision gives a measure of the ability of the detector to detect only the relevant objects. Hence it measures the “correctness” of the detector. It is defined as:

$$precision = \frac{TP}{TP + FP} \quad (40)$$

Recall on the other hand measures the “completeness” of a detector by measuring its ability to detect all ground-truth boxes. It is defined as:

$$recall = \frac{TP}{TP + FN} \quad (41)$$

The precision and recall values can be used to plot a precision x recall curve. For different confidence values given by the object detection framework, the plot can be seen as a trade-off between precision and recall. A good object detector can be defined as one that detects all the boxes given in the ground-truth while at the same time detecting only the relevant ones, which means the precision of the detector should stay high as its recall increases. This can be visualized using the precision x recall curve. The area under

the curve (AUC) is determined to estimate the average precision (AP) of the detector. The curve obtained typically has a zig-zag shape and can be summarized by averaging the maximum precision values observed at each recall level. Such a method has been carried out as per the interpolation method given by (Padilla et al. 2020). The APs of all object categories are averaged to determine the mean average precision (mAP). Since the tracking algorithm focuses only on pedestrians, the number of object categories for detection is 1, thereby making the mAP mathematically equal to the AP.

MOTA and MOTP are two popular metrics for evaluating the the results of a tracking algorithm (Bernardin et al. 2008). MOTP measures the accuracy of the algorithm in localizing the pedestrian using bounding boxes by calculating the average IoU between the detections and the ground-truth boxes. For every detection  $i$  in frame  $t$ , *MOTP* is determined as:

$$MOTP = \frac{\sum_{i,t} IoU_t^i}{\sum_t c_t} \quad (42)$$

where  $c_t$  is the number of matches between the ground-truth and detections in frame  $t$ .

MOTA measures the accuracy of both the tracking and detection algorithms by taking into account, for frame  $t$ , the identity switches *ID* along with the *F*Ps and *F*Ns which are added up and divided by the number of objects present at  $t$ , given by  $g_t$ , to get the total Error rate  $E_{tot}$  given as:

$$E_{tot} = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t g_t} \quad (43)$$

MOTA is then defined as:

$$MOTA = 1 - E_{tot} \quad (44)$$

IDF1 is a metric that measures the consistency of tracks obtained by the tracking algorithm by comparing the obtained identities to the ground-truth tracks. An overlap with the ground-truth track greater than a threshold leads to IDTPs (identity true positives), IDFPs (identity false positives) are non-overlapping tracks and unmatched tracks become IDFNs (identity false negatives) (Luiten et al. 2021). The values can then be used to determine IDF1 as given in the equation below:

$$IDF1 = \frac{IDTP}{IDTP + 0.5 \cdot IDFN + 0.5 \cdot IDFP} \quad (45)$$

The predicted trajectories can be evaluated by comparing them with the ground-truth. Since the predicted trajectories in the 3D space vary from those given by the ground-truth along the  $y$  direction, due to different ways of choosing representative tracking points in the 3D space (a small deviation only along the  $y$  direction still corresponds to the same pedestrian), the comparisons are made along the  $x$ - $z$  plane. The  $z$  axis of such a system is parallel to the optical axes of the cameras.

For every frame including the predicted trajectory, the Euclidean distance between the predicted point and the point given by the ground-truth can be determined on the  $x$ - $z$

plane. This gives the offset of the predicted position from the ground-truth for that frame. This is visualized in Figure 22.

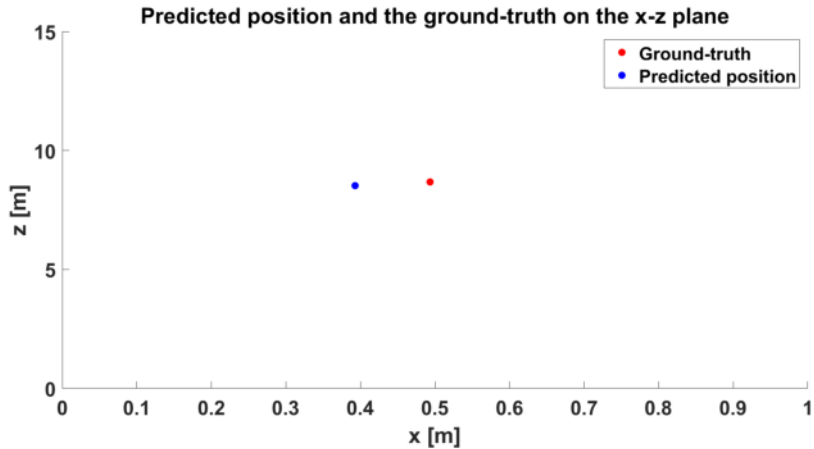


Figure 22: Example of a predicted position and the ground-truth of a pedestrian on the  $x$ - $z$  plane.

Averaging the offsets over all the tracked frames gives an approximation for the deviation of the triangulation results from the ground-truth in terms of the 2D Euclidean distance. In addition, the root mean square error (RMSE) can be determined for the  $x$  and  $z$  coordinates using the values given in the ground-truth for these coordinates. Based on the definition of (Chai et al. 2014), for a total of  $N$  tracked frames, the RMSE for the  $x$  and  $z$  coordinates are given as follows:

$$x_{RMSE} = \sqrt{\frac{\sum_{n=1}^N (x_p - x_{gt})^2}{N}} \quad (46)$$

$$z_{RMSE} = \sqrt{\frac{\sum_{n=1}^N (z_p - z_{gt})^2}{N}} \quad (47)$$

where  $x_p$  and  $z_p$  correspond to the predicted  $x$  and  $z$  coordinates and  $x_{gt}$  and  $z_{gt}$  correspond respectively to their ground-truth coordinates.

## 6 Results

The results obtained during the different stages of implementation of the methodology tested on the KITTI dataset for the sequences mentioned in Section 5 are visualized and analyzed in this section. This includes the results of the Mask-RCNN detections, semi global matching, ground extraction and triangulation, in addition to the final results of the tracking algorithm. The section also includes a discussion of the evaluation metrics (discussed in Section 5) determined for the methodology. The shortcomings of certain implementations and reasons for such eventualities are also pointed out in this section with adequate visualizations. The results obtained in each subsection are interpreted and critically analyzed to form conclusions of the proposed algorithms. Unless otherwise stated, every image presented to show the results of the detection and tracking algorithms belongs to the left camera.

### 6.1 Detection

The detection of pedestrians on the image plane using Mask-RCNN is the first step in the proposed methodology. The quality of its detections, therefore affects the success of all the subsequent stages. As mentioned earlier, the ability of the framework to form segmentation masks of pedestrians even in most cases of overlapping bounding boxes with neighbouring pedestrians is highly advantageous to the matching and tracking algorithms. Figure 23 show the results including the bounding boxes and segmented masks of the Mask-RCNN framework for one frame each from the two test sequences.

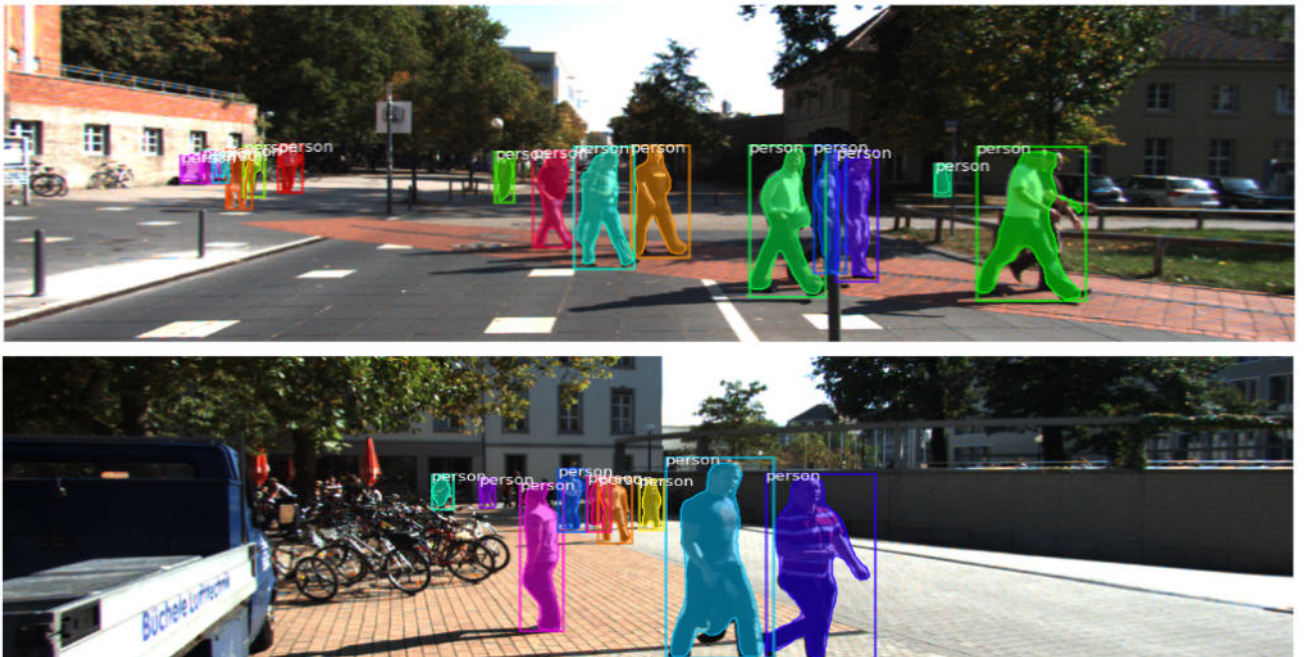


Figure 23: Top: Detection results for a frame in sequence 16. Bottom: Detection results for a frame in sequence 17

From the figures it can be seen how the Mask-RCNN detects and masks out pedestrians in a scene. In some cases, however, it assumes a pedestrian being partially or almost completely occluded by another one as belonging to a single detection as show in Figure 24.

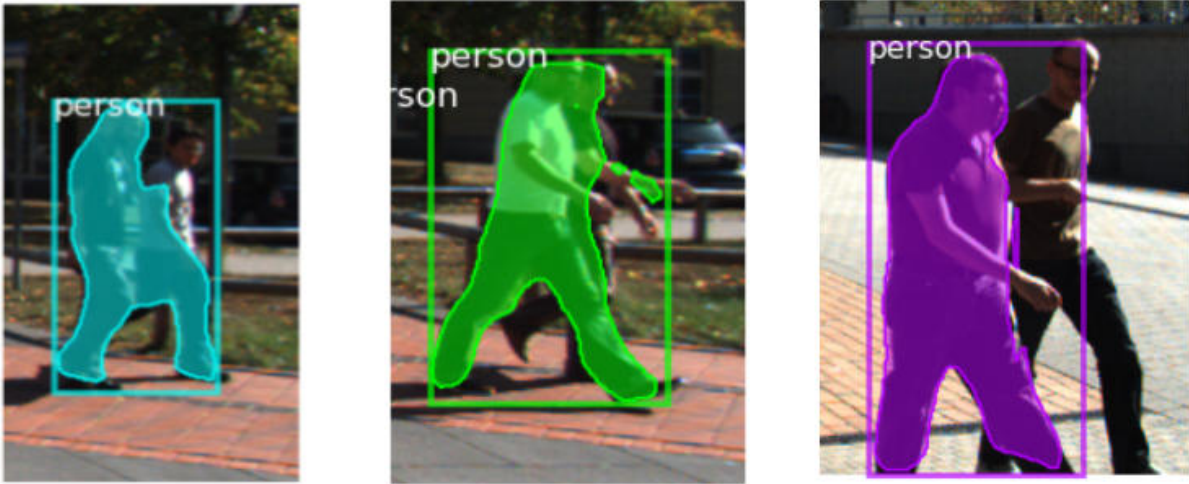


Figure 24: Three cases where two pedestrians were detected as being one due to occlusions

Detection results are also differ due to occlusions between the left and right images in case of large disparities as shown in Figure 25.



Figure 25: Left: Detections for the left frame. Right: Detection for the right frame (frame number: 28, sequence: 16)

It can be seen that when the detector identified the two pedestrians as belonging to two distinct instances in the left image, even when one of them was partially occluded behind the other, it produced a single detection for the two in the right image. Such discrepancies between the left and right images lead to wrong results or no results during triangulation for those identities and in effect, lead to breaks and/or switches in the 3D trajectories.

## 6.2 Dense Stereo Matching and Triangulation

The disparity images for the left and right image pairs determined using semi-global matching are used for triangulation and the ground extraction for each frame in a sequence. Figure 26 shows an example of a disparity image obtained for a frame in the test sequence 16.

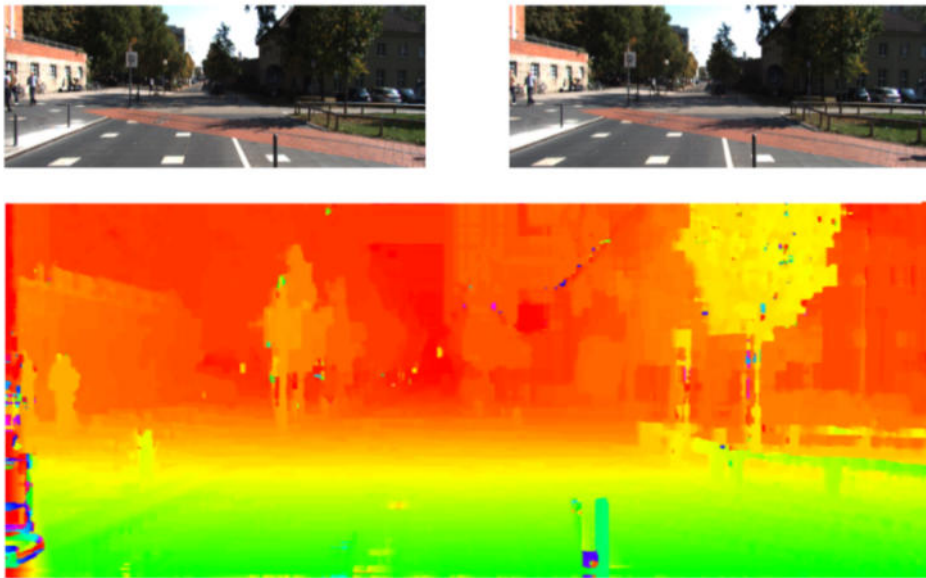


Figure 26: Top: Left and right images of a frame. Bottom: The disparity image for the stereo pair. Pixels with large disparities are given in green color, whereas small disparities are shown in red.

It can be seen that the disparity image is smooth in all areas except where there are sharp discontinuities or boundaries. Such effects are the result of possible occlusions or the algorithm not being able to deal with sharp changes in disparities across boundaries and regions of discontinuities. This could lead to wrong matches of pedestrians in the left images with their conjugates in the right image leading to errors in triangulation, which ultimately give 3D coordinates that are incorrect.

Figure 27 shows two such cases where a pedestrian enclosed by a bounding box on the left image is matched to its conjugate in the right image. Only the disparities of those pixels corresponding to the segmented mask of the pedestrian is used for matching. Matched pixels in the right image are shown as red dots. It can be seen that when the first example produced matches that lay only over the pedestrian in the right image, the

second example shows some pixels that were matched to several background pixels or pixels belonging to other pedestrians.

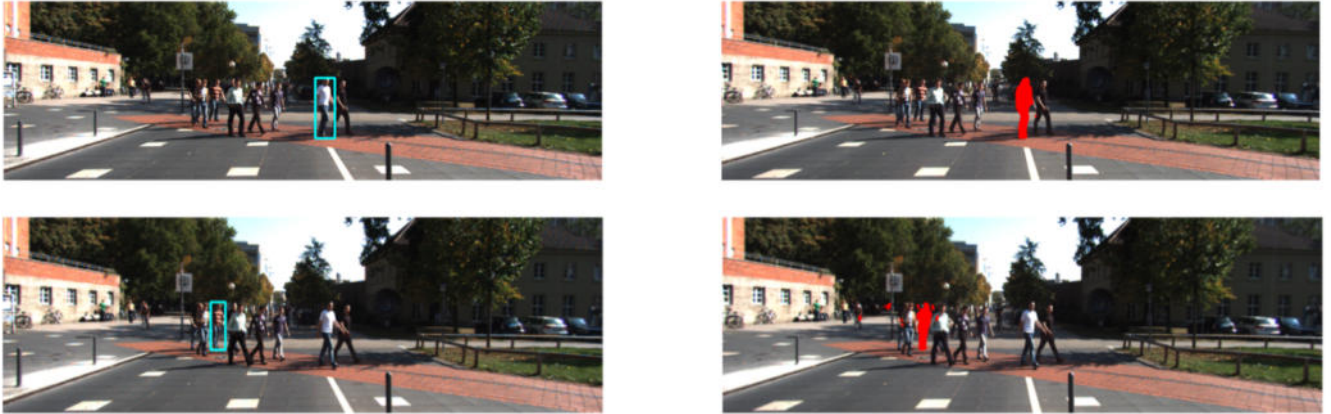


Figure 27: Top: Case in which most pixels on the mask were successfully matched from the left image to the right. Bottom: Case in which stereo matching resulted in points outside the pedestrian.

The triangulation of points matched in the left and right images results in a point cloud in the 3D object space. Figure 28 shows an example of such a reconstruction of a matched pedestrian.

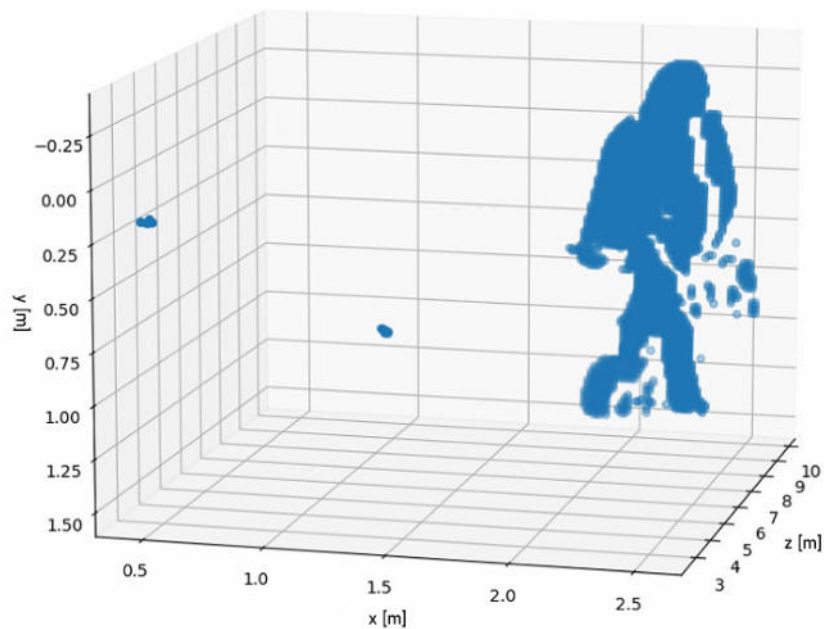


Figure 28: 3D point cloud of a pedestrian obtained after triangulation.

It can be seen that there were several points that were matched to pixels outside the pedestrian leading to outliers. But a high density of points in the region of the pedestrian

leads to an accurate estimate of the representative point, which is used for building the 3D trajectory.

### 6.3 Ground Extraction

Following the steps for extracting the ground plane as explained in Section 4.2, the foot positions of the tracked pedestrians can be determined. The U-disparity and V-disparity images of a disparity image are used for this purpose. Figure 29 shows the V and U-disparity images generated for the disparity image shown in Figure 26.



Figure 29: Left: V-disparity image. Right U-disparity image

Using the obstacle mask calculated from these images, the ground pixels of the left image can be identified. Figure 30 shows the ground pixels identified for the left image. It can be seen how only pixels corresponding to the ground at smaller depths have been identified. A consequence of this is that even though a sufficiently large number of points can be obtained in 3D for a consensus set approximating a plane using RANSAC, the lack of ground points identified further away from the stereo system produces incorrect results for the foot positions of pedestrians when they are at larger depths (around values greater than 20m for the test sequences used).



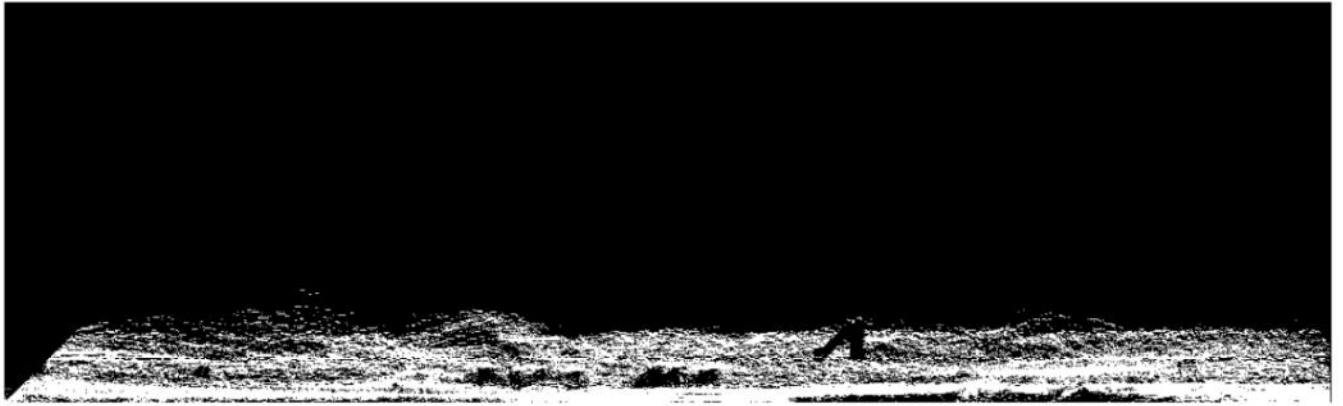


Figure 30: Ground pixels (given in white) identified for the left image

Reconstruction of these pixels in 3D and fitting the points to a plane using RANSAC allows the projection of the 3D pedestrian points on to it, which can then be back-projected on to the image plane to locate the foot positions of the tracked pedestrians. Figure 31 shows pedestrians detected using bounding boxes with the back-projected point indicated as a red dot at the bottom of each box locating the foot positions of the pedestrians.

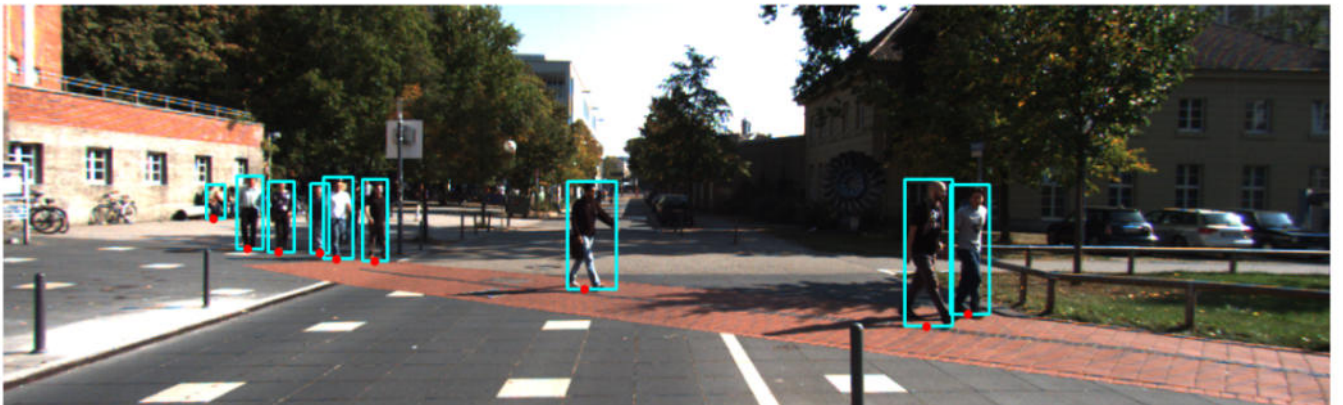


Figure 31: Foot positions identified for each detected pedestrian indicated using red dots.

Predicting the boxes for the next frame using optical flow in combination with this technique allows the refinement of the bounding boxes to enclose the feet of pedestrians a detector may have failed to fit accurately. Figure 32 shows two such cases where the use of the foot positions could improve the bounding boxes.



Figure 32: Left column: Bounding boxes obtained using the Mask-RCNN. Right column: Bounding boxes predicted while tracking.

Even when the detector failed to confine the pedestrians tightly within the bounding boxes due to partial occlusions, the foot positions detected during the tracking algorithm improved the bounding box coordinates.

As it was mentioned earlier the dense matching algorithm and in effect triangulation, have the potential to produce errors with increasing depth from the baseline of the stereo cameras. This in turn also affects the ground extraction and leads to wrong foot positions leading to the bounding boxes not being able to wrap the pedestrian completely or the box enlarging itself beyond the necessary dimensions. Figure 33 shows two such cases where wrong foot positions were back-projected from the 3D object space to produce wrong bounding boxes.

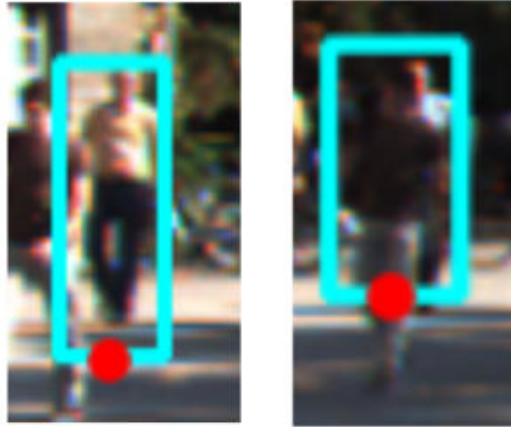


Figure 33: Two cases of pedestrians being incorrectly fit by bounding boxes predicted during tracking due to wrong foot positions.

## 6.4 Optical Flow

The prediction of labels for the upcoming frame using optical flow starts with the detection of reliable points for tracking using the method described in Section 3.5.2. Once the reliable tracking points have been selected for the detected pedestrians, the displacement and direction of their movement between frames can be determined. Figure 34 shows an example of the points suitable for tracking using optical flow selected for a pedestrian.



Figure 34: Points (marked in red) selected for tracking the detected pedestrian using optical flow

Since the region enclosed by a bounding box was used as the search space for this process, the search algorithm is prone to detect points not belonging to the pedestrian if the bounding box does not fit the pedestrian well. This can be seen in Figure 34

where a point on the car that was part of the region enclosed by the box was detected a feature suitable for tracking. Selection of such points on other pedestrians or other entities (including the background) can lead to incorrect flow estimation.

Determining the optical flow as described in Section 3.5 gives the magnitude and direction of the displacement of the selected points between two adjacent frames. Figure 35 visualizes the points detected for the pedestrian shown in Figure 34 for frame  $t_1$  and Figure 36 shows the positions of these points predicted in frame  $t_2$  using optical flow by calculating how much and in which direction they were displaced in between frames.



Figure 35: Reliable points for tracking selected in frame  $t_1$



Figure 36: Predictions of the points in frame  $t_2$

This can be used to predict the bounding box coordinates for the pedestrian in frame  $t_2$ , given his bounding box in  $t_1$ . During the key-frame update, the bounding box coordinates are refined, for example, the width of the boxes are adjusted to accommodate the flow of pedestrians. This step also checks for new detections and assigns new tracks for them. Figure 37 and Figure 38 show two cases where the bounding box coordinates have been updated during a key-frame update.



Figure 37: Left: Bounding box predicted from the previous frame using optical flow. Right: Bounding box refined after the key-frame update.



Figure 38: Left: Bounding box predicted from the previous frame using optical flow. Right: Bounding box refined after the key-frame update.

The foot positions of the tracked pedestrians can be used to form “ground tracks” of pedestrians on the 2D image plane. Figure 39 shows examples of such ground tracks. Each color given to the tracks in the figure corresponds to a different pedestrian.



Figure 39: Examples of ground tracks obtained for tracked pedestrians

## 6.5 Re-identification

The re-identification process is crucial for keeping the consistency of the identities of the tracks under check. Major reasons of missed or switched identities include occlusions, incorrect assignment during the key frame update or change in identity as two pedestrians cross each other or one walks past another. Re-identification using the TriNet model

followed by K-means clustering has been able to fix these problems in several cases. But the resulting tracks still suffer from identity switches. Figure 40 shows a pedestrian who was given an identity of 2 in frame number 5, being occluded behind a cyclist from frame 6 and re-emerging in frame 9 with a new identity of 15. Figure 41 shows how this switch in identity was spotted and the two tracks were merged together and 15 was given as the identity of the track, as it was declared stable by analyzing the later frames.



Figure 40: A pedestrian getting assigned a wrong identity in a later frame due to a missed detection between frames.



Figure 41: The re-identification algorithm resolving the aforementioned switch in identity.

Figure 42 shows, on the other hand, a case where the re-identification method failed to re-identify a pedestrian who was detected before being occluded by a group of two people and detected several frames later with a different identity.

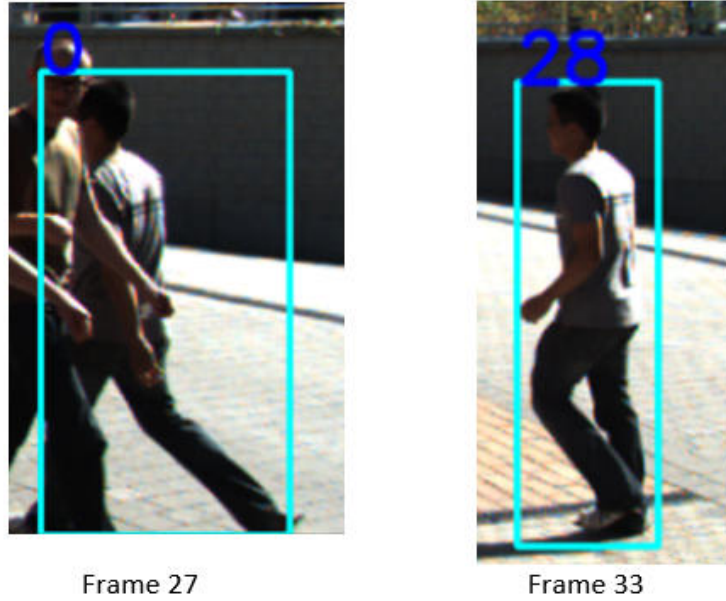


Figure 42: The re-identification algorithm failing to resolve a switch in identity.

It is worth noting that in case of such a scenario, the bounding box enclosing the pedestrian in frame 27, before he was missed includes also the parts of another pedestrian due to a partial occlusion. This results in the inputs being fed to the re-identification model misleading it to make wrong assumptions by forming dissimilar embeddings, eventually leading to different clusters for pedestrians who actually belong to the same identity.

## 6.6 Predicted Trajectories

The tracks obtained from the tracking algorithm on the 2D image plane and the 3D points obtained from triangulation can be used to form 3D trajectories of the tracked pedestrians. Two cases of tracked (for most of the frames) pedestrians, one each from the two test sequences, are discussed in this section. Figure 44 shows the 3D trajectories of a pedestrian from the test sequence 16, with identity 15, obtained using the proposed methodology. Figure 43 shows the pedestrian in frame 0 and later in frame 122 of the sequence for reference. The trajectories shown in Figure 44 follow the centers of gravity obtained for the 3D point clouds and the corresponding foot positions, in every tracked frame.



Figure 43: Top: Pedestrian 15 in frame 0. Bottom: Pedestrian 15 in frame 122

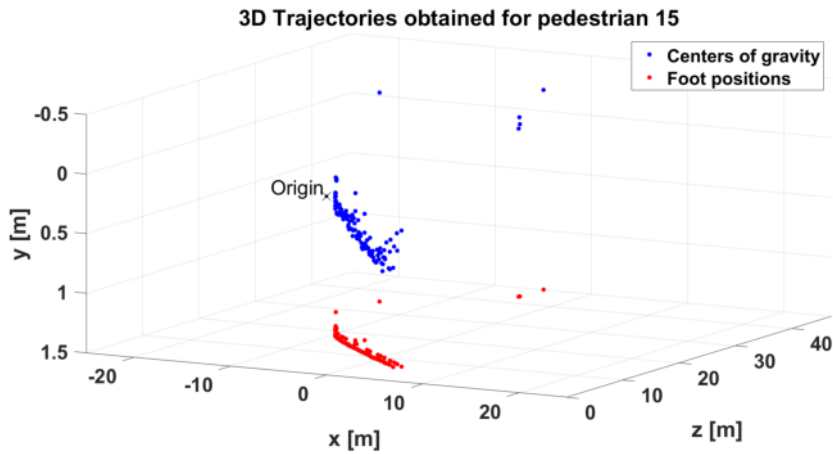


Figure 44: The 3D trajectories for pedestrian 15 following the centers of gravity of the 3D point clouds and the foot positions projected on to the ground plane for every successfully tracked frame.

The origin indicates the position of the projection center of the left camera, based on the definition of the calibration parameters of the stereo system. It can be seen from Figure 44 that the foot positions follow the centers of gravity for every detection. A few outliers can be spotted, especially at larger depths (greater than 20m) suggesting how triangulation failed due to inaccuracies in the dense stereo matching algorithm.

The trajectories can be compared to the ground-truths for the same pedestrian along the  $x$ - $z$  plane. Figure 45 shows the comparison of the tracking results with the ground-truth tracks on the  $x$ - $z$  plane. As mentioned earlier, the  $z$  axis of the coordinate system in the figure is parallel to the optical axes of the stereo cameras.



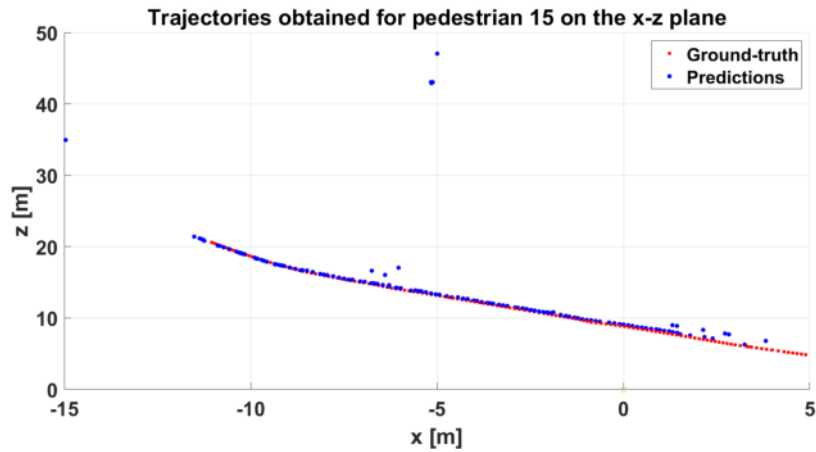


Figure 45: Comparison of the trajectory predicted for pedestrian 15 with the ground-truth positions on the  $x$ - $z$  plane.

Figure 45 shows that the tracking results agree with the ground-truth for most of the frames. Missed detections were due to occlusions as shown in Figure 40 and for smaller  $z$  values, due to the fact that the tracking algorithm was tuned to end tracks when the depth of the tracked pedestrians was less than 2m. The outliers mentioned for the trajectories shown in Figure 44 are also reflected here.

For similar visualizations of another pedestrian from test sequence 17, Figure 46 shows a pedestrian with identity 32 in frame 15 and later in frame 135. Figure 47 shows the trajectories of the centers of gravity and foot positions of the pedestrian in each tracked frame.



Figure 46: Top: Pedestrian 32 in frame 15. Bottom: Pedestrian 32 in frame 135

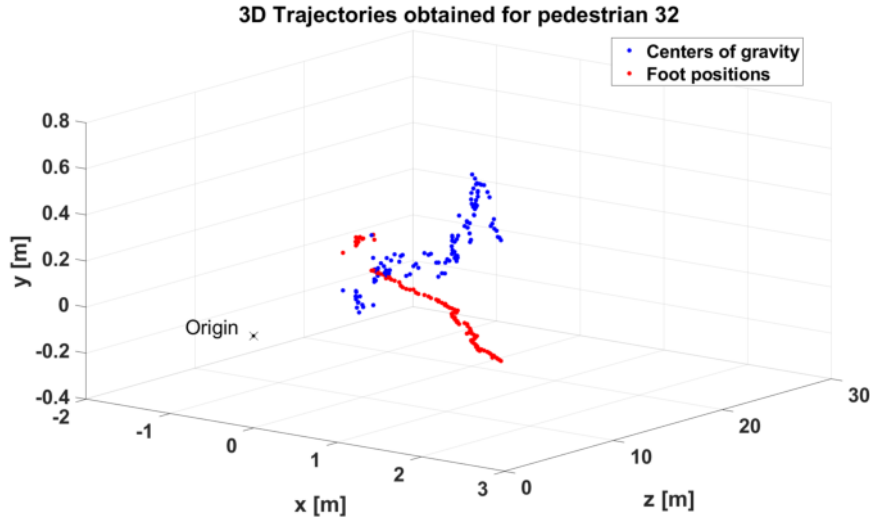


Figure 47: The 3D trajectories for the pedestrian 32 following the centers of gravity of the 3D point clouds and the foot positions projected on to the ground plane for every successfully tracked frame.

Figure 47 shows results of the tracking algorithm in the 3D object space. During the initial frames (when the pedestrian was the furthest away from the stereo cameras), it can be seen that the trajectories of the foot positions and centers of gravity show a strong disagreement (especially along the  $y$ -axis). This disagreement in the trajectories can be attributed to the fact that the ground plane that was fitted using RANSAC was unable to accommodate the foot positions of the pedestrian at depths close to 20m and above. The inlier set obtained from RANSAC approximated a plane which was more suited for pixels at smaller depth values. The absence of pixels identified in the reference image corresponding to the ground, as already mentioned in Section 6.3 and visualized in Figure 30, severely affects the accuracy of the trajectories at such large depth values.

The trajectories are visualized again in Figure 48 and Figure 49 from two different viewing angles, along with the ground plane extracted for the scene. Figure 48 shows how the foot positions followed the centers of gravity of the pedestrian along the ground plane. Figure 49 shows that due to the inaccuracies in stereo matching at larger depths and the fact that the consensus set obtained from RANSAC approximated the plane considering pixels closer to the stereo cameras, the projections of the centers of gravity of the pedestrian at large depths lead to inaccurate results. The centers of gravity falling below the ground plane for several detections at large depth values is a consequence of this problem. It can also be seen how the plane is better approximated for the centers of gravity as the pedestrian walks closer to the cameras.

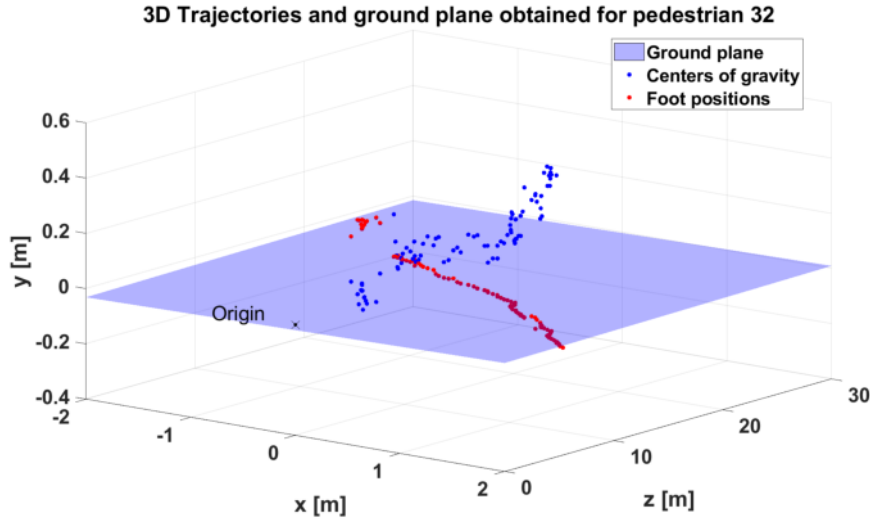


Figure 48: The 3D trajectories for pedestrian 32 following the centers of gravity of the 3D point clouds and the foot positions along the ground plane that was fitted using RANSAC.

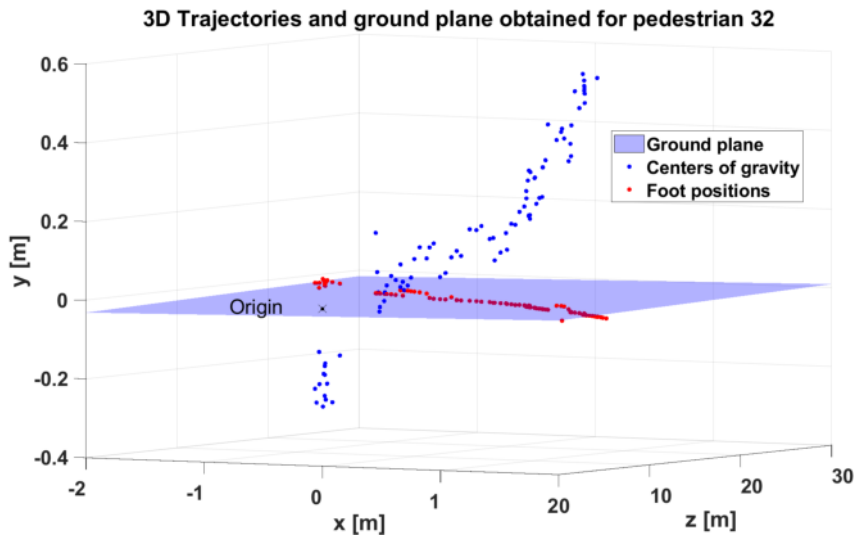


Figure 49: The 3D trajectories for pedestrian 32 and the ground plane viewed from a different angle.

The pedestrian gets missed in several frames due to occlusions from other pedestrians who are closer (and hence appear larger on the image) walking across the scene. The increase and decrease in the positions of the centers of gravity along the  $y$ -axis show how the density of the 3D point cloud varies as the pedestrian moves throughout the sequence.

Figure 50 shows how the tracking algorithm predicted the tracks for pedestrian 32 when compared with the ground-truth positions on the  $x$ - $z$  plane. Gaps in the trajectories can once again be observed for depths greater than and around 20m. This is due to reasons like occlusions from other pedestrians detected closer to the cameras and the tracking algorithm failing at such large depths with limited light conditions during the initial frames of the sequence, as it can be observed from Figure 46.

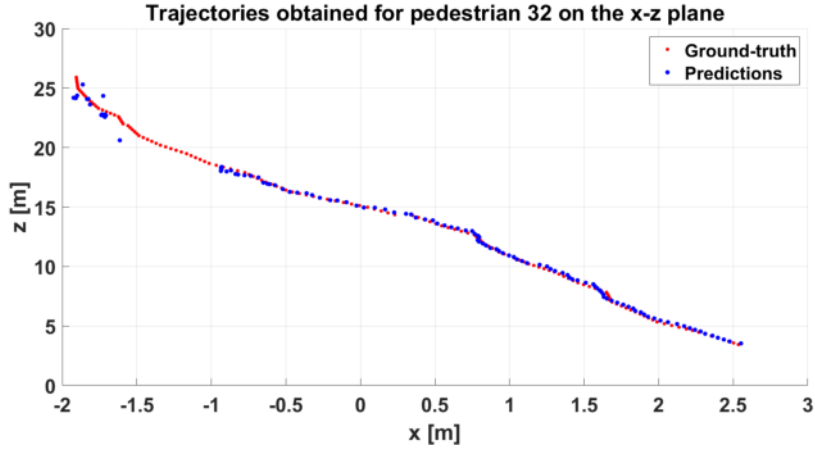


Figure 50: Comparison of the trajectory predicted for pedestrian 32 with the ground-truth positions on the  $x$ - $z$  plane.

## 6.7 Evaluation Metrics

The evaluation metrics mentioned in Section 5.3 were determined to evaluate the efficiency of the detection and tracking algorithms. The metrics for evaluating the pedestrian detections are tabulated in Table 1 for the test sequence 16 and in Table 2 for sequence 17.

Precision	78.36%
Recall	61.51%

Table 1: Metrics evaluating the detections for sequence 16

Precision	66.43%
Recall	73.96%

Table 2: Metrics evaluating the detections for sequence 17

The evaluation indicates a relatively higher precision for the detections in sequence 16 indicating a lower number of FPs as compared to sequence 17. The higher recall value for sequence 17 shows larger number of positives in comparison to sequence 16 leading to a lower precision due to the possibilities of more FPs.

The precision x recall curves of both the sequences were plotted to determine the average precision AP (area under the curve) of the detection framework. Figure 51 and Figure 52 show the precision x recall curves for the detections in sequence 16 and 17 respectively.

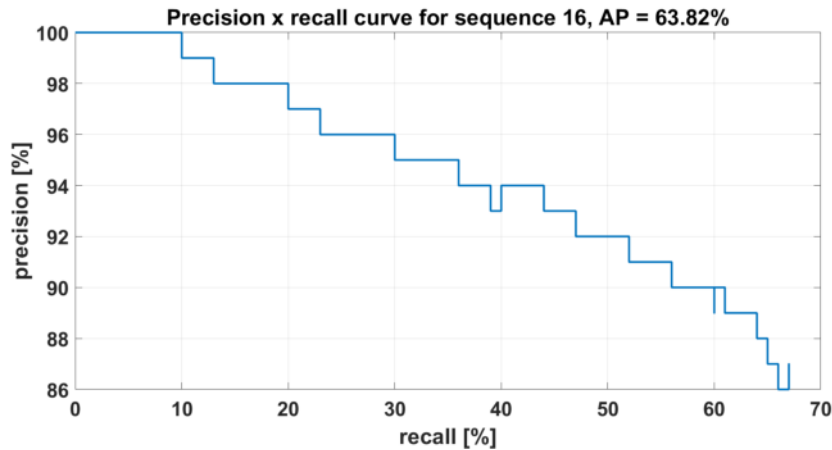


Figure 51: Precision x recall curve for the detections in sequence 16.

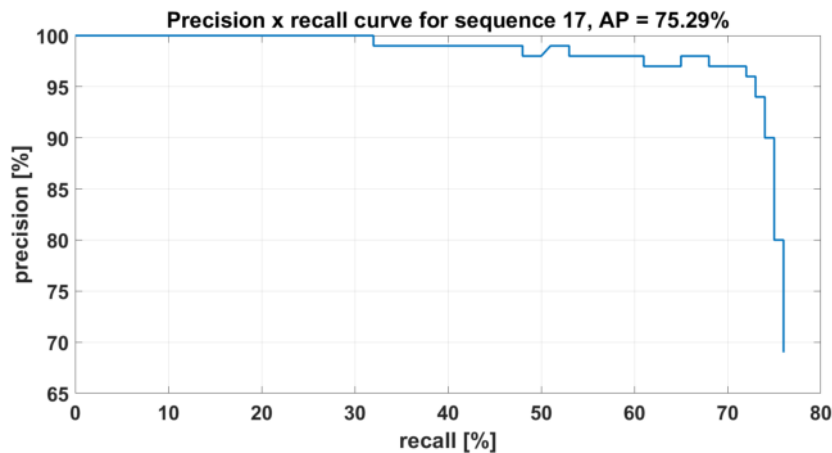


Figure 52: Precision x recall curve for the detections in sequence 17.

The determination of the area under the curve for estimating the average precision ( $AP$ ) resulted in a value of 63.82% for sequence 16 and 75.29% for sequence 17. This indicates better results considering both precision and recall for the latter. The curve in Figure 51 shows a downward trend indicating a decrease in recall values with increasing precision values. This indicates a large number of positive values that have been missed for the sequence. The large number of misses for this sequence could be due to occlusions within small crowds and the fact the tracks for several pedestrians were ended due to the minimum depth (2m) constraint. The curve in Figure 52 shows more stability with high recall values for high precision values, exhibiting a good sign of an object detector, but then has a sharp dip in precision indicating large number of FNs. The detections, which showed better performance in terms of average precision still suffered from problems like pedestrians entering the scene at large depths and under poor light conditions.

It is also worth noting that a careful analysis of the ground-truth provided in the KITTI dataset shows inaccuracies in some frames. Figure 53 shows one such case where the same pedestrian was annotated with two bounding boxes and the neighbouring pedestrian not assigned any. The object detector, however, was able to detect and localize both of

them. Such inaccuracies in annotations can also influence the calculation of the evaluation metrics.



Figure 53: Left: Wrong annotations given in the ground-truth. Right: Boxes obtained by the detection algorithm (frame number: 98, sequence 16)

The tracking algorithm was evaluated by determining MOTA, MOTP and IDF1 scores. These results are tabulated in Table 3 and Table 4 for the sequences 16 and 17 respectively, where an arrow pointing upwards indicates that larger values are desirable, whereas a downward pointing arrow indicates a good performance with lower values.

↑ MOTA	41.36%
↓ MOTP	0.28
↓ IDF1	57.60%

Table 3: MOT metrics for sequence 16

↑ MOTA	34.66%
↓ MOTP	0.24
↓ IDF1	60%

Table 4: MOT metrics for sequence 17

MOTP measures the ability of the algorithm in estimating the position of the objects by comparing the IoU of the detected boxes and the ground-truth averaged over all detections. It therefore, depends on the detection framework involved and not directly on the tracking algorithm. A smaller value of MOTP for sequence 17 shows the relatively larger quantity of error in the detections made for sequence 16.

MOTA gives a measure of the number of missed detections, mismatches in identities and FPs in the tracking algorithm. It can be seen that both sequences suffer from all three issues with the implementation for sequence 17 showing the worst performance of the two. This can be attributed to the relatively larger number of pedestrians entering and leaving

the scene when compared to sequence 16, rendering the tracking and re-identification algorithms incapable of maintaining and re-identifying tracks satisfactorily.

The consistencies of the tracks are measured using the IDF1 score. The value of this metric also points out the improvements that need to be done to the methodology to preserve the identities of tracks and to prevent mismatches or identity switches. The quantitative analysis of the results of the tracking and re-identification algorithms raises the limitations and shortcomings of the proposed methodology which has to be subject to further modifications like the integration of the re-identification scheme to the original tracking workflow or modelling the neighbouring relationships between the tracked pedestrians.

For evaluating the quality of the trajectories obtained, for every frame including the trajectory, the mean deviation of the predicted positions from the ground-truth on the  $x$ - $z$  plane was determined as described in Section 5. For the trajectory obtained for pedestrian 15 in the test sequence 16 (Figure 45) a mean offset of 0.35 m was obtained. This shows that the triangulation results agreed very well with the ground-truth for most frames, despite the outliers. The RMSE for the  $x$  and  $z$  coordinates are given in Table 5.

	RMSE [m]
$x$	0.22
$z$	0.36

Table 5: RMSE of the  $x$  and  $z$ -coordinates obtained for pedestrian 15

Table 5 shows that both the  $x$  and  $z$  coordinates suffered only minor deviations from the ground-truth values. The  $z$  coordinates, giving the depth of the tracked pedestrians suffered slightly larger inaccuracies when compared to the  $x$  coordinates due to inaccuracies in the dense stereo matching algorithm.

Similarly, for pedestrian 32 in the test sequence 17 (with trajectories given in Figure 50), an average deviation of 0.41 m was obtained. This is a slightly worse result when compared to the trajectory obtained for pedestrian 15, showing the difference in the triangulation accuracies for the two sequences as discussed in Section 6.6. The RMSE for the  $x$  and  $z$  coordinates of the trajectory are given in Table 6.

	RMSE [m]
$x$	0.09
$z$	0.47

Table 6: RMSE of the  $x$  and  $z$ -coordinates obtained for pedestrian 32

The RMSE values for pedestrian 32, also show values that agree fairly well with the ground-truth coordinates with only minor deviations. Similar to the previous case, the determination of the depth, given by the  $z$  coordinate shows the largest deviation. This, again indicates the lower performance of the dense matching and in turn, the triangulation algorithms at larger depths.

## 7 Conclusions and Future Work

The experiments carried out as part of this thesis explore several aspects of photogrammetry and computer vision with the help of deep-learning frameworks, to track pedestrians in 3D using stereo images. The competences and flaws of the proposed methodology have been described, visualized and evaluated comprehensively in the different sections. The tracking methodology was tested on the sequences provided in the KITTI dataset and evaluation metrics like MOTA, MOTP and IDF1 were determined. Based on the research going into the proposal of the methodology as well as the experiments carried out on the test data and the analysis and evaluation of the results, several conclusions are drawn and possible extensions and modifications of the proposed methodology and potential research directions in the 3D pedestrian tracking paradigm are described in this section. As far as the objectives of the thesis are concerned, given accurate results on the image plane, the lifting of the tracking mechanism from 2D to 3D was mostly successful with the evaluation showing only minor deviations from the ground-truth whereas, the MOT tracking metrics still showed several possibilities for improvement. Extraction of the ground plane of the scene in combination with the tracking algorithm reliant on optical flow has also proven to improve the localization capabilities of bounding boxes on the images obtained from conventional object detection frameworks, given accurate disparity values. It was also seen how the results of the Mask-RCNN framework and predictions made during tracking complimented each other, with the bounding boxes predicted using optical flow being refined during the key-frame update and the coordinates of the boxes given by the detection stage getting updated by the foot positions of the tracked pedestrians. The inter-dependability of the different stages in the methodology leaves very little margin for error in the final results, as errors in the intermediate steps are inevitably carried over to the subsequent stages.

The results were also highly sensitive to several assumptions and hyper-parameters, like the minimum depth constraint to end tracks, which is something that can be avoided given reliable flow estimations on the image plane, providing more consistent and complete tracks. The availability of segmentation masks for each detected pedestrian can be further utilized in stages like matching pedestrians from the reference image to the matching image, which has shown to produce outliers as given in Section 6.2. These outliers are also carried over to the 3D space after triangulation. Once the segmentation mask of the pedestrian in the matching image has been determined (this can be done, for example, by counting the number of matched pixels that fall into each segmentation mask giving the amount of overlap for each pedestrian and selecting the mask with the maximum overlap as the right mask), each matched pixel can be checked to see if it belongs to the mask. The pixels that fall outside the mask can be considered as outliers and can hence, be discarded. This minimizes the selection of wrong points, especially around boundaries, that are used for triangulation.

Another intermediate step that plays a vital role in the accuracy of the final results is dense stereo matching. As it was discussed, the matching algorithm was prone to errors



with increase in depth of the pedestrians from the baseline of the stereo cameras. The accuracy of disparity estimation has proven to improve when using sub-pixel estimation methods. (Hirschmüller 2005 and Kordelas et al. 2014) propose a quadratic curve fitting method for sub-pixel accuracy in dense stereo matching. The disparity values are calculated based on optimization methods that minimize a cost constraint for possible matches between the left and right images. Sub-pixel estimation involves the fitting of a quadratic curve through the neighbouring costs (given by the next higher or lower disparity) and calculating the minimum. The minimum of this curve gives a disparity estimate with sub-pixel accuracy. Such an estimation can approximate the disparity values in case of large depths and provide better results for triangulation. This enables a more accurate back-projection of the 3D foot positions of the tracked pedestrians on to the image plane improving the 2D bounding boxes.

Even though the technique of extracting the ground plane within a scene using disparity maps has shown to produce results that, in some cases, are better than those given by the object detection algorithm and provides a visualization for the trajectories of the pedestrians following their foot positions, the current methodology is limited to applications where the stereo cameras are stationary, making the ground plane common for every frame. This limits the utilization of the methodology in its present form for applications like autonomous driving, where the cameras are in motion giving rise to a different ground plane for every frame. Although such a scenario, by design, eliminates the possibility of determining ground tracks of tracked pedestrians on a common ground plane using the steps followed in this thesis, further research and experiments can be carried out to modify the method to integrate the ground plane extracted for each new frame into the tracking workflow. This enables the back-projection of the 3D foot position obtained for every pedestrian from a different ground plane in each frame on to the image plane.

It was seen how the use of a pair of stereo cameras enables the lifting of object tracking from the 2D image plane to a 3D object space. The results of the experiments carried out in this thesis show how the determination stereo correspondences followed by ray intersection could refine the 2D detections in case of partial occlusions, when the detections were projected on to the ground plane extracted from the scene. Missed detections due to complete occlusions, however, still persist as a source of error. The possibilities of stereo vision can be extended to multiple views by capturing the scene using multiple pairs of stereo cameras observing from different viewing angles. The detections from each view can be used to form one-to-one matches for all detected pedestrians across all the views using appearance cues and geometrical constraints. Trajectories in 3D can be obtained for a coordinate system with respect to a chosen reference stereo pair or in a common coordinate system obtained for all the pairs. Such a set-up would mean that even when one pair of the cameras failed to detect a particular pedestrian due to complete occlusion behind other pedestrians or obstacles, it is likely to have been captured by at least one of the other pairs observing the scene from another angle, thus contributing to the tracks. (U. D.-X. Nguyen 2020) used a similar approach in her work following

the tracking-by-detection paradigm. Another potential research direction would be to experiment with the sparse reconstruction strategies of the scene with the help of bundle adjustment methods using several partially overlapping images of the scene as described in the works of (Snavely et al. 2008) and (Schönberger 2018). Such a reconstruction of the scene in each frame could provide redundant information for tracking from multiple viewing directions and also help in situations of occlusions as mentioned earlier.

The results shown in Section 6.5 and the calculation of evaluation metrics given in Section 6.7 revealed the existence of identity switches and track inconsistencies using the current implementation. The tracking process also lacks (near) real time capabilities with the current re-identification approach. The possible improvements in tracking results along with better computational times by integrating the re-identification process to the workflow given in Section 4.3 can be investigated. The re-identification method itself can be updated by re-identifying pedestrians in 3D. This is possible, for instance, using the concept of 3D person models to re-identify pedestrians in the 3D space as described by (Z. Zheng et al. 2021). A deep learning model is proposed in the method, which is capable of combining 2D appearance with 3D geometric structure. The model assumes that human beings are rigid 3D objects. Such a human geometry in the 3D space enables the learning of a depth-aware model more robust to real world scenarios. Such a method does not rely solely on 2D information obtained from the image planes, but also shifts the re-identification algorithm to the 3D space. This allows a better utilization of the information that additional viewing directions provide. The involvement of a 3D space also makes the method free of limiting factors like scale and viewpoint. With a stronger re-identification mechanism, the current tracking methodology is expected to produce better results.

## 8 References

- [1] Waleed Abdulla. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN). 2017.
- [2] Saad Ali, Mubarak Shah. “Floor Fields for Tracking in High Density Crowd Scenes”. In: *European Conference on Computer Vision*. 2008.
- [3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, Bernt Schiele. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3686–3693. DOI: 10.1109/CVPR.2014.471.
- [4] H. Harlyn Baker, Thomas O. Binford. “Depth from Edge and Intensity Based Stereo”. In: *International Joint Conference on Artificial Intelligence*. 1981.
- [5] J. Berclaz, F. Fleuret, P. Fua. “Robust People Tracking with Global Trajectory Optimization”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 1. 2006, pp. 744–750. DOI: 10.1109/CVPR.2006.258.
- [6] Jerome Berclaz, Francois Fleuret, Engin Turetken, Pascal Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.9 (2011), pp. 1806–1819. DOI: 10.1109/TPAMI.2011.21.
- [7] Jérôme Berclaz, François Fleuret, P. Fua. Multiple object tracking using flow linear programming. In: *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2009)*, pp. 1–8.
- [8] Philipp Bergmann, Tim Meinhardt, Laura Leal-Taixé. “Tracking Without Bells and Whistles”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 941–951. DOI: 10.1109/ICCV.2019.00103.
- [9] Keni Bernardin, Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. In: *EURASIP Journal on Image and Video Processing* 2008 (Jan. 2008). DOI: 10.1155/2008/246309.
- [10] Christopher M. Bishop, Nasser M. Nasrabadi. Pattern Recognition and Machine Learning. In: *J. Electronic Imaging* 16 (2006), p. 049901.
- [11] Michael D. Breitenstein, Fabian Reichlin, B. Leibe, Esther Koller-Meier, Luc Van Gool. Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011), pp. 1820–1833.
- [12] Zhaowei Cai, Mohammad Saberian, Nuno Vasconcelos. “Learning Complexity-Aware Cascades for Deep Pedestrian Detection”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3361–3369. DOI: 10.1109/ICCV.2015.384.
- [13] Tianfeng Chai, R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? In: *Geosci. Model Dev.* 7 (Jan. 2014). DOI: 10.5194/gmdd-7-1525-2014.
- [14] Tatjana Chavdarova, François Fleuret. *Deep Multi-camera People Detection*. 2017. DOI: 10.48550/ARXIV.1702.04593. URL: <https://arxiv.org/abs/1702.04593>.

- [15] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, Raquel Urtasun. “Monocular 3D Object Detection for Autonomous Driving”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [16] Wongun Choi. “Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3029–3037. DOI: 10.1109/ICCV.2015.347.
- [17] Wongun Choi, Silvio Savarese. “A Unified Framework for Multi-target Tracking and Collective Activity Recognition”. In: *European Conference on Computer Vision*. 2012.
- [18] S. Chopra, R. Hadsell, Y. LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 539–546 vol. 1. DOI: 10.1109/CVPR.2005.202.
- [19] Afshin Dehghan, Shayan Modiri Assari, Mubarak Shah. GMMCP tracker: Globally optimal Generalized Maximum Multi Clique problem for multiple object tracking. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)*, pp. 4091–4099.
- [20] Bin Ding, Huimin Qian, Jun Zhou. “Activation functions and their characteristics in deep neural networks”. In: *2018 Chinese Control And Decision Conference (CCDC)*. 2018, pp. 1836–1841. DOI: 10.1109/CCDC.2018.8407425.
- [21] Shengyong Ding, Liang Lin, Guangrun Wang, Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. In: *Pattern Recognit.* 48 (2015), pp. 2993–3003.
- [22] Andreas Ess, Bastian Leibe, Konrad Schindler, Luc Van Gool. “A mobile vision system for robust multi-person tracking”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587581.
- [23] Xu Fen, Gao Ming. “Pedestrian Tracking Using Particle Filter Algorithm”. In: *2010 International Conference on Electrical and Control Engineering*. 2010, pp. 1478–1481. DOI: 10.1109/iCECE.2010.364.
- [24] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox. *FlowNet: Learning Optical Flow with Convolutional Networks*. 2015. DOI: 10.48550/ARXIV.1504.06852. URL: <https://arxiv.org/abs/1504.06852>.
- [25] Martin A. Fischler, Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Readings in Computer Vision*. Ed. by Martin A. Fischler, Oscar Firschein. San Francisco (CA): Morgan Kaufmann, 1987, pp. 726–740. ISBN: 978-0-08-051581-6. DOI: <https://doi.org/10.1016/B978-0-08-051581-6.50070-2>. URL: <https://www.sciencedirect.com/science/article/pii/B9780080515816500702>.
- [26] Wolfgang Förstner, Bernhard P. Wrobel. “Photogrammetric Computer Vision - Statistics, Geometry, Orientation and Reconstruction”. In: *Geometry and Computing*. 2016.

- [27] T. Fortmann, Y. Bar-Shalom, M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. In: *IEEE Journal of Oceanic Engineering* 8.3 (1983), pp. 173–184. DOI: 10.1109/JOE.1983.1145560.
- [28] Luiz Galvao, Maysam Abbod, Tatiana Kalganova, Vasile Palade, Md Huda. Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—A Review. In: *Sensors* 21 (Oct. 2021), p. 7267. DOI: 10.3390/s21217267.
- [29] Ujwalla Gawande, Kamal Hajari, Yogesh Golhar. “Pedestrian Detection and Tracking in Video Surveillance System: Issues, Comprehensive Review, and Challenges”. In: *Recent Trends in Computational Intelligence*. Ed. by Ali Sadollah, Tilendra Shishir Sinha. Rijeka: IntechOpen, 2020. Chap. 9. DOI: 10.5772/intechopen.90810. URL: <https://doi.org/10.5772/intechopen.90810>.
- [30] Andreas Geiger, Philip Lenz, Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [31] Ross Girshick. “Fast R-CNN”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.
- [32] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81.
- [33] Ross Girshick, Forrest Iandola, Trevor Darrell, Jitendra Malik. *Deformable Part Models are Convolutional Neural Networks*. 2014. arXiv: 1409.5403 [cs.CV].
- [34] Xavier Glorot, Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *International Conference on Artificial Intelligence and Statistics*. 2010.
- [35] Christopher G. Harris, M. J. Stephens. “A Combined Corner and Edge Detector”. In: *Alvey Vision Conference*. 1988.
- [36] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].
- [37] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross B. Girshick. Mask R-CNN. In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: <http://arxiv.org/abs/1703.06870>.
- [38] Kaiming He, Jian Sun. “Convolutional neural networks at constrained time cost”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5353–5360. DOI: 10.1109/CVPR.2015.7299173.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: <https://arxiv.org/abs/1512.03385>.
- [40] Hecht-Nielsen. “Theory of the backpropagation neural network”. In: *International 1989 Joint Conference on Neural Networks*. 1989, 593–605 vol.1. DOI: 10.1109/IJCNN.1989.118638.

- [41] Helbing, Molnár. Social force model for pedestrian dynamics. In: *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics* 51 5 (1995), pp. 4282–4286.
- [42] Alexander Hermans, Lucas Beyer, Bastian Leibe. *In Defense of the Triplet Loss for Person Re-Identification*. 2017. arXiv: 1703.07737 [cs.CV].
- [43] Heiko Hirschmüller. “Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information”. In: *Computer Vision and Pattern Recognition*. 2005.
- [44] Berthold K.P. Horn, Brian G. Schunck. Determining optical flow. In: *Artificial Intelligence* 17.1 (1981), pp. 185–203. ISSN: 0004-3702. DOI: [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2). URL: <https://www.sciencedirect.com/science/article/pii/0004370281900242>.
- [45] Min Hu, Saad Ali, Mubarak Shah. “Detecting global motion patterns in complex videos”. In: *2008 19th International Conference on Pattern Recognition*. 2008, pp. 1–5. DOI: 10.1109/ICPR.2008.4760950.
- [46] Z. Hu, K. Uchimura. “U-V-disparity: an efficient algorithm for stereovision based scene analysis”. In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005*. 2005, pp. 48–54. DOI: 10.1109/IVS.2005.1505076.
- [47] Chang Huang, Bo Wu, Ramakant Nevatia. “Robust Object Tracking by Hierarchical Association of Detection Responses”. In: *European Conference on Computer Vision*. 2008.
- [48] Rabah Iguernaissi, Djamal Merad, Kheireddine Aziz, Pierre Drap. People Tracking in Multi-Camera Systems: A Review. In: *Multimedia Tools Appl.* 78.8 (Apr. 2019), pp. 10773–10793. ISSN: 1380-7501. DOI: 10.1007/s11042-018-6638-5. URL: <https://doi.org/10.1007/s11042-018-6638-5>.
- [49] Hamid Izadinia, Imran Saleemi, Wenhui Li, Mubarak Shah. “(MP) 2 T: Multiple People Multiple Parts Tracker”. In: Oct. 2012. ISBN: 978-3-642-33782-6. DOI: 10.1007/978-3-642-33783-3\_8.
- [50] Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu. *Spatial Transformer Networks*. 2016. arXiv: 1506.02025 [cs.CV].
- [51] Hao Jiang, Sidney S. Fels, J. Little. A Linear Programming Approach for Multiple Object Tracking. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition (2007)*, pp. 1–8.
- [52] Sheng Jin, Wentao Liu, Wanli Ouyang, Chen Qian. Multi-Person Articulated Tracking With Spatial and Temporal Embeddings. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)*, pp. 5657–5666.
- [53] Yonggang Jin, Farzin Mokhtarian. “Variational Particle Filter for Multi-Object Tracking”. In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408952.

- [54] Kiran Kale, Sushant Pawar, Pravin Dhulekar. “Moving object tracking using optical flow and motion vector estimation”. In: *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*. 2015, pp. 1–6. DOI: 10.1109/ICRITO.2015.7359323.
- [55] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME—Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.
- [56] Kiyoshi Kawaguchi. “A multithreaded software model for backpropagation neural network applications”. In: 2000.
- [57] S. Khamis, Cheng-Hao Kuo, Vivek Kumar Singh, Vinay D. Shet, Larry S. Davis. “Joint Learning for Attribute-Consistent Person Re-Identification”. In: *ECCV Workshops*. 2014.
- [58] Andrey Khropov, Anton Shokurov, Victor Lempitskiy, Denis Ivanov. Reconstruction of projective and metric cameras for image triplets. In: (May 2011).
- [59] Chanho Kim, Fuxin Li, Arridhana Ciptadi, James M. Rehg. Multiple Hypothesis Tracking Revisited. In: *2015 IEEE International Conference on Computer Vision (ICCV) (2015)*, pp. 4696–4704.
- [60] Chanho Kim, Fuxin Li, James Rehg. “Multi-object Tracking with Neural Gating Using Bilinear LSTM: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII”. In: Sept. 2018, pp. 208–224. ISBN: 978-3-030-01236-6. DOI: 10.1007/978-3-030-01237-3\_13.
- [61] Georgios A. Kordelas, Petros Daras, Patrycia Klavdianos, Ebroul Izquierdo, Qianni Zhang. “Accurate stereo 3D point cloud generation suitable for multi-view stereo reconstruction”. In: *2014 IEEE Visual Communications and Image Processing Conference*. 2014, pp. 307–310. DOI: 10.1109/VCIP.2014.7051565.
- [62] Louis Kratz, Ko Nishino. Tracking Pedestrians Using Local Spatio-Temporal Motion Patterns in Extremely Crowded Scenes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), pp. 987–1002.
- [63] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In: *Communications of the ACM* 60 (2012), pp. 84–90.
- [64] Nam Do-Hoang Le, Alexandre Heili, Jean-Marc Odobez. “Long-Term Time-Sensitive Costs for CRF-Based Tracking by Detection”. In: *ECCV Workshops*. 2016.
- [65] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. In: *arXiv:1504.01942 [cs]* (Apr. 2015). arXiv: 1504.01942. URL: <http://arxiv.org/abs/1504.01942>.
- [66] Laura Leal-Taixé, Gerard Pons-Moll, Bodo Rosenhahn. “Branch-and-price global optimization for multiview multi-target tracking”. In: 2012.

- [67] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541. eprint: <https://direct.mit.edu/neco/article-pdf/1/4/541/811941/neco.1989.1.4.541.pdf>. URL: <https://doi.org/10.1162/neco.1989.1.4.541>.
- [68] Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep Learning. In: *Nature* 521 (2015), pp. 436–444.
- [69] Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Ng. “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. In: June 2009, p. 77. DOI: 10.1145/1553374.1553453.
- [70] Bohan Li, Yutai Hou, Wanxiang Che. Data augmentation approaches in natural language processing: A survey. In: *AI Open* 3 (2022), pp. 71–90. ISSN: 2666-6510. DOI: <https://doi.org/10.1016/j.aiopen.2022.03.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2666651022000080>.
- [71] Hui Li, Yun Liu, Chuanxu Wang, Shujun Zhang, Xuehong Cui. Tracking Algorithm of Multiple Pedestrians Based on Particle Filters in Video Sequences. In: *Computational Intelligence and Neuroscience* 2016 (Oct. 2016), pp. 1–17. DOI: 10.1155/2016/8163878.
- [72] Wei Li, Rui Zhao, Tong Xiao, Xiaogang Wang. “DeepReID: Deep Filter Pairing Neural Network for Person Re-identification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 152–159. DOI: 10.1109/CVPR.2014.27.
- [73] Youguo Li, Haiyan Wu. A Clustering Method Based on K-Means Algorithm. In: *Physics Procedia* 25 (Dec. 2012), pp. 1104–1109. DOI: 10.1016/j.phpro.2012.03.206.
- [74] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. “Feature Pyramid Networks for Object Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.
- [75] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. *Microsoft COCO: Common Objects in Context*. 2014. DOI: 10.48550/ARXIV.1405.0312. URL: <https://arxiv.org/abs/1405.0312>.
- [76] Jingjing Liu, Shaoting Zhang, Shu Wang, Dimitris N. Metaxas. *Multispectral Deep Neural Networks for Pedestrian Detection*. 2016. arXiv: 1611.02644 [cs.CV].
- [77] Xiaobai Liu. “Multi-View 3D Human Tracking in Crowded Scenes”. In: *AAAI Conference on Artificial Intelligence*. 2016. DOI: <https://doi.org/10.1609/aaai.v30i1.10463>.
- [78] Jonathan Long, Evan Shelhamer, Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].
- [79] Bruce Lucas, Takeo Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)”. In: vol. 81. Apr. 1981.



- [80] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, Bastian Leibe. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. In: *International Journal of Computer Vision* 129 (Feb. 2021), pp. 1–31. DOI: 10.1007/s11263-020-01375-2.
- [81] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Tae-Kyun Kim. Multiple object tracking: A literature review. In: *Artificial Intelligence* 293 (Apr. 2021). DOI: 10.1016/j.artint.2020.103448. URL: <https://doi.org/10.1016%2Fj.artint.2020.103448>.
- [82] Andrii Maksai, Pascal Fua. “Eliminating Exposure Bias and Metric Mismatch in Multiple Object Tracking”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4634–4643. DOI: 10.1109/CVPR.2019.00477.
- [83] J. Chris McGlone. *Manual of Photogrammetry, 6th Edition*. ASPRS, 2013.
- [84] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler. MOT16: A Benchmark for Multi-Object Tracking. In: *arXiv:1603.00831 [cs]* (Mar. 2016). arXiv: 1603.00831. URL: <http://arxiv.org/abs/1603.00831>.
- [85] Dennis Mitzel, B. Leibe. Real-time multi-person tracking with detector assisted structure propagation. In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2011), pp. 974–981.
- [86] Hans P. Moravec. “Obstacle avoidance and navigation in the real world by a seeing robot rover”. In: 1980.
- [87] Arsalan Mousavian, Dragomir Anguelov, John Flynn, Jana Košecká. “3D Bounding Box Estimation Using Deep Learning and Geometry”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5632–5640. DOI: 10.1109/CVPR.2017.597.
- [88] Vinod Nair, Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *International Conference on Machine Learning*. 2010.
- [89] Uyen Nguyen, Franz Rottensteiner, Christian Heipke. CONFIDENCE-AWARE PEDESTRIAN TRACKING USING A STEREO CAMERA. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2019).
- [90] Uyen Dao-Xuan Nguyen. “3D Pedestrian Tracking Using Neighbourhood Constraints”. In: 2020.
- [91] Rafael Padilla, Sergio L. Netto, Eduardo A. B. da Silva. “A Survey on Performance Metrics for Object-Detection Algorithms”. In: *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2020, pp. 237–242. DOI: 10.1109/IWSSIP48289.2020.9145130.
- [92] Sakrapee Paisitkriangkrai, Chunhua Shen, Anton van den Hengel. Learning to rank in person re-identification with metric ensembles. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1846–1855.

- [93] Sankar K. Pal, Anima Pramanik, Jhareswar Maiti, Pabitra Mitra. Deep learning in multi-object detection and tracking: state of the art. In: *Applied Intelligence* 51 (2021), pp. 6400–6429.
- [94] Peixi Peng, Yonghong Tian, Yaowei Wang, Jia Li, Tiejun Huang. Robust multiple cameras pedestrian detection with multi-view Bayesian network. In: *Pattern Recognit.* 48 (2015), pp. 1760–1772.
- [95] Zhen Qin, Christian R. Shelton. Improving multi-target tracking via social grouping. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 1972–1978.
- [96] Yaadhav Raaaj, Haroon Idrees, Gines Hidalgo, Yaser Sheikh. “Efficient Online Multi-Person 2D Pose Tracking With Recurrent Spatio-Temporal Affinity Fields”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4615–4623. DOI: 10.1109/CVPR.2019.00475.
- [97] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV].
- [98] Andreas Reich, Hans-Joachim Wuensche. “Monocular 3D Multi-Object Tracking with an EKF Approach for Long-Term Stable Tracks”. In: *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. 2021, pp. 1–7. DOI: 10.23919/FUSION49465.2021.9626850.
- [99] Vladimir Reilly, Haroon Idrees, Mubarak Shah. “Detection and Tracking of Large Number of Targets in Wide Area Surveillance”. In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, Nikos Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 186–199. ISBN: 978-3-642-15558-1.
- [100] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.
- [101] Brian D. Ripley. “Pattern Recognition and Neural Networks”. In: 1996.
- [102] Mikel D. Rodriguez, Saad Ali, Takeo Kanade. Tracking in unstructured crowded scenes. In: *2009 IEEE 12th International Conference on Computer Vision* (2009), pp. 1389–1396.
- [103] Mikel D. Rodriguez, Josef Sivic, Ivan Laptev, Jean-Yves Audibert. “Data-driven crowd analysis in videos”. In: *IEEE International Conference on Computer Vision*. 2011.
- [104] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological review* 65 6 (1958), pp. 386–408.
- [105] Olga Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.
- [106] D. Scharstein, R. Szeliski, R. Zabih. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. In: *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*. 2001, pp. 131–140. DOI: 10.1109/SMBV.2001.988771.

- [107] Johannes L. Schönberger. “Robust Methods for Accurate and Efficient 3D Modeling from Unstructured Imagery”. In: 2018.
- [108] Florian Schroff, Dmitry Kalenichenko, James Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.
- [109] Jianbo Shi, Tomasi. “Good features to track”. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.
- [110] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily M. Hand, Mubarak Shah. Part-based multiple-person tracking with partial occlusion handling. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (2012)*, pp. 1815–1821.
- [111] Bing Shuai, Andrew G. Berneshawi, Davide Modolo, Joseph Tighe. Multi-Object Tracking with Siamese Track-RCNN. In: *ArXiv abs/2004.07786* (2020).
- [112] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: <https://arxiv.org/abs/1409.1556>.
- [113] Noah Snavely, Steven M. Seitz, Richard Szeliski. Modeling the World from Internet Photo Collections. In: *International Journal of Computer Vision* 80 (2008), pp. 189–210.
- [114] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In: *J. Mach. Learn. Res.* 15 (2014), pp. 1929–1958.
- [115] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, Jan Kautz. *PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume*. 2017. DOI: 10.48550/ARXIV.1709.02371. URL: <https://arxiv.org/abs/1709.02371>.
- [116] Z. Szabo, A. Lorincz. *L1 regularization is better than L2 for learning and predicting chaotic systems*. 2004. DOI: 10.48550/ARXIV.CS/0410015. URL: <https://arxiv.org/abs/cs/0410015>.
- [117] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.
- [118] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Bernt Schiele. Multi-person Tracking by Multicut and Deep Matching. In: *ArXiv abs/1608.05404* (2016).
- [119] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, Bernt Schiele. Multiple People Tracking by Lifted Multicut and Person Re-identification. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 3701–3710.
- [120] Yonglong Tian, Ping Luo, Xiaogang Wang, Xiaoou Tang. “Deep Learning Strong Parts for Pedestrian Detection”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1904–1912. DOI: 10.1109/ICCV.2015.221.

- [121] Amit Satish Unde, Renu M. Rameshan. *MOTS R-CNN: Cosine-margin-triplet loss for multi-object tracking*. 2021. arXiv: 2102.03512 [cs.CV].
- [122] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade. “Three-dimensional scene flow”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 722–729 vol.2. DOI: 10.1109/ICCV.1999.790293.
- [123] P. Viola, W.M. Wells. “Alignment by maximization of mutual information”. In: *Proceedings of IEEE International Conference on Computer Vision*. 1995, pp. 16–23. DOI: 10.1109/ICCV.1995.466930.
- [124] Stefan Walk, Nikodem Majer, Konrad Schindler, Bernt Schiele. New features and insights for pedestrian detection. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 1030–1037.
- [125] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, Jenq-Neng Hwang. Exploit the Connectivity: Multi-Object Tracking with TrackletNet. In: *Proceedings of the 27th ACM International Conference on Multimedia* (2018).
- [126] Haohan Wang, Bhiksha Raj. *On the Origin of Deep Learning*. 2017. DOI: 10.48550/ARXIV.1702.07800. URL: <https://arxiv.org/abs/1702.07800>.
- [127] Kilian Q. Weinberger, Lawrence K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *NIPS*. 2005.
- [128] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, Cordelia Schmid. “DeepFlow: Large Displacement Optical Flow with Deep Matching”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 1385–1392. DOI: 10.1109/ICCV.2013.175.
- [129] Nicolai Wojke, Alex Bewley. “Deep Cosine Metric Learning for Person Re-identification”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2018. DOI: 10.1109/wacv.2018.00087. URL: <https://doi.org/10.1109%2Fwacv.2018.00087>.
- [130] Nicolai Wojke, Alex Bewley, Dietrich Paulus. “Simple online and realtime tracking with a deep association metric”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.
- [131] Zeke Xie, Issei Sato, Masashi Sugiyama. *Understanding and Scheduling Weight Decay*. 2020. DOI: 10.48550/ARXIV.2011.11152. URL: <https://arxiv.org/abs/2011.11152>.
- [132] Bo Yang, Ram Nevatia. “An online learned CRF model for multi-target tracking”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2034–2041. DOI: 10.1109/CVPR.2012.6247907.
- [133] Fan Yang, Wongun Choi, Yuanqing Lin. “Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2129–2137. DOI: 10.1109/CVPR.2016.234.

- [134] Hongkai Yu, Youjie Zhou, Jeff Simmons, Craig P. Przybyla, Yuewei Lin, Xiaochuan Fan, Yang Mi, Song Wang. “Groupwise Tracking of Crowded Similar-Appearance Targets from Low-Continuity Image Sequences”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 952–960. DOI: 10.1109/CVPR.2016.109.
- [135] Min Zhang, Peizhi Liu, Xiaochuan Zhao, Xinxin Zhao, Yuan Zhang. “An obstacle detection algorithm based on U-V disparity map analysis”. In: *2010 IEEE International Conference on Information Theory and Information Security*. 2010, pp. 763–766. DOI: 10.1109/ICITIS.2010.5689679.
- [136] Xuemei Zhao, Dian Gong, Gérard Medioni. “Tracking Using Motion Patterns for Very Crowded Scenes”. In: *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, 2012, pp. 315–328. ISBN: 978-3-642-33709-3.
- [137] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, Qi Tian. “MARS: A Video Benchmark for Large-Scale Person Re-Identification”. In: *European Conference on Computer Vision*. 2016.
- [138] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, Qi Tian. “Scalable Person Re-identification: A Benchmark”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1116–1124. DOI: 10.1109/ICCV.2015.133.
- [139] Zhedong Zheng, Nenggan Zheng, Yi Yang. *Parameter-Efficient Person Re-identification in the 3D Space*. 2021. arXiv: 2006.04569 [cs.CV].
- [140] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, Ming-Hsuan Yang. *Online Multi-Object Tracking with Dual Matching Attention Networks*. 2019. arXiv: 1902.00749 [cs.CV].