



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

A VISION-BASED TEMPORAL CROSS-VIEW SYNCHRONIZATION
FRAMEWORK FOR SPORTS USER GENERATED VIDEOS COMPARING
MOTION AND SEMANTIC REPRESENTATIONS

Elton Dione Nascimento de Alencar

Advisor: Rosiane de Freitas Rodrigues, Prof. D.Sc.

July/ 2026
Manaus - AM

Elton Dione Nascimento de Alencar

A VISION-BASED TEMPORAL CROSS-VIEW SYNCHRONIZATION
FRAMEWORK FOR SPORTS USER GENERATED VIDEOS COMPARING
MOTION AND SEMANTIC REPRESENTATIONS

Dissertation presented to IComp/UFAM as a
partial fulfillment of the requirements for the
degree of Master of Science (M.Sc.)

Advisor: Rosiane de Freitas Rodrigues, Prof. D.Sc.

July/ 2026
Manaus - AM

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

-
- A368v Alencar, Elton Dione Nascimento de
A Vision-Based Temporal Cross-View Synchronization Framework for
Sports User Generated Videos Comparing Motion and Semantic
Representations / Elton Dione Nascimento de Alencar. - 2026.
148 f. : il., color. ; 31 cm.
- Orientador(a): Rosiane de Freitas Rodrigues.
Dissertação (mestrado) - Universidade Federal do Amazonas, Programa
de Pós-Graduação em Informática, MANAUS, 2026.
1. cross-view alignment. 2. multi-object tracking. 3. re-identification
embeddings. 4. self-supervised. 5. tracklet matching. I. Rodrigues, Rosiane
de Freitas. II. Universidade Federal do Amazonas. Programa de Pós-
Graduação em Informática. III. Título
-



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"A VISION-BASED TEMPORAL CROSS-VIEW SYNCHRONIZATION FRAMEWORK FOR SPORTS USER-GENERATED VIDEOS COMPARING MOTION AND SEMANTIC REPRESENTATIONS"

ELTON DIONE NASCIMENTO DE ALENCAR

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof.^a Dr.^a Rosiane de Freitas-Rodrigues - **Presidente**

Prof. Dr. Marco Antônio Pinheiro de Cristo - **Membro Externo**

Prof. Dr. José Luis de Souza Pio - **Membro Externo**

Dr. Jie Yang - **Membro Externo**

Manaus, 29 de Julho de 2026.



Documento assinado eletronicamente por **Rosiane de Freitas Rodrigues, Professor do Magistério Superior**, em 03/08/2026, às 10:18, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marco Antônio Pinheiro de Cristo, Professor do Magistério Superior**, em 12/08/2026, às 12:08, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Jose Luiz de Souza Pio, Professor do Magistério Superior**, em 13/08/2026, às 09:14, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3279303** e o código CRC **9ED81C0E**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroado I Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

Referência: Processo nº 23105.030166/2026-25

SEI nº 3279303

I dedicate this work to my nephews and nieces, Maria Alice, Thiago Emanuel, Alexia Eduarda, Celina, Theo Vinicius, Maitê, and Jefferson Alberto who represent the future I hope Education and Scientific Research will help to build.

ACKNOWLEDGMENTS

First of all, I would like to thank my family, represented here by my grandmother, Dircinha Carmin (*in memoriam*), who has always been the foundation behind my academic/professional achievements. In particular, I would like to express my gratitude to my mother, Gilmara Carmin, whose encouragement, support and many sacrifices made this achievement possible.

I am also grateful to my fiancée and future wife, Emille Amorim, for standing by my side throughout this research, sharing its challenges and achievements, and always being willing to listen, contribute, and support me along the way. Her constant support and presence made this journey considerably more meaningful.

I also would like to thank all my friends who have been there for me during this time. In particular, to Larissa Pessoa, Pedro Matias, and Victória Guimarães, who followed all the research challenges closely and have continued to support me since our undergraduate days.

To my advisor, Rosiane de Freitas Rodrigues, I am sincerely grateful for believing in this research from the beginning, for her valuable contributions to its development, and for constantly encouraging me to pursue the highest quality in our work, particularly throughout our many scientific paper submissions.

Finally, I would like to acknowledge all those who came before me and helped pave the way for me to earn my master's degree. In the same way, I recognize the importance of public policies that support and encourage education and scientific research. Without these initiatives, this journey would undoubtedly have been much more challenging.

*“We can only see a short distance ahead, but we can see plenty there that
needs to be done.”*

Alan Turing

A VISION-BASED TEMPORAL CROSS-VIEW SYNCHRONIZATION
FRAMEWORK FOR SPORTS USER GENERATED VIDEOS COMPARING
MOTION AND SEMANTIC REPRESENTATIONS

Elton Dione Nascimento de Alencar

July/2026

Advisor: Rosiane de Freitas Rodrigues

Abstract: This dissertation introduces Track-UGV-Sync, a search-efficient and object-centric framework for temporal alignment based exclusively on visual features embeddings. The framework combines (i) multi-object tracking to extract significant object trajectories, (ii) visual correspondence using ReID or self-supervised representations, and (iii) robust temporal offset estimation through similarity search and RANSAC (RANdom SAMple Consensus). Its evaluation comprises 531 video pairs from eight sports and three viewpoint conditions. Within the object-centric pipeline, replacing identity-oriented OSNet-AIN embeddings with DINOv2 representations reduced the Mean Absolute Error from 65.33 to 54.80 frames and increased Acc@5 from 8.47% to 13.75%. A complementary comparison with whole-frame representations showed that DINOv2 combined with Dynamic Time Warping achieved the best aggregate visual result, with an MAE of 36.39 frames and Acc@5 of 49.72%. Conversely, object-centric DINOv2 with RANSAC achieved the best performance for same-view pairs, reaching an MAE of 14.78 frames and Acc@5 of 60.87%. These findings indicate that global scene representations are more robust under large viewpoint differences, whereas tracklet representations provide more discriminative temporal evidence when object appearance remains comparable. By restricting the search to frames belonging to significant trajectories, Track-UGV-Sync reduced the number of candidate offset hypotheses from hundreds of thousands to just hundreds. This yields a $93.4\times$ wall-clock speedup.

Keywords: cross-view alignment, multi-object tracking, re-identification embeddings, self-supervised vision transformers, tracklet matching, user-generated videos, video synchronization, visual semantic representation.

Contents

List of Figures	1
List of Tables	6
List of Abbreviations	9
1 Introduction	10
1.1 Research Question and Objectives	13
1.2 Main Contributions and Dissertation Organization	14
2 Theoretical Background	17
2.1 Multi-View User-Generated Videos and its Synchronization	17
2.1.1 Video Synchronization	19
2.2 Video Feature Extraction	21
2.2.1 Convolutional Neural Network	22
2.2.2 Re-Identification Embeddings	23
2.2.3 Vision Transformers and Self-Supervised Representations	25
2.3 Multi-Object Tracking (MOT)	26
2.3.1 YOLO-World: An Open-Vocabulary Object Detector	28
2.3.2 StrongSORT: A tracking-by-detection approach	31
2.3.3 MOT Evaluation	32
2.4 Similarity Metrics Calculation for Visual Features	33
2.4.1 Sequence Alignment Method	35
3 Track-UGV-Sync Overview	36
3.1 Problem Overview and Proposed Framework	36
3.1.1 Mathematical Formalization of the Problem	37

3.1.2	Problem Formulation	42
3.2	Related Work	47
3.2.1	Video Synchronization Based on Visual Features	47
3.2.2	Deep Learning-Based Multi-Object Tracking Algorithms	52
3.2.3	Evaluation Datasets	54
4	Methodology	56
4.1	Research Overview	56
4.2	Methodological Procedures	57
4.2.1	Data Collection and Dataset Construction	58
4.2.2	Datasets and their Role in the Experimental Design	59
4.2.3	Ground-truth synchronization dataset	61
4.2.4	Selection and Implementation of DL-based Models for MOT	64
4.2.5	Operation: UGV Sync Based on Visual Features	65
4.3	Reproducibility	68
5	Framework Setup and Experiments	69
5.1	UGV Dataset Preparation	69
5.1.1	Inclusion Criteria and Selection Procedure	70
5.1.2	I. Collecting and Annotating YouTube UGVs	72
5.1.3	II. Structuring Multi-View YouTube UGVs	75
5.1.4	III. Collecting Multi-View UGVs with Sensor Data	75
5.2	Timestamp-based Temporal Alignment	77
5.3	Integration of MOT Algorithms	78
5.4	Benchmark Evaluation of MOT-UGV	81
5.4.1	Experimental Setup	81
5.4.2	Models and Architectures	83
5.4.3	Computational Resources	84
5.5	Experiments on Tracklet-Based Synchronization	84
5.5.1	Ground-Truth Evaluation Set for Quantitative Analysis	86
5.5.2	Multi-View Synchronization Experiment: Visual Representation	87
5.5.3	Error Analysis Experiment	89
5.5.4	Generalization Experiment on CMU Panoptic	90

6	Results	91
6.1	Development Outcomes	91
6.1.1	Created Datasets and Its Structure	91
6.1.2	MOT Technique: StrongSORT & YOLO-World	98
6.1.3	MOT-UGV: Quantitative Results	102
6.2	MOT-UGV-based Synchronization	105
6.2.1	Chosen Similarity Metric: Euclidean Distance	106
6.2.2	Synchronization based on visual feature correspondence	108
6.3	Multi-View UGV Synchronization	114
6.3.1	UGV-Sync: Custom Annotated Dataset	115
6.3.2	Offset Estimation Strategies	117
6.3.3	Comparison of Visual Representations	122
6.3.4	Representational Analysis of the Cross-View Failure	126
6.3.5	Comparison with VideoSync and Whole-Frame Representations	127
6.3.6	Search Space Reduction	131
7	Conclusions	133
7.1	Future Works	135
7.2	Publications, Datasets, and Tools	136
7.2.1	UGV-Sync Studio: Open-source software	137
	References	138

List of Figures

2.1	An illustration of the multi-camera view setup within a sports stadium, where 4 users record videos from different perspectives.	18
2.2	OSNet-AIN architecture used as the ReID feature extractor: a 256×128 object crop is processed through stages <code>conv1–conv5</code> into a 512-dimensional L2-normalized embedding [Zhou et al. 2022].	24
2.3	DINOv2 (ViT-B/14) Architecture: each input image is cropped (5% padding), resized to 224×224 and split into 256 patches; the Transformer encoder processes them and the [CLS] token is taken as a 768-dimensional, L2-normalized holistic embedding of the crop. [Oquab et al. 2023].	26
2.4	General workflow of a tracking-by-detection paradigm, as StrongSORT, broken down into five main stages: (I) Object Detection, (II) ReID, (III) Data Association, (IV) Motion Prediction, and (V) Track Management.	27
2.5	General overview of an CNN-based object detection process.	28
2.6	Overview of the YOLO-World architecture. (a) YOLOv8 backbone integrated to it for visual feature extraction [Jocher et al. 2023]. (b) Open-vocabulary detection pipeline combining image multi-scale features with CLIP-based text embeddings to infer the bounding boxes and labels [Cheng et al. 2024].	30
2.7	General overview of the processes in tracking-by-detection approach: object detection, Re-identification, data association, motion prediction, array of tracks.	31

3.1	Diagram workflow of the proposed framework for synchronizing multi-view UGVs. In Stage I, both videos are tracked by MOT-UGV (1) and only the most significant trajectories are selected (3). In Stage II, the significant trajectories of V_2 are used as the index and those of V_1 as queries (4); a best visual representation is extracted (5); the similarity is computed to find the most semantically similar frame pairs (6); the temporal offset is estimated by RANSAC inlier voting (7); and the overlap detection enforces a single offset over the whole sequence, producing the synchronized fragments (8).	46
4.1	Applied Research: general Overview of the processes present in the methodology adopted for synchronizing user-generated videos based on deep learning-based MOT techniques.	58
4.2	Overview of the processes present in the data collection and dataset construction phase. (I) Data Generation, (II) Data Collection, (III) Pre-Processing.	58
4.3	Synchronized frame pairs from the UGV sports evaluation set, with bounding-box annotations produced by the MOT pipeline. Each column shows the temporally aligned frames from two views at the manually annotated ground-truth offset.	62
5.1	Workflow of the three dataset creation: (I) YouTube UGV for MOT, (II) Multi-View YouTube UGV, and (III) Multi-View with Sensors.	70
5.2	Dataset construction pipeline for the UGY dataset. The pipeline illustrates the process from video collection, followed by metadata extraction and frame extraction, automated annotation using YOLO-World and StrongSORT, and manual refinement via CVAT. The resulting dataset is structured in MOT format.	72
5.3	Illustration of the Multi-View UGV Dataset with Sensor Data construction.	76
5.4	Example of the timestamp-based temporal alignment for one event, including the overlap point, where at least two cameras were simultaneously active.	78

5.5	Confirmation of the successful implementation and adaptation of the execution script for the baseline tracking-by-detection algorithm, StrongSORT.	80
5.6	Successful implementation and adaptation of the execution script for the baseline tracking-by-attention algorithm, TrackFormer.	80
5.7	Experiment reproduced for the same video: <code>SOURCE_VIDEO1_PATH</code> and <code>SOURCE_VIDEO2_PATH</code> received the same video file.	86
5.8	Video 2 is a cut of the Video 1. The highlighted part is where both videos present the same moment.	86
6.1	Summary of the total videos and its distribution for the UVY.	92
6.2	Example of manual annotation refinement using CVAT. (1) Initial detection by YOLO-World; (2) Corrected bounding box via CVAT; (3) Frame with ID-switch issue where the identity was incorrectly transferred to a different person.	94
6.3	MUVY Stats: summary of the total videos and its distribution.	96
6.4	MUVY Structure Overview: each video is accompanied by a <code>video_</code> -info file, an extracted frames directory, and an object annotations file.	97
6.5	Summary of the total videos and its distribution for the MUVS Dataset.	97
6.6	Overview of the Data Structure in MUVS, Motorola’s Custom Dataset.	98
6.7	MOT Implemented - Output Videos: (a) DeepSORT (YOLO-v5), “sports ball” class detected; (b) TrackFormer result, no ball detected; (c) StrongSORT result, detected the “sports ball” class and improved tracking of multiple objects, including a person positioned near the goal area.	99
6.8	The video processed by DeepSORT presented id switching problems during the process of data association.	100
6.9	StrongSORT results: on the left side YOLOv7 detecting several people, including those in the audience; on the right, YOLO-World detecting only people playing on the court.	101

6.10	StrongSORT results: on the left YOLOv7 detected an average of 20 objects per frame. On the right, YOLO-World detected about 5 objects per frame, focusing mainly on those in the field.	101
6.11	Euclidean distances heatmap, illustrating the variation of the pairwise tracklet comparison.	106
6.12	Cosine Similarity heatmap, illustrating the variation of the pairwise tracklet comparison. All values very close to 1.00.	107
6.13	Mean Squared Error Values Heatmap, illustrating the variation of the pairwise tracklet comparison. Less robust in distinguishing fine differences.	108
6.14	Tracklet Similarity Heatmap: illustrating a diagonal with zero values, which indicates the exact correspondence among tracklet features. . .	109
6.15	Frame-Level Similarities Heatmap: pairwise comparison among frame-level features extracted by two video with the same content, highlighting a diagonal with zero values.	110
6.16	Offset Estimation Result: pairwise comparison of two overlapped frames.	111
6.17	Tracklet Similarity Heatmap: (a) all tracklets features vector from video 1 pairwise compared to the ones from video 2; (b) objects detected and tracked only in video 1; (c) tracklet 22 (video 1) is most similar to the tracklet 3 (video 2); (d) sample of the frame illustrating the object detected for the Tracklet 22 and 3.	112
6.18	Side-by-side comparison of the frames that has the lowest similarity score, detected automatically by the script.	112
6.19	Frame-Level Similarity Heatmap highlighting a diagonal pattern indicating the existence of similar features in the frames.	113
6.20	Synchronized video fragments generated automatically based on the visual features extracted by the MOT-UGV.	114
6.21	Screenshot of the custom UGV-Sync Studio, the tool used to support the validation of synchronized video pairs.	116

6.22	Synchronization accuracy as a function of the tolerance threshold τ for OSNet-AIN and DINOv2, under the best-match and the RANSAC strategies. The RANSAC strategy consistently improves upon the best-match approach, particularly for OSNet-AIN, corroborating the theoretical prediction of Theorem 1.	118
6.23	Synchronization accuracy as a function of the tolerance threshold τ for OSNet-AIN and DINOv2, under the best-match and the RANSAC strategies. The RANSAC strategy consistently improves upon the best-match approach, particularly for OSNet-AIN, corroborating the theoretical prediction of Theorem 1.	119
6.24	UMAP projections of OSNet-AIN tracklet embeddings for all six UGV evaluation pairs (tracklets < 15 frames excluded). Cyan = Video 1; magenta = Video 2.	121
6.25	Synchronization accuracy across tolerance thresholds. The timestamp curve is repeated in both panels because it does not depend on the visual offset estimator.	124
6.26	Mean Absolute Error per sport for OSNet-AIN and DINOv2 (best-match strategy).	125
6.27	Accuracy at a 5-frame tolerance per sport for OSNet-AIN and DINOv2 (best-match strategy).	125
6.28	Two-dimensional projection of the tracklet embeddings of a cross-view pair for OSNet-AIN and DINOv2, coloured by source video. The two videos remain more separated for OSNet-AIN and more interleaved for DINOv2.	126
6.29	Example of a DV-Far UGV pair recorded at the same elevation as the players (field-level), shown in the UGV-Sync Studio. The bottom panel reports the per-pair metrics (ground-truth offset=9 and the OSNet-AIN / DINOv2 estimates under each estimator).	130
6.30	Example of a DV-Far UGV pair recorded at an angle that looks down onto the scene (high-level), shown in the UGV-Sync Studio.	131
7.1	UGV-Sync Studio interface for side-by-side inspection across multi-sports scenarios, offset refinement, and comparison of sync metrics.	137

List of Tables

3.1	Summary of works about visual-based video synchronization and its relation to UGVs and MOT features.	49
3.2	Summary of works about DL-based approaches applied to video synchronization and its relation to UGVs and MOT features.	50
3.3	List of works about multi-object tracking and its relation to multi-view user-generated videos.	52
3.4	Summary of works that introduce a dataset to MOT or Synchronization evaluation and its relation to Multi-View UGVs.	55
5.1	Configurations for StrongSORT+YOLO-World/L	82
6.1	Comparison among public datasets used to evaluate MOT algorithms (including UVY)	93
6.2	MOT Benchmark: proposed pipeline compared to the baseline model on MOT17	103
6.3	MOT Benchmark: proposed pipeline compared to the baseline model on MOT20	103
6.4	StrongSORT parameter tuning on MOT20	104
6.5	Composition of the UGV-Sync dataset by source.	117
6.6	Composition of the UGV-Sync dataset (513 pairs) per sport.	117
6.7	Synchronization results on 6 UGV sports evaluation set. GT is the ground-truth temporal offset in frames (positive = Video 1 leads Video 2). Rec. is the RANSAC-recommended offset. $ MAE $ is the absolute error in frames (lower is better). IR is the RANSAC inlier ratio.	120

6.8	Consolidated synchronization error on the 531 evaluated pairs. MAE is reported in frames and Acc@5 is the percentage of estimates within five frames of the validated offset.	123
6.9	Per-sport results on MUVY-Sync set (best-match strategy). MAE in frames; Acc@5 in percentage. The cricket and hockey pairs are same-view trims (see Subsection 6.3.1).	125
6.10	Overall comparison of object-centric and whole-frame synchronization conditions on the 531 UGV-Sync pairs.	128
6.11	Best-performing visual conditions by viewpoint class.	128
6.12	Empirical search-space reduction of the tracklet-based formulation on the MUVY evaluation set.	132
7.1	Publications and submissions derived from this dissertation.	136

List of Abbreviations

ANN	Approximate Nearest Neighbor
CLIP	Contrastive Language-Image Pre-training
CNN	Convolutional Neural Network
DINO	self-Distillation with NO label
DL	Deep Learning
DTW	Dynamic Time Warping
GPS	Global Positioning System
KLT	Kanade–Lucas–Tomasi feature tracker
MFCC	Mel-Frequency Cepstral Coefficients
MOT	Multi-Object Tracking
MOT-UGV	Multi-Object Tracking Approach Applied to Sports User-Generated Videos
MOTS	Multi-Object Tracking and Segmentation
MS COCO	Microsoft Common Objects in Context Dataset
MUVS	Multi-view User-generated Videos and Sensor Data
MUVY	Multi-view User-generated Videos from YouTube
OSNet	Omni-Scale Network
RANSAC	Random Sample Consensus

ReId	Re-Identification
ResNet	Residual Networks
SIFT	Scale Invariant Feature Transform
SORT	Simple Online and Realtime Tracking
SWPERFI	Artificial Intelligence Techniques for Software Performance Analysis and Optimization Project
UGV	User-Generated Video
UVY	User-generated Videos from YouTube
ViT	Vision Transformer
YOLO	You Only Look Once

Chapter 1

Introduction

Video content, a multimodal structure, is considered a rapid means of sharing information, surpassing traditional combinations of text and images. This coincided with the rapid expansion of video-sharing platforms over the past few years. This growth led to an exponential increase in video production, with millions of videos created every day [Tang et al. 2023]. In addition, most of these sharing platforms (social media) have mobile devices (such as smartphones) as their main source of data generation and consumption, mainly because smartphones have built-in cameras capable of capturing videos quite quickly [Zhang et al. 2015, Wang et al. 2023].

Most of these videos are classified as User-Generated Videos (UGVs), once they are recorded by people with mobile hand-held cameras [Segundo et al. 2017]. When multiple UGVs are recorded at the same event by different users, a set of multi-view videos is generated, for example, multiple videos from various perspectives recorded by users in different positions inside a soccer stadium [Pessoa et al. 2024]. In uncontrolled environments, these videos are often unsynchronized, mainly because the devices that record them are not connected [Guggenberger 2023].

The synchronization of UGVs can contribute to an overall understanding of the recorded scene, especially for highlight moments in a sport recording, such as a goal or a foul [Held et al. 2023]. One practical example is that after the synchronization, these videos can also be used to create a single video that thoroughly covers the event by merging all synced videos (i.e., mashups), allowing viewers to re-watch the match with a multi-perspective view of the recorded event [Alencar and de Freitas 2026a].

Manually processing such a large volume of video content to achieve synchronization is labor-intensive and time-consuming. There is a growing demand for tools that can effectively manage, analyze, and process this abundance of video content [Tang et al. 2023]. Developing tools to automate the identification and extraction of elements composing the scenes is necessary and would optimize part of the video content analysis process [Alencar et al. 2022]. This would also contribute to automatically extracting visual features to be further used in video synchronization [Mei et al. 2023].

To meet the demand mentioned above, deep learning-based video understanding methods have emerged, allowing automatic recognition and interpretation of video content, which reduces the need for manual work [Tang et al. 2023]. One of these methods is Multi-Object Tracking (MOT), which is an approach used to understand video content and, consequently, describe scenes by detecting and tracking objects in the scene. This is done mainly through deep learning-based techniques [Alencar and de Freitas 2025, Alencar et al. 2025].

Furthermore, MOT is critical for video analysis, as it tracks entities across video frames [Amosa et al. 2023]. This capability contributes to the automation of video content understanding and enables a variety of feature-based tracking analysis [Alencar et al. 2025], making it useful in different fields of knowledge [Alencar and de Freitas 2025], such as sports video analysis [Rangasamy et al. 2020], action recognition [Alencar et al. 2022], video summarization, video synchronization [Whitehead et al. 2005], among others.

In appearance-assisted tracking-by-detection methods such as StrongSORT, detections are associated using motion, geometry, and appearance cues [Du et al. 2023]. By this association, we obtain tracklets, formed by linking detections across adjacent frames to construct object trajectories. By representing object movement in compact form, tracklets enable localized reasoning over time and serve as base units for more complex operations such as track merging and behavior analysis. Because they preserve both spatial and temporal features, tracklets can be used to identify visual consistency and overlapping content across different video segments [Bewley et al. 2016].

From the StrongSORT pipeline, the appearance cues used to build and associate

tracklets are produced as re-identification (ReID) embeddings [Du et al. 2023]. These embeddings are trained to discriminate object identity across frames and cameras; that is, they are optimized to determine whether two image crops correspond to the same person [Zhou et al. 2022]. However, the problem of temporal synchronization is distinct from re-identification. In synchronization, the question is not whether two tracklets belong to the same object, but whether they indicate the same temporal moment of an event recorded from different views [Alencar and de Freitas 2026a].

For this reason, ReID embeddings are useful, but not necessarily the most adequate representation for cross-view temporal alignment. This observation motivates investigating which type of visual representation better supports the synchronization task, comparing identity-specialized ReID embeddings [Zhou et al. 2022] with holistic self-supervised representations learned by Vision Transformers [Caron et al. 2021, Oquab et al. 2023]. Candidate tracklet correspondences can be retrieved through nearest-neighbor search (e.g., ANN) in the embedding space [Guo et al. 2026]. The synchronization addressed in this work follows the principle of comparing embeddings to find similar content.

However, the main difference is that for video synchronization requires these individual correspondences to agree on a common temporal offset. The matches must also be consistent in time along the sequence. For this reason, synchronization is closer to a sequence alignment problem than to a retrieval or a visual semantic-logical search problem [Guo et al. 2026].

This intuition is also supported by Shin and Molybog 2025: when general-purpose Vision-Language Models (VLMs) are used directly to predict the temporal offset between two videos, they perform no better than a naive guess. This indicates that strong semantic understanding alone does not solve the temporal alignment, and that temporal consistency along the sequence is what really matters. This relation is used here only to help the understanding of the problem and is revisited with more detail in the analysis of the results.

1.1 Research Question and Objectives

Considering the above, this work introduces a framework to synchronize user-generated videos using object-tracking features extracted by a pre-trained deep learning-based MOT pipeline and to investigate which visual representation better supports cross-view temporal alignment. To this end, this work aims to address the following research question:

Research Question: How effectively can multi-view sports user-generated videos be temporally synchronized using only visual representations extracted from tracked objects, without relying on timestamps, audio, sensor metadata, or camera calibration?

Considering what was introduced before and in order to answer the formulated research question, we present the main and specific objectives defined for this dissertation.

Main Objective: develop and evaluate a vision-based framework for estimating the temporal offset between pairs of multi-view sports UGVs using visual representations extracted from tracked objects. The proposed approach detects and tracks relevant sports objects, represents their trajectories as tracklets, retrieves candidate cross-video correspondences through embedding similarity, and estimates the final offset by identifying temporally coherent matches. The study also evaluates how the choice of visual representation and the difference between camera viewpoints affect synchronization accuracy.

To achieve this, the following **specific objectives** are defined:

- Collaboratively build custom datasets of multi-view UGVs recorded during sports events for fine-tuning and evaluation.
- Build an annotated validation set with ground-truth synchronization offset to provide a set of video pairs with manually annotated offset, across different sports, to enable the quantitative evaluation of the synchronization.
- Adapt existing deep learning-based multi-object tracking techniques to specifically track sports objects (e.g. player in the field, the ball) in user-generated videos.

- Propose an approach for pairwise comparison of tracklets across frame sets to identify overlapping scenes by implementing embedding-based similarity search for matching visual features.

1.2 Main Contributions and Dissertation Organization

Given the research questions and objectives presented above, the main contributions of this dissertation are as follows.

1. The proposal of Track-UGV-Sync, a timestamp-free synchronization framework designed to estimate the temporal offset between pairs of multi-view sports UGVs using only visual information extracted from each visual content, without relying on timestamps, sensor metadata, audio signals, or camera calibration [Alencar and de Freitas 2026a].
2. A custom adapted object tracking algorithm, named MOT-UGV. It is used to detect and track objects of interest (e.g., players, ball) in sports UGVs. MOT-UGV stage combines open-vocabulary object detection (YOLO-World) with a tracking by detection method (StrongSORT) to generate tracklets that represent the movement of relevant objects in the scene [Alencar and de Freitas 2025, Alencar et al. 2025]. These tracklets are then used as the basis for the synchronization process, reducing the need for exhaustive frame-by-frame comparison.
3. Beyond proposing a synchronization pipeline, this research presents a comparative analysis of visual representations for cross-view temporal alignment. This is done by investigating which type of visual representation better supports the synchronization of multi-view sports UGVs. In particular, it compares identity-specialized ReID embeddings, originally produced by the tracking model, with holistic self-supervised Vision Transformer representations.
4. The proposed framework introduces a tracklet-matching and offset estimation procedure based on visual similarity, temporal coherence, and RANSAC-based

inlier voting. This strategy improves the robustness of the synchronization process by reducing the influence of low-quality matches and outliers, allowing the final offset to be selected from the most temporally consistent visual correspondences.

5. This dissertation contributes to the creation and organization of UGV sport datasets. They are UVY (i.e., a dataset of YouTube sports User-generated videos for MOT evaluation [Alencar and de Freitas 2026b]), MUVY (i.e., YouTube Multi-view User-generated Videos [Pessoa et al. 2026a]) and MUVS (i.e., YouTube Multi-view User-generated Video and Sensor data [Pessoa et al. 2026b]). These datasets are not only used as input data, but they also define the experimental conditions under which the comparison of visual representations is performed, since they reproduce the uncontrolled viewpoints, framings and recording conditions that characterize user-generated content. In addition, this work introduces UGV-Sync, a custom annotated evaluation dataset composed of UGV pairs with manually defined ground-truth offsets. UGV-Sync is what makes the central comparison of this dissertation measurable, because without annotated offsets the alignment produced by each representation could only be inspected qualitatively, and metrics such as MAE and $\text{Acc}@7$ could not be computed.
6. This research also introduces the UGV-Sync Studio, a visual inspection and annotation tool created to support the analysis of video synchronization results. The tool allows video pairs to be displayed side by side, together with their estimated or annotated offset, making it possible to visually inspect the behavior of the proposed framework pipeline. Beyond inspection, the UGV-Sync Studio is the instrument through which the ground-truth offsets of UGV-Sync are validated and refined by frame-by-frame comparison. This role is not only operational, since the reliability of the annotated offsets bounds the validity of every metric reported in this dissertation. Tolerance-based metrics such as $\text{Acc}@1$ and $\text{Acc}@3$ would not be meaningful if the ground truth itself carried an error of a few frames, and the tool is what makes this level of annotation accuracy possible.

7. Finally, this work provides an empirical analysis of the main challenges involved in the synchronization of multi-view sports UGVs, especially under different-view and far-view conditions. The analysis helps to identify the remaining limitations of the proposed approach and motivates future directions based on scene-level semantic representations and cross-view synchronization.

It is important mention how these contributions relate to each other, since they are not independent claims of the same nature. The central scientific claim of this dissertation is empirical and comparative, namely that holistic self-supervised representations support the cross-view temporal alignment of sports UGVs better than identity-specialized ReID embeddings, and that the failure of the latter is representational rather than algorithmic. A claim of this type cannot be sustained by the proposed framework alone, because it depends on being measured under conditions that reproduce the phenomenon under study. The datasets described in contribution 5 define these conditions, the annotated offsets of UGV-Sync make the comparison quantifiable, and the UGV-Sync Studio described in contribution 6 supports the reliability of these annotations. To the best of our knowledge, no publicly available benchmark provides ground-truth temporal offsets for multi-view sports user-generated videos under these viewpoint conditions. In this sense, contributions 5 and 6 are not parallel results placed beside the main conclusion, but the conditions under which this conclusion becomes verifiable. These contributions were consolidated through the publications listed in Section 7.2, and are detailed throughout Chapters 3 to 6.

This dissertation is organized as follows. Chapter 2 presents the theoretical background, covering the deep learning models used to automatically extract visual features through multi-object tracking, as well as general concepts about video synchronization and similarity metrics. Chapter 3 describes the problem and the challenges of synchronizing UGVs, presents the proposed framework, and reviews related work on multi-object tracking, vision-based synchronization and multi-view UGV datasets. Chapter 4 describes the adopted methodology, Chapter 5 the experiments carried out to validate the proposal, and Chapter 6 the analysis of their results. Finally, Chapter 7 presents the final considerations and the future work directions.

Chapter 2

Theoretical Background

This chapter provides a theoretical background that is the basis for the framework proposed in this research. The sections are structured to provide a comprehensive overview of the main concepts and subject that guided this research.

2.1 Multi-View User-Generated Videos and its Synchronization

Videos can be considered generalizations of image data in which a temporal component is inherent to a sequence of images. So, they comprise spatio-temporal data. Each frame in a video is an image, and therefore a video receives a sequence of images in time [Aggarwal 2018]. They can vary in length, quality and format, especially when recorded by consumer devices such as smartphones, digital cameras, and tablets [Guggenberger 2023]. In the context of this dissertation, the term video refers to a recording of an event made with a mobile device, which is often shared online.

User-Generated Videos (**UGVs**) are recordings made by individuals, usually with personal mobile devices, like smartphones. These videos are characterized by their diversity in terms of quality, viewing angle and stability, reflecting the variety of devices and perspectives of the users who capture them [Naab and Sehl 2017]. They are often captured at events such as concerts, **sports matches** and among others, providing **multiple perspectives** of the same event. They are often shaky, blurry, contain fast movement, or exhibit compression artifacts, which makes matching of

features across frames challenging [Guggenberger 2023].

In the context of sporting events, such as a football match in a stadium, multiple users can record videos simultaneously from different angles around the field. This scenario leads to the concept of **Multi-Camera Views** or **Multi-View Videos**, where multiple videos are captured by different users positioned in different locations. When combined, these UGVs provide a more comprehensive understanding of the event, enabling analysis of the scenes from various perspectives [Olagoke et al. 2020]. Figure 2.1 illustrates a typical multi-view scenario, where multiple users, positioned in different parts of a stadium, record video simultaneously. The existence of multi-view videos enables the application of synchronization techniques that combine these different points of view into a single timeline, providing a complete and integrated view of the recorded scene [Hosain et al. 2024].

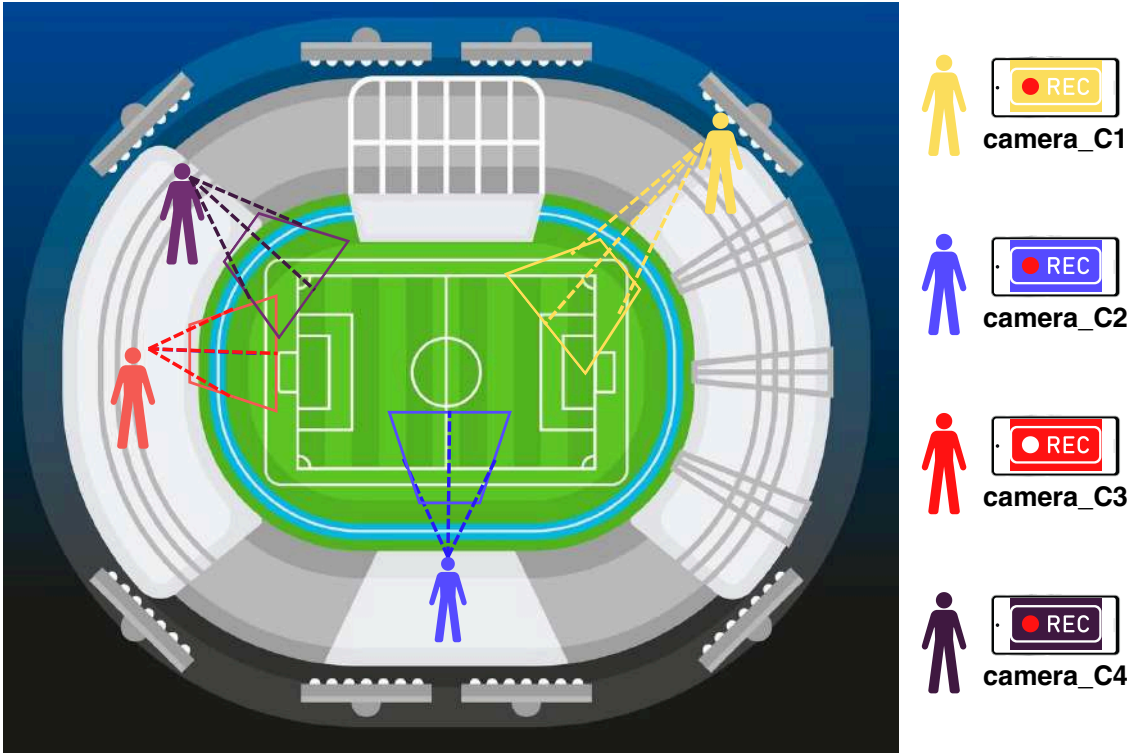


Figure 2.1: An illustration of the multi-camera view setup within a sports stadium, where 4 users record videos from different perspectives.

Overlap refers to fragments where multi-view videos capture a common scene/action in the same period of time. Identifying these overlapping segments is a part of the synchronization process, as they contain common content that can be used to

align videos temporally. Efficient visual-based synchronization relies on detecting these overlaps, which should include portions of the same visual scene with similar visual cues. Meanwhile, **offset** is the time difference between the start points of two recordings. This offset should be taken into account and adjusted so that the videos can be aligned closely on a common timeline [Guggenberger 2023].

2.1.1 Video Synchronization

Video synchronization involves aligning multi-view videos on a common timeline so that their contents are matched both spatially and temporally. Whitehead et al. 2005 defined the frame-based synchronization problem as: given k different video sequences that overlap in time, identify a frame from each sequence that refers to the same point in time. For each video sequence i , the function $T_i(f)$ maps an integral frame number f from sequence i to universal time. Thus, the synchronization equality can be expressed as finding the frame numbers $(f_1, f_2, \dots, f_k)$ in a way that $T_1(f_1) = T_2(f_2) = \dots = T_k(f_k)$ [Whitehead et al. 2005].

However, due to possible minute variations in camera start times and variations in frame rates, perfect integral synchronization does not always exist. In this case, Whitehead et al. 2005 proposes searching for a set of frames whose pairwise differences with respect to synchronization equality are minimized. It is argued that if the restriction of integral frame numbers is removed, the time function of the frame maps real frame values (integral and non-integral) to a point in time. In this case, there will always be a set of real frame numbers such that the synchronization equality is satisfied [Liu et al. 2024].

In his PhD thesis, Guggenberger 2023 highlighted a set of applications and use cases that can benefit from the ability to generate a synchronized set of event recordings. Event reconstruction is an application example, which makes it possible to create a comprehensive view by merging multiple recordings, useful for complete videos of concerts, sporting events, player performance analysis, and other applications. Mashup creation is another example, where the best video tracks from various sources are selected to improve the final video content [Wu et al. 2015]. Additionally, 3D scene reconstruction from synchronized videos from different angles provides a detailed 3D perspective of the event, which is relevant for virtual reality

and forensic analysis [Wang et al. 2016, Liu et al. 2024].

Other use cases include detecting key moments by analyzing the number of simultaneous recordings, creating dynamic videos by switching between different perspectives, and identifying clusters of related recordings to facilitate the management of large volumes of data. The collection and enhancement of personal and semi-professional footage, such as interviews, also benefit from synchronization. In addition, coordinated productions involving multiple recording devices demonstrate how synchronization can create comprehensive event videos. Professional productions, such as music videos made from fan footage, also use crowd-sourced video synchronization to create high-quality content [Guggenberger 2023].

2.1.1.a. Video Synchronization Challenges

Synchronizing user-generated videos presents several challenges. First, the heterogeneous devices used by users vary in recording capabilities, configurations, and formats, which complicates the alignment process. Different devices have different frame rates and resolutions, and the recording quality can vary significantly depending on the device and recording conditions. This, along with the variability of the content, is also a challenge, as differences in audio and video quality, frame rates, and recording conditions make synchronization complex. In addition, manual synchronization of large data sets is laborious and impractical, requiring effective automated solutions [Guggenberger 2023].

Due to the variability of devices, one of the main challenges is clock drift (i.e. offset), which refers to variations in the accuracy of recording times between devices. This drift can cause a gradual divergence in the recordings over time, making accurate synchronization difficult. In addition, metadata such as timestamps (date and time information) are often inaccurate or nonexistent, making it difficult to rely on them for synchronization [Guggenberger 2023, Whitehead et al. 2005].

2.1.1.b. Synchronization Methods

Manual video synchronization involves detailed analysis of audio and video content to identify similar moments and sort them, often using video editing tools. However, this process becomes impractical for large volumes of data, thus requiring automa-

tion. Automatic methods use timestamps or electronic signals to sort recordings, but suffer from the inaccuracy of device clocks and do not guarantee continuous synchronization throughout the recording, especially those recorded by end-user devices.

Audio-based methods, such as audio fingerprinting, offer robustness to noise and compression and are widely used due to the low directionality of microphones, which makes it easier to record a common dominant audio source at an event. Alternative methods use audio feature vectors such as onsets, chromas, and mel-frequency coefficients (MFCCs). There are also visual-based methods, which often face prohibitive restrictions and only work in controlled environments, requiring fixed camera settings, static camera views and large continuous overlaps [Guggenberger 2023].

Some methods use local feature tracking and alignment of their trajectories [Whitehead et al. 2005, Brito et al. 2011], while others employ image brightness changes, phase correlation, static camera flashes, spatial gradients or video fingerprints. However, many of these methods are proof-of-concept and lack comprehensive evaluation, and are far from real-world applicability [Guggenberger 2023].

2.2 Video Feature Extraction

Considered as one of the main research topics in computer vision, Video Understanding aims to **extract** and interpret visual features to have meaningful information from video sequences [Wang et al. 2024]. With the growth of online video platforms and the increasing prevalence of cameras integrated in mobile devices (hand-held devices), such as smartphones, the demand for automatic video analysis methods has increased considerably [Tang et al. 2023]. Deep learning techniques proved to be effective in this context, providing robust and accurate solutions for various tasks, such as the automatic detection of objects in a video and the analysis of their temporal evolution [Wang et al. 2024].

The tasks related to this include classifying videos, recognizing actions, **tracking object**, captioning, and answering questions about the content of videos. In sporting events, deep learning techniques can be applied to the analysis of athlete

performance, reviewing plays/actions, and automatically generating highlight clips, among others [Zhao et al. 2023]. Such applications can face significant challenges, such as scalability for processing large volumes of data, integrating multiple modalities (video, audio, text) for comprehensive analysis [Tang et al. 2023]. The main approaches include convolutional neural networks (CNNs) and Visual Transformer-based models (ViT) dosovitskiy2020ViT. In this work, these approaches are used to obtain the visual representations that are presented and compared in the following subsections: CNN features, re-identification (ReID) embeddings, which are a CNN-based representation, and self-supervised Vision Transformer features.

2.2.1 Convolutional Neural Network

Since the baseline DL-based techniques reviewed in this work are mainly Convolutional Neural Network (CNNs), this subsection has an overview of such approach. CNN is a class of deep learning models especially designed to process data with a grid structure, such as the frames of videos (*i.e.*, images), whose main advantage is its ability to preserve the spatial structure of the data throughout its layers. It is mainly used in computer vision for image classification and object detection [Aggarwal 2018], which extracts visual information from data, enabling the model to identify patterns in order to make automated decisions in various tasks, such as image classification [Krizhevsky et al. 2012], segmentation, object detection and tracking [Girshick et al. 2014].

As partially illustrated in Figure 2.6(a), the main components of CNNs are the convolutional layers, which represent hidden feature maps that encode various types of shapes in the image [Aggarwal 2018]. The kernel filters are used to extract visual features from the input image, such as edges, textures and shapes; the pooling layers are responsible for reducing the dimensionality of the feature maps; the activation layers enable the models to learn complex patterns extracted by the previous components; the fully connected layers are responsible for combining the extracted features to carry out the classification [Krizhevsky et al. 2012].

CNNs have proven effective in video processing, as they facilitate the extraction of visual features from individual frames [Tang et al. 2023]. This process is important for the automatic analysis of image sequences, allowing the model to extract

key visual features and relate them across frames. These features can include edges, contours, and patterns that are consistent over time and can thus be used to identify and track moving objects, for example [Russell and Norvig 2020].

As a result of their advantages, CNNs have been applied in object detection and tracking techniques in recent years. In detection, CNN is used to identify objects in each frame and provide their locations [Ren et al. 2016, Redmon 2016]. In object tracking, CNNs help correlate objects across different frames, enabling tracking of their movements over time [Bewley et al. 2016, Nasser et al. 2021]. A common application of these CNN-based features is the extraction of appearance descriptors used to recognize the same object across frames, known as re-identification (ReID) embeddings, which are presented in the next subsection [Zhou et al. 2022].

2.2.2 Re-Identification Embeddings

Re-Identification (ReID) is the task of recognizing the same object across different frames or even different cameras. To perform this task, ReID models extract appearance feature vectors, here called ReID embeddings, from the image crop of each detected object. These embeddings are used to determine whether two detections correspond to the same object by comparing their vector-space distances. To produce vectors with this property, ReID models are usually trained with metric learning, where the objective is to keep the embeddings of the same identity close in the vector space and the embeddings of different identities far from each other [Zhou et al. 2022].

An example of such model is OSNet [Zhou et al. 2022], an Omni-Scale Network designed to learn discriminative identity features at different scales. Its main building block, the Omni-Scale block, combines several convolutional streams with different receptive field sizes and fuses them via a unified aggregation gate that weights each scale based on the input. In this research the OSNet-AIN variant (`osnet_ain_x1_0`) is used, which adds instance-normalisation layers to OSNet to make the embeddings more robust across different domains (Figure 2.2). OSNet-AIN is used as the appearance feature extractor inside the StrongSORT tracker, described in Section 2.3.2.

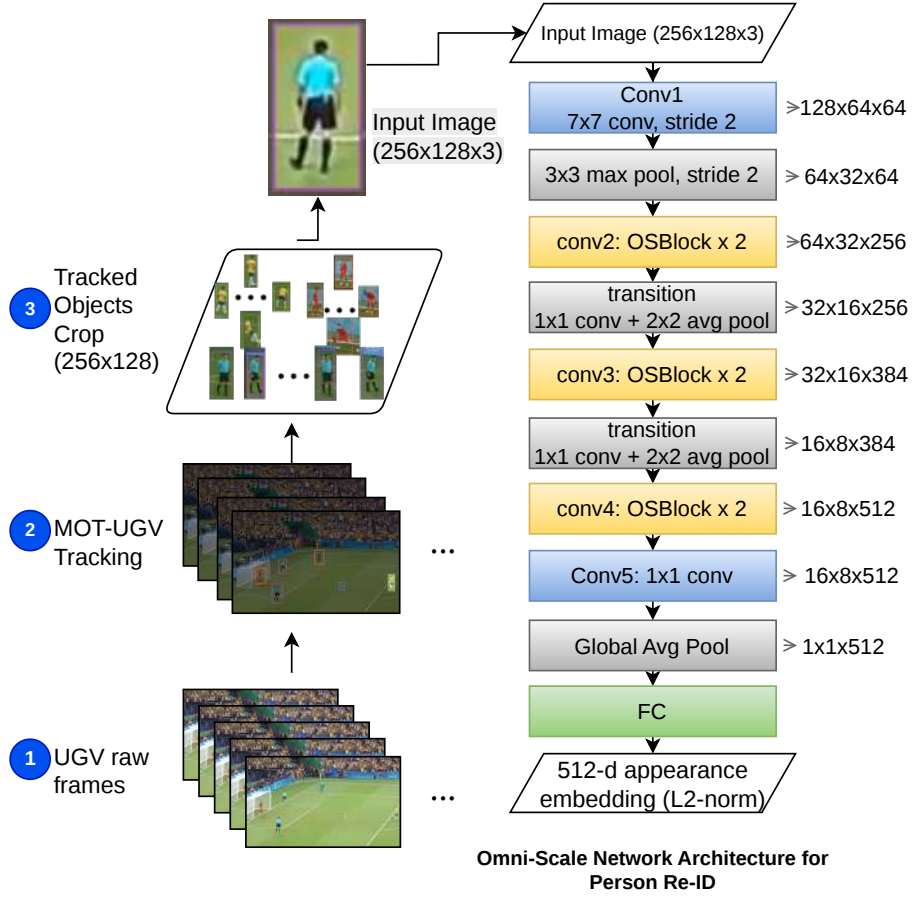


Figure 2.2: OSNet-AIN architecture used as the ReID feature extractor: a 256×128 object crop is processed through stages `conv1`–`conv5` into a 512-dimensional L2-normalized embedding [Zhou et al. 2022].

Figure 2.2 shows how this extractor is integrated into the tracking pipeline and how an image crop is processed within the network. Each object detected and tracked in the UGV frames is cropped and resized to 256×128 , and then used as the OSNet-AIN input. The crop passes through the convolutional stages (`conv1` to `conv5`), where spatial resolution is progressively reduced while the number of channels increases [Zhou et al. 2022]. After a global average pooling and a fully connected layer, the network produces a 512-dimensional appearance embedding, L2-normalized, for each tracked object. This is the vector that is later used to compare detections and, in our framework, to match tracklets between different views.

2.2.3 Vision Transformers and Self-Supervised Representations

In addition to the CNN-based representations presented above, another family of models used for visual representation feature extraction is the Vision Transformer (ViT) [Dosovitskiy 2020]. Unlike CNNs, which extract features in a local and hierarchical way through convolutions, the ViT divides the input image into patches and processes them with a self-attention mechanism [Vaswani et al. 2018]. In this way, the model can relate distant regions of the image since the first layers, capturing a more global context of the scene.

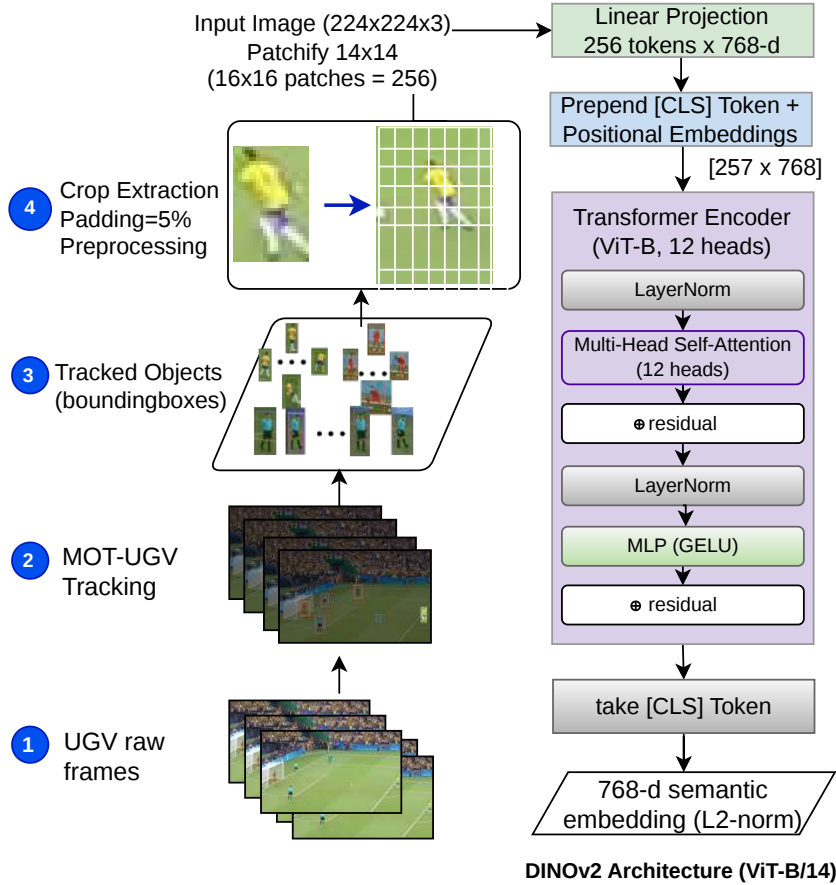
A relevant property of transformer-based models appears when they are trained in a self-supervised way, that is, without using class labels. The DINO method [Caron et al. 2021] showed that ViT trained with self-supervision develops emergent semantic properties, such as information related to the segmentation of objects, which is less evident in CNNs trained with supervision. The DINOv2 model (as illustrated in Figure 2.3) extends this idea by providing general-purpose visual features, trained without explicit supervision on a large amount of data, that can be used in different tasks and domains without specific fine-tuning [Oquab et al. 2023].

In practice, the [CLS] token produced by the ViT serves as a global image descriptor, encoding the entire scene into a single feature vector. In this work, these features are called as holistic, because they represent the image as a whole and not as a set of local parts. For this reason, self-supervised features tend to be more robust to changes of viewpoint, illumination and scale [Oquab et al. 2023], which are common problems in user-generated videos recorded from different positions. This characteristic is particularly relevant for the cross-view scenario treated in this research, and motivates the comparison between identity-specialized ReID embeddings and holistic self-supervised embeddings.

Figure 2.3 illustrates how these holistic features are obtained in this work. Starting from the UGV frames, the tracking stage provides the tracked objects as bounding boxes; each detection is then cropped and passed to the DINOv2 model, which returns a single 768-dimensional embedding that semantically describes the whole crop. Unlike OSNet-AIN, which is trained with identity supervision, DINOv2 is trained in a self-supervised manner and encodes global semantic structure rather

than instance identity.

Figure 2.3: DINOv2 (ViT-B/14) Architecture: each input image is cropped (5% padding), resized to 224×224 and split into 256 patches; the Transformer encoder processes them and the [CLS] token is taken as a 768-dimensional, L2-normalized holistic embedding of the crop. [Oquab et al. 2023].



2.3 Multi-Object Tracking (MOT)

The task of tracking an object (*i.e.* entities or distinct things present in an environment that can be seen or touched), that is, Multi-Object Tracking (MOT), plays a fundamental role in understanding video content and, consequently, in describing scenes. Currently, deep learning-based techniques have been applied in object detection and tracking [Russell and Norvig 2020]. These techniques, which are specialized in detecting, segmenting, and tracking multiple objects in videos, are gain-

ing significant attention due to their academic and commercial potential [Luo et al. 2021]. They facilitate the automation of video content understanding, making them appealing across various areas of application, such as sports video analysis [Rangasamy et al. 2020, Held et al. 2023, Zhao et al. 2023], analysis of multiple videos using tracking-based information to handle temporal video synchronization [Whitehead et al. 2005, Brito et al. 2011], action recognition [Alencar et al. 2022], video summarization, among others [Du et al. 2023, Nguyen et al. 2024]

Thus, there are various MOT approaches, and according to Meinhardt et al. 2022, these approaches can be categorized into four different paradigms: tracking-by-detection, tracking-by-regression, tracking-by-segmentation, and tracking-by-attention. Due to the advances in deep learning-based object detection approaches, **Tracking-by-Detection** has become the most explored paradigm [Wojke et al. 2017, Du et al. 2023]. As illustrated in Figure 2.4, the process for this paradigm begins by detecting the objects of interest. Subsequently, for tracking, a unique identifier is assigned to the object. This detection is propagated throughout the video using a model that maintains an association of the object’s location in subsequent frames [Athar et al. 2023].

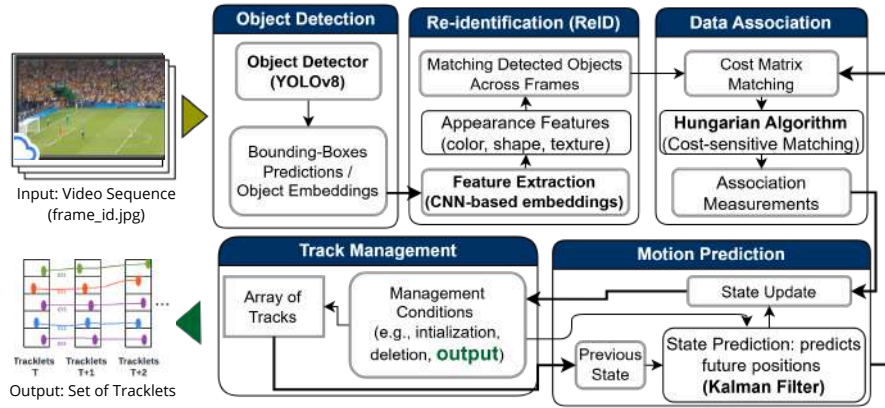


Figure 2.4: General workflow of a tracking-by-detection paradigm, as StrongSORT, broken down into five main stages: (I) Object Detection, (II) ReID, (III) Data Association, (IV) Motion Prediction, and (V) Track Management.

Figure 2.4 presents a general workflow of a tracking-by-detection algorithm, which is the base for most modern object tracking systems. The process is divided into five main stages. First, object detection is performed using a model such

as YOLOv8, which provides bounding boxes and embeddings for each object in the frame. Then, re-identification (ReID) is applied to match the same objects across frames using visual appearance features; these features are usually extracted with a CNN-based embedding network. In the next stage, data association is performed by comparing the detected objects and existing tracks using a cost matrix and the Hungarian algorithm [Kuhn 1955]. After that, the motion prediction step updates and predicts the location of the object using a Kalman Filter [Kalman 1960]. Finally, track management handles initialization, update, or deletion of object tracks, generating the final output as a set of tracklets.

2.3.1 YOLO-World: An Open-Vocabulary Object Detector

Object Detection is present in the process of MOT techniques, especially the tracking-by-detection ones. In their book, Russell and Norvig 2020 affirm that the main goal of object detection models is to search for different objects in a given image, report what the class is for each object found, and also return where each object is in the environment, through a *bounding box* around the detected object. In summary, the model uses a variation of a Convolutional Neural Network as a backbone to extract the visual feature, which divides the image input into smaller regions and then predicts the *bounding boxes* for each object detected, similar to what is illustrated in Figure 2.5.

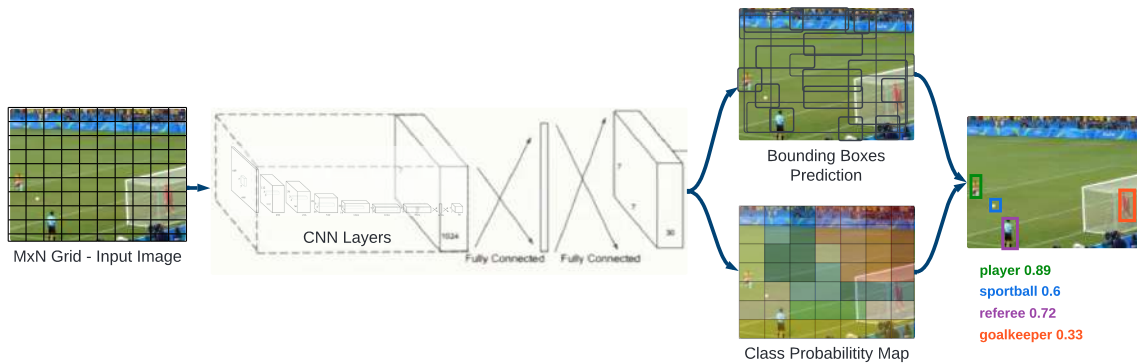


Figure 2.5: General overview of a CNN-based object detection process.

To carry out the object detection process, there are some state-of-the-art solutions available in the literature. The You Only Look Once (YOLO) model [Redmon 2016], is an example of a solution. And its series of improved versions (Yolov1 to

v8), build upon the success of its previous version, have established themselves as efficient and practical tools [Hussain 2024]. Most of these versions (including the latest one) have their pre-trained weights publicly available, and were trained using the MS COCO dataset, a large-scale object detection dataset, with 1.5 million object instances among 80 object categories [Lin et al. 2014]. Therefore, all these pre-trained models are specialized in detecting and classifying only known objects within these categories.

2.3.1.a. YOLO-World

Considering what was mentioned above about the models trained on the MS COCO dataset, YOLO-World (2024), an Open-Vocabulary Object Detector, was introduced by Cheng et al. 2024 to mitigate the limitation of previous object detection models, which is the reliance on predefined and fixed object categories, that restricts their applicability in open-world scenarios. This approach extends the YOLOv8 architecture with open-vocabulary detection capabilities through vision-language modeling and pre-training on large-scale datasets such as CLIP, enabling the detection of new objects beyond the training categories [Cheng et al. 2024]. As illustrated in Figure 2.6 (b), YOLO-World consists of two main input branches: one for the visual domain and the other for the language domain.

Figure 2.6 (a) illustrates the stages in the YOLOv8 architecture, starting with the image input (ideally $\text{height} = 640 \times \text{width} = 640 \times \text{channels} = 3$), which is processed by a sequence of convolutional layers in the backbone. This backbone successively applies Conv blocks (with a 3×3 kernel, stride 2 and padding 1) to reduce the resolution and increase the depth of the representations, followed by C2f blocks that refine these features through multiple internal convolutions with shortcut-type connections. Throughout these processing, the spatial resolution of the image is reduced from 320×320 to 20×20 , while the depth of the channels is progressively increased. At the end of the backbone, the SPPF (Spatial Pyramid Pooling Fast) structure adds multiscale context, expanding the network’s field of view without compromising computational efficiency. After this stage, the activation maps are passed on to the detection head, which generates embeddings of objects with their respective positions and classes [Jocher et al. 2023]. However, standard YOLOv8

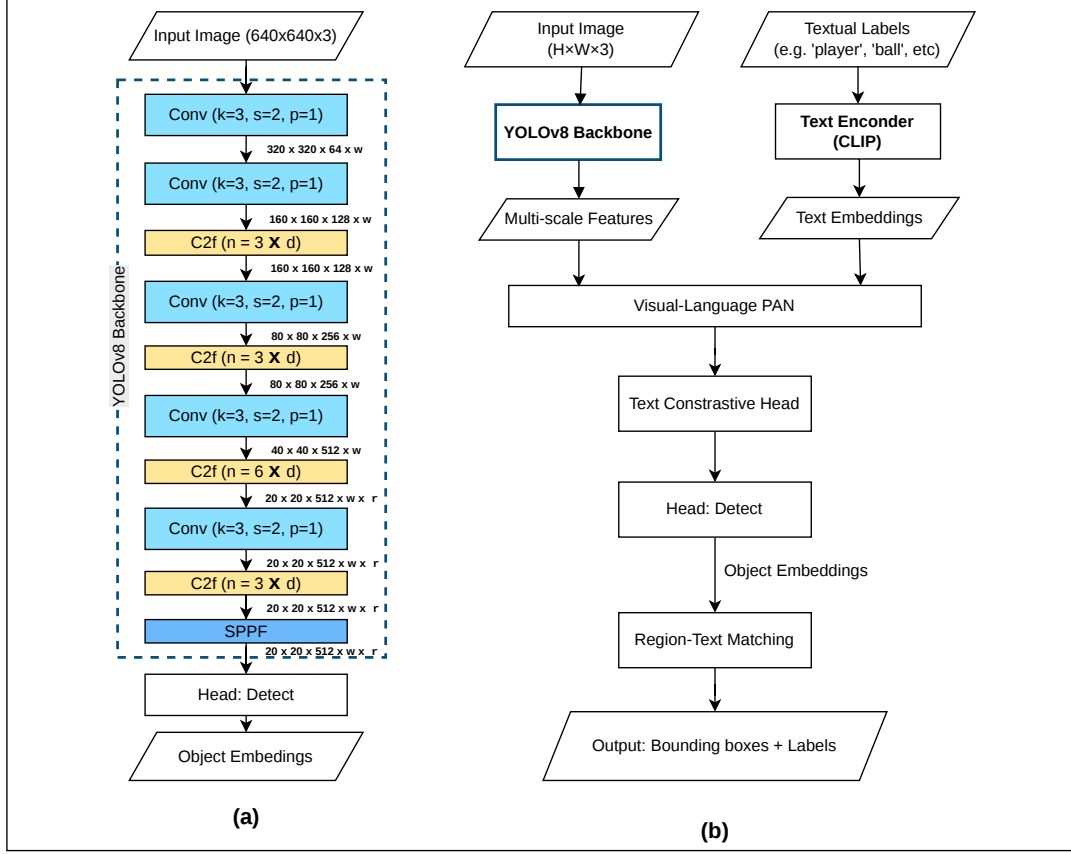


Figure 2.6: Overview of the YOLO-World architecture. (a) YOLOv8 backbone integrated to it for visual feature extraction [Jocher et al. 2023]. (b) Open-vocabulary detection pipeline combining image multi-scale features with CLIP-based text embeddings to infer the bounding boxes and labels [Cheng et al. 2024].

models are limited to closed-set detection, recognizing only object classes seen during training.

YOLO-World addresses this by incorporating CLIP-based text embeddings and a contrastive learning framework, allowing the model to generalize to unseen classes during inference. Figure 2.6 (b) illustrates the YOLO-World processing flow, where the image input is processed by a YOLOv8-based backbone that extracts multi-scale feature representations. Simultaneously, the textual labels (e.g., "player", "ball", "referee") are encoded using a CLIP-based text encoder to generate semantic embeddings. These two streams are then integrated through the Visual-Language PAN module, which performs cross-modal fusion between visual and textual features. Following this fusion, the Text Contrastive Head further aligns the modalities by com-

puting similarity scores, and the fused features are passed to the YOLOv8 detection head. Instead of relying on predefined class output, the detection stage generates region embeddings that are then matched against the text embeddings using the Region-Text Matching module [Cheng et al. 2024].

2.3.2 StrongSORT: A tracking-by-detection approach

A tracking-by-detection model well-established in the literature is StrongSORT [Du et al. 2023], an improved version of DeepSORT [Wojke et al. 2017], which was build on top of SORT (Simple Online and Realtime Tracking) [Bewley et al. 2016], a classic method that predicts the current position based on the target object’s position in the previous frame and matches the detection bounding boxes with the predicted position, all these relying on Kalman Filter and Hungarian Algorithm methods (Figure 2.7). StrongSORT was introduced with the aim of improving the tracking performance in crowded scenes and reducing the frequency of ID switches in real-time tracking task [Li et al. 2024].

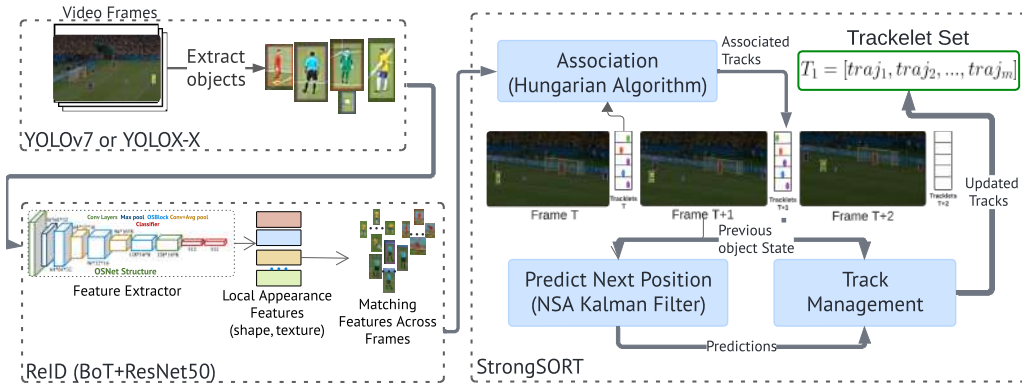


Figure 2.7: General overview of the processes in tracking-by-detection approach: object detection, Re-identification, data association, motion prediction, array of tracks.

Figure 2.7 shows an overview of the StrongSORT approach. It starts by detecting objects in each video frame using a YOLO-based model (e.g., YOLOv7). Each detected object is then represented by local appearance features, extracted by a CNN-based network (i.e., BoT+ResNet50). These features help match objects across consecutive frames. The data association step uses the Hungarian algorithm

to link the detections to existing tracklets. An optimized Kalman filter (NSA version) predicts the next position of each object, and track management updates the states based on new detections and predictions. As a result, the algorithm produces a set of updated tracklets that represent object trajectories over time.

One of the main improvements of StrongSORT is the adoption of a more robust detector and an advanced visual feature extractor, the BoT model, a Batch Normalization Neck (BNNeck), which makes the model more effective at discriminating between similar objects. In addition, the exponential moving average (EMA) is used as a new update process strategy to increase the accuracy of associations over time. These improvements make this technique one of the most precise trackers in benchmarks like MOT17 and MOT20, achieving great results in terms of identification consistency and detection accuracy [Du et al. 2023].

2.3.3 MOT Evaluation

For a given MOT approach, metrics and datasets are required to evaluate its performance quantitatively. In this way, evaluation metrics are used to provide a means for fair quantitative comparison. The Multi-Object Tracking performance can be evaluated by metrics that have been introduced in the literature. Furthermore, since many approaches to MOT employ the tracking-by-detection strategy, they often measure both detection and tracking performance. For this reason, object detection metrics are also adopted in MOT approaches [Luo et al. 2021].

2.3.3.a. Quantitative Evaluation Metrics

Detection accuracy can be evaluated using Precision and Recall, two state-of-the-art metrics that balance false positives and false negatives. In addition, Multiple Object Detection Precision (MODP) measures the precision of predicted detections by calculating the percentage of correctly detected objects among the total objects in the ground truth annotations. [Luo et al. 2021].

Now, for the evaluation of tracking prediction, precision can be assessed using the Identity F1-Score (IDF1), which measures the accuracy of tracking identities by combining precision and recall for object identities [Ristani and Solera 2016]. Multiple Object Tracking Accuracy (MOTA) is an SOTA metric used to quan-

tify the accuracy of MOT results by relating false positives, false negatives, and **identity switches**, providing a comprehensive measure of overall tracking performance [Bernardin and Stiefelhagen 2008]. Another metric used for MOT evaluation is HOTA (Higher Order Tracking Accuracy), a combination of detection score (DetA - Detection Accuracy) and association score (AssA - Association Accuracy), which balances the effects of performing accurate detection and association into a single unified metric [Luiten et al. 2021]. All the code for the implementation of these metrics and others can be found in the TrackEval framework, a public codebase repository [Luiten and Hoffhues 2020].

2.3.3.b. Datasets & Benchmarks

According to [Huang et al. 2024], the most recent tracking algorithms focus mainly on pedestrians or vehicle tracking. These algorithms have made significant progress in public benchmarks, such as MOT16 [Milan et al. 2016], MOTS [Voigtlaender et al. 2019], and MOT20 [Dendorfer et al. 2020]. However, these algorithms face challenges in datasets specifically designed for sports scenarios [Huang et al. 2024]. Therefore, the application of object tracking in sports videos can contribute to the demands of automatic sports analysis, such as tactical and statistical analysis of athlete movement [Zhao et al. 2023].

2.4 Similarity Metrics Calculation for Visual Features

The comparison of visual features extracted from images is part of different processes available in methods of various areas of computer vision. In order to tell whether these visual features are similar or not the literature uses metrics, which are responsible for quantifying how similar or different the characteristics of two images are in terms of their numerical representation, known as feature vectors [Mahendran and Vedaldi 2015, Peng et al. 2020]. Metrics that perform this task have been used in various computer vision applications, such as object recognition, multiple object tracking [Peng et al. 2020], and video synchronization [Boizard et al. 2023]. In this work, three main metrics (Euclidean Distance, Cosine Similarity and Mean Squared

Error) were selected as the point-to-point base metrics for comparing feature vectors, especially due to their use in works that aims to synchronize videos based on frame-level visual features [Mei et al. 2023]. In addition, a sequence alignment method, and a strategy to select the most reliable matches are also considered, as detailed at the end of this section.

The **Euclidean distance** has been used in a variety of image retrieval-based applications, such as pattern recognition, image search, representation and clustering, as it provides a direct measure of how different two vectors of features extracted from images are [Lowe 2004, Wang et al. 2014, Varior et al. 2016]. It is used to measure the similarity between these features vectors by calculating the square root of the sum of squared differences between the corresponding coordinates of two vectors. For example, given two vectors of trajectory features $traj_1$ and $traj_2$, the Euclidean distance formula is represented in the Equation 2.1. The smaller $EucD(traj_1, traj_2)$ is, the more similar the two vectors of features are.

$$EucD(traj_1, traj_2) = \sqrt{\sum_{i=1}^n (traj_{1i} - traj_{2i})^2} \quad (2.1)$$

However, **Cosine Similarity** is applied in information retrieval and pattern recognition tasks, especially when the magnitude of the vectors is not relevant. This metric measures the angle between two vectors, ignoring their magnitudes [Yu et al. 2016, Peng et al. 2020, Boizard et al. 2023]. It is calculated as the cosine of the angle between the vectors, resulting in 1 when the vectors are identical (same direction) and 0 when they are orthogonal (no similarity). This metric is calculated as shown in Equation 2.2.

$$CosineSim(traj_1, traj_2) = \frac{traj_1 \cdot traj_2}{\|traj_1\| \|traj_2\|} \quad (2.2)$$

Finally, the third metric is **Mean Squared Error (MSE)**, which is particularly useful in similarity analysis when quantifying the absolute error between two extracted tracklet features. In this case, the interpretation of the result will be similar to that of the first metric, where the smaller the similarity score is, the more similar the features are. As shown in the equation 2.3, this metric quantifies the mean squared difference between the values of the two trajectories, and is a metric that is sensitive to large differences between values [Mei et al. 2023].

$$MSE(traj_1, traj_2) = \frac{1}{n} \sum_{i=1}^n (traj_{1i} - traj_{2i})^2 \quad (2.3)$$

2.4.1 Sequence Alignment Method

The metrics presented above compare two feature vectors in a point-to-point way, that is, the ReID-based features from one specific frame of a tracklet is compared directly with the ReID-based features of another tracklet. However, the synchronization of videos is not only a problem of finding the most similar pair of frames, but of finding a temporal alignment that is consistent along the sequence. In other words, if a frame of the first video corresponds to a frame of the second video, the following frames should also keep this correspondence. This makes the synchronization closer to a sequence alignment problem than to a pure similarity search.

For this reason, the Dynamic Time Warping (DTW) algorithm [Sakoe and Chiba 1978] is also considered in this work. The DTW compares two sequences of vectors and finds the alignment that minimizes the total distance between them, allowing small temporal distortions, for example when the objects move with slightly different speeds in each video. In this way, the DTW captures the temporal structure of the tracklet trajectory, and not only the similarity of ReID-based features of a isolated frame. A faster approximated version, FastDTW [Salvador and Chan 2007], can be used to reduce computational cost.

When two videos are compared, many pairs of tracklets are obtained, and each pair produces a similarity value and a possible correspondence. Not all of these pairs are reliable: some of them can present a low distance by appearance coincidence, behaving as outliers. For this reason, a strategy to select the most reliable matches is necessary, instead of trusting a single closest pair or a simple average of all of them.

A classical strategy for this kind of problem is the consensus based on RANSAC [Fischler and Bolles 1981], which keeps the candidates that are supported by the agreement of the largest number of other candidates, being robust to the presence of outliers. This idea of robust selection of the closest and most consistent matches has also been used in works of visual-based video synchronization [Whitehead et al. 2005, Brito et al. 2011].

Chapter 3

Track-UGV-Sync Overview

This chapter introduces the general overview of the problem for synchronizing user-generated videos using only visual features and presents the proposed framework designed to mitigate this problem, which was defined based on the related work discussed later in the chapter.

3.1 Problem Overview and Proposed Framework

The number of people using their mobile devices to record videos during large public events, such as sports matches, concerts, or festivals, has increased. During these events, multiple users often capture videos from different perspectives and then share them on online platforms, such as YouTube, resulting in a rich but unsynchronized collection of multi-view user-generated videos. The temporal alignment of these videos can enable multi-perspective analysis, allowing, for example, multi-angle replays, video editing, mashup creation, or other applications that aim to provide a comprehensive understanding of the captured event.

However, synchronizing videos recorded from different perspectives, often without temporal references such as timestamps, is a significant challenge. Traditional synchronization methods that rely on audio-visual tracks or manual alignment are either impractical or unfeasible for processing Multi-View UGVs, given the scale and variability of such content. Since synchronization can be performed using visual features extracted from video frames, and since these video collections often lack reliable timestamps generated from device clocks, this work proposes a frame-

work for synchronizing UGVs using visual information extracted from tracklets by deep learning-based techniques, without relying on timestamp information.

A tracklet is defined as a sequence of object detections across consecutive video frames, where each detection is associated with the same tracked object ID. It represents the trajectory of a detected object within a segment of a video. These tracklets are formed by associating object detections in consecutive frames using a tracking algorithm, ensuring that each detected object maintains a consistent track ID over time.

However, the visual features used to describe these tracklets are not all equivalent for the synchronization task. The ReID embeddings produced by the MOT are optimized to discriminate identity, that is, to decide if two image crops are the same object, which is not the same objective of finding the same temporal moment of the event observed from different views. For this reason, this work does not assume a single representation as given. The same synchronization pipeline is instantiated with two different visual representations, which are then compared under identical conditions.

The first one is the identity-specialized ReID embedding produced by the tracker itself (OSNet-AIN), which is taken as the baseline representation, since it is what a tracking-based pipeline provides. The second one is a self-supervised representation extracted by a ViT-based model (*i.e.*, DINOv2), which is not trained for identity but for general visual correspondence. The experiments reported later show that the holistic representation is at least as accurate as the ReID one in every recording condition, so DINOv2 is adopted as the default representation of the framework, and OSNet-AIN is kept as the comparison baseline. It is important to note that this choice is fixed and made offline. The framework does not select the representation per video pair and, in particular, it does not require any prior knowledge about how the two cameras were positioned. The recording conditions formalized in Definition 5 are used only to stratify the experimental analysis, never as an input of the method.

3.1.1 Mathematical Formalization of the Problem

Before presenting the detailed formulation of the synchronization framework, we give a formal mathematical foundation for the main concepts. This formalization

provides the theoretical basis for the proposed approach and supports the analysis of its properties.

To formalize the problem, we define the fundamental concepts about video, user-generated video (UGV), temporal overlap, and offset as follows.

Definition 1 (Video). *A video V is a finite sequence of frames $V = \{f_1, f_2, \dots, f_F\}$, where $f_t \in \mathbb{R}^{H \times W \times C}$ is an image at the discrete time instant t , with height H , width W , and C color channels. The temporal duration of the video is $\tau(V) = F/T$, where T is the frame rate (fps) and F is the total number of frames.*

Definition 2 (User-Generated Video - UGV). *A User-Generated Video (UGV) is a video V captured by a consumer mobile device (usually, a smartphone) in an uncontrolled environment, by a non-professional user. Typical characteristics include, but are not limited to, variations in illumination, camera motion, compression artifacts, recording quality and the absence of reliable synchronization metadata.*

Definition 3 (Temporal Overlap). *Given two videos V_1 and V_2 , we define the temporal overlap interval $\mathcal{O} \subseteq \mathbb{R}$ as the set of time instants t for which both videos have a frame representing the same real-world event. Formally:*

$$\mathcal{O} = \{t \mid \exists f_t^{(1)} \in V_1, f_t^{(2)} \in V_2, \text{ such that } f_t^{(1)} \equiv f_t^{(2)}\},$$

where $\equiv$ denotes semantic and temporal correspondence between frames.

Definition 4 (Temporal Offset). *The temporal offset δ between two videos V_1 and V_2 is the difference between their recording start times on a universal clock. If t_1 and t_2 are the start times, then:*

$$\delta = t_1 - t_2.$$

Consequently, for the videos to be synchronized, frame $f_t^{(1)}$ must correspond to frame $f_{t-\delta}^{(2)}$. The unit of δ can be expressed in seconds or, more commonly, in number of frames.

[Non-Empty Overlap Assumption] This work addresses the problem of *estimating* the offset between two videos that do record a common portion of the same event, that is, it assumes $\mathcal{O} \neq \emptyset$. Deciding whether two arbitrary videos overlap at all is

a distinct problem, of detection rather than estimation, and it is out of the scope of this dissertation. The framework always returns an offset for the pair it receives, and it does not include a rejection criterion: applied to a non-overlapping pair, it would still report the most consistent hypothesis found, which would be spurious. Accordingly, every pair of the evaluation sets was built so that the two views share a common temporal window, as described in Subsection 6.3.1.

[Frame-Rate Normalization] Expressing δ in number of frames is only well defined when both videos share the same frame rate T . This is not the case for raw UGVs because consumer devices record with different nominal frame rates and frequently produce variable frame rate streams, so the physical duration of one frame is not the same in V_1 and V_2 . Therefore, before any processing, every video is re-encoded to a common constant frame rate (CFR) T^* , and both the estimated and the ground-truth offsets are always expressed in this single reference. Under this condition, the conversion between frames and physical time is direct:

$$\delta_{\text{sec}} = \frac{\delta}{T^*}, \quad \delta = \text{round}(\delta_{\text{sec}} \cdot T^*).$$

In this work $T^* = 30$ fps, which is the lowest nominal frame rate observed among the recording devices of the collection. Normalizing to the lowest nominal rate means that faster recordings are resampled down, instead of duplicating frames of the slower ones, which would create frames that were never captured. The conversion is applied with **FFmpeg** to every clip of the UGV-Sync set at the moment the evaluation clips are produced, and it was verified afterwards, so that both videos of a pair are guaranteed to be CFR at the same rate. This step operates on the video stream only, and does not depend on any metadata written by the recording device.

The difficulty of synchronizing two UGVs also depends on how far apart the two cameras are, so the pairs are characterized according to the spatial relation between the two recordings.

Definition 5 (Viewpoint Relation). *Let $\ell(V) \in \mathcal{L}$ be the discrete viewpoint label of a video V , where $\mathcal{L} = \{\text{LEFT}, \text{BEHIND_LEFT}, \text{CENTER}, \text{RIGHT}, \text{BEHIND_RIGHT}\}$ describes the position of the camera relative to the main activity area of the sport, and let $\pi(V)$ be the capture position from which V was recorded [Pessoa et al. 2026a].*

The viewpoint relation $\mathcal{R}(V_1, V_2)$ of a pair is defined as:

$$\mathcal{R}(V_1, V_2) = \begin{cases} SV, & \text{if } \ell(V_1) = \ell(V_2) \text{ and } \pi(V_1) = \pi(V_2), \\ DV\text{-near}, & \text{if } \ell(V_1) = \ell(V_2) \text{ and } \pi(V_1) \neq \pi(V_2), \\ DV\text{-far}, & \text{if } \ell(V_1) \neq \ell(V_2). \end{cases}$$

The labels in $\mathcal{L}$ are the viewpoint categories defined in the MUVY collection, where they were assigned by human visual inspection of each recording, according to the position of the camera with respect to the field or, for sports with a less standardized spatial layout, with respect to the primary activity area observed in the recording [Pessoa et al. 2026a]. When the two videos of a pair carry the same label, the pair is inspected once more to decide whether both recordings come from the same capture position (SV) or from two distinct positions inside the same region (DV-near). Different labels always indicate a large angular separation, and the pair is assigned to DV-far.

The proposed framework relies on a fundamental assumption about the nature of the synchronization problem. We formalize this assumption as follows:

Axiom 1 (Temporal Consistency). *For a pair of videos V_1 and V_2 recording the same event, there exists a single temporal offset δ^* that is valid for the entire duration of their overlap $\mathcal{O}$. This axiom implies that δ is a global constant and not a function of time, which is a fundamental assumption for any synchronization method that operates without external temporal references.*

From this axiom, we derive the following lemma, which justifies the use of a consensus-based estimation strategy:

Lema 1 (Consistent Temporal Correspondence). *Let $\mathcal{P} = \{(a, b)\}$ be the set of candidate tracklet pairs, where each pair generates an offset hypothesis $\delta_{a,b}$. If δ^* is the true temporal offset, then, for any pair of tracklets that represent the same overlapping object or event, the generated hypothesis will be within a tolerance Δ of δ^* . Formally:*

$$\forall (a, b) \in \mathcal{P}^* \text{ (correct correspondences), } |\delta_{a,b} - \delta^*| \leq \Delta,$$

where $\mathcal{P}^* \subseteq \mathcal{P}$ denotes the set of correct tracklet correspondences.

Proof. The proof follows directly from Axiom 1 and the definition of a tracklet as a temporally-ordered sequence of object detections. If a tracklet in V_1 and one in V_2 correspond to the same physical object or event, then the time difference between their corresponding frames must be constant and equal to the global offset δ^* , up to small estimation errors introduced by the tracking and feature extraction processes. These errors are captured by the tolerance parameter Δ . $\square$

We now state and prove the central theoretical result that justifies the RANSAC-based consensus strategy employed in the proposed framework.

Theorem 1 (Offset Estimation by Consensus). *Given a set of offset hypotheses $\mathcal{H} = \{\delta_{a,b} \mid (a,b) \in \mathcal{P}\}$, where a fraction $r > 0.5$ of the hypotheses are inliers (that is, they satisfy Lemma 1), the consensus voting strategy (RANSAC) that selects the offset δ_h with the largest number of inliers within the tolerance Δ will correctly estimate the true offset δ^* .*

Proof. Let $\mathcal{H}_{\text{in}} \subset \mathcal{H}$ be the subset of inlier hypotheses. By hypothesis, $|\mathcal{H}_{\text{in}}| > |\mathcal{H}|/2$. By Lemma 1, all hypotheses in $\mathcal{H}_{\text{in}}$ are within Δ of δ^* . Therefore, the neighborhood $\mathcal{N}(\delta^*, \Delta)$ contains more than $|\mathcal{H}|/2$ hypotheses.

Now consider any other offset $\delta' \neq \delta^*$. By definition, the inlier hypotheses are not in the neighborhood of δ' , unless δ' is also within Δ of δ^* , which would imply that δ' and δ^* are indistinguishable at the chosen tolerance. In the general case, the maximum number of hypotheses that can be in $\mathcal{N}(\delta', \Delta)$ is bounded by the number of outlier hypotheses, which is $|\mathcal{H}| - |\mathcal{H}_{\text{in}}| < |\mathcal{H}|/2$.

Therefore, $\mathcal{N}(\delta^*, \Delta)$ contains more hypotheses than the neighborhood of any other offset. The RANSAC voting strategy, which selects the offset with the highest inlier count, will consequently select δ^* . $\square$

Corollary 1 (RANSAC Success Condition). *If the fraction of inliers $r = |\mathcal{H}_{\text{in}}|/|\mathcal{H}|$ is greater than 0.5, Theorem 1 guarantees that the RANSAC-based estimation strategy can identify the correct offset δ^* . In practical terms, this condition implies that the majority of tracklet correspondences must be temporally consistent for the estimation to be successful. The probability of RANSAC success increases exponentially with the fraction of inliers.*

Beyond correctness, the proposed framework offers a significant computational advantage over existing exhaustive search methods. We formalize this advantage as follows:

Proposition 1 (Complexity Reduction). *The proposed algorithm reduces the complexity of the search for the temporal offset from $O(F_1 \times F_2)$ to $O(T_1 \times T_2)$, where F_i is the number of frames in video i , and T_i is the number of significant tracklets in video i . Since $T_i \ll F_i$ in typical scenarios (empirically, $T_i < 0.1 \times F_i$), the reduction in the search space is substantial.*

Proof. An exhaustive frame-level search would compare every frame from video V_1 with every frame from video V_2 to find the best temporal alignment, resulting in $F_1 \times F_2$ pairwise comparisons.

The proposed method, however, operates at the tracklet level. It first filters the data to retain only significant tracklets (those with duration $\geq m$ frames, as defined in Equation 3.3). The core matching step, which constitutes the computational bottleneck, compares each significant tracklet from V_1 with each significant tracklet from V_2 , resulting in $T_1 \times T_2$ pairwise comparisons.

As tracklets are far less numerous than total frames (the "significant trajectories" filter discards short, fragmented, and irrelevant tracklets), we have $T_i \ll F_i$, and consequently $O(T_1 \times T_2) \ll O(F_1 \times F_2)$. The empirical reduction factor is reported in Section 6.4. $\square$

3.1.2 Problem Formulation

Let V_1 and V_2 be two UGVs of the same sports event, recorded from different perspectives by independent mobile devices, without shared timestamps or audio cues. Each video V_i contains F_i frames, where F_i can range from hundreds to thousands depending on the duration and frame rate of the video. The goal of timestamp-free synchronization is to find a temporal offset δ such that:

$$\delta = f_1^* - f_2^*, \quad (3.1)$$

where $f_1^* \in V_1$ and $f_2^* \in V_2$ are frames that correspond to the same real-world instant. Once δ is estimated, the overlapping fragments can be extracted from both

videos aligned on a common timeline.

To estimate δ without timestamps, the framework relies on the trajectories of the objects that are tracked in each video. For each video V_i , the multi-object tracker (i.e., MOT-UGV [Alencar and de Freitas 2025]) produces a set of tracklets:

$$T_i = \{ traj_{i,1}, traj_{i,2}, \dots, traj_{i,n_i} \}, \quad traj_{i,k} = (t, b_t)_{t \in \mathcal{F}_{i,k}}, \quad (3.2)$$

where n_i is the number of tracklets in video i , $\mathcal{F}_{i,k}$ is the set of frames in which object k is tracked, and b_t is its bounding box at frame t .

Instead of comparing all frames of one video against all frames of the other, the framework first keeps only the most significant trajectories, that is, the tracklets that are tracked for at least a minimum number of frames m :

$$\hat{T}_i = \{ traj_{i,k} \in T_i : |\mathcal{F}_{i,k}| \geq m \}. \quad (3.3)$$

It is worth to mention that this selection is an important point of the proposed framework, because the frame-level comparison is performed only inside the trajectories that are kept. In this way, the search is reduced from $O(F_1 \times F_2)$ to $O(|\hat{T}_1| \times |\hat{T}_2|)$, which is much smaller in practice.

For each detection of a significant tracklet, a visual feature vector is extracted from the object image crop through an embedding function φ :

$$e_t^{(i,k)} = \varphi(I_i(t, b_t)) \in \mathbb{R}^d, \quad \varphi \in \{ \varphi_{\text{OSNet-AIN}}, \varphi_{\text{DINOv2}} \}, \quad (3.4)$$

where $I_i(t, b_t)$ is the image crop of the object at frame t . In this work, φ is treated as a replaceable visual representation: it can be the identity-specialized ReID encoder $\varphi_{\text{OSNet-AIN}}$, already obtained during tracking, or the self-supervised ViT encoder φ_{DINOv2} , which produces a single descriptor of its input through the [CLS] token.

In the context of φ_{DINOv2} , the holistic term refers to what the descriptor encodes, and not to the spatial extent of what it sees. The two encoders differ in their training objective. OSNet-AIN is trained with identity supervision, so any visual evidence that does not help to discriminate one identity from another, in particular the background and the surrounding context, is treated as nuisance and is suppressed by the objective itself. DINOv2 is trained in a self-supervised way, without identity labels, and its [CLS] token aggregates all patch tokens of the input through global

self-attention. As a consequence, it retains the scene content, the pose and the spatial arrangement that fall inside the crop region, instead of only the cues that identify the object. The crop is also slightly expanded by a fixed padding margin before the extraction, which controls how much surrounding context enters the descriptor. Each tracklet is then represented by its ordered sequence of embeddings $\mathbf{e}_{i,k} = (e_t^{(i,k)})_{t \in \mathcal{F}_{i,k}}$.

Following a retrieval-like setup [Guo et al. 2026], the significant trajectories of V_2 are used as the index and the trajectories of V_1 as queries. Not every pair of trajectories is compared; a candidate pair is only formed when both tracklets carry the same object class label c , which prevents, for example, a tracklet of the ball from being matched against a tracklet of a player. The detector adopted in Stage I is open-vocabulary and is driven by a flat set of natural language prompts $\mathcal{C}$, defined for the sports domain of this work as $\mathcal{C} = \{ \text{"person playing"}, \text{"audience"}, \text{"sports ball"} \}$, where "audience" acts only as a suppression prompt and its detections are discarded before tracking. These class labels is an index into a flat, task-defined prompt vocabulary, with no hierarchy or formal semantics between the prompts. In this context, c is simply the index of the prompt that the detector assigned to the object, propagated by the tracker to every detection of the tracklet. The constraint is therefore a plain equality test $c_a = c_b$ over a task-defined vocabulary, adopted as a filter to reduce spurious correspondences.

Given two tracklets $traj_{1,a}$ and $traj_{2,b}$ that satisfy this constraint, their similarity $S(traj_{1,a}, traj_{2,b})$ is computed from the embedding sequences, either as the minimum point-to-point Euclidean distance between their frames or through Dynamic Time Warping (DTW), which compares the tracklets as temporal sequences instead of isolated frames. The pair of most similar frames (t_1^*, t_2^*) of a matched tracklet pair provides a candidate offset hypothesis:

$$\delta_{a,b} = t_1^* - t_2^*. \quad (3.5)$$

Since a single tracklet pair can produce an incorrect temporal hypothesis, the final offset is not estimated from one match alone. Instead, the framework adopts a RANSAC-inspired consensus strategy over the set of offset hypotheses generated by the most reliable tracklet correspondences. Let $\mathcal{P}$ denote the set of candidate tracklet

pairs, where each pair $(a, b) \in \mathcal{P}$ produces an offset hypothesis $\delta_{a,b}$ and an associated visual feature distance $d_{a,b}$. From this set, the top- K pairs with the smallest feature distances are selected as anchor hypotheses, forming the set of hypothesis $\mathcal{H}_K$.

The final synchronization offset is then selected as:

$$\delta^* = \arg \max_{\delta_h \in \mathcal{H}} \left| \left\{ (a, b) : |\delta_{a,b} - \delta_h| \leq \Delta \right\} \right|, \quad (3.6)$$

where Δ is a tolerance in frames and the selected δ^* is the hypothesis supported by the largest number of consistent tracklet pairs (inliers). Ties are broken by the smallest mean feature distance among the inliers, and, when no consensus is found, the framework falls back to the best-match or the median offset, in this order.

It is important to note that this is not the same as an unconstrained retrieval problem, such as the one solved by RAG [Lewis et al. 2020] or VLM-based approaches [Guo et al. 2026], where each query is matched independently. In the synchronization problem, the matches must respect a temporal relation: the same offset δ^* must hold across the entire sequence of overlapping frames, imposing a temporal-coherence constraint on the selected matches. This estimated δ^* finally defines the set of scenes that overlaps between the two videos.

In this context, synchronization is the process of finding sets of frames, one from each input unsynchronized video, such that the similarity score among the Frame-level visual representation features of each set of tracklets is higher, indicating the detection of an overlapping scene based on the estimated offset.

Figure 3.1 presents the process as a workflow-oriented block diagram, highlighting the method’s modular structure, its connections, and the complete workflow from start to finish. Each block represents a process in the pipeline: from multi-object tracking using MOT-UGV [Alencar and de Freitas 2025], through tracklet embedding and similarity computation, to offset estimation and overlap detection.

In summary, each process works as follows:

- **Stage I – MOT-UGV-based Pre-processing.**

1. The deep learning-based multi-object tracker (MOT-UGV) is applied to the two input videos.

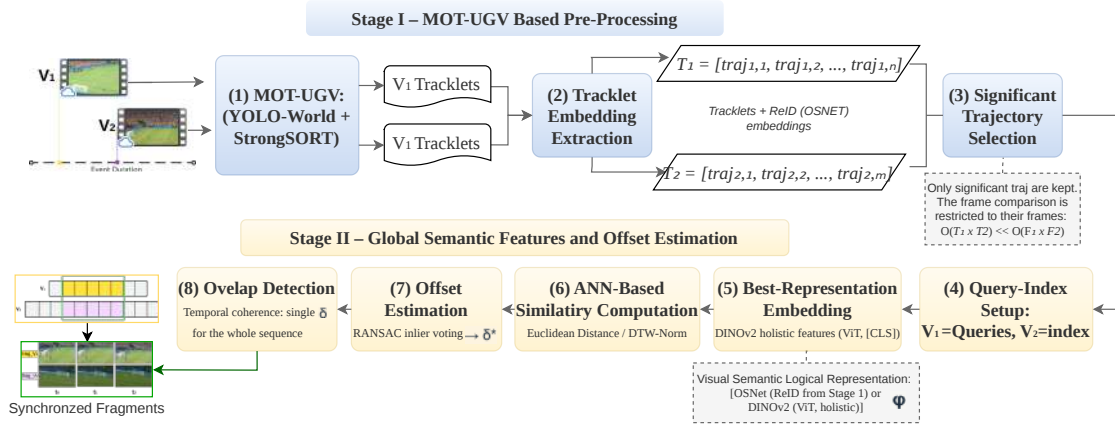


Figure 3.1: Diagram workflow of the proposed framework for synchronizing multi-view UGVs. In Stage I, both videos are tracked by MOT-UGV (1) and only the most significant trajectories are selected (3). In Stage II, the significant trajectories of V_2 are used as the index and those of V_1 as queries (4); a best visual representation is extracted (5); the similarity is computed to find the most semantically similar frame pairs (6); the temporal offset is estimated by RANSAC inlier voting (7); and the overlap detection enforces a single offset over the whole sequence, producing the synchronized fragments (8).

2. As a result of (1), a set of tracklets together with the ReID (OSNet-AIN) appearance embeddings of the tracked objects from each video is returned.
3. From these tracklets, only the most significant trajectories are selected (Equation 3.3). This is the point that reinforces the idea that this proposed approach does not compare all frames of one video against all frames of the other to find the offset. In this case, frame-level comparison is performed only for frames that belong to the selected trajectories.

- **Stage II – Global Semantic Features and Offset Estimation.**

4. The significant trajectories of V_2 are used as the index and the trajectories of V_1 as queries, in a retrieval-like setup.
5. A visual representation is extracted for the frames of these trajectories with a DINOv2-based approach. Both representations receive the same

input, the image crop delimited by the bounding box of the tracked object, what distinguishes them is not the extent of the input but what each descriptor retains from it: the ReID embedding keeps what discriminates identity, while the DINOv2 descriptor summarizes the visual content of the whole input region. The same step is also instantiated with the ReID (OSNet-AIN) embeddings obtained in Stage I, which is the baseline condition used in the comparison (Equation 3.4).

6. The similarity between the trajectories is computed (Euclidean distance and DTW) to find the best correlation pairs, that is, the frames of the trajectories that are the most similar semantically.
7. A RANSAC inlier voting is applied over the offset hypotheses of these pairs to obtain the best offset δ^* (Equation 3.6).
8. The overlap detection applies the temporal relation that characterizes the synchronization problem. Differently from a VLM-based retrieval [Guo et al. 2026], where each query is matched independently, here the estimated offset must be the same for the whole sequence of frames, which imposes a temporal-coherence constraint over the matches.
9. The result is the set of synchronized fragments between the two videos.

3.2 Related Work

This section lists the main works that are strongly related to the objective proposed in this research. It is divided into three main topics: Video Synchronization Based on Visual Features, Multi-Object Tracking Algorithms, and Evaluation Datasets.

3.2.1 Video Synchronization Based on Visual Features

This sub-section provides a chronological overview of the works most closely related to the synchronization of multi-view user-generated videos using multi-object tracking. The following tables list these works, beginning with those that are not deep learning-based (Table 3.1) followed by the ones based on Deep Learning models (Table 3.2), summarizing the main characteristics of the proposed methods.

In summary, existing visual-based synchronization methods demonstrate that temporal alignment can be estimated from video content without relying exclusively on timestamps or sensor information. However, most approaches use global frame representations [Yin et al. 2022, Zhang and Chan 2022, Boizard et al. 2023, Liu et al. 2024], predefined temporal search intervals [Shin and Molybog 2025], or controlled recording conditions [Whitehead et al. 2005, Meyer et al. 2008, Brito et al. 2011]. Moreover, the literature provides limited investigation of object-centered representations and of how substantial viewpoint differences affect synchronization in real multi-view sports UGVs.

Classical Visual Synchronization Methods

Most of the works listed in Table 3.1 propose synchronization methods that rely heavily on background subtraction and frame differencing techniques. While these approaches proved effective in controlled environments [Bruto et al. 2011] and were groundbreaking at the time, they often struggled to handle the variability and complexity introduced by dynamic backgrounds and the lack of timestamps in UGVs [Guggenberger 2023].

As examples of this are the approaches proposed by Whitehead et al. 2005 and Meyer et al. 2008. These are considered baselines for trajectory-based video synchronization, focusing on extracting tracklet-based visual features using classic object tracking techniques, such as the KLT tracker and SIFT. However, they were designed for videos that have been recorded under certain restrictions, such as the fact that the objects being tracked (where, in most cases, only a single object is tracked) should not have movements similar to a pendulum [Whitehead et al. 2005], which suggests that the object cannot leave and enter the scene again. This cannot be guaranteed for user-generated sports videos, which are the focus of this work.

As research in video synchronization advances, more advanced techniques have emerged, incorporating multiple object tracking (MOT) capabilities and leveraging sensor data, as seen in the work of Brito et al. 2011 and Gaddam et al. 2018. These methods exemplify a transition towards more dynamic and context-aware synchronization, using multi-object trajectories and external sensors to improve alignment accuracy [Gaddam et al. 2018]. However, they still face challenges related to adapt-

Table 3.1: Summary of works about visual-based video synchronization and its relation to UGVs and MOT features.

Classical Visual Synchronization Methods			
Title & Reference	UGV	MOT	Sync Method
Temporal Synchronization of Video Sequences in Theory and in Practice [Whitehead et al. 2005]	✗	✓	Method: KLT Tracker, consensus matching. Evaluation: Real-world, synthetic. Focus: Stationary cameras.
Subframe Temporal Alignment of Non-Stationary Cameras [Meyer et al. 2008]	✗	✓	Method: SIFT detector, KLT algorithm. Focus: Trajectory and time matching.
Temporal Synchronization of Non-Overlapping Videos Using Known Object Motion [Brito et al. 2011]	✗	✓	Method: WSL tracker, 3D projection. Focus: Temporal relationships using high-dimensional space.
Camera Synchronization for Panoramic Videos [Gaddam et al. 2018]	✗	✓	Method: Hardware-based trigger, sensor tracking. Focus: Sports analysis, timestamp info used.
Multimodal Alignment of Videos Toward Multimodal Data Synchronization [Guggenberger 2023]	✓	✗	Method: SIFT, image retrieval. Dataset: Jiku Mobile.

ability and scalability, particularly in the presence of content that is constantly becoming more diverse and unpredictable, such as the ones seen in UGVs.

Deep Learning-Based Sync Methods

With the recent introduction of Deep Learning models, the field has seen significant advances, offering more robust and adaptable solutions for video synchronization. The works listed in Table 3.2 exemplify these changes, employing, for example, convolutional neural networks and unsupervised learning techniques to achieve synchronization without explicitly relying on temporal data [Boizard et al. 2023]. However, there are works such as Nazare et al. 2018, which combine visual information, in this case the trajectories of detected objects, with sensor data extracted from external hardware to achieve synchronization, but this forces the system to rely on timestamp information to align the data that will be combined.

Most of the works listed in Table 3.2 use deep learning approaches to visual feature extraction, such as Transformer-based architectures and pose estimation networks [Yin et al. 2022, Liu et al. 2024]. Even with their advances, these mod-

Table 3.2: Summary of works about DL-based approaches applied to video synchronization and its relation to UGVs and MOT features.

Deep Learning-Based Sync Methods			
Title & Reference	UGV	MOT	Sync Method
Content-Based Multi-Camera Video Alignment using Accelerometer Data [Nazare et al. 2018]	✗	✓	Method: Accelerometer data, Faster R-CNN, mean-shift tracking. Focus: Align tracklets with accelerometer signals.
Self-Supervised Human Pose Based Multi-Camera Video Synchronization [Yin et al. 2022]	✗	✗	Method: SeSyn-Net, HRNet (key points). Focus: Feature embedding.
Single-Frame-Based Deep View Synchronization for Unsynchronized Multicamera Surveillance [Zhang and Chan 2022]	✗	✗	Method: FCN-7, content and motion flow matching. Focus: DNN-based models, image correspondence.
Deep Learning-Based Stereo Camera Multi-Video Synchronization [Boizard et al. 2023]	✗	✗	Method: CNN-Siamese, Triplet Loss. Focus: Matching matrix, delay estimation.
Learning Domain-Adaptive Landmark Detection-Based Self-Supervised Video Synchronization for Remote Sensing Panorama [Mei et al. 2023]	✗	✓	Method: Unsupervised learning, matrix diagonal similarity. Focus: Temporal offsets, synthetic datasets.
Advancing Video Synchronization with Fractional Frame Analysis: Introducing a Novel Dataset and Model [Liu et al. 2024]	✗	✗	Method: Transformer, InSynFormer model. Focus: HRNet, IFID dataset, intra-frame alignment.
Beyond Audio and Pose: A General-Purpose Framework for Video Synchronization [Shin and Molybog 2025]	✗	✗	Method: Self-supervised video embeddings (CARL, InternVideo2), per-frame similarity matrix, CNN-based offset classifier. Focus: General-purpose sync, pose/audio-independent. CMU Panoptic.

ern methods maintain a conceptual connection to previous approaches, as they are still based on the fundamental idea of aligning videos by matching visual features extracted from sequences of frames [Zhang and Chan 2022, Mei et al. 2023], which underscores their effectiveness.

Despite these advances, it is important to note that none of these deep learning-based methods explicitly addresses the challenges presented in UGVs. Their focus remains primarily on controlled or synthetic datasets [Mei et al. 2023]. Among the works mentioned in this subsection, only Guggenberger 2023 specifically addresses UGV synchronization, highlighting the importance of tailoring synchronization techniques to the unique characteristics of such videos. However, despite recognizing the

potential of deep learning-based approaches, his proposed method relies entirely on classic image retrieval solutions, such as SIFT. This suggests a gap between the potential benefits of deep learning techniques and their application to UGV challenges [Guggenberger 2023].

More recent work also moves toward replacing task-specific features with general-purpose representations [Shin and Molybog 2025]. Instead of relying on human pose or audio, the authors evaluate self-supervised video embedding models, such as CARL and InternVideo2, feeding a similarity matrix into a classifier to predict the synchronization offset. This is the closest work to the representation question raised in this research, since it explicitly compares different families of visual representations for the synchronization task.

However, it still operates at the level of the whole frame and is evaluated on controlled multi-view datasets, such as NTU RGB+D and CMU Panoptic, without addressing the challenges of user-generated videos or exploring object-centric tracklet features, which are the focus of this work. It is also worth noting that the same authors report that general-purpose VLMs perform no better than a naive guess on precise temporal alignment, which is consistent with the idea that UGVs synchronization requires temporal coherence rather than independent semantic retrieval [Shin and Molybog 2025].

To conclude, while traditional methods provide a foundation, they may not have fully exploited the capabilities of the proposed techniques to deal with the challenges present, especially in UGVs. Moreover, all these methods take the visual representation as a fixed choice and align the videos by matching features extracted exclusively from the whole frames, without questioning which kind of representation is more suitable for the cross-view alignment of videos. This research seeks to contribute to this gap not only by integrating a deep learning-based multi-object tracking approach to explore the features of multi-view UGVs, but also by investigating which visual representation better supports this alignment. This is why it was necessary to survey the state of the art of deep learning-based models for tracking multiple objects, since the tracking stage is what produces one of the representations under investigation.

3.2.2 Deep Learning-Based Multi-Object Tracking Algorithms

Table 3.3 presents a chronological overview of key works on multi-object tracking, highlighting their characteristics in multi-view setups. These works demonstrate the evolution of deep learning-based multi-object tracking algorithms, which initially focused on controlled environments where static cameras are used for tasks like pedestrian tracking, as seen in the MOT16 and MOT20 datasets [Milan et al. 2016, Dendorfer et al. 2020].

Table 3.3: List of works about multi-object tracking and its relation to multi-view user-generated videos.

Multi-Object Tracking Methods (None UGV-based)			
Multi-View	Title & Reference	Evaluation Dataset	Approaches
✗	TrackFormer: Multi-Object Tracking with Transformers [Meinhardt et al. 2022]	MOT17 and MOTS20 (pedestrians).	Tracking-by-attention: Transformer-based. Feature extraction: ResNet-50 (CNN). Track Prediction: Attention mechanism.
✗	MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pre-trained Object Detectors [Zhang et al. 2023]	DanceTrack, BDD100k, and MOT17.	Tracking-by-detection: MOTR (DETR + ResNet-50). Extra object detector: YOLOX.
✗	StrongSORT: Make DeepSORT Great Again [Du et al. 2023]	MOT17 and MOT20.	Tracking-by-detection paradigm. Detector: YOLOX-X. Appearance Feature extraction: BoT + EMA. Prediction: NSA Kalman Filter.
✓	Multi-camera Multi-object Tracking: A Review of Current Trends and Future Advances [Amosa et al. 2023]	DukeMTMCT, CityFlow (pedestrian and vehicle objects).	Survey of 30 MCMT algorithms. Motion and appearance models. Hierarchical techniques dominate.
✓	Online Multi-camera People Tracking with Spatial-temporal Mechanism and Anchor-feature Hierarchical Clustering [Cherdchusakulchai et al. 2024]	2024 AI City Challenge Track 1 (synthetic scenes).	MOT: YOLOv8 + OSNet (Re-ID) + ByteTrack. MTMC: Merges tracklets.
✓	GMT: A Robust Global Association Model for Multi-Target Multi-Camera Tracking [Fan and Zhao 2024]	VisionTrack (recorded by drones).	Tracking-by-attention: Global MTMC. Detector: CenterNet. Feature extraction: Re-ID (Mask R-CNN). Association: Hungarian algorithm.
✗	Iterative Scale-Up ExpansionIoU and Deep Features Association for Multi-Object Tracking in Sports [Huang et al. 2024]	SportsMOT and SoccerNet-Tracking (players).	Tracking-by-detection (sports scenarios). Detector: YOLOX + OSNet (Re-ID). Track prediction: Expansion IoU.

The use of static cameras is evident in works such as “TrackFormer” and “MOTRv2”, which introduced innovations in detection-based tracking, using attention mechanisms and adaptive filtering to improve object association between video frames

[Meinhardt et al. 2022, Zhang et al. 2023]. Over time, these methods have been extended to more complex scenarios, such as sports events [Huang et al. 2024], including multi-view camera setups [Amosa et al. 2023]. Furthermore, all the studies listed emphasize that MOT is a fundamental task in the field of computer vision [Huang et al. 2023].

As demand for MOT has increased, particularly for dynamic environments such as sports and UGVs, new challenges have emerged. According to Huang et al. 2023, multi-object tracking in sports scenarios presents two main challenges: (1) the nonlinear movement of players and (2) the similar appearance of athletes on the field. Thus, MOT for unpredictable environments can present unique challenges. In the work proposed by Huang et al. 2024, these challenges are addressed by replacing the Kalman filter with the iterative ExpansionIoU technique and deep feature extraction. Despite these advances, methods like that can still face limitations when applied to UGVs, where variable capture conditions, such as lighting and camera angles, introduce additional tracking difficulties.

Furthermore, the application of MOT to multi-view environments has also been studied [Amosa et al. 2023, Cherdchusakulchai et al. 2024]. Such approaches require advanced solutions to address issues such as occlusion, perspective variation, and temporal synchronization between cameras, because it is assumed that processed videos are already temporally aligned [Cherdchusakulchai et al. 2024]. Additionally, the datasets used to evaluate these techniques, such as DukeMTMC and CityFlow, are composed of video of pedestrians recorded in controlled environments, limiting the applicability of these methods to more dynamic and complex scenarios.

It is also important to note the type of visual representation on which these tracking methods are based. To associate detections across frames, modern MOT pipelines extract appearance embeddings using re-identification (ReID) networks, such as OSNet [Zhou et al. 2022], which is also the case of the sports-oriented methods listed above [Cherdchusakulchai et al. 2024, Huang et al. 2024]. These embeddings are optimized to discriminate the identity of each object, that is, to distinguish one athlete from another even when they look similar.

However, the cross-view synchronization of UGVs does not require identifying who the object is, but rather recognizing that two videos are observing the same

moment of the scene from different viewpoints. This distinction motivates us to contrast identity-specialized ReID embeddings with a different family of representations. The holistic and visual semantic-logical features produced by self-supervised Vision Transformers [Dosovitskiy 2020], such as DINO [Caron et al. 2021] and DINOv2 [Oquab et al. 2023], which are trained without identity labels and tend to capture the global semantic content of the image [Shin and Molybog 2025].

3.2.3 Evaluation Datasets

Table 3.4 presents a list of related works that use datasets composed of multiple videos for evaluating multi-object tracking or synchronization methods, with an emphasis on those that use user-generated videos (UGVs) and multi-view perspectives. The works are ordered chronologically, starting with those that explicitly use UGV, followed by those that do not. Most of these studies are applied to the validation MOT models for single-view scenarios, where none of these are UGV-based and they have no sensor data. There are also those that were created to be used as ground-truth for the task of synchronizing UGVs [Guggenberger et al. 2015].

The first work listed [Cricri et al. 2013] explores multimodal analysis to classify sport genres in UGV videos recorded with mobile devices. Although the dataset includes Multi-View UGVs and sensor data, such as accelerometers, the focus is primarily on video classification rather than video synchronization or multi-object tracking, which limits its applicability to the context of this research.

Similarly, the work of Guggenberger et al. 2015 offers a set of multi-view UGVs recorded at public concerts, and can be used to evaluate video synchronization approach once it was manually annotated for such task. Although useful for video synchronization, the dataset lacks the object-feature annotations needed to evaluate tracking algorithms specifically. Furthermore, Bailer et al. 2015 and Perera et al. 2020 introduce a dataset for benchmarking and multi-view applications, but combine content generated by professionals and users.

In contrast, works such as Ristani and Solera 2016, Dendorfer et al. 2020, and Tanaka et al. 2022 focus explicitly on multi-object tracking, but do not use user-generated videos. These studies use static cameras and controlled environments, which differ from the context of UGV. The latest published work combines GPS-

Table 3.4: Summary of works that introduce a dataset to MOT or Synchronization evaluation and its relation to Multi-View UGVs.

Datasets for Multi-Object Tracking (MOT) and Video Synchronization in UGVs					
UGV	Title & Reference	Perspective	Sensor Data	MOT	Work Description
✓	Multi-sensor fusion for sport genre classification of user-generated mobile videos [Cricri et al. 2013]	Multi-View	✓	✗	Combines data from mobile video recordings and sensor data (e.g., accelerometer). The application focuses on video classification using SVM-based techniques.
✓	A Synchronization Ground Truth for the Jiku Mobile Video Dataset [Guggenberger et al. 2015]	Multi-View	✓	✗	Includes user-generated videos at events, aligned manually by audio samples for synchronization. The application is for benchmarking and algorithm development.
✓	Multi-sensor Concert Recording Dataset Including Professional and User-generated Content [Bailer et al. 2015]	Multi-View	✓	✗	Combines professional and UGC concert recordings. Synchronization is performed using both audio and visual cues for multi-view video synchronization research.
✗	Performance Measures and a Data Set for Multi-Target Multi Camera Tracking [Ristani and Solera 2016]	Multi-View	✗	✓	Outdoor multi-view videos with bounding boxes and unique IDs for tracking. Used to benchmark tracking algorithms in multi-camera setups.
✗	A Multiviewpoint Outdoor Dataset for Human Action Recognition [Perera et al. 2020]	Multi-View	✗	✗	Multi-view outdoor human actions, including YouTube-sourced videos, annotated with action labels and bounding boxes for training action recognition models.
✗	MOT20: A Benchmark for Multi-object Tracking in Crowded Scenes [Dendorfer et al. 2020]	Single-View	✗	✓	Crowded scene videos annotated for object detection and tracking with bounding boxes and unique IDs across frames for benchmarking tracking algorithms.
✗	MultiSports: A Multi-Person Video Dataset of Spatio-Temporally Localized Sports Actions [Li et al. 2021]	Multi-View	✗	✓	Sports actions with multiple players annotated with bounding boxes and action labels. Used for action detection and tracking in sports scenarios.
✗	Updating a Dataset of Labeled Objects on Raw Video Sequences with Unique Object IDs [Tanaka et al. 2022]	Single-View	✗	✓	Includes object IDs and bounding boxes for tracking, with manual correction for evaluating tracking consistency across frames.
✗	SoccerTrack: A Dataset and Tracking Algorithm for Soccer with Fish-Eye [Scott et al. 2022]	Single-View	✗	✓	Fish-eye videos with player positions and bounding boxes. Focuses on tracking under distortion with validation using GNSS alignment and YOLOv5 detection.

generated data with object detections to track players in soccer videos, but is limited to a single view and does not address the challenge of multiple UGV perspectives, as it is based on official broadcast videos [Scott et al. 2022].

Chapter 4

Methodology

This chapter describes the methodology used to achieve the objectives mentioned in Section 1.1. From the formulation of the problem, the methodological procedures are structured into three main processes: (i) data collection and data generation; (ii) model selection and implementation of multi-object tracking algorithms; and (iii) operation, monitoring and maintenance of the feature processing, so that we have the tracklet-based UGV synchronization.

4.1 Research Overview

Before going into the methodological procedures for the experimental part of this work, it is important to highlight the scientific nature of this research. Based on the objectives defined before and also on the designed experiments, this work is mainly characterized as **applied research**, focusing on the practical application of deep learning-based multi-object tracking techniques used to extract visual features to synchronize user-generated videos.

Given this context, this work has a **descriptive** methodology that aims to describe how visual tracking features extracted from the deep learning-based model can be used in a visual-based synchronization framework. In this context, this work aims to apply theoretical knowledge to develop a framework that is able to synchronize UGVs using tracklet features extracted by MOT-based approaches, especially adapted for multi-view videos captured simultaneously at sports events, without relying on timestamp. To this end, the same pipeline is instantiated with

two visual representations that are compared under identical conditions: the Re-ID embeddings extracted by the MOT model (OSNet-AIN), taken as the baseline, and self-supervised embeddings extracted by a ViT-based model (DINOv2), adopted as the default representation of the framework. This choice is fixed offline and does not depend on the recording condition of the pair, which is used only to stratify the analysis. A more detailed analysis of the behavior of each representation is presented in the results chapter.

In this context, this research is an applied experimental study, combining quantitative evaluation with qualitative error analysis, to evaluate the proposed framework. The quantitative approach involves numerical and statistical analysis of the defined metrics to evaluate the performance of the object tracking-based synchronization frameworks. The qualitative approach focuses on understanding and interpreting the performance of deep learning-based MOT techniques when applied to UGV synchronization.

4.2 Methodological Procedures

The development of a deep learning-based framework requires experiments throughout the development process. Starting from the problem formulation, the methodology begins with data collection and management (Figure 4.1 - I), followed by model selection and its optimization (Figure 4.1 - II), ending with the application and decision-making process (Figure 4.1 - III).

The methodological procedures of this work, illustrated in Figure 4.1, are an adaptation of the one indicated by Russell and Norvig 2020 to be used in machine learning-based project. More details of each of the three main processes are given in the following subsections.

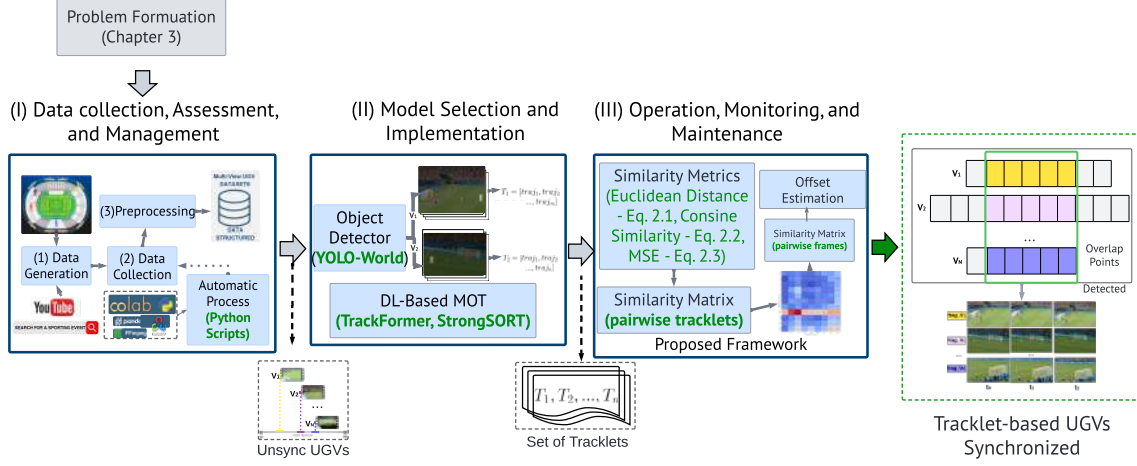


Figure 4.1: Applied Research: general Overview of the processes present in the methodology adopted for synchronizing user-generated videos based on deep learning-based MOT techniques.

4.2.1 Data Collection and Dataset Construction

Once the problem is defined and detailed, the first main process followed in the proposed methodology is the data collection, its assessment, and management, where the idea is to have a custom set of data that follows the characteristics defined for multi-view user-generated videos, that is, the building of a dataset of multi-view user-generated videos from sport events, with overlapping scenes captured from different angles by different users. Taking into account this and also the analysis made in subsection 3.2.3, the overall procedures defined for this phase of the research are illustrated in Figure 4.2.

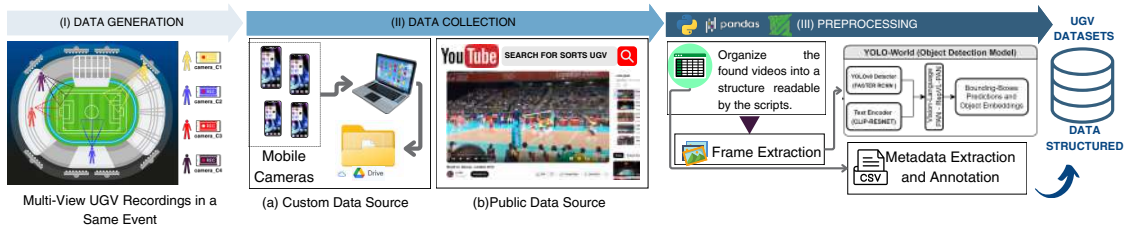


Figure 4.2: Overview of the processes present in the data collection and dataset construction phase. (I) Data Generation, (II) Data Collection, (III) Pre-Processing.

The data generation process is the phase of recording videos by users present in a sports environment, such as a soccer stadium (Figure 4.2 - I). In the context of this research, this process was carried out both by unknown users who made their videos public via YouTube’s Creative Commons category, and by members of the SWPERFI project (including the author of this dissertation) in partnership with Motorola Mobility’s Lab312 research group, who collected the data collaboratively.

The data is then collected (Figures 4.2 - II) from their respective sources, either by extracting (via USB-cable) the videos recorded with devices provided for collaborative recording (Custom Data Source) or by extracting the videos found on YouTube (using the public YouTube Data API accessed via a Python script). In the pre-processing phase (Figure 4.2 - III), this data is structured according to a standard to facilitate its manipulation and management through the relationship between the media file and its metadata.

Still in phase III, the annotation process is carried out, combining automatic pre-annotation with systematic manual review, so that the resulting files follow the standard MOT format and remain compatible with the tracking evaluation tools. The criteria used to decide whether a recording qualifies for these datasets, to confirm that two recordings belong to the same event, and to discard unsuitable material are presented in Subsection 5.1.1, together with the detailed construction workflow of each dataset.

4.2.2 Datasets and their Role in the Experimental Design

Because the proposed framework is training-free, the experimental design of this work does not follow the usual partition into training, validation and test sets. No component of the pipeline is trained or fine-tuned on any of the data described here: the detector, the tracker and the two visual encoders are used with publicly available pre-trained weights, exactly as distributed. Consequently, none of the datasets was used as training data, and all of them play an evaluation role. What must be distinguished instead is which decisions were taken by looking at which data, which is what this subsection makes explicit.

Four collections are used, with distinct purposes:

- **UVY**: a single-view MOT benchmark of sports UGVs, with frame-level bound-

ing box and identity annotations. It supports the adaptation and the evaluation of the tracking pipeline of Stage I, reported in [Alencar and de Freitas 2025], and is *not* used in any synchronization experiment. The StrongSORT parameters and the detection threshold were adjusted on this benchmark and on the public MOT17/MOT20 sequences, that is, on data disjoint from the pairs used to report synchronization results.

- **MUVY**: the multi-view UGV collection from which a subset of 63 pairs was drawn. This is the manually annotated core of the evaluation, and the only subset whose ground-truth offset comes entirely from human annotation.
- **MUVS**: a multi-view collection recorded with sensor timestamp information, from which 468 additional pairs were derived. Here the reference offset follows from the timestamp-based correspondence between fragments, corrected by manual inspection whenever needed, which makes this subset an independent and much larger source of reference offsets, obtained by a different mechanism from the MUVY.

The MUVY-derived 63 pairs and MUVS-derived 468 pairs are unified into a single evaluation manifest, **UGV-Sync**, with 531 pairs, sharing the same schema, the same offset sign convention and the same normalized frame rate (Remark 3.1.1).

Experimental Protocol and Scope of the Analyses

Three groups of decisions must be separated, since they were taken at different moments and on different data. The first group concerns the **tracking pipeline of Stage I**. The class prompt set, the detection confidence threshold and the StrongSORT parameters were established before the synchronization experiments, on the MOT benchmarks mentioned above, and were kept fixed and identical for every pair reported in this work. They were never adjusted per sport, per viewpoint condition or per pair. The second group concerns the **comparison between the two visual representations**. Both representations are evaluated over the same pairs, with the same tracklets, the same significance filter, the same similarity strategies and the same offset estimators. Nothing is tuned in favour of either one, so this comparison

is internally controlled and its conclusion does not depend on a choice made on the data.

The third group concerns **configuration choices that were resolved by inspecting results**, namely the crop padding margin of the DINOv2 extraction and the offset estimator among best-match, median and RANSAC. These were decided by observing performance on the same pairs that are later used to report the final numbers. There is no held-out partition in this work: the evaluation set is small, manually annotated, and splitting it would leave too few pairs per sport and per viewpoint condition for the stratified analysis to be meaningful. This is stated plainly rather than presented as a validation procedure. The comparison between OSNet-AIN and DINOv2 holds under every configuration examined, including every padding value and every estimator, which is why it is the claim the dissertation defends.

Finally, the scope of the evaluation changed substantially after the qualification exam. At that time, the quantitative evaluation relied on a small set of annotated pairs and a single visual representation. The following elements are **post-qualification additions**: the expansion of the manually annotated derived from MUVY; the derivation of 468 further pairs from MUVS and their unification into UGV-Sync; the introduction of the DINOv2 representation and of the controlled comparison against OSNet-AIN; the DTW-based sequence alignment and the RANSAC consensus estimator;

4.2.3 Ground-truth synchronization dataset

In addition to the general dataset construction described above, a specific ground-truth synchronization dataset was built to support the quantitative evaluation and visual validation of the proposed synchronization framework. This was necessary because, at the time of the qualification exam, there was no unified annotated set with frame-level temporal offsets that could be directly used to measure synchronization error in the context of the problem of this work.

To build this validation set, two complementary sources were used. The first is composed of manually selected pairs from the MUVY dataset¹, which were trimmed

¹<https://swperfi-project.github.io/Pages-dev/MuvyDataset-page/>

into short overlapping segments of approximately 15 to 20 seconds. Each pair contains a common scene recorded from different points of view (as illustrated in Figure 4.3). A ground-truth temporal offset between views was manually annotated and organized according to their recording conditions, that is, same-view (SV), different-view near (DV-near) and different-view far (DV-far), following the same definition presented in Subsection 3.1.2.

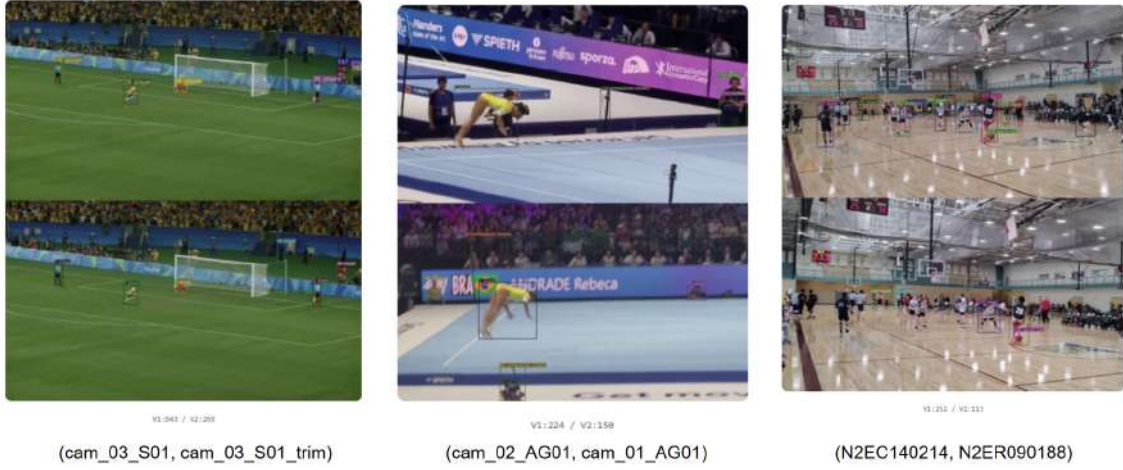


Figure 4.3: Synchronized frame pairs from the UGV sports evaluation set, with bounding-box annotations produced by the MOT pipeline. Each column shows the temporally aligned frames from two views at the manually annotated ground-truth offset.

The second source consists of video fragments extracted from the MUVS dataset², which contain sensor timestamp information. In this case, the initial temporal relationship between cameras is obtained from the timestamp-based synchronization available in the dataset, while the residual cross-camera offset provided in the MUVS annotations is used to define the final ground-truth offset for each generated pair.

The construction and validation of this dataset were supported by a custom inspection tool, **UGV-Sync Studio**. This tool allows the annotator to browse the selected pairs, visualize both videos side by side, switch between the raw clips and the tracking outputs, inspect the estimated offsets produced by the framework, and adjust the reference offset when needed. The currently selected manual offset can be written back to the connected manifest file by updating the `offset` and `offset_sec`

²<https://swperfi-project.github.io/Pages-dev/MuvsDataset-page/>

columns. This makes the tool useful not only for visualization but also for validating and refining the ground-truth annotations used in the experiments.

As illustrated in Figure 4.3, for each selected pair, the annotator verifies whether the frames from V_1 and the frame from V_2 correspond to the same visual event at the annotated offset (e.g., a specific player action or ball position). Pairs for which no reliable common anchor could be visually identified were excluded from the quantitative evaluation.

Annotation Protocol and Reliability

The annotation carried out here is of a different nature from the frame-level object annotation described earlier. What is annotated is not a region or a class, but a single temporal correspondence per pair. The annotator identifies one frame $f_1^* \in V_1$ and one frame $f_2^* \in V_2$ that depict the same real-world instant, and the ground-truth offset follows directly as $\delta = f_1^* - f_2^*$. Only this anchor correspondence is annotated, not every frame of the pair. This is sufficient because of the temporal consistency assumption (Axiom 1): if a single offset holds over the whole overlap, one correctly identified correspondence determines the alignment of the entire sequence. The annotator nevertheless verifies the anchor by playing both clips under the selected offset in **UGV-Sync Studio**, so the correspondence is confirmed over a window rather than on an isolated frame.

The annotation guidelines follow those already established for the MUVY and MUVS collections, including the viewpoint labelling scheme and the criteria for considering that two recordings correspond to the same sporting moment. The anchor is chosen on a visually unambiguous event, such as a ball contact, a landing, or a clearly identifiable player action, visible in both views. Pairs for which no such anchor could be reliably identified were excluded, as described above. Occlusion received no specific treatment at this stage: an occluded athlete simply makes a candidate anchor unusable, and the annotator selects a different instant in the same pair, since a single valid correspondence is enough.

The reliability of these annotations has limits that must be stated. The annotation was carried out by the Vision-AI group, a small team of researchers from the ALGOX/SWPERFI project, familiar with the collections and with the synchroniza-

tion task, and each pair was reviewed visually before being accepted. However, no pair was independently annotated by more than one person, so no inter-annotator agreement was computed, and no annotation error rate was estimated, as this would require a doubly annotated subset that was not produced. The precision of an individual annotation is bounded by the visual granularity of the anchor event: fast actions allow the two views to be aligned within a few frames, while slower or partially occluded actions leave a wider margin of uncertainty. This is one reason why results are reported not only as the mean absolute error but also as accuracy within a tolerance τ , which absorbs annotation uncertainty of a few frames instead of penalizing it as method error. Establishing the reliability of the reference offsets quantitatively, through double annotation of a subset, remains an open point for future work.

4.2.4 Selection and Implementation of DL-based Models for MOT

This phase of the work involved researching the state-of-the-art deep learning-based object tracking techniques available in the literature for implementation and reproduction. Following the analysis made in subsection 3.2.2, priority was given to those models whose method of object tracking is strongly related to the context of this research, in addition to their ability to deal with multiple objects simultaneously.

Another condition considered to select models is the support they have from the community of researchers and developers, which facilitates resolving possible difficulties during the implementation and adaptation phase. It is worth mentioning that, at the time of its conception, the aim of the research was not to train a new MOT model; once the goal is to validate the use of trajectory features for synchronizing UGVs, which is why pre-trained models with available weights and configuration files were utilized for reproducibility.

One of the objectives of this research is to adapt the selected tracking model to address the specific challenges encountered in sports UGVs (as discussed in Section 2.1). Additionally, the model will be adapted to detect and track only objects of interest, such as players on the field and a sports ball. This involves customizing the object detection phase to accurately identify these objects in each frame and

assigning unique IDs to maintain object identity across frames.

4.2.5 Operation: UGV Sync Based on Visual Features

As described in subsection 3.1.2, this phase is the core of the processing pipeline, where the visual features of the videos are processed to detect sets of overlapping frames based on similarity calculations among their features. At this stage of the framework, it is assumed that the object tracking model to extract trajectory information has already been implemented and adapted (subsection 4.2.4). As a result, a set of adjusted video tracklets (tracklet embeddings) is generated for the video synchronization process.

Tracklet Output Format: For each processed frame t , the pipeline writes one line per tracked object to a MOT-format text file, extended with the 512-dimensional appearance embedding $\mathbf{e}_{j,t}$:

$$\underbrace{t, id_j, x, y, w, h, 1, c_j, 1}_{\text{MOT standard}}, \underbrace{e_1, e_2, \dots, e_{512}}_{\text{ReID embedding}} \quad (4.1)$$

where (x, y, w, h) is the bounding box in pixel coordinates, c_j is the class index, and the final 512 values are the OSNet-AIN appearance embedding. This extended format preserves full compatibility with standard MOT evaluation tools while enabling downstream tracklet-based similarity computation.

The operation phase is organized in two stages. The first one is a MOT-based pre-processing, and the second one is responsible for the best representation of the embeddings and for the offset estimation. The formal definition of the synchronization problem and all the equations used in this process are presented in Section 3.1.2, so the description here is focused on the methodological steps of the framework.

Stage I: MOT-based Pre-processing

In the first stage, the MOT-UGV approach is applied to the two input videos in order to obtain their tracklets. After that, only the most significant trajectories of each video are selected. This selection is an important design decision of the framework, because it means that the approach does not perform a frame-by-frame comparison over the entire duration of the videos to find the offset.

The comparison is performed only for the frames that belong to the trajectories that are semantically closest between the two videos, which reduces the search space and the computational cost. The pipeline integrates three components: an open-vocabulary object detector (YOLO-World), a tracking-by-detection algorithm (StrongSORT) and a ReID backbone, all adapted to the specific challenges of sports UGVs [Alencar and de Freitas 2025].

Object Detection: YOLO-World Adaptation: standard object detectors trained on MS-COCO [Lin et al. 2014] are limited to fixed class sets, which prevents them from distinguishing sport-specific roles such as players on the field from audience members in the stands. To address this, the framework adopts YOLO-World/L [Cheng et al. 2024], an open-vocabulary detector that supports arbitrary class prompts via CLIP-based text embeddings at inference time.

The input prompt configured for sports UGV analysis contains the following object classes: "**person playing**" (active players and referees on the field); "**sports ball**" (ball detection for ball-sports events) and "**audience**" (configured as a dummy suppression class to filter crowd detections, which are subsequently removed from the tracking input).

A critical adaptation concerns the detection confidence threshold. YOLO-World returns significantly lower confidence scores for classes outside its COCO pre-training distribution [Roboflow-Team 2024]. Standard thresholds (e.g., 0.25) would suppress most valid detections in this domain. After systematic evaluation, a threshold of $\epsilon = 0.005$ was adopted, combined with Non-Maximum Suppression at $\text{IoU} = 0.45$ to control false positives.

Tracking and ReID: StrongSORT with OSNet-AIN: detection outputs are passed to StrongSORT [Du et al. 2023], a tracking-by-detection algorithm that combines a Kalman filter for motion prediction with a ReID network for appearance-based data association. to ensure robustness under different conditions, a series of iterative fine-tuning experiments was then performed, adjusting key StrongSORT parameters to address specific limitations observed in the baseline. Modifications included reducing the `max_age` parameter to avoid outdated identity persistence in dynamic scenarios, increasing `max_dist` to allow greater flexibility in appearance-based associations under occlusion, and increasing the `n_init` parameter to ensure

greater robustness before confirming tracks.

The ReID backbone was upgraded from the default OSNet-x0.25 to OSNet-AIN x1.0, which incorporates Aggregated Instance Normalization (AIN) that improves feature discrimination across domains, a relevant property when processing UGVs with heterogeneous recording conditions such as varying illumination, motion blur, and resolution. Collectively, these adaptations were designed to systematically explore the trade-offs among detection recall, identity consistency, and robustness across both benchmark datasets and the context of user-generated sports videos [Alencar and de Freitas 2025].

Stage II: Visual-Representation Embedding and Offset Estimation

In the second stage, the trajectories of the second video are used as indices, and the trajectories of the first video are used as queries. The framework predicts the best offset using two complementary visual representations, according to the recording condition of the videos. For same-view (SV) and different-view near (DV-near) pairs, the Re-ID features extracted by the MOT model (OSNet-AIN) are used. For cross-view pairs, that is, different-view far (DV-far), a DINOv2-based approach is used to extract holistic embeddings from the frames of the closest trajectories, since these embeddings are more robust to large viewpoint changes. A more detailed discussion about the behavior of each representation is presented in the results and analysis chapter.

From the selected representation, the similarity between the trajectories is calculated to find the best correspondence pairs, that is, the frames of the trajectories that are semantically similar. For this, two complementary similarity strategies are used: the Euclidean distance (Equation 2.1), which compares the feature vectors point to point, and a sequence-alignment strategy based on Dynamic Time Warping (DTW-Norm), which compares the trajectories as temporal sequences instead of isolated frames. In both cases, smaller values indicate that the objects are more similar. Based on these similarities, the final offset is estimated. Instead of using only the single best match or a simple average of the offsets, which can be easily affected by wrong matches, the offset is selected by a RANSAC-style inlier voting strategy.

Once the offset is defined, the overlap detection is performed. In this step, the temporal relation between the videos is applied. The estimated offset must be the same for the whole sequence of overlapping frames. This temporal consistency constraint is what differentiates the proposed approach from a retrieval-based approach (such as a RAG/VLM pipeline), where each query is matched independently and without a global temporal coherence. Finally, the output of this process is the synchronized fragments of the videos.

4.3 Reproducibility

All the code, configuration files and manifests needed to reproduce the experiments reported in this dissertation are available in a public repository ³. The repository is organized so that each reported experiment corresponds to an executable entry point rather than to a description in prose, and it contains: the implementation of the two pipeline stages; the batch evaluation scripts that produce the reported metrics; the evaluation manifests of the UGV-Sync set, including the reference offset, the frame rate, the viewpoint label and the sport of every pair; the scripts that generate the MUVS-derived pairs and that verify the frame-rate normalization; the UGV-Sync Studio inspection tool; and the analysis scripts that produce the tables and figures of the results chapter.

Two aspects deserve explicit mention. First, every model used is public and pre-trained, and the exact checkpoints and versions are declared in the repository, so no training step needs to be reproduced. Second, the evaluation pipeline writes one result file per pair, containing the estimated offset of each condition, the matched tracklet pairs and the resulting error, which means that the reported aggregate numbers can be recomputed from the stored per-pair outputs without re-running the pipeline. The video files themselves are distributed through the dataset pages of the MUVY and MUVS collections, under their respective licenses, and are not duplicated in the repository; the manifests reference them by relative path. Details of the environment, the dependency versions and the execution order of the scripts are documented in the repository.

³<https://github.com/eltonalencar90/track-ugv-sync>

Chapter 5

Framework Setup and Experiments

This chapter describes the implementation and evaluation stages carried out to support and validate the proposed framework for synchronizing user-generated videos (UGVs) through deep learning-based object tracking features. The chapter is structured into two parts: (i) preliminary setup and data preparation stages, including dataset construction and MOT model integration; and (ii) experiments designed to validate specific components of the framework, such as feature-based similarity metrics and offset estimation.

Additionally, this chapter also introduces the experiments that were carried out after the qualification exam, which focused on the multi-view case, including the comparison of visual representations for cross-view synchronization, the construction of a ground-truth evaluation set, and an error analysis to characterize the conditions in which the framework has more difficulty. Before these experiments, a dedicated section consolidates the models, the development environment and the computational resources used.

5.1 UGV Dataset Preparation

Given the context of applying automated methods to process user-generated videos, which often differ from the recording patterns of professionals, and the analysis of the lack of datasets composed of such videos, the procedures described in this section focus on preparing customized datasets of user-generated videos. These datasets serve as the foundation for the experiments and evaluations carried out in

subsequent sections of the framework. The goal was to construct datasets that reflect the diversity and challenges of real UGVs, including varying quality, resolution, and camera positions.

Figure 5.1 summarizes the steps conducted for each created dataset. Each procedure follows a structured workflow that involves data acquisition, annotation, and organization stages, adapted to the specific goals of the evaluation (single-view MOT, multi-view events, or multi-view with sensors).

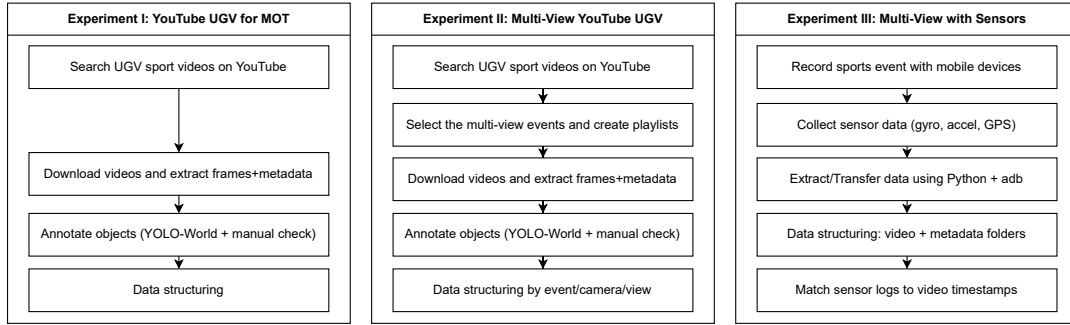


Figure 5.1: Workflow of the three dataset creation: (I) YouTube UGV for MOT, (II) Multi-View YouTube UGV, and (III) Multi-View with Sensors.

5.1.1 Inclusion Criteria and Selection Procedure

Before describing each construction workflow, this subsection states the criteria used to decide whether a recording qualifies for these datasets, how two recordings were confirmed to belong to the same event, and what was discarded.

Operational criteria for a UGV. Definition 2 characterizes a UGV conceptually, but that definition is not directly observable in a video file retrieved from an online platform. In practice, a recording was accepted as a UGV when it satisfied all of the following observable conditions: it was published on a non-institutional channel, that is, not by a broadcaster, a league or a club media department; it consists of a single continuous handheld shot, without cuts between cameras, replays, slow motion or editing typical of broadcast production; it contains no production graphics, such as score overlays, station logos or lower thirds; and the vantage point is one accessible to a spectator, from the stands or the sideline, rather than a dedicated camera position. Camera motion, framing instability, exposure changes and

compression artifacts were not treated as defects, since they are precisely the characteristics the datasets are meant to preserve. The recordings of the third collection (Subsection 5.1.4) are UGVs by construction, since they were captured with handheld consumer devices by non-professional operators under spectator conditions; the only controlled aspect is the prior allocation of recording positions, adopted to guarantee simultaneous multi-view coverage, and this is stated explicitly rather than presented as uncontrolled capture.

Verifying that two recordings belong to the same event. No meta-data was used for this purpose, since upload dates, titles and descriptions of user-generated content are unreliable and frequently absent. For the multi-view collection built from online sources, candidate recordings were first grouped by event through targeted searches and playlist organization, and each candidate group was then confirmed by visual inspection. A group was accepted only when the recordings agreed on multiple independent cues observable in the images themselves: the same venue and field markings, the same team uniforms, the same identifiable athletes, weather conditions, and a consistent progression of the action across views. Recordings whose correspondence could not be established beyond doubt through these cues were not grouped. For the third collection, the correspondence is guaranteed by the acquisition protocol itself, as all devices recorded the same session simultaneously.

Filtering and exclusions. No minimum resolution, duration or bitrate was imposed as an admission threshold, because filtering by technical quality would remove exactly the recordings that make the UGV setting difficult and would bias the datasets toward the easiest cases. The observed heterogeneity is therefore a deliberate property: resolutions range from 640×360 to 1920×1080 , frame rates from approximately 25 to 30 fps, and durations from a few seconds to a little over two minutes. Recordings were nevertheless discarded when they did not contain the content the datasets require, namely: highlight compilations and replay-only clips, since they violate the single continuous shot condition and have no coherent timeline; screen recordings of broadcast material, since the capture device is not the recording device; footage showing only the audience, the venue or pre-game activity, without athlete action. For the multi-view collections, a further exclusion applies at the pair level: a candidate pair was discarded when the two recordings were found

to derive from the same original upload, and when no common visual anchor could be identified between the views, as described in Subsection 4.2.3.

5.1.2 I. Collecting and Annotating YouTube UGVs

During the literature review, no dataset was found that corresponded to what was intended for the experiments to evaluate the proposed framework, that is, unofficial recordings of sports matches (e.g. videos recorded by mobile devices). In fact, most of the main public benchmarks, such as MOT16, MOTS, MOT20, were created for pedestrian tracking or vehicle tracking. To conduct the research, we created a new custom dataset to evaluate object tracking models applied to UGVs.

Following the methodology described in Section 4, Figure 5.2 presents an overview of the five-stage workflow used to construct the UGY dataset: (1) video collection and filtering, (2) pre-processing and frame extraction, (3) automated annotation generation and (4) manual review, correction, and (5) quality assurance.

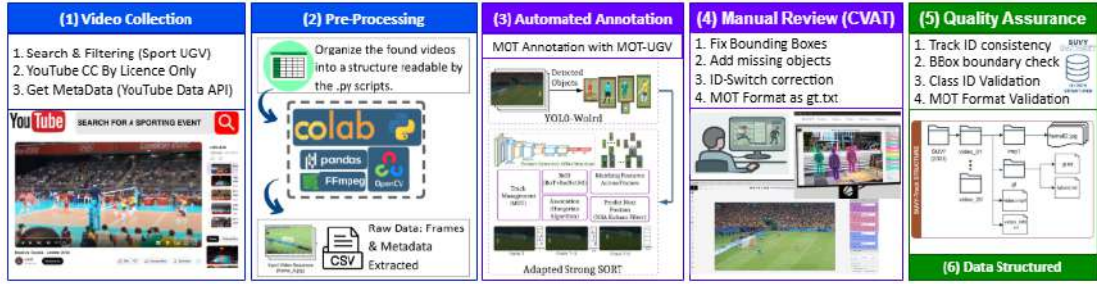


Figure 5.2: Dataset construction pipeline for the UGY dataset. The pipeline illustrates the process from video collection, followed by metadata extraction and frame extraction, automated annotation using YOLO-World and StrongSORT, and manual refinement via CVAT. The resulting dataset is structured in MOT format.

(1) Video Collection and (2) Pre-Processing: videos were collected from YouTube by executing targeted search queries per sport category (e.g., "grass-roots soccer match", "amateur volleyball tournament", "artistic gymnastics training video"). Only videos published under a Creative Commons Attribution license (CC BY) were considered for inclusion, ensuring legal redistribution.

Video download and metadata (i.e., YouTube video ID, title, source URL, duration, frame rate, and resolution) extraction were automated using a Python script

based on yt-dlp¹, an open-source command-line tool and Python library designed to download audiovisual content and retrieve metadata from YouTube. The selection criteria required that the videos contain actual gameplay or athlete activity; highlight reels, replay-only clips, and screen recordings were excluded. After collection, frames were extracted at each video’s native frame rate using FFmpeg 6.0 and OpenCV.

(3) Automated Annotation with YOLO-World + StrongSORT: To automate and accelerate the annotation process, YOLO-World was integrated into the pipeline. As already mentioned, this model can detect arbitrary categories from text prompts. Specifically, we used the "yolo_world/s" variant from the Ultralytics library with the following configured classes: "person playing", "audience", and "sports ball". These classes were selected to reflect key actors in sports scenarios (e.g., players and ball), while "audience" was used as a dummy class to suppress false positives around the stands or crowd areas.

Inference was applied frame-by-frame using a Python script invoking `YOLOWorld.infer()`, filters detections by confidence (set to 0.003) and Non-Maximum Suppression (0.15), and stores results in standard MOT-style format (`<frame_id> <object_class_name> <x1> <y1> <x2> <y2>`) [Cioppa et al. 2022, Dendorfer et al. 2020]. Bounding boxes and class labels were visually verified for quality control during manual review. A very low confidence threshold (0.003) was intentionally set because we noticed that the confidence returned by YOLO-World is significantly lower when querying for classes outside the COCO dataset [Roboflow-Team 2024]

It should be noted that this configuration concerns the *annotation* of the dataset, and is not the same one used inside the synchronization framework. The annotation stage uses the lighter `yolo_world/s` variant with a very permissive threshold, because its output is subsequently reviewed and corrected by hand, so favouring recall over precision reduces the number of missing objects a human has to add. The detection stage of the framework, described in Subsection 4.2.4, uses `YOLO-World/L` with $\epsilon = 0.005$ and NMS at $\text{IoU} = 0.45$, since there is no manual correction downstream and false positives would propagate into the tracklets.

Although YOLO-World provided satisfactory coverage for most of the target ob-

¹<https://github.com/yt-dlp/yt-dlp>

jects, including pseudo-classes such as 'Person playing', 'Sport Ball' and 'Audience', manual review and correction were necessary for all videos, especially to add missing detections such as referees or goalkeepers, which were not detected by the model.

(4) Manual Review and Correction via CVAT: Following automated detection, all annotation files were imported into CVAT (Computer Vision Annotation Tool) for visual inspection. CVAT is an open-source, web-based annotation platform that provides an interface for efficiently reviewing, correcting, and managing annotations on image and video datasets [Corporation 2023]. The review process was applied to all frames of each video sequence and addressed the following categories of annotation error:

1. Inaccurate bounding box dimensions: boxes that under- or over-estimated the spatial extent of athletes were resized to more accurately capture their body silhouette.
2. Identity switches: track IDs incorrectly propagated to a different individual in subsequent frames were corrected.
3. False positive detections: spurious detections caused by crowd elements, advertising boards, referees' flags, or graphical overlays were removed.
4. Missed detections: objects not detected by YOLO-World, particularly referees, goalkeepers, and goal structures, were manually annotated.

These steps allowed for detailed verification and adjustment of the annotations in a user-friendly interface. Once validated, the corrected annotations were exported in the standard MOT format to ensure compatibility with the tracking and evaluation process. This validation phase was needed to address identity switches and false positives problems, which often occurred due to the low confidence threshold used during YOLO-World detection. It ensured that the final annotations were accurate and consistent, especially in challenging visual conditions common in user-generated content.

(5) Quality Assurance and data structure: After CVAT review, annotation files were programmatically validated to confirm: (a) all track IDs are consistent across frames with no gaps longer than 30 frames; (b) bounding box coordinates are

within image boundaries; (c) class IDs correspond to valid entries in the labels.txt file. Sequences with remaining systematic errors (e.g., unreliable ball detection in tennis and cricket due to object speed and motion blur) are flagged in the video_info.txt metadata file.

5.1.3 II. Structuring Multi-View YouTube UGVs

This second procedure focused on collecting multi-camera user-generated videos of the same sports event. The objective was to build a dataset that supports synchronization analysis across independently recorded video streams [Guggenberger 2023]. To this end, the previous workflow was used as a baseline resource to select and extract related content. Videos were searched on YouTube under Creative Commons license, and playlists were created to group multi-view content for the same event (Figure 5.1).

Once playlists were defined, videos and metadata were downloaded and organized by event and camera/view. As in the first dataset created, YOLO-World was used for automatic object detection, the results being stored in standard annotation formats. The information was structured following standard object detection benchmarks. The dataset was structured with subfolders per viewpoint and organized according to sports category and event.

5.1.4 III. Collecting Multi-View UGVs with Sensor Data

As part of the collaboration between SWPERFI Project (UFAM) and Lab 312 Project (Motorola Mobility at Chicago - IL, USA), a dedicated data collection procedure was carried out during our visit to Motorola Mobility’s office. The objective of this workflow was to advance in the creation of a multi-view user-generated video dataset composed of sensor data, captured simultaneously during various sporting events. And the motivation for creating this third dataset is the same as the one mentioned earlier, relating to the challenges and potential of user-generated videos, especially when applied to the context of multi-view analysis.

In addition, the main difference between this dataset and the other two described above is that this data collection includes both video recordings and sensor data, such as gyroscope, accelerometer, magnetometer, and timestamp information, which can

provide a comprehensive set of data for the context of current video synchronization research.

It started by **planning** and **identifying** suitable sporting events to attend while on our visit to Chicago. The SWPERFI researcher members worked closely with Motorola’s team to configure the mobile devices used for the data collection. These devices were equipped with a modified version of the camera app that enabled simultaneous video and sensor data recording.

The modifications to the app were made by Motorola’s camera software development team. The APK version was then shared with researchers to install on the devices used to record the videos. The data collection function was responsible for starting the reading and storage of sensor data (e.g. gyroscope, accelerometer, magnetometer, and GPS) while a video was being recorded. To ensure optimal data capture, the positioning of users around the scene to be recorded was pre-determined (Figure 5.3(a)).

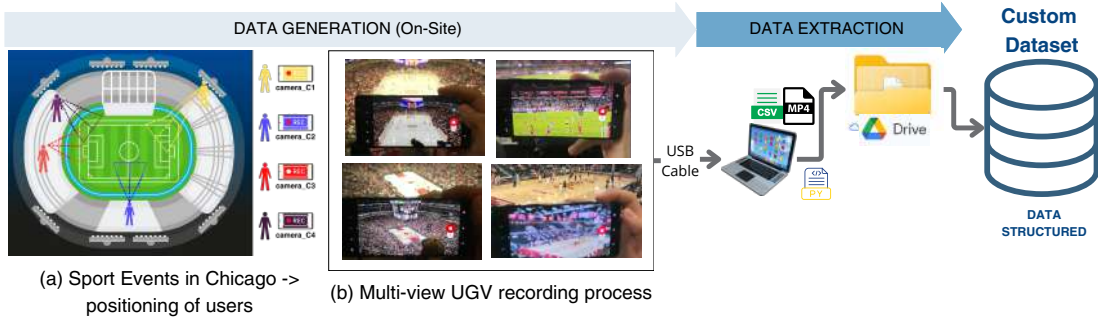


Figure 5.3: Illustration of the Multi-View UGV Dataset with Sensor Data construction.

The acquisition protocol was designed to capture simultaneous recordings of the same event from multiple handheld cameras under real-world conditions. To facilitate multi-view analysis, priority was given to sports played on structured rectangular playing surfaces, such as basketball courts, soccer fields, hockey rinks, and volleyball courts, where multiple spectators could be spatially distributed around the scene while maintaining visual access to the ongoing action. Examples of spectator-operated mobile recordings acquired during live sports events are shown in Figure 5.3(b).

Before each collection session, the event, spectator positions, and device alloca-

tion were planned to increase viewpoint diversity and ensure simultaneous coverage of the same scene. The recordings were performed under natural spectator conditions, without controlling camera motion, framing style, or local illumination, so that the dataset would preserve realistic variability in handheld video capture.

Once the data collection phase was completed, the next step was to reproduce the **extraction and construction** processes, where the video and sensor data were transferred using custom Python scripts with `adb` integration. The dataset was then structured by associating each sensor log file with its corresponding video file and aligning the timestamps. This dataset offers an enriched source for evaluating video synchronization approaches based on both visual and motion data.

With all datasets prepared and annotated, the subsequent sections focus on the application of tracking models and the execution of experiments aimed at validating the key functionalities of the proposed synchronization framework.

5.2 Timestamp-based Temporal Alignment

Temporal alignment across camera streams was established using the absolute timestamps recorded during acquisition (as described in 5.1.4). Since different devices may start and stop recording at different times, the processed data were organized based on overlap periods, defined as time intervals in which at least two cameras were simultaneously active during the same event. These overlap intervals were identified from the temporal boundaries of the videos associated with each event and were used to determine the portions of the recordings that effectively support multi-view analysis. An example of the timestamp-based temporal alignment pipeline for one event is shown in Figure 5.4.

To improve the consistency of the processed dataset, only overlap periods longer than 60 s were retained for downstream processing. These periods define the temporal basis for the next stages of dataset construction, since they represent intervals in which multiple viewpoints are available for the same event and can therefore support camera comparison and selection tasks.

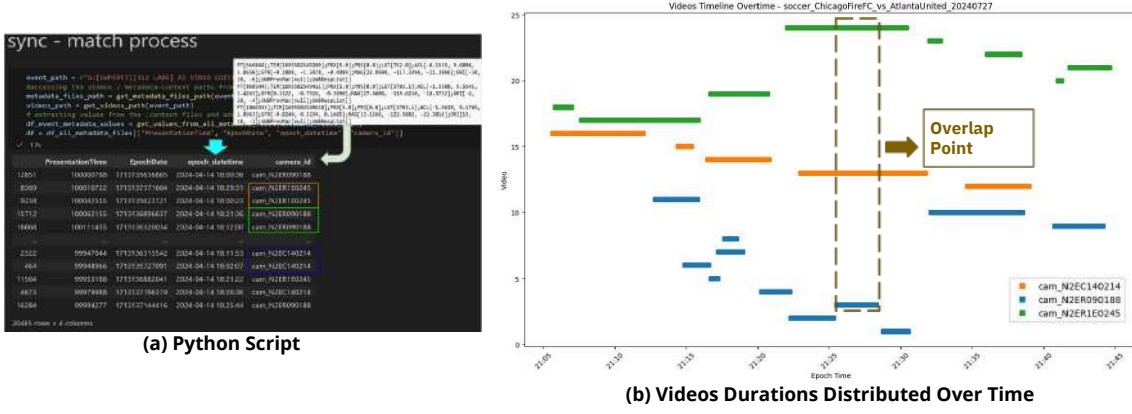


Figure 5.4: Example of the timestamp-based temporal alignment for one event, including the overlap point, where at least two cameras were simultaneously active.

5.3 Integration of MOT Algorithms

This section describes the process of integrating state-of-the-art multi-object tracking (MOT) algorithms into the proposed framework. Rather than developing tracking models from scratch, existing implementations of DeepSORT, StrongSORT, and TrackFormer were adapted to extract tracklets from UGV datasets created before. These models were chosen based on published performance, code availability, compatibility issues, embedding extraction, computational cost, and strong support from the community and forums. These selections aimed to evaluate the performance of such tracking methods in the context of user-generated and multi-camera videos.

The main goal was to establish a functional tracking pipeline capable of generating tracklets for later use in experiments involving tracklet-based synchronization. These steps focused on configuring the detection models, adapting scripts for frame-wise processing, and verifying that the tracking components functioned properly on custom video inputs.

The execution process was conducted in the Google Colaboratory environment to mitigate potential problems with Python library versioning and to leverage GPU support for PyTorch-based models. Each algorithm was tested in separate notebooks, starting with pre-trained weights and adjusting hyperparameters based on preliminary results to optimize performance for our specific datasets.

Detection models (YOLOv5 and YOLOv7) were applied to each frame of the videos to generate bounding boxes around the detected objects. The tracking models

(DeepSORT, StrongSORT) were then used to assign unique identifiers and maintain object trajectories across frames. Both detection and tracking models were iteratively tested with different thresholds and parameters, including confidence scores and association metrics, to improve tracking accuracy. DeepSORT combine Kalman filtering to predict object states with the Hungarian algorithm to solve the assignment problem between predicted tracks and new detections, a strategy that maintains robust tracking in complex and occluded scenarios. StrongSORT, an improved version of DeepSORT, incorporates additional features to better handle occlusions and maintain track consistency. Additionally, TrackFormer, an attention-based tracking method, was included to explore different paradigms of object tracking.

I. Tracking-by-detection: DeepSORT and StrongSORT

To familiarize with object tracking techniques, the DeepSORT algorithm was initially configured using a Kaggle-Code-Repository guide². The detector model was set up to extract tuples containing bounding box coordinates, confidence scores, and class labels for each object in a frame. These tuples were passed to DeepSORT, which the estimation, association, and Tracker ID lifeCycle management. A similar process was followed for StrongSORT (Subsection 2.3.2). Both implementations were prepared using instructions from their respective GitHub repositories³, with execution carried out in Google Colab to ensure compatibility and GPU support. The successful processing of sports videos from the created dataset by both DeepSORT⁴ and StrongSORT⁵ confirmed the setup, as illustrated in Figure 5.5.

II. Tracking-by-attention: TrackFormer

TrackFormer is an object tracking algorithm that uses the concept of Transformer models. Its main contribution lies in the application of self-attention to learn long-range representations and make precise data associations between consecutive frames. This approach makes it possible to capture contextual dependencies be-

²<https://www.kaggle.com/code/nityampareek/using-deepsort-object-tracker-with-yolov5>

³<https://github.com/dyhBUPt/StrongSORT.git>

⁴<https://drive.google.com/file/d/109tnrdW1uVRco2rHu2XwqxXkyjxLO8y1/view>

⁵<https://drive.google.com/file/d/1oWrPohUy1PkkB4Zkb7iT20y5XljG1mjb/view>



Figure 5.5: Confirmation of the successful implementation and adaptation of the execution script for the baseline tracking-by-detection algorithm, StrongSORT.

tween different objects in the scene, resulting in tracking that is more robust to occlusions, changes in appearance and the entry of new objects [Meinhardt et al. 2022]. The algorithm was prepared following instructions from its GitHub repository⁶. A Google Colab notebook was used to configure the code and process sports videos from the created dataset. Successful processing⁷, as shown in Figure 5.6, confirmed the setup.

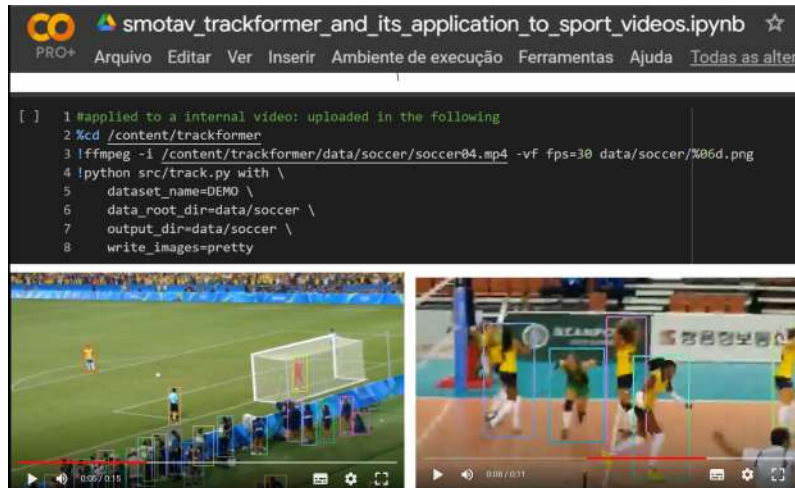


Figure 5.6: Successful implementation and adaptation of the execution script for the baseline tracking-by-attention algorithm, TrackFormer.

⁶<https://github.com/timmeinhardt/trackformer.git>

⁷<https://drive.google.com/file/d/1d1WKbT2T24fZq71kevvAz5zSDfsG7jjz/view>

5.4 Benchmark Evaluation of MOT-UGV

This section presents the experimental benchmarking of the custom tracking framework that integrates the StrongSORT with the YOLO-World as detector. In addition to the proposed configurations, we also reproduced the experimental conditions of the baseline model described by Broström 2022, to be used as a reference point for comparison. The evaluation pipeline follows the MOTChallenge benchmark protocol, using the official TrackEval toolkit to compute the quantitative evaluation metrics introduced before (Section 2.3.3):

- **HOTA**: balances detection and association accuracy in a single score.
- **IDF1**: measures identity preservation across time (higher = better).
- **MOTA**: penalizes missed detections (FN), false positives (FP), and identity switches (IDSW).
- **IDSW**: number of times the identity assigned to a tracked object changes incorrectly.

The evaluation was based on two standard tracking datasets: the MOT20 and MOT17 datasets, which were the ones used by Du et al. 2023 to evaluate their results.

- **MOT17**: Urban surveillance scenes with moderate crowd density and multiple views per sequence.
- **MOT20**: Highly crowded scenes with frequent occlusion, ideal for testing tracking performance in dense environments.

5.4.1 Experimental Setup

In order to have the reference points for comparison, the first experiment of benchmarking was reproduced for the baseline model introduced by Broström 2022. This setup integrates the YOLOv7 object detector with StrongSORT, using the default OSNet ReID backbone (`osnet_x0_25_msmt17.pt`), and serves as a reference point for comparison. All parameters for detection confidence (0.25) and StrongSORT

tracking hyperparameters (`max_age=30`, `max_dist=0.2`, `n_init=3`) were kept consistent with the original repository configuration [Broström 2022].

- **Baseline Evaluation (YOLOv7 + StrongSORT):** The benchmark was executed on both MOT17 and MOT20 datasets using YOLOv7 for detection and OSNet (x0.25) for re-identification. These results act as a baseline to evaluate the improvements proposed by integrating YOLO-World and a stronger ReID backbone.
- **Proposed Evaluation (YOLO-World + StrongSORT):** To evaluate the contribution of open-vocabulary detection and stronger appearance modeling, the pipeline was re-executed using YOLO-World (versions `l` and `m`) and OSNet-AIN. Lower confidence thresholds (e.g., 0.005) were adopted to facilitate broader semantic coverage and recall potential of YOLO-World.

Once evaluated the improvement by replacing the Object Detector, the other benchmark experiments were carried out in iterative rounds of parameters fine tuning to refine the configuration (see Table 5.1) and evaluate the impact of each adaptation. Four different configurations were tested:

Table 5.1: Configurations for StrongSORT+YOLO-World/L

ID	Conf.	ReID Weights	max_age	n_init	max_dist
EXP-Base	0.005	osnet_x1_0	30	3	0.2
DONE-EXP-Base-RobustReID	0.005	osnet_ain_x1_0	30	3	0.2
DONE-EXP-StableID	0.005	osnet_ain_x1_0	30	4	0.3
EXP-AgeLarger	0.005	osnet_ain_x1_0	60	2	0.4

Each configuration change aimed to mitigate specific performance limitations:

- Reduction of `max_age` (70 to 30): reduced temporal tolerance to prevent outdated identity persistence in dynamic crowded scenes (improves IDF1 and reduces IDSW). A shorter `max_age` can help the tracker avoid incorrect identities during occlusions, which improves precision and reduces false positives.
- Increase in `max_dist` (0.2 to 0.3): allowed more flexibility in appearance matching, compensating for view variation and occlusion. This tolerance in-

crease can help match detections across slightly displaced positions caused by jitter or minor occlusions.

- Switch to `osnet_ain_x1_0`: offered superior ReID feature discrimination, especially relevant in dense scenes like MOT20 or sport scenarios.
- Higher `n_init` (3 to 4): increased the robustness of confirmed tracks by requiring an object to be seen in four consecutive frames before being confirmed as a track reduced ID switches and false trajectories..

5.4.2 Models and Architectures

The framework is composed of a detection model, a tracking model, and a visual representation (embedding) model. The conceptual description of each architecture was presented in Chapter 2. In the following is presented an overview of the configuration used in the experiments described in this chapter.

- **Object detection using YOLO-World:** (open-vocabulary, zero-shot), variants `l` and `m`, used with text prompts for the classes of interest (e.g., "`person playing`", "`sports ball`", `texttt`"audience").
- **MOT using StrongSORT**, a tracking-by-detection paradigm, used to assign and maintain object identities and to produce the tracklets.
- **Re-ID representation using OSNet-AIN:** `osnet_ain_x1_0`), producing appearance embeddings of 512 dimensions, used for same-view and near-view pairs.
- **Holistic representation usign DINOv2:** (ViT-B, self-supervised), producing holistic embeddings of 768 dimensions, used for cross-view pairs.
- **Offset estimation** based on RANSAC-style inlier voting (number of hypotheses $K = 5$], frame tolerance $\Delta = 10$ frames), with a fallback to the best-match.
- **Sequence alignment** using a DTW-Norm variant, based on FastDTW with a one-to-one Hungarian assignment between tracklets.

5.4.3 Computational Resources

The local environment used is the Avell A65 AION laptop, provided by the SWPERFI project. This machine is equipped with a 12th generation Intel Core i9-12900HX processor (2.30 GHz), 64 GB of system memory (63.7 GB usable), and an NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB of dedicated memory (NVIDIA driver version 32.0.15, CUDA 11.8), in addition to the integrated Intel UHD Graphics. The operating system is a 64-bit (x64) system. When the Google Colab environment was used for the initial configuration of the tracking notebooks, the GPU made available by the platform at execution time was NVIDIA T4.

5.5 Experiments on Tracklet-Based Synchronization

Following the setup and integration of MOT algorithms, this section focuses on evaluating the proposed framework using the generated tracklets. The objective is to validate whether tracklet-level visual features can be effectively used to estimate the offset between user-generated videos. The experiments were organized in an incremental way, starting from controlled cases (similarity metric and offset estimation on simple inputs) and advancing to the multi-view case, the quantitative evaluation against a ground-truth set, an error analysis of the difficult conditions, and a generalization experiment.

As described in the previous section, the model for extracting visual features, *i.e.*, extracting the trajectories of the objects tracked in each video, has already been adapted. So, we only need to validate the proposed tracklet-based synchronization method. All the experiments in this section describe the experimental design and conditions, the quantitative results and their discussion are presented in Chapter 6.

I. Similarity Metric Definition

For this first experiment, the goal was to evaluate and select the metric that best captures the similarity between feature vectors extracted from objects tracked in the videos. The experiment involved testing three different similarity metrics, Euclidean distance (Eq. 2.1), Cosine Similarity (Eq. 2.2) and MSE (Eq. 2.3), to determine which provided the best performance in identifying corresponding trajectories across

different videos. These metrics were selected based on previous related works that highlights their effectiveness in calculating feature similarity in multi-perspective video scenarios.

Using tracklets extracted by StrongSORT, the features of the tracked objects in each frame were compared using each metric. A similarity matrix was generated for each metric and visualized as a heatmap, enabling a detailed analysis of the differences and correspondences between the tracklets of the videos. To facilitate the initial process of evaluation of the implemented script, the experiments were carried out using only two UGVs as input.

II. Offset Estimation

Once the metric used to calculate similarity had been defined, the next step was to validate the process of estimating the offset between videos. To do this, an experiment was planned with the aim of aligning two videos automatically, using as a basis the features extracted from the objects tracked in each frame of the videos. Similar to the previous experiment, two UGVs were used as input in order to reduce the complexity of the initial implementation, which allowed a controlled and detailed qualitative analysis of the proposed approach.

The experiment begins with the process of extracting tracklets from the input videos. After extraction, the similarity between these trajectories is calculated, using the Euclidean distance as the main metric to compare the features extracted from each tracked object. From this comparison, a similarity matrix was generated using the score of the tracklets distances, which was visualized through a heatmap, highlighting the correspondences between the object trajectories in the two videos. The result of this matrix was used to identify the most similar tracklets in order to obtain the list of frames that contain the objects of the similar trajectories.

In the current version of the framework, instead of relying only on the single most similar frame pair, the final offset is selected by a RANSAC-style inlier voting over the most similar tracklet pairs, as defined in Section 3.1.2, which makes the estimation more robust to wrong matches. The offset value is then used to detect the overlap point, i.e. identifying how long the frames in both videos remained similar before diverging.

This offset estimation was first validated in two controlled cases, in order to reduce the complexity of the initial analysis:

- **Same video processed twice** (Figure 5.7): the same video was processed as the two inputs, so the tracklets and frame features are expected to match exactly. This setup validates the correct functioning of the similarity metrics and the overlap detection, and represents the most basic case.

```
# 1) detect and track objects: get visual features (tracklet set per video):
SOURCE_VIDEO1_PATH = r'C:\Users\elton\Desktop\strongsort_042024\Yolov7_StrongSORT_OSNet\video_C2_clipchamp.mp4'
SOURCE_VIDEO2_PATH = r'C:\Users\elton\Desktop\strongsort_042024\Yolov7_StrongSORT_OSNet\video_C2_clipchamp.mp4'
#Tracking process: StrongSORT
tracklet_video_01_file = get_tracklets_strongSORT_YOLO_WORLD(source=SOURCE_VIDEO1_PATH, detector_model=YOLOWorld_model, save_vid=True, save_txt=True)
tracklet_video_02_file = get_tracklets_strongSORT_YOLO_WORLD(source=SOURCE_VIDEO2_PATH, detector_model=YOLOWorld_model, save_vid=True, save_txt=True)
```

Figure 5.7: Experiment reproduced for the same video: SOURCE_VIDEO01_PATH and SOURCE_VIDEO02_PATH received the same video file.

- **Second video as a cut of the first** (Figure 5.8): the second video is a trimmed version of the first one. This case evaluates the robustness of the overlap detection in a more realistic situation, where Video 2 represents only part of the content of Video 1, requiring the framework to identify the exact moment when the common content begins and ends (overlap point).

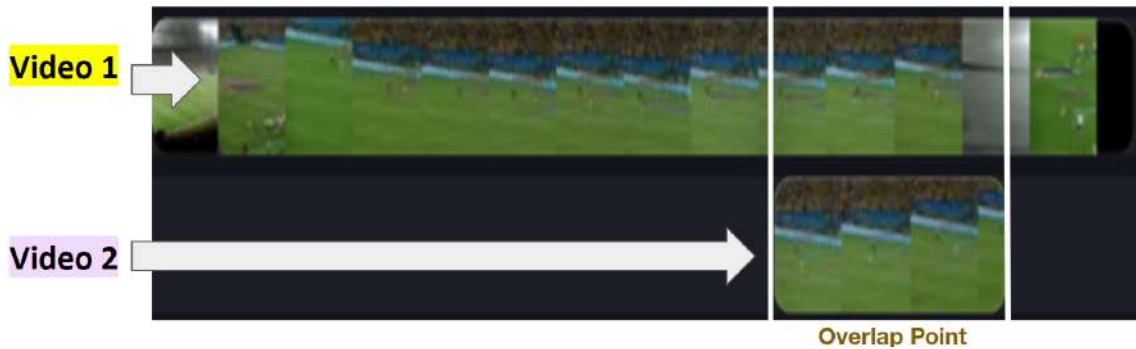


Figure 5.8: Video 2 is a cut of the Video 1. The highlighted part is where both videos present the same moment.

5.5.1 Ground-Truth Evaluation Set for Quantitative Analysis

To support the quantitative evaluation of the proposed framework, a ground-truth evaluation dataset was constructed by combining manually annotated MUVY-Sync

pairs with additional synchronized fragments collected from the MUVS dataset. The goal of this setup was to obtain a unified set of video pairs with known frame-level temporal offsets, allowing the predicted offset of each method to be directly compared against a reference value.

The construction process followed two complementary strategies. First, the selected and manually annotated pairs from MUVY dataset were preserved as a controlled benchmark with explicit viewpoint variation, including same-view (SV), different-view near (DV-near), and different-view far (DV-Far) cases. Second, additional pairs were collected from MUVS dataset by selecting fragments from different cameras that recorded the same event and period (overlap points). For these MUVS-derived pairs, the timestamp-based alignment provided the nominal temporal correspondence, while the residual per-camera offsets available in the annotation files were used to compute the ground-truth offset.

To ensure that the reference offset was valid throughout each generated clip, only windows fully contained inside a stable temporal interval were accepted. Fragments with unreliable offset values, insufficient overlap, or transitions between different offset regions were discarded. All accepted clips were re-encoded to a constant frame rate, making the frame-based offset consistent across both videos of each pair.

The resulting pairs were written in a common manifest format containing the video paths, reference frames, ground-truth offset in frames and seconds, FPS, notes, view class, data source, and sport category. The custom **UG-Sync Studio** tool was then used as a visual inspection and validation interface. It allows the annotator to browse the generated pairs, compare the two videos side by side, inspect the tracking outputs and evaluation results, and update the reference `offset` and `offset_sec` values when manual correction was needed.

5.5.2 Multi-View Synchronization Experiment: Visual Representation

Previous experiments in this work focused on synchronizing videos with similar viewpoints, such as trimmed versions of the same source (Figure 5.8) or recordings from the same physical location. Now, the robustness of the proposed framework were validated in more challenging scenarios. In this subsection, the set of experiments

conducted using videos captured from different viewpoints of the same sport event s presented. These conditions introduce challenges such as partial occlusions, perspective distortions, and differences in object visibility and scale, which are common in user-generated videos.

The main experiment of this phase investigates the visual representation used for cross-view synchronization. The same synchronization pipeline is kept fixed and only the embedding model is changed, in a controlled A/B comparison: the Re-ID embeddings extracted by OSNet-AIN are compared against the holistic embeddings extracted by DINOv2, over the annotated pairs and stratified by the recording condition (SV, DV-near, DV-far). In addition, two variants of the pipeline are compared:

- **Standard:** nearest-neighbour matching based on the minimum Euclidean distance between feature vectors.
- **DTW-Normalized (DTW-Norm):** a spatially augmented Dynamic Time Warping that compares the trajectories as temporal sequences, with a one-to-one Hungarian assignment between tracklets to avoid the hub-tracklet problem (one tracklet matching several others).

Pipeline overview. To carry out this experiment, the same pipeline was applied to each pair:

1. **Object Detection and Tracking:** videos are processed using an adapted version of YOLO-World for zero-shot object detection (with class prompts such as “person playing”, “audience”) followed by an adapted version of StrongSORT for multi-object tracking [Alencar and de Freitas 2025]. The resulting tracklets are stored in the MOT format.
2. **Tracklet Feature Embedding:** each tracklet is transformed into a temporal sequence of feature vectors. For the cross-view condition, the holistic DINOv2 embeddings are extracted from the frames of the closest trajectories.
3. **Tracklet Similarity Calculation:** a pairwise similarity is computed between tracklets of both videos, using the Euclidean distance (Standard) or DTW-

Norm, which accounts for temporal misalignment and differences in motion speed. The most representative pairs are visualized via heatmaps.

4. **Offset Estimation and Overlap Detection:** for the selected pairs, the temporal alignment is estimated by the RANSAC-style inlier voting (offset) and the overlap duration, applying the temporal consistency constraint described in Section 3.1.2.
5. **Quantitative Evaluation:** the final synchronization offset is compared with the ground-truth annotations to measure the synchronization error (MAE and $\text{Acc}@T$).

The quantitative evaluation is performed using a dedicated evaluation module (`batch_sync_eval.py`), which compares the estimated offsets from different synchronization strategies, that is, single best tracklet, overlap-based match, frame-level similarity, and global multi-tracklet (RANSAC) voting, all against the ground truth. To survive interruptions, the module saves the result of each pair immediately (per-pair JSON output) and caches the tracking and DINOv2 features.

5.5.3 Error Analysis Experiment

An additional experiment was designed not to improve the synchronization score, but to understand and characterize the conditions in which the framework has more difficulty. The motivation is that the average error alone does not explain *why* the framework fails in some pairs, which is important for the scientific contribution and for guiding future work.

This experiment has two complementary parts:

- **Stratified error inspection:** the synchronization error is analyzed separately by recording condition (SV, DV-near, DV-far) and by sport, in order to identify which conditions concentrate the largest errors.
- **Embedding separability analysis:** for selected pairs, the tracklet embeddings of both videos are projected into a low-dimensional space using UMAP (with cosine metric) and t-SNE, to visually inspect whether the embeddings of corresponding objects from the two views fall close together or form disjoint

clouds. This analysis is used to support the hypothesis that the failure in the DV-far condition is a *representational* failure (the features do not generalize across large viewpoint changes) rather than an *algorithmic* failure of the synchronization method.

The main goal of this experiment is to connect the numerical error with a visual and interpretable explanation of the behavior of each representation under different viewpoint conditions.

5.5.4 Generalization Experiment on CMU Panoptic

To evaluate whether the conclusions obtained on the MUVY set generalize to a controlled multi-view environment, an additional experiment was conducted on the CMU Panoptic Studio dataset (Pose subset). This dataset provides a large number of calibrated views of the same scene, which allows the construction of many cross-view pairs with a known temporal reference. From this dataset, a set of 779 Video pairs were used.

The same synchronization pipeline and the same evaluation protocol (MAE and $\text{Acc}@ \tau$) used for the MUVY set were applied here, with the pairs organized according to the proximity of the camera viewpoints. The purpose of this experiment is to verify, in a controlled and calibrated setting, whether the behavior observed in the user-generated multi-view set (in particular, the difficulty under large viewpoint changes) is also present, and to provide a second, independent point of evaluation for the proposed framework. All of these experiments were carried out to understand the limitations of the current implementation and to identify the main challenges involved in synchronizing UGVs from different perspectives. The analysis of the results is presented in the following chapter.

Chapter 6

Results

This chapter is divided into two parts. The first presents the development outcomes that supported the implementation of the proposed framework, including dataset creation and integration of multi-object tracking models. The second part presents the results and analysis of the experiments designed to evaluate the effectiveness of the synchronization method based on visual features extracted from MOT algorithms.

6.1 Development Outcomes

This section presents the results obtained from the implementation and preparatory steps that were necessary to enable the experimental evaluation. These outcomes include dataset creation, MOT model integration, and initial validations that guided the design of the final synchronization method.

6.1.1 Created Datasets and Its Structure

This section will describe the results of the procedures used to create a dataset of videos with main characteristics of user-generated content and multi-view. In total, three new datasets were created. The first, UVY, consists of sports videos recorded by users who were present at the event. The other two, MUVY and MUVS, are also composed of videos recorded by users, but only if these videos are multi-view, *i.e.* necessarily part of a set of at least two UGVs recorded at the same sporting event. The following subsections provide more details on the numbers, structures and main

characteristics of each dataset created.

6.1.1.a. UVY: a Dataset of Sport User-Generated Videos from YouTube

As a first result of what was described in Section 5.1, UVY, a Sport User-generated Videos from YouTube dataset was created to evaluate Multi-Object Tracking techniques applied to sport videos extracted from YouTube. As illustrated in Figure 6.1, the constructed dataset comprises 20 videos, covering 7 different sports categories (soccer, basketball, volleyball, tennis, athletics, artgym, and cricket). In total, the dataset contains 31,961 annotated frames, 128,258 bounding boxes, and 378 annotated tracks, with an average of 4.01 objects per frame. The mean video duration is 53.3 seconds, with an average frame rate close to 30 fps, which is consistent with typical consumer-grade recording devices.

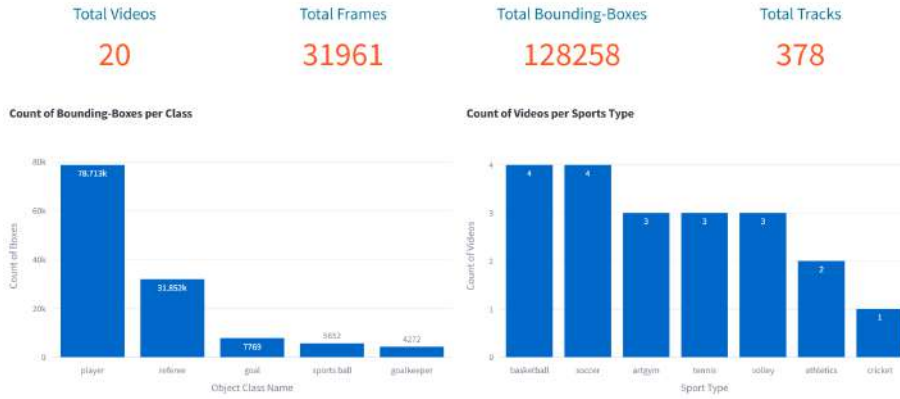


Figure 6.1: Summary of the total videos and its distribution for the UVY.

Regarding object classes, the distribution is dominated by players (78.7K instances, 61%), followed by referees (31.9K instances, 25%). Minority classes include goals (7.8k), sports balls (5.7k) and goalkeepers (4.2k), accounting for approximately 14% of the total annotations. From a sports perspective, the dataset is relatively balanced among team sports, with soccer and basketball contributing 4 videos each, volleyball and tennis contributing 3 videos each, while athletics (2), artgym (3), and cricket (1) are less represented. This distribution highlights the strengths and the limitations of the dataset, where the large number of instances for players and referees provides a substantial number of annotated examples for model development and evaluation. However, the long-tail distribution of minority classes (e.g.,

sports ball and goalkeeper) may hinder performance for those categories, requiring balancing strategies such as oversampling or data augmentation.

From a multi-object tracking (MOT) perspective, the dataset provides 378 annotated tracks with varying lengths, covering both long continuous tracks (e.g., athletics and artgym) and short, fragmented tracks (e.g., basketball and soccer). This diversity makes it particularly useful for evaluating object re-identification and occlusion handling in MOT algorithms. Despite the relatively short average duration of videos, the dataset captures different crowd densities and motion patterns, offering a valuable benchmark for fine-tuning and validating tracking frameworks in unconstrained sports environments.

Table 6.1 shows that the created dataset is comparable in numbers to the other datasets listed. Compared to the other works listed in the mentioned table, UVY has a small number of videos with varying resolutions, indicating challenges with UGVs. In addition, of the two works with sports videos, UVY is the only one with a variation of more than one type of sport. All the datasets listed in Table 6.1 are mainly used to evaluate MOT techniques (see Table 3.3).

Table 6.1: Comparison among public datasets used to evaluate MOT algorithms (including UVY)

MOT Dataset	Application	# Videos	Video Duration	Resolutions	Recording Nature
MOT17 (2017)	Pedestrian tracking (train station, shopping mall, crowded street)	42	MAX: 01min25s, MIN: 15s.	1920x1080 (# 36), 640x480 (# 6).	Pedestrian tracking (train station, shopping mall, crowded street)
MOT20 (2020)	Pedestrian tracking (train station, shopping mall, crowded street)	8	MAX: 02min13s, MIN: 17s.	1920x1080, 1920x734, 1545x1080, 1173x880, 1654x1080.	Crowded environment
MOTS20 (2020)	Pedestrian tracking (train station, shopping mall, crowded street)	8	MAX: 01min25s, MIN: 15s.	1920x1080 (# 6), 640x480 (# 2).	Crowded environment
SoccerNet-Track (2022)	MOT in sports environments	100	30s	1920x1080	Official broadcast of soccer.
UVY Track (Ours)	MOT in sports environments	20	MAX: 02min15s, MIN: 08s.	640x360, 1280x720, 1920x1080.	Sport UGVs.

Although UVY Track currently has a relatively small number of entries (Table 6.1), its data source, YouTube, is a public platform that is fed daily; this number

could increase by as much as the number of Creative Commons’ user-generated videos available on the platform, with the only limitation being that the videos are sports videos. The video files and their related annotation files are structured according to the MOT standard format¹.

This version also improves annotation quality, as all bounding boxes were manually reviewed and corrected in CVAT (Computer Vision Annotation Tool). After the initial detection phase, all annotated bounding boxes were visually inspected and corrected as needed, addressing common issues such as inaccurate bounding box dimensions, identity switches, false positives, and missed detections in challenging scenes. Figure 6.2 illustrates this process. On the left (1), the initial detection made by YOLO-World is shown with a suboptimal bounding box size. The image (2) demonstrates the corrected annotation after manual refinement using CVAT, which more accurately captures the extent of the athlete’s body, allowing for a more robust comparison of features.

Finally, the third image (3) in Figure 6.2, shows an example of an ID-switch issue, where the identity assigned to the athlete is mistakenly transferred to a different person in a subsequent frame. This manual validation process ensures higher precision and consistency across the dataset, making it more robust for training and evaluation of tracking algorithms and multi-view video synchronization.

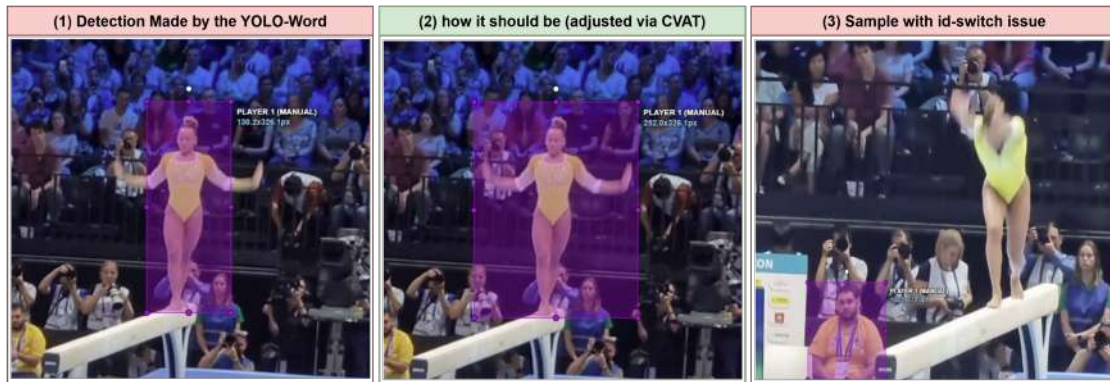


Figure 6.2: Example of manual annotation refinement using CVAT. (1) Initial detection by YOLO-World; (2) Corrected bounding box via CVAT; (3) Frame with ID-switch issue where the identity was incorrectly transferred to a different person.

¹<https://docs.cvat.ai/docs/manual/advanced/formats/format-mot/>

It is worth mentioning that the results obtained from the creation of this first dataset went beyond the files present in the dataset that will be made public. All scripts created for its construction, as well as the experience gained during its execution, are considered successful results obtained from this first dataset. This is because UYV served as the baseline for creating a second dataset, MUYV, which consists of Multi-View UGVs selected from YouTube and created collaboratively by members of the Vision-AI Group, a research sub-group of the SWPERFI Project.

6.1.1.b. Multi-View UGV Custom Datasets

Considering again what was described in Section 5.1 about workflows 2 and 3, two new datasets (MUYV and MUVS) were created collaboratively. These are similar in terms of having at least two user-recorded videos captured at the same event, thus providing the multi-view feature. However, MUYV (**M**ulti-view **U**ser-generated **V**ideos from **Y**ouTube) is composed of UGVs that were selected from YouTube under the condition of multi-view videos. The MUVS (**M**ulti-view **U**ser-Generated **V**ideos & **S**ensor Data), on the other hand, consist of videos recorded by the authors of the dataset (including the author of this work) and also contain metadata collected from the sensors of the smartphones used to record the videos (including the timestamp). In the following, more details are given on the current numbers of each of these datasets, which are still in the expansion phase.

I - MUYV Dataset

MUYV dataset² was created to be used as a source for multi-perspective analysis and applications, including video synchronization. Its latest version includes 141 videos recorded in 31 events, totaling 3h15min of content, all published on YouTube by users. In addition, the dataset comprises videos recorded across 7 types of sports, adding diversity to the selected content and bringing the application and complexity closer to a real-world scenario. This information is summarized in Figure 6.3.

²<https://swperfi-project.github.io/Pages-dev/MuvyDataset-page/>



Figure 6.3: MUVY Stats: summary of the total videos and its distribution.

Beyond its multi-perspective content that can be used to validate video synchronization proposals (which is the case with this research), another result we had with the creation of MUVY was the acceptance of the paper, titled "Exploring multi-camera views from user-generated sports videos", which was accepted for presentation at the 2024 KDMiLe - the Symposium on Knowledge Discovery, Mining, and Learning, thus reinforcing the contribution of this result to the research community [Pessoa et al. 2024]. As an illustration, Figure 6.4 has been added with an example of how the videos and related files are structured.

II - MUVS: Motorola's Custom Dataset

MUVS³, the second Multi-View User-Generated Video dataset was created. As mentioned before, this specific dataset is different from the others because it is composed of videos and also corresponding files that contain a set of sensor values that were collected during the recording processes. During data collection experiments, the research group went to a total of 21 sport events, 340 user-generated videos were recorded during those events. These videos covered events of 5 sport genres (basketball, soccer, hockey, volleyball, and tennis), see Figure 6.5.

³<https://swperfi-project.github.io/Pages-dev/MuvsDataset-page/>

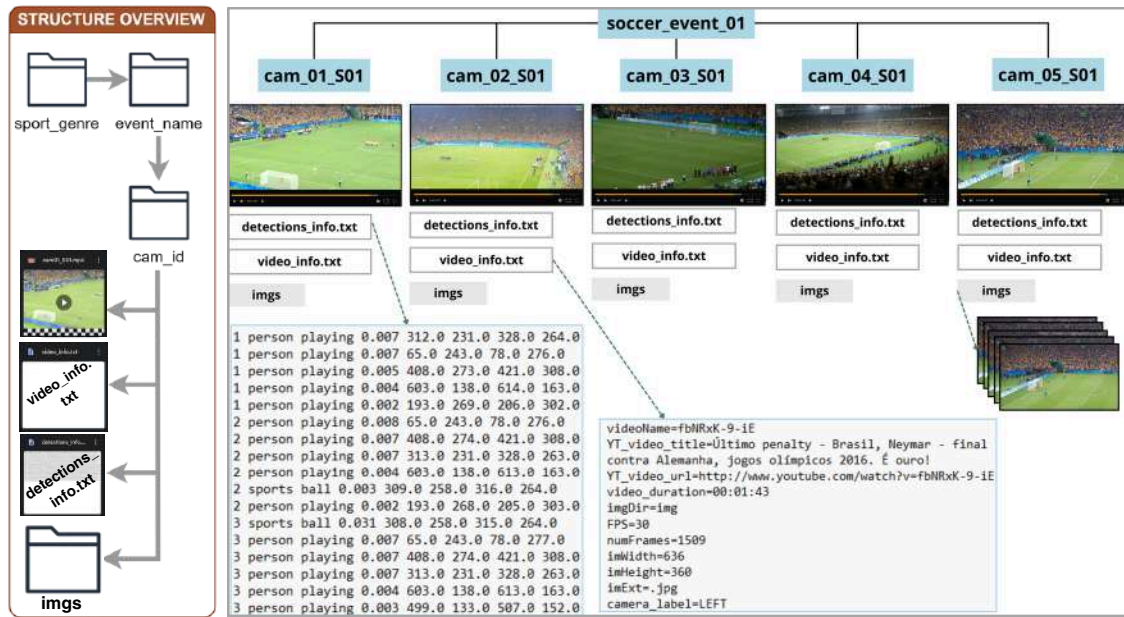


Figure 6.4: MUVY Structure Overview: each video is accompanied by a video_info file, an extracted frames directory, and an object annotations file.

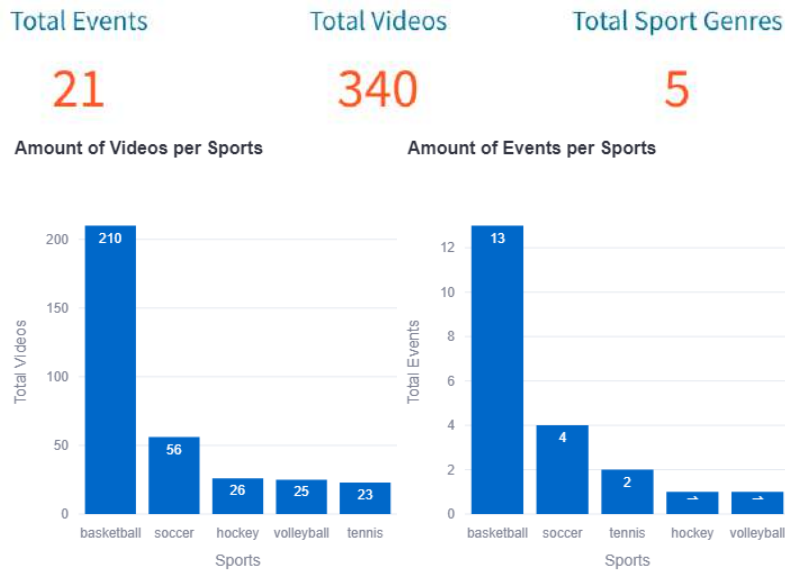


Figure 6.5: Summary of the total videos and its distribution for the MUVS Dataset.

The raw data are structured similarly to the MUVY dataset, with the difference that this dataset includes a directory containing a metadata file (containing the sensor values) for each video recorded by a specific device. Also note that, as illustrated in Figure 6.6, for each smartphone used, there will be cases where more

than one video was recorded at the same event, thus illustrating another difference from this dataset to MUVY.

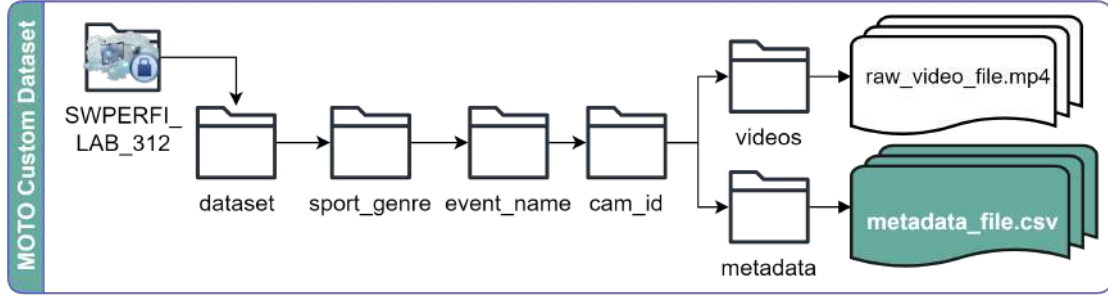


Figure 6.6: Overview of the Data Structure in MUVS, Motorola's Custom Dataset.

6.1.2 MOT Technique: StrongSORT & YOLO-World

From the process of selecting and implementing multiple object tracking techniques, three models were implemented (DeepSORT-2017, TrackFormer-2022 and StrongSORT-2023) and applied to videos from the dataset described in the previous section. Initially, the analysis of the preliminary results was entirely qualitative, but in the future, we intend to evaluate the results by a quantitative analysis of the models when applied to the UYV dataset.

However, during the processing of the videos selected from the new dataset, and from a **comparative** and **qualitative** analysis of the results observed in the output videos returned from each model, it was possible to observe that factors such as the quality of the video, the speed of the objects and the complexity of the scene can influence the performance of the algorithms, which was already expected, since the user-generated videos have varied recording conditions.

Moreover, it was observed that only DeepSORT⁴ and StrongSORT⁵ were able to detect the ball, but this was due to the fact that YOLOv5 and YOLOv7 are pre-trained models specialized to detect objects such as "sports ball" (Figure 6.7).

⁴<https://drive.google.com/file/d/109tnrdW1uVRco2rHu2XwqxXkyjxLO8y1/view>

⁵<https://drive.google.com/file/d/1oWrPohUy1PkkB4Zkb7iT20y5XljG1mjb/view>



Figure 6.7: MOT Implemented - Output Videos: (a) DeepSORT (YOLO-v5), “sports ball” class detected; (b) TrackFormer result, no ball detected; (c) StrongSORT result, detected the “sports ball” class and improved tracking of multiple objects, including a person positioned near the goal area.

While TrackFormer⁶ was only able to detect pedestrians/people, due to the CNN-based model used, which was not specialized to detect objects but rather to extract features from regions similar to a person [Meinhardt et al. 2022]. This highlighted the possibility of adapting tracking-by-detection models (e.g. DeepSORT and StrongSORT) by replacing the detector model by a model capable of detecting objects of interest according to the proposed context.

Now, comparing the results between DeepSORT (YOLOv5) and StrongSORT (YOLOv7), it was confirmed that the former suffers from the id-switch problem during the data association process, i.e. for the same detected object, such as the ball, DeepSORT assigns more than one id (tracklet-ID), as illustrated in Figure 6.8 extracted from its output video². However, this had already been mentioned by the authors of StrongSORT (an improved version of DeepSORT) [Du et al. 2023]. For this reason, we decided to continue using StrongSORT as the technique to extract the trajectories that will be used to synchronize the UGVs.

I. StrongSORT with YOLO-World (2024)

The implemented StrongSORT (YOLOv7) presented promising results that indicated its effectiveness in tracking objects in UGV sports videos⁷. However, thinking in the context of calculating similarity between the trajectories of objects detected in different videos, we observed the necessity of a model that only tracks objects that are common between the different videos, which means that for sports videos,

⁶<https://drive.google.com/file/d/1d1WKbT2T24fZq71kevvAz5zSDfsG7jjz/view>

⁷<https://drive.google.com/file/d/1gLwhI758qYAUFtIsMAsaqh7ju7YScEO4/view>

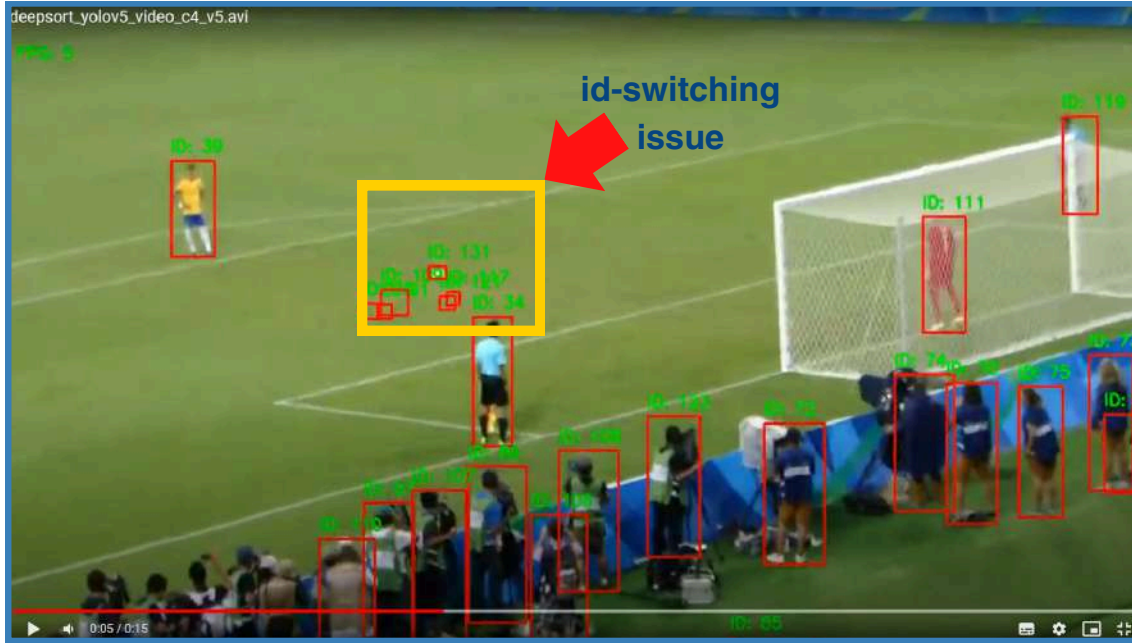


Figure 6.8: The video processed by DeepSORT presented id switching problems during the process of data association.

the ideal is to carry out the similarity calculation process only for objects that are on the field (i.e. “person playing”, “sports ball”, “referee”, “goalkeeper”), especially because the chance of these objects of interest appearing in videos recorded from other angles is greater.

For this reason, a new adapted version of StrongSORT was implemented, where the detector model has been replaced. Previously, YOLO-v7 was used, now there is an adapted version where the objects tracked are those detected by YOLO-World, a zero-shot detector (Section 2.3.1). Figure 6.9 shows the results side by side, with the left side showing the result of a video⁵ processed by the version available in the literature (StrongSORT+YOLOv7), and the right side showing the same video⁸ processed by the adapted version developed in this research (StrongSORT+YOLO-World). Without requiring model training, this version effectively reduces the number of tracked objects, focusing mainly on the objects (*e.g.*, “person playing” and “sports ball”) in the field.

Despite this satisfactory result for the context of this work, it is important to mention that this new version will still have cases in which it detects an object

⁸<https://drive.google.com/file/d/1fAOF3ZhbG0TTSNg17UNWjjRhhFNpfYh2/view>

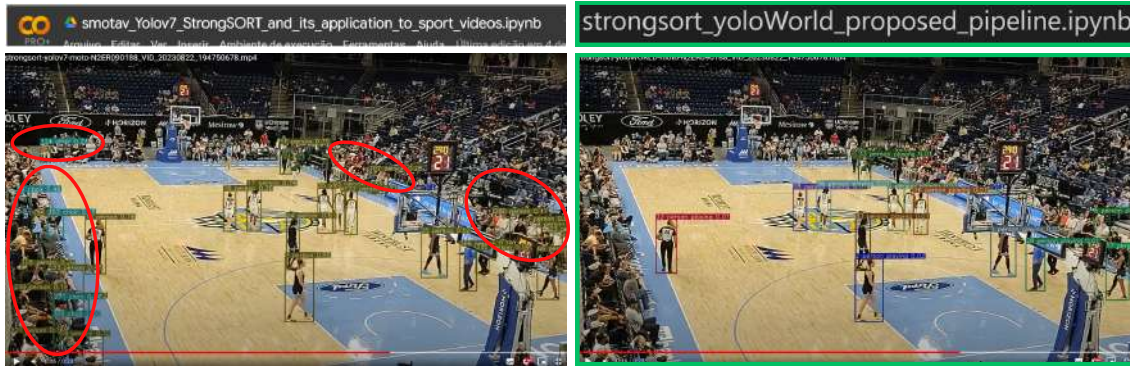


Figure 6.9: StrongSORT results: on the left side YOLOv7 detecting several people, including those in the audience; on the right, YOLO-World detecting only people playing on the court.

that is outside the field of action, such as the case highlighted (yellow) in the image on the right-hand side of Figure 6.10, extracted from its output video ⁹. However, when compared in terms of numbers for this specific case, the number of objects detected by the StrongSORT+YOLO-World is around four times less than the number detected by the StrongSORT+YOLOv7 (literature version).



Figure 6.10: StrongSORT results: on the left YOLOv7 detected an average of 20 objects per frame. On the right, YOLO-World detected about 5 objects per frame, focusing mainly on those in the field.

Therefore, based on the qualitative analysis presented before, it was decided to go

⁹https://drive.google.com/file/d/1i4lsk346wC6_K_mrNaO3vZwTgBp8GXhS/view

with StrongSORT+YOLO-World as the deep learning-based multi-object tracking technique that will be used to extract the visual features required to calculate the similarity for the frame-level alignment. For the next phase of the research, quantitative metrics will be used to evaluate the performance of the proposed framework in order to have a qualitative comparison.

6.1.3 MOT-UGV: Quantitative Results

The benchmark results are divided into two parts. First, a comparative analysis between the proposed tracking algorithm (MOT-UGV [Alencar and de Freitas 2025]) and the baseline model from Broström 2022, which combines YOLOv7 with StrongSORT, is presented. This comparison is performed on both MOT17 and MOT20 datasets to validate how the updated detector and ReID backbone affect overall performance.

The second part then focuses on the results of the experiments aimed at analyzing the effect of tracking parameter tuning. These experiments are restricted to the MOT20 dataset, as it better represents crowded scenes with frequent occlusion and high pedestrian density, challenges that are also present in sports video scenarios. The goal is to analyze how changes to StrongSORT parameters improve robustness in dense environments when combined with YOLO-World.

6.1.3.a. Comparison with Baseline Model

According to the metrics summarized in Table 6.2, the proposed model achieves similar HOTA and IDF1 scores compared to the baseline on MOT17, but increases the number of false positives (FP) and identity switches (IDSW). This is likely due to the lower detection threshold (0.005) and the open-vocabulary nature of YOLO-World, which tends to produce many more detections, some of which may be spurious or overlapped. Despite this, a key improvement was observed in false negatives (FN), which dropped from 67.6K to 49k, indicating better recall. This suggests that YOLO-World is more capable of detecting partially occluded or smaller pedestrians, but at the cost of introducing noise.

Table 6.3 presents the results for the benchmark comparison between the proposed and baseline models on MOT20, which has sequences of dense crowd scenar-

Table 6.2: MOT Benchmark: proposed pipeline compared to the baseline model on MOT17

Model	HOTA	IDF1	MOTA	IDSW ↓	FP ↓	FN ↓
YOLOv7 + OSNet (x0.25)	35.04	38.72	34.21	722	5,161	67,583
YOLOv7 + OSNet-AIN	40.40	46.99	39.88	361	4,573	67,623
YOLO-World + OSNet-AIN (Ours)	34.90	38.08	32.89	1,254	75,702	49,084

ios. In MOT20, the proposed pipeline achieved a significant performance increase. HOTA increased from 9.09 to 33.25, and IDF1 increased from 6.23 to 43.79. The MOTA also improved significantly from 8.58 to 32.46. These are more than three times higher compared to the baseline.

However, the cost of these achievements was a higher increase in ID switches (from 2,656 to 10,576). This again reflects the more permissive detection behavior of YOLO-World under a very low confidence threshold. However, the reduction in false negatives by more than half (from 1.05M to 508k) demonstrates the strength of the new detector in covering densely populated scenes.

Table 6.3: MOT Benchmark: proposed pipeline compared to the baseline model on MOT20

Model	HOTA	IDF1	MOTA	IDSW ↓	FP ↓	FN ↓
YOLOv7 + OSNet (x0.25)	9.09	6.23	8.58	2,656	2,096	1,052,620
YOLOv7 + OSNet-AIN	12.21	11.09	11.71	891	1,421	1,053,454
YOLO-World + OSNet-AIN (Ours)	33.25	43.79	32.46	10,576	115,607	508,860

Although YOLOv7 combined with StrongSORT achieves competitive results in MOT17, its performance drops significantly in MOT20. This can be explained by the fact that MOT20 consists of crowded scenes with occlusion and a high pedestrian density per frame. YOLOv7, although accurate in general contexts, was not specifically optimized for such challenging conditions, causing detection failures or missed instances in cluttered environments. This limitation is evident in the higher false negatives and lower IDF1 scores compared to YOLO-World, which are more adaptable to complex visual scenarios.

These results demonstrate that the integration of YOLO-World/L proposed model combined OSNet-AIN, improves tracking performance in dense and dynamic scenes, especially under occlusions. This is a good achievement for sports UGV

analysis, where multiple players frequently occlude each other and move across the scene. The results in MOT20 suggest that the proposed approach is better suited to the challenges highlighted in this investigation about the content of UGVs sports.

6.1.3.b. Tracking Parameter Tuning on MOT20

Table 6.4 presents a summary of the main metrics obtained for MOT20 for each configuration:

Table 6.4: StrongSORT parameter tuning on MOT20

Model	HOTA	MOTA	IDF1	IDSW	GT_IDs	IDs
SS_YW_MOT20	4.88	1.59	2.05	418	2215	137
SS_YWL_MOT20	19.36	36.93	21.73	22888	2215	1170
SS_OSX1_YWL_MOT20	30.66	42.24	39.01	10985	2215	10596

- The switch to `osnet_ain_x1_0` significantly improved identity association, nearly doubled IDF1 (from 21.7 to 39.0) and reduced the ID switches in half, confirming the importance of robust ReID and association tuning.
- HOTA increased by more than 11 points, indicating improved joint detection-association quality.

These improvements demonstrate the effectiveness of both StrongSORT hyperparameter tuning and the use of enhanced ReID embeddings. These benchmark evaluations demonstrate that careful tuning of StrongSORT parameters and replacing the ReID backbone model can significantly improve tracking performance in crowded scenes. The integration with YOLO-World is especially promising given its prompt-driven detection flexibility.

According to the StrongSORT paper, their best result on MOT20 using `osnet_x1_0` was $MOTA = 61.3$, $IDF1 = 80.9$, with a HOTA of around 56.1 [Du et al. 2023]. Our best configuration achieved $HOTA = 30.66$, which is lower than the SOTA by approximately 15 points. The IDF1 score reached 39.01, a result influenced by the presence of frequent occlusions between sequences. The MOTA value attained was 42.24, which, while showing improvement compared to our baseline, was penalized by the high number of false positives resulting from the intentionally low detection

threshold (0.005). Although the final scores do not yet match the published results of StrongSORT, this experiment demonstrates that our framework generalizes well across datasets such as MOT17 and MOT20. It also validates that both the choice of detection and ReID models, as well as careful tuning of tracking parameters, are critical in achieving competitive performance in crowded tracking benchmarks.

6.2 MOT-UGV-based Synchronization

After defining the datasets and evaluating multiple object tracking techniques, this section presents the results obtained from applying the proposed pipeline for UGV synchronization based on visual features. The main goal of the idea was to extract tracklets (i.e., partial trajectories of objects across frames) and use them as units of comparison between videos, based on their visual similarity. This approach aims to identify potential temporal overlaps between UGVs of the same event.

In summary, based on the analysis of the results obtained from the experiments carried out to validate the proposed framework, especially the similarity calculation and offset estimation process, it was observed that the visual features extracted by MOT-UGV, the proposed tracking algorithm that has as its backbone the YOLO-World as detector and the data association part based on StrongSORT, were sufficient to indicate the most similar frames between two different videos, thus indicating a possible overlap point considering the correspondence of the features of each pairwise frame compared.

The Euclidean distance metric was selected for the matrix similarity process because it effectively captured the similarity between tracklets and frames, and the minimum similarity score provided a reliable estimate of the frame offset. The experiments reproduced for different pairs of videos further validated the robustness and precision of the framework in scenarios considered as the baseline.

More details about these results and their analysis are available in the following subsections.

6.2.1 Chosen Similarity Metric: Euclidean Distance

As described in Section 5.5, the three metrics tested (Euclidean distance, cosine similarity, and mean square error) were chosen because the literature had found works using these metrics to measure the similarity between visual features in video analysis tasks.

The **Euclidean distance** heatmap (Figure 6.11) shows greater variability in the similarity scores between the tracklets in the two videos, suggesting that this metric is sensitive to both the direction and magnitude of the feature vectors. In this experiment, Euclidean distance provided a clear differentiation between tracklets that represent similar tracklet features and those that do not. This was observed through the analysis of the high-similarity zones, which are the blue areas that represent regions where the Euclidean distance value is lower, meaning the tracklets are similar. The red zones represent the low-similarity zones, the areas where the similarity score is higher, indicating dissimilar tracklets.

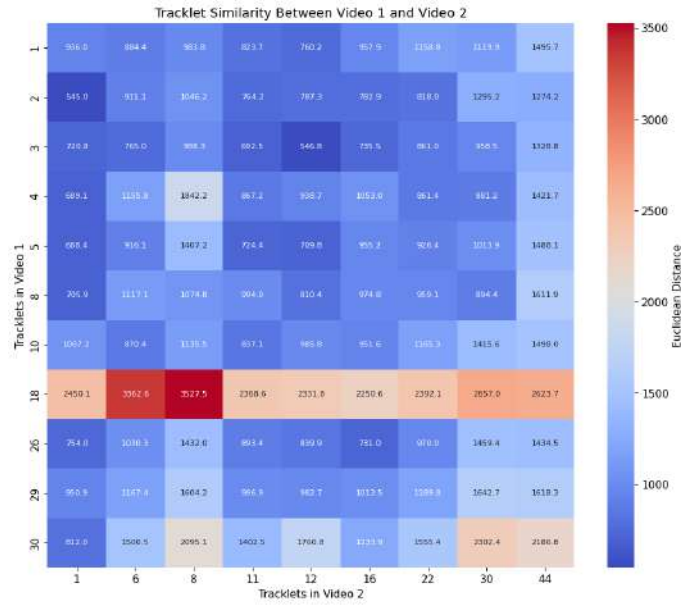


Figure 6.11: Euclidean distances heatmap, illustrating the variation of the pairwise tracklet comparison.

Based on the analysis of Figure 6.11), the Euclidean distance metric revealed that certain tracklets from Video 1 have high similarity to corresponding tracklets from Video 2. For example, with a similarity score = 764.2, tracklet 2 in Video 1

(Y axis) showed a high similarity to tracklet 11 in Video 2 (X axis) and, following the ground-truth annotation, these two tracklets have the features of the exact same object detected in two different videos.

However, the **Cosine similarity** heatmap (Figure 6.12) illustrates values that are consistently close to 1.00. This highlighted the fact that these metrics focus on the angle distance between the vectors, not capturing the variations in magnitude (*e.g.* object size or speed). Consequently, the cosine similarity metric might overlook magnitude variations that could be useful for more precise tracklet correspondence, especially in videos where objects vary significantly in scale or distance, which is the case for the videos available in the datasets created and introduced before.

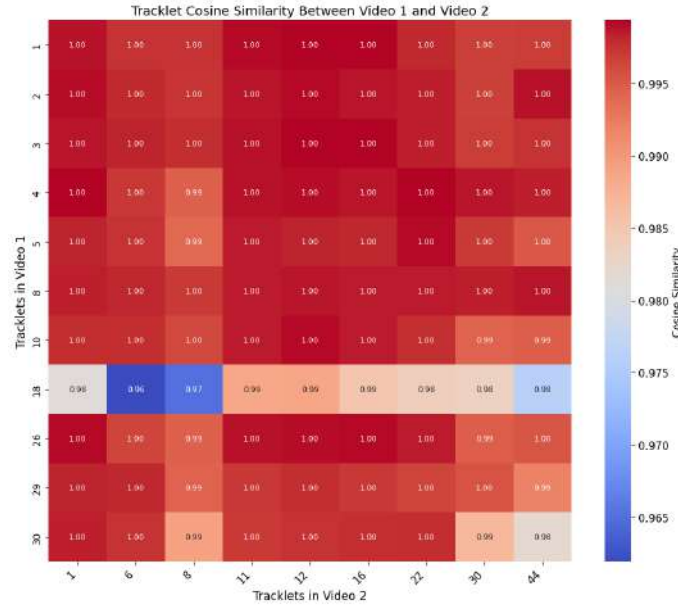


Figure 6.12: Cosine Similarity heatmap, illustrating the variation of the pairwise tracklet comparison. All values very close to 1.00.

And finally, the **heat map of the mean square error (MSE)** (Figure 6.13), although showing some variation similar to that of the first metric, this metric presented a wide range of values for cases of not similar tracklets. This indicates that this metric may be sensitive to outliers, as it penalizes large differences between feature vectors, which can be confirmed by looking at the high values for the most dissimilar cases. This suggests that the MSE may overestimate the differences when there are only small variations between the tracked objects. Therefore, Euclidean distance emerges as the most suitable choice for this experiment, and it will be used

as the primary similarity metric for subsequent stages of the proposed framework.

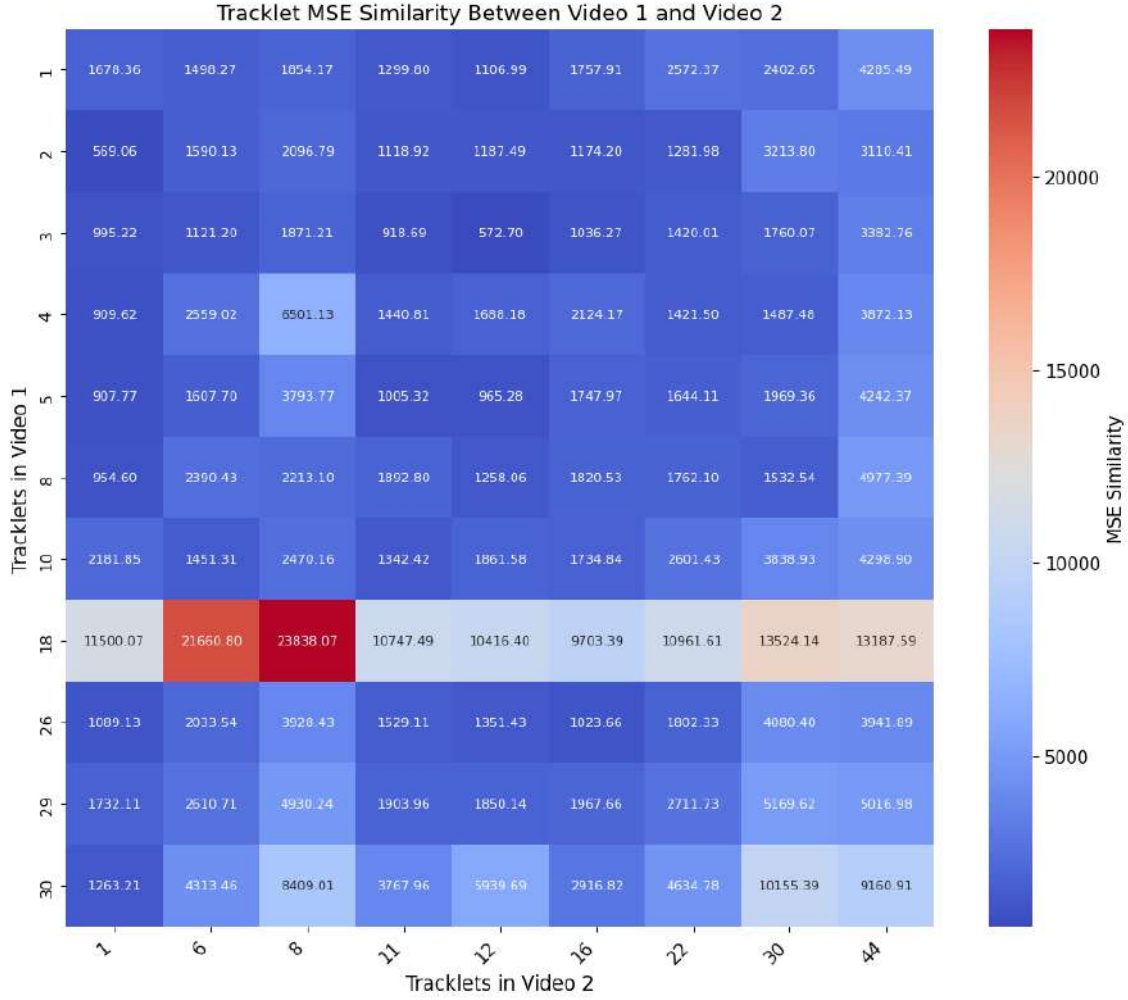


Figure 6.13: Mean Squared Error Values Heatmap, illustrating the variation of the pairwise tracklet comparison. Less robust in distinguishing fine differences.

6.2.2 Synchronization based on visual feature correspondence

The results obtained from the offset estimation experiments (Subsection 5.5) were analyzed to **qualitatively** verify how precise the script developed is in aligning two videos, thus evaluating the match conditions for the similarity calculation between the tracklets extracted from the frames of each video.

6.2.2.a. Similarities Between the Same Video

By analyzing the heatmap returned from the process of calculating the similarity scores between the trajectories of video 1 and video 2, it was possible to see that

the conditions implemented in the script for calculating the similarity matrix work as expected for the easiest case, that is, in which the same video is processed twice, thus resulting in the process of comparing the same content, highlighting the pairs with the lowest similarity scores. This resulted in a matrix with a zero diagonal, indicating the correspondence of the content of the trajectories, as seen in Figure 6.14.

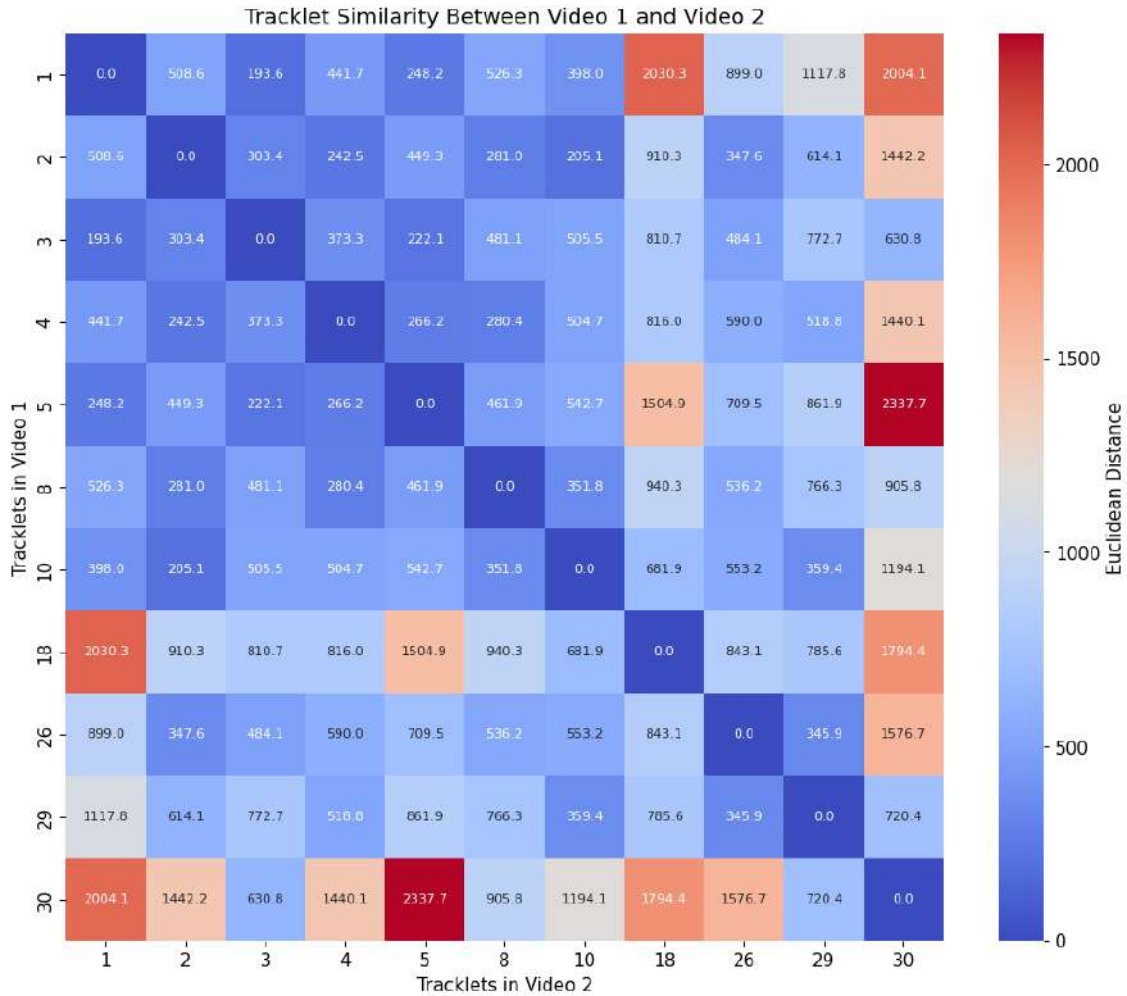


Figure 6.14: Tracklet Similarity Heatmap: illustrating a diagonal with zero values, which indicates the exact correspondence among tracklet features.

The result for the subsequent step, which is to estimate the best offset based on the similarity calculation for the frames that are part of the vectors of the most similar tracklets, indicates a correspondence in the pairwise comparison of the features of a set of frames belonging to the same trajectory, also resulting in a frame-level correspondence heatmap with the diagonal equal to zero. Figure 6.15 presents this

zero diagonal observed for the frames belonging to tracklet 26, that is, the frames where the object 26 appears.

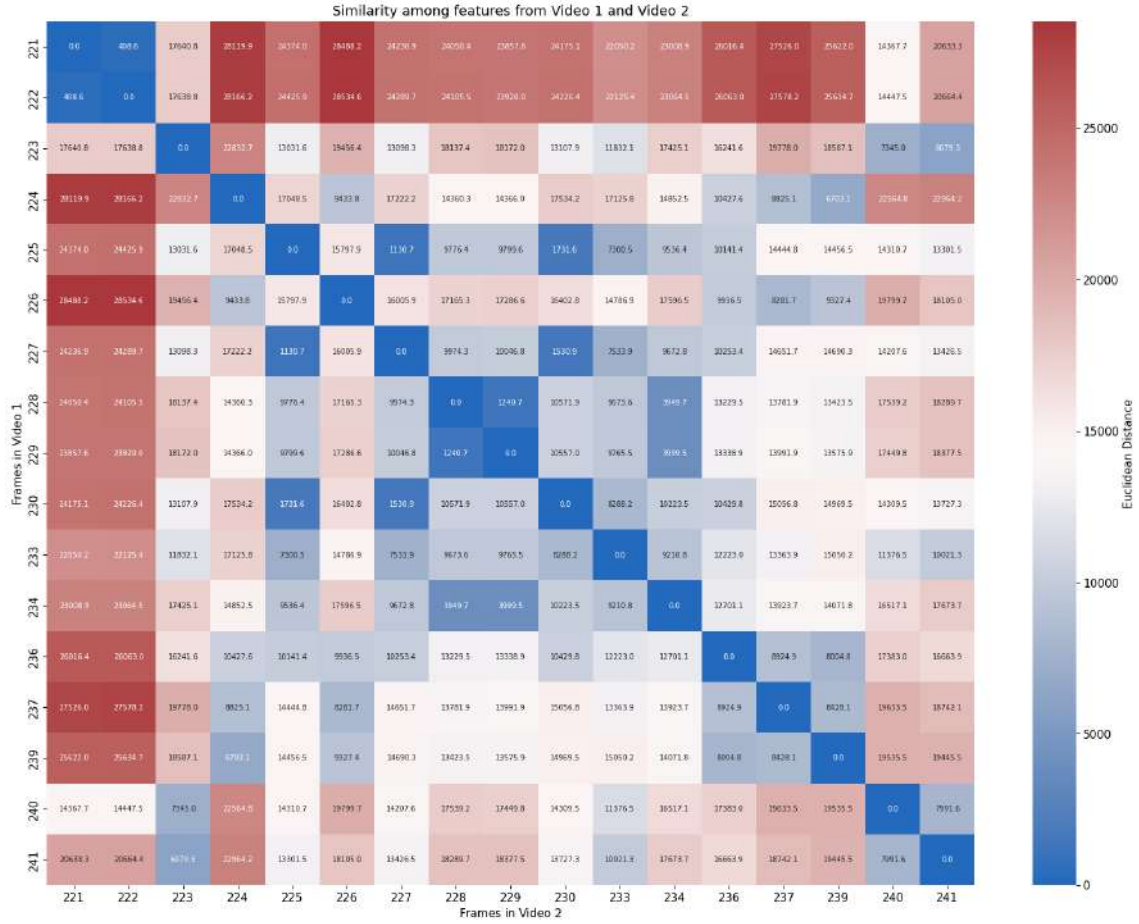


Figure 6.15: Frame-Level Similarities Heatmap: pairwise comparison among frame-level features extracted by two video with the same content, highlighting a diagonal with zero values.

In addition, Figure 6.16 shows a sample of the correspondence of frames according to the estimated offset, which for the case described above is zero. This means that for the set of frames in which the object of tracklet 26 was detected, video 1¹⁰ needs to move zero to overlap its content with video 2¹¹. Figure 6.16 presents a snippet of the script output that indicates the overlap point based on the estimated offset for the case where the processed videos have the exact same visual content.

¹⁰<https://drive.google.com/file/d/18-LR83YoQeoDK70nkTjAW6XThVZ-6j6Z/view>

¹¹<https://drive.google.com/file/d/1xs6xgc4SJ6V3xRmxjPqMePFPnEdoWqL9/view>



Figure 6.16: Offset Estimation Result: pairwise comparison of two overlapped frames.

6.2.2.b. Offset Estimation for a Fragment of the Same Video

A similar analysis was made for the results obtained from the replication of the same experiment, but now for two different videos, with the second video (Video 2¹²) being a fragment cut from the first video (Video 1¹³). Starting with the analysis of the similarity heatmap obtained from the tracklet matching process, it was observed that the script is able to indicate that the objects tracked only in video 1 (the longest video) are very different from the objects tracked in video 2 (images (a) and (b) of Figure 6.17). Now, considering only the most similar objects (similarity score < 1000), it was confirmed that the script was able to correctly indicate, according to the ground-truth value, that Tracklet 22 (video 1) is the closest one to Tracklet 3 (similarity score = 260.3), as highlighted in images (c) and (d) of Figure 6.17.

¹²<https://drive.google.com/file/d/1mXtDi0W3ViP41Q9JRjX65mUI0VLinoJr/view>

¹³<https://drive.google.com/file/d/1EBZ9Y2rgJF2pdmd1jFg7YNq18LqbWtFU/view>

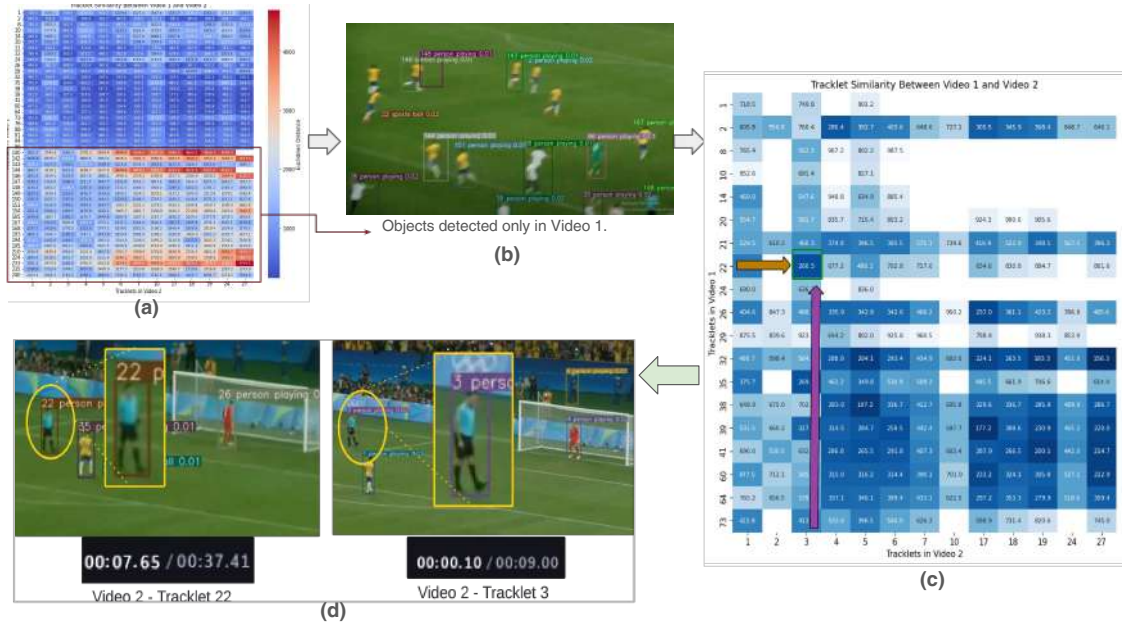


Figure 6.17: Tracklet Similarity Heatmap: (a) all tracklets features vector from video 1 pairwise compared to the ones from video 2; (b) objects detected and tracked only in video 1; (c) tracklet 22 (video 1) is most similar to the tracklet 3 (video 2); (d) sample of the frame illustrating the object detected for the Tracklet 22 and 3.

The result obtained from the offset estimation process, considering only the features of the frames belonging to the trajectories mentioned above, was that, with a Euclidean distance of 298.6, frame 860 (video 1) is the most similar to frame 186 (video 2) (Figure 6.18) from which the estimated offset was 674 (860-186). This means that the content of Video 2 will overlap from frame 674 of video 1.



Figure 6.18: Side-by-side comparison of the frames that has the lowest similarity score, detected automatically by the script.

In addition, by visually analyzing the heatmap returned from the similarity calculation process for the frames (frame-level correspondence), it was possible to identify a pattern that indicated the existence of a diagonal starting in a certain region of the heatmap (Figure 6.19 (a)), which is more evident when we filter out the frames of the video that are more distant similarly from the frames of video 2 (keeping only the frames from 674, for example). This diagonal pattern, visible in dark blue (especially around the middle), suggests that some portions of the two videos align well, meaning that they likely contain similar features in those frames (Figure 6.19 (b)).

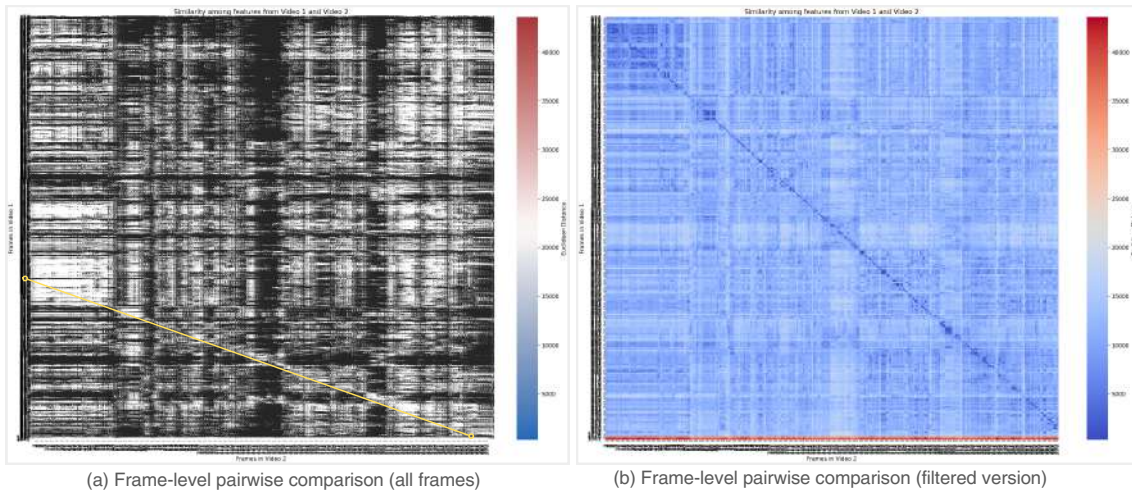


Figure 6.19: Frame-Level Similarity Heatmap highlighting a diagonal pattern indicating the existence of similar features in the frames.

Considering the analysis above, the script was adjusted to created two new fragments of the video 1¹⁴ and video 2¹⁵, respectively, containing the set of frames that has the lowest similarity score from the pairwise comparison of the frame-level feature correspondence process, which should indicate the point of overlap detected based on the visual features from both videos. This can be confirmed by the illustration in Figure 6.20.

¹⁴https://drive.google.com/file/d/1moUJonbsH-bfMRSjd5xwZ_EA91rYwJX8/view

¹⁵https://drive.google.com/file/d/1YGBImjUaJet7EHohszm_c7mNJ8fIC-3A/view



Figure 6.20: Synchronized video fragments generated automatically based on the visual features extracted by the MOT-UGV.

The experiments reproduced for the baseline cases described above confirmed that the visual features extracted by the proposed MOT-UGV pipeline are sufficient to indicate the overlapping point when the two videos share the same visual content. However, these scenarios do not represent the actual problem addressed in this investigation, in which the two videos are recorded by different users, from different cameras, and from different viewpoints of the same event. The next section presents the results obtained when the same pipeline is applied to this harder multi-view, cross-view condition, and analyzes the challenges that arise from it.

6.3 Multi-View UGV Synchronization

The results presented in Section 6.2 validated the proposed pipeline in controlled conditions, where the two processed videos share the same content (the same video processed twice or a fragment cut from the same video). In those cases, the similarity calculation among the tracklets and the offset estimation worked as expected, producing a clear correspondence between frames. However, when the same pipeline was applied to the actual target scenario of this investigation, the problem became a challenge. In a real multi-view setting, the two UGVs are recorded by different users, with different devices and from different positions of the stadium, so the same object appears under different scales, poses, and backgrounds across the two videos.

This difference exposes a limitation of the approach as it was conceived until

the qualification stage of this master’s research. The synchronization based only on MOT-UGV features relies on matching tracklets of the same object across the two videos, that is, it depends on an *intra-class*, identity-oriented correspondence, where the ReID embedding must recognize that “player A in video 1” is the same “player A in video 2”.

This assumption holds when the viewpoints are close, but it breaks down when the cameras are far apart, because the appearance of the same athlete changes substantially and the ReID embedding, trained for person ReID, was not designed to be invariant to such large viewpoint changes. This observation motivates a different hypothesis, described in this section: that cross-view temporal alignment is better supported by a *holistic* representation of the scene as a whole rather than an identity-specific representation of each individual object.

To investigate this, two representations were compared using the same pipeline and evaluation protocol. Those are:

1. OSNet-AIN [Zhou et al. 2022], a re-identification embedding that serves as the baseline and evidences the limitation described above;
2. DINOv2 [Oquab et al. 2023], a self-supervised ViT embedding that encodes the frame as an integrated whole.

6.3.1 UGV-Sync: Custom Annotated Dataset

To enable the quantitative evaluation of the proposed framework, UGV-Sync, a custom multi-view UGV dataset was built by combining the original MUVY¹⁶ samples with additional samples selected from MUVS¹⁷ dataset. The MUVY-derived subset contains 63 video pairs selected (Subsection 6.1.1.b). Each pair contains two views of the same event and was trimmed to a short segment of at most 15 to 20 seconds, keeping the annotation feasible and concentrating the comparison on a temporal window in which the same action is visible in both views.

The temporal offset of the MUVY subset was manually annotated as ground truth. For this purpose, a custom annotation and inspection tool was developed to

¹⁶<https://swperfi-project.github.io/Pages-dev/MuvyDataset-page/>

¹⁷<https://swperfi-project.github.io/Pages-dev/MuvsDataset-page/>

display the two videos side by side and allow frame-by-frame inspection for alignment. In the current version of the evaluation workflow, the web tool, **UGV-Sync Studio**, extends this process by also supporting the MUVS-derived pairs, loading the raw clips, the tracking videos, and the JSON results produced by the evaluation pipeline from a unified folder structure. The tool also allows the annotator to update the manifest directly when the manually selected offset is corrected during visual inspection (See Figure 6.21).

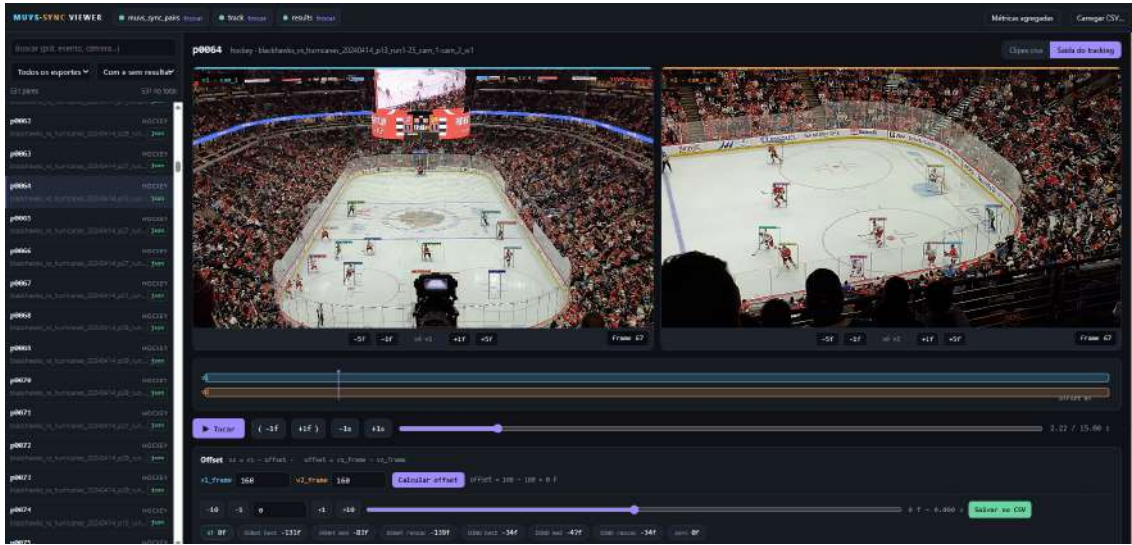


Figure 6.21: Screenshot of the custom UGV-Sync Studio, the tool used to support the validation of synchronized video pairs.

The MUVS-derived subset was constructed automatically from the MUVS fragments¹⁸. For each event and period, the generation script enumerates valid camera pairs and cuts fixed-length clips from temporally corresponding fragment windows. The timestamp-based alignment provides the nominal temporal correspondence, and the per-fragment residual offsets are used to compute the final reference offset. To avoid ambiguous ground truth, only clips entirely contained inside a stable offset run are considered. Each accepted pair is written to the same manifest format used by the MUVY subset.

The final evaluation dataset, therefore, contains both manually annotated MUVY pairs and timestamp-supported MUVS-derived pairs. The MUVS-derived subset contains 468 generated pairs, while the MUVY subset contains 63 manually an-

¹⁸<http://zenodo.org/records/20708683>

notated pairs, resulting in 531 pairs in the unified evaluation manifest. Table 6.5 summarizes the composition by source.

Table 6.5: Composition of the UGV-Sync dataset by source.

Source	Pairs
MUVS (timestamp-supported pairs)	468
MUVY (manually annotated pairs)	63
Total	531

The evaluation set spans eight sports, with an unbalanced number of pairs per sport, which reflects the availability of multi-view content for each modality. Table 6.6 reports the number of pairs annotated pairs per sport.

Table 6.6: Composition of the UGV-Sync dataset (513 pairs) per sport.

Sport	# Pairs
American football	13
Artistic gymnastics	8
Basketball	127
Cricket	1
Hockey	118
Soccer	142
Tennis	2
Volleyball	120
Total	531

6.3.2 Offset Estimation Strategies

Before comparing the two representations, the strategy used to convert the tracklet-level similarities into a single offset estimate is recapitulated, since it was refined after the qualification stage. Up to that point, the offset was obtained from the minimum-distance match and from a simple average (or median) of the offsets suggested by the most similar tracklet pairs. However, this consensus was easily pulled away

from the correct value by low-quality matches, which behave as outliers in the set of candidate offsets.

To make the consensus more robust, a RANSAC-based voting was adopted. Instead of averaging the candidate offsets, each confident pair of tracklets proposes one offset hypothesis, and the hypothesis that gathers the largest number of inliers within a tolerance of $\pm\delta$ frames is selected as the estimated offset. This makes the estimation depend on the agreement among several matches rather than on a single best match, reducing the influence of noise correspondences. In the results that follow, three strategies for completeness are reported: *best match*, which takes the offset of the single lowest-distance tracklet pair; *median*, which takes the median of all candidate offsets; and *RANSAC*, the inlier-voting consensus described above.

For completeness, we report results for three offset estimation strategies in the experiments that follow:

- **Best match:** Takes the offset of the single lowest-distance tracklet pair.
- **Median:** Takes the median of all candidate offsets.
- **RANSAC:** The inlier-voting consensus described above and formalized in Theorem 1.

Figure 6.22: Synchronization accuracy as a function of the tolerance threshold τ for OSNet-AIN and DINOv2, under the best-match and the RANSAC strategies. The RANSAC strategy consistently improves upon the best-match approach, particularly for OSNet-AIN, corroborating the theoretical prediction of Theorem 1.

As demonstrated in Figure 6.23, the RANSAC strategy improves the accuracy of OSNet-AIN at tighter tolerances (Acc@5 from 15.87% to 25.40%), corroborating the theorem’s thesis that consensus among multiple hypotheses is more robust than relying on a single best match.

Validation of the Matching Strategy

A complementary experiment was carried out to validate which strategy better recovers the best offset when matching the tracklet sequences. For this purpose, a

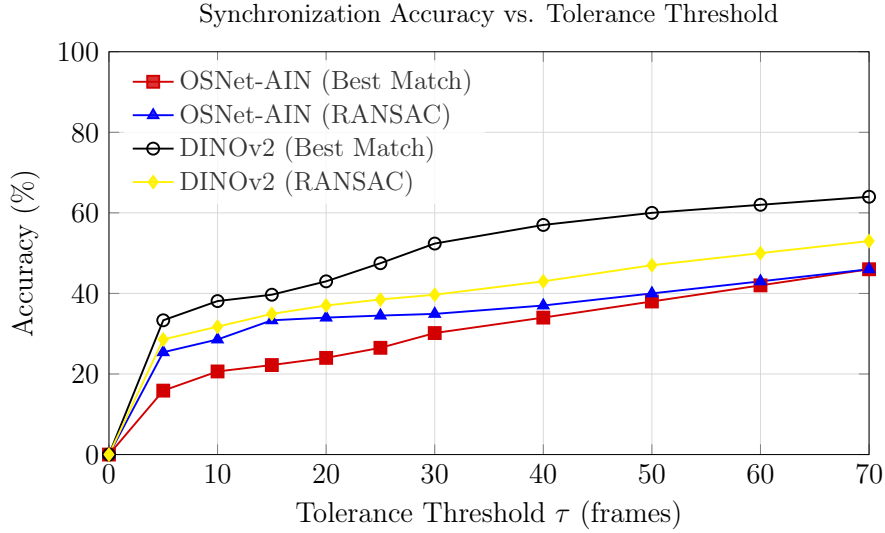


Figure 6.23: Synchronization accuracy as a function of the tolerance threshold τ for OSNet-AIN and DINOv2, under the best-match and the RANSAC strategies. The RANSAC strategy consistently improves upon the best-match approach, particularly for OSNet-AIN, corroborating the theoretical prediction of Theorem 1.

smaller and controlled set of 6 annotated pairs was used to compare two matching strategies.

The first one, referred to as *Standard*, performs the correspondence through the ANN-based Euclidean match between tracklet features. The second one, as called *DTW-Norm*, aligns the full tracklet sequences temporally using a normalized Dynamic Time Warping FastDTW [Salvador and Chan 2007]. The normalization (standardization of the ReID and division of the bounding-box coordinates by the frame dimensions) is applied so that the bounding-box magnitudes do not dominate the distance.

Table 6.7 presents per-pair results in six video pairs (selected from the MUVY-Sync set) that span soccer, artistic gymnastics, and basketball, under three recording conditions: same-view trimmed clips (SV), different views with small angular separation (DV-near), and different views with large angular separation (DV-far).

Same-view (SV): The same-view pair ($GT = 674$) is the most controlled scenario: both clips are extracted from the same camera recording and contain 23 and 14 tracklets (≥ 15 frames), respectively. Both methods recover the exact offset ($MAE = 0$), with RANSAC inlier ratios of 0.21 (Standard) and 0.20 (DTW-Norm),

Table 6.7: Synchronization results on 6 UGV sports evaluation set. **GT** is the ground-truth temporal offset in frames (positive = Video 1 leads Video 2). **Rec.** is the RANSAC-recommended offset. $|MAE|$ is the absolute error in frames (lower is better). **IR** is the RANSAC inlier ratio.

Pair	Sport	GT (fr)	Scenario	Standard			DTW-Norm		
				Rec.	$ MAE $	IR	Rec.	$ MAE $	IR
(cam_03_S01, cam_03_S01_trim)	Soccer	674	SV	674	0	0.21	674	0	0.20
(cam_02_AG01, cam_01_AG01)	Gym	74	DV-near	74	0	0.15	145	71	0.18
(N2EC140214, N2ER090188)	Basket	139	DV-near	138	1	0.24	219	80	0.13
(N2ER090188, N9BA420582)	Soccer	65	DV-near	84	19	0.21	48	17	0.20
(cam_01_AG03, cam_02_AG03)	Gym	181	DV-far	1275	1094	0.13	156	25*	0.17
(cam_02_S01, / cam_04_S01)	Soccer	-206	DV-far	-65	141	0.15	37	243	0.29

based on 62 and 10 matched pairs. The shared embedding space visualized via UMAP (Figure 6.24-A) confirms the strong representational overlap. The clusters in cyan (Video 1) and magenta (Video 2) are closely interleaved in the overlay panel, providing qualitative evidence of the tracklet correspondences that align with the correct offset estimated.

Different views with small angular distance (DV-near): Three pairs fall into this category. For the Artistic Gymnastics pair AG01 (single athlete, close viewpoints, GT = 74), Standard achieves a perfect estimate ($MAE = 0$, 26 matched pairs), while DTW-Norm diverges to $MAE = 71$ fr. The degradation of DTW-Norm is attributable to the minimum-length filter applied during the DTW pass, only 11 tracklets were returned after the 20-frame threshold, reducing the match pool and increasing sensitivity to noise.

For the basketball pair, Standard again outperforms with $MAE = 1$ fr. The UMAP overlay for this pair (Figure 6.24-C) exhibits inter-video cluster alignment, consistent with the large match count. For the pair (N2ER090188, N9BA420582), neither method reaches the exact offset: standard yields $MAE = 19$ and DTW-Norm $MAE = 17$. Both errors correspond to sub-second alignment (<0.7 s at 30 fps), which would be acceptable for many broadcast applications, but the underlying cause a sparse match pool of 11–15 tracklets limits consensus reliability.

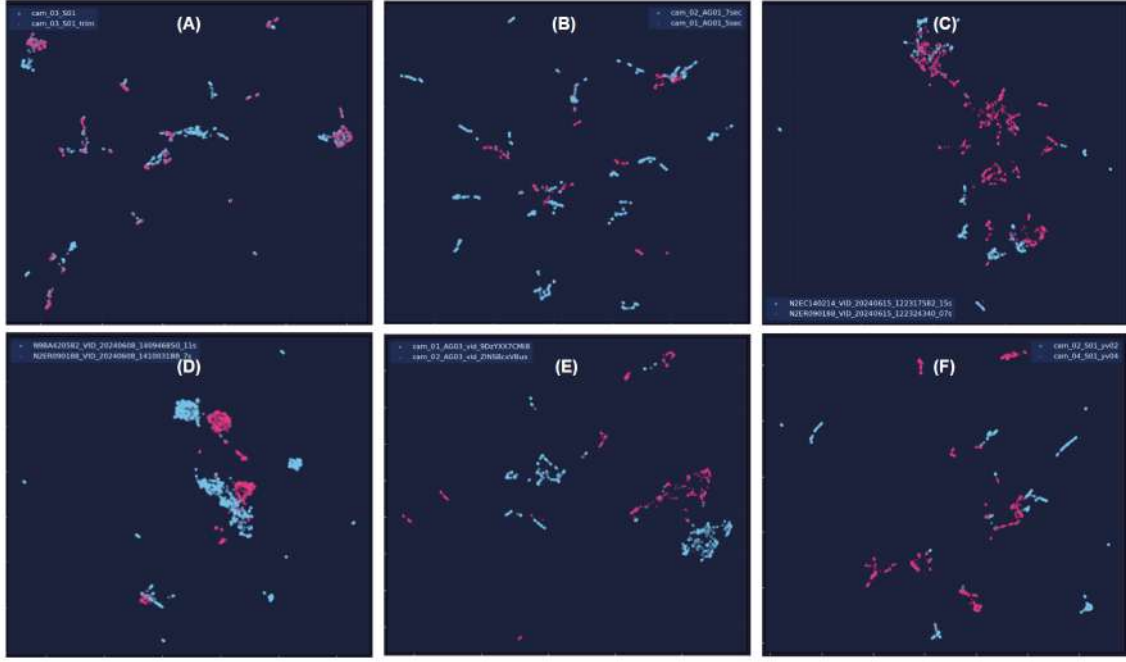


Figure 6.24: UMAP projections of OSNet-AIN tracklet embeddings for all six UGV evaluation pairs (tracklets < 15 frames excluded). Cyan = Video 1; magenta = Video 2.

Different views with large angular distance (DV-far): Both pairs in this condition expose the principal limitation of MOT-UGV-based synchronization. For the Gym pair AG03 (large viewpoint difference, $GT = 181$), Standard presented a high ($|MAE| = 1094$, equivalent to ~ 36 s), while DTW-Norm substantially reduced the gap to $MAE = 25$ (< 1 s at 30 fps). The improvement of DTW-Norm in this case is attributable to the spatial trajectory constraint, that is, by jointly considering feature similarity and bounding-box trajectories, DTW-Norm partially compensates for the appearance inconsistency that arises when the same athlete is seen from divergent angles.

Note that the match pool is small ($N = 6$ for DTW-Norm), so the estimate is highly uncertain. For the Soccer pair (cam_02_S01 / cam_04, $GT = -206$), both methods fail entirely: Standard presented $MAE = 141$ and DTW-Norm $MAE = 243$. The UMAP overlay for this pair (Figure 6.24-F) justifies this failure geometrically: the embedding clouds from these two videos occupy largely disjoint regions of the vector space, confirming that the embedding features returned from OSNet-AIN (which was trained on pedestrian ReID data) do not generalize across

extreme viewpoint changes in multi-player sports scenes, like soccer.

Standard (EuclideanDistance-based) outperforms DTW-Norm in three of the four SV / DV-near scenarios, benefiting from a larger match pool and lower computational cost (< 5 s per pair on GPU vs. up to 32 s for DTW-Norm). The DTW-Norm is preferable in the DV-far regime, where spatial trajectory continuity partially compensates for appearance inconsistency. In general, Track-UGV-SYNC achieves exact or near-exact synchronization (< 1 s at 30 fps) in four of six pairs (Standard) and three of six (DTW-Norm). Failure cases are concentrated in the DV-far condition, where the tracklet embedding space shows minimal inter-video overlap (see Figure 6.24).

6.3.3 Comparison of Visual Representations

The results described in this subsection evaluate the two visual representations analysis on the 531 annotated pairs (5.5.1), keeping every other component of the pipeline unchanged. Both conditions share the same detections and the same track identities produced by MOT-UGV in Stage I of the framework; only the feature vectors used to compare the tracklets differ. In the first condition, the 512-dimensional ReID features extracted by OSNet-AIN during tracking are used. In the second condition, these features are replaced by the 768-dimensional DINOv2 [CLS] embeddings, re-extracted from the same tracked crops.

The consolidated synchronization evaluation contains 531 video pairs from the unified UGV-Sync dataset (Table 6.8). The reference offset was validated through frame-level inspection. For the MUVS-derived pairs, sensor timestamps were used as an initial temporal anchor and subsequently compared with the validated offset. Thus, the timestamp condition should be interpreted as a sensor-derived baseline rather than as an independent ground truth. For the comparison between visual representations, the same detections, tracklets, trajectory filtering, and synchronization pipeline were preserved. Only the embedding representation extracted from the tracked crops was changed, allowing a controlled comparison between identity-oriented OSNet-AIN features and self-supervised DINOv2 features.

Table 6.8: Consolidated synchronization error on the 531 evaluated pairs. MAE is reported in frames and Acc@5 is the percentage of estimates within five frames of the validated offset.

Condition	Estimator	MAE	Acc@5 (%)
Timestamp-based sensor offset	–	46.2	57.1
OSNet-AIN	Best match	142.6	6.2
	Argmax	65.3	8.5
	RANSAC	125.8	7.9
DINOV2	Best match	128.6	9.0
	Argmax	54.8	13.7
	RANSAC	96.1	13.0

The timestamp-based baseline obtained the best aggregate result, with an MAE of 46.2 frames and an Acc@5 of 57.1%. This indicates that, when sensor information is available and sufficiently reliable, it remains more precise than the current visual-only pipeline, particularly at strict tolerances. The relevance of Track-UGV-Sync is therefore not that it uniformly surpasses timestamps, but that it can estimate temporal alignment when timestamps, or any other sensor metadata are unavailable.

Among the visual approaches, DINOV2 consistently outperformed OSNet-AIN under all three offset estimators. With the Argmax estimator, DINOV2 reduced the MAE from 65.3 to 54.8 frames and increased Acc@5 from 8.5% to 13.7%. Under RANSAC, the MAE decreased from 125.8 to 96.1 frames, while Acc@5 increased from 7.9% to 13.0%. These results support the conclusion that self-supervised semantic features are more suitable than identity-oriented ReID features for the evaluated cross-view synchronization task.

Figure 6.25 presents the accuracy over increasing tolerance thresholds. The sensor baseline remains clearly superior at the strictest tolerances. Nevertheless, DINOV2 remains above OSNet-AIN for both best-match and RANSAC estimation. The figure also reveals a relevant estimator trade-off. The Argmax strategy produces the lowest aggregate MAE, whereas the DINOV2–RANSAC combination becomes progressively more competitive as the tolerance increases, reaching approximately 53% accuracy at a tolerance of 60 frames, compared with approximately 36% for OSNet-AIN–RANSAC.

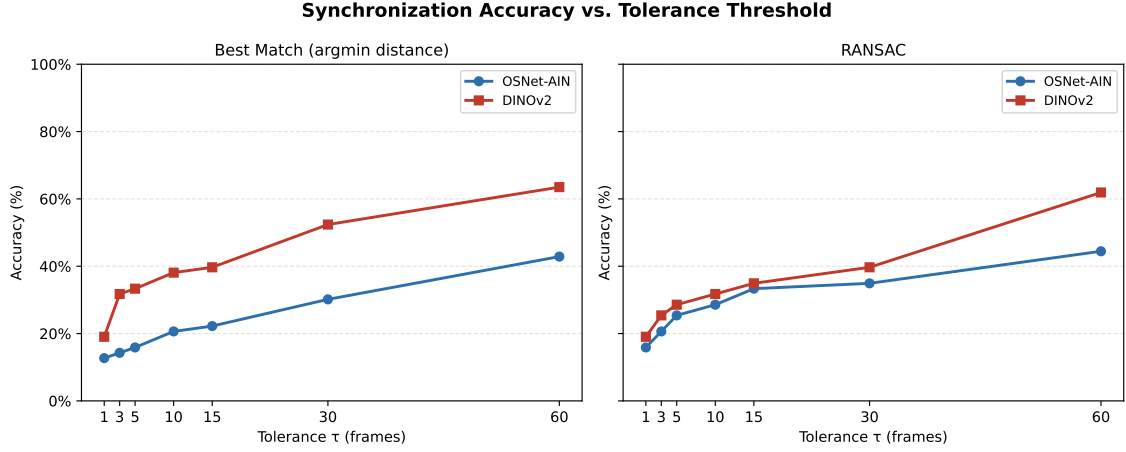


Figure 6.25: Synchronization accuracy across tolerance thresholds. The timestamp curve is repeated in both panels because it does not depend on the visual offset estimator.

A per-sport breakdown analysis showed that the overall gain is not uniform across modalities. Table 6.9, Figures 6.26, and 6.27 report the MAE and the Acc@5 per sport for the best-match strategy. The most expressive improvement is observed in artistic gymnastics, where the MAE drops from 106.50 to 4.38 frames and the Acc@5 increases from 12.50% to 87.50%. Tennis and volleyball also benefit strongly from DINOv2, with the error reduced to a few frames. On the other hand, American football and soccer remain challenging for both representations. For soccer in particular, DINOv2 does not reduce the MAE (it even increases from 89.68 to 107.12 frames), although the median error is reduced by half (from 87 to 44 frames) and the Acc@5 slightly improves (from 12.00% to 16.00%); this behavior is consistent with a small number of pairs producing very large errors that inflate the average.

Table 6.9: Per-sport results on MUVY-Sync set (best-match strategy). MAE in frames; Acc@5 in percentage. The cricket and hockey pairs are same-view trims (see Subsection 6.3.1).

Sport	# Pairs	OSNet MAE	OSNet Acc@5	DINOv2 MAE	DINOv2 Acc@5
American football	13	198.23	0.00	165.23	7.69
Artistic gymnastics	8	106.50	12.50	4.38	87.50
Basketball	10	79.80	10.00	61.20	30.00
Cricket	1	4.00	100.00	0.00	100.00
Hockey	1	0.00	100.00	0.00	100.00
Soccer	25	89.68	12.00	107.12	16.00
Tennis	2	8.50	50.00	1.00	100.00
Volleyball	3	23.67	66.67	2.67	66.67
Total	63	104.14	15.87	87.03	33.33

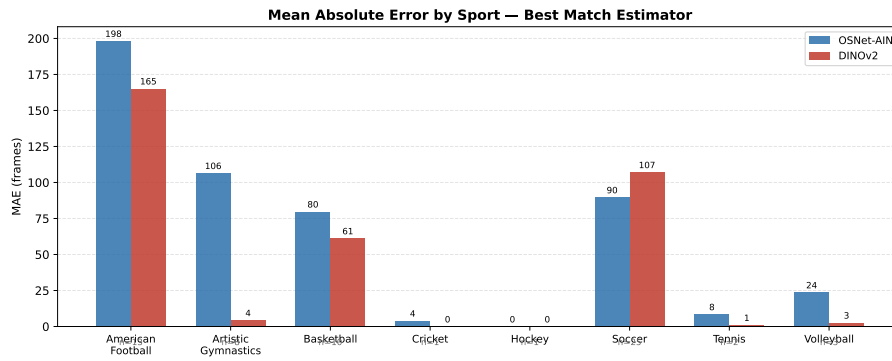


Figure 6.26: Mean Absolute Error per sport for OSNet-AIN and DINOv2 (best-match strategy).

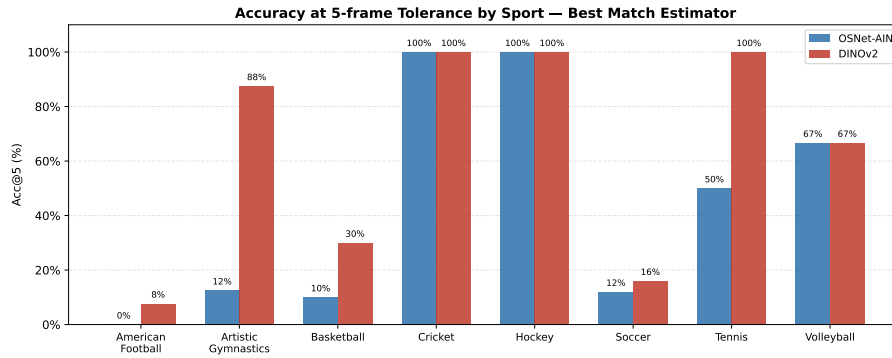


Figure 6.27: Accuracy at a 5-frame tolerance per sport for OSNet-AIN and DINOv2 (best-match strategy).

6.3.4 Representational Analysis of the Cross-View Failure

To understand why the error concentrates in the different-camera pairs, the embedding spaces of both visual representations were inspected by projecting the tracklet features of a representative cross-view pair into two dimensions, using PCA, t-SNE and UMAP [McInnes et al. 2018]. In each projection, the points are colored in three ways: by the source video, by the track identity, and by the frame index.

When the points are colored by source video, the OSNet-AIN projection tends to separate the two videos into distinct regions of the space, that is, the features carry information about which camera produced them rather than about what is happening in the scene. With DINOv2, the points from the two videos are more interleaved, which is consistent with a representation that is more invariant to the viewpoint and that places visually similar moments close together regardless of the camera. This is illustrated in Figure 6.28.

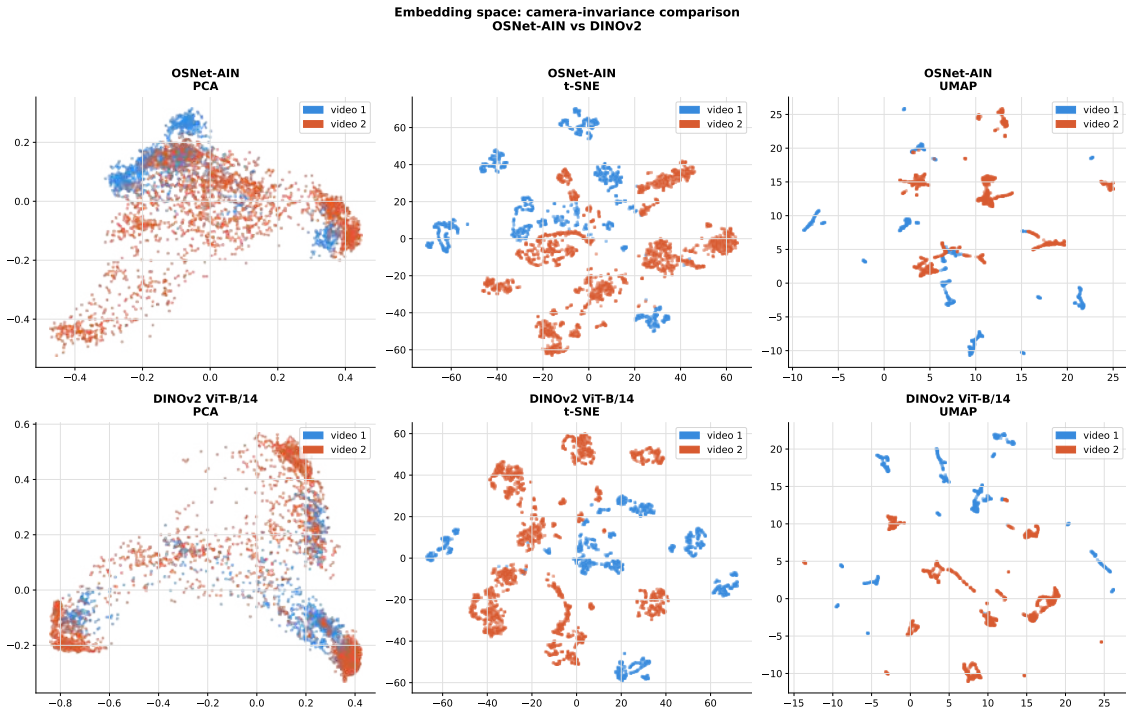


Figure 6.28: Two-dimensional projection of the tracklet embeddings of a cross-view pair for OSNet-AIN and DINOv2, coloured by source video. The two videos remain more separated for OSNet-AIN and more interleaved for DINOv2.

This analysis supports the central interpretation of these results, which is that the failure observed in the most distant cross-view condition is mainly due to the

way the visual features are represented (*i.e.*, representational failure) and due to the algorithm of the proposed workflow. The synchronization procedure is the same in both conditions and works well whenever the representation places corresponding moments close together in the embedding space; what changes is the ability of the representation to remain stable across large viewpoint changes.

OSNet-AIN, specialized in intra-class identity discrimination, was not built for this, whereas DINOv2, by encoding the frame as a whole through global self-attention, provides a more visual, semantically rich, and viewpoint-stable description of the scene. This motivates the idea of comparing videos at the frame level using scene-level scene signatures, that is, a semantic search over the entire frame rather than an intra-class match between individual objects, which is identified as the main direction for future work.

6.3.5 Comparison with VideoSync and Whole-Frame Representations

An additional zero-shot benchmark was conducted to compare the proposed object-centric formulation with whole-frame synchronization approaches. All conditions were evaluated on the same 531 video pairs and against the same validated ground-truth offsets. The comparison included CARL, the frame-level representation adopted by [Shin and Molybog 2025], and a complementary full-frame DINOv2 ablation. The latter uses the same ViT-B/14 backbone adopted in Track-UGV-Sync, but extracts one descriptor from each complete frame. In contrast, the object-centric DINOv2 condition extracts features only from frames belonging to significant tracklets. This comparison therefore helps separate the effect of the visual backbone from the effect of representation granularity.

For the whole-frame conditions, two non-trained offset estimators were evaluated: independent nearest-neighbour matching, denoted as Argmax, and Dynamic Time Warping (DTW). The trained CNN classifier originally proposed by VideoSync was not included because its output space was learned for offsets restricted to $[-30, +30]$ frames [Shin and Molybog 2025]; and would require retraining on the UGV-Sync dataset.

The overall results demonstrate a strong interaction between representation gran-

Table 6.10: Overall comparison of object-centric and whole-frame synchronization conditions on the 531 UGV-Sync pairs.

Representation	Unit	Estimator	MAE ↓	Acc@5 ↑	Acc@15 ↑	Acc@30 ↑
DINOv2	Full frame	DTW	36.39	49.72	65.54	73.82
VideoSync-CARL	Full frame	DTW	36.84	44.82	63.28	73.82
Timestamp	Metadata	Direct	46.16	57.06	63.28	66.10
DINOv2	Significant tracklets	Argmax	54.80	13.75	29.76	46.52
VideoSync-CARL	Full frame	Argmax	63.02	20.15	41.24	53.86
OSNet-AIN	Significant tracklets	Argmax	65.33	8.47	22.03	38.61
DINOv2	Full frame	Argmax	100.57	5.27	12.24	25.05

ularity and temporal alignment. Whole-frame DINOv2 with independent Argmax matching achieved an MAE of 100.57 frames, whereas applying DTW to the same embeddings reduced the MAE to 36.39 frames and increased Acc@5 from 5.27% to 49.72%. CARL exhibited the same tendency, with its MAE decreasing from 63.02 to 36.84 frames when Argmax was replaced by DTW. Consequently, the improvement cannot be attributed to whole-frame representations alone. Explicitly enforcing an ordered temporal path was essential to transform frame-level visual similarities into reliable offset estimates.

The whole-frame DTW conditions achieved the best aggregate visual results. However, this result must be interpreted together with the distribution of the viewpoint classes. The benchmark contains 402 DV-far pairs, 106 DV-near pairs, and only 23 same-view pairs. Therefore, the aggregate result is largely determined by the DV-far condition.

Table 6.11: Best-performing visual conditions by viewpoint class.

View class	Representation	Estimator	MAE ↓	Acc@5 ↑	Acc@30 ↑
DV-far	VideoSync-CARL full frame	DTW	32.51	50.75	75.62
DV-far	DINOv2 full frame	DTW	33.67	53.73	75.37
DV-near	DINOv2 full frame	DTW	35.97	37.74	71.70
DV-near	DINOv2 significant tracklets	Argmax	39.33	18.87	54.72
SV	DINOv2 significant tracklets	RANSAC	14.78	60.87	86.96

For DV-far pairs, CARL-DTW achieved the lowest MAE, while full-frame DINOv2-DTW obtained the highest strict-tolerance accuracy. These results indicate that

global scene information becomes particularly important when individual objects undergo substantial changes in scale, pose, resolution, and background across cameras. Whole-frame descriptors retain the collective configuration of the players, field or court geometry, camera motion, and the global progression of the sporting action, which may remain informative even when direct object correspondence is unreliable.

For DV-near pairs, the difference between full-frame and object-centric DINOv2 was smaller. Full-frame DINOv2-DTW achieved an MAE of 35.97 frames, whereas object-centric DINOv2 reached 39.33 frames. This suggests that significant tracklets still retain useful correspondence information when viewpoint differences are moderate, although global context and DTW improve the proportion of predictions within strict tolerances.

The same-view condition exhibited the opposite behavior. Object-centric DINOv2 combined with RANSAC achieved an MAE of 14.78 frames and an Acc@5 of 60.87%, substantially outperforming all whole-frame conditions. In this scenario, tracked objects remain visually comparable across videos, and removing background information reduces ambiguous scene-level matches. The higher RANSAC inlier ratio observed for same-view pairs also indicates that a dominant set of temporally consistent object correspondences is available. Under large viewpoint differences, this inlier set becomes sparse, preventing RANSAC from recovering a correct offset when most candidate matches are already visually inconsistent.

These findings show that neither representation granularity is universally superior. Object-centric tracklet representations are particularly effective when object appearance remains comparable, while whole-frame representations combined with explicit temporal alignment are more robust under large cross-view differences. This motivates a hybrid formulation in which scene-level DTW provides a global temporal hypothesis and significant tracklets are subsequently used to validate or refine the estimated offset.

Beyond the quantitative comparison above, a qualitative pattern is consistent. Pairs recorded at approximately the same elevation as the players (i.e., from the stands or sideline, at a height close to the action itself), consistently yield offsets closer to the validated ground truth. Figure 6.29 shows a representative example: a pair recorded from the height of the spectator, where tracked subjects remain large,

well separated, and visually distinct between both views.

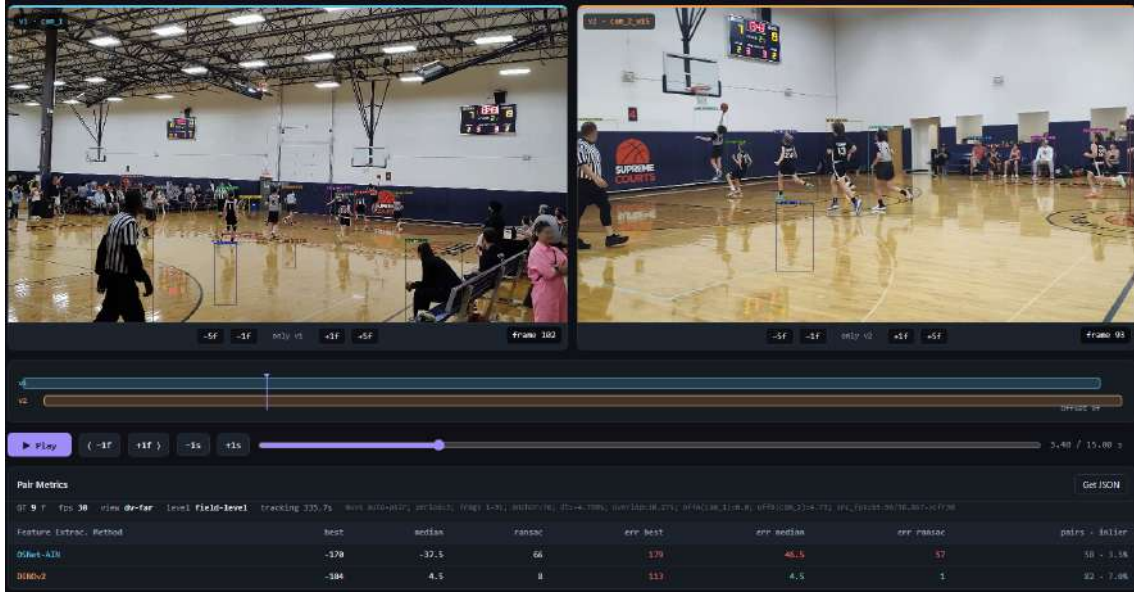


Figure 6.29: Example of a DV-Far UGV pair recorded at the same elevation as the players (field-level), shown in the UGV-Sync Studio. The bottom panel reports the per-pair metrics (ground-truth offset=9 and the OSNet-AIN / DINOv2 estimates under each estimator).

Conversely, Figure 6.30 illustrates an example when a camera is positioned above the field, at an angle that looks down onto the scene, the framework has more difficulty distinguishing individual subjects, especially when those subjects share the same visual appearance (e.g., players wearing the same team uniform). This is consistent with the representational limitation already identified in this work, which is regarding the embeddings used for matching being sensitive to appearance, and a steep, elevated viewpoint reduces the amount of distinctive visual information available per subject, independently of which embedding (OSNet-AIN or DINOv2) is used.

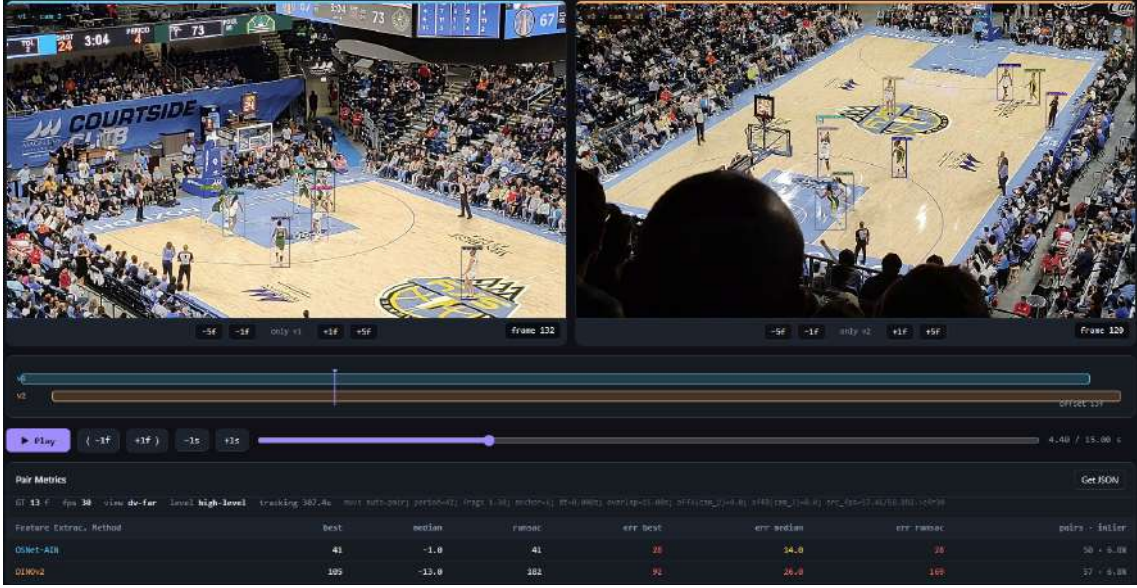


Figure 6.30: Example of a DV-Far UGV pair recorded at an angle that looks down onto the scene (high-level), shown in the UGV-Sync Studio.

6.3.6 Search Space Reduction

The formal analysis of computational efficiency presented in Proposition 1 (Section 3.1.2) predicted a significant reduction in the search space by operating at the tracklet level rather than the frame level. Here, we provide empirical validation of this theoretical claim. Table 6.12 quantifies the reduction achieved by the tracklet-based formulation on the MUVY evaluation set (63 pairs).

Table 6.12 reports the mean and median number of frames per video, raw and significant tracklets, and the resulting candidate reduction factors. F is the number of frames per video. T_{raw} and T_{sig} are the number of tracklets per video before and after the significance filter (≥ 20 frames). The candidate reduction factor is $\rho = (F_1 \times F_2) / (T_1 \times T_2)$, i.e. how many fewer pair hypotheses Stage II enumerates with respect to a frame-exhaustive search. Wall-clock speed-up is measured on a subset of 12 timed pairs.

Table 6.12: Empirical search-space reduction of the tracklet-based formulation on the MUVY evaluation set.

Quantity	Mean	Median
Frames per video, F	390.9	361.5
Tracklets per video, T_{raw}	32.6	20.5
Significant tracklets, T_{sig}	14.4	11.0
Candidate reduction $\rho_{\text{raw}} (\times)$	691.6	352.3
Candidate reduction $\rho_{\text{sig}} (\times)$	5702.5	1843.0
Wall-clock speed-up, Standard $(\times)$	93.4	90.5
Wall-clock speed-up, DTW-Norm $(\times)$	39.4	1.7

The results in Table 6.12 empirically validate Proposition 1. Replacing the exhaustive frame-pair search ($F_1 \times F_2$) with a tracklet-correspondence search ($T_1 \times T_2$) reduces the number of candidate hypotheses by a median factor of approximately $352\times$ (raw tracklets) and $1843\times$ when only significant tracklets are retained. This reduction translates directly into computational savings: in wall-clock terms, the deployed Standard variant runs approximately $90\times$ faster than the frame-exhaustive baseline on the timed subset.

The computational gain comes from candidate reduction, vectorization, and class/length filtering. However, the total number of elementary feature comparisons is not itself reduced, since each tracklet still spans several frames. The key insight is that the search

Chapter 7

Conclusions

This work presented Track-UGV-Sync, a vision-based framework for the temporal synchronization of multi-view sports user-generated videos using visual representations extracted from tracked objects. Its main scientific contribution lies in investigating synchronization under uncontrolled recording conditions that approximate real-world use: independently operated handheld cameras, heterogeneous viewpoints, occlusions, camera motion, scale variation, repeated uniforms, and the absence of shared timestamps (sensor metadata). Although previous visual synchronization methods have demonstrated promising results in controlled, synthetic, or rigorously preprocessed scenarios, these conditions are not jointly addressed in the literature together with object-centered tracklets, explicit viewpoint analysis, and real sports UGVs.

The consolidated evaluation on 531 video pairs showed that visual-only synchronization is feasible, although its effectiveness depends strongly on both the visual representation and the recording condition. Among the object-centered approaches, DINOv2 consistently outperformed the identity-oriented OSNet-AIN embeddings. Using the Argmax estimator, DINOv2 reduced the MAE from 65.3 to 54.8 frames and increased Acc@5 from 8.5% to 13.7%. This result indicates that self-supervised semantic representations are more suitable than ReID features for finding temporal correspondences across different cameras. The MOT-UGV preprocessing stage also improved the main tracking metrics over the original YOLOv7–StrongSORT baseline on MOT20, increasing HOTA from 9.09 to 33.25, IDF1 from 6.23 to 43.79, and MOTA from 8.58 to 32.46. Nevertheless, the direct relationship between tracking

quality and synchronization error remains to be investigated.

The comparison with whole-frame approaches provided a more nuanced answer regarding representation granularity. Full-frame DINOv2 combined with DTW achieved the best overall visual result, with an MAE of 36.39 frames and an Acc@5 of 49.72%, closely followed by VideoSync-CARL with DTW. These results show that global scene information and explicit temporal sequence alignment are particularly important in far cross-view conditions, where direct object correspondence becomes unreliable. Conversely, object-centered DINOv2 with RANSAC achieved the best result in the same-view condition, reaching an MAE of 14.78 frames and an Acc@5 of 60.87%. Therefore, neither global nor local representations are universally superior: global features are more robust to large viewpoint changes, whereas significant tracklets provide more discriminative evidence when object appearance remains comparable. In addition, the tracklet-based formulation reduces the number of candidate offset hypotheses by restricting the search to frames belonging to significant trajectories, rather than exhaustively comparing every frame of one video with every frame of the other.

These findings define the practical operating range of the proposed framework. Track-UGV-Sync is most effective when the videos preserve sufficient visual correspondence between tracked objects, while its performance degrades under large angular differences, severe scale changes, partial visibility, elevated cameras, and scenes containing visually similar players. The results also show that a larger number of matched tracklet pairs does not necessarily imply a lower synchronization error. The quality and temporal discriminability of the correspondences are more important than their quantity. Although sensor timestamps obtained the best strict-tolerance accuracy when available, the relevance of the proposed framework is that it remains applicable when such metadata are missing or unreliable.

Regarding the data used to evaluate these results, they were selected from the three new UGV-based datasets (i.e., UVY, MUVY, and MUVS) developed in collaboration with the ALGOX research group. UVY, the first of these, contributed to the selection, adaptation, and evaluation of the MOT-UGV. The other two served as a source for validating and testing the synchronization experiments and the multi-view analysis, since they contain multiple videos recorded at the same event. UVY and

MUVY contains videos selected exclusively from YouTube, while MUVS consists of videos recorded by the authors of the dataset, with the differential of collecting metadata (sensor data) during the recording process, which includes the timestamp information. The contribution of these datasets was reinforced by their acceptance in the literature (see Table 7.1).

7.1 Future Works

Limitation: The framework currently operates only on video pairs and assumes that the two input videos contain at least one temporal overlap of the same event. It was not evaluated in cases where no common scene exists, and therefore it does not yet include a mechanism for rejecting non-overlapping pairs. The benchmark also contains several pairs derived from the same events and camera sequences, which reduces the effective number of independent observations. Furthermore, the distribution of viewpoint classes is unbalanced, and the relationship between tracking errors, track fragmentation, and downstream synchronization accuracy was not isolated through a controlled ablation.

Future work will prioritize an adaptive synchronization architecture that first estimates the visual complexity and viewpoint relationship of each input pair. This preliminary stage could classify the pair as same-view, near cross-view, far cross-view, or potentially non-overlapping, and then select the most appropriate synchronization strategy. Based on the results obtained here, a promising direction is a coarse-to-fine ensemble in which full-frame semantic features and DTW first generate a restricted interval of plausible offsets, while significant object tracklets are subsequently used to validate and refine the final estimate. Such a formulation would combine the cross-view robustness of global scene representations with the temporal precision and reduced hypothesis space of local object-centered features. Additional directions include detecting the absence of temporal overlap, extending pairwise synchronization to multiple cameras, evaluating the framework on independent events, and quantitatively relating MOT quality to synchronization performance.

7.2 Publications, Datasets, and Tools

Most of the results of this dissertation were consolidated through publications, dataset articles, and supporting tools derived from this research. This section summarizes the publications and research outputs derived from this dissertation. The contributions include journal articles, conference papers, public datasets, and open-source software developed throughout the research. Table 7.1 summarizes the publications and submissions resulting from this work. The venue classification and impact indicators correspond to the information available at¹ the time of submission.

Table 7.1: Publications and submissions derived from this dissertation.

Group	Title, reference, and contribution	Venue / Journal and status	Qualis / Impact
Computer vision application	Comparative Analysis of Deep Learning-Based Multi-Object Tracking Approaches Applied to Sports User-Generated Videos [Alencar et al. 2025]. Contribution: Provides a comparative analysis of deep learning-based MOT approaches applied to sports UGVs, supporting the selection and design of the tracking stage adopted in this dissertation.	<i>International Conference on Computer Vision Theory and Applications (VISAPP)</i> , 2025. Published and presented in Porto, Portugal.	Qualis A2
Computer vision application	MOT-UGV: A Deep Learning-Based MOT Approach Applied to Sports UGVs [Alencar and de Freitas 2025]. Contribution: Presents MOT-UGV, the adapted approach for detecting and tracking relevant objects in sports UGVs, which constitutes Stage I of Track-UGV-Sync.	<i>International Conference on Soft Computing and Machine Intelligence (ISCMI)</i> , 2025. Published and presented in Rio de Janeiro, Brazil.	IF 3.9 (2024) CiteScore 12.1 Scopus Q1
Computer vision application	Track-UGV-SYNC: A Framework for Synchronizing Sports UGVs Using MOT Features [Alencar and de Freitas 2026a]. Contribution: Introduces the first version of Track-UGV-Sync, using visual features extracted by MOT to estimate the temporal offset between pairs of sports UGVs.	<i>International Conference on Computer Vision Theory and Applications (VISAPP)</i> , 2026. Published and presented remotely.	Qualis A2
Dataset	Exploring Multi-Camera Views from User-Generated Sports Videos [Pessoa et al. 2024]. Contribution: Investigates the characteristics of multi-camera and multi-view sports UGVs and supports the construction and organization of the datasets used throughout this dissertation.	<i>Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)</i> , 2024. Published and presented in Belém, Pará, Brazil.	Qualis B2
Dataset	MUVY: A Multi-View User-Generated Sports Video Dataset [Pessoa et al. 2026a]. Contribution: Describes MUVY, a collection of multi-view sports UGVs that provided the basis for the manually annotated synchronization evaluation pairs used in this dissertation.	<i>Data in Brief</i> , 2026. Published in July 2026. Associated dataset DOI: https://doi.org/10.5281/zenodo.13883315 .	Qualis A3
Dataset	MUVS: A Dataset of Multi-View User-Generated Sports Videos and Sensor Data. Contribution: Describes MUVS, which combines multi-view sports UGVs with smartphone sensor data and provides timestamp-based temporal references used in the construction and validation of UGV-Sync.	Submitted in June 2026 to <i>Data in Brief</i> . Associated dataset DOI: https://doi.org/10.5281/zenodo.20708683 .	Qualis A3
Dataset	UVY: A Dataset of Sport UGVs from YouTube for Multi-Object Tracking Evaluation. Contribution: Describes UVY, a sports UGV dataset designed to support the development and evaluation of multi-object tracking approaches under uncontrolled recording conditions.	Submitted in July 2026 to <i>Data in Brief</i> . Associated dataset DOI: https://doi.org/10.5281/zenodo.21303900 .	Qualis A3

¹<https://sucupira.capes.gov.br/avaliacao-quadrienal/qualis-eventos>

7.2.1 UGV-Sync Studio: Open-source software

As a practical contribution, this work also introduces the **UGV-sync Studio**, a visual inspection and annotation tool created to support the analysis of video synchronization results. The tool allows video pairs to be displayed side by side, together with their estimated or annotated offset, making it possible to visually inspect the behavior of the proposed framework pipeline (Figure 7.1). In addition, it supports the validation and refinement of ground-truth offsets through frame-by-frame inspection, helping to ensure the reliability of the annotated synchronization benchmark used in the experiments.

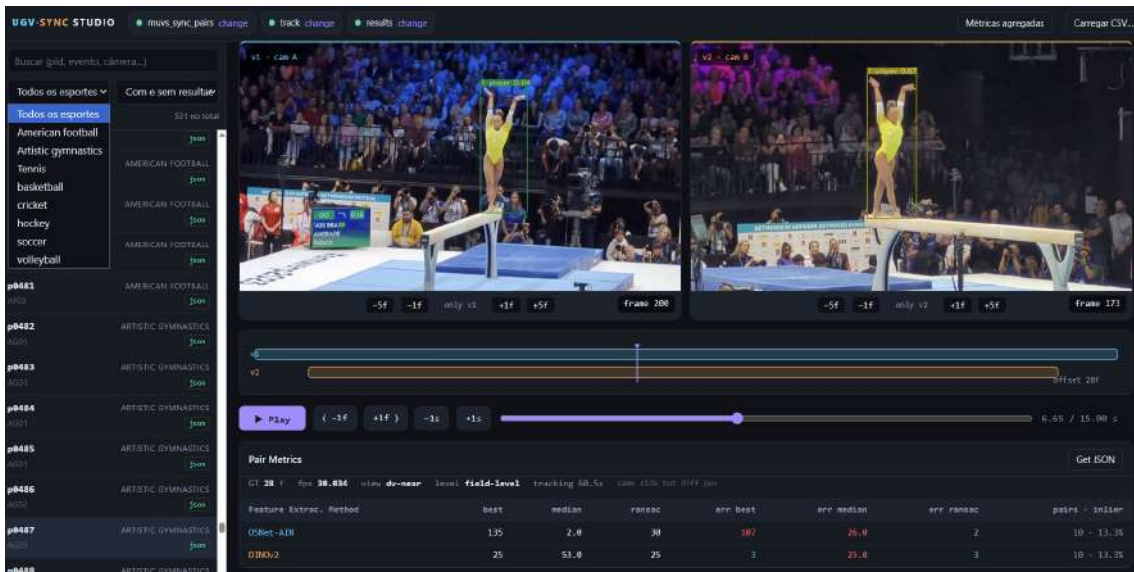


Figure 7.1: UGV-Sync Studio interface for side-by-side inspection across multi-sports scenarios, offset refinement, and comparison of sync metrics.

The source code, experimental configurations, and public datasets associated with this dissertation will be available through the <https://github.com/eltonalencar90/track-ugv-sync.git>.

References

- [Aggarwal 2018] Aggarwal, C. C. (2018). Neural networks and deep learning, volume 10. Springer International Publishing AG, part of Springer Nature 2018.
- [Alencar and de Freitas 2025] Alencar, E. and de Freitas, R. (2025). MOT-UGV: A deep learning-based multi-object tracking approach applied to sports user-generated videos. In 12th Proc. Int. Conf. on Soft Computing Machine Intelligence (ISCMI), pages 454–458.
- [Alencar and de Freitas 2026a] Alencar, E. and de Freitas, R. (2026a). Track-UGV-SYNC: A framework for synchronizing sports user-generated videos using multi-object tracking features. In 21th Proc. Int. Conf. on Computer Vision Theory and Applications - Volume 3: VISAPP, pages 484–491. INSTICC, SciTePress.
- [Alencar and de Freitas 2026b] Alencar, E. and de Freitas, R. (2026b). Uvy: a dataset of sport user-generated videos from youtube for multi-object tracking evaluation.
- [Alencar et al. 2022] Alencar, E. et al. (2022). Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações.
- [Alencar et al. 2025] Alencar, E., Pessoa, L., Costa, F., Souza, G., and de Freitas, R. (2025). Comparative analysis of deep learning-based multi-object tracking approaches applied to sports user-generated videos. In 20th Proc. Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: VISAPP, pages 691–698. INSTICC, SciTePress.

- [Amosa et al. 2023] Amosa, T. I., Sebastian, P., et al. (2023). Multi-camera multi-object tracking: a review of current trends and future advances. Neurocomputing, 552:126558.
- [Athar et al. 2023] Athar, A., Luiten, J., et al. (2023). Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In Proc. Int. Winter Conf. on Applications of Computer Vision (IEEE/CVF), pages 1674–1683.
- [Bailer et al. 2015] Bailer, W., Pike, C., et al. (2015). Multi-sensor concert recording dataset including professional and user-generated content. In 6th ACM Proc. Int. Conf of multimedia systems, pages 201–206.
- [Bernardin and Stiefelhagen 2008] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing, 2008:1–10.
- [Bewley et al. 2016] Bewley, A., Ge, Z., et al. (2016). Simple online and realtime tracking. In IEEE Proc. Int. Conference on Image Processing (ICIP), pages 3464–3468. IEEE.
- [Boizard et al. 2023] Boizard, N., El Haddad, K., et al. (2023). Deep learning-based stereo camera multi-video synchronization. In Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE.
- [Brito et al. 2011] Brito, D. N., Pádua, F. L., et al. (2011). Temporal synchronization of non-overlapping videos using known object motion. Pattern recognition letters, 32(1):38–46.
- [Broström 2022] Broström, M. (2022). Real-time multi-object tracking using yolov7 and strongsort with osnet. https://github.com/mikel-brostrom/Yolov7_StrongSORT_OSNet.
- [Caron et al. 2021] Caron, M., Touvron, H., et al. (2021). Emerging properties in self-supervised vision transformers. In Proc. Int. Conf. on computer vision (IEEE/CVF), pages 9650–9660.

- [Cheng et al. 2024] Cheng, T., Song, L., et al. (2024). YOLO-World: Real-time open-vocabulary object detection. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF - CVPR), pages 16901–16911.
- [Cherdchusakulchai et al. 2024] Cherdchusakulchai, R., Phimsiri, S., et al. (2024). Online multi-camera people tracking with spatial-temporal mechanism and anchor-feature hierarchical clustering. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 7198–7207.
- [Cioppa et al. 2022] Cioppa, A., Giancola, S., et al. (2022). Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 3491–3502.
- [Corporation 2023] Corporation, C. (2023). Computer vision annotation tool (cvat).
- [Cricri et al. 2013] Cricri, F., Roininen, M., et al. (2013). Multi-sensor fusion for sport genre classification of user generated mobile videos. In IEEE Int. Conf. on Multimedia and Expo (ICME), pages 1–6. IEEE.
- [Dendorfer et al. 2020] Dendorfer, P., Rezatofighi, H., et al. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003.
- [Dosovitskiy 2020] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- [Du et al. 2023] Du, Y., Zhao, Z., et al. (2023). Strongsort: Make deepsort great again. IEEE Transactions on Multimedia.
- [Fan and Zhao 2024] Fan, H. and Zhao, T. a. (2024). Gmt: A robust global association model for multi-target multi-camera tracking. arXiv preprint arXiv:2407.01007.
- [Fischler and Bolles 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analy-

- sis and automated cartography. Communications of the ACM, 24(6):381–395.
- [Gaddam et al. 2018] Gaddam, V. R., Langseth, R., et al. (2018). Camera synchronization for panoramic videos. MediaSync: handbook on multimedia synchronization, pages 565–592.
- [Girshick et al. 2014] Girshick, R., Donahue, J., et al. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 580–587.
- [Guggenberger 2023] Guggenberger, M. (2023). Multimodal Alignment of Videos. Doctoral dissertation, Alpen-Adria-Universität Klagenfurt, Klagenfurt am Wörthersee. Toward Multimodal Synchronization of User-Generated Event Recordings.
- [Guggenberger et al. 2015] Guggenberger, M., Lux, M., and Böszörményi, L. (2015). A synchronization ground truth for the jiku mobile video dataset. In 21st Proc. Int. Conf. on MultiMedia Modeling (MMM), pages 87–98. Springer.
- [Guo et al. 2026] Guo, W., Chen, Z., et al. (2026). Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. Advances in Neural Information Processing Systems, 38:124389–124422.
- [Held et al. 2023] Held, J., Cioppa, A., et al. (2023). Vars: Video assistant referee system for automated soccer decision making from multiple views. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 5085–5096.
- [Hosain et al. 2024] Hosain, M. T., Zaman, A., et al. (2024). Synchronizing object detection: Applications, advancements and existing challenges. IEEE Access.

- [Huang et al. 2023] Huang, H.-W., Yang, C.-Y., et al. (2023). Observation centric and central distance recovery for athlete tracking. In Proc. Winter Conf. on Applications of Computer Vision (IEEE/CVF), pages 454–460.
- [Huang et al. 2024] Huang, H.-W., Yang, C.-Y., et al. (2024). Iterative scale-up expansion and deep features association for multi-object tracking in sports. In Proc. eedings of Winter Conf. on Applications of Computer Vision (IEEE/CVF), pages 163–172.
- [Hussain 2024] Hussain, M. (2024). YOLOv1 to v8: Unveiling each variant a comprehensive review of yolo. IEEE Access, 12:42816–42833.
- [Jocher et al. 2023] Jocher, G., Qiu, J., and Chaurasia, A. (2023). Ultralytics yolo v8. <https://ultralytics.com>.
- [Kalman 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.
- [Krizhevsky et al. 2012] Krizhevsky, A. et al. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
- [Kuhn 1955] Kuhn, H. W. (1955). The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97.
- [Lewis et al. 2020] Lewis, P., Perez, E., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33:9459–9474.
- [Li et al. 2024] Li, S., Cao, Y., and Xie, X. (2024). A review of detection-related multiple object tracking in recent times. In 26th Int. Conf. on Advanced Communications Technology (ICACT), pages 122–130. IEEE.
- [Li et al. 2021] Li, Y., Chen, L., et al. (2021). Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In IEEE/CVF Proc. Int. Conf. on Computer Vision, pages 13536–13545.

- [Lin et al. 2014] Lin, T.-Y., Maire, M., et al. (2014). Microsoft coco: Common objects in context. In 13th Proc. European Conf. on Computer Vision (ECCV), pages 740–755. Springer.
- [Liu et al. 2024] Liu, Y., Ai, H., et al. (2024). Advancing video synchronization with fractional frame analysis: Introducing a novel dataset and model. In Proc. Int. Conf. on Artificial Intelligence (AAAI), volume 38, pages 3828–3836.
- [Lowe 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60:91–110.
- [Luiten and Hoffhues 2020] Luiten, J. and Hoffhues, A. (2020). Trackeval. <https://github.com/JonathonLuiten/TrackEval>.
- [Luiten et al. 2021] Luiten, J., Osep, A., et al. (2021). Hota: A higher order metric for evaluating multi-object tracking. International journal of computer vision, 129:548–578.
- [Luo et al. 2021] Luo, W., Xing, J., et al. (2021). Multiple object tracking: A literature review. Artificial intelligence, 293:103448.
- [Mahendran and Vedaldi 2015] Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 5188–5196.
- [McInnes et al. 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.
- [Mei et al. 2023] Mei, L., He, Y., et al. (2023). Learning domain-adaptive landmark detection-based self-supervised video synchronization for remote sensing panorama. Remote Sensing, 15(4):953.
- [Meinhardt et al. 2022] Meinhardt, T., Kirillov, A., et al. (2022). Trackformer: Multi-object tracking with transformers. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 8844–8854.

- [Meyer et al. 2008] Meyer, B., Stich, T., et al. (2008). Subframe temporal alignment of non-stationary cameras. In BMVC, pages 1–10.
- [Milan et al. 2016] Milan, A., Leal-Taixé, L., and other (2016). MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.
- [Naab and Sehl 2017] Naab, T. K. and Sehl, A. (2017). Studies of user-generated content: A systematic review. Journalism, 18(10):1256–1273.
- [Nasseri et al. 2021] Nasseri, M. H., Moradi, H., et al. (2021). Simple on-line and real-time tracking with occlusion handling. arXiv preprint arXiv:2103.04147.
- [Nazare et al. 2018] Nazare, A. C., Costa, F., and Schwartz, W. R. (2018). Content-based multi-camera video alignment using accelerometer data. In 15th IEEE Proc. Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE.
- [Nguyen et al. 2024] Nguyen, P., Quach, K. G., et al. (2024). Type-to-track: Retrieve any object via prompt-based tracking. Advances in Neural Information Processing Systems, 36.
- [Olagoke et al. 2020] Olagoke, A. S., Ibrahim, H., and Teoh, S. S. (2020). Literature survey on multi-camera system and its application. IEEE Access, 8:172892–172922.
- [Oquab et al. 2023] Oquab, M., Darcet, T., et al. (2023). Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193.
- [Peng et al. 2020] Peng, J., Wang, T., et al. (2020). TPM: Multiple object tracking with tracklet-plane matching. Pattern Recognition, 107:107480.
- [Perera et al. 2020] Perera, A. G., Law, Y. W., et al. (2020). A multiviewpoint outdoor dataset for human action recognition. IEEE Transactions on Human-Machine Systems, 50(5):405–413.
- [Pessoa et al. 2026a] Pessoa, L., Alencar, E., and de Freitas, R. (2026a). Muvy: A multi-view user-generated sports video dataset. Data in Brief, 67:113003.

- [Pessoa et al. 2024] Pessoa, L., Alencar, E., et al. (2024). Exploring multi-camera views from user-generated sports videos. In KDMILE, pages 105–112, Porto Alegre, RS, Brasil. SBC.
- [Pessoa et al. 2026b] Pessoa, L., Alencar, E., and Rosiane, F. (2026b). Muvs: a dataset of multi-view user-generated sports video and sensor data.
- [Rangasamy et al. 2020] Rangasamy, K., As’ari, M. A., et al. (2020). Deep learning in sport video analysis: a review. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(4):1926–1933.
- [Redmon 2016] Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF).
- [Ren et al. 2016] Ren, S., He, K., et al. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6):1137–1149.
- [Ristani and Solera 2016] Ristani, E. and Solera, F. a. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In European conference on computer vision, pages 17–35. Springer.
- [Roboflow-Team 2024] Roboflow-Team (2024). Zero-shot object detection with yolo-world. <https://supervision.roboflow.com/develop/notebooks/zero-shot-object-detection-with-yolo-world/>.
- [Russell and Norvig 2020] Russell, S. and Norvig, P. (2020). Artificial Intelligence: A Modern Approach. 4th edition.
- [Sakoe and Chiba 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1):43–49.
- [Salvador and Chan 2007] Salvador, S. and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis, 11(5):561–580.

- [Scott et al. 2022] Scott, A., Uchida, I., et al. (2022). SoccerTrack: A dataset and tracking algorithm for soccer with fish-eye and drone videos. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 3569–3579.
- [Segundo et al. 2017] Segundo, R. M. C., de Amorim, M. N., et al. (2017). Crowdsync: User generated videos synchronization using crowdsourcing. In Proc. Int. Conf. on Systems, Signals and Image Processing (IWSSIP), pages 1–5. IEEE.
- [Shin and Molybog 2025] Shin, Y. and Molybog, I. (2025). Beyond audio and pose: A general-purpose framework for video synchronization. arXiv preprint arXiv:2506.15937.
- [Tanaka et al. 2022] Tanaka, T. et al. (2022). Updating a dataset of labelled objects on raw video sequences with unique object ids. Data in Brief, 41:107892.
- [Tang et al. 2023] Tang, Y., Bi, J., et al. (2023). Video understanding with large language models: A survey. arXiv preprint arXiv:2312.17432.
- [Varior et al. 2016] Varior, R. R., Haloi, M., and Wang, G. (2016). Gated siamese convolutional neural network architecture for human re-identification. In 14th Proc. European Conf. on Computer Vision (ECCV), pages 791–808. Springer.
- [Vaswani et al. 2018] Vaswani, A., Shazeer, N., et al. (2018). Attention is all you need. Advances in neural information processing systems, 30.
- [Voigtlaender et al. 2019] Voigtlaender, P., Krause, M., et al. (2019). Mots: Multi-object tracking and segmentation. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 7942–7951.
- [Wang et al. 2024] Wang, J., Chen, D., et al. (2024). Omnivid: A generative framework for universal video understanding. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 18209–18220.

- [Wang et al. 2014] Wang, J., Song, Y., et al. (2014). Learning fine-grained image similarity with deep ranking. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 1386–1393.
- [Wang et al. 2016] Wang, X., Shi, J., et al. (2016). Motion-based temporal alignment of independently moving cameras. IEEE Transactions on Circuits and Systems for Video Technology, 27(11):2344–2354.
- [Wang et al. 2023] Wang, Y., Huang, Q., et al. (2023). Video stabilization: A comprehensive survey. Neurocomputing, 516:205–230.
- [Whitehead et al. 2005] Whitehead, A. et al. (2005). Temporal synchronization of video sequences in theory and in practice. In 7th IEEE Workshops on Applications of Computer Vision (WACV/MOTION), volume 2, pages 132–137. IEEE.
- [Wojke et al. 2017] Wojke, N. et al. (2017). Simple online and realtime tracking with a deep association metric. In Proc. Int. Conf. on Image Processing (ICIP), pages 3645–3649. IEEE.
- [Wu et al. 2015] Wu, Y., Mei, T., et al. (2015). Movieup: Automatic mobile video mashup. IEEE Transactions on Circuits and Systems for Video Technology, 25(12):1941–1954.
- [Yin et al. 2022] Yin, L., Han, R., Feng, W., and Wang, S. (2022). Self-supervised human pose based multi-camera video synchronization. In 30th ACM Proc. Int. Conf. on Multimedia, pages 1739–1748.
- [Yu et al. 2016] Yu, F., Li, W., et al. (2016). POI: Multiple object tracking with high performance detection and appearance feature. In Workshops Proceedings Computer Vision (ECCV), pages 36–42. Springer.
- [Zhang et al. 2015] Zhang, F.-L., Wang, J., et al. (2015). Simultaneous camera path optimization and distraction removal for improving amateur video. IEEE Transactions on Image Processing, 24(12):5982–5994.
- [Zhang and Chan 2022] Zhang, Q. and Chan, A. B. (2022). Single-frame-based deep view synchronization for unsynchronized multicamera surveillance. IEEE

Transactions on Neural Networks and Learning Systems, 34(12):10653–10667.

[Zhang et al. 2023] Zhang, Y., Wang, T., and Zhang, X. (2023). Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In Proc. Int. Conf. on Computer Vision and Pattern Recognition (IEEE/CVF), pages 22056–22065.

[Zhao et al. 2023] Zhao, Z., Chai, W., et al. (2023). A survey of deep learning in sports applications: Perception, comprehension, and decision. arXiv preprint arXiv:2307.03353.

[Zhou et al. 2022] Zhou, K., Yang, Y., et al. (2022). Learning generalisable omni-scale representations for person re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9):5056–5069.