MASTERARBEIT | MASTER'S THESIS

Titel | Title

Density based subspace-clustering for object tracking

verfasst von | submitted by

Maximilian Bierl BSc BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Inform.Univ. Dr. Claudia Plant
Mitbetreut von | Co-Supervisor:	Dr. Anna Beer BSc MSc

Acknowledgements

I would like to express my deepest gratitude to my father for his constant support, encouragement, and understanding throughout this entire journey. His belief in me and his willingness to stand by my side at all times have been invaluable.

I would also like to express my sincere thanks to my co-supervisor, Dr. Anna Beer, for her valuable support, guidance, and encouragement throughout the entire process of this thesis. Her insights and willingness to engage in every step of the thought process with me were essential to the development of this work.

Finally, I would like to thank Rhea Tebbich for kindly taking the time to proofread my thesis and provide helpful feedback.

Abstract

This work explores how Wasserstein distance can be effectively used for unsupervised clustering, sequencing, and motion analysis in unordered image sequences. We investigate the integration of Wasserstein distance into density-connectivity-based clustering and assess its performance across different preprocessing strategies, color spaces, and fusion techniques. A full pipeline is proposed that combines clustering with temporal ordering and motion extraction based on optimal transport plans. We evaluate the framework on synthetic and real-world datasets, analyzing its strengths and limitations in varying conditions. Results indicate that Wasserstein-based approaches can uncover scene structure and object movement without relying on supervision or semantic features, though performance depends heavily on data characteristics and parameter choices.

Kurzfassung

In dieser Arbeit wird untersucht, wie sich die Wasserstein-Distanz für Verfahren zur automatischen Gruppierung, zeitlichen Anordnung und Bewegungsanalyse in ungeordneten Bildsequenzen einsetzen lässt. Dabei wird ihre Integration in dichte- und verbindungsbasierte Clustering-Methoden betrachtet und die Leistung unter verschiedenen Vorverarbeitungsstrategien, Farbräumen und Fusionsansätzen bewertet. Darüber hinaus wird eine vollständige Verarbeitungspipeline entwickelt, die Clustering, temporale Sortierung und die Extraktion von Bewegungsinformationen auf Basis optimaler Transportpläne kombiniert. Das vorgeschlagene Framework wird anhand von synthetischen und realen Datensätzen evaluiert, um seine Stärken und Grenzen unter unterschiedlichen Bedingungen aufzuzeigen. Die Ergebnisse zeigen, dass wassersteinbasierte Ansätze in der Lage sind, Szenenstrukturen und Objektbewegungen auch ohne manuelle Annotationen oder semantische Informationen zuverlässig zu erkennen. Die erzielte Leistung hängt jedoch stark von den Eigenschaften der Daten sowie den gewählten Parameterkonfigurationen ab.

Contents

List of Tables

List of Figures

1. Introduction

1.1. Motivation

Understanding how objects move across time is a foundational challenge in computer vision, underpinning applications such as surveillance, autonomous driving, sports analysis, and behavioral monitoring. Traditional object tracking methods typically rely on supervised learning, which requires extensive annotated datasets and often depends on domain-specific object detectors or appearance models. These methods are consequently sensitive to variations in visual domains, changes in object appearance, lighting conditions, and occlusions. This dependency also hampers their ability to generalize to new or unlabeled scenarios.

In contrast, human perception often organizes visual scenes based on structural continuity and spatial transformations, without needing explicit knowledge of object categories. Drawing inspiration from this perceptual ability, this thesis explores a fully unsupervised and interpretable approach to motion understanding—focusing not on the identity of objects but on how visual mass shifts and evolves over time.

1.2. Problem Statement

The central objective of this work is to analyze sequences of raw images without any external supervision. The aim is threefold: first, to segment the sequence into coherent scenes based on structural similarity; second, to establish a logical or temporal ordering within each identified scene; and third, to discover and localize regions where motion consistently occurs, all without relying on annotations, labels, or pretrained features.

Achieving this entails developing methods that can meaningfully compare image content and structure, reason about spatial relationships and local densities, and infer motion patterns purely from image sequences.

1.3. Proposed Approach

To address these goals, this thesis introduces a novel framework grounded in optimal transport theory and density-based clustering. The pipeline is designed to analyze image sequences from a structural standpoint, incorporating the following key components:

First, each image is interpreted as a discrete distribution of visual mass, evaluated per color channel. To compare image pairs, the Wasserstein distance (also known as Earth Mover's Distance) is employed. This metric accounts for both pixel intensities and their spatial arrangement, enabling a robust comparison of image structure.

Next, the computed pairwise distances form the basis of a mutual reachability graph, from which density-connectivity distances are derived. A k-center clustering algorithm is then applied, segmenting the image sequence into semantically coherent scenes.

Within each scene, a greedy ordering algorithm leverages the Wasserstein distances to approximate the temporal or visual progression of images. This approach avoids the computational complexity of solving exact sequencing problems, while still preserving meaningful continuity.

Finally, the transport plans generated during the Wasserstein evaluations are analyzed as mass flow vectors. These vectors are clustered to identify consistent motion patterns, allowing for the localization of motion subspaces. This unsupervised analysis enables object motion tracking without any reliance on category-specific cues or labeled data.

The entire framework is modular, interpretable, and fully unsupervised. It eschews the use of pretrained neural networks and handcrafted features, relying instead on geometric, probabilistic, and density-aware reasoning.

1.4. Contributions

This thesis advances the field of unsupervised motion analysis through a set of interconnected contributions. First, it introduces a novel Wasserstein-based method for comparing image frames, treating them as spatial distributions. This approach is applicable to both standard RGB and perceptually uniform color spaces, ensuring broad relevance and robustness. Building on this foundation, the thesis presents a density-based clustering technique that leverages *DC-Distance* and k-center algorithms to segment image sequences into coherent scenes without any form of supervision.

A further contribution lies in the identification of motion subspaces. By clustering transport vectors derived from optimal transport plans, the framework is able to localize motion in a fully interpretable manner, without requiring labels or predefined object categories. These methods are brought together in a unified, modular pipeline that performs scene discovery, temporal sequencing, and motion analysis. Crucially, this framework operates entirely without annotated data, object detectors, or pretrained models, highlighting its potential for generalization across diverse and previously unseen visual domains.

1.5. Structure of this Thesis

The remainder of this thesis is structured to guide the reader through the theoretical foundations, methodological contributions, and experimental validation of the proposed framework. Chapter 2 lays the groundwork by introducing the necessary theoretical background, including optimal transport theory, the Wasserstein distance, and techniques for density-based clustering. Chapter 3 builds upon this foundation to describe the full proposed approach in detail, covering image representation, distance computation, clustering mechanisms, sequence ordering strategies, and motion analysis components.

Chapter 4 presents empirical results obtained from both synthetic and real-world image sequences. These evaluations include both qualitative visualizations and quantitative performance metrics, demonstrating the effectiveness of the approach. Chapter 5 offers a critical discussion of the method's limitations and identifies failure cases, while also proposing possible improvements and future enhancements. Finally, Chapter 6 concludes the thesis by summarizing the findings.

2. Related Work

Density-based clustering methods, such as DBSCAN [EKS$^+$96], have been widely used for discovering arbitrarily shaped clusters and handling noise in static datasets. However, their application to dynamic data, like time series and video, introduces challenges due to evolving cluster structures over time. These limitations have motivated research into more adaptive approaches, including subspace clustering and novel distance measures designed to capture temporal and structural variation. This section reviews prior work in density-based clustering, its extensions to dynamic settings, and recent developments in subspace methods relevant to object tracking.

2.1. Density-Based Clustering Techniques

Density-based clustering algorithms leverage the concept of data density to identify clusters with diverse shapes, sizes, and densities . These methods are robust and widely used in applications like astronomy, earth sciences, and multimedia [BM21].

2.1.1. Characteristics of Density-based clustering algorithms

Density-based clustering algorithms are characterized by their ability to group dense regions in data as clusters, effectively separating them from low-density noise. They excel in identifying arbitrarily shaped clusters and offer flexibility for datasets with varying density levels [BM21]. This ability is especially advantageous for spatial databases, where traditional methods may overlook compact clusters in remote areas [BM21].

Key features of Density-based clustering algorithms include:

- **Granularity Exploration:** Density-based clustering algorithms explore the data space at varying levels of granularity, reconstructing the overall shape of the data distribution (Original work cited in [BM21]- p.3 [Red18]).

- **Noise Filtering:** These algorithms incorporate noise-handling mechanisms, ensuring cleaner clustering outputs (see Figure 2.1) [EKS$^+$96].

- **Arbitrary Shape Detection:** Density-based clustering algorithms can detect clusters of arbitrary shapes, accommodating non-globular and complex structures in data (see Figure 2.2) [ESK03].

Popular examples of Density-based clustering algorithms are:

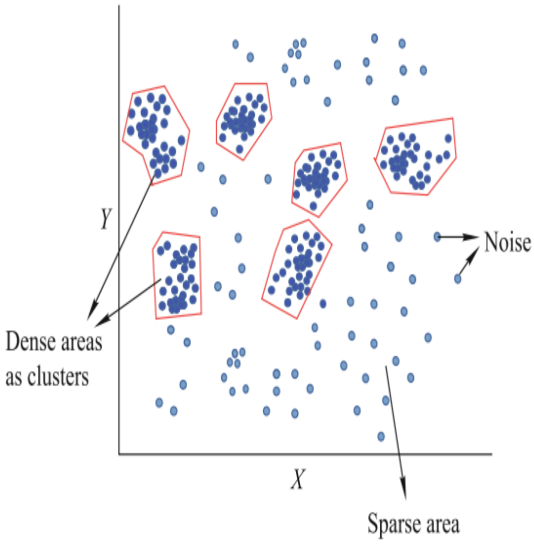

Figure 2.1.: Dense regions resemble clusters. Noises are filtered out. [BM21]

- **DBSCAN:** The foundational algorithm, introduced in 1996, identifies clusters based on a neighborhood radius (ϵ) and a minimum number of points (MinPts). It performs well with spatial data but struggles with varying densities and high-dimensional spaces [EKS+96].
- **HDBSCAN:** This extension of DBSCAN builds a hierarchical clustering tree based on varying density levels. By optimizing cluster stability, it provides more flexibility and eliminates the need to define a fixed ϵ parameter. HDBSCAN also distinguishes between core, noise, and hierarchical structures in the data, making it more robust for datasets with varying densities [CMS13]

2.1.2. Advanced Developments: The Density-Connectivity Distance (DC-Distance)

Recent advancements in density-based clustering introduced the Density-Connectivity Distance , which formalizes the clustering objectives of DBSCAN and unifies it with other clustering paradigms like k-Center and Spectral Clustering. The DC-Distance combines minimax path distances with density constraints, ensuring that only dense paths contribute to cluster formation [BDH+23]. Key-Features of DC-Distance include:

- **Unified Clustering Framework:** DC-Distance demonstrates the equivalence of clustering outcomes across different algorithms such as DBSCAN [EKS+96], k-Center, and Spectral Clustering. This is achieved by framing all these algorithms within the same metric space, defined by density-connectivity constraints.

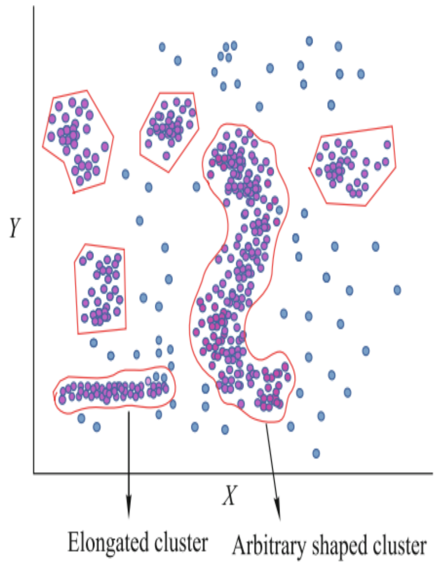

Figure 2.2.: Depiction of clusterrecognition of clusters with varying shapes[BM21]

Such equivalence bridges gaps between heuristic and optimization-based clustering paradigm[BDH+23].

- **Formal Optimization Objective:** Unlike traditional DBSCAN [EKS+96], which relies on heuristic parameter settings like ϵ and MinPts, DC-Distance introduces a clear objective function. It minimizes intra-cluster distances while enforcing density-connectivity, thus formalizing DBSCAN's [EKS+96] clustering objectives into a measurable optimization problem [BDH+23].

- **Ultrametric Properties:** It allows for the simultaneous calculation of DBSCAN [EKS+96] clusterings across all ϵ values, streamlining the tuning process [BDH+23].

Furthermore the usage of DC-Distance has several other beneficial computational properties:

- **Parameter Simplification:** By enabling the calculation of DBSCAN [EKS+96] results across all ϵ values simultaneously, DC-Distance significantly reduces the parameter tuning burden. The metric offers an efficient tree-based representation for exploring clustering solutions over a range of parameter values [BDH+23].

- **Scalability:** The ultrametric properties of DC-Distance limit the number of unique pairwise distances to $O(n)$, as opposed to the quadratic complexity of traditional pairwise metrics. This makes the method computationally feasible for large-scale datasets[BDH+23].

- **Noise Resilience:** The incorporation of density constraints mitigates the single-link effect, which can result in spurious cluster connections in noisy data. By focusing exclusively on dense paths, DC-Distance ensures that clusters reflect true data structures [BDH+23].

2.2. Clustering in Time Series Data

Time series clustering is a pivotal method in data mining, designed to group temporally dependent and often high-dimensional datasets. This clustering approach enables the identification of patterns in domains such as finance, biology and multimedia, where time series data are prevalent [ASW15].

Time series clustering frameworks typically involve four key components: dimensionality reduction, distance measurement, clustering algorithms, and evaluation methods [ASW15]. Dimensionality reduction methods, address the challenges of high dimensionality by transforming raw time series into more compact representations while distance measures such as Euclidean distance and Dynamic Time Warping are essential for quantifying similarity, especially in datasets with varying size and temporal shifts [ASW15].

2.2.1. Density-Based Methods: Characteristics and Challenges

Density-based clustering methods, including Density Peaks, are characterized by their ability to detect arbitrarily shaped clusters without requiring a predefined number of clusters. This makes them particularly advantageous for analyzing complex datasets with irregular structures and noise. Unlike partitional methods, density-based approaches are robust to noise and outliers due to their reliance on local density thresholds, which serve as the basis for cluster formation [ASW15].

However, as noted by Javed et al., density-based methods face significant computational challenges when applied to high-dimensional and large-scale time series datasets. The quadratic complexity of these algorithms often limits their scalability and practicality in handling large datasets. [ASW15].For instance, the promising Density Peaks algorithm [DDXS19]with Dynamic Time Warping [CKHP02] as the distance metric showed high computational demands, requiring substantial optimization to ensure feasibility. Even elaborated pruning strategies, clustering large datasets can take days to process [JLR20].

On the other hand, the application of Density Peaks with Dynamic Time Warping, which was shown to be effective in detecting temporal misalignments in datasets with irregular structures, though it lagged behind partitional methods in Adjusted Rand Index (ARI) scores across the benchmark [JLR20].

Chandrakala and Chandra also explored kernel-based density clustering algorithms tailored for multivariate time series of varying lengths. Their heuristic-driven parameter initialization significantly enhanced clustering quality, illustrating the adaptability of density-based methods in this domain [CS08].

2.2.2. Broader Landscape of Time Series Clustering

Time series clustering also includes hierarchical, partitioning, model-based, and hybrid approaches. Hierarchical methods excel in visualizing clusters but struggle with scalability due to their quadratic computational complexity [ASW15]. Partitioning methods, such as k-means, are computationally efficient but require a predefined number of clusters, which can be challenging for real-world applications [NR07]. Nevertheless, because of their speed, k-Means and k-Medoids are used in many works [ASW15].

In a study by Javed et al. [JLR20], hierarchical clustering with Ward's linkage showed superior performance in average ARI scores compared to partitional and density-based methods . However, the overlap in performance between hierarchical and partitional methods suggests that their utility may be dataset-specific rather than universally superior [JLR20].

2.2.3. Future Directions

Future advancements in density-based time series clustering should might on integrating these methods with advanced dimensionality reduction techniques and hybridizing them with other clustering frameworks. For instance, combining density-based approaches with hierarchical methods could leverage their complementary strengths, enhancing the scalability and adaptiveness of clustering algorithms. As noted in the benchmark study by Javed et al.[JLR20], it was observed that clustering methods that show a wide range of Adjusted Rand Index (ARI) scores across datasets tend to create different types of clusters depending on the dataset. This means such methods can adapt to diverse data structures and reveal patterns that other methods might miss, making them useful for analyzing various kinds of time series data [JLR20].

By addressing computational scalability and integrating tailored dimensionality reduction strategies, density-based clustering can become a robust tool for analyzing high-dimensional, complex, and noisy time series data.

2.3. Subspace Clustering

Subspace clustering has become a cornerstone in handling high-dimensional data and as object tracking can be approached not only by directly comparing scenes, but also by analyzing the underlying structure of the data in feature space. Instead of matching entire frames, one can identify subspaces that correspond to individual objects or coherent motion patterns. In particular, Wasserstein distances and their associated transport plans can reveal how mass shifts between regions across frames, which in turn can be interpreted as object movement. Investigating subspace clustering methods therefore provides valuable insights into alternative ways of segmenting and associating objects in dynamic scenes, and serves as a conceptual foundation for the proposed Wasserstein-based approach.

2.3.1. Overview

According to Sim et al. [SGZC13], subspace clustering addresses two key challenges in high-dimensional data analysis:

- **Irrelevant Attributes:** High-dimensional datasets often contain irrelevant attributes that obscure meaningful clusters. Subspace clustering resolves this by dynamically identifying subsets of attributes relevant to each cluster. This issue was extensively discussed by Kriegel et al. [KKZ09], who analyzed the impact of dimensionality on clustering effectiveness.

- **Cluster Overlap:** Different clusters may exist in overlapping subsets of attributes. Subspace clustering accommodates this flexibility, enabling objects to belong to multiple clusters in different subspaces, as emphasized by Sim et al. [SGZC13] in their survey on enhanced subspace clustering.

2.3.2. Techniques

The survey by Kriegel et al. [KKZ12] describes three types of major techniques used in thelast years:

- **Axis-Parallel Subspace Clustering:** Techniques in this category, such as CLIQUE [AGGR98], rely on identifying dense regions of data within axis-parallel subspaces. These methods operate under the assumption that clusters are well-represented when irrelevant attributes are excluded, as supported by the statistical observations of Moise and Sander [MZK+09].

- **Arbitrarily Oriented Subspaces:** Algorithms like ORCLUS [AY00] extend beyond axis-parallel constraints by incorporating principal component analysis. The adoption of eigenvalue decompositions, as formalized by Jolliffe [Jol02], allows these methods to model clusters in hyperplanes defined by linear dependencies between attributes. Achtert et al. [ABK+06] also contributed significant insights into the quantitative modeling of correlation clusters, by developing methods to derive explicit mathematical models that represent clusters formed by objects exhibiting linear correlations in high-dimensional spaces

- **Soft Projected Clustering:** Approaches like the entropy-weighted k-means algorithm [JNH07] dynamically adjust the importance of attributes during clustering. This method, rooted in optimization techniques, emphasizes a balance between dimensionality reduction and clustering accuracy.

2.3.3. Density-Based Subspace Clustering

Among basic methods, density-based subspace clustering stands out for its ability to identify arbitrarily shaped clusters by focusing on dense regions within subspaces. Techniques such as SUBCLU and DiSH, developed by Kailing et al. [KKK04] and Achtert et al. [ABK+07], adapt DBSCAN's density connectivity principles for high-dimensional data. These methods could be particularly effective in tracking scenarios where object trajectories form non-linear patterns in feature spaces.[SGZC13]

2.3.4. Relevance to Object Tracking

The flexibility of subspace clustering makes it uniquely suited for object tracking. For example, axis-parallel methods can highlight static movement patterns by focusing on the most relevant features, while arbitrarily oriented approaches uncover complex motion trajectories influenced by multi-dimensional correlations. These capabilities are critical in domains where object behavior spans varying feature spaces and scales.[KKZ12]

For example Sparse Subspace Clustering (SSC)[EV13] has emerged as a powerful tool for analyzing motion trajectories in object tracking. By leveraging the self-expressiveness property, SSC clusters high-dimensional data, such as feature trajectories in videos, into low-dimensional subspaces corresponding to distinct motions. This is particularly effective in motion segmentation, where trajectories of rigidly moving objects are segmented based on their subspace affiliations [EV13].

Key applications include well known datasets, where SSC demonstrates its robustness against noise, outliers, and missing data, all common in real-world tracking scenarios. Additionally, SSC's adaptability to affine subspaces enhances its applicability to transformations like scaling and rotation, critical for modern tracking challenges.[EV13] Its ability to handle these complexities could make SSC a compelling choice for object tracking tasks in dynamic environments.

2.3.5. Future Directions

Challenges remain in defining optimal subspaces and addressing redundancy within clustering results.Assent et al. [AMG+10] for example, explores adaptive density thresholds to overcome biases introduced by fixed criteria, which could be particularly relevant for dynamic object tracking scenarios. However, most subspace clustering methods have so far been developed and evaluated in abstract or tabular data domains rather than applied to image data. In the context of object tracking, identifying subspaces can correspond to isolating groups of pixels or features that represent individual objects or coherent motion patterns. This connection has not yet been systematically explored. Bridging this gap could enable subspace clustering techniques to be adapted for visual data, opening new possibilities for unsupervised object discovery and motion segmentation.

2.4. The Role of Wasserstein Distances in Dynamic Clustering and Tracking

2.4.1. Optimal Transport and Wasserstein Distances

The mathematical foundations of optimal transport and Wasserstein distances trace back to the work of Kantorovitch, who redefined the transport problem as a linear programming challenge. This formulation established the modern framework for measuring the "effort" required to map one probability distribution onto another, providing a rigorous basis for various applications such as clustering and tracking (as cited in [BRPP15]). The conceptualization of Wasserstein distances as an optimization problem has profoundly influenced clustering paradigms by enabling more robust metrics for quantifying distributional differences.

Building on this foundation, Benamou and Brenier introduced a time-dependent, convex optimization approach for computing Wasserstein geodesics. This method offered a means to smoothly interpolate between probability measures, making it particularly relevant for applications requiring dynamic modeling of evolving distributions, such as object trajectories. This alignment of smooth transitions between measures is a core requirement for density subspace clustering, where evolving trajectories often exhibit dynamic subspace structures [BB00].

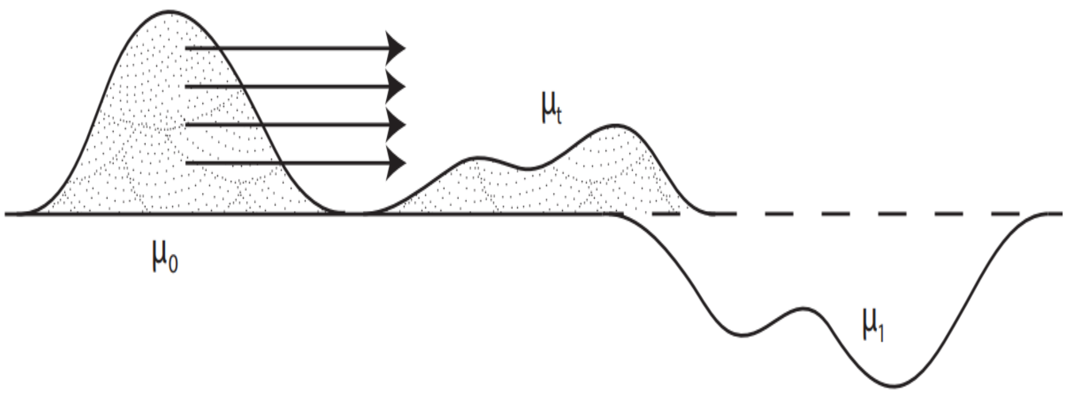

Figure 2.3.: The mass transportation problem involves the optimal transfer of a probability measure μ_0, represented as a pile of sand, to another probability measure μ_1, represented as a hole. At an intermediate time $t \in [0, 1]$, an interpolated probability measure μ_t is formed. [BRPP15]

2.4.2. Wasserstein Barycenters

The concept of Wasserstein barycenters, formulated by Agueh and Carlier, provides the theoretical foundation for aggregating probabilistic representations.[AC11] These barycenters offer existence and uniqueness properties, enabling their use in clustering tasks requiring the synthesis of multiple trajectories or distributional modes, such as in multi-object tracking scenarios. The aggregation of trajectory distributions into a central representation aligns naturally with density subspace clusterings goals.

To address computational challenges, Rabin et al. [RPDB12] developed an efficient method for approximating Wasserstein barycenters using one-dimensional projections.

This technique substantially reduces computational complexity while maintaining effectiveness, making it ideal for scalable clustering tasks. The ability to leverage these computational shortcuts is critical in dynamic clustering scenarios where data streams and trajectories necessitate real-time responsiveness [RPDB12].

2.4.3. Applications in Image and Motion Analysis

The practicality of Wasserstein distances has been demonstrated in various real-world contexts. For instance, Pitié et al. employed one-dimensional Wasserstein distances for histogram matching, laying the groundwork for trajectory alignment tasks in object tracking [PKD05]. These techniques provide a concrete example of how Wasserstein-based measures can bridge the gap between theoretical robustness and practical utility, particularly in tasks involving noisy or incomplete data.

2.4.4. Integration with Subspace Clustering

Dynamic Subspace Evolution

The interpolation of measures through Wasserstein geodesics [BB00] and barycenters [AC11] should directly support the modeling of dynamic subspaces. In object tracking, where object motion is often conceptualized as sequences of evolving subspaces in high-dimensional feature spaces, Wasserstein-based methods provide a powerful framework for identifying transitions and dynamically unifying clusters.

Dimensionality Reduction and Scalability

The one-dimensional projection method proposed by Rabin et al. [RPDB12] parallels subspace clustering techniques like PCA, where dimensionality reduction is used to identify relevant subspaces efficiently. This analogy underscores the scalability of Wasserstein-based clustering methods in handling high-dimensional data streams, a frequent challenge in object tracking applications.

Multiscale Sliced Wasserstein Distances (MS-SWD)

He et al. [HWW$^+$24] introduced the Multiscale Sliced Wasserstein Distance (MS-SWD) as a perceptual color difference measure for photographic images. The method combines multiscale analysis with non-local comparisons, showing strong performance in handling image misalignment and geometric transformations compared to traditional color difference metrics. These results suggest that Wasserstein-based distances can be adapted effectively to tasks where strict spatial correspondence cannot be assumed.

At the same time, the work is primarily situated within the domain of perceptual quality assessment, rather than clustering or tracking. Its evaluation is limited to image datasets and perceptual benchmarks, and the connection to broader distributional modeling tasks remains indirect. While the training-free design and robustness are appealing from a computational perspective, it is not yet clear how well the approach transfers to dynamic, high-dimensional data streams typical in subspace clustering or multi-object tracking.

Robustness in Real-Time Applications

Wasserstein distances are particularly robust to noise and incomplete data, as highlighted by Pitié et al. [PKD05]. This robustness complements density-based subspace clustering techniques, such as SUBCLU and DiSH, which are designed to handle data corrupted by occlusions or inconsistencies. By leveraging the inherent resilience of Wasserstein measures, object tracking can achieve greater reliability in diverse real-world scenarios.

2.5. Object Tracking

The field of object tracking has seen tremendous development. Chen et al. [CWZ+22] differentiate the approaches into generative, discriminative, collaborative, and deep learning-based methods.

2.5.1. Generative Trackers

Generative trackers focus on modeling the target's appearance to find its best match in subsequent frames. For instance, Ross et al.[RLLY08] introduced Incremental Visual Tracking, which utilizes subspace learning to adapt to variations in the target's appearance. Mei and Ling [ML09] proposed the L1 Tracker, leveraging sparse representations for robust tracking. These methods excel in adaptability but face challenges in handling rapid motion or complex scenes.[CWZ+22]

2.5.2. Discriminative Trackers

Discriminative trackers treat object tracking as a classification problem, separating the target from its background [FL19]. Babenko et al. [BYB10] introduced Multiple Instance Learning, employing positive and negative bags to train classifiers for better robustness. Henriques et al. [HCMB14] advanced Kernelized Correlation Filters, which accelerate training and tracking by exploiting Fourier transforms. These approaches seem to provide robust performance under occlusion and cluttered environments.

2.5.3. Collaborative Trackers

Collaborative trackers combine generative and discriminative methodologies for enhanced performance. Zhong et al. [ZLY12] proposed the Sparse Collaborative Model, integrating sparsity-based classifiers for robustness against environmental changes. Hong et al. [HCW+15] developed MUSTer (Multi-Store Tracker), which uses short-term and long-term memory schemes to improve tracking consistency.

2.5.4. Deep Learning-Based Trackers

Deep learning has revolutionized tracking with methods that leverage robust feature extraction and end-to-end learning frameworks. Bertinetto et al. [BVH+16] introduced SiamFC, a fully convolutional Siamese network that learns similarity metrics for tracking. Li et al. [LWW+19] extended this work with SiamRPN, incorporating a Region Proposal Network for precise bounding box predictions. Further advancements include DiMP by Bhat et al. [BDGT19] and ATOM by Danelljan et al. [DBKF19], both of which dynamically update the target model during tracking for improved accuracy and adaptability.

2.5.5. Density-Based Clustering in Object Tracking

Density-based clustering algorithms are already used in object tracking applications, especially in complex environments with varying densities and noise. As described before, these algorithms identify clusters by detecting regions of high point density, effectively distinguishing objects from background noise.

In multi-object tracking scenarios, density-based clustering facilitates the association of detection points across frames, enabling the formation of coherent object trajectories. For instance, the DBSCAN-Based Tracklet Association Annealer enhances multi-object tracking by robustly associating tracklets, even in noisy conditions. [KC21] Similarly, the MeanShift++ algorithm accelerates mode-seeking processes, improving segmentation and tracking efficiency [JJ21].

Moreover, density-based clustering is instrumental in processing data from point clouds, such as those obtained from LiDAR sensors. By clustering points that represent object surfaces, these algorithms assist in detecting and tracking objects in 3D space, which is crucial for applications like autonomous driving. [AC20]

In summary, density-based clustering methods are integral to object tracking research today, offering robust performance in complex and noisy environments by effectively grouping relevant data points and facilitating accurate object trajectory estimation. To the best of our knowledge, however, these methods have not been applied in the way proposed in this work, where Wasserstein-based distances are directly combined with density-connectivity clustering for scene-level comparison and object tracking.

2.6. Summary

Density-based subspace clustering has emerged as a promising approach in addressing the complexities of object tracking in dynamic and high-dimensional data environments. Its ability to detect clusters of arbitrary shapes and handle noise makes it well-suited for scenarios where traditional clustering methods often fail. However, the application of these methods to object tracking comes with significant challenges.

One of the key issues is the dynamic and evolving nature of object tracking data. The time-series-clusters change over time as objects move, interact, or disappear, necessitating algorithms that can adapt to these temporal variations [JLR20]. Traditional density-based methods that are more static seem to struggle to accommodate such dynamics without significant modification.

Another critical challenge lies in the scalability of these methods when applied to high-dimensional data. Object tracking often involves complex feature spaces, and the curse of dimensionality makes clustering computationally intensive and less effective. Subspace clustering methods, such as SUBCLU and DiSH, attempt to mitigate these issues but introduce their own set of challenges, including identifying relevant subspaces and managing overlapping clusters [KKZ12].

Noise and outliers further complicate the process. Real-world tracking data is often noisy due to environmental factors, sensor inconsistencies, or occlusions. Even though density- based approaches should be well equipped for that challange [BM21], these irregularities can fragment clusters or introduce errors in trajectory estimation, which could reduce the reliability of the estimated mappings .

Moreover, parameter sensitivity remains a persistent issue. Algorithms like DBSCAN rely heavily on parameters such as neighborhood radius (ϵ) and minimum points (MinPts), which are difficult to tune in dynamic and noisy contexts. Advances such as the density-connectivity distance offer potential solutions but are still under exploration [BDH+23].

Finally, the need for real-time processing poses a significant barrier. Object tracking applications often require near-instantaneous results, conflicting with the computational demands of traditional density-based clustering methods. Techniques that integrate dimensionality reduction and efficient computational frameworks, such as Wasserstein-based methods, show promise but require further research and development [BRPP15].

Despite these challenges, density-based subspace clustering continues to be a vital area of study in object tracking. Its ability to adapt to diverse data distributions, combined with emerging techniques for scalability and robustness, suggests a strong potential for future advancements. Addressing the outlined challenges will require continued innovation in algorithm design, parameter optimization, and the integration of complementary clustering frameworks.

In this thesis, we build on these ideas by exploring Wasserstein-based distances in combination with density-connectivity clustering. This approach directly addresses several of the identified challenges: it leverages the structural advantages of optimal transport for capturing object movement, improves robustness against noise through density-connectivity, and provides a foundation for both clustering and motion extraction without relying on handcrafted features or supervision. By integrating these elements, the proposed methodology aims to bridge the gap between theoretical advances in distance-based clustering and practical requirements of unsupervised object tracking.

3. Methodology

In this section, we describe the experimental procedure used in our study. An overview of the main steps is illustrated in Figure 3.1. After introducing the overall flow, we will detail each step and its sub-steps in the following subsections.

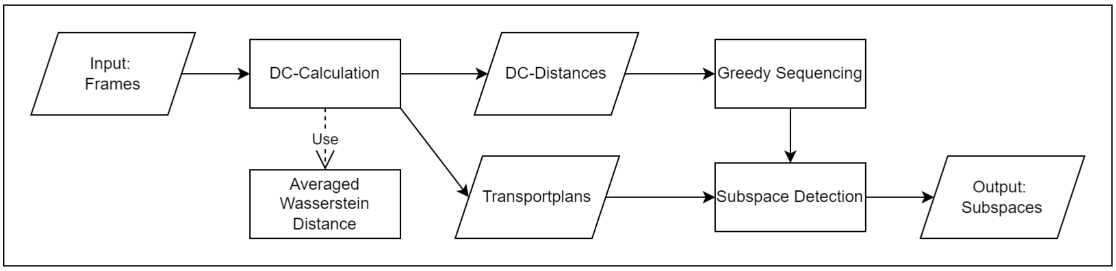

Figure 3.1.: Overview of experimental flow

3.1. Image Comparison with Wasserstein Distance Based on Image Mass Distributions

To determine whether two images originate from the same scene, we compute the Wasserstein distance between them. Each image is treated as a spatial distribution of mass, where pixel intensities correspond to the mass located at each spatial coordinate. This enables a comparison that accounts for both spatial layout and content distribution.

3.1.1. Representing Images as Discrete Distributions

Let $I \in \mathbb{R}^{H\times W\times 3}$ denote an RGB image. For each color channel, we convert the 2D intensity map into a discrete probability distribution:

$$\mu = \sum_{i=1}^{N} a_i \, \delta_{x_i}, \qquad \nu = \sum_{j=1}^{M} b_j \, \delta_{y_j}$$

Here:

- N, M denote the number of pixels in the respective image distributions μ and ν.
- $x_i, y_j \in \mathbb{R}^2$ are pixel coordinates (e.g., (x, y) positions),
- $a_i, b_j \geq 0$ are normalized pixel intensities with $\sum_i a_i = \sum_j b_j = 1$,

- δ_x is the Dirac delta centered at x.

This representation models each image channel as a mass distribution over space. Brighter pixels carry more mass and thus influence the distance computation more strongly. If the total intensity of a channel is zero, we default to a uniform distribution to avoid division by zero.

Since each image has three channels (in the RGB or CIELAB Colorspace), we compute the Wasserstein distance independently for each channel distribution. The final distance between two images I_1 and I_2 is the average over all three channels:

$$W_{\text{Colorspace}}(I_1, I_2) = \frac{1}{3} \sum_{c=1}^{3} W(\mu_c^{(1)}, \mu_c^{(2)})$$

Here, $\mu_c^{(1)}$ and $\mu_c^{(2)}$ denote the discrete distributions of color channel c from the first and second image, respectively.

3.1.2. Computing the Wasserstein Distance

To compare two such distributions, we solve the optimal transport problem using the Wasserstein distance. In our framework we specifically use the 1-Wasserstein distance (the case $p = 1$ of the general p-Wasserstein family), which measures transport cost directly in terms of Euclidean distances.

$$W(\mu, \nu) = \min_{\gamma \in \Pi(\mu,\nu)} \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma_{ij} \, \|x_i - y_j\|$$

Here, $\Pi(\mu, \nu)$ denotes the set of joint distributions with marginals μ and ν, i.e., all transport plans that redistribute the source mass μ onto the target ν.

The minimization is subject to the marginal constraints:

$$\sum_{j=1}^{M} \gamma_{ij} = a_i, \quad \sum_{i=1}^{N} \gamma_{ij} = b_j, \quad \gamma_{ij} \geq 0$$

Here, $\gamma \in \mathbb{R}^{N \times M}$ is the transport plan that indicates how much mass to move from location x_i to y_j. The cost of moving mass is given by the Euclidean distance between pixel coordinates:

$$C_{ij} = \|x_i - y_j\|$$

3.1.3. Practical Considerations for Transport Computation

In practice, the total mass of pixel intensities in an image channel (i.e., the sum of all pixel values) often differs between images. Since Wasserstein distance assumes distributions with equal total mass, we employ the following strategies to enable meaningful comparison:

(a) Normalization to Probability Distributions By default, we normalize each image channel so that the sum of all pixel values equals one:

$$a_i = \frac{I_1(x_i)}{\sum_k I_1(x_k)}, \qquad b_j = \frac{I_2(y_j)}{\sum_l I_2(y_l)}$$

This produces valid discrete probability distributions μ and ν for input to the Wasserstein distance. In the case where the total mass of a channel is zero (e.g., an entirely black channel), we fallback to a uniform distribution:

$$a_i = \frac{1}{N} \quad \text{for all } i$$

(b) Bucket-Based Mass Balancing To preserve absolute intensity differences between images without enforcing normalization, we optionally introduce an artificial *overflow bucket* to absorb excess mass or supply missing mass. This ensures that two images with differing total pixel mass (e.g., due to the appearance or disappearance of objects) can still be compared using Wasserstein distance.

Let $\Delta = \left|\sum_i a_i - \sum_j b_j\right|$ denote the mass imbalance between the two distributions. A dummy location x_* is added to the support of the smaller distribution, along with custom-defined transport costs. We experiment with the following cost strategies for assigning transport mass to the bucket:

- **Maximum cost:** Assign the worst-case penalty for unmatched mass:

$$C_{i,*} = \max_{i,j} \|x_i - y_j\|$$

- **Mean cost:** Assign the average spatial displacement as penalty:

$$C_{i,*} = \frac{1}{NM} \sum_{i,j} \|x_i - y_j\|$$

- **Edge-aware cost:** Assign bucket cost based on the minimum distance from each pixel to the border of the image. This reduces penalty for newly appearing objects at the image boundary and improves scene clustering robustness under object entry/exit:

$$C_{i,*} = \text{dist\_to\_edge}(x_i, H, W), \qquad C_{*,j} = \text{dist\_to\_edge}(y_j, H, W)$$

  where H and W are the image height and width, and:

$$\text{dist\_to\_edge}(x, H, W) = \min(x,\, W - x - 1,\, y,\, H - y - 1)$$

Using a zero cost ($C_{i,*} = 0$) is explicitly avoided. This would imply that unmatched mass can be discarded or generated *without penalty*, which causes degenerate behavior: for example, uniformly colored images (regardless of actual color or location differences) would yield zero distance, eliminating meaningful comparisons. This violates the spatial sensitivity required for clustering and sequencing, especially in structured scenes.

The edge-aware bucket cost provides a spatially meaningful fallback that still respects locality, allowing the framework to tolerate new object appearances near the image border without completely distorting the global transport structure.

(c) Color Space Considerations We support both RGB and CIELAB color spaces:

- **RGB:** Standard pixel-wise intensities in the $[0, 255]$ range.
- **CIELAB:** Perceptually uniform color space. Before processing, each channel is rescaled to $[0, 1]$:

$$L' = \frac{L}{100}, \quad a' = \frac{a + 128}{255}, \quad b' = \frac{b + 128}{255}$$

(d) Distance Metric Variants The cost matrix C used in the Wasserstein formulation can be built using different spatial metrics:

- **Euclidean:** $C_{ij} = \|x_i - y_j\|_2$
- **Squared Euclidean:** $C_{ij} = \|x_i - y_j\|_2^2$

The standard (non-squared) Euclidean distance yields the 1-Wasserstein metric. Using squared distances corresponds to the 2-Wasserstein variant and emphasizes larger displacements.

These design choices offer flexibility depending on the intended application: normalized distributions provide robustness and invariance, while bucket-based balancing preserves absolute differences. We use normalization in all experiments for consistency unless otherwise noted.

3.1.4. Motivation

Overall this method has several advantages:

- **Interpretability:** Pixel intensities map directly to mass; transport reflects visual changes.
- **Spatial sensitivity:** Matching is aware of object location and layout.
- **No learned features:** Works without pre-trained models or embeddings.

The resulting Wasserstein distances serve as a basis for scene similarity and are further used to construct a reachability graph in the next step.

3.2. Image Clustering Using Density-Based Connectivity (DC-Distance)

After computing the pairwise Wasserstein distances between all images (see Section 1), we construct a graph that captures the structural similarity between them. This graph serves as the basis for identifying local density and connectivity in the image set.

3.2.1. Reachability Distance Matrix

Let n be the number of images. From the pairwise Wasserstein distances $D \in \mathbb{R}^{n \times n}$, we define the *mutual reachability distance matrix* R as follows:

- For each image p, compute its *core distance* c_p, defined as the distance to its k-th nearest neighbor in D.
- The mutual reachability distance between image p and q is:

$$R_{pq} = \max\left(D_{pq}, c_p, c_q\right)$$

This adjustment ensures that only pairs of points that are both in locally dense regions will have small reachability distances. The matrix R forms the weighted adjacency matrix of a graph in which all edges correspond to potential density-connected image pairs.

3.2.2. Density-Connectivity Distance (DC-Distance)

To capture the global structure of image similarity while incorporating local density, we compute the *density-connectivity distance* between all image pairs. This is done by computing the minimax distance on the graph defined by R:

$$\text{dc-dist}(p, q) = \min_{P \in \mathcal{P}(p,q)} \max_{(u,v) \in P} R_{uv}$$

where $\mathcal{P}(p, q)$ is the set of all paths connecting node p and q in the graph.

This value corresponds to the largest edge weight on the most "narrow but dense" path between two images. It reflects how difficult it is to connect two images through dense regions of the graph.

3.2.3. Scene Clustering via k-Center Clustering on DC-Distance

To partition the full set of images into scenes, we apply k-center clustering on the density-connectivity distance matrix.

In k-center clustering, the objective is to choose k representative points (centers) such that the maximum distance from any point to its nearest center is minimized [Hak64]. Formally, for a set of n points and a distance matrix D, the goal is to find centers $C = \{c_1, \ldots, c_k\}$ such that:

$$\min_{C \subseteq \{1,\ldots,n\}, |C|=k} \max_{p \in \{1,\ldots,n\}} \min_{c \in C} D_{pc}$$

We apply a greedy 2-approximation algorithm for this purpose, which iteratively selects the point furthest from all existing centers as the next center, until k centers have been chosen. This method is fast, deterministic, and has strong approximation guarantees.

Why k-center is effective here: Following the motivation in of *Beer et al. (2023)* [BDH+23], k-center clustering is well-suited for our case because:

- It produces clusters that are tightly bounded in DC-Distance space, which reflects semantic density and connectivity between images.
- It is robust to variations in cluster shape and density, which is important since scenes may differ in appearance, content, and transitions.
- It scales efficiently to large image sets and does not rely on iterative convergence like k-means or DBSCAN.

Additionally, as noted in the original paper, k-center guarantees that no sample is further than a known bound from its cluster center, making it a good choice when spatial separation in feature space directly reflects scene structure, as is the case with Wasserstein-based DC-Distances.

3.3. Intra-Scene Sequencing via Greedy Path Construction

After clustering the full set of images into scenes using k-center clustering on the DC-Distance matrix, we aim to establish an ordering of images within each scene that reflects their temporal or structural continuity. Since DC-Distance is an ultrametric, applying k-center clustering in this space is equivalent to DBSCAN, as both correspond to different cuts of the same density-connectivity tree. [BDH+23] For sequencing, we then rely solely on the Wasserstein distance matrix D, which was already computed during distance evaluation. We employ a simple greedy sequencing algorithm that incrementally builds a path through the cluster, always extending it in the direction of the closest unvisited image. The goal is to produce a linear ordering of the images that minimizes the total pairwise Wasserstein distance along the path and which allows to recover temporal structure even when explicit frame order is not available.

Pseudocode: Greedy Scene Sequencing

```
Input: Cluster C of images, Distance matrix D
Initialize: Sequence = [any image from C]
While unvisited images remain:
    Look at first and last image in Sequence
    Find the unvisited image with smallest distance to either end
    Add that image to the closer side of Sequence
Return: Ordered Sequence
```

Why this works: This method is valid within a scene because the k-center clustering ensures that all images in a scene lie in a compact, well-connected region of the Wasserstein space. Within such a space, greedy traversal tends to approximate the shortest path well, similar to how nearest-neighbor heuristics perform well on locally coherent subsets in Travelling Salesman-like problems.

Why this does not generalize: If applied to the entire dataset without prior clustering, this method would likely produce incorrect orderings, frequently jumping between unrelated scenes. This is because Wasserstein distances between distant or semantically unrelated images can still be small (e.g., due to shared background or low contrast), misleading the greedy logic.

Moreover, finding the globally optimal ordering over all images across scenes, corresponds to solving the *Hamiltonian path problem*, which is NP-hard. The best-known exact solution has time complexity $\mathcal{O}(n^2 \cdot 2^n)$, which becomes infeasible for even moderately sized datasets. The clustering step prevents this by partitioning the problem into smaller, scene-specific subproblems where greedy heuristics are computationally effective. Without this partitioning, a global greedy traversal could irreversibly attach frames from different scenes in the early steps, preventing later insertion of the correct within-scene neighbors.

3.4. Identifying Object Movement via Transport Plans

After ordering the images within each scene, we analyze the transport plans $\gamma^{(t)}$ computed between successive images (I_t, I_{t+1}) to localize object movement. Each plan indicates how pixel mass is redistributed between two frames, and this information is used to extract motion patterns in a completely unsupervised and interpretable way.

3.4.1. From Transport Plans to Mass Flow Vectors

Each transport plan $\gamma^{(t)}$ is a matrix in $\mathbb{R}^{N \times M}$ that describes how mass moves from pixel positions in image I_t to positions in I_{t+1}. We interpret this as a collection of *mass flow vectors* from source locations x_i to destination locations y_j.

To extract meaningful movement signals, we:

- Filter out all self-transports ($x_i = y_j$) and all entries where $\gamma_{ij} = 0$.
- Compute the total transport mass per flow and retain only those above the mean mass.
- Represent each vector as a pair (x_i, y_j) with assuming direction in order of sequencing. This contains a bias, since the temporal direction is not known during computation, but shows the sequential order either correct or in opposite direction.

The representation focuses on **local structure and displacement**, enabling the detection of spatial changes regardless of exact temporal orientation.

3.4.2. Clustering Transport Vectors into Movement Subspaces

To identify regions of consistent movement, we cluster the transport vectors using HDBSCAN. Each vector is now represented in the standard motion format as a 4D tuple:

$$(x_{\text{src}}, y_{\text{src}}, \Delta x, \Delta y)$$

where $(x_{\text{src}}, y_{\text{src}})$ denotes the starting position of the transported mass, and $(\Delta x, \Delta y)$ encodes the direction and magnitude of the transport.

This representation preserves directional information and spatial locality, allowing HDBSCAN to group vectors that share both similar origins and similar motion patterns. The resulting clusters define movement subspaces—localized regions in the image where coherent and repeated motion occurs. These often correspond to object trajectories, deformation boundaries, or zones of consistent activity across the sequence.

3.3 Visual Interpretation and Benefits

Each transport vector is visualized with a color encoding its angle (hue) and a scaled length. Vectors below the mean transport mass are discarded for clarity. After clustering, vectors from the same group are visualized with consistent colors, revealing structure in the motion pattern.

Key advantages of this approach:

- **Interpretability:** Displacement flows give intuitive, spatially grounded motion signals.
- **Object-independence:** The method does not rely on object detection, feature maps, or semantic labels.
- **Localization:** It identifies *where* changes occur, and groups them without assumptions about object shape or type.

This provides a high-resolution, fully unsupervised method for understanding object motion from transport plans derived via Wasserstein distances.

4. Experiments

4.1. Implementation Environment

All experiments were executed on a Microsoft Azure VM (Standard F4ams v6, 4 vCPUs, 32 GiB RAM, North Europe region). The VM ran Ubuntu 22.04.5 LTS (Jammy Jellyfish) with Linux kernel 6.8.0-1030-azure. The development environment was based on Python 3.10.12. The complete list of installed Python packages is provided in the Appendix.

4.2. Datasets

To evaluate the different stages of the proposed framework (including image comparison, scene clustering, ordering, and subspace motion analysis) we test on a diverse set of datasets. These range from highly controlled synthetic settings to real-world video sequences, covering various object types, motion patterns, and background complexities. The following subsections describe the datasets used.

Evaluation Subset	**Source Dataset**	**Frames**	**Pixels**	**Clusters**	**Channel**
Synthetic Dataset	-	20	15×15	2	3
Synth-B	-	20	15×15	2	3
Coil-100 Subset	Coil-100	64	32×32	8	3
Kitti1	Kitti-Benchmark	114	32×32	11	3
Kitti2	Kitti-Benchmark	30	32×32	3	3
SVIA-Subset	SVIA-Dataset	20	64×64	2	1
MammAlps-Subset	MammAlps-Dataset	30	54×54	3	3

Table 4.1.: Overview of all datasets used in this study, including their key characteristics and relevant details.A dash indicates that the dataset was created by the author and not derived from an external source.

4.2.1. Synthetic Dataset

To establish a controlled test environment with minimal confounding variables, we created a synthetic dataset consisting of 15×15 pixel colored images. Each image contains one or two simple geometric objects (triangles, rectangles or circles in red, green or blue) placed on a black background. A scene consists of 10 consecutive frames, where the objects move by one pixel per frame in a fixed direction. This dataset serves as a minimal baseline to validate core functionality such as Wasserstein-based distance comparison and motion extraction under idealized conditions.

In addition, we constructed a variant called **Synth-B**, which is identical in structure and object motion but introduces slight frame-to-frame variations in the background lightning, adding a controlled degree of visual noise to test robustness against minor illumination changes.

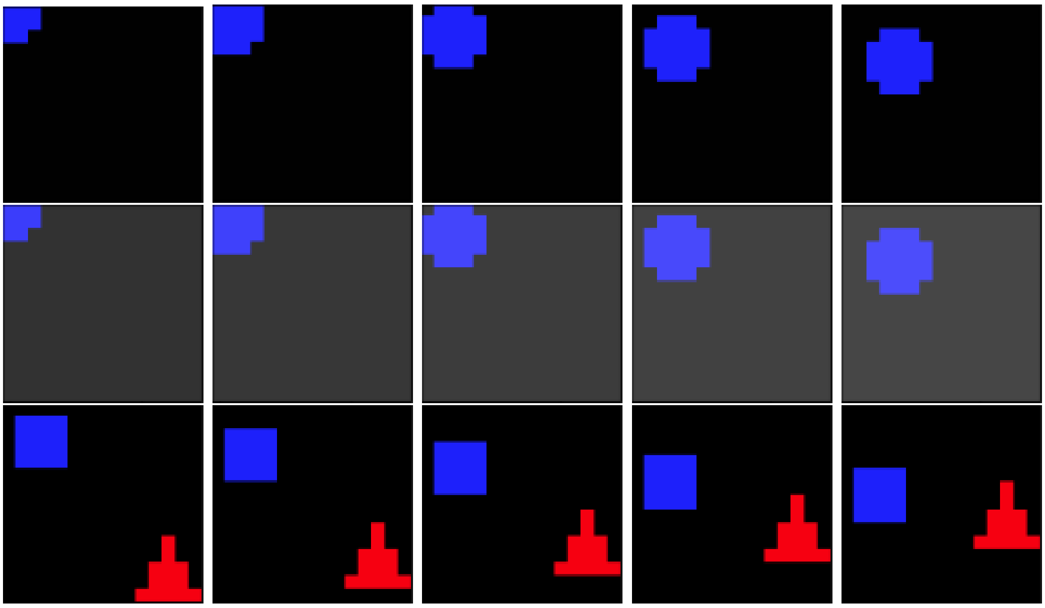

Figure 4.1.: Exemplary frames from the synthetic dataset: first row: one moving object; second row: one moving object with changing lighting (Synth-B); third row: two moving objects.

4.2.2. COIL-100

The Columbia Object Image Library (COIL-100) [NNM96] is a dataset containing images of 100 different objects taken from multiple angles. Each object is photographed against a uniform black background while rotating on a turntable. In our experiments, each scene consists of sequential views of a single object undergoing rotation. This allows us to analyze how well the framework handles intra-object transformation such as rotation while controlling for background variation. We use only a selected subset of objects and scale the images to match the resolution and format used throughout this thesis.

4.2.3. KITTI

The KITTI dataset [GLU12] provides real-world driving scenes captured from a moving vehicle, including object labels for tracking and detection. For our purposes, we extract

Figure 4.2.: Examplary downscaled frames from the Coil-100 Dataset[NNM96]

labeled object patches (e.g., cars, pedestrians) from selected sequences and normalize them linearly to a common resolution of $32 \times 32. Theseobjectcropsareusedasinputforclusteringandorderingexperimen$

4.2.4. SVIA Dataset

The SVIA dataset [CLZ+22] consists of microscopic video recordings designed for computer-aided sperm analysis. Each scene contains a small number of objects (sperm cells) moving against a relatively uniform and low-noise background. This dataset serves as a test case for tracking performance in low-resolution, low-texture scenarios where color and shape information are limited. We use full-frame sequences and focus on testing the motion subspace detection capabilities of our framework under laboratory-like conditions.

4.2.5. MammAlps

The MammAlps dataset [GQF+25] includes wildlife videos collected with synchronized multi-view camera traps in the Swiss Alps. It contains recordings of several mammal species in natural environments, covering different activities such as foraging, moving, or resting. Each event is captured from multiple perspectives and includes challenging factors such as background motion, partial occlusions, and varying lighting conditions. Since the recordings are made with static cameras, background dynamics remain limited, making the dataset particularly practical for evaluating object-movement tracking in real-world outdoor scenarios.

Figure 4.3.: Examplary extracted labeled objects from the Kitti-Dataset[GLU12]

4.3. RQ1: Ablation and Evaluation of Wasserstein Distance in Density-Connectivity Clustering

To what extent does replacing the base distance in density-connectivity clustering with Wasserstein distance improve object tracking and data mining performance on image data?

4.3.1. Motivation and Evaluation Strategy

This research explores whether using Wasserstein distance, known for its sensitivity to spatial structure and mass transport, can enhance the discriminative power of **density-connectivity-based clustering** when applied to image data. In the original "Connecting the Dots" framework [BDH+23], the DC-Distance operates over a mutual reachability graph based on Euclidean metrics. However, Euclidean distance often fails to capture semantic similarity in image spaces. Wasserstein, in contrast, can account for pixel intensity distribution and spatial arrangement, potentially improving the separation between clusters. To evaluate this, we systematically test several preprocessing and representation strategies. Cluster quality is evaluated using:

- **Intra- vs. inter-cluster distance separability** (ground-truth labels) [JD88],
- **Silhouette score** (cohesion vs. separation) [Rou87],
- **Dunn Index** (minimum inter-cluster distance / maximum intra-cluster distance) [Dun74].

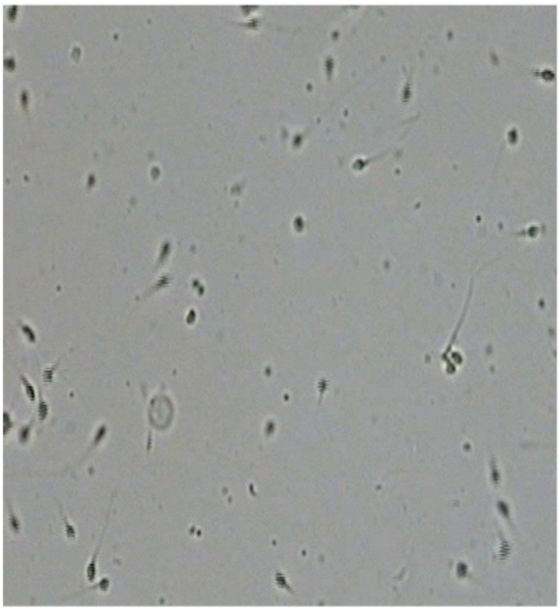

Figure 4.4.: Examplary Frame from the SVIA-Dataset[CLZ+22]

4.3.2. Sub-Questions and Experimental Variants

1. **Is Wasserstein distance a better base metric than Euclidean for DC-Distance in clustering image data?**
   Wasserstein distance measures the optimal transport cost between pixel intensity distributions. Unlike Euclidean distance, which treats all pixel positions equally, Wasserstein reflects global structural shifts, making it potentially more robust to noise, deformation, and minor pixel-level misalignment.

2. **Does normalization of pixel values improve Wasserstein distance calculations more than histogram bucketing when comparing images with different mass?**
   In image comparisons, pixel mass often varies due to lighting conditions or occlusion. Normalizing pixel values scales each image to the same total mass, mitigating lighting variations and making distributions comparable and is also necessary to perform a comparison via wasserstein-distance. Bucketing (binning intensities into histograms) adds discretization, but its granularity choice is non-trivial and can obscure finer differences. We hypothesize that normalization provides more stable and discriminative distance values under varying imaging conditions.

3. **Does computing and averaging Wasserstein distances across channels outperform using barycentric fusion or sliced Wasserstein with random projections ?**
   Instead of reducing color to grayscale, we compute the Wasserstein distance separ-

ately for each RGB channel and then average the three results. This preserves color information while maintaining the interpretability and modularity of Wasserstein transport. We compare this approach against:

- Barycentric fusion (computing a Wasserstein barycenter between distributions), and
- Sliced Wasserstein distance with random projections

to assess which strategy better supports meaningful cluster formation in color image data.

4. **Is RGB color space more suitable than CIELAB for Wasserstein-based dc-clustering in image analysis?**
   Although CIELAB is perceptually uniform, its transformation can distort spatial relationships relevant for transport-based distances. We test both spaces to determine which supports better intra- vs. inter-cluster separability.

5. **Does using Wasserstein-based DC-Distance outperform regular Wasserstein and mutual reachability-based clustering strategies?**
   We compare three approaches:
   - DC-Distance using Wasserstein-distance as base,
   - Wasserstein-distance, and
   - mutual reachability distance (using Wasserstein-distance as base). This experiment tests whether adding density-connectivity constraints improves clustering quality beyond what Wasserstein already provides in isolation.

6. **Is density-connectivity a strong clustering strategy for image data, and how does the minPts parameter influence its performance?**
   DC-Distance provides a formal ultrametric that captures both connectivity and density constraints, offering robustness against noise and the single-link effect. Its ability to yield consistent DBSCAN*, k-center, and spectral clusterings under a unified framework simplifies hyperparameter tuning and improves interpretability [BDH+23]. We test different minPts (µ) settings to control the density threshold for core points, assessing how this influences clustering stability and ground-truth alignment.

In the following, pairwise distance relationships between images are visualized using heatmaps. Each heatmap represents a symmetric distance matrix, where both the x- and y-axes correspond to all images in the dataset. The color intensity encodes the pairwise distances: dark blue indicates high similarity (small distance), while bright yellow indicates low similarity (large distance). To make visual patterns easier to recognize, images belonging to the same scene are arranged consecutively, which leads to a block-diagonal structure when similar scenes cluster together.

4.3.3. Evaluation and Results

Wasserstein vs. Euclidean-distance

To evaluate whether Wasserstein distance improves clustering quality when used as the base metric in DC-Distance computations, we compare its performance against standard Euclidean distance across subsets of four datasets: Synthetic, COIL, and two KITTI subsets. The central hypothesis is that Wasserstein distance, by incorporating spatial structure and intensity distribution, provides more meaningful similarity measures between images than pixel-wise Euclidean distance.

Distance-Measure	Dataset	Sil-Score	Dunn	Sep-Score
Wasserstein	Synth	**0.563**	**0.660**	**2.3**
	COIL	**0.725**	**0.685**	**7.2**
	KITTI1	**0.481**	0.169	**3.0**
	KITTI2	**0.464**	0.325	**2.4**
Euclidean	Synth	0.346	1.123	1.5
	COIL	0.545	0.662	2.9
	KITTI1	0.362	**0.462**	1.9
	KITTI2	0.342	**0.586**	1.7

Table 4.2.: Comparison of clustering metrics

Quantitative Evaluation. Table 4.2 summarizes the clustering results using three standard metrics: Silhouette Score, Dunn Index, and Separation Score. Across most datasets, Wasserstein generally outperforms Euclidean distance, especially in terms of Silhouette- and Separation Scores:

- On the **Synthetic** dataset, Wasserstein achieves significantly higher values for Silhouette Score (0.563 vs. 0.346) and Separation Score (2.3 vs. 1.5), indicating well-defined and cohesive clusters even under minimal complexity.
- On the **COIL** dataset, which features rotating objects against a black background, Wasserstein again leads with a Silhouette Score of 0.725 and a high Separation-Score of 7.2, showing robustness to transformation.
- On the more challenging **KITTI1** and **KITTI2** datasets, Wasserstein continues to outperform Euclidean in Silhouette and Separation metrics. However, for the Dunn Index, Euclidean performs slightly better. This could be attributed to the Dunn Index's sensitivity to outliers and to variations in object scale and background clutter, which may affect the minimal inter-cluster distances that dominate the index.

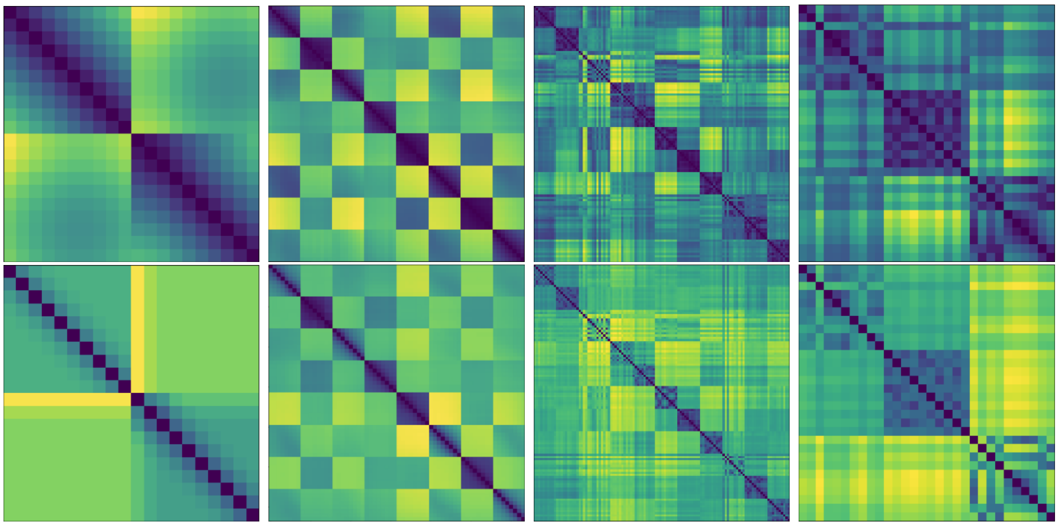

Figure 4.5.: Comparison of pairwise distance heatmaps between images across four datasets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed using Wasserstein distances. Bottom row: Heatmaps computed using Euclidean distances.

Qualitative Insights from Distance Heatmaps. Heatmap visualizations 4.5 of the pairwise distance matrices further support these results. For Wasserstein-based matrices, we observe:

- Clearer separation between visually dissimilar images.
- Smooth gradation along sequentially similar frames.
- A sensitivity to object structure and displacement even when no common subspace is shared.

In contrast, Euclidean distance appears to penalize even slight object deformations inconsistently and often assigns uniform distances to entirely distinct images when subspace overlap is lacking. In practice, this means situations where objects barely intersect or align between consecutive frames. This behavior is especially pronounced in the **Synthetic dataset**, where Wasserstein distance reflects relative movement even across object boundaries, while Euclidean comparison fails to differentiate frames lacking pixel-wise alignment.

Normalization vs. Bucketing

In real-world image sequences, the total pixel mass may vary across frames due to illumination changes, occlusion, or objects entering or leaving the frame. To address this, we evaluate three mass-balancing strategies used prior to Wasserstein distance computation:

- **Normalization:** Each image channel is normalized such that the total pixel mass sums to 1, removing absolute brightness differences and making distributions directly comparable.

- **Max Distance Bucketing:** A dummy bucket absorbs unmatched mass, with a fixed transport cost equal to the maximum pairwise pixel distance. This penalizes unmatched mass uniformly, preserving intensity differences across images.

- **Border-Aware Bucketing:** Each pixel's cost to the dummy bucket is its distance to the closest image border. This approach reduces penalties for new mass appearing at the image edge, encouraging tolerance toward object entry at frame boundaries.

To explicitly test sensitivity to brightness variation, we introduce a variant of the synthetic dataset, called **Synth-B**, which introduces per-frame brightness changes to simulate mild lighting fluctuations.

Mass-Balancing	Dataset	Sil-Score	Dunn	Sep-Score
Normalization	Synth	0.563	0.660	2.3
	Synth-B	**0.502**	**0.506**	**2.0**
	COIL	**0.725**	**0.685**	7.2
	KITTI1	**0.481**	**0.169**	3.0
	KITTI2	0.464	0.325	2.4
Max Loss	Synth	**0.827**	**1.403**	**6.1**
	Synth-B	0.058	0.152	1.1
	COIL	0.715	0.266	**12.2**
	KITTI1	0.377	0.050	**4.4**
	KITTI2	**0.786**	**0.615**	**7.7**
Border Dist	Synth	0.253	0.082	1.5
	Synth-B	0.201	0.045	1.2
	COIL	0.763	0.634	10.7
	KITTI1	0.455	0.079	3.1
	KITTI2	0.345	0.253	2.3

Table 4.3.: Clustering scores across mass-balancing methods.

Quantitative Evaluation. Table 4.3 summarizes the performance of all three strategies across six datasets using Silhouette Score, Dunn Index, and Separation Score.

Key observations:

- On the original **Synthetic** dataset, **Max Distance Bucketing** achieves the best scores across all metrics (Silhouette-Score: 0.827, Dunn-Index: 1.403), confirming its ability to distinguish moving objects through absolute mass changes.

- However, on **Synth-B**, Max Bucketing fails (Silhouette-Score: 0.058), as global brightness variation causes artificial mass imbalances that dominate the transport cost and fragment the clusters.
- **Normalization** handles such brightness shifts more adequately. It achieves the best overall stability on **Synth-B** (Silhouette-Score: 0.502) and performs robustly across COIL and KITTI subsets.
- **Border-Aware Bucketing** performs moderately overall. It helps tolerate object entry at frame edges but suffers when objects continuously move along borders, which can distort transport signals.

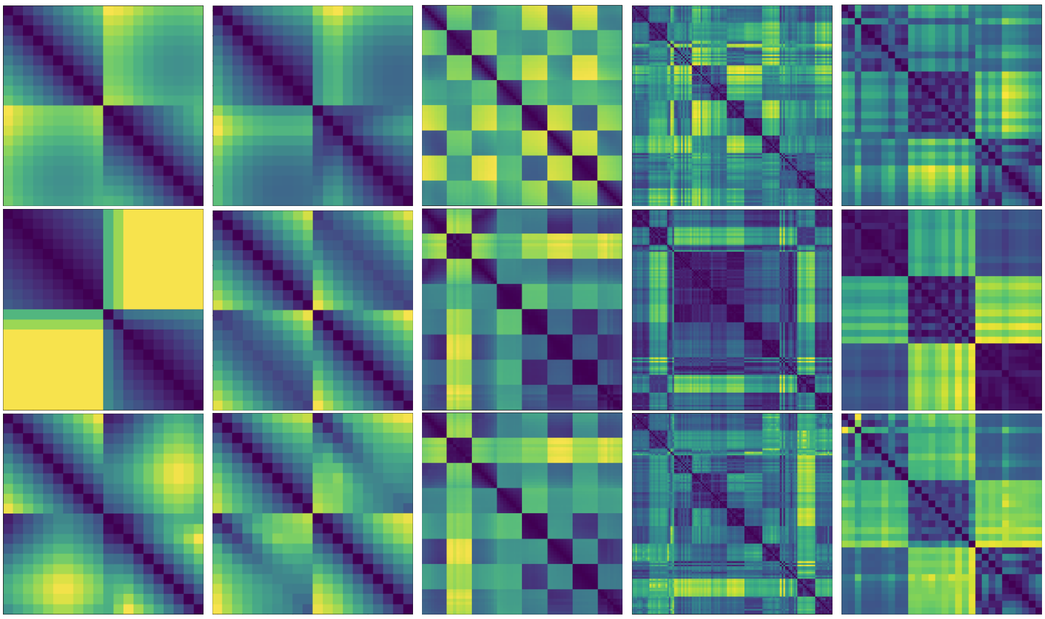

Figure 4.6.: Comparison of pairwise distance heatmaps between images across five datasets: Synthetic, Synth-B, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed using normalization for mass-balancing. Middle row: Heatmaps computed using Maximum-Loss for unbalanced mass. Bottom row: Heatmaps computed using Border Distance-Loss for unbalanced mass.

Qualitative Insights from Distance Heatmaps. Visual inspection of pairwise distance heatmaps reveals that:

- **Max Distance Bucketing** produces sharp cluster blocks on the Synthetic dataset, showing clear separation between scenes even without subspace overlap. This validates its ability to quantify absolute object displacement.

- On **Synth-B**, this same method produces erratic distance patterns where brightness shifts are misinterpreted as semantic changes, breaking cluster coherence.
- **Normalization** produces smoother and more coherent heatmaps across both synthetic variants. This highlights its strength in suppressing global intensity variation while preserving structural similarity.
- **Border-Aware Bucketing** results in ambiguous transitions when objects move along edges. It distinguishes scene entry/exit well but fails to reflect smooth motion along borders due to variable cost weighting.

Averaging vs. Barycenter vs. Sliced

To incorporate color information into the Wasserstein-based image comparison, we explore three strategies for combining the three channels (RGB or perceptually uniform color spaces):

- **Per-Channel Averaging:** Compute Wasserstein distance independently for each channel and average the three results. This modular approach retains interpretability and avoids complex fusion operations.
- **Barycentric Fusion:** Fuse the three channel-wise distributions into a single joint barycenter before comparing images. This method has theoretical appeal but may suffer in structured scenes where most mass lies in one dimension of the color space.
- **Sliced Wasserstein Distance:** For each channel, the 2D mass distribution is projected along multiple random directions to obtain 1D histograms. Wasserstein distances are computed along each slice and averaged per channel, followed by averaging across all channels. This method approximates full optimal transport while being more efficient and often more robust to high-dimensional noise.

Quantitative Evaluation. Table 4.4 compares the three fusion strategies across all datasets using Silhouette Score, Dunn Index, and Separation Score.

Key observations:

- On the **Synthetic** dataset, **Barycentric Fusion** fails to distinguish any meaningful clusters (Silhouette-Score: -0.042, Dunn-Index: 0.010), highlighting a major shortcoming. Since synthetic objects are rendered with solid colors primarily expressed in a single color-channel, the barycentric fusion collapses their structure into degenerate forms, losing crucial spatial separation.
- **Per-Channel Averaging** performs well across datasets, showing strong results on COIL (Silhouette-Score: 0.725, Dunn-Index: 0.685) and good performance on KITTI subsets. This suggests that averaging per-channel transport captures structural similarity while preserving color sensitivity.

Channel-Calculation	Dataset	Sil-Score	Dunn	Sep-Score
Averaging	Synth	0.563	0.660	2.3
	COIL	**0.725**	0.685	**7.2**
	KITTI1	**0.481**	**0.169**	3.0
	KITTI2	**0.464**	**0.325**	**2.4**
Barycenter	Synth	-0.042	0.010	0.9
	COIL	0.554	0.135	5.5
	KITTI1	0.432	0.068	**3.2**
	KITTI2	0.401	0.194	2.5
Sliced	Synth	**0.586**	**0.709**	**2.4**
	COIL	0.717	**0.723**	6.7
	KITTI1	0.460	0.137	2.9
	KITTI2	0.424	0.275	**2.4**

Table 4.4.: Clustering scores for different channel combination methods.

- **Sliced Wasserstein Distance** achieves the best Dunn Index on COIL (0.723) and overall solid performance across all datasets. Its slight edge over averaging on some datasets (e.g., Synthetic) supports the idea that directional projections offer robustness while preserving discriminative power.

Qualitative Insights from Distance Heatmaps. Visual inspection of the distance matrices confirms the quantitative findings:

- **Barycentric Fusion** produces unclear, noisy distance maps on the Synthetic dataset, with minimal differentiation between image pairs. This is expected when color information is sparse or unevenly distributed across channels.
- **Per-Channel Averaging** leads to more structured and interpretable distance matrices, especially in datasets with clear geometric object placement and consistent appearance (e.g., COIL, KITTI1).
- **Sliced Wasserstein** produces clean cluster separation patterns, especially on COIL, where object rotation and consistent lighting benefit from robust directional projections.

RGB vs. CIELAB

CIELAB is a perceptually uniform color space, meaning that Euclidean distances between colors correlate more strongly with human perception than in RGB. However, this perceptual uniformity does not necessarily imply suitability for optimal transport calculations. The transformation from RGB to CIELAB is nonlinear [CIE76] and may distort pixel intensity distributions in a way that affects Wasserstein-based distance metrics. In this

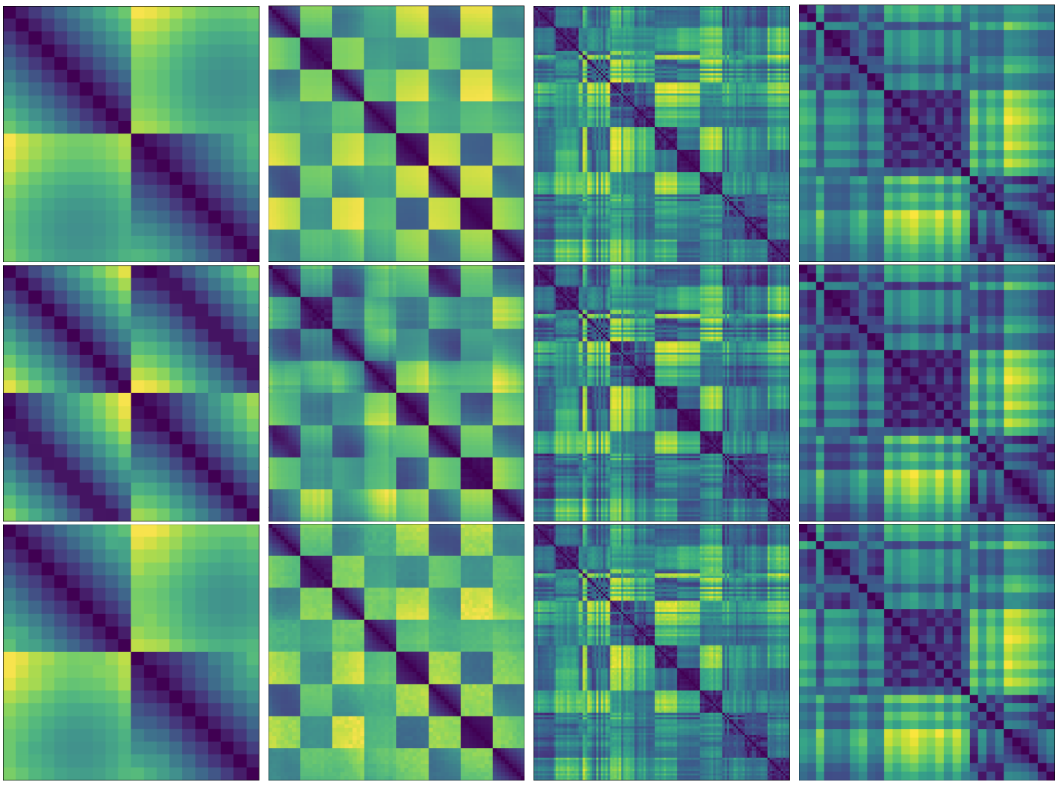

Figure 4.7.: Comparison of pairwise distance heatmaps between images across four datasets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed using averaging for channel-merging. Middle row:Heatmaps computed using a wasserstein-barycenter for channel-merging. Bottom row: Heatmaps computed using sliced wasserstein.

experiment, we evaluate whether RGB or CIELAB space provides better clustering performance when used in density-connectivity clustering.

Color Space Transformation. To make CIELAB comparable to RGB in Wasserstein computation, we normalize each channel independently to the $[0, 255]$ range, ensuring:

- All values are non-negative (Wasserstein requires non-negative mass).
- All three channels have similar dynamic ranges.

After this transformation, Wasserstein distances are computed per channel and averaged across channels, following the same procedure as in the RGB experiments.

Quantitative Evaluation. Table 4.5 shows clustering results across four datasets using Silhouette-, Dunn-, and Separation Scores.

Key observations:

Colorspace	Dataset	Sil-Score	Dunn	Sep-Score
RGB	Synth	**0.563**	**0.660**	**2.3**
	COIL	**0.725**	**0.685**	**7.2**
	KITTI1	**0.481**	0.169	**3.0**
	KITTI2	0.464	0.325	2.4
CIELAB	Synth	0.018	0.094	1.0
	COIL	0.700	0.666	6.6
	KITTI1	0.475	**0.190**	2.9
	KITTI2	**0.478**	**0.380**	**2.5**

Table 4.5.: Comparison of RGB and CIELAB colorspaces for Wasserstein-based clustering.

- On the **Synthetic** dataset, RGB significantly outperforms CIELAB (Silhouette-Score: 0.563 vs. 0.018). This is likely due to the fact that synthetic images were generated using high-saturation RGB primitives. The CIELAB transformation spreads these into less linearly separable regions, harming transport-based distance calculation.

- On the **COIL**, **KITTI1**, and **KITTI2** datasets, the performance differences between RGB and CIELAB are minimal. While CIELAB slightly outperforms RGB on KITTI2 and has a marginally higher Dunn Index in KITTI1, RGB is slightly better in terms of Silhouette- and Separation Scores on both. On COIL, RGB and CIELAB are nearly tied. These results suggest that both spaces perform comparably on real-world data, with no clear overall advantage.

- Normalization was essential in CIELAB to avoid negative values and balance the influence of each channel. Without it, transport distances would be skewed due to channel imbalance or undefined behavior in negative mass space.

Qualitative Insights from Distance Heatmaps. Visual inspection confirms the numerical results:

- In **Synthetic**, distance matrices in CIELAB show little to no visible clustering, likely because pixel values that are extreme in RGB do not map to equally extreme values in CIELAB, reducing separability.

- Distance heatmaps in CIELAB and RGB are visually similar on natural datasets. No consistently clearer separation pattern is visible across them, confirming that the clustering quality is comparable between the two color spaces in real-world imagery.

- RGB remains more intuitive and reliable for data created or labeled in RGB space, particularly synthetic or graphics-like datasets.

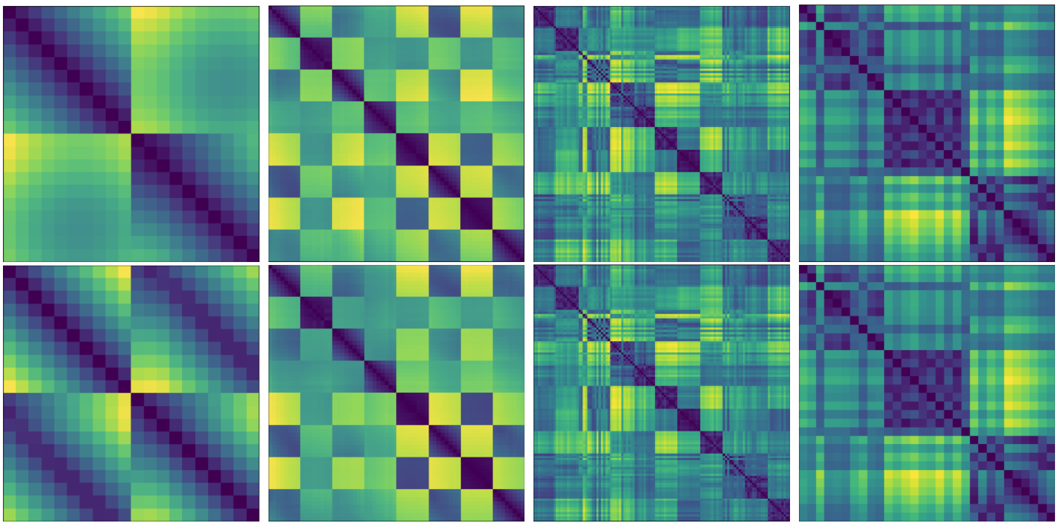

Figure 4.8.: Comparison of pairwise distance heatmaps between images across four datasets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed in the RGB-Colorspace. Bottom row: Heatmaps computed in the CieLAB-Colorspace.

Density-Connectivity

This experiment investigates whether incorporating Wasserstein distance into the density-connectivity distance framework yields improvements over using Wasserstein distance alone. We evaluate both on the same datasets and compare standard clustering quality metrics.

Distance-Measure	Dataset	Sil-Score	Dunn	Sep-Score
Wasserstein	Synth	0.563	0.660	2.3
	COIL	0.725	0.685	7.2
	KITTI1	0.481	0.169	**3.0**
	KITTI2	0.464	0.325	**2.4**
DC-Distance	Synth	**0.832**	**5.867**	**6.0**
	COIL	**0.854**	**2.364**	**10.7**
	KITTI1	**0.550**	**0.416**	2.6
	KITTI2	**0.501**	**0.823**	2.1

Table 4.6.: Comparison of clustering metrics across two distance measures.

Quantitative Evaluation. The results demonstrate that DC-Distance consistently outperforms plain Wasserstein distance across all datasets. The improvements are particularly pronounced on structured or synthetic data. For example, on the **Synthetic** dataset, the Silhouette Score rises from 0.563 to 0.832, and the Dunn Index from 0.660 to 5.867.

On **COIL**, the improvements are similarly strong (Silhouette-Score: 0.725 to 0.854, Dunn-Index: 0.685 to 2.364). These results suggest that density-connectivity introduces stronger cluster cohesion and better separation compared to direct pairwise comparisons.

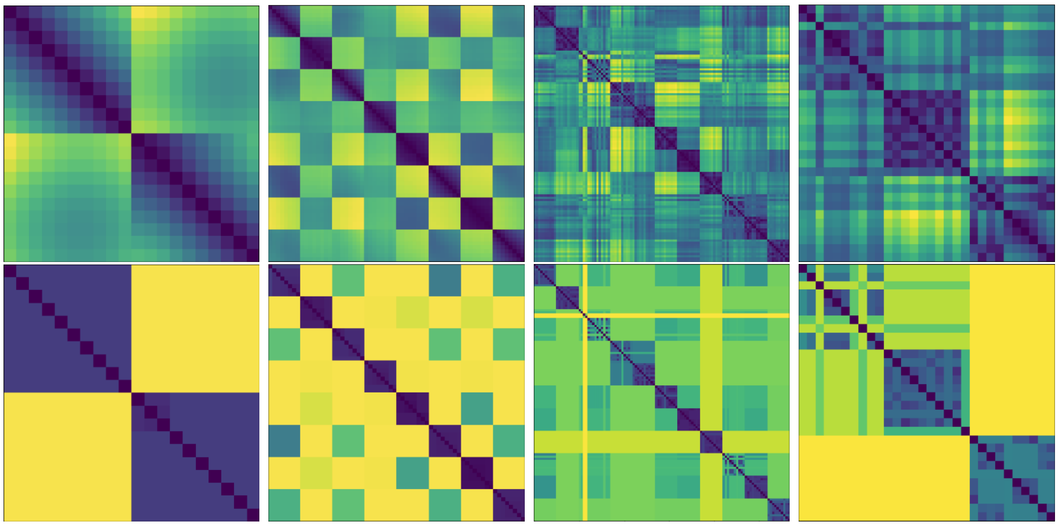

Figure 4.9.: Comparison of pairwise distance heatmaps between images across four datasets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed using Wasserstein-Disctance. Bottom row: Heatmaps computed using DC-Distance based on Wasserstein-Distance.

Qualitative Insights from Distance Heatmaps. The DC-Distance matrices reveal more uniform and block-like structures than the plain Wasserstein matrices. Especially in the Synthetic and COIL datasets, the clusters appear more sharply separated. These improvements confirm the benefits discussed in [BDH+23], where DC-Distance is shown to promote global consistency by integrating local density structure.

Degree of Density-Connectivity

We evaluate whether DC-Distance is a suitable foundation for image-based clustering and how its density sensitivity, controlled by the *minPts* parameter, impacts clustering quality.

Table 4.7 highlights a clear trend: as *minPts* increases, clustering performance consistently drops across all datasets. The best results are achieved with **minPts = 1 or 2**, which yield the highest Silhouette- and Dunn scores and largest inter-cluster separability. For example, on the **COIL** dataset, the Silhouette score drops from 0.854 at $minPts = 1/2$ to 0.442 at $minPts = 8$.

Why Do *minPts* = 1 and 2 Yield Identical Results? This equivalence arises from the nature of the image data. In DC-Distance, the mutual reachability between two points

MinPoints	Dataset	Sil-Score	Dunn	Sep-Score
1 or 2	Synth	**0.832**	**5.867**	**6.0**
	COIL	**0.854**	**2.364**	**10.7**
	KITTI1	**0.550**	**0.416**	**2.6**
	KITTI2	**0.501**	0.823	**2.1**
5	Synth	0.502	1.467	2.0
	COIL	0.645	0.998	4.5
	KITTI1	0.355	0.363	1.8
	KITTI2	0.253	**0.961**	1.3
8	Synth	0.076	0.849	1.1
	COIL	0.442	0.703	2.7
	KITTI1	0.218	0.402	1.4
	KITTI2	0.050	0.646	1.1

Table 4.7.: Clustering results for different MinPoints values.

depends on the maximum of their core distances (to their μ-th nearest neighbor) and the base metric. In short video sequences or synthetic image sets, each frame is strongly connected to its immediate neighbors, the preceding and following frames. Since both of these neighbors are spatially and semantically very similar to the current frame, the core distances for $\mu = 1$ and $\mu = 2$ differ minimally, if at all. This leads to identical mutual reachability values and consequently to an identical DC-Distance matrix and clustering outcome.

Qualitative Insights from Distance Heatmaps. In the distance heatmaps, we observe that small *minPts* values (1-2) lead to clear block structures with strong contrast between intra- and inter-cluster distances. This reflects tight cohesion within scenes and clear separation between them. As *minPts* increases, these patterns degrade: the blocks blur and contrast diminishes, especially on Synthetic and KITTI datasets.

This phenomenon can be attributed to the nature of video data: temporally adjacent frames form natural chains due to their minimal visual differences. These frame-to-frame links effectively form a **single-linkage chain**, connecting all images within a scene. When *minPts* is small, this sequential structure is sufficient to capture the core connectivity of a cluster. However, increasing *minPts* forces the distance graph to include additional neighbors, which leads to weak inter-scene links and spurious cluster merging. This reduces overall contrast and cluster fidelity.

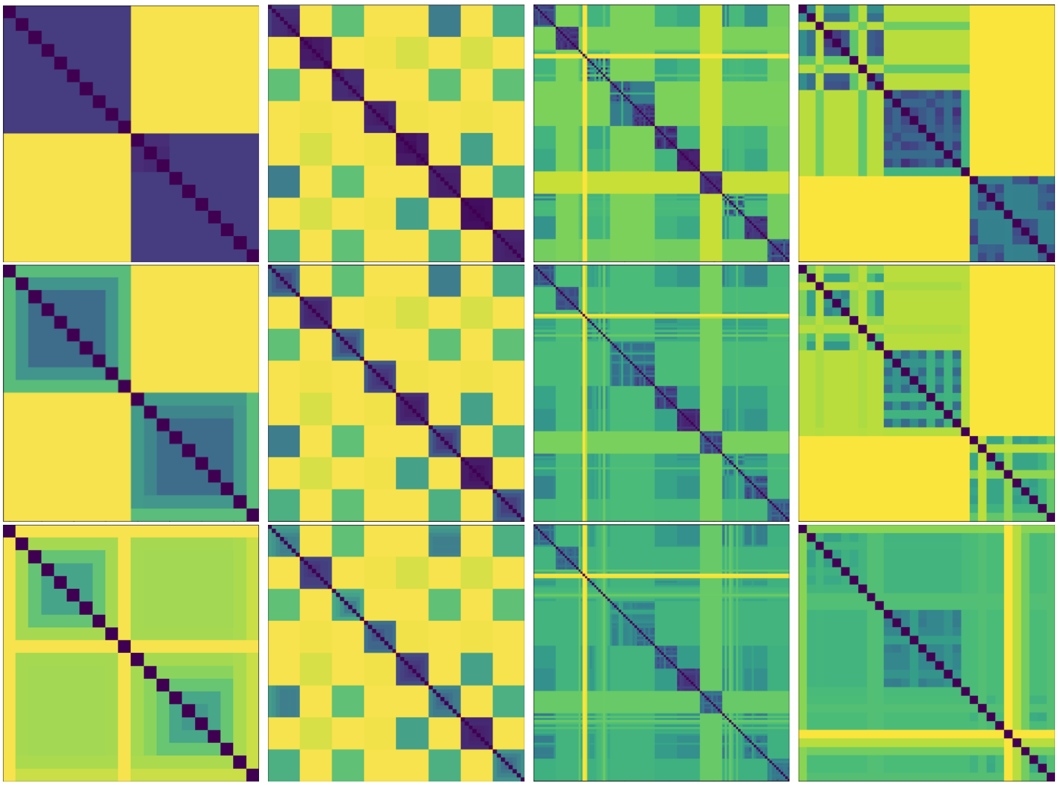

Figure 4.10.: Comparison of pairwise distance heatmaps between images across four datasets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Top row: Heatmaps computed using DC-Distance based on Wasserstein-Distance with min-Points 1 or 2. Middle row: Heatmaps computed using DC-Distance based on Wasserstein-Distance with min-Points 5. Bottom row: Heatmaps computed using DC-Distance based on Wasserstein-Distance with min-Points 8.

4.4. RQ2: Performance Evaluation Against other Methods

4.4.1. Research Question 2: Comparative Evaluation of Wasserstein Variants

In the first part of this research question, the proposed Wasserstein-based DC distance was compared against a broad set of baseline methods, ranging from handcrafted descriptors to pre-trained embeddings. This established its robustness and competitiveness in realistic clustering and tracking scenarios.

In the second part, we move from broad comparisons to a more targeted analysis. Here, the proposed Wasserstein formulation is directly contrasted with the Multi-Scale Sliced Wasserstein Distance (MS-SWD), both as a base metric and within the DC-Distance framework. This step allows us to disentangle two aspects: (1) whether the proposed

Wasserstein distance itself provides advantages over MS-SWD, and (2) whether these differences persist or change when combined with clustering based on DC-Distance with the respective metric as a base. Together, this focused comparison complements the baseline study and provides a deeper understanding of the role of different Wasserstein variants in density-connectivity clustering.

4.4.2. Comparison of Wasserstein-DC Distance with Established Clustering Approaches

To evaluate the effectiveness of the proposed Wasserstein-based density-connectivity distance, we compare it against a set of baseline methods on multiple datasets. For this comparison, we focus on the two previously used subsets from the KITTI-benchmark and the subset of the COIL-100 dataset. In contrast, the synthetic dataset is deliberately excluded, since its highly regular and simplified structures would not provide a meaningful reference for assessing clustering performance in realistic scenarios.

Importantly, we restrict our comparison to methods that do not require task-specific training. Deep learning–based trackers or clustering approaches are excluded, as they would introduce additional learning overhead and are not representative baselines for an unsupervised, training-free setting. Instead, the chosen baselines cover both handcrafted descriptors and pre-trained but fixed feature embeddings, ensuring fair and transparent conditions.

All images are downscaled to two resolutions, 32×32 pixels and 4×4 pixels. This choice serves two purposes: First, it ensures computational efficiency and allows us to apply a broad range of methods under identical conditions. Second, it provides insight into the robustness of different approaches under severe information constraints. Such low-resolution setups also reflect realistic constraints encountered in practical scenarios such as video surveillance, wildlife monitoring, or other resource-limited environments.

The baseline methods span both classical feature- and similarity-based techniques (HOG, color histograms, perceptual hashing, and structural similarity measures) as well as modern pre-trained feature embeddings (ImageNet-based CNN features and DINOv2 representations). Together, these baselines establish a broad and complementary reference frame against which the proposed Wasserstein DC distance can be systematically evaluated.

To apply clustering based on Wasserstein DC-Distances, the pairwise distance matrix $D = [d_{ij}]$ is converted into an affinity matrix A using a Gaussian RBF-Kernel. The affinity is defined as

$$A_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right),$$

where σ is set to the median of all non-zero distances in D. This mapping assigns high similarity values to small distances and exponentially suppresses large distances, resulting in a dense affinity matrix that reflects the neighborhood structure between images. Spectral clustering is then applied to A, as it operates directly on pairwise similarities and does not require an underlying Euclidean feature space, making it well suited for Wasserstein-based distances.

Baseline Methods

The comparison includes a set of lightweight and training-free baselines, ranging from classical feature-based descriptors to pre-trained embeddings. These methods provide complementary perspectives and allow for a transparent evaluation of the Wasserstein-based approach.

Histogram of Oriented Gradients (HOG). HOG descriptors capture local edge orientations and are widely used in object detection tasks [DT05]. They provide a compact representation of shape information while being computationally efficient. Each image is converted into a numerical feature vector by concatenating local gradient histograms, which lie in a well-defined Euclidean space. Because this representation is suited for centroid-based partitioning, we apply K-Means clustering to group the resulting feature vectors.

Color Histograms. Color distributions have long been employed for image retrieval and recognition [SB91]. They encode global appearance and are simple yet effective for distinguishing images based on dominant color composition. For each image, a global color histogram is extracted, resulting in a numerical feature vector. Since these vectors inhabit an Euclidean feature space, K-Means is used to partition them into clusters.

Perceptual Hashing (P-Hash). Perceptual hashing methods generate compact hash codes that preserve visual similarity. P-Hash in particular has been shown to be robust against minor geometric or photometric changes [Zau10]. Each image is converted into a fixed-length binary hash (e.g., 64 bits), which captures coarse visual structure. Because these binary representations do not reside in an Euclidean space, clustering is performed using spectral clustering on an affinity matrix derived from pairwise Hamming distances between hashes.

Structural Similarity (SSIM). The Structural Similarity Index (SSIM) measures similarity in terms of luminance, contrast, and structure [WBSS04]. We evaluate both grayscale SSIM and its multiscale extension [WSB03] to capture similarity at different levels of detail. Unlike feature-based methods, SSIM does not produce embeddings for individual images; instead, it yields scalar similarity scores for each image pair, forming a similarity matrix over the dataset. Since this representation is inherently pairwise, spectral clustering is applied directly to this matrix.

ImageNet ViT Embeddings. For this baseline we employ embeddings extracted from a Vision Transformer (ViT-B/16) [DBK+20] pre-trained on the ImageNet-1K benchmark

[DDS+09]. The model is used in a frozen manner, i.e., without any task-specific fine-tuning, providing a strong and widely used representation for evaluating clustering performance. Each image is transformed into a high-dimensional feature vector. Because these embeddings define a metric space where Euclidean distance reflects semantic similarity, K-Means is applied to cluster them.

DINOv2 Features. DINOv2 is a recent self-supervised vision transformer that produces robust and transferable visual representations [ODM+23]. Similar to ImageNet features, each image is mapped to a high-dimensional feature vector. Since these embeddings reside in a continuous vector space, K-Means is used to partition the feature space into clusters.

Experimental Comparison

Method	Dataset	ARI	NMI	DISCO
DC-Distance (Wasserstein) + Spectral Clustering	Coil-100	**1.000**	**1.000**	0.752
	Kitti1	0.916	0.955	**0.433**
	Kitti2	0.808	0.841	0.427
HOG & Color + K-means	Coil-100	0.842	0.918	0.262
	Kitti1	0.926	**0.959**	0.275
	Kitti2	**1.000**	**1.000**	0.196
P-Hash + Spectral Clustering	Coil-100	0.771	0.866	0.387
	Kitti1	0.914	0.886	0.189
	Kitti2	0.808	0.841	0.143
SSIM GreyScale + Spectral Clustering	Coil-100	0.874	0.934	0.698
	Kitti1	0.764	0.892	0.364
	Kitti2	**1.000**	**1.000**	0.498
SSIM MultiScale + Spectral Clustering	Coil-100	**1.000**	**1.000**	**0.834**
	Kitti1	0.799	0.921	0.420
	Kitti2	**1.000**	**1.000**	**0.523**
Imagenet Feature Embeddings + K-means	Coil-100	**1.000**	**1.000**	0.756
	Kitti1	**0.938**	0.958	0.238
	Kitti2	0.898	0.899	0.292
DINOV2 Feature Embeddings + K-means	Coil-100	0.898	0.942	0.583
	Kitti1	0.858	0.927	0.265
	Kitti2	1.000	1.000	0.405

Table 4.8.: Evaluation of clustering performance across different methods on images with resolution 32×32 pixels. Highlighted numbers indicate the best performance for each dataset across all methods.

Tables 4.8 and 4.9 summarize the clustering performance of the proposed Wasserstein-based DC distance in comparison to the selected baseline methods on the COIL-100 subset and two KITTI subsets, evaluated at resolutions of 32×32 and 4×4 pixels.

Method	Dataset	ARI	NMI	DISCO
DC-Distance (Wasserstein) + Spectral Clustering	Coil-100	**1.000**	**1.000**	**0.857**
	Kitti1	0.860	0.931	**0.552**
	Kitti2	0.808	0.841	0.590
HOG & Color + K-means	Coil-100	0.600	0.784	0.132
	Kitti1	0.791	0.894	0.275
	Kitti2	0.801	0.800	0.189
P-Hash + Spectral Clustering	Coil-100	0.711	0.840	0.117
	Kitti1	0.590	0.766	0.083
	Kitti2	0.667	0.764	0.159
SSIM GreyScale + Spectral Clustering	Coil-100	0.780	0.884	0.599
	Kitti1	0.865	0.932	0.537
	Kitti2	**1.000**	**1.000**	0.694
SSIM MultiScale + Spectral Clustering	Coil-100	0.848	0.923	0.683
	Kitti1	**0.920**	**0.961**	0.550
	Kitti2	**1.000**	**1.000**	**0.702**
Imagenet Feature Embeddings + K-means	Coil-100	0.962	0.977	0.796
	Kitti1	0.773	0.872	0.271
	Kitti2	0.898	0.899	0.292
DINOV2 Feature Embeddings + K-means	Coil-100	0.849	0.920	0.611
	Kitti1	0.758	0.860	0.203
	Kitti2	**1.000**	**1.000**	0.278

Table 4.9.: Evaluation of clustering performance across different methods on images with resolution 4×4 pixels. Highlighted numbers indicate the best performance for each dataset across all methods.

COIL-100 Results As the COIL-100 subset provides static objects imaged against a clean black background while being rotated on a turntable. This creates consistent object appearances, where the global structure remains stable and only the viewing angle changes. In contrast to KITTI, where object crops often include varying parts of the surrounding background, COIL-100 objects are isolated, and the background remains constant. This favors methods that are sensitive to pixel distributions or color information, since no background clutter interferes with the feature representations. As a result, almost all methods achieve relatively high scores on COIL-100, especially at 32×32 resolution. At 4×4 resolution, however, handcrafted descriptors and hash-based approaches (e.g., P-Hash) lose discriminative power, since color and edge information is heavily compressed. In contrast, SSIM (particularly its multiscale variant) and the DC-Dist based on Wasserstein remain effective, as they are better suited to capture global structural similarities. Pre-trained embeddings from ImageNet and DINOv2 also maintain strong results here, benefitting from their ability to represent object shape and appearance even under resolution constraints.

KITTI Results The KITTI crops present a more challenging scenario.As objects are extracted from real driving sequences, and although they are resized to a fixed target dimension, their apparent size and detail level varies depending on motion, distance, and occlusions. This resizing effect can distort features and particularly impacts methods that rely on consistent low-level patterns. HOG+Color and P-Hash degrade considerably, which is likely due to their reliance on local edges and global appearance distributions. These characteristics may make them more sensitive to scale variations, background changes, and partial occlusions observed in the KITTI crops. SSIM performs better, as it evaluates relative structural similarity rather than absolute pixel distributions, making it more robust against resizing artifacts. The multiscale SSIM variant achieves near-perfect performance on KITTI2 at both resolutions, highlighting the benefit of capturing similarity at multiple spatial scales.

Pre-trained embeddings show mixed behavior on KITTI. ImageNet features retain strong discriminative capacity on KITTI1 but perform less consistently on KITTI2, likely due to domain mismatch between natural image training data and the specific visual conditions in KITTI. DINOv2 embeddings, although robust on COIL-100, are more affected by the limited detail in KITTI crops, especially at 4×4 resolution, where the transformer-based embeddings cannot fully leverage their representational richness.

Wasserstein DC Distance Across all scenarios, the Wasserstein-based DC distance demonstrates high robustness and flexibility. On COIL-100, it achieves perfect clustering, consistent with the clean structure of the dataset. More importantly, it remains stable on KITTI sequences even under extreme downscaling. Unlike handcrafted descriptors, it does not rely on fixed low-level features, and unlike pre-trained embeddings, it does not suffer from domain mismatch.

By combining optimal transport with density-connectivity clustering, it effectively captures the relative distribution of pixel intensities and structural relationships, making it resilient to resizing effects and noise.

Wasserstein DC Distance Across all scenarios, the Wasserstein-based DC distance demonstrates high robustness and flexibility. On COIL-100, it achieves perfect clustering, consistent with the clean structure of the dataset. More importantly, it remains stable on KITTI sequences even under extreme downscaling. Unlike handcrafted descriptors, it does not rely on fixed low-level features, and unlike pre-trained embeddings, it does not suffer from domain mismatch.

By combining optimal transport with density-connectivity clustering, the method proves resilient to resizing effects and noise, which explains its consistently strong performance in challenging real-world conditions.

Summary The experimental results underline three key insights: (1) simple handcrafted descriptors fail under extreme compression and scale variation, (2) SSIM-based measures and pre-trained embeddings remain competitive but are dataset-dependent, and (3) the Wasserstein DC distance achieves consistently superior or at least comparable performance, demonstrating its strength as a robust, training-free approach to clustering in object tracking scenarios with real-world constraints such as low resolution, noise, and scale variation.

4.4.3. Is Wasserstein or Multiscale Sliced Wasserstein Distance (MS-SWD) the better standalone Metric for Clustering?

MinPoints	Dataset	Sil-Score	Dunn	Sep-Score
Wasserstein	Synth	0.563	0.660	2.3
	COIL	0.725	0.685	7.2
	KITTI1	0.481	0.169	3.0
	KITTI2	0.464	0.325	2.4
DC-Distance (Wasserstein)	Synth	**0.832**	**5.867**	**6.0**
	COIL	**0.854**	**2.364**	10.7
	KITTI1	**0.550**	**0.416**	2.6
	KITTI2	0.501	0.823	2.1
MS-SWD	Synth	0.578	0.666	2.4
	COIL	0.725	0.382	12.1
	KITTI1	0.411	0.095	**3.9**
	KITTI2	0.710	0.856	5.5
DC-Distance (MS-SWD)	Synth	0.725	2.022	3.6
	COIL	0.789	1.395	**15.2**
	KITTI1	0.429	0.179	2.7
	KITTI2	**0.790**	**2.601**	**6.9**

Table 4.10.: Evaluation of clustering performance across various distance variants.

Quantitative Analysis. We compare plain Wasserstein distance (averaged over RGB channels) to Multiscale Sliced Wasserstein Distance (MS-SWD) as standalone clustering metrics.

- **Synthetic:** MS-SWD marginally outperforms Wasserstein across all metrics (Silhouette-Score: 0.578 vs. 0.563, Dunn-Index: 0.666 vs. 0.660, Separation-Score: 2.4 vs. 2.3).
- **COIL:** Silhouette scores are identical (0.725), but MS-SWD achieves a much higher Separation-Score (12.1 vs. 7.2), suggesting better inter-cluster distinction.
- **KITTI1:** Wasserstein performs better in Separation-Score (3.0 vs. 3.9), but MS-SWD has a slightly higher Dunn Index (0.095 vs. 0.169).
- **KITTI2:** MS-SWD clearly outperforms across all three metrics (Silhouette-Score: 0.710 vs. 0.464, Dunn-Index: 0.856 vs. 0.325, Separation-Score: 5.5 vs. 2.4).

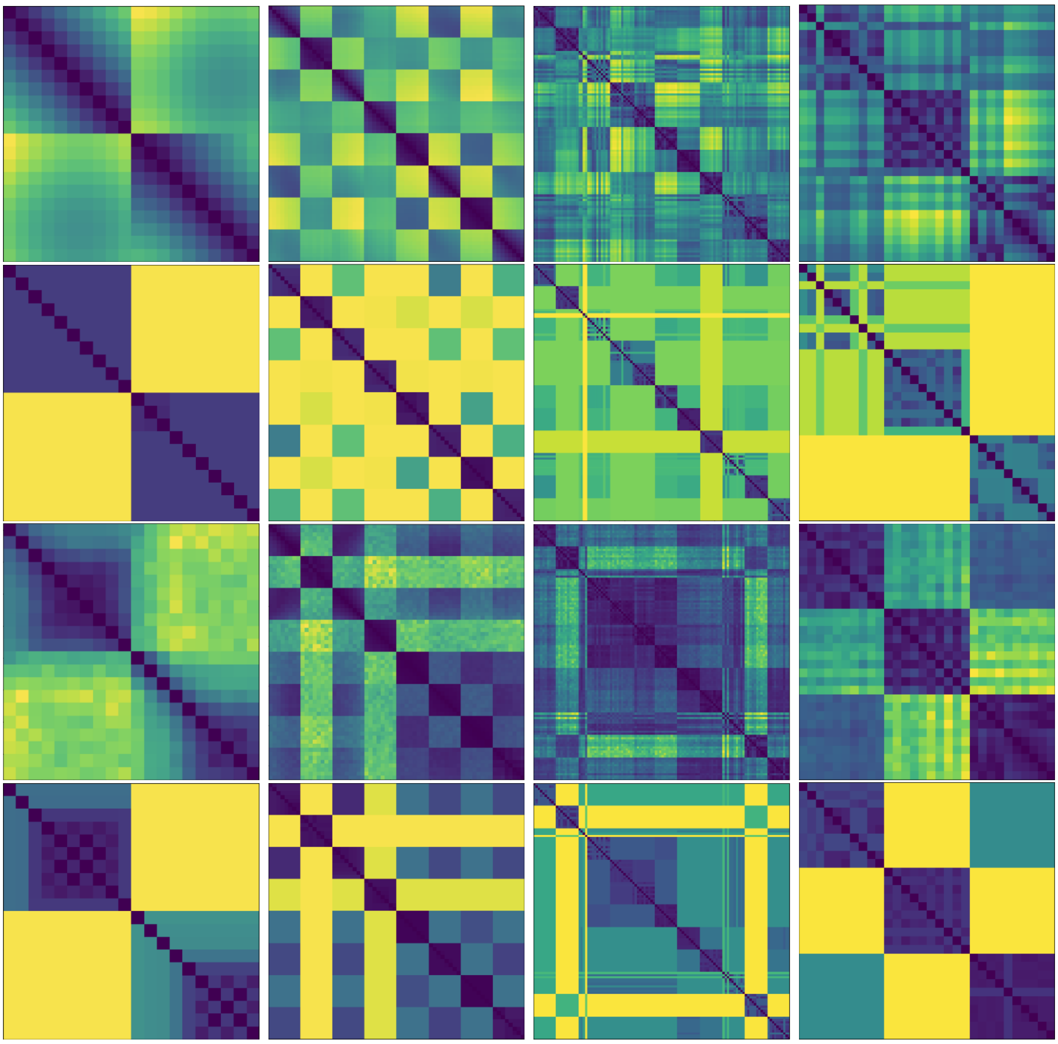

Figure 4.11.: Comparison of pairwise distance heatmaps between images across four data- sets: Synthetic, COIL, KITTI-Subset A, and KITTI-Subset B (left to right). Row 1: Heatmaps computed using Wasserstein-Distance. Row 2: Heatmaps computed using DC-Distance based on Wasserstein-Distance. Row 3: Heatmaps computed using MS-SWD. Row 4: Heatmaps computed using DC-Distance based on MS-SWD.

Qualitative Insights from Distance Heatmaps. On the Synthetic dataset, plain MS-SWD fails to capture temporal smoothness between frames. The heatmap lacks a clear block-diagonal structure and instead shows irregular transitions between frames that should be sequentially similar. This is likely due to the patch-based random projection in MS-SWD, which is less sensitive to small spatial shifts which is a key factor in synthetic motion. In contrast, plain Wasserstein distance shows much stronger continuity and block-wise structure across sequential frames, indicating better temporal cohesion. On the COIL dataset, both distances struggle, but MS-SWD appears slightly noisier. Its inability to align well with high-saturation RGB variations may lead to misgroupings. On KITTI, MS-SWD outperforms plain Wasserstein, with cleaner inter-cluster margins and more contrast between scenes. This leads to the assumption that MS-SWD's design is more effective on natural, diverse image sequences than on minimal synthetic data.

4.4.4. Is Wasserstein or MS-SWD a Better Base Metric for DC-Distance Clustering?

Quantitative Analysis. We compare DC-Distance computed over Wasserstein vs. MS-SWD as base distances.

- **Synthetic:** DC-Distance (Wasserstein) performs significantly better (Silhouette-Score: 0.832 vs. 0.725, Dunn-Index: 5.867 vs. 2.922, Separation-Score: 6.0 vs. 3.6), indicating stronger separation and internal coherence.
- **COIL:** Wasserstein wins in Silhouette-Score and Dunn-Index (0.854 vs. 0.789 and 2.364 vs. 1.395), but MS-SWD has a notably higher Separation-Score (15.2 vs. 10.7), suggesting sharper inter-cluster contrast.
- **KITTI1:** Results are mixed. Wasserstein has a better Silhouette-Score (0.550 vs. 0.429), but MS-SWD achieves a better Dunn-Index (1.179 vs. 0.416), hinting at tighter, denser clusters.
- **KITTI2:** MS-SWD again outperforms in all metrics (Silhouette-Score: 0.790 vs. 0.501, Dunn-Index: 2.601 vs. 0.823, Separation-Score: 6.9 vs. 2.1).

Qualitative Insights from Distance Heatmaps. On the Synthetic dataset, DC-Distance (Wasserstein) creates strong, clean block structures, accurately capturing the continuity and segmentation between frame sequences. DC-Distance (MS-SWD) produces sharper segmentation than MS-SWD alone, but the intra-cluster consistency is visibly weaker than with Wasserstein. On COIL, both DC variants show improvement over their standalone counterparts, but DC-Distance (Wasserstein) still yields more distinct clusters. KITTI, however, highlights DC-Distance (MS-SWD)'s strengths: it produces extremely sharp transitions between scenes, with strong inter-cluster margins. Yet even here, DC-Distance (Wasserstein) seems to achieve better internal coherence, confirming its suitability for dense sequential data where preserving local continuity is essential.

4.5. RQ3: Framework for Object Tracking Based on Wasserstein

To what extent can clustering with DC-Distance and Wasserstein distance recover scene structure and enable object tracking from unordered image sequences?

4.5.1. Motivation and Evaluation Strategy

This research examines whether the combination of DC-Distance and Wasserstein distance can be used to group unordered frames into coherent scenes and track object motion by analyzing the optimal transport plans between sequentially ordered images. The approach consists of three steps:

1. clustering images into scenes using DC-Distance based on Wasserstein distance,
2. reconstructing the temporal order of frames within each scene by arranging them so that consecutive frames have minimal pairwise Wasserstein distance, and
3. extracting object motion from the transport plans between adjacent frames.

The core insight is that Wasserstein transport captures pixel-level mass displacement, which can be interpreted as the movement of objects or regions across time. This makes it particularly suitable for identifying object trajectories without relying on explicit object detection or feature tracking, especially in cases where annotations or pretrained models are unavailable.

In this part of the study, we restrict ourselves to using the averaged Wasserstein distance across RGB channels. Although methods such as Multiscale Sliced Wasserstein Distance (MS-SWD) or barycentric fusion showed competitive performance in clustering tasks (see RQ1 and RQ2), they do not produce transport plans, which are essential for the motion extraction step. Using them would therefore require recomputing standard Wasserstein distances additionally, leading to a significant computational overhead. Since optimal transport computations already represent a major bottleneck in terms of memory and runtime, we opt for a unified approach based solely on Wasserstein averaging.

We evaluate the full pipeline using the following criteria:

- **Scene-Clustering:** Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Disco-Score are used for clustering-evaluation.
- **Temporal ordering:** Kendall's Tau rank correlation (measures how similar two sequences are based on pairwise agreements) and Levenshtein distance (counts the minimum edits needed to turn one sequence into another) with ground-truth frame order.
- **Tracking performance:** Evaluation of extracted motion paths using Intersection-over-Union (IoU) over time or trajectory similarity to ground truth.
- **Transport visualization:** Qualitative inspection of displacement patterns between adjacent frames to assess semantic plausibility of motion.

4.5.2. Sub-Questions and Experimental Variants

Can this framework successfully recover scenes and track object motion in:

1. our synthetic testbed with constant mass and high foreground contrast, and
2. real-world data such as the SVIA sperm microscopy dataset and the MammAlps Wildlife documentary subset?

These datasets differ significantly in appearance, structure, and signal-to-noise ratio. The synthetic dataset provides a controlled environment with minimal background and clean object motion, making it ideal for verifying theoretical assumptions. The real datasets test the robustness of the pipeline under natural variations such as lighting changes, motion blur, and subtle object appearance.

4.5.3. Evaluation and Results

POC on synthetic Data

To evaluate the proposed tracking framework under idealized conditions, we applied it to our synthetic dataset. Each scene in this dataset consists of a series of 10 frames, depicting 1-2 geometric shapes moving on a black background. Before clustering, all images were shuffled to remove temporal information.

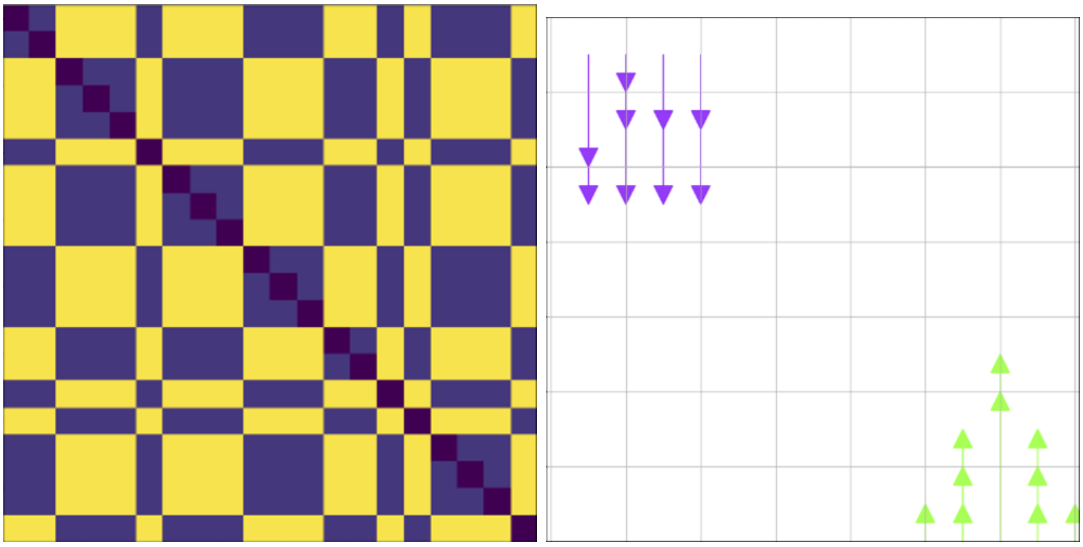

Figure 4.12.: Both figures are based on exemplary frames of the synthetic dataset. Left: Heatmap computed using DC-Distance based on Wasserstein-Distance (Frames from one Scene not sorted block-diagonally). Right: Transport-Movement-plan of one exemplary frame-comparison.

Scene Clustering with Wasserstein-Based DC-Distance. Using density-connectivity distance with Wasserstein as base metric, we computed the pairwise distance matrix between all frames. As shown in the heatmap, the resulting distances formed clearly separated blocks corresponding to scenes, with no overlap or ambiguous transitions. We then applied greedy 2-approximate k-center clustering to assign images to clusters. All images were assigned to their correct original scene (ARI = 1.0, NMI = 1.0), demonstrating that Wasserstein-based connectivity captures the underlying sequence semantics even in unordered data. The DISCO-score of 0.338 further confirms the strong alignment between distance structure and scene labels. The residual inter-scene overlap reflected in this value is likely attributable to the identical background shared across scenes, which reduces overall inter-cluster contrast despite perfect scene assignments.

Temporal Ordering Within Clusters. Once scenes were recovered, we applied our greedy ordering algorithm based on pairwise Wasserstein distances within each cluster. Starting from a random image, the algorithm incrementally selected the frame with the shortest distance to either endpoint of the current sequence. This process yielded a perfect reordering of the original sequences. Quantitatively, this corresponds to a Kendall's Tau correlation of $\tau = 1.0$ and a Levenshtein distance of 0 between the predicted and ground-truth frame order in every cluster.

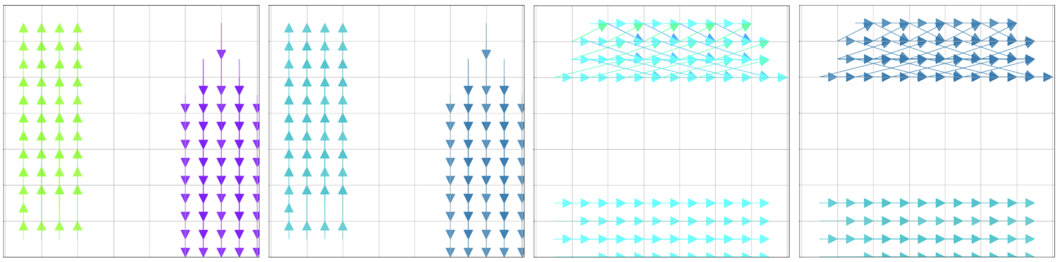

Figure 4.13.: Figures are based on exemplary frames of the synthetic dataset. From Left to Right: Unclustered combined Transport-plans from Scene 1 (color depicting direction).; Clustered combined Transport-plans from Scene 1.; Unclustered combined Transport-plans from Scene 2 (color depicting direction).; Clustered combined Transport-plans from Scene 2.

Object Tracking via Transport Plans. Transport plans between adjacent images in the recovered order provided direct access to pixel-level displacement. For every frame pair, we computed the optimal transport plan across all three RGB channels and averaged them. Figure 4.11 shows the transport plan between two adjacent frames, visualized as mass displacement vectors. The directions of these vectors uniformly point in the object's true direction of movement, despite no supervision or explicit motion model, validating the semantic alignment of transport with visual motion.

To assess overall tracking consistency, we combined the transport vectors from all adjacent image pairs in each cluster. This yielded dense displacement maps 4.13, clearly visualizing object trajectories across time.

We clustered the filtered transport vectors using HDBSCAN, applied to the 4D representation of each arrow consisting of its start position and displacement vector, i.e.

$$\mathbf{v}_k = \big(x_k,\, y_k,\, \Delta x_k,\, \Delta y_k\big),$$

where (x_k, y_k) denotes the source location and $(\Delta x_k, \Delta y_k)$ the corresponding transport displacement. This representation groups arrows that originate from similar positions and point in similar directions, effectively identifying coherent motion patterns.

Each object's motion formed a distinct group, with no false positives or intermixing.

Sperm Dataset

Scene Clustering with Wasserstein-Based DC-Distance. Clustering on the SVIA sperm tracking dataset using DC-Distance based on Wasserstein successfully produced a perfect separation of scenes (ARI = 1.0, NMI = 1.0). The resulting clustering achieved a DISCO score of 0.331, indicating solid alignment between the induced distance structure and the clustered scene grouping, especially in conjunction with the perfect ARI and NMI values. While this value reflects better-than-random separability, it also suggests that substantial inter-scene overlap remains, likely due to the inherently low variation in grayscale intensity and the small spatial extent of moving objects. Nevertheless, this improvement allowed the framework to proceed with subsequent sequencing and tracking steps on this dataset.

Temporal Ordering Within Clusters. Using the greedy ordering algorithm based on pairwise Wasserstein distances within the two original clusters yielded poor results. The distance information again proved insufficient to reflect the temporal structure, resulting in a Kendall's Tau of -0.333 with a non-significant p-value of 0.216 and a relative Levenshtein-distance of $0,75$. This suggests that Wasserstein distances were too uniform or noisy to meaningfully capture frame order in this dataset. This may be partly due to the low color variation in the data, as the images are in black and white, and also possibly due to the failure of the ordering process itself. We thus reverted to the original temporal order for the motion tracking step.

Object Tracking via Transport Plans. We computed the transport plans between sequentially ordered frames. However, due to intensity normalization across frames, the same object might carry slightly different mass in each frame. This caused high-frequency noise in the resulting transport vectors, as tiny intensity mismatches were balanced via many small, scattered mass flows. To recover meaningful object motion, we applied aggressive thresholding: discarding 97% of the smallest mass flows. The threshold was selected via a grid search over the range $[90\%, 99.9\%]$, balancing noise suppression against preservation of coherent motion. As a selection criterion, we evaluated how well the resulting transport vectors aligned with visually apparent object displacements across frames. After filtering, we observed interpretable mass displacement patterns, clearly capturing

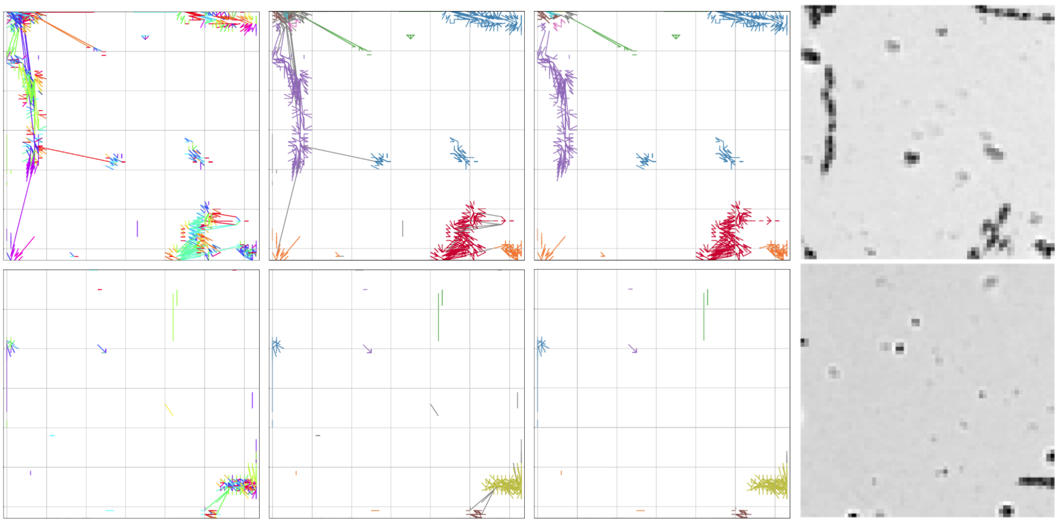

Figure 4.14.: Figures are based on exemplary frames of the SVIA-dataset. Top: Scene 1; Bottom: Scene 2; From Left to Right: Unclustered combined Transport-plans(color depicting direction).; Clustered combined Transport-plans.; Clustered combined Transport-plans with filtering.; Original frames layered on top of each other.

the dominant sperm movements across frames (see Figure 4.14). HDBSCAN was able to accurately cluster most sperm trajectories. Only a few ambiguous paths, particularly where trajectories intersected, were misclustered. Additionally, it was noticeable that most displacement vectors pointed radially inward rather than consistently forward, a side effect of Wasserstein's optimization, which minimizes cost rather than directionality. Thus, while trajectory regions were correctly captured, directional information was often lost.

Wildlife Trap Cameras

Scene Clustering with Wasserstein-Based DC-Distance. The subset consisted of three scenes: two taken from the same camera setup (same background and conditions) and one from a different environment. Using Wasserstein-based DC-Distance, the clustering resulted in clearly distinguishable clusters that matched the underlying scenes well. The corresponding heatmap exhibits strong block structure for each cluster. However, inter-cluster distances between the two scenes captured by the same camera were notably smaller than those involving the third scene, indicating a clear but weaker separation when visual backgrounds are similar. This confirms that the method is capable of identifying semantic structure even in real-world conditions, while also being sensitive to background similarity. The final clustering using k-center based on the computed DC-Distance matrix yielded a flawless assignment (ARI $= 1.0$, NMI $= 1.0$), with all frames correctly grouped according to their respective scenes. The DISCO-score of 0.871 further supports this result, reflecting a very high alignment between the learned distance structure and the ground-truth scene labels. The slight deviation from a perfect score likely stems from the reduced inter-cluster margins as two of the 3 scenes had the same background, which introduces overlap in the distance distribution despite perfect cluster assignments.

Temporal Ordering Within Clusters. The greedy sequencing algorithm achieved a moderate Kendall's Tau of 0.378 ($p = 0.156$) and a relative Levenshtein distance of 0.27. Upon closer inspection, most of the deviation originated from Scene 2, where strong background movement (such as grass swaying in the wind) introduced noise into the Wasserstein distance calculations. Scene 1 was ordered perfectly, and Scene 3 contained only one out-of-place frame, which caused a global shift in the sequence and disproportionately impacted the Kendall's Tau score. Despite this, the intra-scene ordering was visually coherent. To ensure consistent downstream evaluation, the ground-truth frame order was used in the following analysis.

Object Tracking via Transport Plans. Using the ground-truth sequential frame order, transport plans between adjacent frames were combined to visualize motion trajectories. As in the sperm dataset, small intensity fluctuations from frame-to-frame normalization and background motion (e.g., vegetation) introduced numerous low-mass transport vectors. These were filtered by applying a threshold on transport mass per scene, adjusted via a grid search in the range of $[90\%, 99.9\%]$ in increments of 0.1%. For each scene, the optimal threshold was selected based on the clearest separation between meaningful object motion and background noise, determined through visual inspection and qualitative trajectory coherence.

The resulting displacement maps successfully highlighted regions of consistent object movement, notably animal motion (e.g., deer) across frames. HDBSCAN was applied to cluster these mass flow vectors, yielding good results overall. As expected, errors occurred when multiple animals crossed paths in close spatial proximity, leading to occasional cluster merging. Additionally, the transport vectors remained radial rather than strictly directional, again due to the optimization behavior of the Wasserstein transport which prioritizes cost-efficient mass movement over temporal continuity.

An interesting observation occurred in Scene 1, where a large moving object (e.g., a deer) covered extensive regions of the image. Due to its uniform color and minimal pixel-level change, not all motion was captured by the transport vectors, which highlights a known limitation of purely intensity-based mass movement tracking [PC$^+$19]. Nevertheless, across all scenes, spatial motion areas could be detected accurately, as illustrated in Figure 4.15.

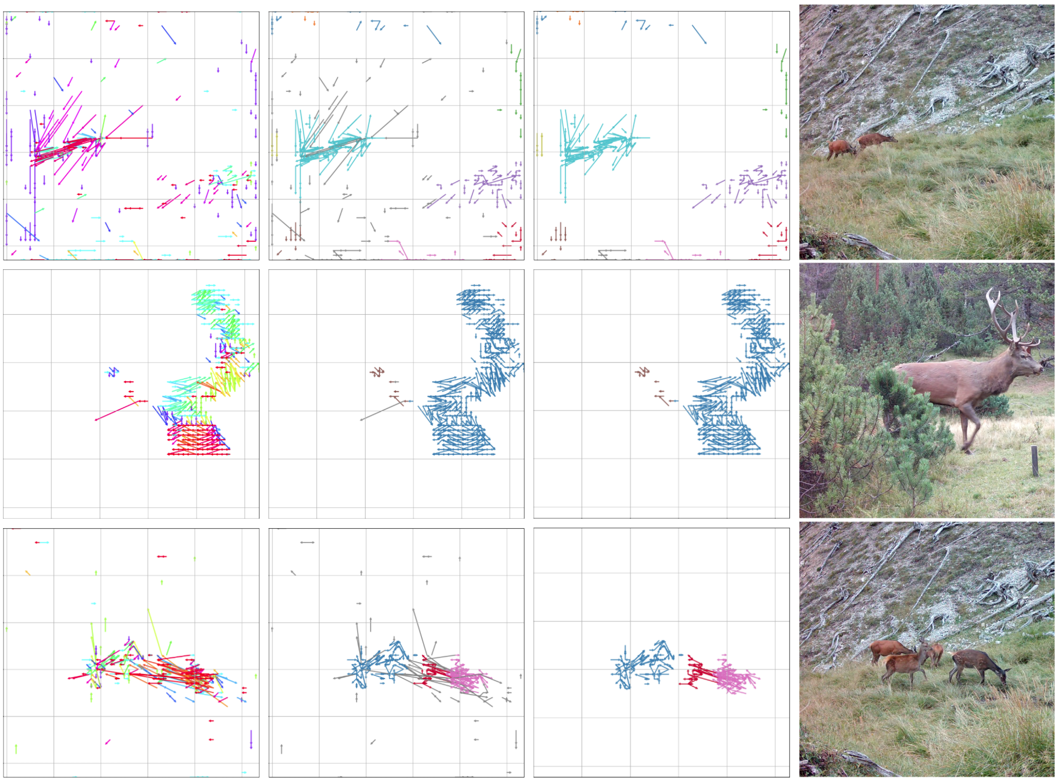

Figure 4.15.: Figures are based on exemplary frames of the MammAlps-dataset. Top: Scene 1; Middle: Scene 2; Bottom: Scene 3; Column 1: Unclustered combined Transport-plans (color depicting direction); Column 2: Clustered combined Transport-plans; Column 3: Clustered combined Transport-plans with filtering; Column 4: Exemplary frame of the scene.

5. Discussion

5.1. General Interpretation of Results

Wasserstein distance consistently outperforms Euclidean distance as a base metric in clustering image data. Unlike pixel-wise Euclidean comparisons, it captures spatial mass displacement and structure, which is critical in tasks involving object motion, deformation, or partial overlap. Especially on synthetic sequences with spatial shifts, Wasserstein maintains a more semantically aligned similarity structure, justifying its central role throughout this thesis.

Among the tested mass-alignment strategies, normalization showed the most consistent and stable performance across datasets. While max-cost bucketing yielded strong separation on clean, synthetic data, it collapsed under slight appearance changes, such as lighting fluctuations (e.g., Synth-B). Border-aware bucketing introduced valuable flexibility (e.g., in cases of object entry at frame borders), but also introduced instability when edge movement dominated the frames. These results justify using normalization as the default mechanism in mixed or noisy settings.

Averaging Wasserstein distances across color channels (RGB) emerges as a simple and robust solution. In contrast, barycentric fusion struggled on low-dimensional or sparse-color data (e.g., synthetic), likely due to the strong channel-wise color dominance and low diversity. Sliced Wasserstein Distance (SWD), despite its approximate nature, delivered surprisingly good results across all datasets and proved efficient and scalable, making it a strong alternative when exact transport plans are not required. However, for applications like tracking (requiring full transport plans), only full-channel averaging with Wasserstein was viable.

CIELAB's perceptual uniformity does not always translate to better clustering performance. It failed to distinguish synthetic clusters, likely due to the fact that synthetic frames are constructed directly in RGB primaries, which CIELAB distorts through nonlinear mapping. On real-world datasets such as COIL and KITTI, however, both RGB and CIELAB yielded comparable clustering quality, provided the CIELAB channels were normalized to a comparable range. RGB remains the practical default for graphics-derived content, while CIELAB is viable for perceptual domains under proper preprocessing.

While Wasserstein distance alone performs well in perceptual comparison, its integration into the DC-Distance framework significantly improves separability and reduces sensitivity to noise. The DC-Distance (Wasserstein) variant yields sharper intra-cluster consistency and more stable scene groupings, particularly in temporally coherent data. DC-Distance (MS-SWD) also performed well in real-world scenes, offering sharper contrast between clusters, but it lacked the continuity and internal structure needed for sequencing or transport-based motion analysis.

The *minPts* parameter in DC-Distance strongly influences clustering behavior. Values of 1-2 were most effective for image sequences, as they exploit the inherent temporal adjacency between frames (i.e., frame t is always similar to $t-1$ and $t+1$). This effectively creates natural single-linkage chains. As *minPts* increases, these fine-grained links are overridden by denser connectivity assumptions, leading to blocky heatmaps, cluster merging, and degraded separability. Thus, minimal *minPts* settings are preferred in sequence-based applications.

Wasserstein DC-Distance exhibited a stable performance across all datasets and settings copared to baseline methods. While some achieved higher scores in individual cases, Wasserstein DC-Distance remained consistently strong and reliable, maintaining high clustering quality even in challenging scenarios with limited detail or background variation. This balance between robustness and accuracy makes it a dependable reference point among the evaluated methods.

Plain MS-SWD outperforms plain Wasserstein on complex and natural datasets like KITTI, especially when dealing with large spatial and photometric transformations. Its multiscale patch-based comparison provides robustness in shifting scenes. However, Wasserstein performs better in synthetic and structured domains due to its pixel-level grounding and interpretability. When embedded into DC-Distance, Wasserstein maintains strong internal consistency and scene progression, whereas MS-SWD sharpens inter-cluster contrast. Each has its merits depending on dataset characteristics.

The synthetic dataset validated every component of the framework: scene clustering with DC-Distance was flawless; greedy sequencing reproduced original order with zero error (Kendall's $\tau = 1.0$); and the combined transport plans clearly outlined object motion. HDBSCAN clustered movement vectors perfectly. This validates the complete pipeline under ideal conditions.

Despite being grayscale and simple, the SVIA sperm dataset produced usable clustering results after applying contrast enhancement. Sequencing, however, still failed due to low inter-frame variation. Tracking worked after strong filtering (removing 97% of low-mass vectors), but directional information was lost as radial mass movement dominated in Wasserstein transport. This highlights challenges with grayscale and intensity-unstable sequences, where even with improved clustering, sequencing and motion direction recovery remain difficult.

The Wildlife Trap Camera dataset produced well-separated scene clusters. Sequencing was moderate overall (Kendall's $\tau = 0.378$, $p = 0.156$), with strong results in two scenes and one problematic due to moving grass. Tracking required per-scene threshold tuning to suppress background noise. Despite these challenges, object movement was clearly captured in transport maps. The system successfully detected and clustered multiple animals, with occasional confusion when trajectories overlapped or occlusions occurred.

Across datasets, the proposed framework, built on Wasserstein-based DC-Distance, greedy sequencing, and transport-based tracking, proves conceptually sound and robust in most practical scenarios. Challenges remain in grayscale and highly variable scenes, but core capabilities like unsupervised scene grouping and motion subspace extraction were validated.

5.2. Methodological Strengths

One of the core methodological contributions of this work is the integration of Wasserstein distance into the density-connectivity framework. By replacing the base metric in DC-Distance with Wasserstein, the method gains sensitivity to both pixel intensity distributions and spatial displacement, improving semantic clustering, particularly for image sequences and structurally similar data. This allows the framework to maintain robust intra-cluster consistency while clearly separating scene transitions, even in low-signal or noisy environments.

The use of **DC-Distance** brings significant advantages over traditional clustering approaches. It integrates mutual reachability with local density constraints, which helps mitigate noise and suppress the chaining effect that often undermines single-linkage methods. Especially in temporally structured data, it preserves meaningful relationships by leveraging natural adjacency, leading to more coherent and interpretable clusters.

The proposed **greedy sequencing algorithm** successfully reconstructs frame order from unordered sets, given that Wasserstein distance reflects meaningful inter-frame similarity if the underlying Wasserstein-distances are meaningful. On the synthetic dataset, this method achieved perfect recovery of the ground-truth order. Its simplicity and effectiveness, without requiring supervised learning or temporal metadata, make it a practical tool for reordering frames in applications such as microscopy or wildlife monitoring.

Furthermore, clustering transport vectors using **HDBSCAN** allows for the extraction of object motion trajectories from purely unsupervised input. The clustering of transport plans revealed coherent sub-trajectories corresponding to individual moving entities, such as sperms or animals in natural scenes. Even without labels or tracking annotations, the movement subspaces were often correctly segmented, illustrating the power of motion-based representation for clustering.

The **visual analysis of transport plans** adds a unique interpretability layer to the system. By inspecting the displacement maps between frames, users can intuitively validate the presence and directionality of motion. These transport fields show how objects moved within the frame, offering an interpretable alternative to black-box tracking models.

Finally, the framework's modularity, using clustering, sequencing, and tracking as separable steps, increases flexibility and adaptability across domains. Each component contributes to the overall robustness, and their combination facilitates a general-purpose unsupervised pipeline for motion-centric image analysis.

5.3. Methodological Limitations

Despite its strengths, the proposed framework also exhibits several limitations that must be acknowledged.

First, the reliance on **mass normalization**, while necessary for comparing images with varying intensities, can lead to artifacts in *low-contrast grayscale data*. This was particularly evident in the SVIA dataset, where sperm scenes lacked sufficient pixel variance. After normalization, minimal intensity differences were exaggerated, resulting in unstable Wasserstein distances that failed to reflect true scene differences. This undermined both clustering and temporal ordering, indicating the need for adaptive or alternative normalization strategies in such contexts.

Second, although Wasserstein **transport plans capture the locality of pixel movement**, they do not preserve directional information in a consistent or interpretable manner. In several real-world datasets, especially Wildlife and SVIA, the resulting mass flow between adjacent frames was predominantly *radial* to the object path. This occurs because the optimal transport prioritizes cost minimization over vector consistency, and often redistributes mass through shortest local paths instead of along coherent motion lines. Consequently, directional tracking using raw transport vectors is unreliable, limiting the framework's utility for direction-sensitive tasks.

In addition, **minor variations in lighting, background activity, or scene composition** can produce disproportionately large effects in the transport plans. This was particularly noticeable in outdoor settings such as the Wildlife dataset, where subtle grass movements caused dense low-magnitude mass flows. These dominated the transport map and obscured meaningful object motion. As such, the method exhibits sensitivity to environmental noise and benefits significantly from post-filtering.

Another major drawback lies in the **computational overhead** of the approach. Computing full transport plans between all image pairs, especially when used for DC-Distance calculations, introduces significant time and memory demands. For large datasets or high-resolution frames, this becomes prohibitive. Optimizations such as sliced Wasserstein or Sinkhorn approximations may alleviate this issue [PC+19], but at the potential cost of transport fidelity and nor resulting transport-plan for the framework to use.

Lastly, the use of transport-based tracking necessitates **threshold tuning for mass filtering**, particularly when aggregating transport vectors across frames. The proportion of transport mass that needs to be retained to obtain meaningful motion trajectories varies substantially between scenes. This is because the distribution of transported mass depends on factors such as background dynamics and object salience. As observed in both SVIA and Wildlife datasets, setting a global cutoff leads to either noise inclusion or motion loss. Thus, the framework currently depends on manual or dataset-specific parameter tuning, limiting its automation and scalability.

5.4. Broader Implications

The findings of this thesis suggest some implications for image analysis in settings where labeled data, semantic understanding, or temporal metadata are unavailable.

First, the experiments demonstrate that **unsupervised clustering and tracking of image sequences is achievable without any semantic features, pretrained mod-**

els, or ground-truth ordering. By relying purely on Wasserstein-based comparisons and density-connectivity structure, the framework is able to group visually similar images, recover likely temporal orders, and extract plausible motion paths—solely from spatial pixel mass distributions. This makes it applicable to scientific, surveillance, medical or ecological domains where annotation is costly or infeasible, and semantic concepts may not be clearly defined.

Moreover, the use of **transport plans** as a representational tool provides a novel and interpretable lens into motion and structural shifts across frames. Unlike traditional optical flow or keypoint tracking, the transport vectors emerge directly from the underlying image distribution and do not require object segmentation or classification. This allows motion to be analyzed in a label-free, geometry-driven manner, exposing coherent displacement patterns even in scenes with partial occlusions or appearance variation.

In sum, this work points to the viability of **geometry-aware, distributional techniques** for low-supervision image understanding, and establishes a foundation for future unsupervised pipelines in complex temporal data.

5.5. Future Work

Several promising directions emerge from the findings and limitations of this study, which could further enhance the proposed framework's scalability, precision, and interpretability:

- **Hybrid Distance Strategies.** Future work could explore combining the efficiency and robustness of sliced Wasserstein or MS-SWD with the structural detail captured by full Wasserstein transport. For example, sliced-based clustering followed by full transport analysis could balance speed and semantic richness.

- **Global Sequencing Optimization.** While the greedy algorithm for intra-cluster ordering proved effective in structured datasets, its local nature may fail in noisier or ambiguous scenarios. Global optimization techniques (e.g., via solving a Traveling Salesman-style ordering over Wasserstein distances) could improve sequencing robustness, albeit at higher computational cost.

- **Motion Direction Estimation.** Since optimal transport plans only reflect source-to-target mass movement, directionality remains ambiguous in many cases. Integrating temporal color gradients, optical flow priors, or weak supervision could help infer motion vectors with orientation, enabling more accurate tracking.

- **Computational Improvements.** One major bottleneck is the computational load of full Wasserstein transport, especially for high-resolution images and long sequences. Exploring faster approximations like entropically regularized Optimal Transport (Sinkhorn), sliced OT with transport plan recovery, or GPU-based acceleration would be essential for practical deployment at scale. Additionally, storage of transport plans can be optimized by leveraging sparse matrix representations: only transport entries exceeding a significance threshold need to be retained. This drastically reduces memory consumption and allows focusing on semantically relevant mass movements, particularly in long or dense sequences.

6. Conclusion

This thesis presents a novel unsupervised framework for analyzing unordered image sequences through clustering, sequencing, and object tracking, exclusively leveraging Wasserstein-based optimal transport and DC-Distance. The overarching goal of this work is to demonstrate that scene structure and motion cues can be reliably extracted from raw pixel distributions without relying on supervised labels, object detection, or semantic features.

Main Contributions. The proposed method offers the following key contributions:

- **Wasserstein-based DC-Distance.** We introduce a clustering metric that integrates local density and global structure, enabling robust unsupervised grouping of images with spatial similarity.
- **Transport-Based Motion Tracking.** Without any object annotations, motion paths are extracted through clustering of transport vectors derived from adjacent image pairs.
- **Cross-Dataset Validation.** The entire pipeline is tested across synthetic, microscopic, and real-world wildlife datasets, demonstrating generalization and adaptability.

Core Results. The evaluation shows that the framework:

- Outperforms Euclidean and multiscale sliced Wasserstein distance (MS-SWD) baselines in clustering performance, particularly in capturing intra-sequence coherence.
- Recovers ground-truth temporal order with perfect accuracy on synthetic data and acceptable alignment in real-world scenes.
- Extracts motion subspaces without supervision, yielding semantically meaningful transport paths and movement zones directly from raw pixel input.

Final Takeaway This method provides a viable alternative to object-centric tracking pipelines. It eliminates the need for pretraining, segmentation, or keypoint matching by focusing on transport geometry. Its interpretability and domain-independence make it well-suited for low-data applications such as microscopy, scientific imaging, and wildlife observation if the necessary computational resources are available.

Bibliography

[ABK+06] Elke Achtert, Christian Böhm, Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Deriving quantitative models for correlation clusters. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 4–13, 2006.

[ABK+07] Elke Achtert, Christian Böhm, Hans-Peter Kriegel, Peer Kröger, Ina Müller-Gorman, and Arthur Zimek. Detection and visualization of subspace cluster hierarchies. In *Advances in Databases: Concepts, Systems and Applications: 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007. Proceedings 12*, pages 152–163. Springer, 2007.

[AC11] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[AC20] Syeda Mariam Ahmed and Chee Meng Chew. Density-based clustering for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10608–10617, 2020.

[AGGR98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105, 1998.

[AMG+10] Ira Assent, Emmanuel Müller, Stephan Günnemann, Ralph Krieger, and Thomas Seidl. Less is more: Nonredundant subspace clustering. In *Proc. ACM SIGKDD Workshop MultiClust.* Citeseer, 2010.

[ASW15] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

[AY00] Charu C Aggarwal and Philip S Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 70–81, 2000.

[BB00] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[BDGT19] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6182–6191, 2019.

[BDH+23] Anna Beer, Andrew Draganov, Ellen Hohma, Philipp Jahn, Christian MM Frey, and Ira Assent. Connecting the dots–density-connectivity distance unifies dbscan, k-center and spectral clustering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 80–92, 2023.

[BM21] Panthadeep Bhattacharjee and Pinaki Mitra. A survey of density based clustering algorithms. *Frontiers of Computer Science*, 15:1–27, 2021.

[BRPP15] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

[BVH+16] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016.

[BYB10] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2010.

[CIE76] OJ CIE. Official recommendations on uniform color space, color difference equations and metric color terms. *CIE Publ*, 15:1976, 1976.

[CKHP02] Selina Chu, Eamonn Keogh, David Hart, and Michael Pazzani. Iterative deepening dynamic time warping for time series. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 195–212. SIAM, 2002.

[CLZ+22] Ao Chen, Chen Li, Shuojia Zou, Md Mamunur Rahaman, Yudong Yao, Haoyuan Chen, Hechen Yang, Peng Zhao, Weiming Hu, Wanli Liu, et al. Svia dataset: a new dataset of microscopic videos and images for computer-aided sperm analysis. *Biocybernetics and Biomedical Engineering*, 42(1):204–214, 2022.

[CMS13] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[CS08] S Chandrakala and C Chandra Sekhar. A density based method for multivariate time series clustering in kernel feature space. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1885–1890. IEEE, 2008.

[CWZ+22] Fei Chen, Xiaodong Wang, Yunxiang Zhao, Shaohe Lv, and Xin Niu. Visual object tracking: A survey. *Computer Vision and Image Understanding*, 222:103508, 2022.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DBKF19] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4660–4669, 2019.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[DDXS19] Mingjing Du, Shifei Ding, Yu Xue, and Zhongzhi Shi. A novel density peaks clustering with sensitivity of local density and density-adaptive metric. *Knowledge and Information Systems*, 59:285–309, 2019.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[Dun74] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[ESK03] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 47–58. SIAM, 2003.

[EV13] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[FL19] Heng Fan and Haibin Ling. Parallel tracking and verifying. *IEEE Transactions on Image Processing*, 28(8):4130–4144, 2019.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[GQF+25] Valentin Gabeff, Haozhe Qi, Brendan Flaherty, Gencer Sumbül, Alexander Mathis, and Devis Tuia. Mammalps: A multi-view video behavior monitoring dataset of wild mammals in the swiss alps. *arXiv*, 2025.

[Hak64] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.

[HCMB14] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[HCW+15] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 749–758, 2015.

[HWW+24] Jiaqi He, Zhihua Wang, Leon Wang, Tsein-I Liu, Yuming Fang, Qilin Sun, and Kede Ma. Multiscale sliced wasserstein distances as perceptual color difference measures. In *European Conference on Computer Vision*, pages 425–442. Springer, 2024.

[JD88] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[JJ21] Jennifer Jang and Heinrich Jiang. Meanshift++: Extremely fast mode-seeking with applications to segmentation and object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4102–4113, 2021.

[JLR20] Ali Javed, Byung Suk Lee, and Donna M Rizzo. A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001, 2020.

[JNH07] Liping Jing, Michael K Ng, and Joshua Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering*, 19(8):1026–1041, 2007.

[Jol02] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[KC21] Jongwon Kim and Jeongho Cho. Dbscan-based tracklet association annealer for advanced multi-object tracking. *Sensors*, 21(17):5715, 2021.

[KKK04] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 246–256. SIAM, 2004.

[KKZ09] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *Acm transactions on knowledge discovery from data (tkdd)*, 3(1):1–58, 2009.

[KKZ12] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):351–364, 2012.

[LWW+19] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4282–4291, 2019.

[ML09] Xue Mei and Haibin Ling. Robust visual tracking using ℓ 1 minimization. In *2009 IEEE 12th international conference on computer vision*, pages 1436–1443. IEEE, 2009.

[MZK+09] Gabriela Moise, Arthur Zimek, Peer Kröger, Hans-Peter Kriegel, and Jörg Sander. Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and Information Systems*, 21:299–326, 2009.

[NNM96] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Columbia University, 1996.

[NR07] Vit Niennattrakul and Chotirat Ann Ratanamahatana. Inaccuracies of shape averaging method using dynamic time warping for time series data. In *Computational Science–ICCS 2007: 7th International Conference, Beijing, China, May 27-30, 2007, Proceedings, Part I 7*, pages 513–520. Springer, 2007.

[ODM+23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[PC+19] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[PKD05] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439. IEEE, 2005.

[Red18] Chandan K Reddy. *Data clustering: algorithms and applications*. Chapman and Hall/CRC, 2018.

[RLLY08] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International journal of computer vision*, 77:125–141, 2008.

[Rou87] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[RPDB12] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer, 2012.

[SB91] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[SGZC13] Kelvin Sim, Vivekanand Gopalkrishnan, Arthur Zimek, and Gao Cong. A survey on enhanced subspace clustering. *Data mining and knowledge discovery*, 26:332–397, 2013.

[WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[WSB03] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[Zau10] Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. 2010.

[ZLY12] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *2012 IEEE Conference on Computer vision and pattern recognition*, pages 1838–1845. IEEE, 2012.

A. Appendix

A.1. Parameters used

Procedure	Parameters
K-Means	Number of clusters set to ground truth; initialization = k-means++; n_init = 10; max_iter = 300.
Spectral Clustering	Number of clusters set to ground truth.
HDBSCAN	`min_cluster_size = 5`; `min_samples` equals `min_cluster_size` .
DC-Distance (Wasserstein)	`minPts = 2` for DC-distance computation, unless explicitly stated otherwise.

Table A.1.: Parameters used for each procedure. The listed values apply uniformly to all experiments in which the respective procedure was used, unless explicitly stated otherwise. Specific parameter adjustments that apply only to individual experiments are stated alongside the corresponding experimental description.

A.2. Python Packages

The Python environment used for the experiments is listed below:

```
absl-py==2.3.1
aiohappyeyeballs==2.6.1
aiohttp==3.12.15
aiosignal==1.4.0
ansicolors==1.1.8
anyio==4.9.0
argon2-cffi==23.1.0
argon2-cffi-bindings==21.2.0
arrow==1.3.0
asttokens==3.0.0
async-lru==2.0.5
async-timeout==5.0.1
attrs==25.3.0
Automat==20.2.0
babel==2.17.0
bcrypt==3.2.0
```

```
beautifulsoup4==4.13.4
bleach==6.2.0
blinker==1.4
certifi==2020.6.20
cffi==1.17.1
chardet==4.0.0
charset-normalizer==3.4.1
click==8.0.3
cloud-init==24.4.1
colorama==0.4.4
colormath==3.0.0
colour-science==0.4.6
comm==0.2.2
command-not-found==0.3
configobj==5.0.6
constantly==15.1.0
contourpy==1.3.2
cryptography==3.4.8
cycler==0.12.1
dbus-python==1.2.18
debugpy==1.8.14
decorator==5.2.1
defusedxml==0.7.1
distro==1.7.0
distro-info==1.1+ubuntu0.2
entrypoints==0.4
exceptiongroup==1.2.2
executing==2.2.0
fastjsonschema==2.21.1
filelock==3.18.0
fonttools==4.57.0
fqdn==1.5.1
frozenlist==1.7.0
fsspec==2025.3.2
h11==0.14.0
h5py==3.14.0
hdbscan==0.8.40
hf-xet==1.1.10
httpcore==1.0.8
httplib2==0.20.2
httpx==0.28.1
huggingface-hub==0.35.1
hyperlink==21.0.0
```

```
idna==3.3
ImageHash==4.3.2
imageio==2.37.0
importlib-metadata==4.6.4
incremental==21.3.0
ipykernel==6.29.5
ipython==8.35.0
isoduration==20.11.0
jedi==0.19.2
jeepney==0.7.1
Jinja2==3.0.3
joblib==1.4.2
json5==0.12.0
jsonpatch==1.32
jsonpointer==2.0
jsonschema==4.23.0
jsonschema-specifications==2024.10.1
jupyter-events==0.12.0
jupyter-lsp==2.2.5
jupyter_client==8.6.3
jupyter_core==5.7.2
jupyter_server==2.15.0
jupyter_server_terminals==0.5.3
jupyterlab==4.4.0
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
keras==3.11.3
keyring==23.5.0
kiwisolver==1.4.8
launchpadlib==1.10.16
lazr.restfulclient==0.14.4
lazr.uri==1.0.6
lazy_loader==0.4
lightning-utilities==0.15.2
markdown-it-py==4.0.0
MarkupSafe==2.0.1
matplotlib==3.10.1
matplotlib-inline==0.1.7
mdurl==0.1.2
mistune==3.1.3
ml_dtypes==0.5.3
more-itertools==8.10.0
mpmath==1.3.0
```

```
multidict==6.6.4
namex==0.1.0
nbclient==0.10.2
nbconvert==7.16.6
nbformat==5.10.4
nest-asyncio==1.6.0
netifaces==0.11.0
networkx==3.4.2
notebook==7.4.0
notebook_shim==0.2.4
numpy==2.2.5
nvidia-cublas-cu12==12.4.5.8
nvidia-cuda-cupti-cu12==12.4.127
nvidia-cuda-nvrtc-cu12==12.4.127
nvidia-cuda-runtime-cu12==12.4.127
nvidia-cudnn-cu12==9.1.0.70
nvidia-cufft-cu12==11.2.1.3
nvidia-curand-cu12==10.3.5.147
nvidia-cusolver-cu12==11.6.1.9
nvidia-cusparse-cu12==12.3.1.170
nvidia-cusparselt-cu12==0.6.2
nvidia-nccl-cu12==2.21.5
nvidia-nvjitlink-cu12==12.4.127
nvidia-nvtx-cu12==12.4.127
oauthlib==3.2.0
opencv-contrib-python==4.12.0.88
opencv-python==4.11.0.86
optree==0.17.0
overrides==7.7.0
packaging==25.0
pandocfilters==1.5.1
papermill==2.6.0
parso==0.8.4
pexpect==4.8.0
pillow==11.2.1
platformdirs==4.3.7
POT==0.9.5
prometheus_client==0.21.1
prompt_toolkit==3.0.51
propcache==0.3.2
psutil==7.0.0
ptyprocess==0.7.0
pure_eval==0.2.3
```

```
pyasn1==0.4.8
pyasn1-modules==0.2.1
pycparser==2.22
Pygments==2.19.1
PyGObject==3.42.1
PyHamcrest==2.0.2
PyJWT==2.3.0
pyOpenSSL==21.0.0
pyparsing==2.4.7
pyparted==3.11.7
pyrsistent==0.18.1
pyserial==3.5
python-apt==2.4.0+ubuntu4
python-dateutil==2.9.0.post0
python-debian==0.1.43+ubuntu1.1
python-json-logger==3.3.0
python-magic==0.4.24
pytorch-lightning==2.3.0
pytz==2022.1
PyWavelets==1.8.0
PyYAML==5.4.1
pyzmq==26.4.0
referencing==0.36.2
requests==2.32.3
rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
rich==14.1.0
rpds-py==0.24.0
safetensors==0.6.2
scikit-image==0.25.2
scikit-learn==1.6.1
scipy==1.15.2
SecretStorage==3.3.1
Send2Trash==1.8.3
service-identity==18.1.0
six==1.16.0
sniffio==1.3.1
sos==4.8.2
soupsieve==2.6
ssh-import-id==5.11
stack-data==0.6.3
sympy==1.13.1
systemd-python==234
```

```
tenacity==9.1.2
terminado==0.18.1
threadpoolctl==3.6.0
tifffile==2025.3.30
timm==1.0.20
tinycss2==1.4.0
tomli==2.2.1
torch==2.6.0
torchmetrics==1.4.0.post0
torchvision==0.21.0
tornado==6.4.2
tqdm==4.67.1
traitlets==5.14.3
triton==3.2.0
Twisted==22.1.0
types-python-dateutil==2.9.0.20241206
typing_extensions==4.13.2
ubuntu-pro-client==8001
ufw==0.36.1
unattended-upgrades==0.1
uri-template==1.3.0
urllib3==1.26.5
wadllib==1.3.6
WALinuxAgent==2.2.46
wcwidth==0.2.13
webcolors==24.11.1
webencodings==0.5.1
websocket-client==1.8.0
yarl==1.20.1
zipp==1.0.0
zope.interface==5.4.0
```