Zero-Shot Monocular Motion Segmentation: A Fusion of Deep Learning and Geometric Approaches

by

Yuxiang Huang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis incorporates findings from the following research papers. I was the primary contributor to the design, implementation, experimentation and analysis of the methods proposed in these papers. Prof. John Zelek and Prof. Yuhao Chen provided advice on the methodology and contributed to the drafting and revision of these papers.

Yuxiang Huang, John Zelek, ”Motion Segmentation from a Moving Monocular Camera”, Workshop on Robotic Perception and Mapping: Frontier Vision and Learning Techniques, the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023

Yuxiang Huang, John Zelek, ”A Unified Model Selection Technique for Spectral Clustering Based Motion Segmentation”, Journal of Computational Vision and Imaging Systems, 2023

Yuxiang Huang, Yuhao Chen, John Zelek, ”Dense Monocular Motion Segmentation Using Optical Flow and Pseudo Depth Map: A Zero-Shot Approach”, accepted by the 21st Conference on Robots and Vision (CRV), 2024

Yuxiang Huang, Yuhao Chen, John Zelek, ”Zero-Shot Monocular Motion Segmentation in the Wild by Combining Deep Learning with Geometric Motion Model Fusion”, accepted by the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), 2024

Abstract

Identifying and segmenting moving objects from a moving monocular camera is difficult when there is unknown camera motion, different types of object motions and complex scene structures. Deep learning methods achieve impressive results for generic motion segmentation, but they require massive training data and do not generalize well to novel scenes and objects. Conversely, recent geometric methods show promising results by fusing different geometric models together, but they require manually corrected point trajectories and cannot generate a coherent segmentation mask.

This work proposes an innovative zero-shot motion segmentation approach that seamlessly combines the strengths of deep learning and geometric methods. The proposed method first generates object proposals for every video frame by using state-of-the-art foundation models, and then extracts different object-specific motion cues. Finally, the method uses multi-view spectral clustering to synergistically fuse different motion cues together to cluster objects into distinct motion groups, resulting in a coherent segmentation. The key contributions of this work are as follows:

- Proposing the first zero-shot motion segmentation pipeline that performs dense motion segmentation on different scenes and object classes without any training.
- This work is the first to combine epipolar geometry and optical flow-based motion models for motion segmentation. Multi-view spectral clustering is used to effectively combine different motion models to achieve good motion segmentation results in complex scenes.

Through extensive experimentation and comparative analysis, we validate the efficacy of the proposed method. Despite not being trained on any data, the method is able to achieve competitive results on real-world datasets, some of which are even better than those of the state-of-the-art motion segmentation methods trained in a supervised manner. This work not only contributes to the advancement of monocular motion segmentation, but also shows that combining different geometric motion models and motion cues is very important in analyzing the motions of objects.

Acknowledgements

I would like to sincerely thank my supervisor, Prof. John Zelek, for his guidance and support throughout my master's study. His encouragement, enthusiasm and insights have really inspired me to strive for excellence in every aspect of my academic journey. I would also like to sincerely thank Prof. Yuhao Chen for supporting my work and providing a lot of useful feedback, which greatly strengthened this work. Finally, I would like to thank all my colleagues and friends at the VIP lab for all the interesting discussions and fun game nights. Thank you for making it such a wonderful time.

Dedication

To my family and friends

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Scope

This thesis is focused on developing a dense monocular motion segmentation framework that works regardless of camera motion, object appearance, motion type and scene geometry, without any training. More specifically, given a video from a monocular camera, the goal is to create a dense segmentation mask of every object that is moving independently. If only part of the object is moving, the entire object needs to be segmented. Since segmenting moving objects from a static camera is a well-studied problem, the main focus of this thesis will be on segmenting moving objects from a moving camera. That said, the proposed method will also work with a static camera.

A closely related area of research to motion segmentation is video object segmentation (VOS) [85]. However, it is important to distinguish between the two: The goal of VOS is to segment only the moving objects in the foreground, while motion segmentation focuses on segmenting any object that is moving independently, regardless of whether it is in the foreground or not.

1.2 Motivation

Being able to identify and segment moving objects from a moving camera is crucial for various applications such as autonomous navigation, robotics, SLAM and scene understanding in general. In a dynamic scene, the video camera is moving at an unknown velocity with respect to the environment. Such scenarios pose many challenges to motion segmentation methods such as motion degeneracy and motion parallax [25]. Existing monocular motion segmentation methods have two major limitations in dynamic environments: 1) They either do not perform well due to geometrical limitations and noises [4, 79, 80, 82], or 2) they require end-to-end training (often fully supervised) to produce good results, which is computation-intensive and reduces the generalizability of the method [49, 74, 59].

It is important to address these limitations in order to achieve accurate monocular motion segmentation in diverse natural scenes with different object motions, object appearances and scene geometry, as well as to minimize the training cost. To tackle these challenges, we draw inspiration from the recently proposed deep learning foundation models and the well-established geometry-based motion segmentation approaches. The recent computer vision foundation models are very good at discovering, segmenting and tracking objects in videos, but they cannot distinguish between static and moving objects. On the other hand, a significant number of geometry-based approaches have been proposed during the past years on motion clustering and segmentation, and recent studies have shown impressive results by fusing different geometric models together, especially on challenging scenes. However, such methods require manually corrected point trajectories as inputs and cannot generate dense segmentation masks. Based on these reasons, we propose to combine the advantages of both deep learning and geometric methods to perform motion segmentation by applying geometric model fusion on object proposals generated by computer vision foundation models, which results in a zero-shot motion segmentation approach.

1.3 Overview

The proposed approach fuses multiple geometric models to cluster object proposals into different motion groups. More specifically, our motion segmentation pipeline first uses foundation models to detect, recognize and segment all common objects and the background in the video sequence, then tracks these objects using an object tracker. For each object in each frame, we obtain two types of motion cues – one set of object-specific point trajectories and one set of object-specific optical flow masks. We then synergistically fuse these two types of motion cues to cluster different objects in the proposal into different motion groups. The key contributions of this thesis are the following:

1. Modeling multiple complex motions in challenging scenes by combining epipolar geometry and optical flow based motion models using multi-view spectral clustering.

2. Proposing the first zero-shot motion segmentation pipeline that does not need any training and achieve generalization on different scenes and object classes.

In the following sections, this thesis first provides a literature review on notable existing motion segmentation methods as well as some necessary knowledge on motion segmentation. It then describes the detailed methodology for generating per-frame object proposals and clustering them into different motion groups. Finally, it shows experimental results of the proposed method, compares it with other state-of-the-art motion segmentation methods, provides discussion, conclusion and future research directions.

Chapter 2

Literature Review

Monocular motion segmentation can be broadly categorized into three groups: (1) Intensity based methods [51, 64, 55, 7, 9], (2) sparse correspondence based methods [18, 34, 33, 11, 20, 53, 40, 82, 3] and (3) deep learning based methods [74, 65, 10, 59, 16, 8, 21, 49, 52, 47].

2.1 Intensity Based Motion Segmentation

Intensity based methods are based on the brightness constancy constraint, which assumes the pixels brightness stays consistent over time. Intensity based methods can be further categorized into direct and indirect methods. Direct methods [1, 51, 27, 26, 75] directly take a pair of images as input and combine the two processes of optimizing for the brightness constancy constraint and estimating the motion models together. In contrast, indirect methods [64, 78, 55, 7] rely on pixel-wise correspondences as input, and produce a pixel-wise segmentation mask indicating different motion groups. Such pixel correspondences are usually obtained from optical flow, which is based on the brightness constancy assumption.

2.1.1 Direct Methods

Early works of direct methods [1, 51, 27] assume there is no more than one independent moving object in the scene (including the camera) and directly compute the camera motion parameters using a pair of of image frames, without computing optical flow. [51] develops a novel approach for estimating the movement of an observer relative to a planar surface using image brightness derivatives, bypassing the traditional optical flow computation. The method analyzes spatial and temporal brightness variations of at least eight points. The method uses nine nonlinear equations to extract the motion and surface parameters by optimizing the least squares loss. [27] introduces methods to determine the various types of motions of an observer in a static environment, including pure rotation, pure translation, and complex motions with known rotation. These methods primarily focus on minimizing the discrepancy between the observed and predicted pixel temporal brightness changes, using first-order brightness derivatives without establishing point correspondences or estimating optical flow. The research emphasizes the significance of a broad field of view for accurate motion component recovery and the relevance of points with minimal brightness change over time. Additionally, it addresses the challenges of large depth ranges and the importance of effective spatial and temporal filtering of image data to avoid aliasing.

More recently, [75] proposes a method to perform motion segmentation on multiple moving objects. The authors propose a closed-form solution to perform motion segmentation using the proposed multi-body brightness constancy constraint, a polynomial relationship that connects pixel coordinates, image derivatives and motion models independently from the segmentation of image data, achieving multi-label motion segmentation in both static and dynamic scenes directly from a sequence of raw image data, without intermediate steps like keypoint tracking or optical flow computation.

2.1.2 Indirect Methods

Comparing to direct methods, most recent works on intensity based methods use the indirect approach [63, 64, 78, 55, 7], possibly due to the fast advance in optical flow estimation [19, 69, 60, 32, 70, 67]. Indirect methods first computes optical flow as an intermediate output and perform motion segmentation using the optical flow map as input.

A popular indirect motion segmentation approach is the variational method, which iteratively optimizes the motion parameters of different moving objects using a regularized energy function. [63, 64] propose a variational approach to minimize an energy function that describes the discrepancy between the reconstructed optical flow field and the ground truth. A functional containing two terms is proposed – the first term describes the difference between the interpreted and the ground truth 3D screw motion parameters, and the second term is a regularizer based anisotropic diffusion aiming to preserve the object boundaries. The authors iteratively optimize a system of equations derived from this functional using the half-quadratic algorithm after a random initialization. The method does need to know the ground truth number of motions in the image sequence.

Some approaches combine optical flow and appearance cues as input: In [55], the authors use both motion and appearance cues (i.e., pixel intensity values) to perform temporal consistent binary motion segmentation, which is able to segment objects even when they pause their movements in some frames. The authors propose a method to perform binary motion segmentation using both motion boundaries and an object appearance model. The method produces the binary segmentation in four steps: 1) Computing the motion boundaries of the foreground object using the gradients of the direction and magnitude of the optical flow; 2) compute an inside-outside map to label the pixels as foreground or background based on it; 3) build an appearance model using two GMMs using the inside-outside map and the labeled pixels in the current and nearby frames, and build a motion prior based on the percentage of the superpixel inside the object and the propagation from previous superpixels 4) Minimize the energy combining these two unary potentials and two pairwise potentials for smoothness.

More recent approaches have achieved improved results by using probabilistic motion models to determine if a region in the image belong to a certain moving object. In [7],

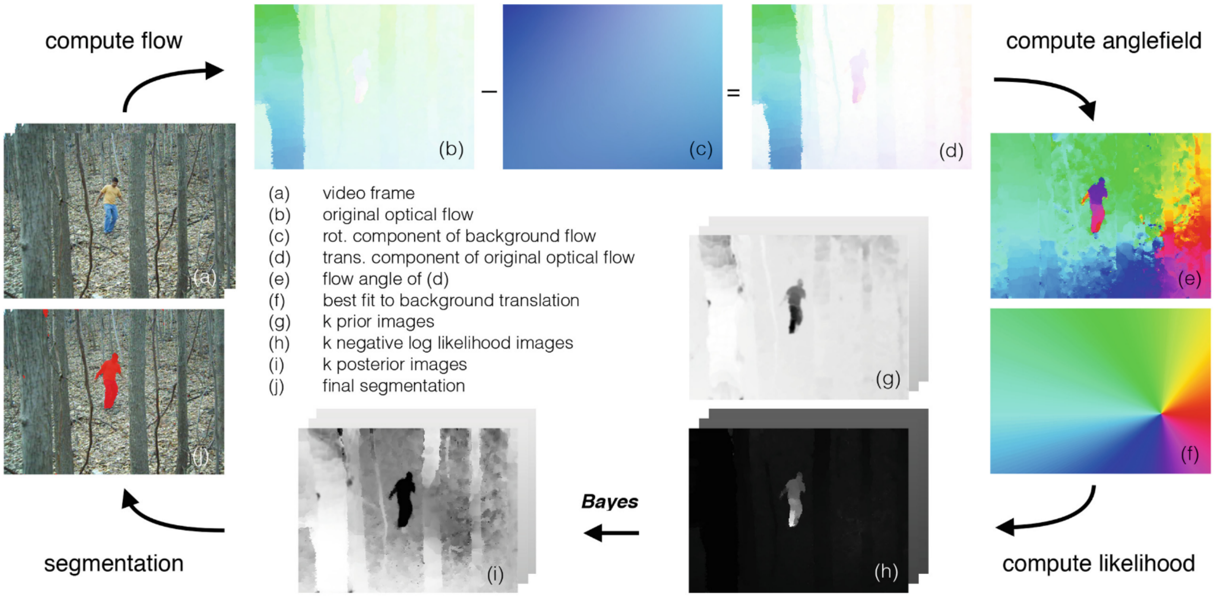

Figure 2.1: An example of indirect binary motion segmentation method proposed by [7]. The authors first solve for an optical flow angle field to estimate the background motion, then compute the possibility of each pixel belonging to the background motion using the Bayesian model

the authors proposed a probabilistic motion model using the lengths and angle field of the rotation-compensated optical flow vectors, together with sequential RANSAC to perform binary motion segmentation. Figure 3.1 shows the framework of [7]. The authors first solve for an optical flow angle field to estimate the background motion, then compute the possibility of each pixel belonging to the background motion using the Bayesian model.

Removing the rotation part of the optical flow can improve the robustness of the method in scenes with high depth variation, but will also result in additional noises in the motion segmentation process and prolonged processing time.

In general, There are three main drawbacks to intensity based motion segmentation methods: 1) Intensity based methods cannot handle strong depth variations from a moving camera – if the scene contains these elements (e.g. road scenes), these methods will fail to distinguish if a part of the image is moving independently or is just at a different depth from its surroundings, because the motion flow vectors projected to a 2D image from the 3D space are determined by both the depth and the screw motion of the object [48]. 2) Typically, intensity based methods assume the brightness constancy constraint, which states the intensity (brightness) of any pixel in an image remains constant between consecutive frames, even though its position may change due to motion, however, this assumption is not always satisfied. 3) Current intensity based methods mostly are mostly focused on binary motion segmentation and do not perform well when there are multiple moving objects. Additionally, optical flow based methods heavily rely on accurate optical flow prediction. Most optical flow based motion segmentation methods can still suffer heavily from noise and produce incorrect results given noisy optical flows [2].

2.2 Sparse Correspondence Based Motion Segmentation

Unlike intensity based methods or deep learning methods, sparse correspondence based methods output clusters of predefined kepypoints corresponding to different motion groups instead of dense segmentation masks. These methods can be further categorized into two-frame based methods and multi-frame based methods.

2.2.1 Two-Frame Correspondence Based Methods

Two-frame based methods [71, 18, 34, 6] usually recover motion parameters by solving an iterative energy minimization problem of finding a certain number of geometric models (e.g., fundamental matrices) on a set of matched feature points, to minimize an energy function that evaluates the quality of the overall clustering of corresponding feature points.

Early works of two-frame motion segmentation typically uses generalized RANSAC methods to optimize the geometric model fitting losses between a pair of images. For example, [71] proposed to apply RANSAC sequentially on a set of matched keypoints between a pair of images. This approach involves selecting a model through random sampling at each iteration, aiming to maximize the number of inliers to a specific fundamental matrix given a specified threshold. Once a model and its associated inliers have been identified, these inliers are then removed from the pool of data points. This entire process is then iteratively repeated to uncover additional models.

Other than relying on RANSAC, some more recent methods use the α-expansion algorithm to minimize a more sophisticated joint loss function for better results. [18] and [34] propose an extension of the α-expansion algorithm to simultaneously minimize a joint loss consisting of a data cost, a smooth cost and a label cost in order to approximate a set of different motion models in a pair of images. The motion models are proposed by randomly sampling from the matched keypoints. The data cost is defined as the the mean squared Sampson's distance of keypoints with respect to fundamental matrix fitting. The authors proposed the smooth cost and the label cost in addition to the data cost in order to penalize overly complex motion models and to encourage spatial coherence respectively.

[6] improves upon [18] and [34] to better deal with more complex scenario where there are many potential geometric motion models by progressively propose new potential models. More specifically. the authors use GC-RANSAC [5] with NAPSAC sampling [50] to progressively propose new motion models one by one, and use MSAC [72] instead of the traditional inlier counting method to evaluate the quality of each proposed model. The authors also use or propose new criteria for proposal validation and termination in order to make this method an any-time algorithm, which means the algorithm can be terminated at any given time and still returns the current best solution.

Two-frame based methods are usually focused on geometric model fitting given a set of matched keypoints from a pair of images. While geometric model fitting approaches can be used on motion segmentation, they can also be used on other applications such as plane detection by fitting homographies. However, a fundamental flaw of these methods is that they are only able to fit one type of geometric model on a given pair of image. When they are applied to motion segmentation, the only type of geometric model that can be used is fundamental matrix. However, fundamental matrix can only capture different motions if the motion is not confined to epipolar geometry [25], which is often not the case in reality.

2.2.2 Multi-Frame Correspondence Based Methods

Unlike two-frame based methods, multi-frame based methods [40, 82, 3, 36, 79, 20, 61, 73, 76, 11, 53] usually establish point correspondences over multiple frames using an optical flow based point tracking. Noisy, occluded and unwanted points are often manually removed to produce a sparse set of completely noise-free point trajectories. The main difference between multi-frame based methods and two-frame based methods is that multi-frame based methods have more matched keypoints to work with – the keypoints are obtained from a sequence of images and can be used to analyze long-term motions. Multi-frame based methods not only use different geometric models, but also use spatio-temporal similarities to uncover different motions, Moreover, unlike two-frame based methods which only rely on epipolar geometry to detect different motions, multi-frame based methods are able to combine different geometric models together and achieve better performance in more challenging scenarios.

Subspace clustering is a popular method for multi-frame based motion segmentation [73, 61, 76, 20]. These methods are based on the assumption that in a video with F frames, the 2D coordinates of feature points trajectories lying on a single rigid object belong to an affine subspace of $\mathbb{R}^{2F}$ of dimension three [20]. This assumption is true when the feature points are projected onto the 2D camera plane using an affine camera model, where the depths of the feature points are approximated to be constant. Due to this reason, such algorithms work well when the camera motion remains close to the camera plane, or when the moving objects' motions do not involve strong depth changes. However, if these assumptions are not fulfilled to a certain extent, subspace clustering based methods will fail to deliver satisfying results. usually use spectral clustering on affinity matrices constructed using the results of geometric model fitting [40, 82, 3], subspace fitting [20, 61, 73, 76] or pairwise affinities derived from spatio-temporal motion cues and appearance cues [11, 53].

Some multi-frame based methods also use spatio-temporal motion cues to determine if different regions of the images belong to different motions [11, 53, 37, 38]. [11] and [53] propose a pairwise trajectory motion affinity function computed using spatially regularized pairwise trajectory motion difference. The motion difference between a pair of trajectories is computed as the greatest squared difference between the two trajectories motion vector (i.e. optical flow) throughout a certain number of frames. Since optical flow vectors only account for motions on the camera plane (i.e., motions on the x and y axes only, but not the z axis associated with the depth), in order to factor in motions on the depth axis, the authors multiply the trajectory motion difference with the average spatial distance between the two trajectories. The final pairwise motion affinity score is calculated as the exponential of the normalized and regularized motion difference value. Spectral clustering

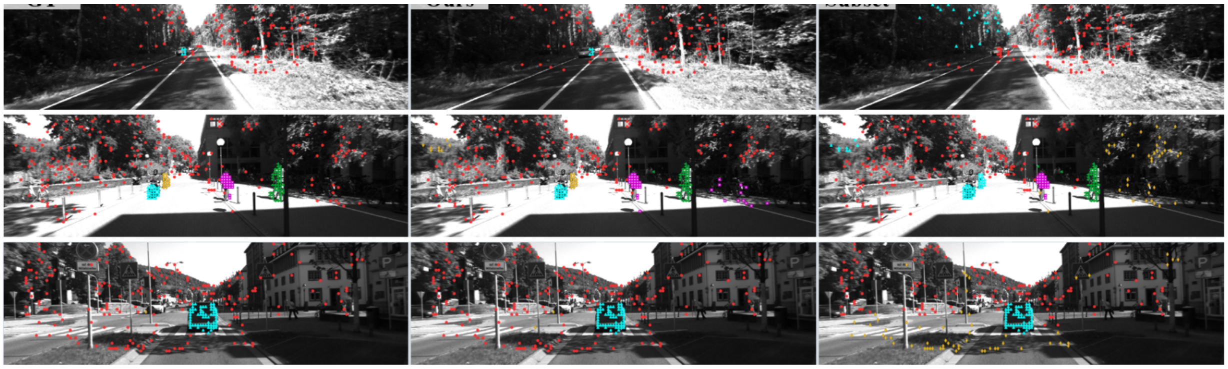

Figure 2.2: Example Outputs of State-of-the-Art multi-frame correspondence based methods on the KT3DMoSeg dataset [82]. The left column is the groundtruth, the middle column is the results of [36], and the right column is the results of [82]. The KT3DMoSeg dataset is a challenging dataset due to its various degenerate motions and motion parallax effects, but both methods are able to achieve good results by fusing multiple geometric models.

is used to cluster all the point trajectories into segments of different motions after obtaining the pairwise motion affinity scores of every pair of trajectories. The authors also propose a unique model selection technique to determine the number of motion clusters automatically, by minimizing a spatial regularity term separately from the main motion clustering objective. [37] and [38] improve upon [11] and [53] to provide more accurate motion segmentation results on small moving objects. More specifically, they propose to use minimum cost multicut to segment the point trajectories into different motion groups, where every trajectory is considered as a vertex in the graph. The authors also propose a new pairwise affinity function between trajectories by incorporating all three of color, spatial and motion similarities. For the motion similarity score, the authors remove the spatial distance regularizer used in [53] to avoid over-segmentation by minimum cost multicut.

Although spatio-temporal affinity based motion segmentation methods show success in some general cases, they usually struggle to produce coherent motion segmentation results at the object level and tend to over segment the objects. This is mainly due to two reasons: 1) the spatio-temporal motion model does not work well on non-translational motions (i.e., rotation or motion on the depth axis) in general due to its nature; 2) The moving object itself may consist of different motions, for example, when a human walks, the torso and the limbs belong to different rigid motion groups, so they tend to be over segmented.

In contrast to subspace clustering based methods or methods using spatio-temporal motion cues, most recent multi-frame based methods typically combine multiple different geometric models to achieve the state-of-the-art results [82, 36, 79, 30]. [82] propose to fuse three different geometric models (homography, fundamental matrix and affine camera model) on consecutive frame pairs using multi-view spectral clustering to perform motion segmentation. This method achieves state-of-the-art results on all rigid body motion segmentation tasks, surpassing all previous methods, especially on challenging road scenes where rotations and forward and backward motions are prominent. [36] improves upon [82] by focusing on finding a consensus motion affinity matrix from all three geometric models and also proposed a new optimization scheme to optimize the clustering process using co-regularized spectral clustering. [79] employs a two-stage approach by first over segmenting the trajectories points using multiple fundamental matrix fitting on selected samples and then merging the over-segmented clusters using optical flow. [30] proposes to improve the segmentation coherency by first using an initial grouping of the point trajectories using an object segmentation mask, then using fundamental matrix fitting and spectral clustering on to obtain the final motion segmentation. Geometric model fusion techniques achieve the best results on rigid motion segmentation given trajectory points in both general scenes and challenging road scenes, however, since they are mostly based on geometric models, they may still struggle to distinguish non-rigid motions.

In conclusion, multi-frame correspondence based motion segmentation methods have achieved impressive results under experimental settings largely thanks to fusing different geometric models together, however, the performance of sparse correspondence based methods is still strongly determined by the accuracy of the point correspondence, since most existing methods rely on manually corrected point correspondences and cannot handle outliers and excessive noise. Another drawback is of point correspondence based methods in general is that they typically do not have an accurate automatic model selection scheme to automatically infer the number of motions in the scene. Many methods require a groundtruth number of motions as input [82, 36, 79, 30], but even for the methods that propose custom model selection techniques do not usually perform well under practical settings.

2.3 Deep Learning Based Methods

Deep learning based methods usually takes a pair or a sequence of input frames (sometimes also their optical flow masks) as input and directly produces a either a binary segmentation mask of moving vs static objects [74, 65, 59, 21, 10], or a multi-label segmentation mask showing different objects of different motions [16, 49, 52, 83, 15, 47].

Many deep learning based methods adopt a fully supervised approach [74, 65, 16, 49]. These methods typically train a CNN-based encoder-decoder network to perform end-to-end learning, which is computation-intensive. Their network architecture usually have the following components: (1) a module to extract the motion information from consecutive frames, (2) a module to extract appearance information from the same frames, (3) a module to fuse the appearance and motion information, and (4) a decoder to generate the final segmentation.These methods perform very well on scenes similar to the datasets they are trained on, but cannot scale well to unseen environment where there are different motion patterns or object classes. Moreover, the data collection and training process are very time-consuming and computation intensive, which make them not an ideal method.

Aside from supervised methods, some methods use a self-supervised and unsupervised approach. In [8], the authors extended their previous work [7] by proposing an self-supervised approach to train a neural network to perform motion segmentation on synthetic angle fields, given that most optical flows can be reduced to rotation-compensated angle fields. The rotation-compensated angle field is obtained using the following steps. First, the camera rotation is computed using the known camera intrinsic and extrinsic matrix. Then, the rotation components are subtracted from the optical flow fields to obtain the rotation compensated optical flow fields. Finally, each optical vector in the optical flow field is normalized to obtain the rotation compensated angle field. In [46], the authors proposed an unsupervised learning method to solve multi-label motion segmentation problem. The method first relies on the Expectation-Maximization (EM) algorithm to produce motion segmentation, and then trains a motion segmentation network using these generated results to avoid running the slow EM-algorithm during the run time. However, these two methods purely rely on optical flow for motion information and thereby inheriting the limitations of optical flow. In order to alleviate this limitation, [15] propose to train image segmentation and motion segmentation models together using both optical flow and raw video frames as inputs due to the fact that motion and appearance cues are usually highly related in practice. The unsupervised training is done in a very similar way as [46] using the EM-algorithm.

One of the state-of-the-art deep learning based motion segmentation network is Raptor [52], where the authors train a CNN model on the 3 consecutive image frames, their optical flow maps and monocular depth maps. Using the depth map is a crucial since it compensates for the weakness of optical flow in distinguishing motion parallax from a moving camera. More specifically, if the scene contains significant depth variation, using only optical flow is insufficient in distinguishing if a part of the image is moving independently or is just at a different depth from its surroundings, because the motion flow vectors projected to a 2D image from the 3D space are determined by both the depth and

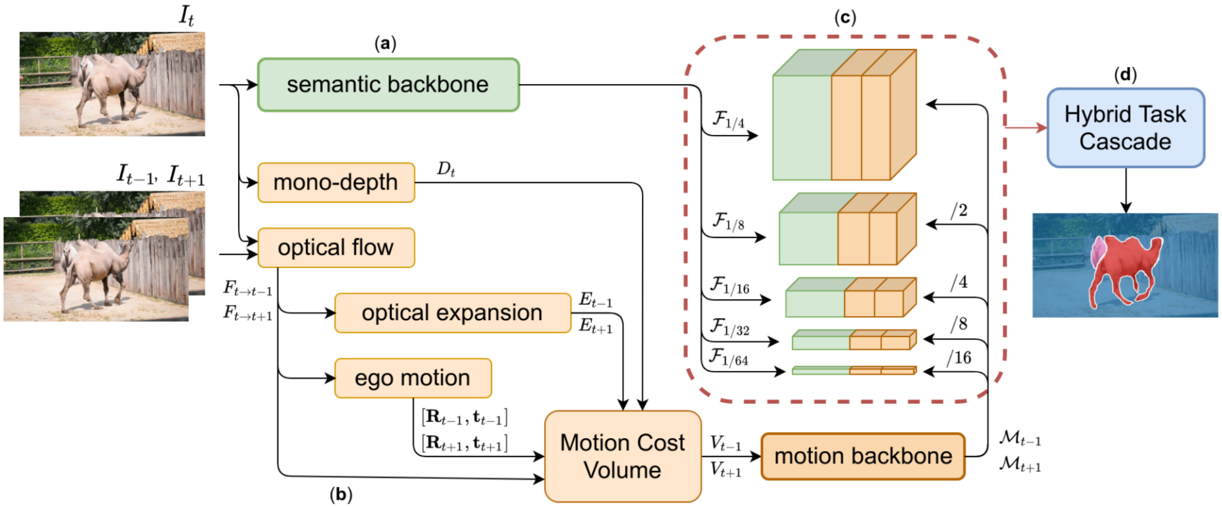

Figure 2.3: The network architecture of Raptor [8], one of the state-of-the-art monocular motion segmentation network. The network takes 3 consecutive video frames as well as their monocular depth map and optical flow field as inputs, and outputs a multi-label motion segmentation mask indicating moving objects in different motions.

the screw motion of the object [48]. The authors also incorporate the image appearance information in their model by pretraining a class-agnostic object segmentation module on the COCO object segmentation dataset [43]. The rest of the training is done on synthetic datasets only and the results translate relatively well to other datasets.

Deep learning based methods are currently the state-of-the-art methods for dense monocular motion segmentation from a moving camera, however, they do have limitations such as requiring training and some form of supervision. Moreover, many current methods rely on optical flow as the sole input for motion information, which makes their model inherit the drawbacks of optical.

2.4 Summary

Monocular motion segmentation is a fundamental task in computer vision. It has evolved significantly and has branched into three distinct categories: intensity-based methods, sparse correspondence-based methods, and deep learning-based methods.

Intensity-based methods leverage the brightness constancy constraint to distinguish motion segments. This category is divided into direct methods, which optimize the brightness constancy constraint and motion model estimation simultaneously without optical flow, and indirect methods, which rely on pixel-wise correspondences from optical flow for segmentation. Most recent methods use the indirect approach. Despite the advancements, intensity-based methods struggle with scenes of high depth variation and fail to uphold the brightness constancy in dynamic lighting conditions, limiting their applicability.

Sparse Correspondence-based methods focus on clustering predefined keypoints into different motion groups, with two-frame and multi-frame approaches offering solutions based on geometric model fitting and spatio-temporal similarities. Multi-frame based approaches have demonstrated impressive results, particularly when fusing different geometric models. However, their performance heavily depends on the accuracy of point correspondences and the manual removal of outliers, posing challenges in noisy or complex scenes. They also cannot produce dense segmentation masks due to the nature of these methods. The lack of effective automatic model selection schemes further limits their application in diverse settings.

Deep Learning-based Methods represent the forefront of monocular motion segmentation, directly producing dense segmentation masks from image sequences. Fully supervised approaches have shown exceptional performance on familiar scenes, yet they struggle with generalization to new environments due to the intensive data and computational demands. Unsupervised and self-supervised strategies offer promising results by using synthetic data and innovative training techniques to overcome the limitations of supervised methods. Incorporating depth information and integrating appearance cues are effective ways to enhance model robustness.

Despite the progress, the field faces challenges like model generalizability and the need of supervision. Future research aims to integrate the strengths of these methods and explore new techniques for more adaptable and efficient motion segmentation methods.

Chapter 3

Fundamentals of Monocular Motion Estimation

In order to identify and segment different moving objects from a moving camera, we first need to understand how to estimate the motion of a moving object and the moving camera from a sequence of RGB images. How to accurately estimate these motions is a challenging problem. In this chapter, we first explain the theories behind monocular motion estimation. We then discuss the geometric and mathematical techniques that can be used to distinguish different moving objects from a monocular camera on a fundamental level.

3.1 Three-Dimensional Interpretation of Optical Flow

3.1.1 Camera Projection model

Figure 3.1 shows an 3D orthogonal coordinate system with basis vectors **I**, **J** and **K** on X, Y and Z axes. This coordinate system represents the 3D world coordinate system, while $\boldsymbol{\pi}$ represents the image plane at a distance f from the world origin **O**. Axis Z represents the depth and is perpendicular to image plane $\boldsymbol{\pi}$, while axes X and Y are parallel to the image plane. Let the origin of the image plane **o** be on the depth axis Z.

Let **P** be a random point on a random surface in the 3D world coordinate system with 3D coordinate (X, Y, Z), and let **p** be the 2D projection of **P** on the image plane $\boldsymbol{\pi}$ with

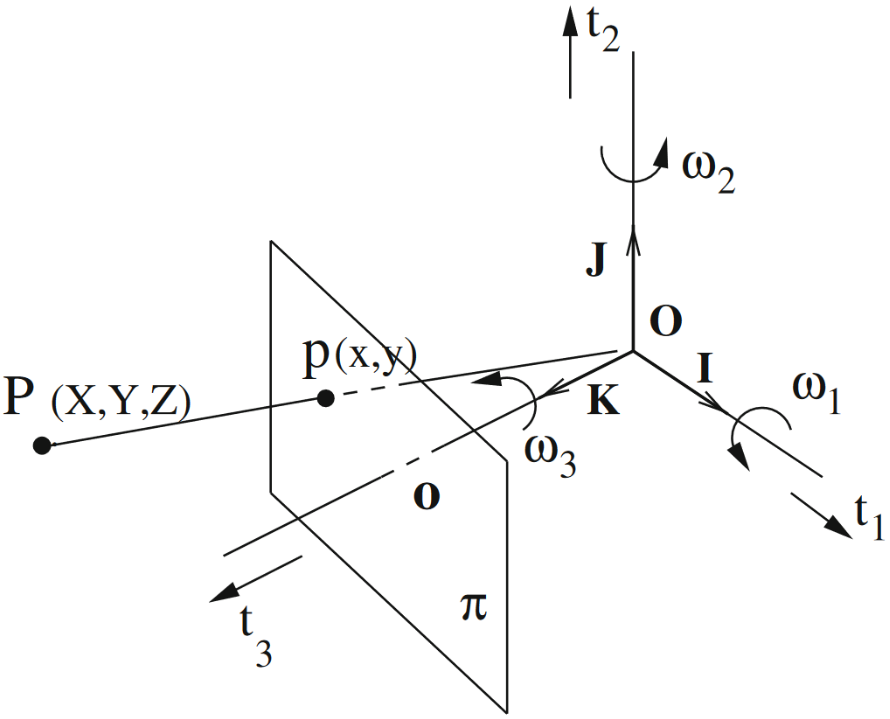

Figure 3.1: Camera Projection Model [48]

its 3D coordinate being (x, y, f). Thus, we can establish the following relationship among points **P**, **p** and **O**:

$$\frac{X-0}{x-0} = \frac{Y-0}{y-0} = \frac{Z-0}{f-0} \tag{3.1}$$

And we can represent the projected 2D coordinate of **p** on the image plane $\boldsymbol{\pi}$ as follows:

$$\begin{aligned} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z} \end{aligned} \tag{3.2}$$

By taking the derivatives on both sides of the projection equations above, we can obtain the following equations, where u and v correspond to the 2D velocities of $\mathbf{p}$ projected from $\mathbf{P}$ on the image plane along the X and Y axes (also known as the motion flow):

$$\begin{aligned} u &= f\frac{Z\frac{dX}{dt} - X\frac{dZ}{dt}}{Z^2} = f\frac{ZU - XW}{Z^2} = f\frac{U}{Z} - x\frac{W}{Z} \\ v &= f\frac{Z\frac{dY}{dt} - Y\frac{dZ}{dt}}{Z^2} = f\frac{ZV - YW}{Z^2} = f\frac{U}{Z} - y\frac{W}{Z} \end{aligned} \tag{3.3}$$

where

$$U = \frac{dX}{dt}, V = \frac{dY}{dt}, W = \frac{dZ}{dt} \tag{3.4}$$

U, V and W represent the velocity of $\mathbf{P}$ in the 3D world coordinate system along the X, Y, Z axes respectively, while u and v represent the velocity of $\mathbf{p}$ projected from $\mathbf{P}$ onto the 2D image plane. When the brightness constancy constraint is satisfied, the motion flow vectors u and v will be equal to the optical flow at $\mathbf{p}$.

If we assume the surface where $\mathbf{P}$ lies belongs to a rigid body, we would be able to model its motion with the following rigid body motion model. More specifically, the velocity of $\mathbf{P}$ can be decomposed into two components – translational component and the rotational component. Let OXYZ be the Cartesian coordinate system of the 3D world. Let $T = (\tau_1, \tau_2, \tau_3)$ be the translational velocity of $\mathbf{P}$ relative to OXYZ and let $\omega = (\omega_1, \omega_2, \omega_3)$ be the rotational velocity of $\mathbf{P}$. τ_1, τ_2 and τ_3 are the translational velocities along the X, Y, Z axes, and ω_1, ω_2 and ω_3 are the rotational velocities along the X, Y, Z axes respectively. Then the velocity of $\mathbf{P}$ can be represented with the following equation:

$$\frac{d\mathbf{P}}{dt} = T + \omega \times \mathbf{OP} \tag{3.5}$$

Which expands to:

$$\begin{aligned} U &= \tau_1 + Z\omega_2 - Y\omega_3 \\ V &= \tau_2 + X\omega_3 - Z\omega_1 \\ W &= \tau_3 + Y\omega_1 - X\omega_2 \end{aligned} \tag{3.6}$$

Substituting (2) in (6) and then substitutisng the resulting expression in (3) results in the Longuet-Higgins and Pruzdny model equations [45]:

$$\begin{aligned} u &= -\frac{xy}{f}\omega_1 + \frac{f^2 + x^2}{f}\omega_2 - y\omega_3 + \frac{f\tau_1 - x\tau_3}{Z} \\ v &= -\frac{f^2 + y^2}{f}\omega_1 + \frac{xy}{f}\omega_2 + x\omega_3 + \frac{f\tau_2 - y\tau_3}{Z} \end{aligned} \tag{3.7}$$

The Longuet-Higgins and Pruzdny model equations represent the relationship between the 3D motion of a point on a rigid body, its relative depth and its projected optical flow. One thing worth notice is that if a set of 3D rigid motion parameters (T, ω) and a depth scalar Z satisfy equation (7), then (αT, ω, αZ) will also satisfy equation (7). This ambiguity of scale makes it only possible to recover the depth of a 3D point or its translational velocity up to a relative scale.

3.1.2 Estimating Three-Dimensional Screw Motions from Optical Flow

Equation 3.5 demonstrates the mathematical relationship between the 2D motion flow vectors and the 3D screw motions of an object in the scene, the camera focal length and the depth of the object. For the rest of this chapter, we assume the brightness constancy constraint is satisfied, thus the motion flow is equal to the optical flow. For each pixel in an image frame, we can establish two equations using its horizontal and vertical optical components u and v. Since all pixels that belong to the same rigid object share the same screw motion $\tau_1, \tau_2, \tau_3, \omega_1, \omega_2, \omega_3$ but potentially different depth values, we can see that we need at least 6 pixels on the same rigid body and their corresponding optical flow values to solve for the screw motion of the object, assuming the camera focal length is also unknown. More specifically, for 6 pixels on the same rigid body, we will have 12 unknowns, which includes 6 screw motion variables, 6 depth values, 1 focal length, and minus 1 ambiguity of scale.

We are only interested in solving for the relative 3D screw motion variables since the goal is to segment objects with different motions. As mentioned earlier, one rigid object has six instantaneous screw motion parameters . For every pixel that belongs to the rigid object, we can construct two equations as in 3.7, and this results in a large number of equations if we consider all pixels within the object mask. The problem is that equation

3.7 is non-linear due to the unknown pixel depth being the denominator in some of its terms, which poses significant challenge in optimizing for its solutions. However, if we can linearize these two equations using a known depth value, such optimization will become fast and straightforward. If the relative depth value of every pixel is known, we will be able to linearize equation 3.7 with respect to the screw motion parameters T and ω in the following form:

$$\begin{aligned} u &= a + \frac{b}{z} - \frac{cx}{z} - dy + ex^2 - fxy \\ v &= g + \frac{h}{z} - \frac{cy}{z} + dx + fy^2 + cxy \end{aligned} \tag{3.8}$$

Where a, b, c, d, e, f, and g denote unknown parameters that encapsulate arithmetic combinations of the six screw motion parameters, and z is the known relative depth of the pixel [28]. Each unknown parameter thus encodes a composite measure derived from the underlying screw motion parameters, reflecting specific algebraic relationships that represent the object's screw motions. We choose to use the relative depth here since estimating the relative depth values of pixels can be done accurately by the state-of-the-art monocular depth estimation models, while estimating the true depth of pixels is still an unsolved problem. Theoretically, either using relative pixel depth or using true pixel depth will have the same result in distinguishing different object motions. When solving for the object screw motion parameters, using relative pixel depth

3.2 Epipolar Geometry and Its Application in Motion Analysis

3.2.1 Epipolar Geometry

Epipolar geometry is a branch of geometry that focuses on understanding the intrinsic projective geometry between two different views captured by the camera [25]. It is fundamental in the field of computer vision and structure from motion (SfM), providing a mathematical framework to analyze how points in a three-dimensional space are projected onto two images from different perspectives. This framework is built upon the concepts of epipolar lines and epipolar planes, which are critical for determining correspondences between points in the two views. By leveraging these correspondences, it becomes possible to reconstruct the three-dimensional structure of the scene, analyze the ego-motion of the camera or the motion of any moving objects in the scene.

The fundamental concept in epipolar geometry is the epipolar plane, which is defined by the line connecting two camera centers (the baseline) and any point in space. This plane intersects each camera's image plane along a line, known as the epipolar line. For any given point in one image, its corresponding point in the other image must lie along the epipolar line. Figure 3.2 demonstrates the concept of epipolar plane and epipolar line by showing how a 3D point is projected onto two different camera planes (a) and how a 2D point on the camera plane is back projected into 3D space and then onto the camera plane of another camera.

Mathematically, epipolar geometry can be represented as fundamental matrix [25]. Given a set of corresponding points on two images from two different camera views, we can establish the following equation:

$$x'^{\intercal} F x = 0 \tag{3.9}$$

where F is a 3×3 matrix called fundamental matrix and x and x' are the 2D homogeneous coordinates of the two corresponding points. Once the fundamental matrix is obtained between two different camera views, we will be able to also recover the translation and rotation parameters between these two views using singular value decomposition. The translation and rotation parameters directly show the relative position or motion between the two camera views. Detailed proof can be found in [25].

3.2.2 Distinguishing Different Motions from Epipolar Geometry

We showed how epipolar geometry can be used to described the relationship between a 3D point and its 3D projection from 2 different cameras. Now imagine when there is only one camera which is moving, the same principle can be applied in this scenario if we consider the same camera at two different timestamps as two different camera views. In this way, we can infer the ego-motion of the camera as well as the motion of potential moving objects in the scene using the established point correspondences and epipolar geometry.

Between two camera views, one fundamental matrix normally can be recovered for each rigid motion if enough corresponding points can be established on the same rigid body. The fundamental matrix F has 7 degrees of freedom [25] and can be solved if more than

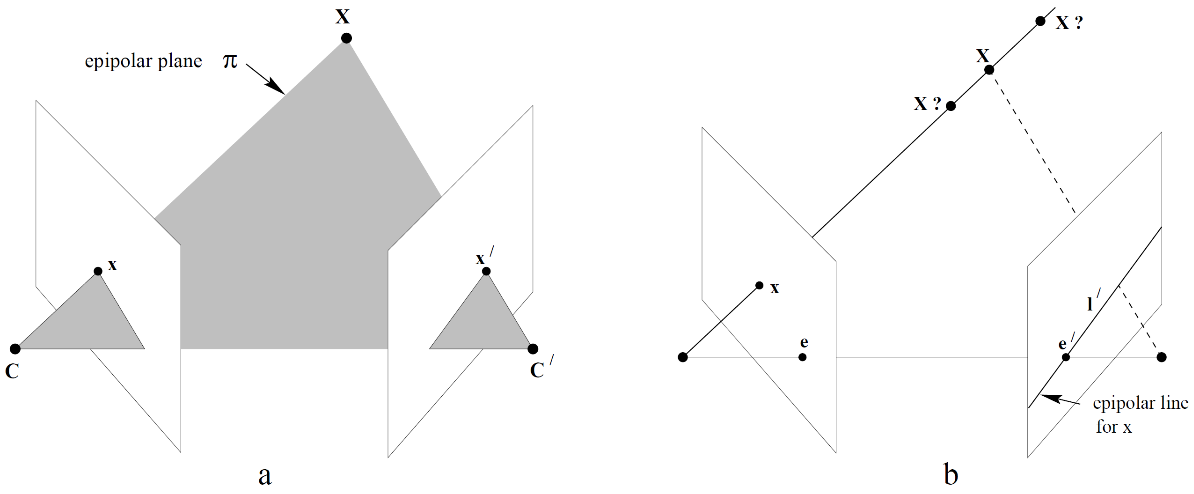

Figure 3.2: Epipolar Geometry [25]

7 pairs of corresponding points can be established on the same rigid body. Typically, a combination of seven-point algorithm or eight-point algorithm and RANSAC [22] is used to solve for the fundamental matrix by minimizing the algebraic error $x'^{\mathsf{T}}Fx$ after an overdetermined linear system of equations is established on a set of corresponding points.

Although algebraic error is the most widely adopted way to compute the fundamental matrix due to being fast and decently accurate, it is not always accurate in measuring how well the fundamental matrix fits the corresponding points, because it can be affected by the absolute values of homogeneous coordinates of the corresponding points, even after normalization. A more accurate way to evaluate how well the fundamental matrix and the corresponding points fit each other is the Sampson distance [25]:

$$d_{\text{Sampson}} = \frac{(x'^{T}Fx)^2}{(Fx)_1^2 + (Fx)_2^2 + (F^Tx')_1^2 + (F^Tx')_2^2}$$

The Sampson distance is the first order approximation of geometric error, which is the actual geometric distance between the corresponding points and the epipolar lines on the image plane we want to minimize. However, the geometric error is hard to minimize due to its non-linear nature, and the Sampson distance offers a closer approximation than the algebraic error, while also being faster to compute than geometric error.

If we compute one fundamental matrix for each individual rigid object between each frame pair of the video, we will be able to use these fundamental matrices to represent the motion of each individual object. For example, given a set of corresponding feature points P_A detected on an object A at frame m and the fundamental matrix F_A computed on these feature correspondences between frames m and n, we can compute the epipolar lines of these feature points at frame n. For object A, its feature points must lie closely to the epipolar lines on both frames m and n, because this is how its fundamental matrix is computed. However, if we look at another object B that is moving differently from object A, the corresponding feature points of object B between frames m and n (denoted as P_B and P'_B) will not be matched with the fundamental matrix of object A. That is, If we compute the mean Sampson distance using P_B and P'_B and the fundamental matrix of object A (F_A), it will most probably result in a much larger number than that of object A, indicating the point correspondences do not lie closely to the epipolar lines on either frame. If we see such a discrepancy, we will know that object A and B are moving differently between frame m and n. However, this method does have limitations: First, it only uses epipolar geometry to determine if there is relative motion between two objects. If the two objects are both moving on the epipolar plane of the two camera views, this method will not be able to recognize their relative motion since the mean Sampson distance of both objects will be close to zero. Second, it assumes the motion is rigid. If both objects have more non-rigid motions than rigid motions, both Sampson distances will be large.

3.3 Summary

In this chapter, we have discussed the fundamental theory of monocular motion segmentation, as well as two basic methods of monocular motion estimation from a moving camera. The first method uses the optical flow and the relative depth map to compute the relative screw motion between a rigid object and the camera. The second method uses corresponding feature points on a rigid object between two different camera views to compute a fundamental matrix, from which we can recover the object's translation and rotation parameters relative to the camera.

We also discussed how such motion estimation methods can be used to perform monocular motion segmentation. The ability to estimate the motions of individual objects in the scene is crucial for motion segmentation. When multiple objects are present in the scene, motion estimation techniques can be used to estimate the motion of each individual object. By analyzing each object's motions, we can determine which objects are moving in a similar pattern (most likely part of the background) and which objects are moving independently.

In the following chapter, we will discuss how we synergistically combine these traditional motion estimation methods with deep learning to achieve state-of-the-art zero-shot monocular dense motion segmentation without needing any training.

Chapter 4

Zero-Shot Monocular Motion Segmentation by Object Proposal Clustering

4.1 Methodology

This study propose a zero-shot monocular motion segmentation approach that uses both object appearance information and a combination of epipolar geometry and optical flow based motion models to perform in-the-wild motion segmentation without any assumptions of the motion or the scene [29].

The proposed motion segmentation pipeline first generates an initial segmentation of the background and all common objects in the scene using foundation models, and then tracks these objects throughout the whole video using an object tracker. For each object in each frame, we obtain a set of object-specific trajectory points, an optical flow mask and a depth map. We then compute two types of motion models for each object in the scene: one based on fundamental matrix fitting using point trajectories, and the other based on fitting optical flow and a depth map to our proposed parametric equations. By fitting each object's motion models on every other object and analysing the residuals of the model fitting, we are able to compute two pairwise affinity scores between every pair of objects and construct two motion affinity matrices for the two types of motion models respectively. Lastly, we fuse the two affinity matrices using co-regularized multi-view spectral clustering to obtain the final segmentation. Figure 4.1 shows a diagram of the motion segmentation pipeline.

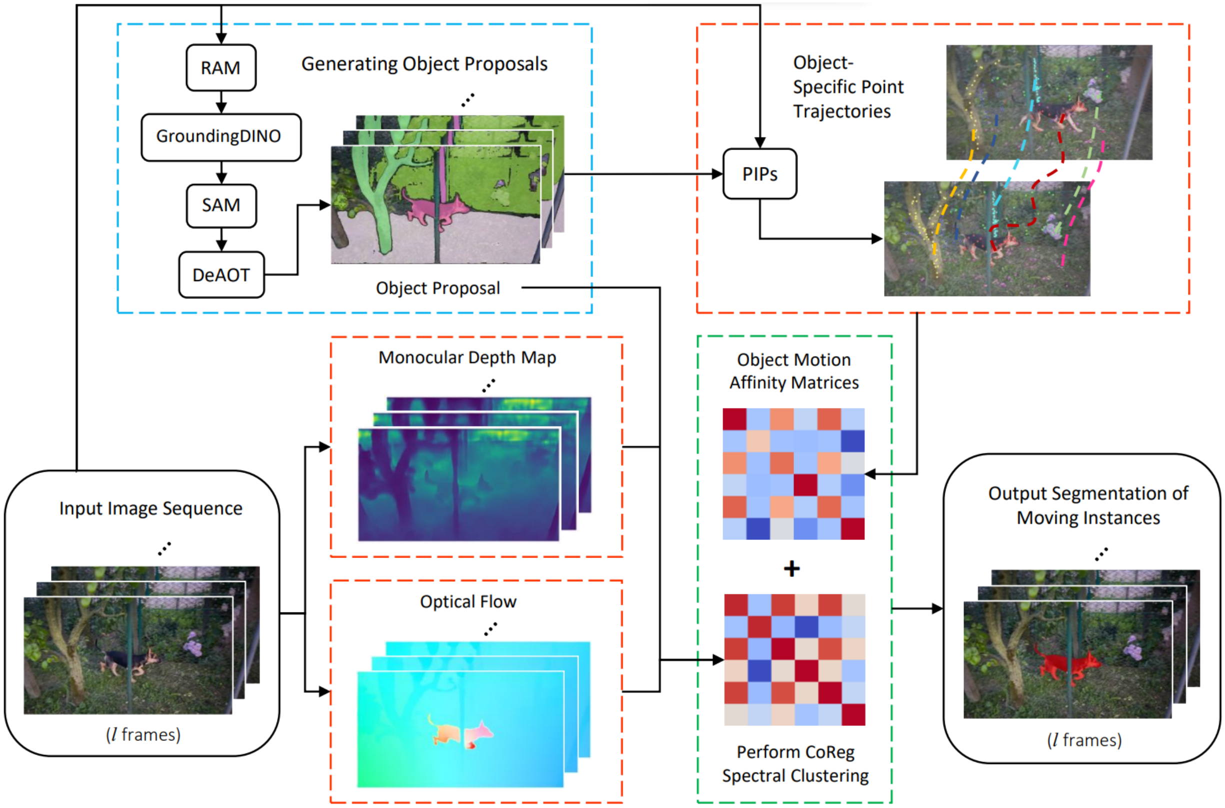

Figure 4.1: Our Motion Segmentation Pipeline. The motion segmentation method can be summarized to three main steps: 1) given a sequence of video frames, we produce an object proposal by automatically detecting, segmenting and tracking common objects in the video. 2) we compute object-specific point trajectories, optical flow and monocular depth maps for every frame. 3) we compute pairwise object motion similarity scores using two motion models (one based on point trajectories and the other based on optical flow and depth map), and use them to construct two motion affinity matrices. The two matrices are fused using multi-view spectral clustering to cluster objects into different motion groups.

4.1.1 Generating Object Proposals

In order to identify all motions in a video sequence at object level, we first identify every common object in the video and track their movements throughout the video. We achieve this by using the recent foundational models in object recognition (Recognize Anything Model)[86], detection (Grounding DINO model) [44] and segmentation (Segment Anything Model) [58], and a state-of-the-art object tracker (DeAOT) [84]. We adapt our preprocessing pipeline from Segment and Track Anything (SAMTrack) [13], which is an object segmentation and tracking framework based on the Grounding DINO model, Segment Anything Model (SAM) and the DeAOT tracker. SAMTrack allows the user to segment and track any specific objects in the video with a text prompt. To make our system fully automatic, we avoid using the user-defined text prompt by adding RAM at the beginning of our pipeline to automatically recognize any common objects in the video frame. In summary, our whole preprocessing pipeline consists of the following steps: 1) Use RAM to recognize any common objects in the first frame of the video; 2) Feed the output of RAM as a text prompt to the Grounding DINO model to obtain object bounding boxes; 3) Feed these bounding boxes to SAM to obtain an instance segmentation mask of the first frame. Non-max suppression was used to remove objects with an IoU score > 0.5 or with a mask area larger than half the image size; 4) Use the DeAOT tracker to track each object's mask throughout the entire video. In order to account for potential new objects entering the scene in the middle of the video, we split the video into multiple parts of l frames each and perform steps 1) to 4) on each part separately. The number l can be video-specific, for example, more dynamic videos with many objects entering scene in the middle of the video will benefit from a smaller l.

4.1.2 Object-Specific Motion Cues

Once we have an object proposal for every frame of the video, we will then obtain object-specific motion cues for every object in the object proposal. We propose to use point trajectories, optical flow and monocular depth map automatically generated by off-the-shelf networks as motion cues, in order to model objects' motions in two complementary ways.

Object-Specific Point Trajectories

A set of sparse point trajectories is generated for every object using PIPs [24]. PIPs is a state-of-the-art point tracker which tracks individual pixels given their initial locations in a video frame. A mixture of Shi-Tomasi [35] and K-Medoids [56] sampling method is used to obtain the initial pixels from each object as it showed good experimental results from previous works in similar tasks [58]. These tracked pixels can be used as object-specific feature points to fit fundamental matrices for every object in frame pairs to describe their motions. One limitation of PIPs is that does not handle occlusion well if the tracked video is more than 8 frames. To overcome this issue, we check for every point if it is inside its corresponding object's mask area every 8 frames. If not, we remove that point and sample a new point inside the object's mask. We also remove any point that is near the edge of the frame since the tracking accuracy of PIPs drops significantly in this case.

Object-Specific Optical Flow and Depth Map

We also generate a dense optical flow mask and a monocular depth map for every frame, from which we can extract object-specific optical flow vectors and depth maps. We use a state-of-the-art optical flow estimator [68] to obtain optical flow, and a state-of-the-art monocular depth estimator, DINOv2 [54], to extract the depth maps. We use monocular depth estimation to estimate the scene depth from a single frame since our goal is to perform motion segmentation from a moving monocular camera. DINOv2 outputs a relative depth map, which is sufficient for our application. Our experiment shows improved results when both optical flow and depth map are used to compute the motion model, comparing to only optical flow. We show how a depth map can be used to improve the motion model based solely on optical flow in the next section.

4.1.3 Motion Model Fitting

After obtaining object-specific point trajectories, optical flow vectors and depth maps, for each frame pair, we compute two motion models of each object based on epipolar geometry and optical flow respectively, to model its motion throughout the video. To compute the epipolar geometry based motion models using point trajectories, we compute a fundamental matrix of each object between every f frames by solving $p'^T F p = 0$ using the eight-point algorithm with RANSAC [22], where p and p' are the normalized 2D homogeneous coordinates of the same tracked point in the two frames. If a degenerate case is encountered for the fundamental matrix, we do not use it.

For the optical flow based motion model, we propose a modified version of the Longuet-Higgins and Pruzdny model equations [45], which model's the instantaneous screw motion of rigid objects at arbitrary depth. The original Longuet-Higgins and Pruzdny model equations establish a relationship between the optical flow, the instantaneous screw motion of rigid objects and the pixel depth value:

$$\begin{aligned} u &= -\frac{xy}{f}\omega_1 + \frac{f^2 + x^2}{f}\omega_2 - y\omega_3 + \frac{f\tau_1 - x\tau_3}{z} \\ v &= -\frac{f^2 + y^2}{f}\omega_1 + \frac{xy}{f}\omega_2 + x\omega_3 + \frac{f\tau_2 - y\tau_3}{z} \end{aligned} \tag{4.1}$$

where u and v are the optical flow vectors on the x and y axes, z is the pixel depth, f is the focal length of the camera, and τ_1, τ_2, τ_3 and ω_1, ω_2, ω_3 are the translational and rotational screw motions of the object. However, in practice, we often do not know the absolute pixel depth, thus we cannot use the complete model to compute the screw motion of the object. To deal with this issue, existing works often use a parametric motion to directly model the object motion from optical flow. For example, [47] uses a piecewise set of the following parametric equation with 12 parameters to direct fit the optical flow field, but their motion model is not theoretically correct and cannot handle scenes with large depth variations. [7, 9] uses a less complex parametric motion equation to model the rotation compensated optical flow angle field, but it requires a known camera matrix, which is also not practical. To have a better motion model that is both theoretically sound and does not need camera information, we propose to linearize the Longuet-Higgins and Pruzdny equations using the monocular depth map generated from DINOv2. With known relative pixel depths, (4.1) can be rewritten as the following linear parametric equations, as previously mentioned in 3.1.2:

$$\begin{aligned} u &= a + b\frac{1}{z} - c\frac{x}{z} - dy + ex^2 - fxy \\ v &= g + h\frac{1}{z} - c\frac{y}{z} - dx + exy + fy^2 \end{aligned} \tag{4.2}$$

Although the depth map from DINOv2 is relative, it can still be used to model object motion in this case – our goal is to cluster the objects' motions into different motion groups instead of computing each object's screw motion, so we do not need to care about the uncertainty of scale. For convenience, we still refer to this motion model as our "optical flow motion model", although it uses both optical flow vectors and pixel depth maps.

4.1.4 Constructing Affinity Matrices

After all fundamental matrices and optical flow motion models are computed, each object will have a fundamental matrix between every p frames and an optical flow motion model between every two frames. By fitting every object's trajectory points, optical flow vectors and depth maps to every other object's fundamental matrix and optical flow motion model on the same frame pair, we can obtain the residuals of every object to all other objects' motion models respectively. We use Sampson distance [25] as the residual for the fundamental matrix and mean squared error for the optical flow motion model. Assuming there are k objects in the scene, for the i-th object at the m-th frame pair, we obtain the following residual vectors under the fundamental matrix and optical flow motion models:

$$\boldsymbol{r}_{o_i}^{\;m} = [r_{o_{i,1}}^{\;m}, r_{o_{i,2}}^{\;m}, ..., r_{o_{i,k}}^{\;m}],$$

$$\boldsymbol{r}_{f_i}^{\;m} = [r_{f_{i,1}}^{\;m}, r_{f_{i,2}}^{\;m}, ..., r_{f_{i,k}}^{\;m}]$$

where $r_{o_{i,k}}^{\;m}$ is the mean residual for fitting the parametric motion model of object i on the optical flow vectors of object k between frames m and $m+1$, and $r_{f_{i,k}}^{\;m}$ is the mean Sampson error for fitting the fundamental matrix of object i on the trajectory points of object k between frames m and $m+p$. We construct two affinity matrices encapsulating the pairwise motion affinities between each pair of objects using a modified version of ordered residual kernal (ORK) [14]. Specifically, for each object, we sort its residual vectors in ascending order and define a threshold to select the smallest t-th residual as inliers. We define $\boldsymbol{c}_i = \{0, max(t - n_i, 0)\}^K$ as an inlier score vector whose length is the same as the number of objects K. n_i is the rank of object k in the residual vector of object i, penalizing different inlier distributions between objects. The pairwise motion affinity between objects i and j can thus be computed as $\boldsymbol{a}_{ij} = \boldsymbol{c}_i^\intercal \boldsymbol{c}_j$, which denotes a weighted co-occurrence score between two objects as inliers of all motion models. Our proposed weighted ORK is robust to outliers and makes the affinity matrix more adaptive to different scenes by reducing the need to set scene specific inlier thresholds.

4.1.5 Co-Regularized Multi-view Spectral Clustering

After constructing the affinity matrices, we normalize them using row normalization [77] and adapt co-regularized multi-view spectral clustering [39] to fuse the two affinity matrices together. With the number of motion groups in the scene given as an input, we are able to obtain the final clustering of moving objects. Co-regularized multi-view spectral clustering uses an regularization term to encourage consensus between different views and is shown to perform well on fusing multiple geometric models for a consistent representation of motion information [82].

4.2 Experiments

Our method is tested on three benchmarks: DAVIS-Moving, YTVOS-Moving and the extended KT3DMoSeg. We first briefly introduce these datasets, then show both quantitative and qualitative comparisons between our method and other state-of-the-art methods. Lastly, we present an ablation study to compare the performance of different individual motion models and the fused motion model.

Method	**Training**	**DAVIS-Moving**		
		Pu	**Ru**	**Fu**
MoSeg [16]	Supervised	**78.30**	78.80	78.10
Raptor [52]	Supervised Features	75.90	79.67	75.93
RigidMask [83]		59.03	49.89	50.01
Ours	Zero-Shot (no training)	78.27	**81.58**	**79.40**

Table 4.1: Quantitative results of our method and state-of-the-art motion segmentation methods on the DAVIS-Moving validation dataset. The best result for each metric is in bold and the second best result is in underscore. The quantitave results of the models being compared are directly cited from Neoral's work [52].

4.2.1 Datasets

DAVIS-Moving and YTVOS-Moving are both proposed by [16] as datasets for generic instance motion detection and segmentation. DAVIS-Moving and YTVOS-Moving are subsets of the DAVIS 17 dataset [57] and the YTVOS dataset [81], where all moving instances in the video sequence are labeled and no static objects are labeled. These two recently proposed datasets are very challenging due to their diverse object classes, occlusions and non-rigid motions. To the author's knowledge, the DAVIS-Moving and YTVOS-Moving datasets are the most recent datasets solely focused on motion segmentation.

In addition to these two datasets, we also evaluate our method on an extended version of the KT3DMoSeg dataset. The original KT3DMoSeg dataset [82] is designed to test point trajectory based motion segmentation methods on complex road scenes. It contains manually corrected point trajectories on selected moving instances in road scenes and includes significant degenerate motions and depth variations. In order to test the performance of our method in such environments, we extend the KT3DMoSeg dataset by adding a pixel-level segmentation mask to every moving instance in the scene. We refer to this extended dataset as the KT3DInsMoSeg dataset in the following sections.

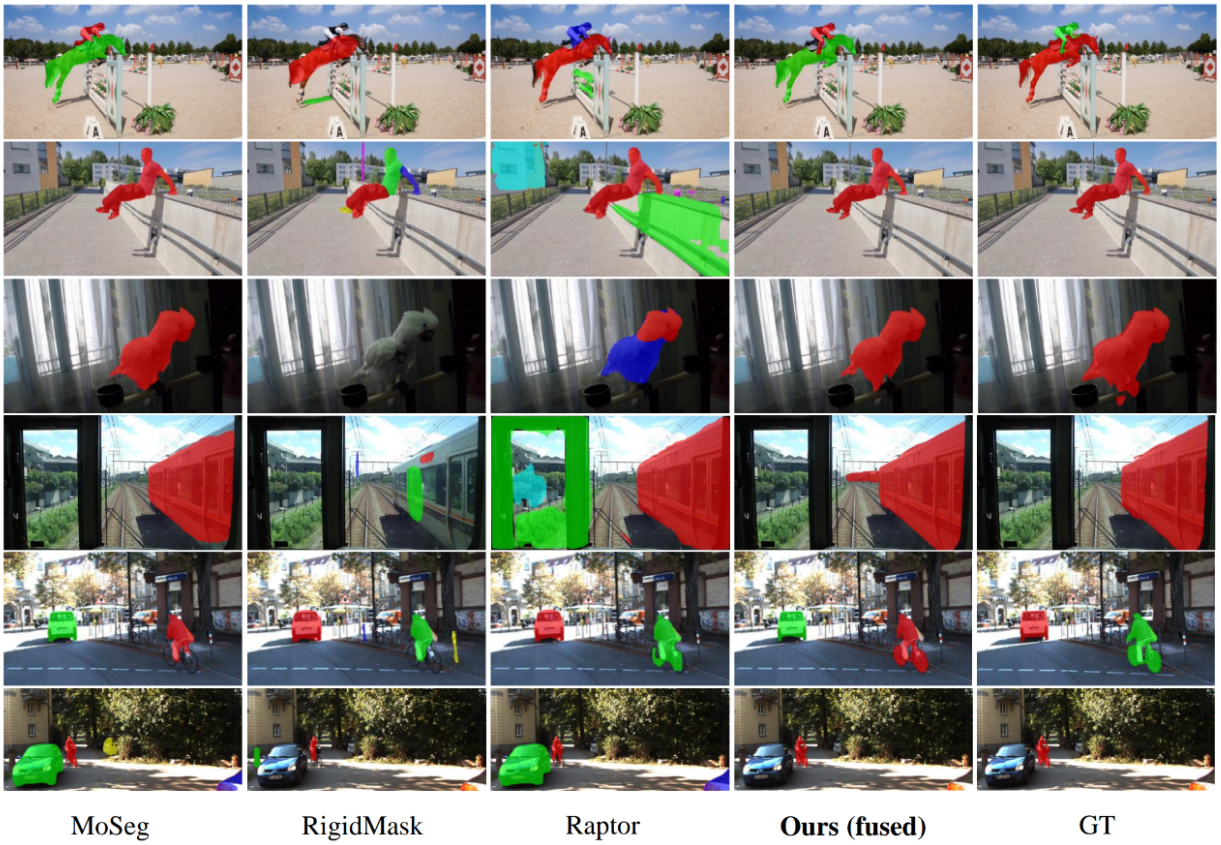

Figure 4.2: Qualitative results of different methods on DAVIS-Moving (row 1, 2), YTVOS-Moving (row 3, 4) and the extended KT3DMoSeg (row 5, 6) datasets. MoSeg often mistakenly label static objects as dynamic when there is degenerate camera motion. RigidMask fails to detect or coherently segment objects with non-rigid motions. Similarly, Raptor also has these problems, although to a lesser extent overall. Our method, despite not being trained on any data, performs well when facing these challenges.

4.2.2 Results

To evaluate our method, we adopt the *precision* (Pu), *recall* (Ru) and *F-measure* (Fu) proposed in [16] which penalizes false positives. The *F-measure* combines both *precision* and *recall* and indicates the method's overall performance. Table 4.1, 4.2 and 4.3 show quantitative results of our method and other state-of-the-art methods on the three benchmarks. In selecting benchmark methods for comparison with our proposed motion segmentation approach, two primary criteria were used to ensure relevance and reproducibility.

Method	**Training**	**YTVOS-Moving**		
		Pu	**Ru**	**Fu**
MoSeg [16]	Supervised	**74.50**	**66.40**	**66.38**
Raptor [52]	Supervised Features	64.43	60.94	60.35
RigidMask [83]		29.88	17.88	18.70
Ours	Zero-Shot (no training)	64.12	61.10	60.62

Table 4.2: Quantitative results of our method and state-of-the-art motion segmentation methods on the YTVOS-Moving validation dataset. The best result for each metric is in bold and the second best result is in underscore. The quantitave results of the models being compared are directly cited from Neoral's work [52].

Method	**Training**	**KT3DInsMoSeg**		
		Pu	**Ru**	**Fu**
MoSeg [16]	Supervised	63.73	78.24	66.85
Raptor [52]	Supervised Features	71.52	**88.27**	**75.82**
RigidMask [83]		65.14	83.34	70.91
Ours	Zero-Shot (no training)	**72.93**	71.02	71.89

Table 4.3: Quantitative results of our method and state-of-the-art motion segmentation methods on the proposed KT3DInsMoSeg dataset. The best result for each metric is in bold and the second best result is in underscore.

First, priority was given to recently proposed methods with publicly available source code, enabling straightforward implementation and verification. Second, for recently proposed methods without publicly available source code, we only select those having been tested on the DAVIS-Moving and YTVOS-Moving datasets, which are the datasets we use for evaluation due to the reasons previously mentioned.

Although we did conduct any training, our method achieves the best result on the DAVIS-Moving dataset, surpassing even the fully-supervised method [16]. On the YTVOS-Moving dataset, our method achieves the second best result, surpassing Raptor [52] with a slight 0.27 % higher Fu score. On the KT3DInsMoSeg dataset, our method achieves the second best result only after Raptor.

We also qualitatively compare our method with these methods and we show the results in Figure 4.2. MoSeg [16] produces good results on segmenting moving objects from DAVIS-Moving and YTVOS-Moving datasets, but it fails to identify the motions of cars and bikes in the two bottom rows when the scene contains degenerate motions (e.g., forward or backward camera motion), as such scenario is not part of its training dataset, whereas our method successfully identifies the cars as being static and the bike as being in the same motion group as the person. RigidMask [83] fails to produce coherent segmentation of the person on the second row, and also fails to detect the motions of the parrot, the train and the bicycle in the middle rows, whereas our method successfully detects and segments all of them coherently. Although RigidMask [83] performs relatively well on the KT3DInsMoSeg dataset due to most motions in the dataset being rigid, it does not produce a segmentation mask as accurate as our method for objects with complex contours (e.g., person). Raptor [52] is able to detect most objects in the scene thanks to its powerful semantic backbone, however, it still over-segments non-rigid objects such as the parrot. Similar to other methods, It also falsely identifies the static cars on the bottom row as being dynamic.

4.2.3 Ablation Study

We present both quantitative (Table 4.4) and qualitative (Fig. 4.3) comparisons between different individual motions models and the fused motion model for their performances on the three benchmarks.

We find that on both DAVIS-Moving and KT3DInsMoSeg datasets, our model fusion technique (fused) is able to significantly boost the Fu score comparing to using only a single model (27.28% and 28.52 % respectively), while on YTVOS-Moving, the Fu score only had a 9.79 % increase. Upon further inspection, we found that this can be attributed to some motion labels in the YTVOS-Moving dataset actually being mostly static throughout the video sequence. Since our method clusters moving objects purely using motion cues, it groups these objects together with the background as expected. Additionally, the YTVOS-Moving dataset also contains some videos with significant camera zooming, which violates the geometric assumptions of both our motion models. Our motion model fusion technique

Method	**DAVIS-Moving**			**YTVOS-Moving**			**KT3DInsMoSeg**		
	Pu	**Ru**	**Fu**	**Pu**	**Ru**	**Fu**	**Pu**	**Ru**	**Fu**
Fused	**78.27**	81.58	**79.40**	**64.12**	61.10	**60.62**	**72.93**	71.02	**71.89**
OC+Depth	71.53	75.66	73.18	63.54	58.94	56.06	48.04	61.54	49.26
OC	58.25	59.22	57.08	61.79	54.64	53.74	36.44	39.97	34.78
Trajs	65.99	75.51	68.47	54.67	52.92	50.05	42.31	**73.66**	45.24
Baseline	43.17	**86.24**	52.12	48.49	**73.01**	50.82	38.97	70.97	43.37

Table 4.4: Quantitative ablation study of the motion segmentation results from using different motion cues. Baseline results are obtained by directly using the raw object proposals as the final motion segmentation mask. Bold numbers are the best results and the underscored numbers are the second best results.

is able to achieve better results than any single motion model on all three datasets, showing its effectiveness.

We also show the performance comparison between the motion model purely based on optical flow (OC) and the motion model based on a combination of optical flow and relative monocular depth information (OC + Depth). For optical flow based motion model (OC), we use the optical flow motion model of [47], which is a state-of-the-art unsupervised motion segmentation method using only optical flow as input. Their motion model is a 12-parameter quadratic parametric equation modified from the Longuet-Higgins and Pruzdny model equations, by modelling the unknown depth information as parameters. Results show that the motion model based on a combination of optical flow and depth (OC + Depth) outperforms OC by a large margin in all three metrics on both DAVIS-Moving and KT3DInsMoSeg. However, on YTVOS-Moving, the difference in performance between these two motion models is not significant. This shows that the depth information is not a key limiting factor for segmenting moving objects in this dataset. In fact, besides some labeled objects being mostly static, the YTVOS-Moving dataset also contains significant occlusions in many scenes and hard-to-detect objects like camouflaged animals. It is likely that these factors outweigh the unknown depth information in preventing the method from generating accurate motion segmentation.

Both point trajectory based (Trajs as in Table 4.4) and optical flow based (OC as in Table 4.4) motion models perform poorly on the KT3DInsMoSeg dataset, due to sig-

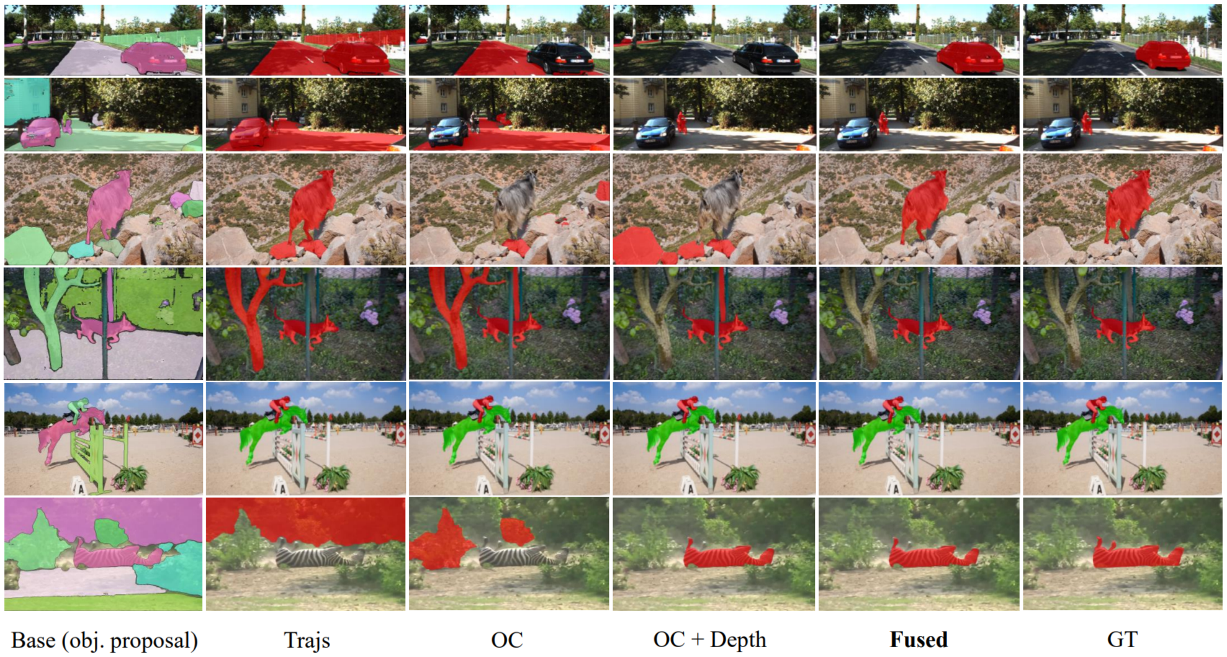

Figure 4.3: Qualitative ablation study between motion models based on different motion cues. Pure optical flow based motion model (OC) suffers on scenes with objects at varying depths. Combining optical flow with depth information (OC + Depth) only alleviates this problem to some extent. Pure point trajectory based motion model (Trajs) suffers from motions near the epipolar plane and inaccurate trajectory estimation. Motion model fusion effectively mitigates these problem by combining the advantages of both motion models and outperforms any single model.

nificant motion degeneracy (e.g., forward motion) and depth variations on road scenes. Incorporating depth information in this case proves to be an effective way to reduce motion ambiguity for the optical flow based motion model, boosting its F-score from 34.78% to 49.26%. Fusing the combined OC + Depth motion model with the point trajectory motion model based on epipolar geometry significantly enhances the performance in this case as well.

4.3 Summary and Future Work

In this study, we propose the first zero-shot approach to solving the problem of instance motion segmentation from a moving monocular camera. The proposed method combines the advantages of both deep learning foundation models and geometric approaches, which results in a zero-shot motion segmentation approach that performs motion model fusion on object proposals. Performance comparisons between the fused motion model and each individual motion model demonstrate significant improvements with the fused motion model, highlighting the effectiveness of the motion model fusion technique. Although this method is a zero-shot method, experimental results show that it surpasses most state-of-the-art methods and highly competitive with others. Future research will investigate in incorporating additional motion models, such as the trifocal tensor, to further enhance the motion segmentation performance.

Two primary limitations are identified with the proposed zero-shot motion segmentation method. First, it requires the number of distinct moving objects to be known to achieve optimal results. This limitation stems from the usage of spectral clustering, which, akin to many clustering methods, necessitates the input of the exact number of clusters to yield the best performance. This is not practical, since we often do not know the number of motions in the scene and need the model to discover such information by itself. Second, the method suffers from slow inference speed since it needs to integrate multiple deep learning models and perform computations across several stages. This multi-stage process inherently prolongs the inference time, presenting a significant drawback in scenarios where fast processing is required.

In order to address the first limitation regarding the determination of motion clusters, a novel model selection technique specifically designed for spectral clustering-based motion segmentation is proposed. This technique aims to autonomously determine the number of distinct motions within a scene. The overview, implementation, and comparative analysis of this model selection technique will be comprehensively discussed in the following chapter. Although this advancement significantly mitigates one of the key challenges, addressing the limitation of inference speed remains an open area for future research.

Chapter 5

A Unified Model Selection Technique for Motion Clustering

5.1 Motivation

Currently, motion segmentation is still a challenging problem when a moving camera is present, due to unknown camera motion. One popular technique to solve the motion segmentation problem in such scenario is to perform spectral clustering on motion affinity matrices constructed with motion models [11, 76, 41, 53, 82, 36, 79, 30, 42]. These methods typically take manually corrected point trajectories as input and build custom motion affinity matrices using one or more types of motion cues such as geometric models, spatio-temporal similarities or optical flow. Recently, spectral clustering based methods have shown remarkable results in segmenting motions in challenging dynamic environment containing significant motion degeneracy and complex scene structures [82, 36, 79, 42, 30], largely thanks to its ability of synergetically fusing multiple types of motion cues together. However, all of these methods cannot automatically infer the number of motions present in the scene (i.e., model selection) and rely on user input for such information. [11, 76, 41, 53] do propose model selection techniques, but those techniques are specifically suited for their respective methods, which do not perform well in complex dynamic scenes. To address this issue, we propose a general unified model selection technique by combining the strengths of multiple existing criteria [31], to automate the model selection process for the current spectral clustering based motion segmentation methods relying on either single or multiple types of motion affinities.

5.2 Methodology

We first briefly introduce the motion segmentation method being used as a foundation and baseline for our model selection technique, then discuss the proposed model selection technique in detail.

5.2.1 Motion Segmentation Pipeline

We use our previously proposed motion segmentation method [30] as the baseline. [30] performs motion segmentation by clustering different objects into different motion groups according to their pairwise motion similarities. More specifically, it first generates an object proposal for every frame of the video sequence denoting all common objects present in the scene, using a combination of off-the-shelf object recognizer, detector, segmentor and tracker. After all the potential objects in the video are segmented and tracked, object-specific point trajectories and optical flow mask for each labeled object in the video are generated as motion cues. From these two types of motion cues, two robust affinity matrices are constructed to encode the pairwise object motion affinities throughout the whole video using epipolar geometry and the optical flow based parametric motion model. Finally, co-regularized multi-view spectral clustering is used to fuse the two affinity matrices and obtain the final clustering. Figure 5.1 shows a diagram of this motion segmentation pipeline. This method achieves state-of-the-art results on the challenging KT3DMoSeg dataset by fusing multiple motion models together using multi-view spectral clustering, similar to other recent methods. Therefore, it is an ideal baseline to evaluate our model selection method.

5.2.2 Model Selection

We propose a general unified model selection method by combining four widely used model selection methods, i.e., the silhouette score [62], eigengap heuristic [77], Davies-Bouldin index [17] and Calinski-Harabasz index [12], to obtain an improved accuracy in determining the number of motion groups in the scene. We choose to use these four methods since they are all widely used criteria to evaluate the quality of clustering as well as to determine the optimal number of clusters. Given a motion affinity matrix, we first compute a confidence score for each criterion on a range of possible number of motions that may be present in the scene, we then compute the average of all four confidence scores corresponding for every possible number of motions, and select the one with the the highest confidence as the number of clusters to perform spectral clustering. We briefly introduce

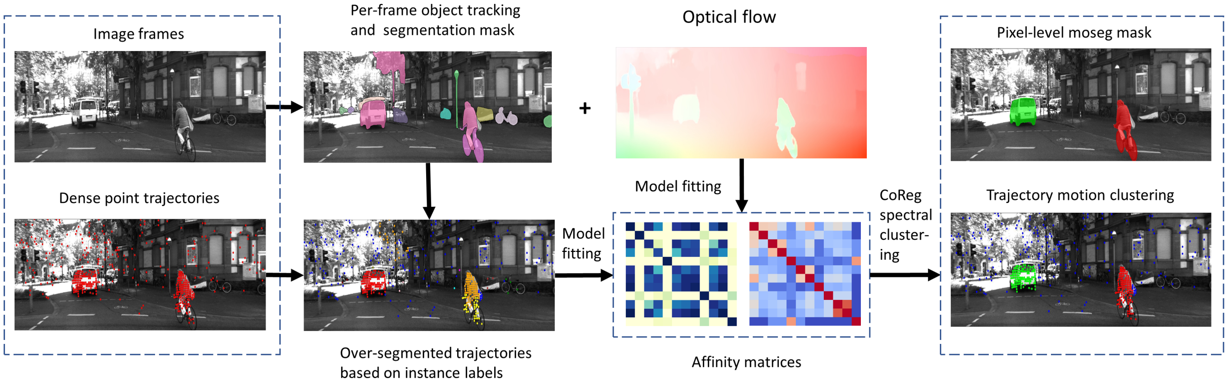
Image frames
Dense point trajectories
Per-frame object tracking and segmentation mask
Over-segmented trajectories based on instance labels
Optical flow
+
Model fitting
Model fitting
Affinity matrices
CoReg spectral cluster-ing
Pixel-level moseg mask
Trajectory motion clustering

Figure 5.1: Motion Segmentation Pipeline Used to Evaluation Model Selection Techniques. Given a sequence of video frames, 1) generate an object proposal for every frame, 2) obtain object-specific point trajectories and optical flow as two types of motion cues, 3) construct two motion affinity matrices using pair-wise object motion affinities, 4) perform co-regularized spectral clustering on the two motion affinity matrices to obtain the final segmentation

these four model selection criteria and further discuss our proposed method in the following sections.

Silhouette Score

The silhouette score measures how closely related each sample is to other samples in the same cluster comparing samples in other clusters. A higher silhouette score indicates higher similarity among samples within each cluster and lower similarity among samples in different clusters, hence better clustering quality. The mean Silhouette score for the clustering can be written as follows:

$$S = \frac{1}{N}\sum_{i=1}^{N}\frac{b(i)-a(i)}{\max(a(i),b(i))} \tag{5.1}$$

where N is the total number of samples, $a(i)$ is the mean distance between sample i and all other points in the same cluster, and $b(i)$ is the smallest mean distance between sample i and any other points in any other cluster, representing the separation from neighboring clusters. Silhouette score has a range between -1 and 1.

Eigengap Heuristic

Eigengap heuristic is a heuristic method for selecting the optimal number of clusters in clustering methods. According to the matrix perturbation theory [66], if the eigengap of affinity matrix's graph Laplacian is larger, then the subspaces spanned by its corresponding eigenvectors will be closer to being ideal. Let λ_i and λ_{i+1} be two consecutive eigenvalues of the Laplacian matrix of the affinity matrix, their eigengap is:

$$\delta_i = |\lambda_{i+1} - \lambda_i| \tag{5.2}$$

Let N be the total number of samples in the dataset, $\delta_1, ..., \delta_{N-1}$ is then the set of all possible eigengap values, and the ideal number of clusters K can be derived as follows:

$$K = argmax(\delta_i) \tag{5.3}$$

The Laplacian matrix L can be computed computed as $L = D - A$, where A is the affinity matrix and D is the degree matrix of the affinity matrix. More details can be found in the work of Luxburg [77].

Davies-Bouldin Index

Davies-Bouldin index is another quantitative measure of the clustering quality with similar intuition as the silhouette score of minimizing the within cluster distances and maximizing the between cluster distances. The Davies-Bouldin index can be written as the following formula:

$$DB = \frac{1}{N} \sum_{i=1}^{N} \max_{i \neq j} \frac{d(i) + d(j)}{D(c_i, c_j)} \tag{5.4}$$

where DB is the Davies-Bouldin index of the clustering, N is the number of clusters, $d(i)$ and $d(j)$ are the within-cluster distances of cluster i and it's most similar cluster j, and $D(c_i, c_j)$ is the distance between the centroids of cluster i and j. A lower DB score means better clustering quality.

Calinski-Harabasz Index

Calinski-Harabasz Index (also known as Variance Ratio Criterion) evaluates the clustering quality by estimating the ratio between ”between cluster variance” and ”within cluster variance". It can be described with the following formula:

$$CH(K) = \frac{\sum_{k=1}^{K} n_k \cdot D(c_k, c)/(K-1)}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} D(x_i, c_k)/(N-K)} \tag{5.5}$$

where CH(K) is the Calinski-Harabasz index for cluster K, n_k is the number of samples in cluster K, $D(c_k, c)$ is the distance between the centroid of cluster K and the centroid of all samples, and x_i is a sample in cluster K. A higher CH score indicates better clustering quality.

Combining Different Model Selection Criteria

We propose to combine the above four different model selection criteria by first computing a confidence score for each criterion on the motion affinity matrix for a range of possible number of motions that may be present in the scene, then selecting the number with the highest average confidence score as the number of motion groups present in the scene, and use this as the number of clusters to perform spectral clustering.

To calculate the above model selection metrics given a motion affinity matrix, we first need to transform the affinity matrix into a "distance matrix", due to the fact that the silhouette score, Davies-Bouldin index and Calinski-Harabasz index operate on distances among samples and clusters, instead of their similarities. Since all motion affinity matrices are normalized (i.e., having pairwise object motion affinity values between 0 and 1), we simply compute the pairwise object motion distance as $1 - affinity$. Then, we use this distance matrix to compute the normalized confidence score corresponding to each of the three criteria. Each normalized confidence score is valued between 0 and 1 with higher value indicating higher confidence. For eigengap heuristic, since it is not a quantitative measurement of the clustering quality, we compute its confidence score by checking how close the current number of motion clusters is to the optimal number of motion clusters (the one with the largest eigengap). Since we have a predefined range of how many motions may be present in the scene, it is easy to compute a normalized confidence score for eigengap heuristic in the same way as other criteria.

The above method is works for automatic model selection given a single motion affinity matrix. In cases of multiple multiple affinity matrices, we propose to first add these affinity matrices together, then perform row normalization [77] to obtain a normalized fused affinity matrix. We then perform the same procedure as above to infer the optimal number of motions using the fused affinity matrix.

5.3 Experiments

We evaluate our model selection method on the KT3DMoSeg dataset [82]. We chose this dataset for evaluation since it is a challenging monocular motion segmentation dataset proposed recently, focusing on real world scenes with strong motion degeneracy and motion parallax. The dataset contains manually corrected point trajectories obtained from an optical flow tracker on 22 video sequences selected from the KITTI dataset [23]. Each video sequence contains 2 to 5 different motion groups. Our evaluations are based on three criteria: 1) The mean squared error (MSE) of each method in predicting the number of motions; 2) The percentage of video sequences each method succeeds in predicting the exact number of motions correctly; 3) The overall motion segmentation error rates of different model selection techniques, versus that achieved by the baseline motion segmentation pipeline given the groundtruth number of motion clusters. The overall motion segmentation error rate is computed as the average error rate of all 22 sequences in the dataset, and the error rate of each sequence is computed as the percentage ratio between the number of wrongly clustered trajectories and the total number of trajectories in the sequence. This metric is adopted from Xu's work [82].

The motion segmentation pipeline computes two motion affinity matrices using epipolar geometry and optical flow respectively. We evaluate our motion selection method both individually on each of the two matrices, and on the fused affinity matrix. The fused affinity matrix is computed by taking the element-wise mean of the two matrices.

We also compare our proposed method of combining different model selection criteria with a consensus voting method and random guessing. The consensus voting method chooses the most frequent optimal number of motion clusters computed by all four criteria. If there is not a most frequent number, it chooses the smaller median value. The random guessing method simply uses a random number between 2 and 5 (inclusive) as the number of motions for each video sequence.

Table 5.1 shows the mean squared errors of different model selection methods on different motion affinity matrices. Our proposed method (Average) achieves the best overall result in predicting the number of motions using the fused affinity matrix, followed by the consensus voting method and the silhouette method.

Table 5.2 shows the accuracy of predicting the exact number of motions from different model selection methods on different motion affinity matrices. Silhouette score achieves the best result in terms of correctly predicting the exact number of motions in the scene. Our proposed method (Average) achieves the second best result.

Table 5.3 shows the final motion segmentation error rate of the motion segmentation

Methods	**Aff. F**	**Aff. OC**	**Fused Aff.**	**Avg. MSE**
Silhouette	1.364	**1.136**	1.091	1.197
Eigengap	1.318	1.455	1.636	1.470
DB	**1.091**	1.818	1.500	1.470
CH	1.364	1.318	1.227	1.303
Random	3.909	2.455	3.091	3.152
Voting	**1.091**	1.455	**1.046**	1.197
Average	**1.091**	1.364	1.091	**1.182**

Table 5.1: MSE of different model selection methods on different motion affinity matrices (lower is better). Aff. F is the motion affinity matrix obtained using epipolar geometry, Aff. OC is the motion affinity matrix obtained using optical flow, and Fused Aff. is the fused motion affinity matrix by taking the mean of the affinity scores of these two matrices

Methods	**Aff. F**	**Aff. OC**	**Fused Aff.**	**Avg. Acc.**
Silhouette	**54.55**	54.55	59.09	**56.06**
Eigengap	45.45	**59.09**	40.91	48.48
DB	54.55	31.82	40.91	42.42
CH	**54.55**	31.82	**68.18**	51.52
Random	31.82	31.82	27.27	30.30
Voting	54.55	40.91	59.09	51.52
Average	**54.55**	45.45	63.64	54.54

Table 5.2: Prediction accuracy of different model selection methods on different motion affinity matrices (higher is better).

pipeline using different model selection methods. The motion segmentation error rate is computed as classification error [82] of the input point trajectories. Our proposed method achieves the best results on two out of three types of motion affinity matrices, close to the baseline which takes the groundtruth number of motions as input. It is worth noting that, while the overall average error rate of the proposed averaging method (average) across all

Methods	**Aff. F**	**Aff. OC**	**Fused Aff.**	**Avg. Error**
Silhouette	15.99	**19.68**	12.78	16.16
Eigengap	16.36	25.01	16.47	19.28
DB	14.70	26.16	14.11	18.32
CH	18.03	26.88	12.09	19.01
Random	27.05	26.08	21.54	24.89
Voting	15.06	24.01	12.04	17.04
Average	**13.89**	20.59	**12.03**	**15.50**
Baseline	9.86	13.47	5.78	9.71

Table 5.3: Overall motion segmentation error rates of different model selection methods vs. the error rate obtained from known groundtruth number of motions (lower is better)

three distinct motion affinity matrices is lower compared to the consensus voting method, the error rate observed with the proposed method on the fused motion affinity matrix is very similar to that of the consensus voting method (12.03% versus 12.04%). Further statistical analysis may be required to investigate the effectiveness of these two methods in this specific case. The silhouette method and the consensus voting method are the second and third best methods, indicating their strengths as well, which is consistent with the results in Tables 5.1 and 5.2.

To further investigate the strengths and weaknesses of our method, we also analyze the evaluation results in more detail by comparing the performance of each method on sequences containing different numbers of motions. Out of the 22 sequences, 12 sequences contain 2 motion groups, 4 sequences contain 3 motion groups, 5 sequence contains 4 motion groups and 1 sequence contains 5 motion groups. We show the MSE and the overall motion segmentation error rate of each method on sequences containing each number of motions in table 5.4 and table 5.5 respectively. The results are evaluated using only the fused affinity matrix since the best motion segmentation results are usually obtained by fusing both affinity matrices together, thereby making the fused matrix more important and useful.

Our proposed method performs well when there are only 2 motion groups in the sequence, which accounts for around half of the dataset. For sequences containing 3 or 5 motion groups, our method also performs decently well, being above average. However, for sequences containing 4 motion groups, our method does not perform well. In fact, most

Methods	**Number of Motions**			
	2	**3**	**4**	**5**
Silhouette	0.00	1.75	3.20	1.00
Eigengap	0.33	0.75	4.0	9.00
DB	1.167	3.25	1.00	1.00
CH	0.75	1.50	2.40	0.00
Voting	0.00	1.50	3.20	1.00
Average	0.00	1.75	3.20	1.00
Avg. MSE	0.375	1.75	2.83	2.17

Table 5.4: MSE of different model selection methods on different numbers of motions. Avg. MSEs are computed using all 6 methods. Evaluated on the fused motion affinity matrix only.

Methods	**Number of Motions**			
	2	**3**	**4**	**5**
Silhouette	6.10	20.03	23.43	10.52
Eigengap	10.74	24.09	22.51	24.61
DB	10.40	20.44	18.67	10.52
CH	7.95	17.72	17.75	10.96
Voting	6.10	18.21	21.67	10.52
Average	6.10	18.16	21.67	10.52
Avg. Error	7.90	19.78	20.95	12.94
Baseline	3.31	8.23	13.75	6.04

Table 5.5: Overall error rates of different model selection methods on different numbers of motions. Avg. Errors are computed using all 6 methods. Evaluated on the fused motion affinity matrix only.

methods do not perform well on these sequences. This is mostly likely due to the fact that these video sequences generally contain more challenging scenes (e.g., more motion degeneracy or motion parallax) for the motion segmentation algorithm, resulting in motion affinity matrices of lower quality. As shown in table 5.5, the baseline method where the groundtruth number of motions is given also performs worst on these sequences.

5.4 Summary and Future Work

We propose a unified model selection technique for spectral clustering based motion segmentation methods, to automatically infer the number of motions in the scene. We combine four existing model selection criteria by computing custom confidence scores on a range of possible numbers of motions, and select the number with the highest average confidence among all four criteria as the optimal number of motions. This inferred number is then used to perform spectral clustering to obtain the final motion segmentation. Our method is tested with a state-of-the-art sparse correspondence based motion segmentation method we previously proposed on the challenging KT3DMoSeg dataset, and achieves competitive results, producing an overall error rate close to the baseline which takes the groundtruth number of motions as input.

The preliminary results have confirmed the efficacy of the proposed model selection technique on the KT3DMoSeg dataset, when it is used in conjunction with state-of-the-art sparse correspondence-based motion segmentation method. While the proposed method achieves competitive results close to the baseline which uses the known ground truth number of motions, further statistical analysis is necessary in order to rigorously compare the effectiveness of the proposed averaging technique with the consensus voting technique. The closeness in error rates between these two methods may indicate that variations in the results could stem from random fluctuations inherent in the evaluation process. Additionally, in order to better assess the versatility and applicability of the proposed model selection method, it is necessary to extend the scope of evaluation. This includes a more comprehensive evaluation across a larger group of spectral clustering-based motion segmentation methods, as well as on additional datasets. Regrettably, due to time constraint, an assessment of the proposed model selection technique in the context of the innovative zero-shot monocular dense motion segmentation method introduced in Chapter 4 was not feasible. As such, we leave the of conducting more thorough evaluations as future work.

Chapter 6

Conclusions and Future Work

This thesis presented a novel zero-shot approach to performing monocular dense motion segmentation from a moving camera. The proposed approach synergistically combines the advantages of both deep learning foundation models and geometric approaches, resulting in a zero-shot motion segmentation approach that performs motion model fusion on object proposals. This approach achieved state-of-the-art performance without the need for any training. Although the proposed approach achieves impressive motion segmentation results in a zero-shot manner, one primary limitation is that it needs the number of distinct moving objects in the scene to produce optimal results. In order to mitigate this limitation, we proposed a model selection technique that automatically determines the number of motions for spectral clustering-based monocular motion segmentation.

Our experimental results demonstrate that the proposed motion segmentation methods outperform some state-of-the-art supervised motion segmentation methods on certain dataset and is highly competitive with others. In particular, fusing different motion models showed a significant performance improvement, validating the effectiveness of our motion model fusion technique.

Looking ahead, there are several promising directions for future work. One potential avenue is to incorporate additional motion models, such as the trifocal tensor, to further enhance the motion segmentation performance. Additionally, exploring ways to improve the robustness and accuracy of our methods in more challenging scenarios, such as in scenes with multiple moving objects or under varying lighting conditions, could also be beneficial. Lastly, model selection remains a major limitation, although primary results suggest the effectiveness of our proposed model selection technique, more comprehensive evaluation needs to be performed on our proposed model selection technique. While deep learning-based methods do indeed rely on extensive training data and computational resources, they may be the most effective solution to the model selection problem. In such cases, our in-depth analysis and comparison of various motion models and the motion model fusion techniques could provide valuable insights into the design of loss functions for training monocular motion segmentation networks.

In conclusion, this thesis contributes to the field of monocular motion segmentation from a moving camera by proposing novel methods that leverage both traditional geometric approaches and modern deep learning techniques. These methods represent a significant step forward in our ability to accurately segment motion in complex scenes using a single moving camera. We believe that the ideas and techniques presented in this thesis will inspire and inform future research in this important area of computer vision.

References

[1] Aloimonos and C. M. Brown. Direct processing of curvilinear sensor motion from a sequence of perspective images, 1984. Volume: 72.

[2] Shivangi Anthwal and Dinesh Ganotra. An overview of optical flow-based approaches for motion segmentation. *The Imaging Science Journal*, 67(5):284–294, July 2019.

[3] Federica Arrigoni, Luca Magri, and Tomas Pajdla. On the Usage of the Trifocal Tensor in Motion Segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12365, pages 514–530. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[4] Federica Arrigoni, Elisa Ricci, and Tomas Pajdla. Multi-frame Motion Segmentation by Combining Two-Frame Results. *International Journal of Computer Vision*, 130(3):696–728, March 2022.

[5] Daniel Barath and Jiri Matas. Graph-Cut RANSAC. pages 6733–6741. IEEE Computer Society, June 2018.

[6] Daniel Barath and Jiri Matas. Progressive-X: Efficient, Anytime, Multi-Model Fitting Algorithm. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3779–3787, Seoul, Korea (South), October 2019. IEEE.

[7] Pia Bideau and Erik Learned-Miller. It's Moving! A Probabilistic Model for Causal Motion Segmentation in Moving Camera Videos. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9912, pages 433–449. Springer International Publishing, Cham, 2016. Series Title: Lecture Notes in Computer Science.

[8] Pia Bideau, Rakesh R. Menon, and Erik Learned-Miller. MoA-Net: Self-supervised Motion Segmentation. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, volume 11134, pages 715–730. Springer International Publishing, Cham, 2019. Series Title: Lecture Notes in Computer Science.

[9] Pia Bideau, Aruni RoyChowdhury, Rakesh R. Menon, and Erik Learned-Miller. The Best of Both Worlds: Combining CNNs and Geometric Constraints for Hierarchical Motion Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 508–517, Salt Lake City, UT, USA, June 2018. IEEE.

[10] Markus Bosch. Deep Learning for Robust Motion Segmentation with Non-Static Cameras, February 2021. arXiv:2102.10929 [cs].

[11] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 282–295, Berlin, Heidelberg, September 2010. Springer-Verlag.

[12] T. Caliński and J Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, January 1974. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101.

[13] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and Track Anything, May 2023. arXiv:2305.06558 [cs].

[14] Tat-jun Chin, Hanzi Wang, and David Suter. The Ordered Residual Kernel for Robust Motion Subspace Clustering. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[15] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion, May 2022. arXiv:2205.07844 [cs].

[16] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards Segmenting Anything That Moves. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1493–1502, Seoul, Korea (South), October 2019. IEEE.

[17] David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[18] Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2173–2180, June 2010. ISSN: 1063-6919.

[19] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, December 2015. ISSN: 2380-7504.

[20] E. Elhamifar and R. Vidal. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, November 2013.

[21] Muhammad Faisal, Ijaz Akhter, Mohsen Ali, and Richard Hartley. EpO-Net: Exploiting Geometric Constraints on Dense Trajectories for Motion Saliency. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1873–1882, Snowmass Village, CO, USA, March 2020. IEEE.

[22] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[23] 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, June 2012. ISSN: 1063-6919.

[24] Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13682, pages 59–75. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science.

[25] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[26] Joachim Heel. Direct Estimation of Structure and Motion from Multiple Frames. March 1990. Accepted: 2004-10-04T14:35:38Z.

[27] Berthold K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, June 1988.

[28] Yuxiang Huang, Yuhao Chen, and John Zelek. Dense Monocular Motion Segmentation Using Optical Flow and Pseudo Depth Map: A Zero-Shot Approach. In *21st Conference on Robots and Vision (CRV)*, Guelph, ON, Canada, May 2024. IEEE.

[29] Yuxiang Huang, Yuhao Chen, and John Zelek. Zero-shot Monocular Motion Segmentation in the Wild by Combining Deep Learning with Geometric Motion Model Fusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, June 2024. IEEE.

[30] Yuxiang Huang and John Zelek. Motion Segmentation from a Moving Monocular Camera. In *IROS 2023 Workshop on Robotic Perception and Mapping: Frontier Vision and Learning Techniques*. arXiv, October 2023. arXiv:2309.13772 [cs].

[31] Yuxiang Huang and John Zelek. A Unified Model Selection Technique for Spectral Clustering Based Motion Segmentation. *Journal of Computational Vision and Imaging Systems*, 9(1), 2023.

[32] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, Salt Lake City, UT, USA, June 2018. IEEE.

[33] David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Thomas Brox, and Jitendra Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6315, pages 282–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. Series Title: Lecture Notes in Computer Science.

[34] Hossam Isack and Yuri Boykov. Energy-Based Geometric Multi-model Fitting. *International Journal of Computer Vision*, 97(2):123–147, April 2012.

[35] Jianbo Shi and Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, pages 593–600, Seattle, WA, USA, 1994. IEEE Comput. Soc. Press.

[36] Yangbangyan Jiang, Qianqian Xu, Ke Ma, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. What to Select: Pursuing Consistent Motion Segmentation from Multiple Geometric Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1708–1716, May 2021. Number: 2.

[37] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion Trajectory Segmentation via Minimum Cost Multicuts. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3271–3279, December 2015. ISSN: 2380-7504.

[38] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A Multi-cut Formulation for Joint Segmentation and Tracking of Multiple Objects, July 2016. arXiv:1607.06317 [cs].

[39] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[40] Taotao Lai, Hanzi Wang, Yan Yan, Tat-Jun Chin, and Wan-Lei Zhao. Motion Segmentation Via a Sparsity Constraint. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):973–983, April 2017. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[41] Zhuwen Li, Jiaming Guo, Loong-Fah Cheong, and Steven Zhiying Zhou. Perspective Motion Segmentation via Collaborative Clustering. In *2013 IEEE International Conference on Computer Vision*, pages 1369–1376, Sydney, Australia, December 2013. IEEE.

[42] Shuyuan Lin, Anjia Yang, Taotao Lai, Jian Weng, and Hanzi Wang. Multi-motion Segmentation via Co-attention-induced Heterogeneous Model Fitting. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, February 2015. arXiv:1405.0312 [cs].

[44] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, March 2023. arXiv:2303.05499 [cs].

[45] H. C. Longuet-Higgins and K. Prazdny. The Interpretation of a Moving Retinal Image. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 208(1173):385–397, 1980. Publisher: The Royal Society.

[46] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. EM-driven unsupervised learning for efficient motion segmentation, March 2022. arXiv:2201.02074 [cs].

[47] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. EM-Driven Unsupervised Learning for Efficient Motion Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4462–4473, April 2023. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[48] Amar Mitiche and J.K. Aggarwal. *Computer Vision Analysis of Image Motion by Variational Methods*, volume 10 of *Springer Topics in Signal Processing.* Springer International Publishing, Cham, 2014.

[49] Eslam Mohamed, Mahmoud Ewaisha, Mennatullah Siam, Hazem Rashed, Senthil Yogamani, Waleed Hamdy, Mohamed El-Dakdouky, and Ahmad El-Sallab. Monocular Instance Motion Segmentation for Autonomous Driving: KITTI InstanceMotSeg Dataset and Multi-Task Baseline. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 114–121, Nagoya, Japan, July 2021. IEEE Press.

[50] D. Myatt, Philip Torr, Slawomir Nasuto, John Bishop, and R. Craddock. NAPSAC: High Noise, High Dimensional Robust Estimation - it's in the Bag. January 2002.

[51] Shahriar Negahdaripour and Berthold K. P. Horn. Direct Passive Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):168–176, January 1987.

[52] Michal Neoral. Monocular Arbitrary Moving Object Discovery and Segmentation. In *The British Machine Vision Conference (BMVC)*, 2021.

[53] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of Moving Objects by Long Term Video Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, June 2014. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu,

Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, April 2023.

[55] Anestis Papazoglou and Vittorio Ferrari. V.: Fast object segmentation in unconstrained video. In *In: ICCV (2013*.

[56] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, March 2009.

[57] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS Challenge on Video Object Segmentation, March 2018. arXiv:1704.00675 [cs].

[58] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment Anything Meets Point Tracking, July 2023. arXiv:2307.01197 [cs].

[59] Mohamed Ramzy, Hazem Rashed, Ahmad El Sallab, and Senthil Yogamani. RST-MODNet: Real-time Spatio-temporal Moving Object Detection for Autonomous Driving, December 2019. arXiv:1912.00438 [cs, stat] version: 1.

[60] Anurag Ranjan and Michael J. Black. Optical Flow Estimation Using a Spatial Pyramid Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, Honolulu, HI, July 2017. IEEE.

[61] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, October 2010. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[62] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987.

[63] Hicham Sekkati and Amar Mitiche. A variational method for the recovery of dense 3D structure from motion, 2006.

[64] Hicham Sekkati and Amar Mitiche. A variational method for the recovery of dense 3D structure from motion. *Robotics and Autonomous Systems*, 55(7):597–607, July 2007.

[65] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. MODNet: Motion and Appearance based Moving Object Detection Network for Autonomous Driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864, November 2018. ISSN: 2153-0017.

[66] G. W. Stewart and Jiguang Sun. *Matrix Perturbation Theory.* New York: Academic Press, 1990.

[67] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J. Fleet, and William T. Freeman. Disentangling Architecture and Training for Optical Flow. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13682, pages 165–182. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science.

[68] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J. Fleet, and William T. Freeman. Disentangling Architecture and Training for Optical Flow. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 165–182, Cham, 2022. Springer Nature Switzerland.

[69] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, Salt Lake City, UT, USA, June 2018. IEEE.

[70] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12347, pages 402–419. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[71] P. H. S. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, May 1998.

[72] P.H.S. Torr. Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting. *International Journal of Computer Vision*, 50(1):35–61, October 2002.

[73] Roberto Tron and Rene Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. ISSN: 1063-6919.

[74] Johan Vertens, Abhinav Valada, and Wolfram Burgard. SMSnet: Semantic motion segmentation using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 582–589, September 2017. ISSN: 2153-0866.

[75] R. Vidal and D. Singaraju. A closed form solution to direct motion segmentation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 510–515 vol. 2, June 2005. ISSN: 1063-6919.

[76] Rene Vidal. Subspace Clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, March 2011.

[77] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[78] Andreas Wedel, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers. Detection and Segmentation of Independently Moving Objects from Dense Scene Flow. pages 14–27, August 2009.

[79] Zhao Xi, Jun Liu, Bin Luo, and Qianqing Qin. Multi-Motion Segmentation: Combining Geometric Model-Fitting and Optical Flow for RGB Sensors. *IEEE Sensors Journal*, 22(7):6952–6963, April 2022. Conference Name: IEEE Sensors Journal.

[80] Fuyuan Xu, Guohua Gu, Kan Ren, and Weixian Qian. Motion Segmentation by New Three-View Constraint from a Moving Camera. *Mathematical Problems in Engineering*, 2015:1–14, 2015.

[81] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11209, pages 603–619. Springer International Publishing, Cham, 2018. Series Title: Lecture Notes in Computer Science.

[82] Xun Xu, Loong Fah Cheong, and Zhuwen Li. Motion Segmentation by Exploiting Complementary Geometric Models. In *2018 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, pages 2859–2867, Salt Lake City, UT, USA, June 2018. IEEE.

[83] Gengshan Yang and Deva Ramanan. Learning to Segment Rigid Motions from Two Frames. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1266–1275, Nashville, TN, USA, June 2021. IEEE.

[84] Zongxin Yang and Yi Yang. Decoupling Features in Hierarchical Propagation for Video Object Segmentation. *Advances in Neural Information Processing Systems*, 35:36324–36336, December 2022.

[85] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video Object Segmentation and Tracking: A Survey. *ACM Transactions on Intelligent Systems and Technology*, 11(4):36:1–36:47, May 2020.

[86] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, Yandong Guo, and Lei Zhang. Recognize Anything: A Strong Image Tagging Model, June 2023. arXiv:2306.03514 [cs].