EXPLORING MOTION SIGNATURES FOR VISION-BASED TRACKING, RECOGNITION AND NAVIGATION

A Dissertation

by

WEN LI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Dezhen Song
Committee Members,	Thomas Ioerger
	Dylan Shell
	Wei Yan
Head of Department,	Nancy Amato

August 2014

Major Subject: Computer Engineering

ABSTRACT

As cameras become more and more popular in intelligent systems, algorithms and systems for understanding video data become more and more important. There is a broad range of applications, including object detection, tracking, scene understanding, and robot navigation. Besides the stationary information, video data contains rich motion information of the environment. Biological visual systems, like human and animal eyes, are very sensitive to the motion information. This inspires active research on vision-based motion analysis in recent years. The main focus of motion analysis has been on low level motion representations of pixels and image regions. However, the motion signatures can benefit a broader range of applications if further in-depth analysis techniques are developed.

In this dissertation, we mainly discuss how to exploit motion signatures to solve problems in two applications: object recognition and robot navigation.

First, we use bird species recognition as the application to explore motion signatures for object recognition. We begin with study of the periodic wingbeat motion of flying birds. To analyze the wing motion of a flying bird, we establish kinematics models for bird wings, and obtain wingbeat periodicity in image frames after the perspective projection. Time series of salient extremities on bird images are extracted, and the wingbeat frequency is acquired for species classification. Physical experiments show that the frequency based recognition method is robust to segmentation errors and measurement lost up to 30%. In addition to the wing motion, the body motion of the bird is also analyzed to extract the flying velocity in 3D space. An interacting multi-model approach is then designed to capture the combined object motion patterns and different environment conditions. The proposed systems and algorithms are tested in physical experiments, and the results show a false positive rate of around 20% with a low false negative rate close to zero.

Second, we explore motion signatures for vision-based vehicle navigation. We discover that motion vectors (MVs) encoded in Moving Picture Experts Group (MPEG) videos provide rich information of the motion in the environment, which can be used to reconstruct the vehicle ego-motion and the structure of the scene. However, MVs suffer from high noise level. To handle the challenge, an error propagation model for MVs is first proposed. Several steps, including MV merging, plane-at-infinity elimination, and planar region extraction, are designed to further reduce noises. The extracted planes are used as landmarks in an extended Kalman filter (EKF) for simultaneous localization and mapping. Results show that the algorithm performs localization and plane mapping with a relative trajectory error below 5.1%.

Exploiting the fact that MVs encodes both environment information and moving obstacles, we further propose to track moving objects at the same time of localization and mapping. This enables the two critical navigation functionalities, localization and obstacle avoidance, to be performed in a single framework. MVs are labeled as stationary or moving according to their consistency to geometric constraints. Therefore, the extracted planes are separated into moving objects and the stationary scene. Multiple EKFs are used to track the static scene and the moving objects simultaneously. In physical experiments, we show a detection rate of moving objects at 96.6% and a mean absolute localization error below 3.5 meters.

DEDICATION

To My Dear Parents and Aibo

ACKNOWLEDGEMENTS

I would like to extend my profound thanks to the people who helped my through my Ph.D. study. This work would not have been possible without the support of all my advisors, colleagues, mentors and friends.

Specially, I would like to express my deepest gratitude to Dezhen Song for his continuous support and mentorship. Dez has been encouraging me through the entire Ph.D. experience. He is always willing to spend time with me and inspire me to explore new problems. His professional guidance assisted, armed me with invaluable tools for life, and filled me with an exciting anticipation for the future to come.

Sincere thanks to Dr. Wei Yan, Dr. Thomas Ioerger and Dr. Dylan Shell for serving as my dissertation committee members and their insightful suggestions and inspiring discussions on advancing my research.

Thanks to A. Tian for his insightful discussions and help in collecting experiment data. Thanks to Y. Xu for his inspiration in my research work. Thanks to Y. Lu, J. Lee, M. Hielsberg, Z. Gui, S.-H. Mun, S. Jacob, M. Midori Hirami, P. Peelen, A. Tuan, and X. Wang for their inputs and contributions to the NetBot Laboratory and my research projects. Finally, thanks to all my friends and colleagues in Texas A&M University, who have been supportive and encouraging all the time in my study and life.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

As mobile imaging devices become more compact and affordable, the capturing and sharing of video streams become much easier and popular. The area of robotics benefits from the technology evolution. Vision, as an important sensing modality for many applications, attracts research in many directions. Unlike data acquired from other sensors, such as radar and laser range finder, video streams not only provide rich information of the environment, but also contain continuous motion information for objects in sight. Vision-based motion analysis research becomes active in recent years, and mainly focuses on low level motion representations of pixels or image regions. However, if further in-depth analysis techniques are developed, the motion signatures can benefit a broader range of applications. In this dissertation, we explore novel motion signatures and algorithms to solve problems in two applications: object recognition and robot navigation.

1.1 Bird Species Recognition for Autonomous Nature Observation

We first explore complex motion signatures for the bird species recognition problem. Imagine a flying bird is captured by an untrained amateur with a hand-held camera. We want to automatically extract the bird species information from the video, which will help ornithologists to study how local birds change their range as the result of climate change. The bird is a free-flying non-rigid object, and thus shapes of its projection onto images change from time to time. With various lighting conditions caused by different weather, time of day, and camera perspectives in field, the bird appears differently in the video frames. Conventional appearance-based tracking and recognition methods tackle the problem by matching similarities of features, such as key-points, color statistics, and textures. However, these features are usually not reliable in outdoor field environments. In addition, often times, there is not a lot of training data due to the uncontrolled objects and environments. These challenges motivate us to explore this object recognition problem for outdoor field environment using in-depth analysis of motion signatures.

We use bird species recognition as an application, and introduce two tracking and recognition algorithms based on motion signatures.

1.1.1 Filtering Using Periodic Motion of Salient Extremeties

We first analyze the articulated 3D motion of bird wings using kinematics models, and obtain wing motion periodicity in image frames after the perspective projection. We propose an approach using the periodicity of salient extremities for animals. For birds, the salient extremities are represented by inter-wing tip distance (IWTD) whose periodic motion is often characterized by wingbeat frequency (WF). We show that the probability that the salient extremity can be recognized is an increasing function of video data amount except ignorable degenerating cases. We model the body-wing structure of a bird using a 3 degree-of-freedom (DOF) kinematics model. We formally prove that the periodicity in salient extremities (i.e., IWTD) in the image frames is determined by the wingbeat frequency (WF) in the world frame. The periodicity is invariant to camera parameters. These analyses enable us to develop an automatic species filtering method consisted of two algorithms. The first algorithm recognizes the IWTD time series from motion segmented bird body contours in the video sequence. The second algorithm applies Fast Fourier transform (FFT) to the IWTD series, and classifies bird species using likelihood ratios. The algorithm returns a ranked short candidate list of species. We have implemented the algorithms. Experimental results are satisfying and the algorithm is also very robust to data loss: it is capable of overcoming up to 30% of data loss in the tests.

1.1.2 Multi-Model Filtering Using Videos from Uncalibrated Cameras

The frequency obtained from the time series of IWTD captures the wingbeat motion of the flying bird. However, a bird flying motion often contains a mixture of gliding and wing flapping motions. Simply using wing flapping frequency-based approach might not be efficient. Therefore, we further propose a multi-model approach to catch complex object motion patterns and different environment conditions. The new multi-model algorithm tracks both body and wing motion of a flying bird to form signatures for species filtering. In the body motion model, we consider both cases when background motion introduced by the camera can and cannot be directly recognized using feature key point matching. We are also able to recover intrinsic camera parameters during the body motion tracking. In the wing motion model, we consider both periodic wing flapping and gliding motion patterns. These models are combined to form a multi-model framework fused by interacting multiple model-based extended Kalman filters (IMM-EKF). We have implemented the algorithm and compared its performance with single model approaches in physical experiments. Results show that the new algorithm significantly reduces the false positive (FP) rate while maintaining a low false negative (FN) rate. The area under ROC curve is 92.86%.

1.2 Vision-based Vehicle Navigation Using Monocular Camera

We then explore motion signatures for the vision-based robot navigation problem. Robot navigation is a fundamental problem to many applications in the area of robotics, and vision-based navigation algorithms are important to most mobile robots in GPS-challenged environments. The robot navigation task includes perceiving the surrounding environment and estimating the robot ego-motion. Many visual navigation approaches rely on correspondence of features between individual images to establish geometric understandings of image data. To do that, image data are often first reduced to a feature set such as points. Then extensive statistical approaches such as random sample consensus (RANSAC) are employed to search for feature matches that satisfy the expected geometric relationships. Such geometric relationships enable us to derive robot/camera ego-motion estimation or scene understandings. The inherent drawback of these approaches is the expensive computation load and robustness of feature extraction, which is often hindered by varying lighting conditions and occlusions.

Recent streaming videos are transmitted after complex compression. These algorithms exploit similarities between blocks of pixels in adjacent frame sets, which are characterized as motion vectors (MVs), to reduce bandwidth needs. Compared with optical flows, MVs have lower spatial resolution (per block vs. per pixel) but higher temporal resolution because MVs are extracted from multiple frames instead of mere two adjacent frames. MVs carry the correspondence information and are readily available from the encoded video data. This motivates us to explore using the MVs from video streams as inputs for the visual navigation problem. In this dissertation, we introduce two MV-based vehicle navigation algorithms.

1.2.1 Toward Featureless Visual Navigation: Simultaneous Localization and Planar Surface Extraction

In a stationary environment, the navigation algorithm is usually conducted under the simultaneous localization and mapping (SLAM) framework. The challenges of using MVs in SLAM framework rise from their low spatial resolution and high noise level. To deal with challenges, we first establish an error propagation model for MVs, and propose to merge MVs from different frames to improve signal quality. Then, MV thresholding is applied to remove the far field in the scene that is not sensitive to robot motion. Instead of using isolated image features, we propose to use homography filtering to extracted planar regions in the scene as landmarks. The homography constraint helps to further improve MV quality. An extended Kalman filter (EKF) based approach is then introduced to simultaneously track robot motion and planes. We have implemented the proposed method and tested it in physical experiments. Results show that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.

1.2.2 Simultaneous Localization, Planar Surface Extraction, and Moving Obstacle Tracking

In environments that contain moving objects, SLAM and obstacle avoidance are two critical navigation functionalities. They are often handled separately due to the limitation of existing methods. This artificial separation can lead to problems such as synchronization or redundant processing of information, which are not desirable. We develop a new algorithm that is capable of performing the SLAM task and obstacle tracking in a single framework using MVs. We first extracts planes from MVs. We label the MVs as stationary and moving according to their consistency to geometric constraints, and the extracted planar regions are separated into stationary and moving groups by the majority voting of their MVs. Multiple EKFs are applied parallel to track the stationary scene and moving objects. Exchanging between the filters is also designed as the objects may start to move or becomes static during the process. The system is implemented and tested in public dataset. Results show that the proposed method performs a mean absolute trajectory error below 3.5 meters, which is less than 2.53% of the trajectory length. In addition, the true positive rate of moving object detection reaches 96.6%.

1.3 Dissertation Organization

The rest of this dissertation is organized as follows. Related work on object tracking, recognition and robot navigation is discussed in Section 2. Sections 3 and 4 address the bird species recognition problem. In Section 3, we propose the frequency based bird species recognition method using the wing flapping motion. Then, we introduce the interacting multi-model method for tracking the combined object motions in Section 4. The vehicle navigation problem is discussed in Sections 5 and 6. In Section 5, an algorithm is proposed for simultaneous localization and planar surface extraction in stationary environments. Section 6 extends the algorithm to dynamic environments with moving object tracking. Section 7 concludes this dissertation with a summary and future work.

2. RELATED WORK

The first part of our work relates to vision-based tracking and recognition of wild animals, and the second part of this work is relevant to vision-based robot navigation and 3D reconstruction. In this section, we develop a review of related literatures in three sections. First, we discuss existing methods for vision-based object tracking and recognition. Our discussion of vision-based object tracking and recognition includes conventional appearance based methods and motion signature based methods. In addition, we provide a brief review of wild animal tracking techniques using non-vision based sensors. Then, in the last section, we explore existing methods for vision-based robot navigation and scene reconstruction.

2.1 Vision-based Object Tracking and Recognition

Vision-based object tracking and recognition have been active topics in computer vision and robotics research. Techniques for object recognition and tracking mutually benefit the study in each field. On one hand, the recognition algorithms help to measure similarity between different observation regions, thus recommend possible correspondences for object tracking. On the other hand, the tracking algorithms help to locate the target objects in multiple frames, such that local representations can be extracted for object recognition.

General vision-based object tracking and recognition methods can be classified into different catgories, according to the their observation space i.e. 2D tracking or 3D tracking, the number of sensors i.e. single camera and multiple cameras, the motion of the cameras i.e. static camera and mobile camera, the environment they are primarily designed to i.e. indoor, outdoor and airborne, and their image representation of objects i.e. point, template and silhouette. A comprehensive survey of object tracking and recognition algorithms before 2006 can be found in [3]. Here we discuss conventional methods used in animal tracking and recognition in the following categories.

In point based tracking methods, an object is represented by a point, and algorithms associate points in consecutive frames to represent animal movement. According to [3], the algorithms used to establish point associations can be classified into deterministic methods and statistical methods. Deterministic methods use constraints such as common motion [4], constant velocity [5] or proximity [6] to find correspondences across frames. Statistical methods use probabilistic filtering to predict object location w.r.t. previous frames. For example, extended kalman filter is used in [7] to track small flying animals with multiple cameras, and particle filter is used in [8] to track multiple flying birds simultaneously. Point based methods provide animal location changes w.r.t. time, and are used in many animal tracking problems. Since no appearance information is considered, this category of methods usually has low computational cost especially when multiple targets are tracked at the same time. However, they are sensitive to occlusion and measurement errors, and are usually not applied for recognition.

Model based tracking methods work on regions in images. To track an particular animal in the video, a specific model has to be built first. Types of object models include articulated model, skeleton template, appearance statistics and point distribution. Motion tracking is performed by searching for regions that best match the given model. For example, in [9], a point distribution model is learned to describe the top view of a pig, and the target pig is located by finding the region that best fits the model. Similarly, in [10], key point based face detection is utilized in tracking and recognizing lion faces. In [11], a 3D kinematics model is built to track the leg motion of a hopping wallaby. In [12], a joint skeleton model is used to track the pose of a dog. In [13], appearance model is established in rectangular window to track the motion and interaction of ants. Model based tracking methods are usually robust to non-rigid animal motion and partial occlusion. Moreover, models like skeleton help to learn the kinematics in animal motion. However, to obtain robust tracking, models have to be learned from sufficient training data, to describe all viewing perspectives and motion possibilities. In addition, model based tracking and recognition methods usually require pre-processing of image frames, such that the target object is translated and scaled for model fitting.

Silhouette based tracking methods emphasis on the tracking of animal shape and contour in images. In initialization, object silhouette or contour can be extracted by segmentation methods, such as edge detection [14] or active contour [15] algorithms. In consecutive frames, silhouettes are tracked by shape matching or local contour evolution methods [3]. For example, in [16], the shape of a sheep in its side-projection is extracted and matched in every frame. Silhouette based animal tracking methods provide complete regions of the animal. Therefore, they are effective for observing detailed motion of the animal body. The advantage of using silhouette based method is their flexibility to handle a large variety of animal shapes, using contour evolution. However, for animals with rapid shape changes, the iterative silhouette evolution in each frame may be time-consuming. Moreover, under moving cameras or fast animal motion, the object may be lost in contour evolution, and the extracted contour may be lead to unrelated objects in cluttered background.

Note that, most existing animal tracking and recognition methods work on the classification of animal genus, rather than species level filtering in our work.

2.1.1 Motion Signature

Motion information has been studied for vision-based detection, tracking and segmentation for decades. In recent years, motion signatures are also introduced to vision-based recognition problems. Different from appearance, motion signatures cannot be directly observed from single image. The extraction of motion signatures usually involves analysis across frames. Considering the level of the features, popular motion features used in object tracking and recognition can be categorized into the following two types.

Pixel-level motion feature is first explored by researchers. The design of pixel-level motion feature can be traced back to 1980s, when computation methods for optical flow are proposed in [17] and [18]. Researchers have been working on efficient and robust algorithms [19–21] for optical flow estimation. Variational optical flow is introduced in [22] and improved in [23] to provide 3D motion estimation in stereo view. For robust point extraction, SIFT [24] method is combined with optical concept to provide a SIFT flow [21] motion feature. These features are effective in object tracking. Built upon the motion estimation of sparse interest points, short moving trajectories of key points, named trajecton, are extracted for recognizing object activities in [25] and [26]. However, trajectory based recognition usually requires static camera setting or preprocessing like image stabilization.

To handle the sensitivity of pixel-level motion features, layer-level motion features are designed to express the complex motion in different parts of image. In [27], pixels with similar velocity are grouped to form a layer and the median velocity of those pixels is calculated to represent the motion of the layer. Similarly, in [28], interest points with coherent motions are clustered using their trajectories, which enables selective magnification of certain motion type. A mid-level motion feature is proposed in [29] for object action recognition. In this work, optical flow is first extracted for every pixel in every frame. Then, a mid-level feature is extracted by weighted combination of thresholded optical flows in all frames within an image region. Mid-level features in all regions are then used together to represent the object motion. As an alternative to optical flow, gradient provides another way of describing motion. In [30], temporal derivatives are calculated on every object pixel in every frame, and a spatio-temporal volume is composed by concatenating the frames of a single complete cycle of an action. Similarly, in [31], gradient based 3D spatial-temporal features are designed to describe the activity of a mouse. Layer-based motion signatures are robust to partial occlusion and are expressive in complex object motion. However, they usually require preprocessing of translation and scaling, and the computation cost for volume matching should be considered.

Special motion properties such as periodicity, are also explored for recognition and tracking. In [32], periodic motion analysis is performed for recognition of pedestrian and running dogs. The authors use region similarity to detect the periodic motion in image sequences. In [33], motion periodicity is used as a feature to guide tracking and segmentation of target objects.

Although motion information has been used in object tracking for a long time, in-depth discussion of the projection property of 3D motion and complex object behaviors is still limited. Moreover, the integration of motion signatures into object recognition problems is also intriguing.

2.2 Wild Animal Tracking Techniques

Our work relates wild animal tracking and recognition. The main goal of wild animal tracking is to collect and store animal behavior data, such as location and movement. As the growth of communication techniques, methods for wild animal tracking vary for different environment settings and needs. Besides vision-based techniques, multiple types of sensors can be utilized in wild animal tracking. Here, we provide a brief survey of wild animal tracking techniques with non-vision sensors.

2.2.1 Radio Telemetry

A large number of wild animal tracking methods rely on radio telemetry. This category of methods uses radio transmitters to track target animals. The basic method of radio telemetry based animal tracking appears in the 1950s [34]. Animal-mounted transmitters are designed and tagged to target animals. The design of transmitters usually satisfy the following requirements: (1) periodically emit radio signals of particular frequency [35], (2) provide transmission for significant movement of animal [34], (3) has very low power rate for long time using and (4) has light weight for the animal to carry. Radio antennas have to be built on top of towers [34] to collect animal angular location, then triangulation is applied to determine distance location [36]. Applications include tracking of deer, rabbit and fox [35]. These methods are effective for homing in on animal for observing behaviour and daily activity to correlate with physiological and environmental data. However, they are limited to record animal location, movement and time information, and are very sensitive to environment and weather changes. The accuracy is not satisfying and the cost is high.

As the development of transmitters and sensors, current radio telemetry based animal tracking methods fall in two major branches: satellite-based methods and wireless sensor network based methods.

Satellite-based wild animal tracking methods are introduced in 1970s. In 1978, the Argos system [37] is created for scientific data collection around the world. Small platform transmitter terminals (PTTs) are developed to attach to animals and programmed to send signals to satellite at periodic intervals. Satellites are used to collect data and send it to processing centers via receiving stations. In 1980s and 1990s, many wildlife tracking methods [38] [39] have been designed to integrate various sensors, such as acceleration, temperature and humidity sensor, into the Argos satellite system. In 1990s, Global Positioning System (GPS) is introduced and integrated with the Argos system [40]. GPS collar based telemetry systems for animal tracking begin in 1991 and continue into the 21st century. Applications include tracking of moose [36], elk and wolf [41]. As the size and weight of PTTs become smaller, the applications also extend to small animals, like eagles [38]. The usage of satellites enables long time and distance tracking of animal behavior, such as migrations. The position information has better accuracy and the system is more stable for different weather and environment, such as ocean, desert and polar regions. However, research shows that the PTT units affect animals behaviors to some extent [42]. Moreover, satellite-based tracking methods rely on very complex and expensive system support, and are not good for vegetarian-covered areas, like forest and bushes.

Another branch of radio telemetry based animal tracking methods uses wireless sensor network to localize and track animal behaviors. For slow moving animals in limited active regions, local wireless sensor network (WSN) can be established to track the behavior of the animals. In [43], WSN is used to track the motion of turtles. Four motion sensors are placed at the corner of the animal active region, and a mobile sensor is placed on the animal body. An incremental grid based approach is designed to detect the motion and location of the turtles, and trajectories are visualized in local map. Similar tracking method is used for cows in [44]. Small wireless sensors are lighter than PTTs and GPS units, therefore they can be used on small animals. Moreover, WSN provide better performance in areas with weak satellite signals, such as bushy areas.

As an alternative to global positioning system, the global system for mobile communications (GSM) introduces a new way to track animal behaviors in long distance with low cost. In [45], GSM phone tags are placed on animals and programmed to send text messages of location and animal ID to laboratory at regular intervals. GSM based animal tracking methods over-perform satellite based methods in residential areas where cellphone signal is strong, and can be combined with GPS-based methods [46–48].

Overall, radio telemetry based animal tracking methods can be applied for both local and global animal tracking. They are especially effective in continuous tracking of a particular target animal. Besides, position tracking can have high accuracy. Multiple types of sensors, such as motion sensor, heat sensor and chemical sensor, can be integrated in the tags or collars for different observation purposes. However, both active and passive radio telemetry methods need to place tags on target animals, which limits their application in animal species and numbers. Moreover, data collected from radio telemetry based methods usually does not include visual information of animal behavior, which makes them not effective for the study of ego-motion.

2.2.2 Other Sensors for Wild Animal Tracking

Besides the radio telemetry based methods, other sensors are utilized in wild animal tracking tasks.

A light-level geolocator can also be placed on animal bodies for tracking [49]. A light-level geolocator utilizes lighting to locate animal movements. Since the sensor does not use radio technology, it can be designed much smaller and lighter. Due to its small size and light weight, it is mainly used in tracking bird migration. Although the accuracy of geolocator is not as good as GPS or Argos, it provides a much cheaper way of tracking small size animal in a far longer time. The main disadvantage is the results could be affected by cloud, shading and artificial light.

Infrared sensor provides an effective way of tracking animals hidden in habitat. Application examples include [50] and [51]. Unlike radio telemetry based methods, infrared sensor provides a low-cost way of monitoring the behavior of animals in non-visible places. However, the accuracy of the tracking results may be affected by artificial lights.

Fluorescent pigments are used in [52] to track mammals. This method is effective in following the movement of small mammals at night. Unlike infrared video based methods, this type of methods provides better accuracy, especially in lighted areas.

2.3 Vision-based Navigation and Scene Reconstruction

A primary goal of visual navigation is to understand scene structure and estimate robot motion. A typical framework employed by robot visual navigation is vision-based simultaneous localization and mapping (SLAM) [53]. Existing approaches to the visual SLAM problem can be classified into different categories, according to the methodology, sensor, and landmark representations. Here, we provide a brief review of methods in each category.

2.3.1 Methodology

Existing approaches in visual SLAM mainly rely on two dominant frameworks: probabilistic filtering and bundle adjustment. An analysis of the advantage of each framework is provided by Strasdat et al. in [54]. In the general problem of visual SLAM, a state vector is used to include the camera pose and the parameters of landmarks. Features extracted from images are represented in another vector as observations. The general goal of SLAM is to estimate the state vector in each frame, while observations stream come in.

2.3.2 Sensors for Visual SLAM

The core sensor for vision-based SLAM approach is the camera equipped on the robot. According to the types and number of cameras used, visual SLAM methods can be grouped into several categories as follows.

A collection of visual SLAM algorithms use a single camera as their sensor. The camera can be a common low cost camera [55] and is usually modeled as a pinhole camera. Monocular camera is the simplest camera setting, and enables lowest computation cost with no synchronization needed. However, the main disadvantages of using monocular camera are the missing of depth and absolute scale, and the limited camera view. Without absolute measurement, the results from monocular SLAM methods are up to scale, and suffer scale drift for long distance estimation. Many methods are introduced to handle the scale drifting problem, including [55–57], and the main ideas focus on using loop closure to correct scales and using active approach to search for observations.

Stereo vision becomes a favorable choice for many visual SLAM applications [58,59]. With well configured and calibrated stereo cameras, the estimation can be provided in absolute world scale, which is useful for real-world application. In addition, with a fixed scale, the drifting problem in monocular SLAM can be reduced, which makes the method suitable for long distance. Also, as [54] discusses, the distribution of landmarks is more important than the number of landmarks. A stereo view provides wider camera view and potentially better observation distribution. Besides stereo cameras, multiple cameras is another choice to provide comprehensive information of overlapping observations. The multiple cameras can be mounted on a rig with fixed relative positions [60], or distributed independently on different platform [61]. The view of multiple cameras can cover a wide angle, or even the panoramic view [62]. The disadvantage of stereo or multiple cameras falls in the synchronization and calibration, as well as the computation cost for real-time application.

Omnidirectional camera is used in some works [63, 64] to provide sufficient observations of the environment. However, the images suffer strong distortion and request careful calibration. For omnidirectional camera, spherical projection model [65] has to be used and integral into the common filtering or bundle adjustment framework.

In recent year, RGB-D cameras attract attentions in the visual SLAM field [66]. RGB-D cameras provide the depth map of the scene together with the regular RGB image, which helps to reduce the scale and depth ambiguity in monocular visual SLAM problem. However, the depth map is usually noise and needs careful synchronization with the RGB image.

2.3.3 Landmark Representation

In a regular SLAM framework, the physical world is represented by a collection of landmarks which are primarily features observed from images. Various types of image features have been studied in literatures.

As the most common features for visual SLAM, Interest points are used as features in many existing works, such as [2, 67]. A comprehensive study of different point detectors is provided in [60], where features like Harris corner, SUSAN, SIFT and SURF are compared in aspects of stability and discover rates.

Low level features like edgelets [68], straight line segments [68–71] and curves [72] are also studied for visual SLAM. To better utilize line features, line segments are grouped according to their 3D directions, such as vertical [73], on-floor [74], wall and ceiling [75], and used separately in visual SLAM.

In recent years, high level features like 3D lines and planes [76–80] are introduced to vSLAM works to construct hierarchical environment representations. In some methods, the extraction of planar surface relies on 3D sensors [81–83], while others uses the fitting of 3D points [76] to detect 3D planes. Hierarchical methods, such as [80], have been studied to integral features in different levels and establish correlations.

In conclusion, this dissertation relates two main topics in the area of robotics: visual tracking and visual navigation. In this section, we discussed popular wild animal tracking techniques using different types of sensors, especially vision-based tracking and recognition methods, as well as approaches for vision-based navigation problem. With the related works discussed above, we start the main part of this dissertation in the following section. Let us begin with the analysis of motion periodicity.

3. BIRD SPECIES FILTERING USING PERIODIC MOTION OF SALIENT EXTREMITIES*

3.1 Introduction

We use bird species detection as an application, since a free flying bird processes the properties of a far field non-rigid object with full degree of freedom in 3D space and complex behaviors. Moreover, the environment of bird flying video is usually outdoor and uncontrolled, with significantly different lighting conditions.

The motion of a flying bird usually includes periodic wing flapping. In this section, we use articulated model of bird wings to study how the 3D periodicity of wing flapping motion is projected to 2D images. The analysis enables us to extract the motion periodicity from the time series of the salient extremities on segmented object contours. For birds, the salient extremities are represented by inter wing tip distance (IWTD) whose periodic motion is often characterized by wingbeat frequency (WF) (see Figure 3.1). Thus bird species is filtered by different wingbeat frequencies.

Before elaborating our work, we start with a brief review of related work on periodic motion analysis.

3.2 Related Work

Our automatic bird species detection method is based on the analysis of the periodicity of the salient extremities of the object. As an active research area [84, 85], periodic motion (PM) analysis provides clues to many vision problems, such as tracking and segmentation [33], single view 3D reconstruction [86–88], human/animal activity recognition [32], and pedestrian detection [89]. Our method extends existing recognition problems to a new

*Reprinted with permission from “Automatic Bird Species Detection from Crowd Sourced Videos” by Wen Li and Dezhen Song, 2014. IEEE Transactions on Automation Science and Engineering, Volume 11, Number 2, Page 348-358, Copyright 2014 by IEEE.

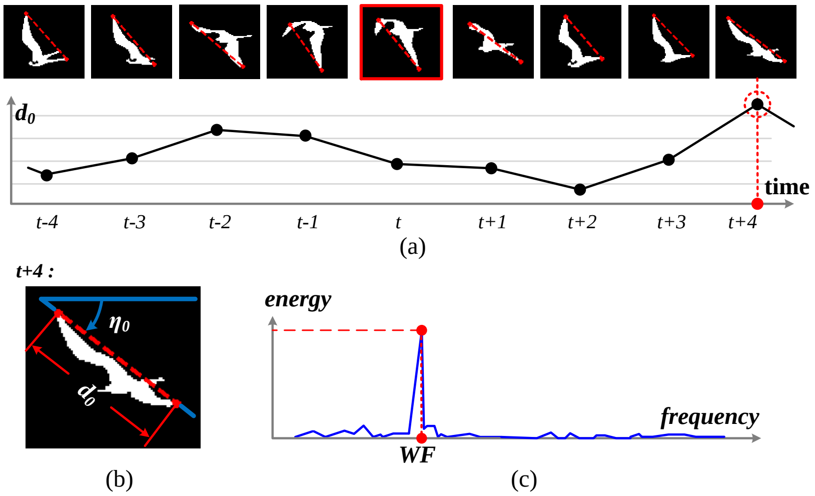


Figure 3.1: Recognizing salient extremities: (a) IWTD varies periodically according to WF. (b) IWTD is extracted as the primary feature. (c) WF is obtained through FFT.

domain: bird species recognition.

PM detection is nontrivial, and methods can be very different due to various camera settings and motion assumptions. Previous works can be classified into categories according to feature correspondence types. *Point correspondence* is used to estimate motion trajectories in [90–92]. However, as stated in [93], feature correspondence estimation is sensitive to illumination changes, reflectance, and especially occlusion. *Template based methods* are proposed in [94, 95]. Since the skeleton models capture the underlying bone structure, these methods serve well in motion capture and tracking applications for humans or animals. However, template based methods usually suffer high computational cost due to large searching and scaling space in the matching process.

Region correspondence based methods are introduced by Polana and Nelson [96], and further extended by Cutler and Davis [97]. These works assume that the object with repetitive motion should appear similar with its corresponding phrase in every period, and use a “similarity plot” to find period. These methods have certain robustness to image blurring and small background motion. However, they require 1) translation and scaling preprocessing, 2) small changing of background texture, and 3) stable viewing perspective. Some also rely on linear motion trajectory. Briassouli and Ahuja [93] avoid the translation and scaling by projecting images into 1D signals and analyzing the short term time-frequency distribution. However, their experiments do not show robustness to perspective changes, and the stationary camera assumption limits background motion.

Under a different application context, our work has to deal with an arbitrary moving camera and a free flying object, thus the viewing angle and trajectory are both subject to significant changes. We analyze the motion periodicity by tracking the movement of salient extremities, which in turn helps to avoid the stationary background requirement. Our feature analysis in frequency domain does not require pre-translation or rescaling. Since we do not use the similarity plot, the restrictive consistent viewing angle is no longer needed.

It is also worth noting that frequency-based methods are very robust to segmentation error. Existing results [98] and [99] show that the periodic frequency still can be extracted from the frequency spectrum even under small ($-10 \sim 10$ dB) signal-to-noise ratio. These results show that frequency-based methods are robust and worth considering in species filtering applications.

Our group has developed systems and algorithms for networked robotic cameras in nature observation applications [100–103]. Our previous work on bird species prediction [103] utilizes the bird body length and is limited to stationary camera with known parameters. This work extends our previous study to more general camera/scene settings.

3.3 Problem Description

The input of the system is a sequence of video frames. The output of the system is a list of candidate species, which is ranked from the most to the least likely.

3.3.1 Assumptions and Prior Knowledge

We assume that the bird in the given video is in steady flight under normal weather, which includes gliding, soaring, circling, cruising and level-flight, but excludes landing and taking off. Also, wing flapping motion should exist in the video. We assume that only one flying bird appears in the motion sequence. If there are multiple birds in the video, we can apply existing multiple target tracking techniques, such as [104], to separate individual bird sequence beforehand. The camera frame rate should be at least two times of WF [105]. Since the WFs of most bird species are lower than 15 Hz, a normal camera with 30 frames per second (fps) works for most cases.

Table 3.1: Prior knowledge of bird WFs. s is species id, and μ and σ are the mean and the standard deviation of the WF, respectively.

s	**μ (Hz)**	**σ (Hz)**	**Species**
6	3.18	0.227	Kittiwake
8	3.05	0.129	Herring Gull
12	4.58	0.183	Fulmar
...	...	...	...

We use the WF tables in [106,107] as the prior knowledge for our algorithm. The tables are obtained by experts' manually counting of continuous flapping motion (See Table 3.1 for a few examples).

3.3.2 Problem Definition

Denote $d(t)$ to be the IWTD at time/frame t in pixel coordinates. Define N_s as the number of candidate species in the prior information, $\mathcal{S} = \{1, ..., N_s\}$ the candidate species set, and $L'(\cdot|\cdot)$ the likelihood that a bird with WF f_0 and WF error bound f_e belongs to species s. The bird species recognition problem can be defined as two sub problems,

Definition 1 (Recognition of Salient Extremities)**.** *Given a bird flying image sequence, recognize time series* $d(t)$*.*

Definition 2 (Species Prediction)**.** *Given* $d(t)$ *and the candidate set* $\{(\mu_s, \sigma_s) : s = 1, ..., N_s\}$*, estimate* f_0*,* f_e*, and compute* $L'(\mu_s, \sigma_s | f_0, f_e), \forall s \in \mathcal{S}$*.*

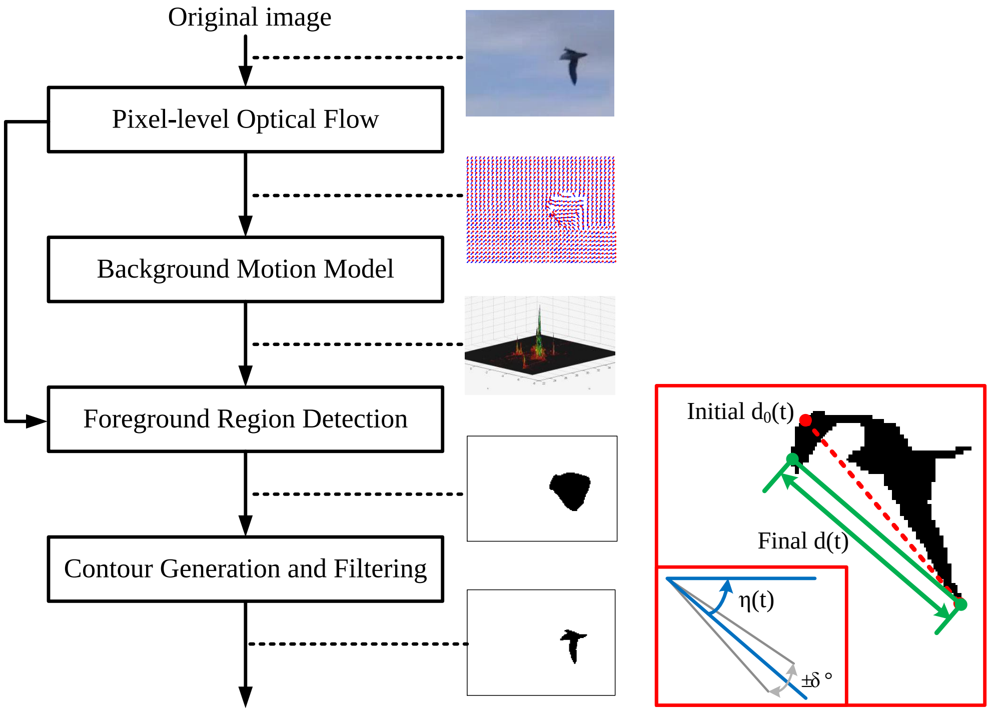
Let us begin with the first problem.

3.4 Recognition of Salient Extremities

The extraction of salient extremities has two steps: 1) motion segmentation that extracts the bird boundary from every frame, and 2) recognizing IWTD from bird boundaries.

3.4.1 Motion Segmentation

Since a flying bird is highly dynamic in appearance and shape, and camera motion is unknown, many segmentation methods are not applicable. We propose an unsupervised method for motion segmentation. Figure 3.2a illustrates the four-step process. For each image frame, optical flow algorithm [21] is applied to calculate the flow on each grayscale pixel. Since background pixels share a similar motion pattern, a background motion model is estimated by iteratively minimizing the covariance of a 2D Gaussian distribution [108]. The Mahalanobis distance between a flow vector and the background model is measured. For those distances that fall out of a flexible quantile [109] of the χ^2 distribution, we label their corresponding pixels as foreground. Active Contour algorithm [15, 110] is then applied to generate a smooth boundary of the foreground area. The foreground objects are further filtered by size and color consistence within consecutive frames, so that false detections are eliminated.



(a) Motion Segmentation

(b) Searching for IWTD

Figure 3.2: (a) A block diagram of motion segmentation. Thumbnails to the right of the block diagram indicate intermediate results. Black pixels in last two thumbnails indicate labeled foreground. (b) Searching for IWTD using WSD $\eta(t)$. The initial $d_0(t)$ is corrected by searching for $d(t)$ in the δ-neighborhood of $\eta(t)$.

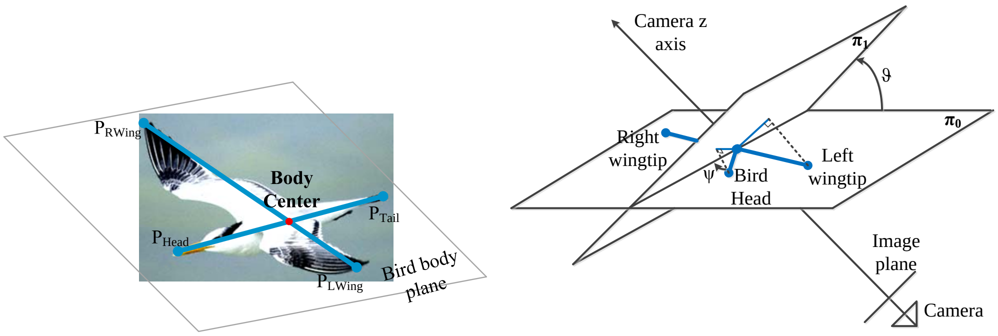
3.4.2 Recognizing Salient Extremities

With the bird boundary extracted, we can search for the salient extremities, namely, the IWTD for a bird. Define D for IWTD and L_B for bird body length in the 3D coordinate. The corresponding notations in the image coordinate system are d and l_B, respectively. Recognizing IWTD in image frames is nontrivial because camera relative perspectives to the bird are unknown and may change from time to time. We cannot identify the salient extremities by simply looking for the longest distance on the bird boundary in an arbitrary frame.

3.4.2.1 Finding the maximum IWTD across frames in a wingbeat period

If the video length is longer than a wingbeat period, the moment when the flying bird fully extends the wing should exist in the video. The moment offers the best opportunity to recognize IWTD. In fact, we can derive the following lower bound for the probability that the IWTD is the longest distance on the bird contour.

Lemma 1. *With a single period, the probability that the IWTD is the longest distance on the bird contour at the moment when the flying bird fully extends its wings is no less than* $1 - \frac{2}{\pi}\arctan(\frac{L_B}{D})$ *for an arbitrary camera perspective.*



(a) Bird Body Plane

(b) Bird Plane Angle Analysis

Figure 3.3: (a) Bird body plane and wing-body stick model (b) Projecting bird onto an intermediate plane that parallel to the image plane. ψ is the angle between the bird flying trajectory and the image/intermediate plane. ϑ is the angle between the bird body plane and the image/intermediate plane.

Proof. When the wingspan reaches its maximum in steady flight, the wing spreading direction (WSD) is perpendicular to the bird body axis. Model the bird skeleton by a cross (Figure 3.3a) with two orthogonal bars. The two bars determine a bird body plane. Recall that the perspective camera follows a pinhole camera model and the bird is in far field of

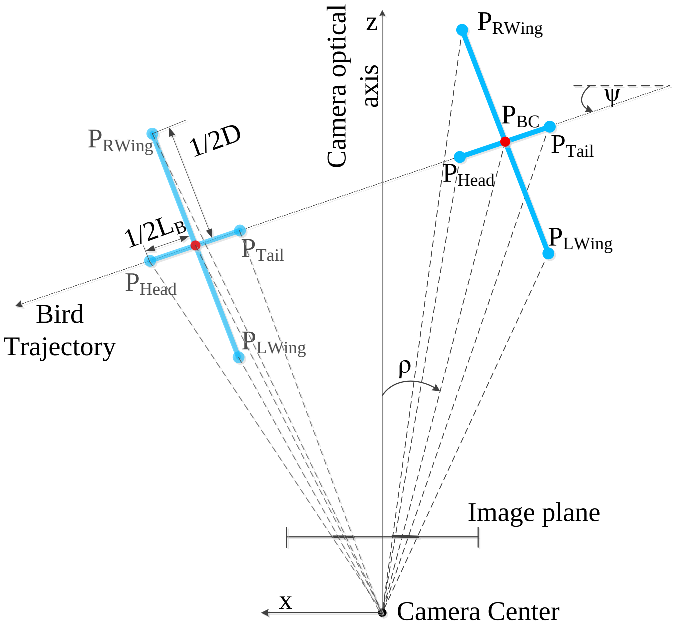


Figure 3.4: Illustration of projecting bird onto the image plane.

the camera view. The relationship between a 3D point $P = [X, Y, Z]^T$ and its 2D projection $p = [x, y]^T$ in the image follows $x = fX/Z$ and $y = fY/Z$ where f is the camera focal length. Notations in figures are defined as follows:

- $P_{BC} = [X_{BC}, Y_{BC}, Z_{BC}]^T$ is the 3D coordinate of bird body center.
- P_{LWing} and P_{RWing} are bird left and right wing tips in 3D, respectively.
- P_{Head} and P_{Tail} are bird head and tail end points in 3D, respectively.
- ψ is the angle of bird flying trajectory w.r.t. the image plane (see Figure 3.3b).
- ρ is the angle of bird body center projection ray w.r.t. the camera optical axis (z axis), see Figure 3.4.
- D is the length of IWTD in 3D, while L_B is the length of bird body in 3D.
- d is the length of IWTD on image, while l_B is the length of bird body on image.

Since the bird is in steady flight, its body plane is horizontal. The camera has a tilting angle ϑ w.r.t. the horizontal plane. We first analyze how the camera's tilting angle affects the probability of successful recognition of salient extremities. Consider $\rho = 0$ (the analysis is similar when $\rho \neq 0$). Plane π_0 represents the bird body plane in Figure 3.3b, while π_1 is parallel to the image plane and intersects π_0 at the bird body center. From Figure 3.3b, then the projection of bird body length L'_B and IWTD D' on π_1 are:

$$L'_B = L_B\sqrt{\cos^2\psi + \sin^2\psi\cos^2\vartheta} \tag{3.1}$$

$$D' = D\sqrt{\sin^2\psi + \cos^2\psi\cos^2\vartheta} \tag{3.2}$$

By geometry similarity, the ratio between d and l_B can be approximated by D'/L'_B (since the bird is in far view). To ensure $d/l_B > 1$, it must satisfy the following:

$$\tan\psi > \sqrt{\frac{1 - (D/L_B)^2\cos^2\vartheta}{(D/L_B)^2 - \cos^2\vartheta}}. \tag{3.3}$$

As $|\vartheta|$ grows larger from 0 to $90°$, the threshold becomes larger and the region of bird trajectory orientation ψ for successful recognition becomes smaller. When $\vartheta = 90°$ (that is the bird plane is perpendicular to the image plane), the probability of success reaches the minimum.

In the following proof, we analyze this worst scenario only, because it gives a lower bound of the probability of successful recognition of salient extremities. Figure 3.4 shows the top view of the setting in our analysis, where the image plane is perpendicular to the paper plane, and bird body plane is parallel to the paper plane. The bird trajectory is assumed to be a straight line in a short time period.

By the projection relationship between 3D and 2D points, we have

$$l_B = \left| \frac{f(X_{BC} + 1/2L_B \cos\psi)}{Z_{BC} - 1/2L_B \sin\psi} - \frac{f(X_{BC} - 1/2L_B \cos\psi)}{Z_{BC} + 1/2L_B \sin\psi} \right| \tag{3.4}$$

$$d = \left| \frac{f(X_{BC} + 1/2D \sin\psi)}{Z_{BC} + 1/2L_B \cos\psi} - \frac{f(X_{BC} - 1/2L_B \sin\psi)}{Z_{BC} - 1/2L_B \cos\psi} \right| \tag{3.5}$$

That is

$$l_B = \left| \frac{fL_B(X_{BC} \sin\psi + Z_{BC} \cos\psi)}{Z_{BC}^2 - (1/2L_B)^2 \sin^2\psi} \right| \tag{3.6}$$

$$d = \left| \frac{fD(X_{BC} \cos\psi - Z_{BC} \sin\psi)}{Z_{BC}^2 - (1/2D)^2 \cos^2\psi} \right| \tag{3.7}$$

Since the bird is in far field of the camera view, $D, L_B \ll Z_{BC}$, we ignore the second term in the denominator. Therefore, the ratio between d and l_B is

$$\begin{aligned} \frac{d}{l_B} &\approx \frac{D}{L_B} \left| \frac{X_{BC} \cos\psi - Z_{BC} \sin\psi}{X_{BC} \sin\psi + Z_{BC} \cos\psi} \right| \\ &= \frac{D}{L_B} \left| \frac{\tan\rho - \tan\psi}{\tan\rho \tan\psi + 1} \right| \\ &= \frac{D}{L_B} \left| \tan(\psi - \rho) \right| \end{aligned} \tag{3.8}$$

To successfully recognize the salient extremity as the IWTD, the ratio d/l_B should be greater than 1. Thus, we have

$$|\tan(\psi - \rho)| \geq \frac{L_B}{D} \tag{3.9}$$

Consider ψ is uniformly distributed on $(-\pi/2, \pi/2)$, and ρ is uniformly distributed on $(-\Theta_h, \Theta_h)$ where $0 < 2\Theta_h < \pi$ is the horizontal field of view of the camera. Let $\zeta =$

$(\psi - \rho)$, then ζ follows the triangle distribution, with the probability density function

$$f_\zeta = \begin{cases} \frac{\zeta+\Theta_h}{2\Theta_h\pi} + \frac{1}{4\Theta_h} & \textit{if } -\frac{\pi}{2} - \Theta_h \leq \zeta < -\frac{\pi}{2} + \Theta_h \\ \frac{1}{\pi} & \textit{if } -\frac{\pi}{2} + \Theta_h \leq \zeta < \frac{\pi}{2} - \Theta_h \\ \frac{-\zeta+\Theta_h}{2\Theta_h\pi} + \frac{1}{4\Theta_h} & \textit{if } \frac{\pi}{2} - \Theta_h \leq \zeta \leq \frac{\pi}{2} + \Theta_h \\ 0 & \textit{otherwise} \end{cases} \tag{3.10}$$

Define the indicator variable $\mathbf{I}_\zeta$ as

$$\mathbf{I}_\zeta = \begin{cases} 1 & \textit{if } d \geq l_B \\ 0 & \textit{otherwise} \end{cases} \tag{3.11}$$

Then, given a ratio L_B/D, the probability of successful recognition of salient extremities in a wingbeat period is the integral

$$Pr\{\mathbf{I}_\zeta = 1 | \frac{L_B}{D}\} = \int_{-\infty}^{+\infty} \mathbf{I}_\zeta f_\zeta d\zeta \tag{3.12}$$

$$= \int_{|\tan\zeta| \geq \frac{L_B}{D}} f_\zeta d\zeta \tag{3.13}$$

$$= \int_{|\tan\zeta| \geq \frac{L_B}{D}, \zeta \geq 0} f_\zeta d\zeta + \int_{|\tan\zeta| \geq \frac{L_B}{D}, \zeta < 0} f_\zeta d\zeta \tag{3.14}$$

Since the absolute tangent function is symmetric, the two parts in (3.14) are equal. Therefore, we only consider the integral on $\zeta \geq 0$ as follows

$$Pr\{\mathbf{I}_\zeta = 1 | \frac{L_B}{D}\} = 2 \int_{|\tan\zeta| \geq \frac{L_B}{D}, \zeta \geq 0} f_\zeta d\zeta \tag{3.15}$$

$$= 2 \int_{\arctan(\frac{L_B}{D})}^{\pi - \arctan(\frac{L_B}{D})} f_\zeta d\zeta \tag{3.16}$$

On the positive axis of ζ, the triangle distribution density changes at the point $\zeta = \frac{\pi}{2} - \Theta_h$.

Therefore, it is necessary to compare $\arctan(\frac{L_B}{D})$ with $\frac{\pi}{2} - \Theta_h$, in order to calculate the probability. We have two cases here:

Case 1: if $\arctan(\frac{L_B}{D}) \leq \frac{\pi}{2} - \Theta_h$, then (3.16) can be unfolded to

$$Pr\{\mathbf{I}_\zeta = 1 | \frac{L_B}{D}\} \tag{3.17}$$

$$= 2\int_{\arctan(\frac{L_B}{D})}^{\frac{\pi}{2}-\Theta_h} \frac{1}{\pi} d\zeta + 2\int_{\frac{\pi}{2}-\Theta_h}^{\frac{\pi}{2}+\Theta_h} \frac{-\zeta + \Theta_h}{2\Theta_h \pi} + \frac{1}{4\Theta_h} d\zeta \tag{3.18}$$

$$= 1 - \frac{2}{\pi}\arctan(\frac{L_B}{D}) \tag{3.19}$$

Case 2: if $\arctan(\frac{L_B}{D}) > \frac{\pi}{2} - \Theta_h$, then (3.16) is

$$Pr\{\mathbf{I}_\zeta = 1 | \frac{L_B}{D}\} \tag{3.20}$$

$$= 2\int_{\arctan(\frac{L_B}{D})}^{\pi-\arctan(\frac{L_B}{D})} \frac{-\zeta + \Theta_h}{2\Theta_h \pi} + \frac{1}{4\Theta_h} d\zeta \tag{3.21}$$

$$= 1 - \frac{2}{\pi}\arctan(\frac{L_B}{D}) \tag{3.22}$$

The two cases result in the same probability equation that is independent of Θ_h. The larger the ratio D/L_B is, the higher the successful probability can reach. Therefore, Lemma 1 is true. □

When more data are available, we have the following corollary.

Corollary 1. *The probability lower bound that the IWTD is the longest distance on the bird contour across k wingbeat periods with independent camera perspectives is*

$$1 - (\frac{2}{\pi}\arctan(\frac{L_B}{D}))^k. \tag{3.23}$$

This conclusion can be straightforwardly derived from Lemma 1. For k wingbeat periods with independent perspectives, if $d > l_B$ holds in at least one period then we can obtain correct IWTD in the image. In fact, according to [111], the ratio D/L_B is larger than 1.09 for all species in the book. That means using 2 independent wingbeat periods will achieve at least a successful rate of 0.777.

It is also worth noting that this probability lower bound in Lemma 1 is not a tight bound. In fact, failure cases only happen when the bird flies near parallel to the image plane with the IWTD near perpendicular to the image plane. This pose occurs in video with very small probability. Providing multiple periods, the probability of failure is even smaller, because the relative perspective between camera and bird keeps changing as the bird flies. From experiments, we find that one wingbeat period is sufficient for extracting IWTD for a majority of bird species.

Corollary 1 suggests that we can search IWTD across frames to find the frame when the bird fully extends its wing. Denote l_{ij} to be the Euclidean pixel distance between two boundary points i and j. For a particular frame t, we first extract an initial IWTD, denoted as $d_0(t)$:

$$d_0(t) = \max_{1 \le i,j \le m(t)} l_{ij}(t), \tag{3.24}$$

where $m(t)$ is the index set of points of the bird boundary in frame t. Its orientation $\eta_0(t)$ can be trivially computed. Figure 3.1(a) shows examples of $d_0(t)$'s in red dashed lines for a 9-frame sequence.

Then, we extract

$$d_{max}(t) = \max_{-\Delta \le i \le \Delta} d_0(t+i), \tag{3.25}$$

to be the IWTD for the moment that the bird fully extends its wing in the period centered at frame t. Δ is the half size of the searching window in terms of frames, and has a lower bound $\Delta \ge \frac{r}{2f_0} - \frac{1}{2}$ which ensures the sequence with frame rate r covers at least a period for the target species.

3.4.2.2 Recognizing IWTD series for the entire period

We introduce wing spreading direction (WSD) to describe the direction along which IWTD is to be extracted. In the image space, WSD in a frame is represented by the tilting angle of IWTD in that frame, denoted as $\eta(t)$. For a single period, WSD is viewed as a constant. Therefore, for a frame t, we can obtain its WSD by computing the angle of $d_{max}(t)$. In the example shown in Figure 3.1(a), frame $t+4$ has the maximum $d_0(t)$. Hence $\eta(t)$ is assigned by $\eta_0(t+4)$.

With WSD obtained, we can search for IWTDs. Since IWTD is the distance between extreme points on the bird, it should correspond to the longest distance between boundary points along the WSD in each frame (see Figure 3.2b). On the other hand, the actual WSD on each frame may be slightly different from WSD obtained from the maximum IWTD because the discrepancy caused by the discretization error of WSD due to the limited frame rate and by small changes in flying poses and camera perspectives exists. Therefore, $d(t)$ is obtained by searching a δ-neighborhood of the obtained WSD:

$$d(t) = \max_{|\varphi_{ij}(t)-\eta(t)|<\delta} l_{ij}(t) \tag{3.26}$$

where δ is a pre-set small threshold of angular difference, φ_{ij} is the orientation from point i to j. δ is selected to cover the aforementioned discrepancy. In our experiment, WSD searching range δ is set to 5°. It is worth noting that this procedure, to some extent, overcomes the self-occlusion problem when one of the wing tip is occluded by the bird body.

3.5 Periodicity Analysis

We show that $d(t)$ shares the same periodic property of the wingbeat motion regardless of camera parameters, so that a frequency analysis can be conducted. We begin with a kinematic model of the bird wing.

3.5.1 Kinematic Modeling of Bird Wings

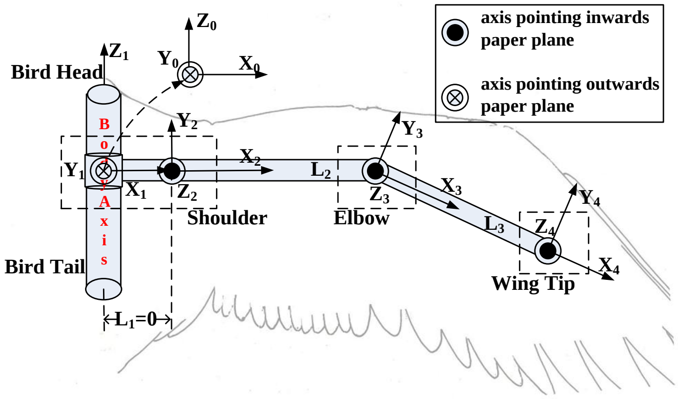


Figure 3.5: A kinematic model of the right wing of a bird.

Following the steady-flight skeleton model in [112], we model a bird wing using three revolute joints (see Figure 3.5). Frame 0 is the bird coordinate system (BCS) with its origin attached to the intersection point between the wing and the body axis of the bird and its Z-axis pointing to the direction of the bird head. Other frames are assigned by following Denavit-Hartenberg notations in [113], see Figure 3.5.

This model has 3 DOFs: joint angles θ_1 and θ_2 at the shoulder and θ_3 at the elbow. The lengths of upper- and fore- arms are L_2 and L_3, respectively. The coordinate of right wing tip in frame 4 is $[0, 0, 0, 1]^T$ in the homogeneous form. Applying the forward kinematics [113] to transform coordinates from frame 4 to frame 0, we have

$$\mathbf{X}_{rw} = \begin{bmatrix} L_2 c\theta_1 c\theta_2 + L_3 c\theta_1 c(\theta_2 + \theta_3) \\ L_2 s\theta_1 c\theta_2 + L_3 s\theta_1 c(\theta_2 + \theta_3) \\ L_2 s\theta_2 + L_3 s(\theta_2 + \theta_3) \\ 1 \end{bmatrix}, \tag{3.27}$$

where $c\theta$ means $\cos\theta$, $s\theta$ means $\sin\theta$, $c(\cdot)$ means $\cos(\cdot)$, and $s(\cdot)$ means $\sin(\cdot)$. Symmetrically, we can obtain left wing tip $\mathbf{X}_{lw}$ in BCS, which is the same as $\mathbf{X}_{rw}$ except that the first element is negative. Therefore, the IWTD in 3D space is

$$D = 2(L_2 c\theta_1 c\theta_2 + L_3 c\theta_1 c(\theta_2 + \theta_3)). \tag{3.28}$$

Now let us project D into the image coordinate. Since the distance from a flying bird to the camera is always significantly larger than the bird size, we can approximate the perspective projection using an affine camera model. Then, the camera transformation can be written as a 3×4 matrix P with its last row as $[0, 0, 0, 1]$.

Let $\mathbf{x}_{rw} := P\mathbf{X}_{rw}$ and $\mathbf{x}_{lw} := P\mathbf{X}_{lw}$ be right and left wing tip positions in the image, respectively. Recalling that $d = \mathbf{x}_{rw} - \mathbf{x}_{lw}$ is the distance between them, we have

$$d = 2(L_2 c\theta_1 c\theta_2 + L_3 c\theta_1 c(\theta_2 + \theta_3))\|\mathbf{p_1}\|_2 = D\|\mathbf{p_1}\|_2, \tag{3.29}$$

where $\mathbf{p_1}$ is the first column of P. Next we will show that d is a periodic function and reflects the WF.

3.5.2 Periodicity Analysis

In steady flight, a bird flaps its wings in a periodic pattern. Denote the period length as τ_0 and the corresponding circular frequency as ω_0. Pennycuick [106] shows that τ_0 and ω_0

are constants in steady flight. Liu et al. [112] show that all joint angle $\theta_i(t)$'s are periodic functions and can be expressed by a Fourier series,

$$\theta_i(t) = \alpha_i + \beta_i \sin(\omega_0 t + \phi_{i1}) + \gamma_i \sin(2\omega_0 t + \phi_{i2}), \tag{3.30}$$

where α_i, β_i, γ_i, ϕ_{i1}, and ϕ_{i2} are constants for $i = 1, 2, 3$. α_i's are phases. Since we only care about the basic WF (ω_0), we drop the harmonic frequency component in the last component and simplify (3.30) to the following,

$$\theta_i(t) = \alpha_i + \beta_i \sin(\omega_0 t + \phi_i). \tag{3.31}$$

Let τ_d be the period length of $D(t)$, we have the following.

Theorem 1. *For a bird in steady flight, the IWTD, $D(t)$, is a periodic function sharing the same period length of the wingbeat motion $\tau_d = \tau_0$ except that $\tau_d = \frac{1}{2}\tau_0$ if the following logic expression is true*

$$(\alpha_1 + \alpha_2 = k\pi) \cdot (\alpha_1 - \alpha_2 = k\pi) \cdot (\alpha_3 = k\pi),$$

where $k \in \mathcal{Z}$ and '$\cdot$' is 'AND' operator.

Considering the geometric constraints and limits on wing joints, we know: $\alpha_1 \in (-\pi/2, \pi/2)$ because the up-stroke/down-stroke of the wing can only reach to the angle that is perpendicular to the body plane, $\alpha_2 \in (-\pi/2, \pi/2)$ because the forward/backward moving of the wing can only reach parallel to the body axis, and similarly, $\alpha_3 \in (-\pi/2, \pi)$ and $\beta_i \in (0, \pi/2]$. Therefore, the degenerate angle in Theorem 1 occurs only when $\alpha_1 = \alpha_2 = \alpha_3 = 0$, which is of small probability.

To proof Theorem 1, we have the following two lemmas first.

Lemma 2. *Define function $f(t) = \cos(\alpha + \beta \sin(\omega t + \phi))$. Then it is a periodic function with the following period length*

$$\tau_f = \begin{cases} \tau & \textit{if } \alpha \neq k\pi \\ \frac{1}{2}\tau & \textit{if } \alpha = k\pi \end{cases} \tag{3.32}$$

where $k \in \mathcal{Z}$ and $\tau = 2\pi/\omega$, the integer set.

Proof. The function $f(t)$ repeats at least every τ time because

$$f(t+\tau) = \cos(\alpha + \beta \sin(\omega t + \omega\tau + \phi)) \tag{3.33}$$

$$= \cos(\alpha + \beta \sin(\omega t + 2\pi + \phi)) \tag{3.34}$$

$$= \cos(\alpha + \beta \sin(\omega t + \phi)) = f(t). \tag{3.35}$$

Suppose the period length of $f(t)$ is τ_f, it is trivial that $0 < \tau_f \leq \tau$. We also have the following equation for all t.

$$\cos(\alpha + \beta \sin(\omega t + \phi)) = \cos(\alpha + \beta \sin(\omega(t + \tau_f) + \phi)) \tag{3.36}$$

Considering $\beta \in (0, \pi/2]$, if (3.36) is true, then either of the following cases must be true:

$$\alpha + \beta \sin(\omega t + \phi) = \begin{cases} \text{case 1:} \quad \alpha + \beta \sin(\omega t + \omega\tau_f + \phi), \\ \text{case 2:} \quad -\alpha - \beta \sin(\omega t + \omega\tau_f + \phi) + 2k\pi, \end{cases} \tag{3.37}$$

where $k \in \mathcal{Z}$. When the first case in (3.37) happens, we have

$$\sin(\omega t + \phi) = \sin(\omega t + \omega\tau_f + \phi) \tag{3.38}$$

Then there are two solutions for (3.38):

$$\omega t + \phi = \omega t + \omega\tau_f + \phi + 2k'\pi, \tag{3.39}$$

and

$$\omega t + \phi = \pi - \omega t - \omega\tau_f - \phi + 2k'\pi, \tag{3.40}$$

where $k' \in \mathcal{Z}$. However, Eq. (3.40) cannot be true for all t because all parameters except t are constants. Therefore, only (3.39) is true and it becomes,

$$\tau_f = k'\tau. \tag{3.41}$$

Since $0 < \tau_f \leq \tau$, k' can only have the value of 1. Therefore, $\tau_f = \tau$, for all α in this case.

Similarly, if the second case in (3.37) happens, we can prove

$$\tau_f = \frac{(2k'-1)}{2}\tau = \frac{1}{2}\tau \text{ , and } \alpha = k\pi \tag{3.42}$$

Combining (3.41) and (3.42), Lemma 2 is proved. □

Before we introduce the next lemma, let us define the following functions to simplify notations,

$$\begin{aligned}
\Psi_c(\alpha,\beta,\phi) &= f(t) \\
\Psi_s(\alpha,\beta,\phi) &= \sin(\alpha + \beta\sin(\omega t + \phi)) \\
\tau = 2\pi/\omega, &\quad \alpha_{1\pm 2} = \alpha_1 \pm \alpha_2 \\
g(t) &= \Psi_c(\alpha_1,\beta_1,\phi_1)\Psi_c(\alpha_2,\beta_2,\phi_2),
\end{aligned} \tag{3.43}$$

where $\beta_i \in (0, \pi/2]$. Then we have the following lemma,

Lemma 3. *Function $g(t)$ is a periodic function with its period length $\tau_g = \tau$, except when Boolean function $\Gamma(\alpha_1, \alpha_2, \beta_1, \beta_2, \phi_1, \phi_2)$ is true where $\Gamma(\alpha_1, \alpha_2, \beta_1, \beta_2, \phi_1, \phi_2) = \Gamma_1 + \Gamma_2 + \Gamma_3$, '$+$' is logical 'OR', and*

$$\begin{aligned}
&\Gamma_1 = (\alpha_{1+2} = k_1\pi) \cdot (\alpha_{1-2} = k_2\pi), \\
&\Gamma_2 = (\beta_1 = \beta_2) \cdot (\phi_1 = \phi_2 + (2k_1 + 1)\pi) \cdot (\alpha_{1-2} = k_2\pi), \\
&\Gamma_3 = (\beta_1 = \beta_2) \cdot (\phi_1 = \phi_2 + 2k_1\pi) \cdot (\alpha_{1+2} = k_2\pi),
\end{aligned}$$

where '$\cdot$' is logical 'AND' and $k_1, k_2 \in \mathcal{Z}$.

Proof. Let us decompose $g(t)$,

$$\begin{aligned}
g(t) = &\, c\alpha_1 c\alpha_2 \Psi_c(0, \beta_1, \phi_1)\Psi_c(0, \beta_2, \phi_2) \\
&- s\alpha_1 c\alpha_2 \Psi_s(0, \beta_1, \phi_1)\Psi_c(0, \beta_2, \phi_2) \\
&- c\alpha_1 s\alpha_2 \Psi_c(0, \beta_1, \phi_1)\Psi_s(0, \beta_2, \phi_2) \\
&+ s\alpha_1 s\alpha_2 \Psi_s(0, \beta_1, \phi_1)\Psi_s(0, \beta_2, \phi_2)
\end{aligned} \tag{3.44}$$

and define the following intermediate variables,

$$\begin{aligned}
\kappa_1 = \beta_1 c(\phi_1) + \beta_2 c(\phi_2); &\quad \kappa_2 = \beta_1 s(\phi_1) + \beta_2 s(\phi_2); \\
\kappa_3 = \beta_1 c(\phi_1) - \beta_2 c(\phi_2); &\quad \kappa_4 = \beta_1 s(\phi_1) - \beta_2 s(\phi_2); \\
\kappa_{12} = \sqrt{\kappa_1^2 + \kappa_2^2}; &\quad \kappa_{34} = \sqrt{\kappa_3^2 + \kappa_4^2}; \\
\phi_{\kappa_{12}} = \arctan(\kappa_2/\kappa_1); &\quad \phi_{\kappa_{34}} = \arctan(\kappa_4/\kappa_3);
\end{aligned}$$

Then, (3.44) can be transformed to

$$\frac{1}{4}(\Psi_c(\alpha_{1+2}, \kappa_{12}, \phi_{\kappa_{12}}) + \Psi_c(\alpha_{1-2}, \kappa_{34}, \phi_{\kappa_{34}})) \tag{3.45}$$

We have the following cases:

- $\kappa_{12} = 0$: This happens if and only if $\beta_1 = \beta_2$ and $\phi_1 = \phi_2 + (2k_1 + 1)\pi$. Then, from Lemma 2, $\tau_g = \tau$ unless $\alpha_{1-2} = k_2\pi$.

- $\kappa_{34} = 0$: This happens if and only if $\beta_1 = \beta_2$ and $\phi_1 = \phi_2 + 2k_1\pi$. Then $\tau_g = \tau$ unless $\alpha_{1+2} = k_2\pi$.

- Otherwise, $\kappa_{12} \neq 0$ and $\kappa_{34} \neq 0$ Then, the first component of (3.45) has period length of $\tau/2$ only when $\alpha_{1+2} = k_1\pi$. The second component has period length $\tau/2$ only when $\alpha_{1-2} = k_2\pi$. Therefore, $g(t)$ has period length $\tau_g = \tau$ unless $\alpha_{1+2} = k_1\pi$ and $\alpha_{1-2} = k_2\pi$.

Therefore, Lemma 3 is proved. □

With the two lemmas, the proof of Theorem 1 is simple.

Proof. Eq. (3.28) has two periodic components: the first part is $c\theta_1 c\theta_2$ and the second part is $c\theta_1 c(\theta_2 + \theta_3)$. Denote τ_{d1} and τ_{d2} to be the period lengths of the first and the second parts, respectively. For $c\theta_1 c\theta_2$, we can apply the two lemmas and obtain the following,

$$\tau_{d1} = \begin{cases} \frac{1}{2}\tau_0 & \text{If } \Gamma(\alpha_1, \alpha_2, \beta_1, \beta_2, \phi_1, \phi_2) \text{ is true,} \\ \tau_0 & \text{otherwise.} \end{cases}$$

For $c\theta_1 c(\theta_2 + \theta_3)$, let us define the following variables,

$$\kappa_5 = \beta_2 c(\phi_2) + \beta_3 c(\phi_3); \quad \kappa_6 = \beta_2 s(\phi_2) + \beta_3 s(\phi_3);$$
$$\kappa_{56} = \sqrt{\kappa_5^2 + \kappa_6^2}; \qquad \phi_{\kappa_{56}} = \arctan(\kappa_6/\kappa_5).$$

Now we can again apply the two lemmas and have

$$\tau_{d2} = \begin{cases} \frac{1}{2}\tau_0 & \text{If } \Gamma(\alpha_1, \alpha_{2+3}, \beta_1, \kappa_{56}, \phi_1, \phi_{\kappa_{56}}) \text{ is true,} \\ \tau_0 & \text{otherwise.} \end{cases}$$

In steady flight, we know $\beta_3 \neq 0$ because the elbow joint does not fix at an angle. Therefore τ_d should be the least common multiple of τ_{d1} and τ_{d2}. Because $\beta_1 = \beta_2$ and $\beta_1 = \sqrt{\kappa_5^2 + \kappa_6^2}$ do not happen simultaneously, Theorem 1 is proved. □

Remark 1. *For a fixed camera w.r.t the bird, the projective matrix does not change. Therefore, $\|\mathbf{p_1}\|_2$ remains constant and $d(t)$ shares the period length with $D(t)$ based on (3.29).*

Remark 2. *If the camera or the bird moves, the changing of perspective introduces the frequency distribution of $\|\mathbf{p_1}\|_2(t)$, and the frequency property of $d(t)$ should be the convolution of the bird motion and the camera motion. As long as the changing of the camera perspective is not strictly periodic, the convolution preserves the dominant frequency component [88] of wing flapping motions, except for a few isolated special degenerate cases. This ensures that we can obtain WF f_0 by applying FFT to the extracted $d(t)$.*

Actually, camera motions are usually slow when people track a bird at a distance. Most birds have a WF significantly higher than 1 Hz. Using a high pass filter of 1 Hz, we filter out the noise introduced by bird gliding and camera motion while preserving WF. Then WF can be extracted by the first energy peak above frequency threshold. Figure 3.1(c) shows the extracted WF and the frequency distribution of the signal from video in Figure 3.1(a).

In theorem, we proved the existence of energy peak at f_0. The harmonic component in eq. 3.30 could lead to another energy peak at $2f_0$, under similar analysis with Theorem 1. We omit this part in the above analysis because the proof is similar but over lengthy. The relative height of the two peaks is affected by β_i's and γ_i's in the model, which is uncertain because different birds have different parameter configurations of their wing models.

In observation, we discover that the highest energy peak above frequency threshold always corresponds to one of these two peaks (f_0 or $2f_0$). Therefore, In practice, we extract WF in a more robust way: 1) finding the frequency f_0 with the highest energy peak and 2) resetting $f_0 = f_0/2$ if there exists another peak at $f_0/2$.

3.6 Species Prediction

Due to noise and discreteness, we perform a variance-based error analysis before the actual species detection with trustable measurements.

Step 1: Error Bound Analysis: Due to the discreteness in frequency domain, the extracted WF has an error bound (f_e) equal to the half of the frequency interval after FFT:

$$f_e = \frac{r}{2N}. \tag{3.46}$$

where N is total number of frames rounded up to a power of 2, and r is the frame rate. Eq. (3.46) is quite intuitive. For a video clip with a fixed frame rate, the more frames we have, the smaller error we can get. Since the extracted WF is uniformly distributed within the error bound, the variance of the extracted WF is

$$Var(f_0) = \frac{1}{12}((f_0 + f_e) - (f_0 - f_e))^2 = \frac{1}{3}f_e^2. \tag{3.47}$$

For a known species s, its reference WF from the aforementioned prior knowledge has a variance of σ_s^2. We believe that a measured WF is reliable only if its variance is less than that of the reference. Hence we establish the error bound for reliable measurements:

Definition 3 (Error Bound for Measurements)**.** *An extracted WF measurement is trustable for species prediction if*

$$\frac{1}{3}f_e^2 \leq \sigma_s^2 \tag{3.48}$$

The species prediction is only performed using trustable measurements. Since more frames mean a smaller f_e, the least number of frames for a fixed rate video can be calculated inversely. For a 30 fps video, 100 frames approximately result in a measurement variance of 0.1 Hz, which is comparable to that of most species.

Step 2: Species Prediction: Had f_0 been error-free, the likelihood that the bird belongs to a species $\{\mu_s, \sigma_s\}$ is

$$L(\mu_s, \sigma_s | f_0) = \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{-\frac{(f_0-\mu_s)^2}{2\sigma_s^2}}. \tag{3.49}$$

However, the true WF is uniformly distributed in $(f_0 - f_e, f_0 + f_e)$, the likelihood function becomes

$$L'(\mu_s, \sigma_s | f_0, f_e) = \int_{f_0-f_e}^{f_0+f_e} \frac{1}{2f_e} L(\mu_s, \sigma_s | f) df. \tag{3.50}$$

Define $G(\cdot)$ as the cumulative probability function for the Gaussian distribution. Then we have,

$$L'(\mu_s, \sigma_s | f_0, f_e) = \frac{1}{2f_e}[G(\frac{f_0 + f_e - \mu_s}{\sigma_s\sqrt{2}}) - G(\frac{f_0 - f_e - \mu_s}{\sigma_s\sqrt{2}})]. \tag{3.51}$$

As the metric for species prediction, the likelihood is used to rank all candidate species. The resulting ranked list is the species prediction outcome. The reason for keeping a short candidate list instead of reporting only the top ranked candidate is that some species share close or equal WF distributions, and it is not desired to miss many false negative predictions.

3.7 Algorithms

To summarize the proposed method, we present two algorithms: Salient Extremity Recognition Algorithm (SERA) and Species Prediction Algorithm (SPA), which correspond to the two problems defined in Section 3.3. The motion segmentation algorithm described in Section 3.4.1 outputs bird boundary points for each frame, which are the input to SERA. As its output, SEAR returns the sequence of extracted 2D IWTD as detailed in Algorithm 1.

Algorithm 1 SERA

Input: N sets of bird boundary points
Output: a sequence of 2D IWTD

```
  $d_0[N], d_{max}[N], d[N] \leftarrow 0$                                $O(N)$
  $\eta_0[N], \eta[N] \leftarrow 0$                                      $O(N)$
  for all frame $i$ in $1, ..., N$ do
    calculate $d_0(i), \eta_0(i)$ according to (3.24)
  end for                                                        $O(M^2N)$
  for frame $i = 1$ to $N$ do
    if $i \leq \Delta$ or $i > n - \Delta$ then
      $\eta(i) = \eta_0(i)$
    else
      $d_{max}(i) = \max_{-\Delta \leq j \leq \Delta} d_0(i + j)$
      $\eta(i) \leftarrow$ the tilting angle of $d_{max}(i)$
    end if
    calculate $d(i)$ according to (3.26)
  end for                                                        $O(M^2N)$
  return $d[N]$
```

From the pseudo code, we have the following theorem.

Theorem 2. *Given a motion segmented video with N frames, M is the maximum number of boundary points for a bird in a frame ($M = \max_{1 \leq t \leq N} |m(t)|$), then the overall time complexity of salient extremity recognition algorithm is $O(M^2N)$.*

It is worth noting that since the bird size is usually small in image, M is approximately between 10^2 and 10^3. For species prediction, Algorithm 2 details the pseudo code, which leads to the following complexity result.

Theorem 3. *SPA runs in $O(N \log N + |\mathcal{S}| \log |\mathcal{S}|)$ time, where N is the number frames in the video and $|\mathcal{S}|$ is the size of the reference species set.*

Algorithm 2 Species Prediction

```
Input: $d[N]$, Candidate species set $\mathcal{S}$, return list length $l$
Output: a ranked list of potential species
  compute FFT on sequence $d[N]$                                                    $O(N \log N)$
  high-pass filtering
  $peak\_set \leftarrow$ FindEnergyPeaks($d$)
  if $peak\_set$ is empty then
    return $-1$
  end if
  $f_0 \leftarrow$ FindMaxPeak($peak\_set$)                                          $O(N)$
  if $f_0/2$ is in $peak\_set$ then
    $f_0 = f_0/2$                                                                   $O(1)$
  end if
  $f_e = \frac{r}{2N}$ according to (3.46)                                          $O(1)$
  likelihood arary $L[N_S] \leftarrow 0$                                            $O(|\mathcal{S}|)$
  likelihood rank index array $I[N_S] \leftarrow 0$                                 $O(|\mathcal{S}|)$
  for all $s \in \mathcal{S}$ do
    if (3.48) then
      calculate $L(s)$ by (3.51)
    else
      $L(s) = 0$
    end if
  end for                                                                           $O(|\mathcal{S}|)$
  sort $L[N_S]$ in descend order, record corresponding species index $I[N_S]$   $O(|\mathcal{S}| \log |\mathcal{S}|)$
  return first $[I(1), ..., I(l)]$ candidate species
```

3.8 Experiments

We have implemented the two algorithms using Matlab on a PC. The *prior knowledge* (extended version of Table 3.1) from [107] contains WF means and variances for 32 different species of birds. Their WF means vary from 2.24 Hz to 9.19 Hz. Since there is no existing video data set to benchmark and compare bird species recognition methods, we collect our data from online video. Original videos are downloaded from YouTube and Internet Bird Collection (`http://ibc.lynxeds.com/`). All videos are recorded by moving cameras. Video 1 to 18 contain different flying birds, covering 6 species in [107]. Video 19 to 27 are non-bird videos including 4 air-planes, 3 moving clouds and 2 walking human. The video dataset consists of 378 flying periods, and 5006 video frames in total. Frame-rates of the videos vary from 15 fps to 30 fps. The IWTDs of the birds in the video range from 105 cm to 229 cm while WFs range from 2.24 to 4.58 Hz. It is worth noting that this WF range covers a majority of bird species ($> 60\%$) which makes it a challenging data set because there are many overlapping WFs among species.

3.8.1 IWTD and WF Extraction

Our algorithm successfully extracts IWTD series, their WFs, and their WF error bounds. Table 3.2 shows the number of wingbeat periods (NoWP) in each video. In fact, we only need one period to recognize IWTD for each frame, and the obtained IWTD series is successful for WF extraction, which agrees with the prediction given by Corallary 1. Figure 3.6 visualizes how the extracted WFs are covered within the true WF distributions. Video 1 to 18 are bird videos, Figure 3.6 shows that the extract WFs are mostly covered in 2σ of the true species WF, and therefore lead to high likelihood of true species, except Video 7. Video 19 to 27 are non-bird videos, for each of which, the Species Prediction algorithm returns empty set for the energy peak. Video 19 to 22 contain a flying air-plane. Motion segmentation is able to extract the plane from background, however, frequency extraction fails because there is no strong energy peak in their spectrum. Video 23 to 25 contain moving clouds. Since clouds move very slow w.r.t. background, the motion segmentation step returns no foreground, thus no measurement is extracted. For video 26 and 27 of walking human, the measurement along its salient dimension is the height of the human and does not provide energy peak in spectrum. Overall, the results show that the system is capable of extracting WFs from different camera perspectives. It also shows that WF is a robust signature for the species recognition.

Table 3.2: RoCS, MLR, and NoWP for testing videos

video	1	2	3	4	5	6
MLR	0.096	0	0.1485	0.1094	0.0609	0.128
RoCS	2	4	2	5	2	2
NoWP	45	19	10	12	15	20

video	7	8	9	10	11	12
MLR	0.1667	0.1	0.0472	0.1714	0.0418	0.171
RoCS	8	8	1	1	1	9
NoWP	18	39	8	10	12	12

video	13	14	15	16	17	18
MLR	0.2904	0.2467	0.0615	0.2541	0.2577	0.3034
RoCS	2	2	6	9	2	3
NoWP	9	28	15	15	79	12

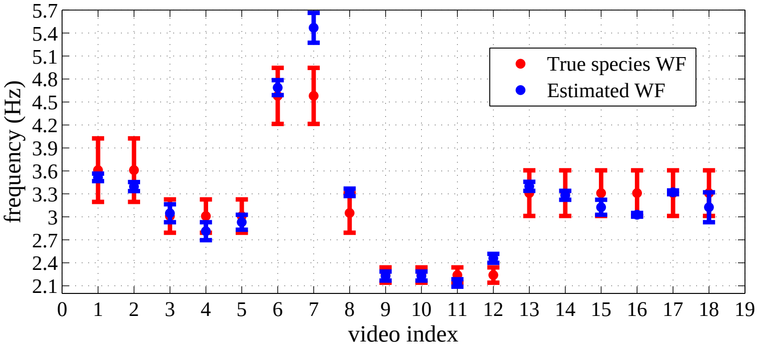


Figure 3.6: Comparison between true species WF and extracted WF. Red bars shows the frequency covered in $\mu \pm 2\sigma$ of the true species, blue bars shows the frequency covered in the extracted $f_0 \pm f_e$ of the target bird. Since no frequency is extracted from video 19 to 27, no bars are drawn for these videos

The WF extracted from video 7 shows the WF of the bird in the video, however, this bird is flying at a different wingbeat frequency from its species distribution. Statistically, individuals that are far from the center of their species distribution exist with minor probability. For those minor cases, such as video 7, WF only is not sufficient for species recognition.

3.8.2 Robustness to Segmentation Error

Since our method relies on the extraction of pixel distance, the temporal feature is inevitably affected by the foreground segmentation error at the bird wing tip. This error happens when image resolution is low or motion blur appears. The error influences the accuracy of pixel distance $d(t)$. We use simulation to evaluate on how segmentation errors affect WF results. Considering the segmentation error at a wing tip to follow a zero-mean Gaussian distribution, the Euclidean distance between wing tips follows Gaussian distribution as well. The simulation is conducted on a real signal from test video 11, where we manually annotated the wing tip positions in every image in the video. A sequence of $d(t)$ is therefore calculated upon the annotation and treated as a low noise ground truth signal as illustrated by the blue solid curve in Figure 3.7. Mean value of this signal is subtracted for illustration purpose. The maximum and the minimum values in $d(t)$ are 154.1 and 60.5, respectively, while the mean of $d(t)$ is 108.82. Different levels of Gaussian noise are added to the signal. The red dotted curve in Figure 3.7 shows the simulated signal when the error standard deviation is 10. We gradually increase the noise standard deviation and measure the ratio between the WF peak energy and the average spectrum energy (Figure 3.8). It is shown that with noise standard derivation varying from 0 to 100 pixels, the WF energy is still larger than average spectrum energy. While in our experiments in previous subsection, the mean segmentation error of this sequence is 4.12 pixels, and the maximum error in a frame is 37.06 pixels, which are much smaller than the simulated error. This simulation demonstrates the robustness of the proposed WF extraction method in the presence of segmentation errors.

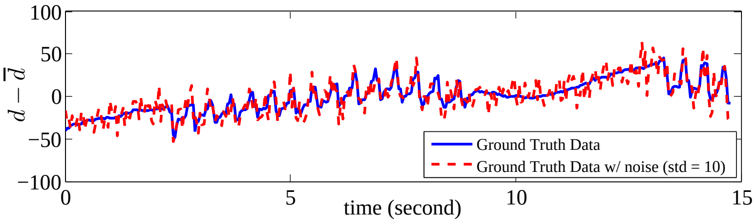


Figure 3.7: Simulation: injecting segmentation error to the ground truth data in simulation. Blue solid curve: the ground true of $d(t) - \bar{d}$. Red dotted curve: after adding Gaussian noise with zero mean and a standard deviation of 10 pixels to the blue curve.

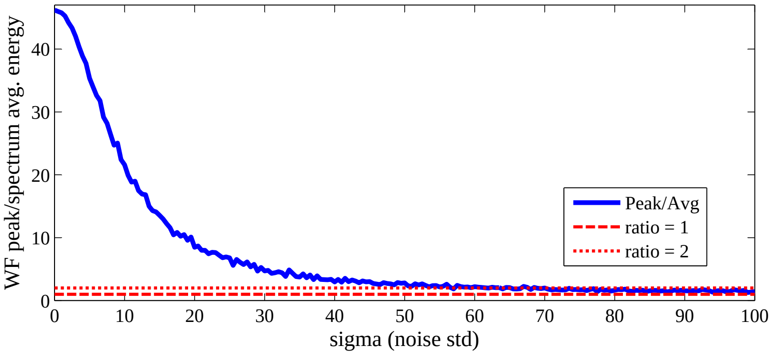


Figure 3.8: Simulation: signal energy vs. background energy. The ratio of WF peak and the average of spectrum energy, as the noise deviation increases from 0 to 100. The ratio is always above 1 and is above 2 when noise deviation is lower than 55 pixels.

3.8.3 Species Prediction

To evaluate the accuracy of the ranked candidate list, we define hit rate as the percentage of returned candidate lists that contain the correct species. To our best knowledge, there is no existing algorithm for flying bird species recognition for videos taken by moving cameras. Previous methods on object recognition or motion analysis cannot be directly applied on the bird species recognition problem. Therefore, the comparison experiment is compared with random guess only. We compare our algorithm output with a short list of the same length which is generated from independent random guesses from the 32 candidate species. The results are showed in Figure 3.9. It is clear that our algorithm significantly outperforms the random guess.

The rank of the correct species (RoCS) for each video is showed in Table 3.2. Combining with Figure 3.6, Table 3.2 shows that as long as the extracted WF is close to the mean of its species distribution, the proposed likelihood metric is able to rank the true species in the top of the candidate list. This proves the feasibility of the proposed species prediction algorithm. For the birds with WF far from its species distribution, such as video 7, 8 and 12, their RoCS's are not high (still among top 1/3), due to similar WFs between some species. However, in real application, the rankings can be improved by adding prior knowledge of local bird distribution, because different bird species have different inhabitation regions and species with similar WFs may not appear in the same geometric region.

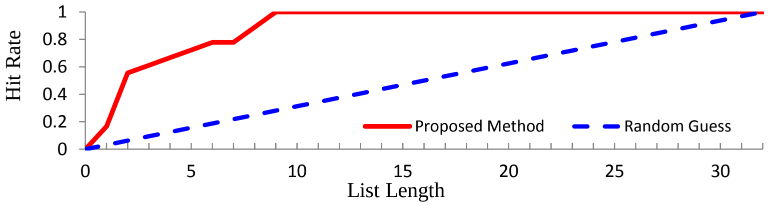


Figure 3.9: Hit rate vs. list length.

3.8.4 Robustness to Data Loss

Inevitably, some frames of bird videos may be too blurred to segment the bird which leads to the loss of IWTD measurements. If so, our system assigns the measurement of this frame using its nearest successful antecedent. Our frequency-based analysis is very robust to data loss. The measurement lost rate (MLR) in each testing video is listed in Table 3.2. The loss rate varies from 0 to 30%, and the maximum number of consecutive lost frames is 21 in our data. Table 3.2 shows that the RoCS's are not obviously influenced by the MLR, since for the video with most data lost (video 18), its RoCS is still among the top three. This proves the robustness of the proposed frequency based species prediction approach.

3.9 Conclusion and Future Work

We developed the bird species filtering method that takes crowd sourced videos from cameras with unknown parameters as input and outputs likelihood of candidate species. The method first recognized and extracted the time series of salient extremities from the videos without prior knowledge on camera motion and perspective changes. The second algorithm applied FFT to observed IWTD series to obtain wingbeat frequency. We also proposed a species prediction metric using likelihood ratio. We have implemented the algorithm and tested it in experiments which validated our design and analysis.

In the future, we will develop recognition methods using other features such as flying speed, color and shape in combination with frequency signatures to achieve more precise prediction. Note that the method also has the potential to be applied to other animals with frequency characteristics.

4. AUTOMATIC MULTI-MODEL BIRD SPECIES FILTERING USING VIDEOS FROM UNCALIBRATED CAMERAS

4.1 Introduction

For objects that process a mixture of behaviours, simply using single motion property, such as periodic motion, may be sufficient for tracking and recognition. The flying birds provide a good example of a combination of wing flapping and gliding motion. In this section, we propose a multi-model approach to catch complex object behaviour patterns and different environment. Figure 4.1 shows the system diagram. This multi-model approach tracks both the body and wing motion of a flying bird to form signatures for species filtering. Before elaborating our model, we discuss related work on bird species detection and monocular tracking.

4.2 Related Work

Our bird species filtering problem relates to vision-based animal recognition and monocular 3D object tracking.

Algorithms have been introduced for automatic wildlife observation, based on audio signals [114], static images [115], and videos [10]. For vision-based wildlife recognition, 2D features, such as point and texture [10], 2D kinematic chain [116] and templates [16], are extracted and matched for tracking and learning. However, most of the existing animal detection works focus on the classification of animal genus, such as dog, lion or bird, and are not sufficient for species filtering.

Our bird species filtering relies on monocular tracking of the bird motion in videos. The most relevant monocular 3D tracking approaches fall into two categories: model-based bundle adjustment and probabilistic filtering. The former has been used in piecewise tracking, based on the extraction of point [117], edge and texture [118] correspondences.

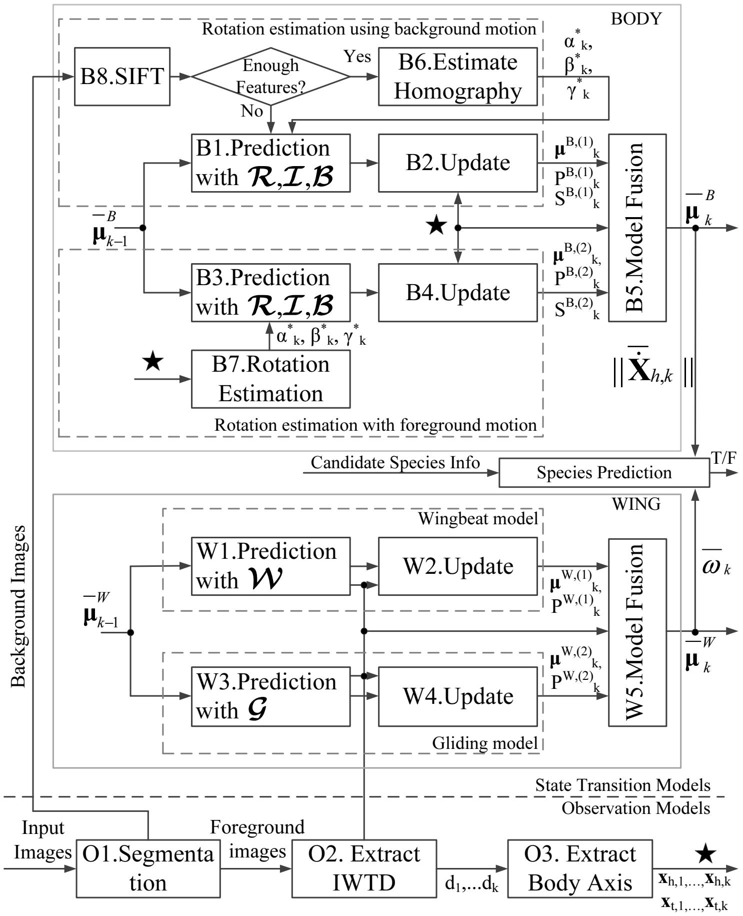

BODY
Rotation estimation using background motion
B8.SIFT
Enough Features?
Yes
No
B6.Estimate Homography
α^*_k, β^*_k, γ^*_k
B1.Prediction with $\mathcal{R},\mathcal{I},\mathcal{B}$
B2.Update
$\boldsymbol{\mu}^{B,(1)}_k$, $P^{B,(1)}_k$, $S^{B,(1)}_k$
$\bar{\boldsymbol{\mu}}^B_{k-1}$
★
B5.Model Fusion
$\bar{\boldsymbol{\mu}}^B_k$
$\boldsymbol{\mu}^{B,(2)}_k$, $P^{B,(2)}_k$, $S^{B,(2)}_k$
B3.Prediction with $\mathcal{R},\mathcal{I},\mathcal{B}$
B4.Update
α^*_k, β^*_k, γ^*_k
★
B7.Rotation Estimation
$\|\bar{\dot{\mathbf{X}}}_{h,k}\|$
Rotation estimation with foreground motion
Candidate Species Info
Species Prediction
T/F
WING
Wingbeat model
W1.Prediction with $\mathcal{W}$
W2.Update
$\boldsymbol{\mu}^{W,(1)}_k$, $P^{W,(1)}_k$
$\bar{\omega}_k$
$\bar{\boldsymbol{\mu}}^W_{k-1}$
W5.Model Fusion
$\bar{\boldsymbol{\mu}}^W_k$
W3.Prediction with $\mathcal{G}$
W4.Update
$\boldsymbol{\mu}^{W,(2)}_k$, $P^{W,(2)}_k$
Gliding model
Background Images
State Transition Models
Observation Models
Input Images
O1.Segmentation
Foreground images
O2. Extract IWTD
$d_1,...d_k$
O3. Extract Body Axis
★
$\mathbf{x}_{h,1},...,\mathbf{x}_{h,k}$
$\mathbf{x}_{t,1},...,\mathbf{x}_{t,k}$

Figure 4.1: System diagram. The solid star represents the output of body axis extraction, which is also the observation input to body motion model.

Special properties such as planar homography [119] are utilized to facilitate correspondence estimation. These methods are proved to have stable performance when sufficient corresponding features are available across multiple frames. However, in our problem, the number of features is often limited, because of the far field view of the bird and feature-less sky.

The probabilistic filtering [120, 121] builds on Bayesian methods such as extended Kalman filter (EKF) and particle filter (PF), for tracking. The IMM-EKF [122] is introduced to fuse multiple models, which provides an open framework to cover cases such as the object performing multiple behaviors [123, 124], observations come from different types of sensors [125], and tracking with different sampling periods [126]. Here we extend the IMM-EKF filtering for our bird species filter. Our IMM-EKF can simultaneously estimate intrinsic matrix of the camera, track object motion, and separate the object motion from camera motion to recover both in a fixed world coordinate system.

Our group has developed systems and algorithms for bird species filtering. Our previous work [127] introduces an autonomous bird observation system using motion detection. We propose [103] a bird species recognition algorithm based on tracking of the bird flying speed with a stationary calibrated camera. To handle uncalibrated cameras with unknown motions, our recent work [128] presents a frequency based method which recognizes wing flapping motion. However, the method still suffers high FP issue. To overcome the issue without imposing more assumptions on camera parameters, we present a new combined framework utilizing both body and wing motion signatures to significantly reduce the FP rate.

4.3 Problem Definition

4.3.1 Assumptions

We assume input video frames contain only one flying bird. If multiple birds appear, each individual bird sequence is separated beforehand, by applying existing multiple target tracking methods such as [104]. We assume the bird is in steady flight under normal weather condition, which includes gliding, level-flying and cruising, but excludes landing, taking off and diving. Video length is sufficiently long to cover wing flapping motion. The uncalibrated camera is assumed to have constant intrinsic parameters. The camera translation is assumed to be negligible in comparison with object motion, so that the camera motion can be approximated by pure rotation with a same camera center. This is usually satisfied because birds are far-field and fast-moving objects. The camera follows a pinhole model and its intrinsic matrix can be approximated as

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

where f is the focal length in pixel units, and (u_0, v_0) are the coordinates of the principal point in pixel units.

4.3.2 Notations and Coordinate System

Let $\{C_k\}$ be the camera coordinate system (CCS) at frame k. Each $\{C_k\}$ has its origin located at the optical center, its x and y axes parallel to u and v axes of the image coordinate system, respectively, and its z axis being the camera optical axis pointing from the origin to infinity and perpendicular to the image plane. $\mathbf{X}_k$ denotes the position of a 3D point in $\{C_k\}$. The camera motion from $\{C_{k-1}\}$ to $\{C_k\}$ is represented by a rotation $R(\alpha, \beta, \gamma)$ in Y-X-Z Euler angle convention, where α, β and γ denote the rotation angles about x, y and z axes, respectively. The world coordinate system $\{W\}$ coincides with $\{C_0\}$.

The image coordinate system (ICS) has its origin at the center of the image, and its u and v axes parallel to the horizontal and vertical directions of the image, respectively. A point in frame k is denoted as $\mathbf{x}_k = [u, v, 1]^\mathsf{T}$.

For all variables in this paper, subscripts h and t mean the extreme point of bird head and tail, respectively.

Therefore, the problem is defined as follows:

Definition 4. *Given an image sequence of a flying bird, and the information of a candidate species including its average body length (L), flying speed range ($[V_{min}, V_{max}]$) and wingbeat frequency range ($[F_{min}, F_{max}]$), track the motion of the bird and camera, and predict whether the bird belongs to the candidate species or not.*

4.4 System Overview

As illustrated in Figure 4.1, the overall system is composed of a state transition model and an observation model.

The observation model is illustrated at the bottom of Figure 4.1. Image segmentation (Box O1 in Figure 4.1) using method in Section 3 is performed on input video frames, resulting in separated foreground and background images. Foreground images are utilized to extract inter-wing tip distance (IWTD) (Box O2 in Figure 4.1) and body axis (Box O3 in Figure 4.1) measurements. Background images are passed to scale invariant feature transform (SIFT) keypoint extraction (Box B8 in Figure 4.1).

The state transition model has two subsystems: bird body motion model (BODY in Figure 4.1) and wing motion model (WING in Figure 4.1). The BODY model tracks the 3D motion of the bird body and the unknown camera parameters simultaneously. In order to reduce tracking ambiguity, two methods for camera rotation estimation are proposed using different image information (Boxes B6 and B7 in Figure 4.1). The transition results are fused in Box B5 in Figure 4.1. The WING model tracks the wing motion of the bird. Since a flying bird may perform either gliding or wing flapping motion, two different behavior models are used in EKF prediction, to describe both translational (Box W3 in Figure 4.1) and periodic (Box W1 in Figure 4.1) patterns. The transition results are fused in Box W5 in Figure 4.1. The tracking results from both BODY and WING are utilized in species prediction w.r.t. a candidate bird species.

To handle the fusion of different state transition methods, both BODY and WING models employ the IMM-EKF framework. Before elaborating our models, let us briefly review the IMM-EKF framework [122]. Let $\boldsymbol{\mu}$ and P denote the state vector and its covariance for a model, $\mathbf{z}$ denote the observation, and w denote the model fusion weight. A $\hat{}$ stands for the predicted value, and a $^{-}$stands for the output of model fusion. As a convention in this paper, a superscript j on a variable stands for model j in the corresponding subsystem, and a subscript k denotes the variable at frame k. Let G and H be Jacobians of the state transition function g and the observation function h, respectively. Assuming uniform model transition probability, an IMM-EKF framework contains the following:

- EKF Prediction:

$$\hat{\boldsymbol{\mu}}_k^{(j)} = g^{(j)}(\bar{\boldsymbol{\mu}}_{k-1}), \tag{4.2}$$

$$\hat{P}_k^{(j)} = G_k^{(j)} \bar{P}_{k-1} (G_k^{(j)})^\mathsf{T} + \Omega_k, \tag{4.3}$$

- EKF Update:

$$S_k^{(j)} = H_k \hat{P}_k^{(j)} H_k^\mathsf{T} + Q_k, \tag{4.4}$$

$$\boldsymbol{\mu}_k^{(j)} = \hat{\boldsymbol{\mu}}_k^{(j)} + \hat{P}_k^{(j)} H_k^\mathsf{T} (S_k^{(j)})^{-1} (\mathbf{z}_k - h(\hat{\boldsymbol{\mu}}_k^{(j)})), \tag{4.5}$$

$$P_k^{(j)} = (I - \hat{P}_k^{(j)} H_k^\mathsf{T} (S_k^{(j)})^{-1} H_k) \hat{P}_k^{(j)}, \tag{4.6}$$

- Model Fusion:

$$\textit{update weight } w_k^{(j)}, \tag{4.7}$$

$$\bar{\boldsymbol{\mu}}_k = \Sigma_j \boldsymbol{\mu}_k^{(j)} w_k^{(j)}, \tag{4.8}$$

$$\bar{P}_k = \Sigma_j w_k^{(j)} (P_k^{(j)} + (\boldsymbol{\mu}_k^{(j)} - \bar{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k^{(j)} - \bar{\boldsymbol{\mu}}_k)^\mathsf{T}), \tag{4.9}$$

where Ω_k and Q_k denote the covariance matrices of system transition noise and observation noise, respectively. Weight $w_k^{(j)}$ distribution over each model j is determined by application and will be discussed later.

4.5 Body Motion Model

The body motion model tracks the 3D motion of the bird body in $\{C_k\}$ and the unknown camera intrinsic parameters (f, u_0, v_0) and extrinsic parameters (α, β, γ). Using a superscript B to indicate system variables in the body motion model, we have state variables

$$\boldsymbol{\mu}^B = [\mathbf{X}_h^\mathsf{T}, \dot{\mathbf{X}}_h^\mathsf{T}, \ddot{\mathbf{X}}_h^\mathsf{T}, f, u_0, v_0, \alpha, \beta, \gamma]^\mathsf{T}, \tag{4.10}$$

where $\mathbf{X}_h$, $\dot{\mathbf{X}}_h$ and $\ddot{\mathbf{X}}_h$ describe the position, velocity, and acceleration of the bird head, respectively.

4.5.1 State Transition Model for Body Motion

Eq. (4.2) of $\boldsymbol{\mu}^B$ can be divided into three subsystems corresponding to the three groups of state variables.

$$\boldsymbol{\mathcal{I}} : \begin{cases} \hat{f}_k = \bar{f}_{k-1} \\ \hat{u}_{0,k} = \bar{u}_{0,k-1} \\ \hat{v}_{0,k} = \bar{v}_{0,k-1} \end{cases},$$

$$\boldsymbol{\mathcal{R}} : \begin{cases} \hat{\alpha}_k = \bar{\alpha}_{k-1} \\ \hat{\beta}_k = \bar{\beta}_{k-1} \\ \hat{\gamma}_k = \bar{\gamma}_{k-1} \end{cases}, \text{ and}$$

$$\boldsymbol{\mathcal{B}} : \begin{cases} \hat{\mathbf{X}}_{h,k} = R_k^{-1}(\bar{\mathbf{X}}_{h,k-1} + \tau_k \bar{\dot{\mathbf{X}}}_{h,k-1} + \frac{1}{2}\tau_k^2 \bar{\ddot{\mathbf{X}}}_{h,k-1}) \\ \hat{\dot{\mathbf{X}}}_{h,k} = R_k^{-1}(\bar{\dot{\mathbf{X}}}_{h,k-1} + \tau_k \bar{\ddot{\mathbf{X}}}_{h,k-1}) \\ \hat{\ddot{\mathbf{X}}}_{h,k} = R_k^{-1}(\bar{\ddot{\mathbf{X}}}_{h,k-1}) \end{cases},$$

where τ_k denotes the time interval between frames $k-1$ and k. Since the intrinsic camera parameters are assumed to be constant, the state transition model follows system $\boldsymbol{\mathcal{I}}$. The rotation of the camera is modeled as a constant angular velocity rotation. Thus the rotation angles remain the same as the previous frame as shown in system $\boldsymbol{\mathcal{R}}$. System $\boldsymbol{\mathcal{B}}$ describes body motion, where we assume the movement of bird body follows a constant acceleration in $\{W\}$, with piecewise constant acceleration increment error. This model is sufficiently broad to cover constant velocity, constant acceleration, and smooth acceleration changing scenarios. The transition from frame $k-1$ to frame k is described as a rigid body motion.

Recall that R_k denotes camera rotation from $\{C_{k-1}\}$ to $\{C_k\}$, and is a function of the extrinsic parameters:

$$\begin{aligned} R_k &= R_k(\bar{\alpha}_{k-1}, \bar{\beta}_{k-1}, \bar{\gamma}_{k-1}) \\ &= \begin{bmatrix} s_\alpha s_\beta s_\gamma + c_\alpha c_\beta & s_\alpha s_\beta c_\gamma - s_\alpha c_\beta & c_\alpha s_\beta \\ c_\alpha s_\gamma & c_\alpha c_\gamma & -s_\alpha \\ s_\alpha c_\beta s_\gamma - c_\alpha s_\beta & s_\alpha c_\beta c_\gamma + s_\beta s_\gamma & c_\alpha c_\beta \end{bmatrix}, \end{aligned}$$

where $c_\alpha := \cos(\bar{\alpha}_{k-1})$ and $s_\alpha := \sin(\bar{\alpha}_{k-1})$, and the same notation convention is used for angles β and γ.

The Jacobian G_k^B in (4.3) is therefore obtained by taking partial derivative on $\boldsymbol{\mathcal{I}}$, $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{B}}$.

However, with a moving object and a moving camera, there exists motion ambiguity when both movements are unknown. To deal with the ambiguity, two methods are designed to estimate camera rotation.

4.5.1.1 Rotation estimation with background correspondences

By definition, background is static in $\{W\}$. SIFT features and their correspondences [24] can be extracted from background in frames $k-1$ and k (Box B8 in Figure 4.1). From projective geometry, we know that corresponding background points in consecutive frames conform to a same homography due to the shared camera center in a pure rotation. If the background is feature-rich, it is possible to estimate the camera rotation from background features separately (Box B6 in Figure 4.1). Thus the bird motion in camera frame can be tracked without ambiguity.

For a background point, the relationship between its coordinate in $\{C_{k-1}\}$ and in $\{C_k\}$ follows: $\mathbf{X}_{k-1} = R_k\mathbf{X}_k$. Since the intrinsic camera matrix remains unchanged during the video period, $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ satisfy the following:

$$\eta\mathbf{x}_{k-1} = KR_kK^{-1}\mathbf{x}_k = \mathcal{H}_k\mathbf{x}_k,$$

where η is a scalar and $\mathcal{H}_k = KR_kK^{-1}$ is the homography matrix, $K = \bar{K}_{k-1}$ is the estimated intrinsic matrix.

The minimum solution for homography estimation requires four pairs of corresponding key points. If the minimum requirement is satisfied, the optimal homography $\mathcal{H}_k^*$ can be calculated following the RANSAC-based algorithm in [129]. The rotation matrix R_k^* is computed by decomposing $\mathcal{H}_k^*$,

$$R_k^* = K^{-1}\mathcal{H}_k^*K, \tag{4.11}$$

and the rotation angles are obtained by decomposing R_k^*,

$$\begin{cases} \alpha_k^* = \arctan(-R_{23,k}^*/\sqrt{(R_{13,k}^*)^2 + (R_{33,k}^*)^2}) \\ \beta_k^* = \arctan(R_{13,k}^*/R_{33,k}^*) \\ \gamma_k^* = \arctan(R_{21,k}^*/R_{22,k}^*) \end{cases}, \tag{4.12}$$

where R_{ij}^* denotes the (i, j)-th entry of R^*. The output can be used to replace the initial prediction in $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{B}}$. If there are fewer than four pairs of SIFT points, the rotation angles estimation sticks to the initial prediction in $\boldsymbol{\mathcal{R}}$.

4.5.1.2 Rotation estimation with foreground motion constraint

The aforementioned homography estimation will fail when the background is a featureless sky. An alternative solution is proposed to estimate camera rotation using bird flying constraints (Box B7 in Figure 4.1).

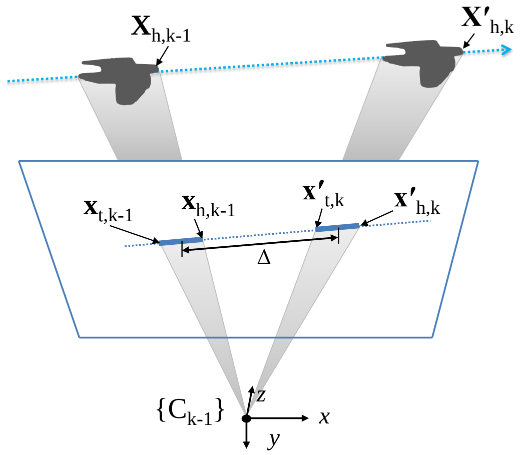


Figure 4.2: Illustration of foreground motion constraints.

Let $\mathbf{X}'_k$ and x'_k be the projection of $\mathbf{X}_k$ and x_k to $\{C_{k-1}\}$, respectively. Recall that the transformation from $\{C_{k-1}\}$ to $\{C_k\}$ follows a rotation matrix R_k. We have

$$\mathbf{X}'_k = R_k \mathbf{X}_k, \tag{4.13}$$

$$s\mathbf{x}'_k = K R_k K^{-1} \mathbf{x}_k. \tag{4.14}$$

Intuitively, x'_k describes the expected position of the point if there is no camera rotation between frames $k-1$ and k. Therefore, the difference between x'_k and x_k is the result of the camera rotation (Figure 4.2).

Linear trajectory constraint: For a flying bird, we know that its trajectory in a short duration is approximately linear along the direction of its body axis. This is reasonable because a smooth trajectory can always be approximated by piece-wise linear segments, due to the fast sampling rate.

Therefore, in $\{C_{k-1}\}$, the coordinates $\mathbf{x}_{h,k-1}$, $\mathbf{x}_{t,k-1}$, $\mathbf{x}'_{h,k}$, and $\mathbf{x}'_{t,k}$ should be approximately collinear in frame $k-1$. Note that we augment subscript by adding h and t to indicate the head and the tail of the bird, respectively.

Denote an image line as $\mathbf{l} = [a, b, c]^\mathsf{T}$. The residual error of the linear fitting is defined by a function $\Gamma_1(\mathbf{l})$ as follows:

$$\Gamma_1(\mathbf{l}) = \mathbf{l}^\mathsf{T}[\mathbf{x}_{h,k-1}, \mathbf{x}_{t,k-1}, \mathbf{x}'_{h,k}, \mathbf{x}'_{t,k}]. \tag{4.15}$$

The optimal line that fits the four body points can be computed by

$$\mathbf{l}^* = \arg\min_{\mathbf{l}} \|\Gamma_1(\mathbf{l})\|_2, \tag{4.16}$$

$$\text{s.t. } a^2 + b^2 = 1. \tag{4.17}$$

and the residual error to the line fitting is

$$\Gamma_1^* = \Gamma_1(\mathbf{l}^*). \tag{4.18}$$

Constant velocity constraint: Denote Δ to be the expected pixel displacement of the center of bird body axis in $\{C_{k-1}\}$. Given the 3D velocity $\dot{\bar{\mathbf{X}}}_{h,k-1}$ of the bird in frame $k-1$, Δ can be approximated by

$$\Delta \approx \begin{bmatrix} f_{k-1} & 0 & u_{0,k-1} \\ 0 & f_{k-1} & v_{0,k-1} \end{bmatrix} (\tau_k \dot{\bar{\mathbf{X}}}_{h,k-1})/z, \tag{4.19}$$

where

$$z = \frac{\left\| \begin{bmatrix} f_{k-1} & 0 & u_{0,k-1} \\ 0 & f_{k-1} & v_{0,k-1} \end{bmatrix} L \frac{\bar{\dot{\mathbf{x}}}_{h,k-1}}{\|\bar{\dot{\mathbf{x}}}_{h,k-1}\|} \right\|}{\|\mathbf{x}_{h,k-1} - \mathbf{x}_{t,k-1}\|}, \tag{4.20}$$

is the approximated depth of the bird in $\{C_{k-1}\}$.

Projecting $\mathbf{x}_{h,k}$ and $\mathbf{x}_{t,k}$ to $\{C_{k-1}\}$ by (4.14), the residual error between the approximated Δ and the measured displacement can be computed by the following function

$$\Gamma_2 = \|\Delta - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \frac{(\mathbf{x}'_{h,k} + \mathbf{x}'_{t,k}) - (\mathbf{x}_{h,k-1} + \mathbf{x}_{t,k-1})}{2})\|. \tag{4.21}$$

Residual errors from (4.18) and (4.21) compose a 5-dimension residual error vector

$$\Gamma = [\Gamma_1^*, \Gamma_2]^{\mathsf{T}}, \tag{4.22}$$

where each dimension describes a pixel-level error.

Therefore, the optimal rotation matrix is estimated by minimizing the L2 norm of the entire residual vector

$$R_k^* = \arg\min_{R_k} \|\Gamma\|_2. \tag{4.23}$$

The estimated values of rotation angles are calculated by (4.12), and are used to replace the values in $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{B}}$.

4.5.2 Model Fusion for Body Tracking

Now, let us fuse the two rotation estimating methods (Box B5 in Figure 4.1) using the IMM-EKF framework in (4.7-4.9). The fusion is consistent with the ratio of their residual in predicting states. We design the model transition probability as a uniform distribution, the model fusion follows the likelihood-based weighting in [122], where the likelihood is calculated as follows:

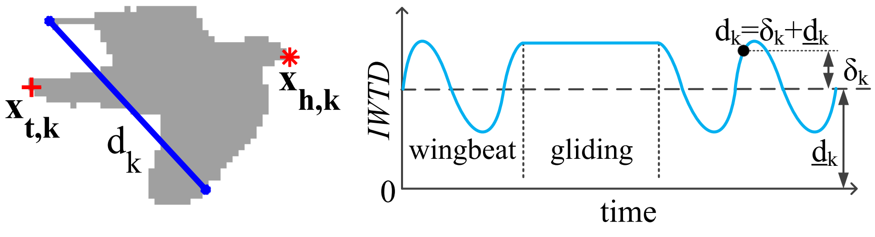
$$
\begin{aligned}
\Lambda_k^{(j)} &= \frac{1}{\sqrt{2\pi|S_k^{(j)}|}} e^{-\frac{1}{2}(\mathbf{z}_k^B - h(\hat{\boldsymbol{\mu}}_k^{(j)}))^\top (S_k^{(j)})^{-1} (\mathbf{z}_k^B - h(\hat{\boldsymbol{\mu}}_k^{(j)}))}, \\
w_k^{B,(j)} &= \frac{w_{k-1}^{B,(j)} \Lambda_k^j}{\Sigma_i w_{k-1}^{B,(i)} \Lambda_k^i},
\end{aligned}
$$

where $j = 1$ represents rotation estimation using background points, while $j = 2$ represents using foreground motion constraint.

4.6 Wing Motion Model

The wing motion model captures the periodic wing flapping motion as well as the gliding motion of the bird. As shown in previous section, the wing motion is characterized by IWTD. Denote d to represent the IWTD in ICS. When the bird is gliding, d should remain constant in a short duration. When the bird is flapping its wings, d changes in a periodic pattern with a constant baseline length. As shown in Figure 4.3b, $d_k = \delta_k + \underline{d}_k$, where δ_k is the periodically changing part of d_k, and $\underline{d}_k$ represents the baseline length. The state for wing motion tracking is designed to include δ and $\dot{\delta}$, the angular wingbeat frequency (WF) ω, and the baseline length $\underline{d}$. Here, we superscribe a W to indicate variables in the wing motion model.

$$\boldsymbol{\mu}^W = [\delta, \dot{\delta}, \omega, \underline{d}]^\top. \tag{4.24}$$



(a) (b)

Figure 4.3: (a) An example of the observations on segmentation result in frame k, where the gray region is the foreground. The blue solid line shows the extracted IWTD d_k, The red asterisk shows the extracted head position $\mathbf{x}_{h,k}$ and the red plus shows the tail position $\mathbf{x}_{t,k}$. (b) The state variables and two models for wing motion.

4.6.1 State Transition Model for Wing Motion

The wing motion model includes both wingbeat and gliding motion.

4.6.1.1 Wingbeat model

During wingbeat periods, d changes in a periodic pattern. Liu et al. [112] show that the wingbeat signal is approximately a sine wave. Therefore, the dynamics of d during wingbeat is approximated by a sine function and its system model (Box W1 in Figure 4.1) is system $\mathcal{W}$ below where the second order Taylor expansion is used for frequency tracking.

$$\mathcal{W}: \begin{cases} \hat{\delta}_k^{(1)} = \bar{\delta}_{k-1} + \tau_k \bar{\dot{\delta}}_{k-1} - \frac{1}{2}\tau_k^2 \bar{\omega}_{k-1}^2 \bar{\delta}_{k-1} \\ \hat{\dot{\delta}}_k^{(1)} = \bar{\dot{\delta}}_{k-1} - \tau_k \bar{\omega}_{k-1}^2 \bar{\delta}_{k-1} - \frac{1}{2}\tau_k^2 \bar{\omega}_{k-1}^2 \bar{\dot{\delta}}_{k-1} \\ \hat{\omega}_k^{(1)} = \bar{\omega}_{k-1} \\ \hat{\underline{d}}_k^{(1)} = \bar{\underline{d}}_{k-1} \end{cases} . \tag{4.25}$$

4.6.1.2 Gliding model

In gliding periods, d is modeled by a constant value, with smooth piecewise incremental errors. Thus, each dimension of the state vector is constant. The dynamic of the system (Box W3 in Figure 4.1) is

$$\boldsymbol{\mathcal{G}} : \begin{cases} \hat{\delta}_k^{(2)} = \bar{\delta}_{k-1} \\ \hat{\dot{\delta}}_k^{(2)} = \bar{\dot{\delta}}_{k-1} \\ \hat{\omega}_k^{(2)} = \bar{\omega}_{k-1} \\ \hat{\underline{d}}_k^{(2)} = \bar{\underline{d}}_{k-1} \end{cases} . \tag{4.26}$$

Note that superscripts (1) and (2) used in systems $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{G}}$ are system index (i.e. value of j in IMM framework (4.2)).

4.6.2 Model Fusion for Wing Motion

When the bird is gliding, d should be approximately constant with a small measurement variance. We assume a short sequence of d in gliding mode conforms to i.i.d. Normal distribution with variance σ^2. Given a sample sequence d_k to d_{k+n}, the sample variance, denoted as s_k^2, should conform to a Chi-square distribution of n degrees of freedom $\frac{ns_k^2}{\sigma^2} \sim \chi_n^2$ (Ch. 2 in [130]).

When calculating the model fusion weight (Box W5 in Figure 4.1), a one-sided χ^2 test is conducted on each frame, and accepts the motion as gliding if the corresponding $\frac{ns_k^2}{\sigma^2}$ is within the left 95% confidence interval. Otherwise, the motion is more likely to be wingbeat. To ensure the smoothness of tracking in both models, the model fusion weight is given by a sigmoid function, rather than a binary value. Denote the 95% threshold for

n-degree χ^2 test as $\varepsilon_{n,95}$, the weight $w^{W,(1)}$ for wingbeat model is assigned as

$$w_k^{W,(1)} = 1/(1+e^{-(\frac{ns_k^2}{\sigma^2}-\varepsilon_{n,95})}), \tag{4.27}$$

$$n = \lceil \frac{1}{2}\frac{r}{(F_{min}+F_{max})/2} \rceil, \tag{4.28}$$

where r is the frame rate of the video, and n is selected to cover at least a half period of a wingbeat cycle. The weight for the gliding model is $w^{W,(2)} = 1 - w^{W,(1)}$.

4.7 Observation Model and Adaptive Sample Rate

4.7.1 Observation Model

Images provide observations of bird body axis and IWTD. At frame k, the observation vector for body motion and wing motion are

$$\mathbf{z}_k^B = [\mathbf{x}_{h,k}^{\mathsf{T}}, \mathbf{x}_{t,k}^{\mathsf{T}}]^{\mathsf{T}}, \tag{4.29}$$

$$\mathbf{z}_k^W = d_k, \tag{4.30}$$

where $\mathbf{x}_{h,k}$ and $\mathbf{x}_{t,k}$ describe the bird head and tail tip position in frame k (Figure 4.3a), and d_k is simply the measured IWTD in frame k.

According to [103], $\mathbf{x}_{t,k}$ can be expressed by a function of $\mathbf{x}_{h,k}$, $\dot{\mathbf{X}}_k$ and the bird body length L. The observation models are as below.

$$\mathbf{z}_k^B = \begin{bmatrix} \frac{[f_k,0,u_{0,k}]\mathbf{X}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}} \\ \frac{[0,f_k,v_{0,k}]\mathbf{X}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}} \\ \frac{[f_k,0,u_{0,k}](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)}{[0,0,1](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)} \\ \frac{[0,f_k,v_{0,k}](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)}{[0,0,1](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)} \end{bmatrix}, \tag{4.31}$$

$$\mathbf{z}_k^W = \delta_k + \underline{d}_k. \tag{4.32}$$

4.7.2 Adaptive Sample Rate

For far field body tracking, small noise in observation may lead to large error in state estimation, if the relative motion between frames is too small. To control the signal-noise ratio under reasonable range, the body observation sequence is down-sampled by an adaptive sample rate before tracking. Given the bird flying velocity $\dot{\mathbf{X}}_{h,k}$, its position $\mathbf{X}_{h,k}$ and a time interval τ_{k+1}, the relative pixel displacement of the bird body can be approximated:

$$\Delta_k = \sqrt{(\frac{[f_k, 0, u_{0,k}]\tau_{k+1}\dot{\mathbf{X}}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}})^2 + (\frac{[0, f_k, v_{0,k}]\tau_{k+1}\dot{\mathbf{X}}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}})^2} \tag{4.33}$$

Assume the measurement noise along u and v axis is i.i.d $N(0, \sigma_z)$. To ensure the noise-to-signal ratio is approximately below a threshold ξ, the time interval τ_{k+1} for frame k is chosen such that $\sigma_z/\Delta_k \leq \xi$, and the observation sequence is down-sampled. In our algorithm, $\xi = 0.05$ is selected.

4.8 Initialization and Species Prediction

4.8.1 Initialization

The body and wing motion models are initialized separately as below.

4.8.1.1 Body and camera

The first few frames are used for initializing the state vector. Follow our assumption that the bird trajectory is linear, velocity and camera intrinsic parameters are constant in a short time period. The acceleration is initialized to $[0, 0, 0]$, and the initial camera rotation angles are 0s. For a frame k, the 3D information of bird head $\mathbf{X}_{h,k}$ and $\dot{\mathbf{X}}_{h,k}$ can be expressed by a translation and rotation of $\mathbf{X}_{h,1}$ in frame 1:

$$\mathbf{X}_{h,k} = R_{1:k}(\mathbf{X}_{h,1} + \tau_{1:k}\dot{\mathbf{X}}_{h,1}) \tag{4.34}$$

$$\dot{\mathbf{X}}_{h,k} = R_{1:k}\dot{\mathbf{X}}_{h,1} \tag{4.35}$$

where $R_{1:k}$ denotes the rotation from frame 1 to k, and $\tau_{1:k}$ denotes the time interval from frame 1 to k. Therefore, the projection of the bird in image k is:

$$\begin{bmatrix} \mathbf{x}_{h,k} \\ \mathbf{x}_{t,k} \end{bmatrix} = \begin{bmatrix} \frac{[f,0,u_0]\mathbf{X}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}} \\ \frac{[0,f,v_0]\mathbf{X}_{h,k}}{[0,0,1]\mathbf{X}_{h,k}} \\ \frac{[f,0,u_0](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)}{[0,0,1](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)} \\ \frac{[0,f,v_0](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)}{[0,0,1](\mathbf{X}_{h,k}-L\dot{\mathbf{X}}_{h,k}/\|\dot{\mathbf{X}}_{h,k}\|)} \end{bmatrix} \tag{4.36}$$

There are 3 unknowns for $\mathbf{X}_{h,1}$, 3 unknowns for $\dot{\mathbf{X}}_{h,1}$, 3 unknowns for the camera intrinsic parameters. Each frame $k(> 1)$ introduces 3 unknowns for $R_{1:k}$ by the Euler angle representation. Using n frames, the total numbers of unknowns is $9 + 3(n - 1)$, while the number of equations is $4n$. The minimum number of frames needed to provide a solution to initial state is 6. For more stable performance, the first 7 frames are used for initializing camera and bird body state variables, the solution is achieved by non-linear optimization. The computed initial values are applied to each model in body motion tracking.

4.8.1.2 Wing

The initialization of wing state variables are based on the observations and the species information of the candidate bird.

$$\begin{cases} \delta_1 & = d_1 - \underline{d}_1 \\ \dot{\delta}_1 & = (d_2 - d_1)/\tau_2 \\ \omega_1 & = 2\pi(F_{max} + F_{min})/2 \\ \underline{d}_1 & = \frac{1}{N}\Sigma_{i=1}^{N} d_i \end{cases} \tag{4.37}$$

where N is the number of frames that covers a wingbeat period. The initial values are applied to each model in wing motion tracking.

4.8.2 Species Prediction

The BODY model tracks the flying velocity of the bird, while the WING model tracks the wingbeat frequency. The species prediction is based on the tracking results from both the BODY and WING models. The bird is considered to belong to the input candidate species if both models converge, and the converged velocity and wingbeat frequency fall in the given range:

$$|\|\bar{\dot{\mathbf{X}}}_{h,N_h}\| - \|\bar{\dot{\mathbf{X}}}_{h,N_h-1}\|| \leq \epsilon_h, \tag{4.38a}$$

$$\|\bar{\dot{\mathbf{X}}}_{h,N_h}\| \in [V_{min}, V_{max}], \tag{4.38b}$$

$$|\bar{\omega}_{N_\omega} - \bar{\omega}_{N_\omega-1}| \leq \epsilon_\omega, \tag{4.38c}$$

$$\frac{\bar{\omega}_{N_\omega}}{2\pi} \in [F_{min}, F_{max}], \tag{4.38d}$$

where N_h and N_ω are the total number of frames in BODY and WING tracking, respectively, ϵ_h and ϵ_ω are the convergence thresholds.

4.9 Experiments

The proposed tracking and recognition algorithm is implemented and tested on both simulated data and the real data from field experiments. The computer used in the tests is a desktop PC. The algorithm is implemented in Matlab.

4.9.1 Simulation on Synthetic Data

Since the objects are free flying birds in our study, the ground truth of real data is difficult to acquire. Simulated inputs allow us to numerically evaluate the performance of 3D tracking under full range of possible parameter settings.

4.9.1.1 Data generation

The simulated camera resolution is 640×480, frame rate is 100 frames/sec, focal length is 5333, and the principal point is $[-2, 1]$. The camera angle of view is set to $10°$. The length of the sequence is set to 3 seconds.

The candidate species includes two popular species in south Texas region, and their information is showed in Table 4.1. The two species represents small-size and mid-size birds, with disjoint WF range and large flying speed coverage. The species information is obtained from [103, 107]. For the herring gull, the WF is set to cover 3σ (99.7%) in the given distribution in [107].

Table 4.1: Candidate species for simulation

species	L (cm)	$[V_{min}, V_{max}]$ (m/s)	$[F_{min}, F_{max}]$ (Hz)
Rock Pigeon	33	$[6.67, 15.67]$	$[5, 8]$
Herring Gull	60	$[8.94, 17.88]$	$[2.66, 3.44]$

To generate a sequence of synthetic data, a body length, a flying speed and a wingbeat frequency are first randomly generated. The initial depth is a random number between $[20, 40]$ meters. The initial 3D position and flying direction of the bird in $\{W\}$ are randomly generated within the camera field of view. Acceleration is set to $[0, 0, 0]^\mathsf{T}$. Random wingbeat periods are generated. The camera rotation in the first frame is $[0, 0, 0]$. We mimic the real bird tracking videos by generating random rotation angles that make the bird appear around the center of every frame. The observations are then acquired by projecting 3D points to images. Moreover, background keypoint pairs are randomly generated for random selected frames.

4.9.1.2 Tracking performance

The convergence of tracked velocity and wingbeat frequency is important to recognition. Given a sequence of simulated data and its true species information, we test the accuracy of the tracked velocity and WF at noise std equals to 2 pixels. Table 4.2 shows the mean relative errors of the converged values w.r.t. the ground truth. Simulated results show that both models are able to converge to near true values given the correct species information.

Table 4.2: Relative tracking errors under noise std = 2px

	$\|\hat{\dot{X}}_{h,N_h}\|$	$\hat{\omega}_{N_\omega}/2\pi$	$\hat{f}_{N_h}$
mean relative error	0.09	0.01	0.04

4.9.1.3 Recognition performance

We test the species recognition performance under 2 pixel of segmentation error. A data set of 100 rock pigeon trails, 100 herring gull trails and 100 trails of birds in other species are generated. To evaluate the recognition performance in a large species range, the 100 trails of other species are generated using randomly selected the bird body length within $[0.1, 0.9]$ meters, the bird flying speed within $[4, 24]$ m/s and the bird WF within $[2, 10]$Hz for each trail. Rock pigeon and herring gull are used as candidate species to test the generated data set.

The proposed method uses both flying speed and WF as species prediction criteria. In comparison to [103] and Section 3, the species prediction using flying speed (4.38a) and (4.38b) alone, and using WF (4.38c) and (4.38d) alone are also tested. Figure 4.4 demonstrates how the false positive rate (FP) and false negative rate (FN) for each candidate species change according to ϵ_h and ϵ_ω.

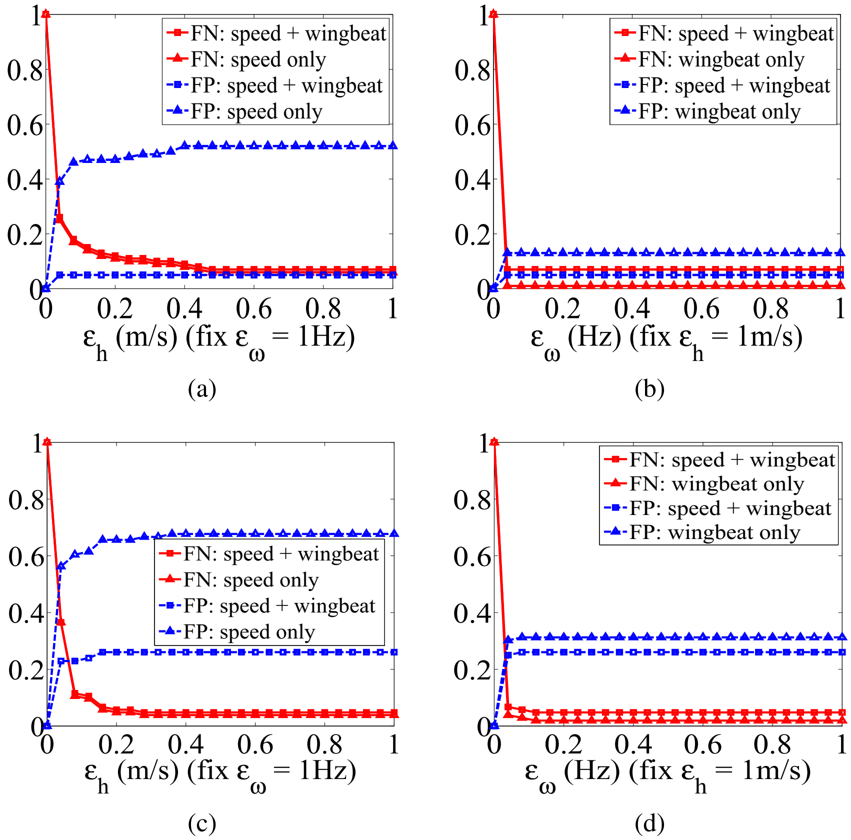


Figure 4.4: FP and FN rates w.r.t convergence threshold. The upper figures show the recognition results for Herring Gull, and the lower figures show that for Rock Pigeon.

For the proposed method, the FP and FN rate for pigeon is 23% and 4.8%, respectively. The FP and FN rate for herring gull is 5% and 7%, respectively. The FN rate is similar, however, the FP rate is significantly improved comparing to methods that use flying speed or WF alone as species prediction criteria. The FP rate for rock pigeon is a little higher than that of herring gull, because the rock pigeon has larger WF range.

Figure 4.5 shows an example of bird body motion tracking of a rock pigeon. The upper figure shows the number of background features in each frame. The lower figure shows weights assigned to the background based model. The weights for the background based model are not necessarily high, when background features exist. However, background features often raise the weight when the rotation estimation accuracy from background correspondence outperforms that from foreground constraints w.r.t the observations.

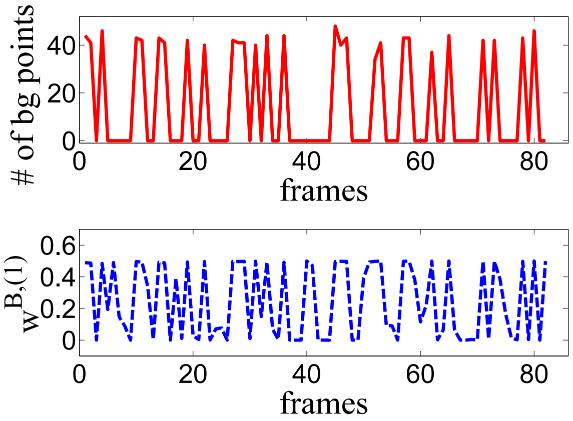


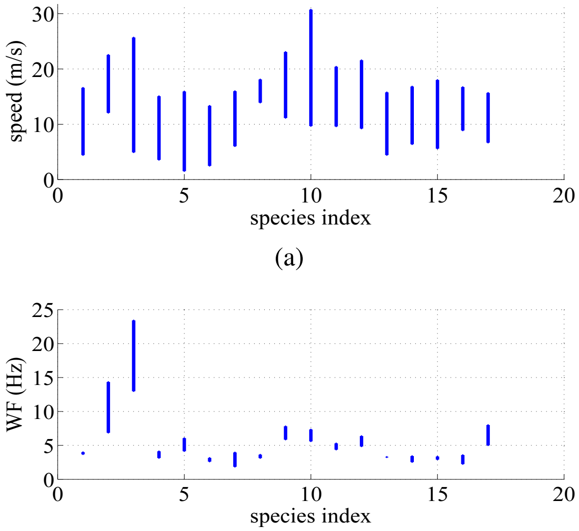
Figure 4.5: An example of bird body motion tracking. The upper figure shows the number of background correspondences in red solid line. The lower figure shows the weights to the background based model during the tracking.

4.9.2 Experiments on Real Videos

Testing on simulated data proves the convergence and accuracy of the proposed system. We also conduct physical experiment on a small set of real flying bird videos. Since there is no existing benchmark for the evaluation of bird species recognition under our problem settings, the testing videos are manually collected on TAMU campus and YouTube. The camera to capture videos on TAMU campus is a Casio EX-ZR200, which runs at 120 frames per second with a 640×480 pixel resolution. The ground truth species is obtained by using human inputs on each video.

The testing dataset contains 11 sequences of flying birds, including Rock Pigeons, Fish Crows, Magnificent Frigatebirds, Golden Eagles and Ducks. The total number of frames is 2188. The collected videos are tested with 17 candidate species provided in [131] and [103], resulting in 187 prediction results. Figure 4.6a and 4.6b show the speed and WF range of each candidate species. The candidate species covers 0.14 to 1.6 meters in body length, 1.6 to 30.7 m/s in flying speed, and 2.8 to 23.4 Hz in WF.

Figure 4.7 shows an example of wing motion tracking of a Magnificent Frigatebird. The bird starts flying with wingbeat motion and the IWTD shows its periodicity. The bird switches to gliding motion at around frame 160 and the IWTD sequence becomes a slowly changing curve. At the end of the sequence, the bird flaps wings again and the IWTD begins to change. The lower figure in Figure 4.7 shows the model fusion weights successfully distinguish the two types of wing motion. The wingbeat weights are high with occasional drops at the beginning and the end of the sequence, and become low in the gliding period. The weight drops in wingbeat motion mainly occur at the extreme values of the sine wave, where the sample covariance is relatively small. This weight distribution clearly shows that the algorithm successfully recognizes the gliding and the wingbeat motion patterns, and correctly adjusts trust levels on the corresponding models.



(a)

Figure 4.6: The speed and WF ranges of each candidate species. (a) shows the speed ranges, and (b) shows the WF ranges. The x-axis is the index of species.

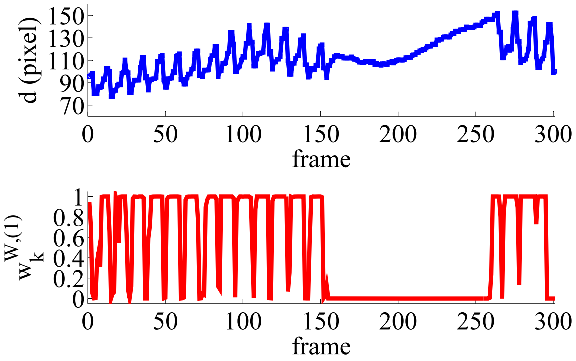
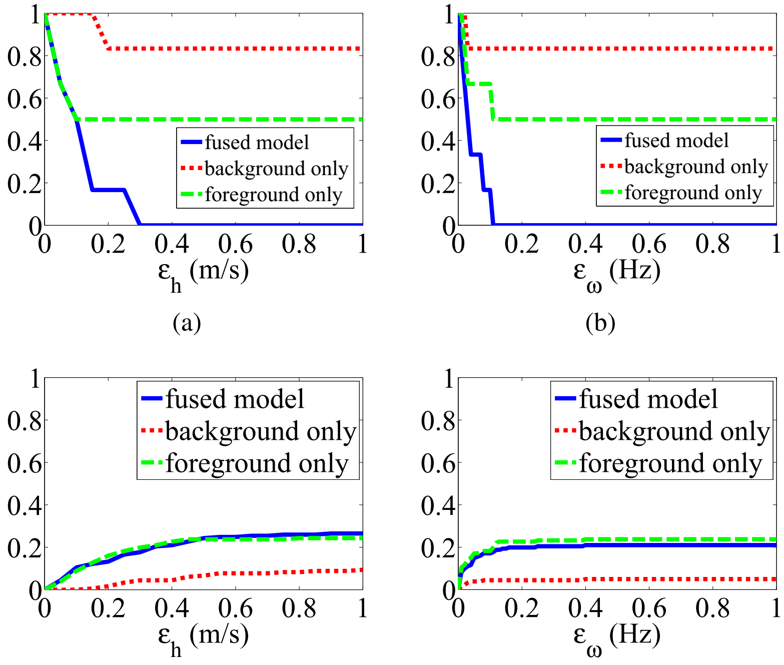


Figure 4.7: An example of wing motion tracking. The upper figure shows the observed d in blue solid line. The lower figure shows the weights to the wingbeat model during the tracking.

In the body motion model, we proposed two methods to estimate camera rotation using separated background and foreground images, and resulting state transitions are fused. The two rotation estimation methods work mutually on the tracking of body motion, and the weights to each method depend on the their accuracy and consistency w.r.t. the observations. To evaluate this hybrid approach, we compare the species recognition performance using fused method with using only one camera rotation estimation method. Figure 4.8 shows the FN and FP rates using different camera rotation estimation methods. The solid lines show the rates using the fused model, while the dotted/dashed lines show the rates using only background/foreground information for rotation estimation. In the results, the fused model shows better trade-off between low FN rate and acceptable FP rate. The single method models have better or equal performance in FP rates, however results in significantly high FN rates.



(a) (b)

(c) (d)

Figure 4.8: FP and FN rates w.r.t convergence threshold using different camera rotation estimation methods. (a) and (b) show the FN rates, while (c) and (d) show the FP rates. Solid lines show the rates using fused model, dotted lines show the rates using background only for camera rotation estimation, and the dashed lines show the rates using foreground constraints only.

The tracked flying speed from BODY and WF from WING form a combined signature for species prediction. To evaluate the prediction performance, FN and FP rates are shown in Figure 4.9a and 4.9b w.r.t. different convergence thresholds. Comparisons are also conducted with the species prediction results using speed or WF only. Results show that the combined signatures from body and wing motions outperform either single motion model by significantly improving FP rates while maintaining low FN rates, which is consistent with the simulation result. Setting the thresholds $\epsilon_h = 0.3 m/s$, $\epsilon_\omega = 0.11 Hz$, the FN rate reaches 0 and the FP rate reaches 0.18.

The ROC curve for the proposed system is presented in Figure 4.10. The area under the ROC curve is 92.86%, which is satisfying.

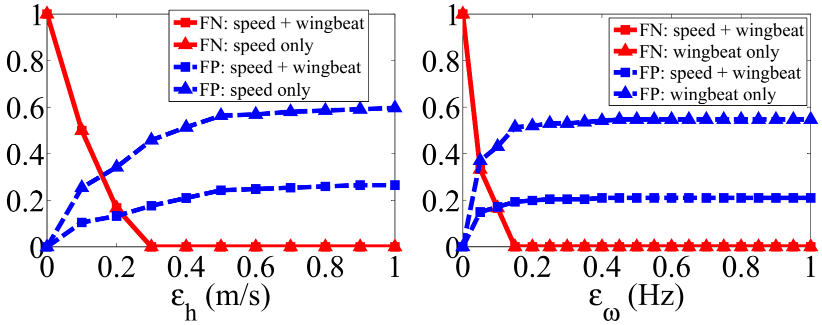


Figure 4.9: FP and FN rates w.r.t convergence threshold in physical experiment.

For the videos collected on TAMU campus, the mean relative error of estimated camera focal length is 3.31%.

4.10 Conclusion and Future Work

We reported an algorithm for bird species detection using videos captured by uncalibrated moving cameras. The algorithm tracks both body and wing motion of a flying bird to form signatures for species filtering. In the body model, we considered both cases when

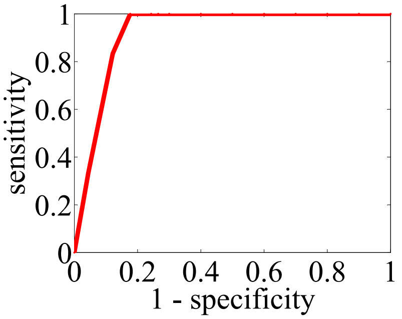


Figure 4.10: ROC curve for physical experiments.

background motion introduced by the camera can and cannot be directly recognized using background key point matching. We were also able to recover intrinsic camera parameters in the body motion tracking. In the wing model, we considered both periodic wing flapping and gliding motion patterns. These models were combined to form a multi-model framework. We tested the algorithm and compared its performance with single model approaches in physical experiments. Results showed that the new algorithm significantly reduced the FP rate while maintaining low a FN rate.

In the future, we plan to further augment camera intrinsic parameter model by incorporating lens distortion estimation and tracking of changing focal length.

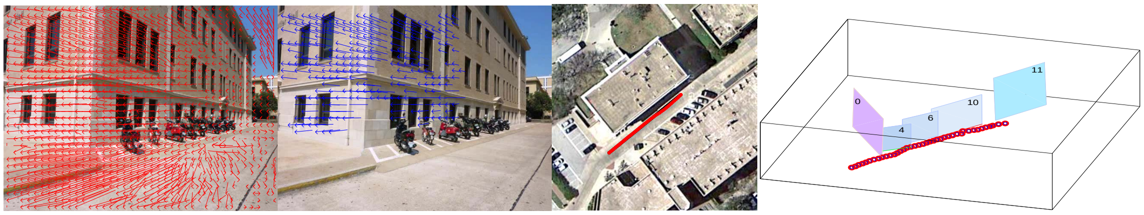
5. TOWARD FEATURELESS VISUAL NAVIGATION: SIMULTANEOUS LOCALIZATION AND PLANAR SURFACE EXTRACTION USING MOTION VECTORS IN VIDEO STREAMS

5.1 Introduction

Beside the visual tracking and recognition problem, motion signature plays an important role in vision-based robot navigation problems. In this section, we focus on the visual navigation problem, and design new algorithms with analysis of a new motion signature.

Many visual navigation approaches rely on correspondence of features between individual images to establish geometric understandings of image data. To do that, image data are often first reduced to a feature set such as points. Then extensive statistical approaches such as random sample consensus (RANSAC) are employed to search for feature matches that satisfy the expected geometry relationships. Such geometric relationships enable us to derive robot/camera ego-motion estimation or scene understandings in different applications such as visual odometry or simultaneous localization and mapping (SLAM) [53]. The inherent drawback of these approaches is the expensive computation load and robustness of feature extraction, which is often hindered by varying lighting conditions and occlusions. Contrarily, human biological visual processing does not follow such an elaborated process. Humans are very aware of changes from frame to frame and rarely process data as feature sets. The natural representation for changes between adjacent frames is optical flow. However, optical flow is notoriously expensive in computation requirement, which prohibits real time applications.

On the other hand, recent streaming videos are transmitted after complex compression. To reduce bandwidth needs, the image frame is equally gridded into blocks, name macroblocks (MBs). The compression algorithms exploit similarities between MBs in ad-



(a) (b) (c) (d)

Figure 5.1: (a) Original MVs represented by red arrows. (b) Filtered MVs represented by blue arrows. (c) Satellite image of an experiment site in Google Maps. The black line is manually measured ground truth camera trajectory, and the red line is the estimated trajectory. (d) Estimated plane positions and camera trajectory.

jacent frame sets, characterized as motion vectors (MVs)(Figure 5.1a). Therefore, one or more MVs will be used to describe the motion of a MB, and the dense MVs reveal the motion of the entire scene. Compared with optical flows, MVs have lower spatial resolution (per block vs. per pixel) but higher temporal resolution because MVs are extracted from multiple frames instead of mere two adjacent frames. MVs carry the correspondence information and are readily available from the encoded video data. These inspire us to explore MVs as a new motion signature for visual navigation problems.

Despite all the aforementioned advantages, MVs are not easy to use because of their low spatial resolution and relatively high noise. Here we explore how to use MVs for simultaneous localization and planar surface extraction (SLAPSE) for a mobile robot equipped with a single camera. We establish the MV noise models to capture the observation error. We formulate the SLAPSE problem and study how to extract planes from MVs using planar homography filtering. We then develop an extended Kalman filter (EKF) based approach with planes and robot motion as state variables. We have implemented our algorithm using C/C++ on a PC platform and tested the algorithm in physical experiments. The results show that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.

5.2 Related Work

SLAPSE relates to recent progress in visual navigation for mobile robots, MPEG compression, and dense 3D reconstruction.

SLAPSE can be viewed as visual SLAM with special observation inputs. In a regular SLAM framework, the physical world is represented by a collection of landmarks which are primarily features observed from images, such as key points [2, 55, 60, 67], line segments [68–70, 73–75], curves [72], and surfaces [76]. In these feature-based approaches, SLAM performance is largely dependent of feature distributions and correspondences. Building on these approaches, our SLAPSE takes advantage of the fact that MVs encode correspondences of segmented scene by overcoming the noise in the MV data.

Many efforts have been made to improve the accuracy and speed of MV computation in MPEG encoding. However, few studies have been conducted on utilizing MVs in complex vision problems. The main reason is because MVs are very noisy and have spatially low resolution. MVs have been applied in fast image-based camera rotation estimation [132], 2D object tracking [133], and image stabilization [134]. All of these approaches employ voting or averaging like strategies with region-based smoothing to obtain either foreground or background information separately. SLAPSE problems need to recover both the scene structure and the robot motion which require MVs with much less errors. We merge MVs across multiple adjacent frames to improve the signal to noise ratio, analyze errors on merged MVs, and utilize geometry relationship for better noise filtering.

MVs directly provide correspondences between pixel blocks. Once planes are identified through MVs in the SLAPSE problem, their corresponding pixel blocks are subsequently reconstructed in 3D. This is close to feature-based dense reconstruction, which usually requires precise dense correspondence between images. Recent dense reconstruction approaches start with a sparse set of salient points, and construct dense surfaces using photoconsistency and geometrical constraints [135]. More relevant works [136] utilize variational optical flow [19] to establish dense surface meshes from point clouds. These works inspire us to use MVs in scene mapping.

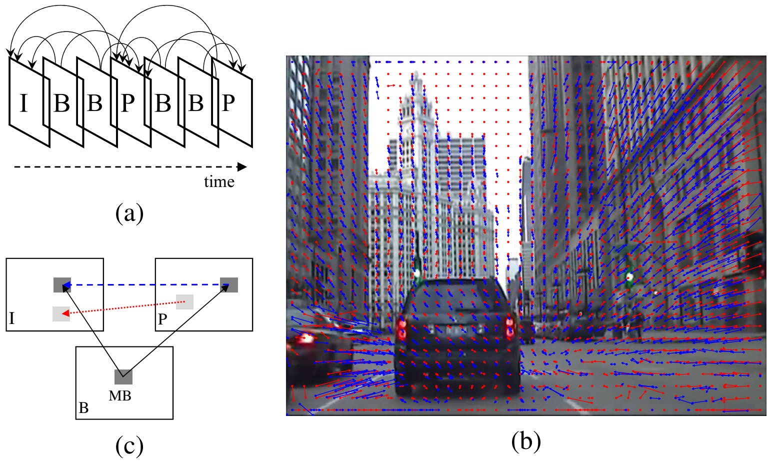
Our group focuses on developing monocular visual navigation techniques for energy and computation constrained robots. Using a vector-field approach [137], we develop a lightweight visual navigation algorithm for an autonomous motorcycle. We also address depth ambiguity problem through planning for small robot systems [138]. We have attempted different features for visual odometry such as vertical line segments [139, 140] and high level features [79, 141] to improve robustness. Through the process, we have learned shortcomings of feature-based approaches, which has motivated this work.

5.3 Background and Problem Definition

5.3.1 A Brief Introduction to Motion Vectors

Video encoders such as MPEG 1/2/4 often utilize block motion compensation (BMC) to achieve better data compression. BMC partitions each frame into small macroblocks (MB) (e.g. each MB is 16×16 pixels for MPEG 2). During encoding, block matching is employed to search for similar MBs in anchor frames. If a matching block is found, an MV is established. Note that each MV only represents a 2D shift in the image frame.

We use MPEG 2 as an example, and our analysis can be easily extended to other BMC-based encoding formats. There are often three types of frames (or slices of a frame): intra coded, predictive coded, and bidirectionally predictive coded, namely, I, P, and B frames, respectively. P and B frames consist of MBs defined by MVs pointing to their anchor frames. I and P frames are used as anchor frames for block matching. As illustrated in Figure 5.2a, a P frame is always predicted from the closest previous P or I frame and each MB has only one MV referring to the past. To achieve more compression, B frames utilize the closest P or I frames from both the past and the future as anchor frames. Each MB



(a)

(c)

(b)

Figure 5.2: (a) GOP structure for an MPEG 2 video stream. Note that the arrows on top of the frames refer to reference relationship in computing MVs. (b) Sample MVs overlaid on top of their video frame. Line segments and circles represent MVs and their pointing direction. (c) MVs between adjacent I and P frames can be obtained either directly (e.g. red dotted lines) or indirectly through B frames (e.g. blue dashed lines).

in B frame has up to two MVs point to both future and past anchor frames. The frame sequencing structure is referred to as group of pictures (GOP) in the MPEG protocols. In more advanced video format (e.g. MPEG 4), an MB can have as many as 16 MVs pointing to many reference frames.

5.3.2 Modeling Noise in Motion Vectors

If an MB centered at (u_i, v_i) in frame i finds the corresponding position (u_j, v_j) in the anchor frame j through block matching algorithm (BMA), then the resulting l-th MV can be defined as

$$m_l^{i \to j}(u_i, v_i) = \begin{bmatrix} \Delta_u \\ \Delta_v \end{bmatrix} = \begin{bmatrix} u_j - u_i \\ v_j - v_i \end{bmatrix}, \tag{5.1}$$

where u and v are frame coordinates. For simplicity, we sometimes use $m_l^{i \to j}$ to represent an MV between the two frames. An MB may contain many MVs. Some of them originate from the center of the MB and others may not (e.g. the reverse MV of $m_l^{i\to j}(u_i, v_i)$ is not necessarily located at the center of an MB in frame j).

Although containing image correspondence information, MVs are difficult to use due to noise introduced by BMA, which searches the most similar block in a given range. When video frames contain repetitive patterns, false matches can be generated. This is not a problem for video compression but presents a huge challenge to scene understandings. Sometimes, occlusions and scene changes may cause BMA to fail to find a matching. Say that BMA finds the correct matching with probability p, which is defined as event E_M. It is worth noting that p is also often affected by robot/camera moving speed. To avoid that, we can set frame rate proportional to the moving speed to reduce the variation in p. As observed from data, a regular street driving in urban area often has $p > 0.6$.

Even when a correct matching is found, BMA still has limited accuracy. MPEG 2 and 4 warrant 0.5 and 0.25 pixel accuracy, respectively. When the correct matching is found, this error $\boldsymbol{e}_l^{i\to j} = m_l^{i\to j} - \bar{m}_l^{i\to j}$ can be modeled as a 2D zero mean Gaussian

$$\boldsymbol{e}_l^{i\to j}|E_M \sim N(\mathbf{0}_{2\times 1}, \boldsymbol{\Sigma}), \tag{5.2}$$

where term $\cdot|E_M$ indicates that this is a conditional distribution, $\bar{m}_l^{i\to j}$ is the true mean of the MV, and covariance matrix $\boldsymbol{\Sigma} = diag\{\sigma^2, \sigma^2\}$ is a diagonal matrix. We set $\sigma = 0.25$ to conservatively capture the 0.5 pixel accuracy for MPEG 2. This accuracy level is sufficient for video presentation. However, due to the small time difference in adjacent frames, the motion parallax can be as small as 2-4 pixels, which leads to large relative error. Compounded with false matches, MVs are too noisy to be directly used for scene understanding.

5.3.3 Problem Formulation

To formulate SLAPSE problem, we assume that the intrinsic matrix of the camera is known as K through pre-calibration and the scene is dominated by planes, such as building facade and paved roads. Thus, the understanding of scene structure relies on estimating 3D planes.

Here all the 3D coordinate systems are right hand systems. Let us define

- $\{C_k\}$ as the 3D camera coordinate system (CCS) in frame k. For each CCS, its origin locates at the camera optical center, z-axis coincides with the optical axis and points to the forward direction of the camera, its x-axis and y-axis are parallel to the horizontal and vertical directions of the CCD sensor plane, respectively,
- R_k and $\boldsymbol{t}_k$ as the rotation and translation of $\{C_k\}$ w.r.t. frame $\{C_{k-1}\}$,
- $\boldsymbol{\pi}_{i,k} = [\boldsymbol{n}_{i,k}^{\mathsf{T}}, d_{i,k}]^{\mathsf{T}}$ is the i-th 3D plane in $\{C_k\}$, where $\boldsymbol{n}_{i,k}$ is the plane normal and $d_{i,k}$ is the plane depth, and
- $\tilde{\boldsymbol{\pi}}_{i,k} = \boldsymbol{n}_{i,k}/d_{i,k}$ as the inhomogeneous form of a plane.

Therefore, the problem is defined as below:

Definition 5. *Given the set of MVs up to time/frame k, $\{m_l^{i \to j} | i, j \leq k\}$, extract planes, estimate plane equations and camera pose R_k and $\boldsymbol{t}_k$ in each frame.*

5.4 System Architecture

The SLAPSE problem can be solved using an EKF-based filtering approach as shown in Figure 5.3a. The system takes MVs as the input, and tracks the 3D configuration of planes and camera poses. A key issue of the procedure is how to extract planes from MVs, which is detailed in Figure 5.3b. Let us start with the planar surface extraction.

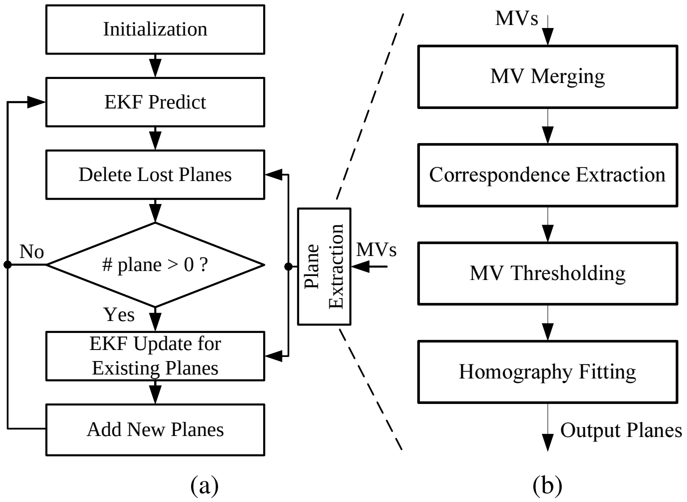


Figure 5.3: System diagrams: (a) Overall SLAPSE diagram based on EKF. (b) A blowup view of plane extraction.

5.5 Planar Surface Extraction

Planes are identified through MVs. Given that MVs may have multiple reference frames, we need to merge them to facilitate the plane extraction. Moreover, it is necessary to understand how errors in MVs are accumulated and propagated in the MV merging process.

5.5.1 Motion Vector Merging

According to the noise model in Section 5.3.2, an MV represents correct MB correspondence between the current B or P frame and its reference frame with probability p. We name MVs with correct correspondence as in-line MVs (IMVs). From scene understanding point of view, IMVs have limited spatial resolution and relatively high noise. However, IMV set is actually temporally abundant. The adjacent frames differ by 1/30 or 1/25 seconds. If done properly, we can utilize IMV's temporal abundance to further reduce noise level. Since IMV accuracy determines the accuracy of scene structure, it is

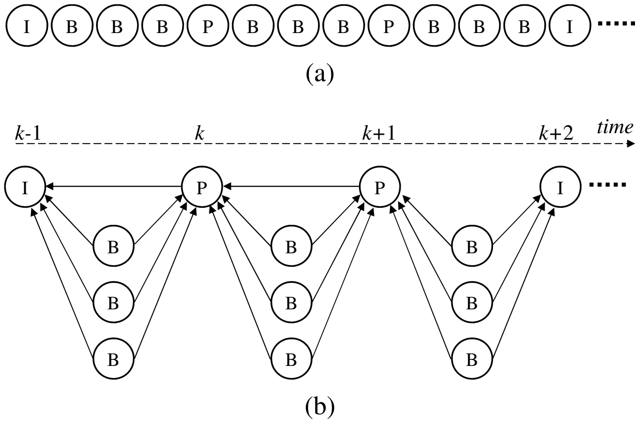


Figure 5.4: MVs in B frames are merged into the nearest P and I frames. Arrows indicate the MV referencing directions. (a) A sample GOP. (b) The GOP can be decomposed into IP, PP and PI types.

important to monitor the IMV variance level. Therefore, the subsequent questions are 1) what is the probability that the IMVs exist across multiple frames and 2) how accurate are these IMVs.

We begin with question 1). For a sample GOP in Figure 5.4a, we can draw the MV reference relationship in Figure 5.4b. Interestingly, the continuous frame sequence can be broken into segments with each segment beginning with an I/P frame and ending with the nearest subsequent I/P frame. Segments overlap by sharing common I or P frames. Let n_{B} be the number of B frames in each segment. $n_{\mathrm{B}} = 3$ in Figure 5.4. Utilizing these natural segments, we check IMV existence every $n_{\mathrm{B}}+1$ frames as defined by each segment. There are three types of segments according to beginning/ending frame types: IP, PP, and PI. IP and PP share a similar structure: a direct reference between the two and n_{B} indirect references from B frames. PI pairs do not have the direct reference because I frames are not constructed from MBs. Define events E_{IP}, E_{PP}, and E_{PI} for the existence of IMV for an MB across the nearest IP, PP, and PI frames, respectively. We have the following lemma.

Lemma 4. *For an MB, the probability of existing at least one IMV across the nearest I/P* *frame pair is,*

$$P(E_{IP}) = P(E_{PP}) = 1 - (1-p)(1-p^2)^{n_B}, \tag{5.3}$$

$$P(E_{PI}) = 1 - (1-p^2)^{n_B}. \tag{5.4}$$

Proof. We can view the MV reference relationship in Figure 5.4b as a probability graph where each edge has a probability of p that the MV is a correct correspondence. Therefore, for each path passing B frames, the probability that both left and right edges are correct is p^2. Subsequently, the probability that the path is incorrect is $1-p^2$. $\overline{E}_{\text{PI}}$ happens if all paths passing B frames are incorrect. Hence $P(\overline{E}_{\text{PI}}) = (1-p^2)^{n_{\text{B}}}$. Eq. (5.4) holds. Similarly, we can obtain $P(E_{\text{IP}})$ and $P(E_{\text{PP}})$. □

Lemma 4 indicates that using B frames can increase the probability of IMV existence. In fact, we often have more than one IMV for each MB. Let us define frame index (also used as time index) variable k and $k+1$ corresponding to an adjacent P/I pair in a segment (see Figure 5.4b). Define set L_{IMV} as the set of IMVs for the MB. We know that IMVs are from two sources: the direct reference between I or P frames and indirect references from B frames. The error in the former follows $N(\mathbf{0}_{2\times1}, \mathbf{\Sigma})$ in (5.2) whereas the error in the latter is the summation of two independent 2D Gaussian in (5.2) and hence follows $N(\mathbf{0}_{2\times1}, 2\mathbf{\Sigma})$. We define event E_D if there exists a correct direct reference and d as the index for the MV. For each MB, we aggregate MVs at I or P frames by minimizing the Mahalanobis distance,

$$m_l^{k+1\to k}|E_D = \frac{\sqrt{2}m_d^{k+1\to k} + \sum_{\eta\in L_{\text{IMV}},\eta\neq d} m_\eta^{k+1\to k}}{\sqrt{2}+|L_{\text{IMV}}|-1}, \tag{5.5}$$

$$m_l^{k+1\to k}|\overline{E}_D = \frac{1}{|L_{\text{IMV}}|}\sum_{\eta\in L_{\text{IMV}}} m_\eta^{k+1\to k}. \tag{5.6}$$

The aggregation results in the following error distribution:

Lemma 5. *The error* $\boldsymbol{e}_l^{k+1\to k} = m_l^{k+1\to k} - \bar{m}_l^{k+1\to k}$ *of the resulting MV is distributed with zero mean:*

$$\boldsymbol{e}_l^{k+1\to k}|E_* \sim N(\mathbf{0}_{2\times 1}, \mathbf{\Sigma}_{\boldsymbol{*}}|E_*), \tag{5.7}$$

where condition '$*$' represents IP, PP, and PI pairs, and three conditional covariance matrices are:

$$\mathbf{\Sigma}_{PI}|E_{PI} = \left[\sum_{i=1}^{n_B} \frac{2}{i} \begin{pmatrix} n_B \\ i \end{pmatrix} \frac{p^{2i}(1-p^2)^{n_B-i-1}}{1-(1-p^2)^{n_B}}\right] \mathbf{\Sigma}, \tag{5.8}$$

$$\begin{aligned} \mathbf{\Sigma}_{IP}|E_{IP} = \mathbf{\Sigma}_{PP}|E_{PP} &= (1-p)\mathbf{\Sigma}_{PI}|E_{PI} \\ &+p\left[\sum_{i=0}^{n_B} \frac{2+i}{(i+\sqrt{2})^2} \begin{pmatrix} n_B \\ i \end{pmatrix} p^{2i}(1-p^2)^{n_B-i}\right] \mathbf{\Sigma}. \end{aligned} \tag{5.9}$$

Proof. Let us begin with $\mathbf{\Sigma}_{\text{PI}}$. Denote $\xi = |L_{\text{IMV}}|$ as the number of IMV for the MB. ξ is conformal to binomial distribution $B(n_{\text{B}}, p^2)$,

$$P(\xi = i) = \begin{pmatrix} n_{\text{B}} \\ i \end{pmatrix} p^{2i}(1-p^2)^{n_{\text{B}}-i}. \tag{5.10}$$

Event E_{PI} means $\xi \geq 1$. Therefore, we have,

$$P(\xi = i|\xi \geq 1) = \frac{P(\xi = i, \xi \geq 1)}{1 - P(\xi = 0)} = \begin{pmatrix} n_{\text{B}} \\ i \end{pmatrix} \frac{p^{2i}(1-p^2)^{n_{\text{B}}-i}}{1-(1-p^2)^{n_{\text{B}}}}, \text{ for } i = 1, ..., n_{\text{B}}. \tag{5.11}$$

Recall that $\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}} = \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}})$ where $\text{Var}(\cdot)$ means the covariance of the random rector. Conditioning on the value of ξ, from the property of conditional variance, we know

$$\begin{aligned} \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}) &= E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)) + \text{Var}(E(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)) && (5.12)\\ &= E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)), && (5.13) \end{aligned}$$

where $E(\cdot)$ means expectation of the random vector. Eq. (5.13) is true because

$$\text{Var}(E(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)) = \mathbf{0}_{2\times 2} \tag{5.14}$$

due to the fact that each $\boldsymbol{e}_l^{k+1\to k}$ is zero mean. From the property of conditional expectation, we have,

$$E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)) = \sum_{i=1}^{n_{\text{B}}} \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi)P(\xi = i|\xi \geq 1). \tag{5.15}$$

According to (5.6), $\boldsymbol{e}_l^{k+1\to k}|(E_{\text{PI}},\xi)$ is an average of i independent Gaussian $N(\mathbf{0}_{2\times 1}, 2\boldsymbol{\Sigma})$. Hence the resulting vector is still Gaussian with

$$\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}},\xi) = \frac{2\boldsymbol{\Sigma}}{i}. \tag{5.16}$$

Combining (5.11-5.16), we obtain (5.8).

It is clear that $\boldsymbol{\Sigma}_{\text{IP}}|E_{\text{IP}}$ and $\boldsymbol{\Sigma}_{\text{PP}}|E_{\text{PP}}$ share the same value due to the same structure shown in Figure 5.4b. We use $\boldsymbol{\Sigma}_{\text{IP}}|E_{\text{IP}}$ to show the proof process. Conditioning on the event E_D, we have

$$\begin{aligned}\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}} &= E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)) &(5.17)\\ &= \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)P(E_D)\\ &\quad + \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, \bar{E}_D)P(\bar{E}_D)\\ &= \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)p + \mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}}(1-p). &(5.18)\end{aligned}$$

Note that (5.17) is true because the zero mean property is applied to conditional variance computation (similar to (5.13)). Also, when $\bar{E}_D$ occurs, $\mathbf{\Sigma}_{\text{IP}}|E_{\text{IP}}$ is reduced to $\mathbf{\Sigma}_{\text{PI}}|E_{\text{PI}}$ according to Figure 5.4b.

To compute $\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{PI}}, E_D)$, we can further condition on ξ, which is similar to how (5.8) has been derived. However, there are two different scenarios: the first is that (5.10) becomes

$$P(\xi - 1 = i) = \begin{pmatrix} n_{\text{B}} \\ i \end{pmatrix} p^{2i}(1-p^2)^{n_{\text{B}}-i} \tag{5.19}$$

and we do not need to use the conditional binomial defined in (5.11) because E_D means $\xi - 1 \geq 0$ is always true. Consequently, (5.15) is modified as

$$E(\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi)) = \sum_{i=0}^{n_{\text{B}}} \text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi)P(\xi - 1 = i). \tag{5.20}$$

The second difference is the fact that we employ (5.5) to aggregate heterogeneous Gaussian distributions which include one error vector in $N(\mathbf{0}_{2\times1}, \mathbf{\Sigma})$ and $\xi - 1$ error vectors in $N(\mathbf{0}_{2\times1}, 2\mathbf{\Sigma})$. Therefore, (5.16) is changed to the following,

$$\text{Var}(\boldsymbol{e}_l^{k+1\to k}|E_{\text{IP}}, E_D, \xi) = \frac{2+i}{(\sqrt{2}+i)^2}\mathbf{\Sigma} \tag{5.21}$$

because (5.5) is just a linear combination of independent Gaussian distributions. Combining these equations, we obtain (5.9). □

Remark 3. *Actually, both (5.8) and (5.9) are decreasing functions of n_B. This means that merging MVs from B frames into the nearest I/P frames reduces error variance. This process allows us to exchange the redundant temporary resolution to better spatial resolution.*

This allows us to obtain a set of merged MVs which are denoted as $\mathcal{M}^{k+1\to k} = \{m^{k+1\to k}\}$ for each adjacent frames $k+1$ and k. Lemmas 4 and 5 ensure IMV existence and derive the corresponding error. A merged MV $m^{k+1\to k}$ provides a correspondence relationship between an MB in $k+1$ and an MB in k which naturally leads to correspondence extraction step.

5.5.2 Correspondence Extraction and MV Thresholding

Define $\boldsymbol{x}_k$ to be the homogeneous form of a point in image k. We represent the motion correspondence by a point pair:

$$\boldsymbol{x}_k = \boldsymbol{x}^c_{k+1} + \begin{bmatrix} m^{k+1\to k} \\ 0 \end{bmatrix}, \tag{5.22}$$

where $\boldsymbol{x}^c_{k+1}$ is the center of $m^{k+1\to k}$'s MB in $k+1$, and $\boldsymbol{x}_k$ is its corresponding position in frame k. Therefore, a set of correspondences between frame k and $k+1$ is obtained:

$$\mathcal{C}^{k+1\to k} =\{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}^c_{k+1} : m^{k+1\to k} \in \mathcal{M}^{k+1\to k}\}. \tag{5.23}$$

To reduce the influence of MV noise in plane estimation, we only consider planes with sufficient motion parallax. This is handled by eliminating MVs belonging to the plane at infinity which is defined as $\boldsymbol{\pi}_\infty$.

According to [129], points in $\boldsymbol{\pi}_\infty$ remain still during camera translation, therefore, they can be detected if the camera rotation is eliminated from the images.

For a pair of adjacent frames k and $k+1$, their fundamental matrix is first estimated using correspondence $\mathcal{C}^{k+1\to k}$. Camera rotation and translation are then decomposed using [142]. We re-project all $\boldsymbol{x}_k$'s to frame $k+1$ using only the rotation matrix, which results in a set of points $\boldsymbol{x}'_{k+1}$.

$$\boldsymbol{x}'_{k+1} = sK({}^{k}_{k+1}R)^{-1}K^{-1}\boldsymbol{x}_k, \tag{5.24}$$

where s is a scalar, and $({}^{k}_{k+1}R)$ is the matrix that rotates $\{C_k\}$ to $\{C_{k+1}\}$ according to the convention used in [113].

The distance between $\boldsymbol{x}'_{k+1}$ and $\boldsymbol{x}^c_{k+1}$ is calculated, and the MV is considered in $\boldsymbol{\pi}_\infty$ if the distance is below a threshold ϵ_m. Denote the correspondence set for $\boldsymbol{\pi}_\infty$ as

$$\mathcal{C}^{k+1\to k}_{\infty} =\{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}^c_{k+1} : \|\boldsymbol{x}'_{k+1} - \boldsymbol{x}^c_{k+1}\| < \epsilon_m\}, \tag{5.25}$$

where subscript ∞ means it corresponds to the plane at infinity and $\|\cdot\|$ represents the L_2 norm. Hence the set of correspondences is further reduced to

$$\mathcal{C}^{k+1\to k}_{m} = \mathcal{C}^{k+1\to k} \setminus \mathcal{C}^{k+1\to k}_{\infty}, \tag{5.26}$$

where subscript m means the thresholded correspondence set with sufficient motion parallax.

5.5.3 Homography Fitting

With the correspondence set extracted, plane extraction can be performed by verifying the homography relationship. The extraction of planes also helps filter IMVs from the correspondence set.

Consider two adjacent frames (IP, PP or PI) after MV merging and thresholding (Figure 5.3b). We have the correspondence set $\mathcal{C}_m^{k+1\to k}$. We apply RANSAC framework to extract 2D planes and IMVs. RANSAC first samples a minimum set of correspondences to obtain a homography that represents the coplanar relationship

$$\boldsymbol{x}_k = \lambda H \boldsymbol{x}_{k+1}^c, \tag{5.27}$$

where H is a 3×3 matrix and λ is a scalar.

Each correspondence provides two equations to (5.27). Since a homography H has at most 8 degrees of freedom (DoFs), only four correspondences are needed to determine a minimal solution. A normalized direct linear transformation (DLT) can be applied to obtain an initial H (page. 109 of [129]). Then, a correspondence resulting in an error below a given threshold:

$$\|\boldsymbol{x}_k - \lambda H \boldsymbol{x}_{k+1}^c\| < \epsilon_h, \tag{5.28}$$

is labeled as an inlier to the plane.

To extract multiple planes, RANSAC is applied iteratively until it reaches a given maximum iteration number or there are not enough unlabeled correspondences to form a minimum solution. Denote the correspondence set $\mathcal{C}_{\pi,i}^{k+1\to k}$ for plane π_i (defined by homography H_i) as

$$\mathcal{C}_{\pi,i}^{k+1\to k} = \{\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}_{k+1}^c : \|\boldsymbol{x}_k - \lambda H_i \boldsymbol{x}_{k+1}^c\| < \epsilon_h\}. \tag{5.29}$$

Hence we obtain a set of N_{k+1} planes with correspondences $\{\mathcal{C}_{\pi,1}^{k+1\to k}, ..., \mathcal{C}_{\pi,N_{k+1}}^{k+1\to k}\}$ from frame k and $k+1$.

Note, if a set of planes with correspondences $\{\mathcal{C}_{\pi,1}^{k\to k-1},...,\mathcal{C}_{\pi,N_k}^{k\to k-1}\}$ have been extracted between frames $k-1$ and k, we first run RANSAC to sample the minimum solutions only from MBs of existing planes. Thus every existing plane π_i has a chance to find its corresponding plane correspondence set $\mathcal{C}_{\pi,i}^{k+1\to k}$ in frame $k+1$. Then a regular RANSAC is applied to the remaining correspondences to discover new planes between frames k and $k+1$.

5.6 Plane Tracking with EKF

With planes extracted, we can feed them as observations to an EKF framework to estimate the global plane equations and camera poses. An EKF filtering approach usually consists of prediction and update steps.

5.6.1 EKF Prediction

In the state space description, we define state vector $\boldsymbol{\mu}_k$ to be consisted of plane equations in inhomogeneous form, camera rotation angles and angular velocity, and camera translation and its velocity in frame k,

$$\boldsymbol{\mu}_k = [\tilde{\boldsymbol{\pi}}_{1,k}^\mathsf{T},...,\tilde{\boldsymbol{\pi}}_{N_k,k}^\mathsf{T},\boldsymbol{r}_k^\mathsf{T},\boldsymbol{t}_k^\mathsf{T},\dot{\boldsymbol{r}}_k^\mathsf{T},\dot{\boldsymbol{t}}_k^\mathsf{T}]^\mathsf{T}, \tag{5.30}$$

where $\boldsymbol{r}=[\alpha,\beta,\gamma]^\mathsf{T}$ defines the Euler rotation angles in $X'Y'Z'$ order, $\boldsymbol{t}=[t_x,t_y,t_z]^\mathsf{T}$ defines the camera translation w.r.t. previous frame, and $\dot{\boldsymbol{t}}$ defines translation velocity in current frame.

Denote Euler rotation matrix $\bar{R}_k = R(\tau\dot{\boldsymbol{r}}_k)$ in $Y'X'Z'$ order. The state transition of the $i-$th plane equation is

$$\tilde{\boldsymbol{\pi}}_{i,k+1} = \frac{\bar{R}_k^\mathsf{T}\tilde{\boldsymbol{\pi}}_{i,k}}{\tau\dot{\boldsymbol{t}}_k^\mathsf{T}\bar{R}_k^\mathsf{T}\tilde{\boldsymbol{\pi}}_{i,k}+1}. \tag{5.31}$$

We assume the camera follows constant angular velocity and linear translation velocity. Hence the state transition is,

$$\begin{cases} \boldsymbol{r}_{k+1} &= \tau \dot{\boldsymbol{r}}_k \\ \boldsymbol{t}_{k+1} &= \tau \dot{\boldsymbol{t}}_k \\ \dot{\boldsymbol{r}}_{k+1} &= \dot{\boldsymbol{r}}_k \\ \dot{\boldsymbol{t}}_{k+1} &= \bar{R}_k \dot{\boldsymbol{t}}_k \end{cases} . \tag{5.32}$$

5.6.2 EKF Update

To utilize rich information from MVs, we do not consider simply making a direct observation of the plane equations. Instead, we use the correspondence sets $\mathcal{C}_{\pi,i}^{k\to k-1}$'s to update the state vectors.

For frame k, the observation of a plane $\boldsymbol{\pi}_{i,k}$ is a set of points $\{\boldsymbol{x}_{k-1}\}$ from $\mathcal{C}_{\pi,i}^{k\to k-1}$. Define rotation matrix $R_k = R(\boldsymbol{r}_k)$ following the $Y'X'Z'$ Euler form. The observation model for plane $\boldsymbol{\pi}_{i,k}$ takes the state vector $\boldsymbol{\mu}_k$ and an additional variable $\boldsymbol{x}_k^c$ as input:

$$\boldsymbol{x}_{k-1} = h(\boldsymbol{\mu}_k, \boldsymbol{x}_k^c) = K[R_k - \boldsymbol{t}_k \tilde{\boldsymbol{\pi}}_{i,k}^{\mathsf{T}}]K^{-1}\boldsymbol{x}_k^c, \tag{5.33}$$

where K is the intrinsic matrix of the camera. The Jacobian matrix is computed by taking partial derivatives on $\boldsymbol{\mu}_k$.

Lemma 5 in Section 5.5.1 provides the error model for the merged MVs, and is applied in setting the noise covariance for the EKF observation.

Note that, since the camera rotation and translation are involved in the observation model for each plane, $\boldsymbol{r}_k$ and $\boldsymbol{t}_k$ are also updated with observations.

5.6.3 Deleting and Adding Planes

Similar to landmark management in SLAM, planes have finite lifespan in the continuous video stream. We need to handle the appearance and disappearance of planes in camera views (see Figure 5.3a).

When transiting from frame k to $k+1$, if $\tilde{\boldsymbol{\pi}}_{i,k}$ has corresponding set $\mathcal{C}_{\pi,i}^{k+1\to k} = \emptyset$ in frame $k+1$, then $\tilde{\boldsymbol{\pi}}_{i,k+1}$ in the state vector and its corresponding dimensions in the state covariance matrix are deleted, before EKF update.

After EKF update in frame k, if a new plane is discovered in frame k, its initialized plane equation and variance are added to the state vector and state covariance matrix. Moreover, since the filter formulation relies purely on planes in EKF updating step, the update is skipped if there are no planes in current state vector. This is not an issue as long as building facades are in the field of view.

5.7 Experiments

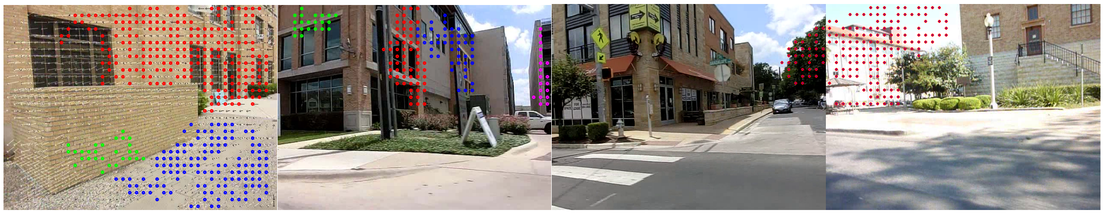
The proposed method is implemented in C/C++ on a desktop PC. Videos and images are acquired with Casio Ex-ZR200 and Panasonic DMC-ZS3 cameras, with a resolution of 640×480 pixel captured at 30 frames per second. Cameras travel in an urban area at a speed between 25 and 50 kph.

5.7.1 Plane Extraction

To evaluate the performance of plane extraction, 7 videos of different scenes in MPEG-2 format have been acquired. We sample 50 pairs of adjacent frames from the videos, and manually label planes in images as ground truth. Figure 5.5 shows sample thumbnails from the dataset. In this experiment, MVs in $\boldsymbol{\pi}_\infty$ have not been filtered out.

As the error threshold of RANSAC changes, the number of extracted planes and the true positive (TP) rates vary. Table 5.1 shows how the plane extraction result is influenced

Figure 5.5: Sample images in plane extraction dataset.



(a) $\epsilon_h = 1$ (b) $\epsilon_h = 2$ (c) $\epsilon_h = 2$ (d) $\epsilon_h = 4$

Figure 5.6: Example of extracted planes. Dots with different colors indicate different extracted planes. (a-b) show all planes extracted in the frame. (c-d) show two incorrect extractions.

by ϵ_h. Note that we restrict the minimum size of an extracted plane to be 20 MBs.

Table 5.1: Plane extract results w.r.t. ϵ_h

ϵ_h (pixel)	1	2	3	4
# extracted planes	101	183	174	215
TP rate (%)	91.09	83.61	73.56	72.09

Figure 5.6 shows four example frames. Dots in the same color indicate an extracted plane. It is clear that the algorithm is able to extract primary planes. However, it may miss some reflective glass/mirror surfaces, such as the leftmost wall in Figure 5.6b, and texture-

less surfaces such as the ground. Some false extractions, such as Figure 5.6c, claim trees as a plane due to far depth. In fact, Figure 5.6d shows the necessity of MV thresholding with $\boldsymbol{\pi}_\infty$ (Section 5.5.2), because far field objects tend to mix together when ϵ_h is not tight enough.

5.7.2 SLAPSE Results

To evaluate overall system performance, we perform field tests in two sites. Ground truth is manually acquired with meters and Bosch ZLR225 laser distance measurer with an accuracy of ± 1.5 mm.

The 3D estimation is up to scale of the initial camera translation. Sample results from the first site are shown in Figure 5.1. It is clear that the system is able to extract dominant planes in the scene.

We project the camera trajectories to $\{C_0\}$ and scale the results by the camera translation in the first step. Comparison with manually measured ground truth is showed in Table 5.2. We denote D as the total traveled distance in each site, and a $\hat{}$ on a variable stands for the ground truth value. Denote $\boldsymbol{t}^{0\to k}$ as the estimated camera translation from frame 0 to k. The mean relative error of camera location is defined as:

$$\epsilon_D = \frac{1}{N}\Sigma_k \frac{\|\boldsymbol{t}^{0\to k} - \hat{\boldsymbol{t}}^{0\to k}\|}{\|\hat{\boldsymbol{t}}^{0\to k}\|}, \tag{5.34}$$

where N is the total number of tracked frames.

We evaluate the estimated building facades and road segments which appear in the camera scene for at least half a second. The number of evaluated planes in each site is shown in Table 5.2. Define the mean absolute error of plane depth ϵ_d and plane orientation

ϵ_n as:

$$\epsilon_d = \frac{1}{\Sigma_i N_i} \Sigma_i \Sigma_k |d_{i,k} - \hat{d}_{i,k}|, \tag{5.35}$$

$$\epsilon_n = \frac{1}{\Sigma_i N_i} \Sigma_i \Sigma_k |\arccos(\boldsymbol{n}_{i,k}^{\mathsf{T}} \cdot \hat{\boldsymbol{n}}_{i,k})|, \tag{5.36}$$

where N_i is the number of frames plane i appears. Table 5.2 shows the mean errors for each site, where the depth errors are less than 0.65 meters and orientation errors are less than 7.07 degrees.

Table 5.2: SLAPSE results

Site	D (m)	$\epsilon_D(\%)$	# planes	ϵ_d (m)	ϵ_n (degs.)
1	42.1	2.9	5	0.61	7.07
2	37.5	5.1	4	0.65	3.26

5.8 Conclusions and Future Work

We explored how to use MVs from video streams for SLAPSE for a mobile robot equipped with a single camera. Using MVs in the MPEG-2 protocol as an example, we established the MV noise models to capture the observation error. We formulated the SLAPSE problem and studied how to extract planes from MVs using planar homography filtering. We then developed an extended Kalman filter (EKF) based approach with planes and robot motion as state variables. We implemented our algorithm using C/C++ on a PC platform, and tested the algorithm in physical experiments in two sites. The results showed that the system is capable of performing robot localization and plane mapping with a relative trajectory error of less than 5.1%.

In the future, we plan to utilize the MVs in the plane at infinity for rotation estimation.

We can also detect moving obstacles by group MVs with similar motion. The entire system can be merged under an interactive multi-model (IMM) EKF to improve results and provide a comprehensive navigation solution. Local bundle adjustment can be embedded as a post-processing step to improve plane estimation accuracy. We will also include the plane segmentation through the re-projection of MBs. Also, the MVs can be combined with feature-based approaches and/or other sensors to form hybrid methods.

6. SIMULTANEOUS LOCALIZATION, PLANAR SURFACE EXTRACTION, AND MOVING OBSTACLE TRACKING

6.1 Introduction

For most mobile robots in GPS-challenged environments, simultaneous localization and mapping (SLAM) and obstacle avoidance are two critical navigation functionalities. They are often handled separately because SLAM usually views moving obstacles as noises in the environment whereas obstacle avoidance only concerns the relative motion between the robot and obstacles. This artificial separation was mostly due to the limitation of existing methods. Both SLAM results and obstacle motion information should be considered together when planning robot trajectories in real applications. In fact, the artificial separation can lead to problems such as synchronization or redundant processing of information, which are not desirable for time, power, and computation constrained mobile robots.

Motion vectors (MVs) characterize the movement of pixel blocks in video streams, which are readily available. With a monocular camera as the only sensor, we have employed MVs from video streams to create a new featureless SLAM method for visual navigation in previous section. However, the method still assumes a stationary environment despite that MVs encode motion information for both the environment and moving objects.

Here we show that MVs allow us to develop a new algorithm that is capable of performing the SLAM task and obstacle tracking in a single framework by simultaneous localization, planar surface extraction, and tracking of moving objects. Assuming a quasi-rectilinear urban environment, this method first extracts planes from MVs and their corresponding pixel micro blocks (MBs). We classify MBs as stationary or moving. These steps are based on geometric constraints and properties of plane-induced homographies under random sample consensus (RANSAC) framework. Planes are labeled as part of the stationary scene or moving obstacles using an MB voting process. This allows us to establish planes as observations for extended Kalman filters (EKFs) for both the stationary scene and moving objects. We have implemented the proposed method and compared it with the state-of-the-art 1-Point EKF [1]. The results show that the proposed method achieves similar localization accuracy. The relative absolute error is less than 2.53%. At the same time, our method can directly provide plane-based rectilinear scene structure, which is a higher level of scene understanding, and is capable of detecting moving obstacles at a true positive rate of 96.6%.

6.2 Related Work

For many vSLAM works, a common assumption is that the environment is stationary. This assumption becomes invalid when a robot navigates in an urban environment due to moving vehicles and pedestrians. In recent years, vSLAM in dynamic environments receives increasing research attention. In existing methods, this problem is separated as a vSLAM in a stationary environment and a 3D visual tracking problem for each moving object [143, 144]. Our work is similar to these works in that we use multiple filters to track stationary and moving objects separately. However, existing methods do not perform motion separation and only work when the stationary landmarks are fixed or the moving objects' templates are given. To integrate motion separation with vSLAM, Zhou et al. [61] propose a multi-camera based approach using multiple views to triangulate points and compare the reprojection error between frames to differentiate stationary and moving points. For a monocular camera, triangulation is not available within a single frame. Therefore, our work relies on an MV-based motion segmentation method using adjacent frames.

The motion separation in our work relates to motion-based object detection in monocular vision. Many existing MV-based object detection approaches require stationary background [133, 145–147]. Assuming that MBs on an object have the same motion, different clustering methods, such as expectation-maximization (EM) [145] and mean-shift [147], are used to classify foreground MVs into different regions. With the given object regions, the tracking can be performed by searching along all MVs in the object region [133]. However, these methods do not apply to our problem because the background is not stationary in our videos, and the object motion cannot be approximated by affine motion. Similar to MVs, optical flows (OFs) enable many motion-based object detection work [148–150]. When the camera moves, OFs are used to detect a single dominant plane with the homography constraint [149]. When the dominating plane is the ground plane in [150], an OF model for the ground plane movement is estimated according to the camera motion where all mis-matchings to the model are detected as obstacles. Considering the low accuracy of MVs, we also use planes as landmarks in our work. However, the camera motion is unknown in our model.

6.3 Problem Formulation

6.3.1 System Overview and Introduction to Motion Vectors

Figure 6.1 shows that the proposed system consists of three parts: the plane extraction and camera motion estimation (top), the stationary scene filter (middle), and the moving object filter (bottom). The plane extraction and camera motion estimation takes MVs as input and outputs labeled stationary/moving planes and the estimated camera motion between the adjacent frames. The extracted stationary planes and camera motion information are fed into the stationary scene filter to perform localization and mapping tasks. The extracted moving planes are entered to the moving object filter for tracking. Since moving and stationary planes are not permanent in applications (e.g. a moving car may come to

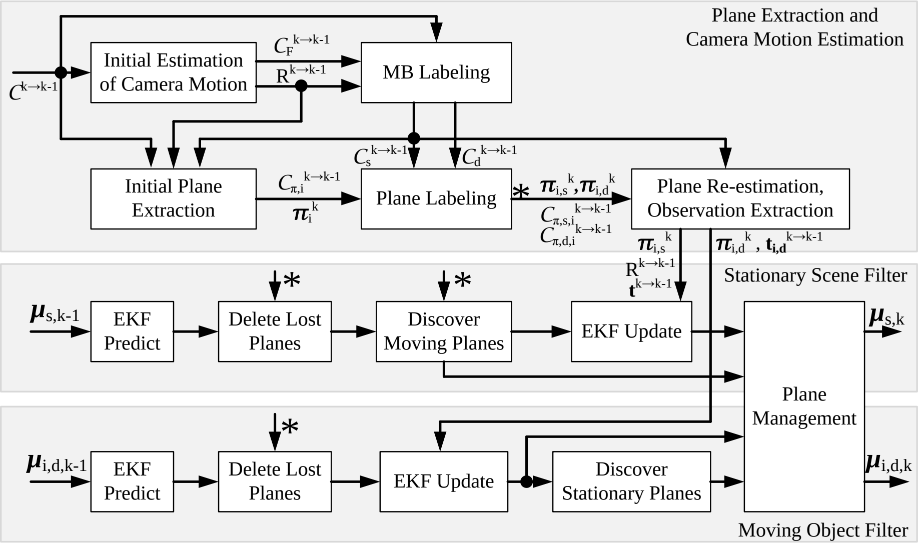


Figure 6.1: System diagram. The $*$ represents the output of plane labeling, which is also the input to three sub-blocks below.

a stop), a plane management module is introduced to allow us to add, remove, verify, and re-label them according to EKF outputs.

Filtered MVs are the input to the entire system. Let us briefly introduce MVs here. Detailed description and the filtering process can be found in [151]. Moving Picture Experts Group (MPEG) stands for a class of video compression algorithms that are the most popular in use today. To achieve compression, each frame is partitioned into MBs in MPEG-1/2/4 standards (e.g. MPEG-2 codec uses 16×16-pixel MB). During encoding, block matching is performed to find similar MBs in reference frames. An MV is then established to represent a 2D shift of an MB with respect to (w.r.t) the reference frame. Depending on group of picture structure in different MPEG protocols, raw MVs may point to multiple future or past reference frames. It is worth noting that MVs are often noisy or missing due to the fact that MVs are computed purely based on the similarity of MBs. The similarity could be corrupted by occlusion, lighting, and large perspective changes or tricked by repetitive patterns.

Comparing to optical flows, MVs are readily available. However, MVs are sparser in spatial resolution but denser in temporal dimension. In [151], we have showed how to exploit this characteristic to reduce noise in MVs, which results in the filtered MVs. Actually, filtered MVs represent the set of corresponding MBs between key frames k and $k-1$, and are denoted by

$$\mathcal{C}^{k \to k-1} := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c\}, \tag{6.1}$$

where $\boldsymbol{x}_k^c$ indicates the center of the MB and $\boldsymbol{x}_{k-1}$ shows its corresponding position in reference frame $k-1$.

6.3.2 Problem Definition

To formulate the problem, we assume the urban scene can be approximated using planes: stationary or moving. A set of stationary planes is a good representation of quasi-rectilinear urban environments and always exists in sight. Moving planes can approximate vehicle exteriors. We consider there are more stationary planes than moving objects. We also assume that moving planes follow pure translation in the short duration of observation. The intrinsic camera matrix K is constant and known through pre-calibration. All 3D coordinate systems are right-handed coordinates, and common notations are defined as follows:

- *Coordinate systems*: $\{\Phi_k\}$ is a camera coordinate system (CCS) at frame k. For each CCS, its origin locates at the camera optical center, z-axis coincides with the optical axis and points to the forward direction of the camera, its x-axis and y-axis are parallel to the horizontal and vertical directions of the CCD sensor plane, respectively. The world coordinate system (WCS) $\{W\}$ coincides with $\{\Phi_0\}$. To differentiate variables in CCS and WCS, a superscription k means the variable is in $\{\Phi_k\}$ or its corresponding image coordinate system, while no superscription is default for $\{W\}$. In addition, a superscription $k \to k-1$ means from $\{\Phi_k\}$ to

$\{\Phi_{k-1}\}$

- *Image coordinate system*: $\boldsymbol{x} \in \mathbb{P}^2$ is the homogeneous representation of an image coordinate where $\mathbb{P}^2$ is 2D projective space.
- *3D planes*: $\boldsymbol{\pi} = [\boldsymbol{n}^\mathsf{T}, d]^\mathsf{T}$ represents a 3D plane, where $\boldsymbol{n} \in \mathbb{R}^3$ is the plane normal vector and d is the plane depth. $\tilde{\boldsymbol{\pi}} = \boldsymbol{n}/d$ is the inhomogeneous form.
- *Subscripts*: k is the time/frame index. To distinguish stationary scene and moving objects, a subscript s stands for stationary and a subscript d represents dynamically moving. For example, $\boldsymbol{\pi}_{s,k}$ is a stationary plane at frame k.
- $\varepsilon_F(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, F)$ denotes the Sampson's error (p. 287 in [129]) for fundamental matrix F, where $\boldsymbol{x}_k^\mathsf{T} F \boldsymbol{x}_{k-1} = 0$. $\varepsilon_H(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, H)$ denotes the Sampson's error (p. 99 in [129]) for homography matrix H, where $\boldsymbol{x}_{k-1} = H\boldsymbol{x}_k$.

With the notations defined, we formulate the problem as below:

Definition 6. *Given the set of MVs, $\mathcal{C}^{k \to k-1}$, up to time/frame k, estimate camera pose R_k and $\boldsymbol{t}_k$ in each frame, identify/label MBs for each plane, and reconstruct stationary and moving planes.*

To solve this problem, we begin with planar surface extraction and camera motion estimation (top box in Figure 6.1).

6.4 Planar Surface Extraction and Camera Motion Estimation

Since MVs are often too noisy to be used directly, we exploit the coplanar property of MBs in each adjacent key frame pair to filter MVs. We estimate camera motion first and then use the motion information to label MBs by identifying whether they belong to stationary scene or moving objects. This allows us to establish planes as observations for the later EKF-based approach.

6.4.1 Initial Estimation of Camera Motion

With the input MVs $\mathcal{C}^{k\to k-1}$ defined in (6.1), let us estimate camera motion between two adjacent frames. The correct MV for the stationary scene across adjacent frames should conform the relation

$$(\boldsymbol{x}_k^c)^\mathsf{T} F^{k\to k-1} \boldsymbol{x}_{k-1} = 0, \tag{6.2}$$

where $F^{k\to k-1}$ is the fundamental matrix between the two frames. We first obtain an initial $F^{k\to k-1}$ using normalized 8-point algorithm under RANSAC framework (p. 281 in [129]). This gives the inlier correspondence set for $F^{k\to k-1}$:

$$\mathcal{C}_F^{k\to k-1} := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c : \|(\boldsymbol{x}_k^c)^\mathsf{T} F^{k\to k-1} \boldsymbol{x}_{k-1}\| < \epsilon_f, \boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}^{k\to k-1}\}, \tag{6.3}$$

where ϵ_f is an error threshold and $\|\cdot\|$ represents the l^2 norm. This verification filters out many non-static MBs and noisy MVs that do not move along the epipolar line, such as the black arrows in Figure 6.2a.

The fundamental matrix can be parameterized by camera rotation and translation as follows:

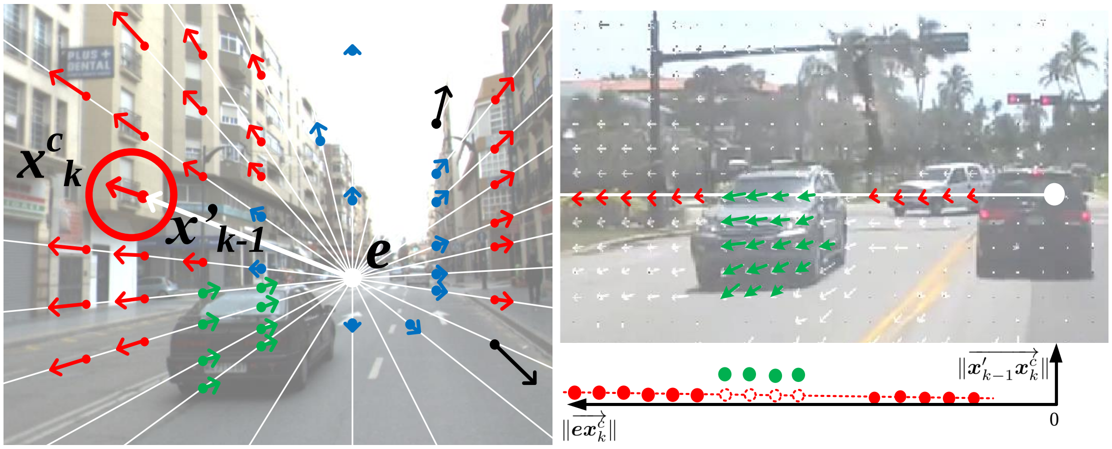
$$F^{k\to k-1} = K^{-\mathsf{T}} [\boldsymbol{t}^{k\to k-1}]_\times R^{k\to k-1} K^{-1} \tag{6.4}$$

where $R^{k\to k-1}$ is the camera rotation matrix from $\{\Phi_k\}$ to $\{\Phi_{k-1}\}$, $\boldsymbol{t}^{k\to k-1}$ is the camera translation from $\{\Phi_k\}$ to $\{\Phi_{k-1}\}$ measured in $\{\Phi_k\}$, and $[\cdot]_\times$ stands for the skew-symmetric matrix representation of the cross product.

Therefore, by minimizing Sampson's error on set $\mathcal{C}_F^{k\to k-1}$:

$$\min_{R^{k\to k-1}, \boldsymbol{t}^{k\to k-1}} \sum_{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_F^{k\to k-1}} \varepsilon_F(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k^c, F^{k\to k-1}), \tag{6.5}$$

we obtain an initial estimation of camera motion between adjacent frames.


x^c_k
x'_{k-1}
e
$\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}\|$
$\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}^c_k}\|$
0

(a) (b)

Figure 6.2: Illustration of MB labeling (best viewed in color). The white dot and lines are the epipole and epipolar lines, respectively. Arrows indicate the movement of MBs between two adjacent frames. (a) MV direction constraint illustration: The camera motion is voted to be “forward”, and red MBs are labeled stationary MBs, green and black MBs are moving MBs, and blue MBs are detected to be on the plane at infinity. (b) MV magnitude constraint illustration. Red arrows are labeled stationary, and the green arrows are moving. The red dashed line illustrates the fitted relationship between $\|\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k\|$ and $\|\boldsymbol{e}\boldsymbol{x}^c_k\|$ along the white epipolar line.

6.4.2 MB Labeling for Stationary and Moving Objects

Before estimating planes, we need to properly classify MBs that belong to moving objects or the stationary scene. The simple verification in (6.3) cannot filter out all MBs on moving objects from the stationary background. If a vehicle moves along the epipolar line, then the corresponding MBs also satisfy (6.3). This happens frequently when a vehicle is in front of the camera and moves in the same direction with the camera on a straight road. The green arrows on the vehicle in Figure 6.2a show a sample case. Since there are two cases: passing vehicles from the same direction of camera motion and approaching vehicles in the opposite direction, we verify the direction and magnitude of the MVs to identify them, respectively.

For a passing vehicle on a straight road, the MVs of the vehicle move along the epipolar line in an opposite direction with the background (e.g. shown by the green arrows in Figure 6.2a). If we know the camera moving direction, these MVs can be detected by checking direction consistency. Therefore, we start with detecting the camera moving direction. Since we know camera rotation from (6.5) and are only interested in camera translation, we can remove the effect of camera rotation first. This is done by projecting $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}'_{k-1}$

$$\boldsymbol{x}'_{k-1} = sKR^{k \to k-1}K^{-1}\boldsymbol{x}_{k-1} \tag{6.6}$$

where s is a scalar. After the projection, the displacement between $\boldsymbol{x}'_{k-1}$ and $\boldsymbol{x}^c_k$ is caused by pure camera translation for stationary MBs. According to epiploar geometry (p. 247 in [129]), when the camera performs a pure translation, the epipole $\boldsymbol{e}$ should be a fixed point, and all stationary MBs should appear to move along lines radiating from the epipole (see Figure 6.2a). The colored dots in the figure are $\boldsymbol{x}'_{k-1}$ and the arrows point to $\boldsymbol{x}^c_k$, an illustration of MVs.

If the camera moves forward along its optical axis, vectors $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ should be in the same direction, as the red arrows in the highlighted circle shown in Figure 6.2a. If the camera moves backward, $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ should be in the opposite direction. Denote the absolute angle between $\overrightarrow{\boldsymbol{e}\boldsymbol{x}'_{k-1}}$ and $\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}$ as α. Of course, the perfect collinear relationship may not hold due to noises in the system. α is always somewhere between $0°$ and $180°$. We examine each MV $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^c_k \in \mathcal{C}^{k \to k-1}_F$. If its α is less than $90°$, a vote of "forward" is assigned, otherwise a "backward" vote is assigned. Then the camera moving direction is obtained as the majority direction from all inlier correspondences. Figure 6.2a shows the camera moving direction is voted as "forward" because most of the MBs move away from the epipole. With the detected camera moving direction, we can identify MBs belonging to passing vehicles easily. However, this would not work for vehicles approaching the camera along the direction parallel to camera motion vector. The MVs on the approaching vehicles also move along the epipolar line and share the same direction as the background motion. For such cases, we need to verify the magnitude of MVs.

The additional motion introduced by the object results in sudden changes of MV magnitude along the epipolar line. To detect this type of moving objects, we start with computing the magnitude of MVs after removing camera rotation. Denote the MV magnitude of $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c$ as $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$, and the Euclidean distance between the MB and the epipole as $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$. From projective geometry we know that closer objects have larger displacements under the same camera motion. Therefore, along one epipolar line, $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ should gradually increase as $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ increases. For each epipolar line, we approximate the 2D relationship between $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ and $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ using RANSAC-based line fitting. An example of the fitted relationship is shown by the dashed line at the bottom of Figure 6.2b. Therefore, for a given $\|\overrightarrow{\boldsymbol{e}\boldsymbol{x}_k^c}\|$ on the epipolar line, an predicted MV magnitude $\|\overset{\sim}{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}}\|$ can be obtained from the fitted relationship (dashed circles in Figure 6.2b). If the difference between $\|\overset{\sim}{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}}\|$ and $\|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}_k^c}\|$ is greater than a threshold ϵ_e, we consider the corresponding MB is potentially moving. In the example shown in Figure 6.2b, the green MBs have magnitudes much greater than the expected red dashed line, and thus labeled as moving MBs.

With the above constraints, we can label every MB and partition the set $\mathcal{C}^{k\to k-1}$ into a stationary correspondence set $\mathcal{C}_s^{k\to k-1}$ and a moving correspondence set $\mathcal{C}_d^{k\to k-1}$, where $\mathcal{C}_s^{k\to k-1} \bigcup \mathcal{C}_d^{k\to k-1} = \mathcal{C}^{k\to k-1}$:

Definition 7 (MB Labeling)**.** *For an MV $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}^{k\to k-1}$ and its corresponding MBs, they are labeled as stationary $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_s^{k\to k-1}$ if the following three conditions are all satisfied:*

1) $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_F^{k\to k-1}$, *2)* $\alpha < 90^\circ$ *if camera moves forward or* $\alpha \geq 90^\circ$ *if camera moves backward,*

3) $|\|\overset{\sim}{\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}}\| - \|\overrightarrow{\boldsymbol{x}'_{k-1}\boldsymbol{x}^c_k}\|| < \epsilon_e$.

Otherwise, the MB belongs to moving objects: $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^c_k \in \mathcal{C}^{k\to k-1}_d$.

In Figure 6.2a, the MBs on building facades are labeled as stationary with red arrows whereas the MBs on the vehicle are labeled as moving.

6.4.3 Initial Plane Extraction and Labeling

With the labeled MB correspondences, we are able to extract planar regions. Since MBs in the plane at infinity π_∞ have very low signal-to-noise ratio for camera translation estimation, they should be removed before plane extraction for better accuracy. Denote the set of correspondences in π_∞ as $\mathcal{C}_\infty$,

$$\mathcal{C}^{k\to k-1}_\infty := \{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^c_k : \|\boldsymbol{x}'_{k-1} - \boldsymbol{x}^c_k\| < \epsilon_m, \boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^c_k \in \mathcal{C}^{k\to k-1}_s\} \tag{6.7}$$

where ϵ_m is the motion threshold. Figure 6.2a shows the detected π_∞ in blue arrows.

On the rest of correspondences $\mathcal{C}^{k\to k-1} \setminus \mathcal{C}^{k\to k-1}_\infty$, RANSAC is applied iteratively to extract all possible planes. To extract one plane, four correspondences are sampled, and an homography H is obtained using normalized direct linear transformation (p. 109 in [129]). Then, all correspondence resulting in an error below a given threshold: $\|\boldsymbol{x}_{k-1} - \lambda H \boldsymbol{x}^c_k\| < \epsilon_h$, is labeled as an inlier to the plane. In each RANSAC iteration, one largest plane is extracted, and its inliers are removed before next RANSAC iteration. This iterative RANSAC procedure can be replaced by J-linkage [152] if needed.

Then a set of planes, $\mathit{\Pi}^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}^k_i, i \in \mathcal{I}\}$ is initially constructed from $\{\Phi_k\}$. We use $\mathcal{I}$ to denote the index set for planes, and $i \in \mathcal{I}$ is the i-th plane. For each extracted plane $\tilde{\boldsymbol{\pi}}^k_i$, we denote its corresponding MV set as $\mathcal{C}^{k\to k-1}_{\pi,i}$. Thus, $\bigcup\limits_{i\in\mathcal{I}} \mathcal{C}^{k\to k-1}_{\pi,i} \subseteq \mathcal{C}^{k\to k-1} \setminus \mathcal{C}^{k\to k-1}_\infty$.

To perform tracking and improve plane estimation, all the planes need to be labeled either stationary or moving. With the MB labeling result $\mathcal{C}_s^{k\to k-1}$ and $\mathcal{C}_d^{k\to k-1}$, the plane labeling is designed by a majority voting of labeled MBs:

Definition 8 (Plane Labeling)**.** *A plane $\tilde{\boldsymbol{\pi}}_i^k \in \mathit{\Pi}^{k\to k-1}$ and its corresponding MV set $\mathcal{C}_{\pi,i}^{k\to k-1}$ are labeled as stationary $\tilde{\boldsymbol{\pi}}_{i,s}^k$ and $\mathcal{C}_{\pi,i,s}^{k\to k-1}$, respectively, if $|\mathcal{C}_{\pi,i}^{k\to k-1} \bigcap \mathcal{C}_s^{k\to k-1}| > |\mathcal{C}_{\pi,i}^{k\to k-1} \bigcap \mathcal{C}_d^{k\to k-1}|$. Otherwise, they are labeled as moving objects, $\tilde{\boldsymbol{\pi}}_{i,d}^k$ and $\mathcal{C}_{\pi,i,d}^{k\to k-1}$, respectively.*

After the labeling step, the set of all planes $\mathit{\Pi}^{k\to k-1}$ is partitioned into

$$\mathit{\Pi}^{k\to k-1} = \mathit{\Pi}_s^{k\to k-1} \bigcup \mathit{\Pi}_d^{k\to k-1}, \tag{6.8}$$

where $\mathit{\Pi}_s^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}_{i,s}^k\}$ is the set of stationary planes and $\mathit{\Pi}_d^{k\to k-1} = \{\tilde{\boldsymbol{\pi}}_{i,d}^k\}$ denotes the set of moving planes.

6.4.4 Plane Re-estimation and Observation Extraction

With the labeled planes, we can refine all estimations and prepare observations for EKFs. We start with the stationary scene and the camera motion. For a stationary plane $\tilde{\boldsymbol{\pi}}_{i,s}^k$, the correspondences $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}_k^c \in \mathcal{C}_{\pi,i,s}^{k\to k-1}$ conform to homography relation:

$$\boldsymbol{x}_{k-1} = H_i^{k\to k-1}\boldsymbol{x}_k^c = K(R^{k\to k-1})^{-1}[I_{3\times 3} + \boldsymbol{t}^{k\to k-1}(\tilde{\boldsymbol{\pi}}_{i,s}^k)^{\mathsf{T}}]K^{-1}\boldsymbol{x}_k^c, \tag{6.9}$$

where $H_i^{k\to k-1}$ is the homography matrix introduced by the plane, $I_{3\times 3}$ is a 3-dimensional identity matrix. Therefore, for the stationary scene, the observations of relative camera motion and stationary plane equations can be estimated by minimizing the total errors of fundamental relationship in all stationary correspondences and homography relationship in all planar correspondences:

$$\begin{aligned}\min_{R^{k\to k-1}, \boldsymbol{t}^{k\to k-1}, \tilde{\boldsymbol{\pi}}^{k}_{i,s} \in \mathit{\Pi}^{k\to k-1}_{s}} & \sum_{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^{c}_{k} \in \mathcal{C}^{k\to k-1}_{s}} \varepsilon_F(\boldsymbol{x}_{k-1}, \boldsymbol{x}^{c}_{k}, F^{k\to k-1}) \\ & + \sum_{i} \sum_{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^{c}_{k} \in \mathcal{C}^{k\to k-1}_{\pi,i,s}} \varepsilon_H(\boldsymbol{x}_{k-1}, \boldsymbol{x}^{c}_{k}, H^{k\to k-1}_{i}) \end{aligned} \tag{6.10}$$

where $F^{k\to k-1}$ and $H^{k\to k-1}_{i}$ are from (6.4) and (6.9), respectively. The resulting optimal $R^{k\to k-1}$, $\boldsymbol{t}^{k\to k-1}$ and $\tilde{\boldsymbol{\pi}}^{k}_{i,s}$'s are inputs to the stationary EKF in the next section.

For a moving plane $\tilde{\boldsymbol{\pi}}^{k}_{i,d}$, denote its translation as $\boldsymbol{t}_d$. If we back shift the plane by $-\boldsymbol{t}_d$, then a homography relationship can be established for $\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^{c}_{k} \in \mathcal{C}^{k\to k-1}_{\pi,i,d}$,

$$H^{k\to k-1}_{i} = K(R^{k\to k-1})^{-1}[I_{3\times 3} + (\boldsymbol{t}^{k\to k-1} - \boldsymbol{t}^{k\to k-1}_{i,d})(\tilde{\boldsymbol{\pi}}^{k}_{i,d})^{\mathsf{T}}]K^{-1}, \tag{6.11}$$

Therefore, a moving plane is estimated by minimizing the following,

$$\min_{\tilde{\boldsymbol{\pi}}^{k}_{i,d}, \boldsymbol{t}^{k\to k-1}_{i,d}} \sum_{\boldsymbol{x}_{k-1} \leftrightarrow \boldsymbol{x}^{c}_{k} \in \mathcal{C}^{k\to k-1}_{\pi,i,d}} \varepsilon_H(\boldsymbol{x}_{k-1}, \boldsymbol{x}^{c}_{k}, H^{k\to k-1}_{i}) \tag{6.12}$$

where $H^{k\to k-1}_{i}$ is from (6.11) with the estimated camera motion from (6.10). The resulting optimal plane equations and translations are inputs to the individual moving object filters later.

6.5 EKF-based Localization and Tracking

With the planes and camera motions extracted for adjacent key frame pairs, we can feed them as observations to EKFs for global robot localization, stationary plane mapping, and moving object tracking. As Figure 6.1 shows, the robot localization and stationary plane mapping are handled by one single EKF below.

6.5.1 Camera Localization and Static Scene Mapping

Based on stationary planes, this part is similar to the traditional visual SLAM problem. Following an EKF framework, we define the state vector $\boldsymbol{\mu}_k$ for the EKF filter as follows:

$$\boldsymbol{\mu}_{s,k} = [..., \tilde{\boldsymbol{\pi}}_{i,s,k}^{\mathsf{T}}, ..., \boldsymbol{r}_k^{\mathsf{T}}, \boldsymbol{t}_k^{\mathsf{T}}, \dot{\boldsymbol{r}}_k^{\mathsf{T}}, \dot{\boldsymbol{t}}_k^{\mathsf{T}}]^{\mathsf{T}}, \tag{6.13}$$

which includes the plane equations in $\{W\}$, the y-x-z Euler angles $\boldsymbol{r}_k$ for camera rotation from $\{W\}$ to $\{\Phi_k\}$, the camera location $\boldsymbol{t}_k$ in $\{W\}$, camera motion velocity $\dot{\boldsymbol{t}}_k$ in $\{W\}$, and the angular velocity of the camera $\dot{\boldsymbol{r}}_k$ in $\{\Phi_k\}$.

We assume the camera motion follows a constant linear velocity and a constant angular velocity. Therefore, the transition of state can be described as:

$$\boldsymbol{\mu}_{s,k} = \begin{bmatrix} ... \\ \tilde{\boldsymbol{\pi}}_{i,s,k} \\ ... \\ \boldsymbol{r}_k \\ \boldsymbol{t}_k \\ \dot{\boldsymbol{r}}_k \\ \dot{\boldsymbol{t}}_k \end{bmatrix} = \begin{bmatrix} ... \\ \tilde{\boldsymbol{\pi}}_{i,s,k-1} \\ ... \\ r(R(\boldsymbol{r_{k-1}})R(\dot{\boldsymbol{r}}_{k-1}\tau)) \\ \boldsymbol{t}_{k-1} + \dot{\boldsymbol{t}}_{k-1}\tau \\ \dot{\boldsymbol{r}}_{k-1} \\ \dot{\boldsymbol{t}}_{k-1} \end{bmatrix} \tag{6.14}$$

where τ is the time interval, $R(\boldsymbol{r})$ denotes the y-x-z Euler rotation matrix defined by $\boldsymbol{r}$, and $r(\cdot)$ is the decomposition of a rotation matrix to Euler angles.

The observations $\boldsymbol{z}_{s,k}$ to stationary filter include the plane equations in $\{\Phi_k\}$ and camera motion between $\{\Phi_{k-1}\}$ and $\{\Phi_k\}$:

$$\boldsymbol{z}_{s,k} = [..., (\tilde{\boldsymbol{\pi}}_{i,s}^{k})^{\mathsf{T}}, ..., (\boldsymbol{r}^{k\to k-1})^{\mathsf{T}}, (\boldsymbol{t}^{k\to k-1})^{\mathsf{T}}]^{\mathsf{T}}, \tag{6.15}$$

where $\boldsymbol{r}^{k\to k-1}$ is the quaternion of rotation matrix $R^{k\to k-1}$.

The observations are all in local coordinate system, and the observation function provides the transform to $\{W\}$ as follows:

$$\boldsymbol{z}_{s,k} = h(\boldsymbol{\mu}_{s,k}) = \begin{bmatrix} \ldots \\ R(\boldsymbol{r}_k)^{-1}\tilde{\boldsymbol{\pi}}_{i,s,k}/(1+\tilde{\boldsymbol{\pi}}_{i,s,k}^{\mathsf{T}}\boldsymbol{t}_k) \\ \ldots \\ r(-\dot{\boldsymbol{r}}_k\tau) \\ -R(\boldsymbol{r}_k)^{-1}\cdot(\dot{\boldsymbol{t}}_k\tau) \end{bmatrix}. \tag{6.16}$$

Note that the variables refer to the planes lost in current frame are deleted before update.

6.5.2 Moving Object Tracking

Similarly, this step is also handled using EKF (the bottom part of Figure 6.1). Moving objects are considered to move independently w.r.t to the camera and each other. We employ one EKF to track each moving object individually. In each EKF, one global plane equation and one velocity vector are tracked. Here, we assume the motion of moving plane follows a constant linear velocity in $\{W\}$ without rotation, which is usually true for pedestrians or vehicles appearing in the camera view for a short period of time. The state vector for a single moving plane filter becomes

$$\boldsymbol{\mu}_{i,d,k} = [\tilde{\boldsymbol{\pi}}_{i,d,k}^{\mathsf{T}}, \boldsymbol{v}_{i,d,k}^{\mathsf{T}}]^{\mathsf{T}}, \tag{6.17}$$

where $\boldsymbol{v}_{i,d,k}$ is the velocity of the i-th object in $\{W\}$. The state transition for the moving object i is straightforward:

$$\begin{cases} \tilde{\boldsymbol{\pi}}_{i,d,k} = \tilde{\boldsymbol{\pi}}_{i,d,k-1}/(1 - \tilde{\boldsymbol{\pi}}_{i,d,k-1}^{\mathsf{T}} \boldsymbol{v}_{i,d,k-1}\tau) \\ \boldsymbol{v}_{i,d,k} = \boldsymbol{v}_{i,d,k-1} \end{cases} . \tag{6.18}$$

The observations for the moving object filters are the estimated plane equations in $\{\Phi_k\}$, and the observation function is the transform between coordinate systems given the camera rotation and translation:

$$\boldsymbol{z}_{i,d,k} = [(\tilde{\boldsymbol{\pi}}_{i,d}^{k})^{\mathsf{T}}, (\boldsymbol{t}_{i,d}^{k \to k-1})^{\mathsf{T}}]^{\mathsf{T}} = \begin{bmatrix} R(\boldsymbol{r}_k)^{-1} \tilde{\boldsymbol{\pi}}_{i,d,k}/(1 + \tilde{\boldsymbol{\pi}}_{i,d,k}^{\mathsf{T}} \boldsymbol{t}_k) \\ -\tau R(\boldsymbol{r}_k)^{-1} \boldsymbol{v}_{i,d,k} \end{bmatrix} . \tag{6.19}$$

6.5.3 Plane Management

Apart from removal of planes that are no longer in the sight from the corresponding EKFs, plane labels are not permanent as a moving object may come to a stop or a parked vehicle may start moving. Since each plane has a stationary/moving label, plane label exchange happens when the label of an existing plane is not consistent with the outcome of the EKF. A moving plane's label will also be changed to stationary if its velocity is close to zero. When a plane changes its label, the corresponding state variables are moved from previous EKF filter to a new EKF, with an initialized velocity if necessary. For each newly discovered plane, its parameters are added into the corresponding EKF according to its label.

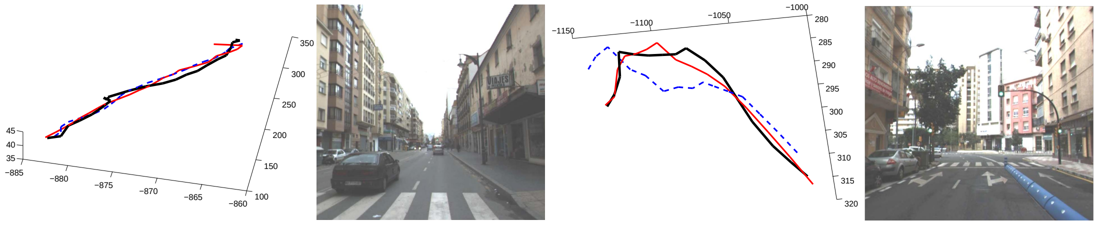
6.6 Experiments

We have implemented the proposed system using C/C++ in Cygwin environment under Microsoft Windows 7. To test the performance of the method, evaluation is conducted in the following three aspects: the localization error, the stationary plane estimation error, and the detection of moving planes.

6.6.1 Localization Evaluation

6.6.1.1 Dataset

We perform the evaluation using the Màlaga urban dataset [153] which provides stereo videos from vehicle driving in a dense urban area. The video frame rate is 20 fps. Images with a resolution of 1024×768 are rectified and the intrinsic camera matrix after rectification is provided. Ground truth data is collected using multiple sensors including GPS, IMU, and laser range finder. Since we assume the scene is quasi-rectilinear with many static planes, two typical urban scenes from the data set are used in the experiment. Since our method is monocular, we only use the images from the left camera in the dataset. Sample thumbnails of frames in the experiment are shown in Figure 6.3. The lengths (i.e. travel distance) of the two sequences are provided in Table 6.1.



(a) (b) (c) (d)

Figure 6.3: Trajectories and sample frame thumbnails. (a) and (c) are the camera trajectories in the two sequences. Black lines are the GPS ground truth, red solid lines are the estimated trajectories using our method and the blue dashed lines are trajectories estimated using [1]. (b) and (d) are the sample image frames in the two sequences.

6.6.1.2 Metric

The localization result is compared with GPS data. The GPS data is sampled once per second, and the image time stamps are aligned according to the GPS clock. The errors are measured using the absolute trajectory error (ATE) [1]. We define the GPS coordinate system by $\{G\}$ and the camera position in $\{G\}$ as $\hat{\boldsymbol{t}}_k^G$. For the estimated camera position $\boldsymbol{t}_k$ in $\{W\}$, a similarity transformation (rotation $R^{W \to G}$, translation $\boldsymbol{t}^{W \to G}$ and scale s) is applied to transform the position to the GPS coordinate $\boldsymbol{t}_k^G = sR^{W \to G}\boldsymbol{t}_k + \boldsymbol{t}^{W \to G}$. The rotation, translation and scale are obtained via a non-linear optimization that minimizes the total error between the GPS data $\hat{\boldsymbol{t}}_k^G$ and the transformed estimation result $\boldsymbol{t}_k^G$. Therefore, the ATE for a frame k is defined as $e_k = \|\boldsymbol{t}_k^G - \hat{\boldsymbol{t}}_k^G\|$.

6.6.1.3 Comparison

We compare our result with the popular 1-Point EKF [1] since both methods are EKF-based. The 1-point EKF [1] approach uses feature points as landmarks. Their system is tested under long distance trajectories with robust performance. The code for 1-Point EKF is acquired from the authors' website and is directly run in Matlab on our testing dataset. Table 6.1 shows the mean and maximum ATE for each sequence for both methods. The results show that the mean ATEs of our method are below 3.5 meters for both sequences and are below 3% of the overall trajectory length, which is comparable to [1]. In the first sequence, the vehicle travels on a mostly straight road, with occasional lane changes. In this case, our method and [1] perform similar, with [1] slightly better. In the second sequence, the vehicle starts from straight driving and experiences curved road later. In this case, our method outperforms [1] over 5 meters in average. This experiment confirms that MV-based featureless navigation method is feasible.

Table 6.1: Localization results using the màlaga dataset

	length (m)	#frames	method	mean ATE	max ATE	% over distance
seq 1	201.08	497	Our method	2.87m	6.33m	1.43%
			1-Point EKF	1.99m	3.67m	0.99%
	length (m)	#frames	method	mean ATE	max ATE	% over distance
seq 2	133.76	318	Our method	3.38m	4.99m	2.53%
			1-Point EKF	9.08m	12.30m	6.80%

6.6.2 Stationary Plane Estimation

To evaluate plane mapping accuracy, we compare our method with our previous work [151] which is referred as SLAPSE method since it only performs localization and plane mapping without ability of tracking moving objects. We use the dataset from [151] for comparison where ground truth is computed by points measured using a laser distance measurer with ±1 mm accuracy. The reason that we do not use the Màlaga urban dataset here is because there is no ground truth data for planes. Similar to [151], we only consider the planes that appear in more than 3 continuous frames. The same error functions in [151] for plane depth and angles are used:

$$\epsilon_d = \frac{1}{\sum_i N_i}\sum_i\sum_k |d^k_{i,k} - \hat{d}^k_{i,k}|, \text{ and } \epsilon_n = \frac{1}{\sum_i N_i}\sum_i\sum_k |\arccos((\boldsymbol{n}^k_{i,k})^\mathsf{T} \cdot \hat{\boldsymbol{n}}^k_{i,k})|, \quad (6.20)$$

where N_i is the number of frames plane i appears, and $\hat{}$ stands for the ground truth. The number of planes extracted in the site and the estimation errors are shown in Table 6.2. The comparison results show our method improves the estimation of scene planes in both depth and orientation accuracy.

6.6.3 Moving Object Detection

To evaluate the performance of moving object detection, the test is focused on the plane labeling algorithm as EKF-based tracking performance is determined by the labeling

Table 6.2: Static plane estimation results

method	# planes	ϵ_d (m)	ϵ_n (degs.)
Our method	5	0.55	6.80
SLAPSE	5	0.61	7.07

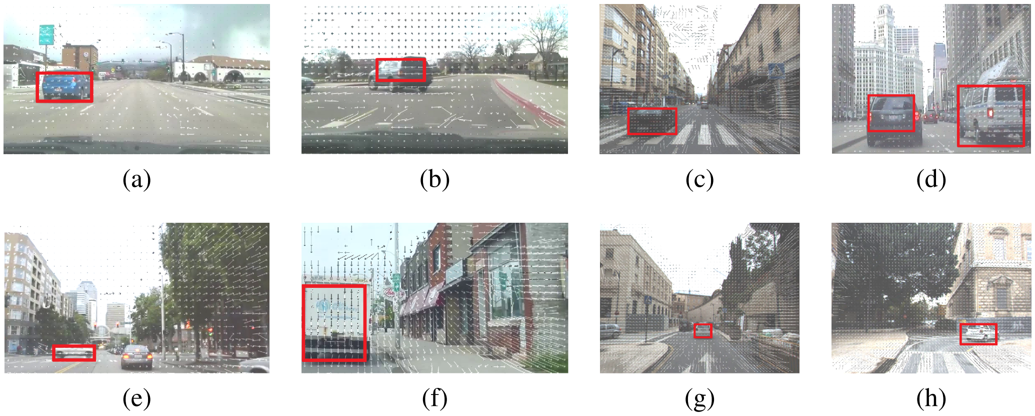

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.4: Detected moving objects are highlighted with red rectangles.

correctness. A dataset of 64 video clips are manually collected from the Internet, such as YouTube. All the video clips are recorded by cameras mounted on vehicles driving in urban environments. The frame rates vary between 23 and 30 fps, and the image resolution is between 640×360 and 1024×768. From all videos, there are a total of 88 moving vehicles that are manually identified as a ground truth. Note that the vehicles parking at red light or curbside are not labeled as moving objects, and the vehicles that are very far are not labeled because they are not objects of interest for collision avoidance.

Then the plane extraction and labeling method in Section 6.4 is applied to extract stationary and moving planes. Among 88 labeled moving objects, 85 are detected and labeled as moving planes, and the detection rate is 96.6%. Among the 3 failure cases, 2 cases are caused by lack of correct MVs on the vehicles. This situation happens when the vehicle is too texture-less and has a color either similar to the ground or with large saturation.

Another 1 case happens because the vehicle is relatively stationary to the camera, thus the MVs on it are not distinguishable from those on the infinite plane. The right most vehicle in Figure 6.2b shows an example of this situation. Actually, due to the zero relative speed, that vehicle is not a concern for collision avoidance purpose.

Figure 6.4 shows some examples of the detected moving planes in a bounding box. The detection of moving object helps to separate outliers and wrong MVs that influence the static localization and mapping results.

6.7 Conclusions and Future Work

We presented a new algorithm that is capable of performing SLAM task and obstacle tracking using MVs as inputs. This algorithm simultaneously localizes the robot, establishes scene understanding through planar surface extraction, and tracks moving objects. To achieve this, we first extracted planes from MVs and their corresponding pixel MBs. We labeled MBs as either stationary or moving using geometric constraints and properties of plane-induced homographies. Similarly, planes were also labeled as either stationary or moving using an MB voting process. This allows us to establish planes as observations for extended Kalman filters (EKFs) for both stationary scene mapping and moving object tracking. We implemented the proposed method and compared it with the state-of-the-art 1-point EKF. The results showed that the proposed method achieved similar localization accuracy. However, our method can directly provide plane-based rectilinear scene structure, which is a higher level of scene understanding, and is capable of detection moving obstacles at a true positive rate of 96.6%.

In the future, we plan to adopt a local bundle adjustment approach to further improve localization accuracy. We will combine MVs with appearance data to establish higher level scene mapping. Fusing with other sensors such as depth or inertial sensors is also under consideration.

7. CONCLUSION AND FUTURE WORK

In this dissertation, we explored motion signatures and algorithms for problems in two applications: object recognition and robot navigation.

7.1 Conclusion: Object Recognition

Using bird species recognition as an application for object recognition, we proposed two systems with different motion signatures. In the first system, the periodicity of the wingbeat motion is studied for species classification. We established Kinematics models of bird wings, and proved the consistency of periodicity between the 3D model and its 2D projection. Time series of salient extremities on bird images are extracted, and the wingbeat frequency is obtained via frequency analysis on the time series. Then, the species classification is performed via wingbeat frequency based likelihood ratios.

In the second system, an interacting multi-model Kalman filter approach is designed to capture the complex motion of birds during the fight. The system tracks both the body motion and the wing motion of the bird, and deals with different background conditions. The two systems are tested in physical experiments, and results show false positive rates of around 20% with low false negative rates.

7.2 Conclusion: Robot Navigation

We also explored motion signatures for vision-based robot navigation. We propose to use MVs encoded in MPEG videos for localization and mapping. To reduce the noise level in MVs, we performed error analysis of the MVs, and merged MVs across frames. To further reduce noises, we proposed to eliminate MVs in far field, and use homography fitting to extract planes as landmarks. The algorithm performed localization with a relative trajectory error below 5.1% in physical experiments.

Since MVs provide not only the motion of the stationary environment, but also that on moving objects. We further proposed to perform localization and obstacle avoidance in a single framework via simultaneous localization, mapping and moving object tracking. Methods are designed to separate MVs and the extracted planes into stationary and moving groups. Multiple Kalman filters are used parallel to track objects in different groups. Physical experiments with public datasets show a mean absolute localization error below 3.5 meters and a moving object detection rate at 96.6%.

7.3 Discussion and Future Work

We discuss the future work of this dissertation from the following perspectives:

7.3.1 Species Recognition for Bird Flock

For the bird species recognition application, we discussed the species recognition of individual birds in video sequences. In future work, it is interesting to study the simultaneous tracking and recognition of a flock of birds, such as the example in Figure 7.1. In this case, multiple object tracking algorithms are necessary. Motion signatures can be key features for tracking similar-looking objects under different moving trajectories.

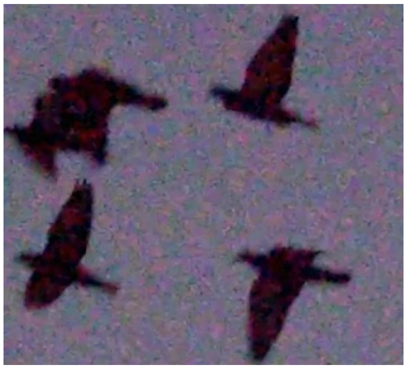


Figure 7.1: An example of bird flocks and occlusions.

7.3.2 Motion-assisted Segmentation of Deformable Objects

In the situation of occlusion, such as Figure 7.1, the segmentation of individual birds needs further development. If proper motion model is designed for represent the motion patterns on different part of the bird body, it would help to detect occlusion, and provide better segmentation of the deformable object.

7.3.3 Motion Information from Different Video Compressions

In the robot navigation application, we explored using the motion vectors in MPEG-2 videos to solve SLAM problem. However, there are other video compression formats that also encode motion correspondences in the environment. For example, in MPEG-4 format, more MVs are established with better accuracy for each macro block, and the block size is also adaptive to different scenes. So, it might provide better support for landmark estimation. In the SLAM problem, we mostly care about the accuracy of individual feature, the spatial distribution of the features and the overall computation load for mobile robots. Therefore, it would be interesting to compare different video compression formats, and decide a trade off between the three factors for different application scenarios.

7.3.4 MVs for Dense Reconstruction in SLAM

In traditional SLAM algorithms where complex image features are used, they can only sample a sparse set of image features to construct the map, because of the computation load in feature extraction. Therefore, the generated 3D map only has sparse landmarks to capture the general structure of the scene, like Figure 7.2a shows. Taking advantage of the readily available MVs all over the images, we save computation in feature extraction, and can trade it for a dense reconstruction of the scene under limited power. Figure 7.2b shows an initial attempt to use the MVs in extracted planar regions to reconstruct the building surface.

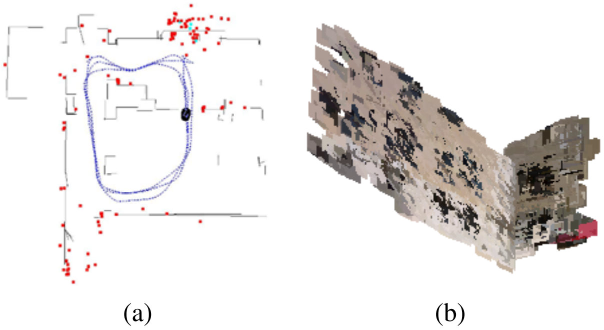
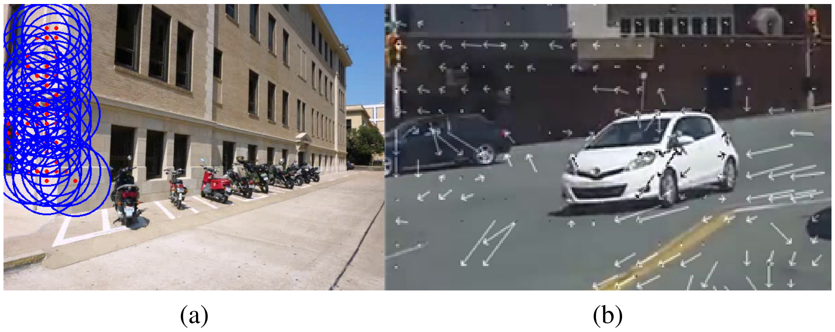


Figure 7.2: (a) An example of sparse landmarks [2]. (b) An example of dense reconstruction using MVs.

7.3.5 SLAM with Environment Models

For most SLAM works, the structure of the environment is unknown. But in urban environments, the exterior structures of the objects (like buildings, vehicles, and signs) usually follow certain patterns. For example, the exterior of a building usually contains vertical planes that perpendicular to each other, and the traffic signs are usually perpendicular to the ground. These prior knowledges can guide the searching of features in SLAM. For example, in Figure 7.3a, we extract and estimate a planar region on one side of the building, but we miss the planar regions on the other side. With the prior knowledge of building structure, it can assist us to actively search for planes in the perpendicular direction, and result in more extractions. Therefore, the reconstruction could reveal more structures and approximate the scene with better accuracy. As a mutual benefit, the reconstruction can then help to update the given building information. Moreover, for some buildings, the computer-aided design (CAD) data is widely available. This data provides value information of the absolute scale of the object, and can largely benefit the scale-drifting problem in monocular SLAM.



(a) (b)

Figure 7.3: (a) An example of plane extraction. (b) An example of fail detected plane.

7.3.6 SLAM with Motion and Appearance Information

When we discussed the failure cases, such as the vehicle in Figure 7.3b, in Section 6.6, one reason is the color of the moving vehicle is too bright or too dark, and no correct MVs exist on it. Due to the lack of MVs, we cannot detect it using motion signatures. In this case, it is interesting to consider combining the appearance information with MVs in the extraction process. Moreover, the appearance information can help to separate MVs on different building facades and objects. Thus, the segmentation of planar regions and the reconstruct of the scene can be more accurate.

7.3.7 Motion Representation from Theoretic Perspective

As one direction of future study, more high-level representations of motion information need to be explored. For example, in this dissertation, we mainly discussed motion signatures for individual objects. However, for groups of objects, their motions are often correlated and affected by each other. For example, for a crowd of vehicles on the road, they usually move in similar velocities, and the lane changing of one vehicle may influence the movement of vehicles around it. It is interesting to consider the representation for motion correlation between different objects or regions. This can assist the study of groups of objects, as well as the understanding of complex scene structure.

REFERENCES

[1] J. Civera, O. Grasa, A. Davison, and J. Montiel. 1-point RANSAC for extended Kalman filtering: application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, 2010.

[2] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman. A framework for vision based bearing only 3D SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1944–1950, Orlando, FL, 2006.

[3] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *ACM Computing Surveys*, 38(4):1–45, 2006.

[4] C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[5] K. Rangarajan and M. Shah. Establishing motion correspondence. *CVGIP: Image Understanding*, 54(1):56–73, 1991.

[6] I. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.

[7] A. Straw, K. Branson, T. Neumann, and M. Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of the Royal Society Interface*, 8(56):395–409, 2011.

[8] D. Tweed and A. Calway. Tracking many objects using subordinated condensation. In *British Machine Vision Conference*, pages 1–10, Cardiff, UK, 2002.

[9] R. Tillett, C. Onyango, and J. Marchant. Using model-based image processing to track animal movements. *Computers and Electronics in Agriculture*, 17(2):249–261, 1997.

[10] T. Burghardt and J. Ćalić. Analysing animal behaviour in wildlife videos using face detection and tracking. *IEE Proceedings-Vision, Image and Signal Processing*, 153(3):305–312, 2006.

[11] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.

[12] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753, Miami, FL, 2009.

[13] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[15] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[16] M. Dunn, J. Billingsley, and N. Finch. Machine vision classification of animals. In *Annual Conference on Mechatronics and Machine Vision in Practice*, pages 157–163, Baldock, UK, 2003.

[17] B. Horn and B. Schunck. Determining optical flow. In *Technical Symposium East*, pages 319–331, Washington, DC, 1981.

[18] B. Lucas and T. Kanade. An interative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, 1981.

[19] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr. Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005.

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36, Prague, Czech Republic, 2004.

[21] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[22] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *European Conference on Computer Vision*, pages 582–595, Crete, Greece, 2010.

[23] T. Muller, J. Rannacher, C. Rabe, and U. Franke. Feature and depth supported modified total variation optical flow for 3D motion field estimation in real scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1193–1200, Colorado Springs, CO, 2011.

[24] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[25] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[26] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: action recognition through the motion analysis of tracked features. In *IEEE International Conference*

on Computer Vision Workshops, pages 514–521, Kyoto, Japan, 2009.

[27] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[28] C. Liu, A. Torralba, W. Freeman, F. Durand, and E. Adelson. Motion magnification. *ACM Transactions on Graphics*, 24(3):519–526, 2005.

[29] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008.

[30] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008.

[31] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, Beijing, China, 2005.

[32] J. Davis, A. Bobick, and W. Richards. Categorical representation and recognition of oscillatory motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, Hilton Head Island, SC, 2000.

[33] A. Briassouli and N. Ahuja. Fusion of frequency and spatial domain information for motion analysis. In *International Conference on Pattern Recognition*, pages 175–178, Cambridge, UK, 2004.

[34] F. Knowlton, P. Martin, and J. Haug. A telemetric monitor for determining animal activity. *The Journal of Wildlife Management*, 32(4):943–948, 1968.

[35] W. Cochran, D. Warner, J. Tester, and V. Kuechle. Automatic radio-tracking system for monitoring animal movements. *BioScience*, 15(2):98–100, 1965.

[36] R. Moen, J. Pastor, Y. Cohen, and C. Schwartz. Effects of moose movement and habitat use on GPS collar performance. *The Journal of Wildlife Management*, 60(3):659–668, 1996.

[37] ARGOS System, 2012. http://www.argos-system.org/.

[38] W. Seegar, P. Cutchis, M. Fuller, J. Surer, V. Bhatnagar, and J. Wall. Fifteen years of satellite tracking development and application to wildlife research and conservation. *Johns Hopkins APL Technical Digest*, 17(4):401–411, 1996.

[39] H. Buechner, F. Craighead, J. Craighead, and C. Cote. Satellites for research on free-roaming animals. *BioScience*, 21(24):1201–1205, 1971.

[40] A. Rodgers. Tracking animals with GPS: the first 10 years. In *Conference on Tracking Animals with GPS*, pages 1–10, Aberdeen, UK, 2001.

[41] H. Sand, B. Zimmermann, P. Wabakken, H. Andrèn, and H. Pedersen. Using GPS technology and GIS cluster analyses to estimate kill rates in wolf-ungulate ecosystems. *Wildlife Society Bulletin*, 33(3):914–925, 2005.

[42] C. Brooks, C. Bonyongo, and S. Harris. Effects of global positioning system collar weight on zebra behavior and location error. *The Journal of Wildlife Management*, 72(2):527–534, 2008.

[43] A. Joshi, I. VishnuKanth, N. Samdaria, S. Bagla, and P. Ranjan. GPS-less animal tracking system. In *International Conference on Wireless Communication and Sensor Networks*, pages 120–125, Las Vegas, NV, 2008.

[44] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain. Animal behaviour understanding using wireless sensor networks. In *IEEE Conference on Local Computer Networks*, pages 607–614, Tampa, FL, 2006.

[45] B. Mcconnell, R. Beaton, E. Bryant, C. Hunter, P. Lovell, and A. Hall. Phoning home - a new GSM mobile phone telemetry system to collect mark-recapture data. *Marine Mammal Science*, 20(2):274–283, 2004.

[46] J. Sundell, I. Kojola, and I. Hanski. A new GPS-GSM-based method to study behavior of brown bears. *Wildlife Society Bulletin*, 34(2):446–450, 2006.

[47] W. Fiedler. New technologies for monitoring bird migration and behaviour. *Ringing & Migration*, 24(3):175–179, 2009.

[48] H. Dettki, G. Ericsson, and L. Edenius. Real-time moose tracking: an internet based mapping application using GPS/GSM-collars in Sweden. *Alces*, 40(1):13–21, 2004.

[49] C. Heckscher, S. Taylor, J. Fox, and V. Afanasyev. Veery (catharus fuscescens) wintering locations, migratory connectivity, and a revision of its winter range using geolocator technology. *The Auk*, 128(3):531–542, 2011.

[50] W. Conner and W. Masters. Infrared video viewing. *Science*, 199(4332):1004–1004, 1978.

[51] A. Weisenberger, B. Kross, S. Gleason, J. Goddard, S. Majewski, S. Meikle, M. Paulus, M. Pomper, V. Popov, M. Smith, B. Welch, and R. Wojcik. Development and testing of a restraint free small animal spect imaging system with infrared based motion tracking. In *IEEE Nuclear Science Symposium Conference Record*, pages 2090–2094, Portland, OR, 2003.

[52] C. Lemen and P. Freeman. Tracking mammals with fluorescent pigments: a new technique. *Journal of Mammalogy*, 66(1):134–136, 1985.

[53] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, MA, 2005.

[54] H. Strasdat, J. Montiel, and A. Davison. Visual SLAM: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[55] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[56] H. Strasdat, J. Montiel, and A. Davison. Scale drift-aware large scale monocular SLAM. In *Robotics: Science and Systems*, pages 5–12, Zaragoza, Spain, 2010.

[57] S. Frintrop and P. Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Transactions on Robotics*, 24(5):1054–1065, 2008.

[58] D. Schleicher, L. Bergasa, M. Ocaña, R. Barea, and E. López. Real-time hierarchical stereo visual SLAM in large-scale environments. *Robotics and Autonomous Systems*, 58(8):991–1002, 2010.

[59] B. Clipp, J. Lim, J. Frahm, and M. Pollefeys. Parallel, real-time visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3961–3968, Taipei, Taiwan, 2010.

[60] A. Gil, O. Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Machine Vision and Applications*, 21(6):905–920, 2010.

[61] D. Zou and P. Tan. CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):354–366, 2013.

[62] M. Kaess and F. Dellaert. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding*, 114(2):286–296, 2010.

[63] A. Rituerto, L. Puig, and J. Guerrero. Visual SLAM with an omnidirectional camera. In *International Conference on Pattern Recognition*, pages 348–351, Istanbul, Turkey, 2010.

[64] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2531–2538, Nice, France, 2008.

[65] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. In *European Conference on Computer Vision*, pages 445–461, Dublin, Ireland, 2000.

[66] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, Vila Moura, Portugal, 2012.

[67] E. Eade and T. Drummond. Scalable monocular SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476, New York City, NY, 2006.

[68] E. Eade and T. Drummond. Edge landmarks in monocular SLAM. In *British Machine Vision Conference*, pages 7–16, Edinburgh, UK, 2006.

[69] P. Smith, I. Reid, and A. Davison. Real-time monocular SLAM with straight lines. In *British Machine Vision Conference*, pages 17–26, Edinburgh, UK, 2006.

[70] T. Lemaire and S. Lacroix. Monocular-vision based SLAM using line segments. In *IEEE International Conference on Robotics and Automation*, pages 2791–2796,

Roma, Italy, 2007.

[71] W. Jeong and K. Lee. Visual SLAM with line and corner features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2570–2575, Beijing, China, 2006.

[72] D. Rao, S. Chung, and S. Hutchinson. CurveSLAM: an approach for vision-based navigation without point features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4198–4204, Vila Moura, Portuga, 2012.

[73] G. Zhang and I. Suh. Building a partial 3D line-based map using a monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1497–1502, Shanghai, China, 2011.

[74] G. Zhang and I. Suh. Sof-SLAM: segments-on-floor-based monocular SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2083–2088, Taipei, Taiwan, 2010.

[75] A. Flint, C. Mei, I. Reid, and D. Murray. Growing semantically meaningful models for visual SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 467–474, San Francisco, CA, 2010.

[76] A. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering higher level structure in visual SLAM. *IEEE Transactions on Robotics*, 24(5):980–990, 2008.

[77] A. Gee, D. Chekhlov, W. Mayol-Cuevas, and A. Calway. Discovering planes and collapsing the state space in visual SLAM. In *British Machine Vision Conference*, pages 1–10, University of Warwick, UK, 2007.

[78] T. Pietzsch. Planar features for visual SLAM. In *Annual German Conference on Advances in Artificial Intelligence*, pages 119–126, Kaiserslautern, Germany, 2008.

[79] H. Li, D. Song, Y. Lu, and J. Liu. A two-view based multilayer feature graph for robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 3580–3587, St. Paul, MN, 2012.

[80] Y. Lu, D. Song, and J. Yi. High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment. In *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014.

[81] T. Lee, S. Lim, S. Lee, S. An, and S. Oh. Indoor mapping using planes extracted from noisy RGB-D sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1727–1733, Vila Moura, Portugal, 2012.

[82] E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. González-Jiménez. Fast place recognition with plane-based maps. In *IEEE International Conference on Robotics and Automation*, pages 2719–2724, Karlsruhe, Germany, 2013.

[83] A. Trevor, J. Rogers, and H. Christensen. Planar surface SLAM with 3D and 2D sensors. In *IEEE International Conference on Robotics and Automation*, pages 3041–3048, St. Paul, MN, 2012.

[84] X. Ji and H. Liu. Advances in view-invariant human motion analysis: a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(1):13–24, 2010.

[85] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[86] E. Ribnick and N. Papanikolopoulos. Estimating 3D trajectories of periodic motions from stationary monocular views. In *European Conference on Computer Vision*, pages 546–559, Marseille, France, 2008.

[87] E. Ribnick and N. Papanikolopoulos. View-invariant analysis of periodic motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1903–1908, St. Louis, MO, 2009.

[88] E. Ribnick and N. Papanikolopoulos. 3D reconstruction of periodic motion from a single view. *International Journal of Computer Vision*, 90(1):28–44, 2010.

[89] Y. Ran, I. Weiss, Q. Zheng, and L. Davis. Pedestrian detection via periodic motion analysis. *International Journal of Computer Vision*, 71(2):143–160, 2007.

[90] S. Seitz and C. Dyer. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):231–251, 1997.

[91] P. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic motion detection for motion based recognition. *Pattern Recognition*, 27(12):1591–1603, 1994.

[92] I. Laptev, S. Belongie, P. Perez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *IEEE International Conference on Computer Vision*, pages 816–823, Beijing, China, 2005.

[93] A. Briassouli and N. Ahuja. Extraction and analysis of multiple periodic motions in video sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1244–1261, 2007.

[94] D. Ormoneit, M. Black, T. Hastie, and H. Kjellström. Representing cyclic human motion using functional analysis. *Image and Vision Computing*, 23(14):1264–1276, 2005.

[95] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[96] R. Polana and R. Nelson. Detection and recognition of periodic, nonrigid motion. *International Journal of Computer Vision*, 23(3):261–282, 1997.

[97] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.

[98] L. Gammaitoni, E. Menichella-Saetta, S. Santucci, and F. Marchesoni. Extraction of periodic signals from a noise background. *Physics Letters A*, 142(2):59–62, 1989.

[99] T. Yoshizawa, S. Hirobayashi, and T. Misawa. Noise reduction for periodic signals using high-resolution frequency analysis. *Journal on Audio, Speech, and Music Processing*, 2011(1):1–19, 2011.

[100] D. Song and K. Goldberg. Networked robotic cameras for collaborative observation of natural environments. In *International Symposium of Robotics Research*, pages 510–519, San Francisco, CA, 2005.

[101] D. Song, N. Qin, Y. Xu, C. Kim, D. Luneau, and K. Goldberg. System and algorithms for an autonomous observatory assisting the search for the ivory-billed woodpecker. In *IEEE International Conference on Automation Science and Engineering*, pages 200–205, Washington, DC, 2008.

[102] S. Faridani, B. Lee, S. Glasscock, J. Rappole, D. Song, and K. Goldberg. A networked telerobotic observatory for collaborative remote observation of avian activity and range change. In *International Federation of Automatic Control Workshop on Networked Robotics*, pages 56–61, Golden, CO, 2009.

[103] D. Song and Y. Xu. A low false negative filter for detecting rare bird species from short video segments using a probable observation data set-based EKF method. *IEEE Transactions on Image Processing*, 19(9):2321–2331, 2010.

[104] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[105] A. Jerri. The shannon sampling theorem - its various extensions and applications: a tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.

[106] C. Pennycuick. Wingbeat frequency of birds in steady cruising flight: new data and improved predictions. *Journal of Experimental Biology*, 199(7):1613–1618, 1996.

[107] C. Pennycuick. Predicting wingbeat frequency and wavelength of birds. *Journal of Experimental Biology*, 150(1):171–185, 1990.

[108] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection.* Wiley, Hoboken, NJ, 1987.

[109] P. Filzmoser, R. Garrett, and C. Reimann. Multivariate outlier detection in exploration geochemistry. *Computers & Geosciences*, 31(5):579–587, 2005.

[110] S. Lankton. Sparse field methods. Technical report, Georgia Institute of Technology, 2009.

[111] J. Dunn and J. Alderfer. *Field Guide to the Birds of Eastern North America*. National Geographic, Washington, DC, 2008.

[112] T. Liu, K. Kuykendoll, R. Rhew, and S. Jones. Avian wing geometry and kinematics. *AIAA Journal*, 44(5):954–963, 2006.

[113] J. Craig. *Introduction to robotics: mechanics and control.* Pearson/Prentice Hall, Upper Saddle River, NJ, 2005.

[114] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang. Sensor network for the monitoring of ecosystem: bird species recognition. In *International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 293–298, Melbourne, Australia, 2007.

[115] U. Nadimpalli, R. Price, S. Hall, and P. Bomma. A comparison of image processing techniques for bird recognition. *Biotechnology Progress*, 22(1):9–13, 2006.

[116] D. Ramanan and D. Forsyth. Using temporal coherence to build models of animals. In *IEEE International Conference on Computer Vision*, pages 338–345, Nice, France, 2003.

[117] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004.

[118] M. Pressigout and E. Marchand. Real-time 3D model-based tracking: combining edge and texture information. In *IEEE International Conference on Robotics and Automation*, pages 2726–2731, Orlando, FL, 2006.

[119] I. Mondragón, P. Campoy, C. Martinez, and M. Olivares-Méndez. 3D pose estimation based on planar object tracking for UAVs control. In *IEEE International Conference on Robotics and Automation*, pages 35–41, Anchorage, AK, 2010.

[120] M. Manz, T. Luettel, F. von Hundelshausen, and H. Wuensche. Monocular model-based 3D vehicle tracking for autonomous vehicles in unstructured environment. In *IEEE International Conference on Robotics and Automation*, pages 2465–2471, Shanghai, China, 2011.

[121] M. Tu, C. Huang, and L. Fu. Online 3D tracking of human arms with a single camera. In *IEEE International Conference on Robotics and Automation*, pages 1378–1383, St. Paul, MN, 2012.

[122] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

[123] J. Civera, A. Davison, and J. Montiel. Interacting multiple model monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 3704–3709, Pasadena, CA, 2008.

[124] Z. Jia, A. Balasuriya, and S. Challa. Vision based data fusion for autonomous vehicles target tracking using interacting multiple dynamic models. *Computer Vision and Image Understanding*, 109(1):1–21, 2008.

[125] P. Bauer and J. Bokor. Multi-mode extended kalman filter for aircraft attitude estimation. In *World Congress of the International Federation of Automatic Control*, pages 7244–7249, Milano, Italy, 2011.

[126] L. Hong, N. Cui, N. Bakich, and J. Layne. Multirate interacting multiple model particle filter for terrain-based ground target tracking. *IEE Proceedings-Control Theory and Applications*, 153(6):721–731, 2006.

[127] D. Song, N. Qin, Y. Xu, C. Kim, D. Luneau, and K. Goldberg. System and algorithms for an autonomous observatory assisting the search for the ivory-billed woodpecker. In *IEEE International Conference on Automation Science and Engineering*, pages 200–205, Washington, DC, 2008.

[128] W. Li and D. Song. Automatic bird species detection using periodicity of salient extremities. In *IEEE International Conference on Robotics and Automation*, pages 5775–5780, Karlsruhe, Germany, 2013.

[129] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, MA, 2003.

[130] K. Knight. *Mathematical Statistics*. Chapman and Hall, New York City, NY, 2000.

[131] C. Pennycuick. Speeds and wingbeat frequencies of migrating birds compared with calculated benchmarks. *Journal of Experimental Biology*, 204(19):3283–3294,

2001.

[132] R. Wang and T. Huang. Fast camera motion analysis in MPEG domain. In *International Conference on Image Processing*, pages 691–694, Kobe, Japan, 1999.

[133] L. Favalli, A. Mecocci, and F. Moschetti. Object tracking for retrieval applications in MPEG-2. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3):427–432, 2000.

[134] F. Vella, A. Castorina, M. Mancuso, and G. Messina. Digital image stabilization by adaptive block motion vectors filtering. *IEEE Transactions on Consumer Electronics*, 48(3):796–801, 2002.

[135] A. Argiles, J. Civera, and L. Montesano. Dense multi-planar scene estimation from a sparse set of images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4448–4454, San Francisco, CA, 2011.

[136] R. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1505, San Francisco, CA, 2010.

[137] D. Song, H. Lee, J. Yi, and A. Levandowski. Vision-based motion planning for an autonomous motorcycle on ill-structured roads. *Autonomous Robots*, 23(3):197–212, 2007.

[138] D. Song, H. Lee, and J. Yi. On the analysis of the depth error on the road plane for monocular vision-based robot navigation. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 301–315, Guanajuato, Mexico, 2008.

[139] J. Zhang and D. Song. On the error analysis of vertical line pair-based monocular visual odometry in urban area. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3486–3491, St. Louis, MO, 2009.

[140] J. Zhang and D. Song. Error aware monocular visual odometry using vertical line pairs for small robots in urban areas. In *AAAI Conference on Artificial Intelligence*, pages 2531–2538, Atlanta, Georgia, 2010.

[141] Y. Lu, Y. Song, D.and Xu, A. Perera, and S. Oh. Automatic building exterior mapping using multilayer feature graphs. In *IEEE International Conference on Automation Science and Engineering*, pages 162–167, Madison, WI, 2013.

[142] Vincent Rabaud. Structure from Motion Toolbox, 2013. http://vision.ucsd.edu/ vrabaud/toolbox/.

[143] S. Wangsiripitak and D. Murray. Avoiding moving outliers in visual SLAM by tracking moving objects. In *IEEE International Conference on Robotics and Automation*, pages 375–380, Kobe, Japan, 2009.

[144] Y. Wang, M. Lin, and R. Ju. Visual SLAM and moving object detection for a small-size humanoid robot. *International Journal of Advanced Robotic Systems*, 7(2):133–138, 2010.

[145] R. Venkatesh Babu and K. Ramakrishnan. Compressed domain motion segmentation for video object extraction. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3788–3791, Orlando, FL, 2002.

[146] T. Yokoyama, T. Iwasaki, and T. Watanabe. Motion vector based moving object detection and tracking in the MPEG compressed domain. In *International Workshop on Content-based Multimedia Indexing*, pages 201–206, Chania, Crete, 2009.

[147] S. Park and J. Lee. Object tracking in MPEG compressed video using mean-shift algorithm. In *Joint Conference of International Conference on Information, Communications and Signal Processing, and Pacific Rim Conference on Multimedia*, pages 748–752, Singapore, 2003.

[148] S. Denman, C. Fookes, and S. Sridharan. Improved simultaneous computation of motion detection and optical flow for object tracking. In *Digital Image Computing: Techniques and Applications*, pages 175–182, Melbourne, Australia, 2009.

[149] N. Ohnishi and A. Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recognition Letters*, 27(9):1009–1021, 2006.

[150] C. Braillon, C. Pradalier, J. Crowley, and C. Laugier. Real-time moving obstacle detection using optical flow models. In *IEEE Intelligent Vehicles Symposium*, pages 466–471, Tokyo, Japan, 2006.

[151] W. Li and D. Song. Toward featureless visual navigation: simultaneous localization and planar surface extraction using motion vectors in video streams. In *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014.

[152] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *European Conference on Computer Vision*, pages 537–547, Marseille, France, 2008.

[153] J. Blanco-Claraco, F. Moreno-Dueñas, and J. González-Jiménez. The Málaga urban dataset: high-rate stereo and LiDAR in a realistic urban scenario. *International Journal of Robotics Research*, 33(2):207–214, 2014.