From Motion to Human Activity Recognition

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Taha Alhersh
aus Irbid, Jordanien

Mannheim, 2021

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Prof. Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent: Prof. Dr. Samir Belhaouari, Hamad Bin Khalifa University, Qatar

Tag der mündlichen Prüfung: 26.03.2021

Acknowledgment

I would like to thank The Islamic Development Bank (IsDB) for awarding me the IDB Merit Scholarship Program for the first three years of my study.

I would also like to thank my supervisor, Heiner Stuckenschmidt, whose expertise was invaluable in producing this research work. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

Furthermore, I would like to thank my second supervisor, Samir Brahim Belhaouari, for all the support he provided me in terms of fruitful discussion and guidance. He was always there for my support.

In addition, my sincere thanks to my parents, Taleb and Shefa, for their continuous support to me during this journey. My brothers were integral components of this support, so big thanks, too, to you buddies.

Very special thanks goes to my wife, Heba, and my children, Mousa and Maria, for their patience, support, and understanding during my study.

Finally, I would like to thank special friends for their support: Abdalhamid Abdalrahmann, Hamada Al-Absi, Melisachew Vudage Chekol, Nabeel Alghail and Yaser Oulabi.

Abstract

Advancements in wearable technology have the potential to transform the quality of life, business, and the global economy. Body sensors can be used in human activity recognition, which has direct impact on various application domains such as surveillance systems, healthcare systems, robotics, and other physical and metrological applications.

Human activity recognition can be considered as a problem in both computer vision and pervasive computing. In this research, we started from a computer vision problem based on optical flow, and then introduced open issues in respect to existing techniques. We went on to study the relation between optical flow and human activity recognition, taking into consideration the effectiveness of using optical flow combined with other wearable sensors. As a result, feasible solutions have been presented to solve those problems, and then useful insights have been given for implementing corresponding techniques.

A comprehensive set of experiments and discussions were performed during the research. Firstly, we suggested an unsupervised optical flow fine-tuning that overcomes the need for a ground truth for training on the one hand and enhanced motion boundaries on the other.

Secondly, we provided theoretical justification for optical flow evaluation metrics. Moreover, we suggested novel optical flow performance metrics that have been evaluated alongside current metrics. Our empirical findings examined the performance of all metrics with regard to their sensitivity to change in motion between estimated optical flow and the ground truth.

Finally, we investigated methods regarding feature extraction for Inertial Measurement Units (IMUs) and visual data captured from wearable sensors, for instance, statistical features, local visual descriptors, and features extracted from deep learning. The features generated were tested for human activity recogni-

tion using Support Vector Machines and Recurrent Neural Network as the main recognition methods.

Zusammenfassung

Fortschritte im Bereich der Wearable Technology können die Lebensqualität und die globale Wirtschaft verändern. Körpersensoren können bei der Erkennung menschlicher Aktivitäten eingesetzt werden, was direkte Auswirkungen auf verschiedene Anwendungsbereiche wie Überwachungssysteme, Gesundheitssysteme, Robotik und andere physikalische und messtechnische Anwendungen hat.

Die Erkennung menschlicher Aktivitäten (Human Activity Recognition) kann als Problem in den Bereichen Computer Vision und Pervasive Computing betrachtet werden. In dieser Arbeit erforschen wir zunächst den Einsatz von Computer-Vision-Methoden in Form von Optical Flow, und gehen auf offene Fragen in Bezug auf bestehende Techniken ein. Daraufhin untersuchen wir im Kontext der Erkennung menschlicher Aktivitäten die Wirksamkeit der Verwendung von Optical Flow unter Berücksichtigung mehrerer tragbarer Sensoren. Dabei führen wir neuartige Lösungen und Methoden für die Erkennung menschlicher Aktivitäten ein, und erörtern für die Implementierung solcher Methoden wichtige Erkenntnisse und Beobachtungen.

In dieser Arbeit wurden eine umfassende Reihe von Experimenten und Diskussionen durchgeführt. Zunächst wurde eine Methode für das unüberwachte Fine-Tuning des Optical Flows vorgeschlagen. Diese überwindet auf der einen Seite die Voraussetzung einer Ground Truth, und verbessert dabei die Motion Boundaries des Optical Flows.

Zweitens haben wir eine theoretische Rechtfertigung für die Bewertungsmetriken des Optical Flows erörtert. Darüber hinaus haben wir neue Metriken vorgeschlagen, die zusammen mit den aktuellen Metriken empirisch ausgewertet und verglichen wurden. Dabei untersuchen wir die Leistung aller Metriken im Hinblick auf ihrer Empfindlichkeit gegenüber Bewegungsänderungen zwischen dem geschätzten Optical Flow und der Ground Truth.

Schließlich untersuchen wir für Inertial Measurement Units (IMUs) und visuelle Daten, die von tragbaren Sensoren erfasst werden, Methoden zur Extraktion von Features. Diese beinhalten z.B. statistische Merkmale, lokale visuelle Deskriptoren und Features welche auf Basis von Deep Learning extrahiert wurden. Die generierten Features wurden für die Erkennung menschlicher Aktivitäten getestet, wobei Support Vector Machines und Recurrent Neural Network als Haupterkennungsmethoden verwendet wurden.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The rapid development of ubiquitous mobile and sensor-rich devices has increased demand for human activity recognition (HAR). A wide range of applications can benefit from HAR, including mobile or ambient-assisted living, health support, human-computer interaction, video surveillance, industrial settings, smart homes, and rehabilitation. Low-cost wearable devices offering custom preferences in regard to size, weight, low power consumption, etc. are becoming increasingly available, which in turn is stimulating demand for - and hence the production of - more mobile wearable devices, in addition to embedded sensing for smart environments.

Many approaches have been adopted in the area of human activity recognition research, with vision based [140] and sensor based [139] being the most common. These approaches can be categorized into two main methods based on the design or technology used. One is machine learning methods, which include (but are not limited to) k-nearest neighbor (K-NN); decision trees (DT); support vector machine (SVM), and hidden Markov models (HMM). The other is neural network methods, which include (but are not limited to) artificial neural networks (ANN); convolutional neural networks (CNN); and recurrent neural networks (RNN) [66].

Vision-based human activity recognition approaches are based on the use of visual sensing technologies, such as video cameras, to monitor an actor's behavior and environmental changes. The sensor data generated takes the form of video sequences or digitized visual data. The approaches in this category exploit computer vision techniques, including feature extraction, structural modeling, movement segmentation, activity extraction, and movement tracking to analyze visual observations for pattern recognition [31].

Advancements in image representation and classification methods have also gained increasing attention. Ordinarily, literature on image representation methods follows research trajectories based on global and local representations. Early beginning research studies attempted to model whole images or silhouettes and represent human activities in a global manner. One approach is approximating the real physical motion projected onto the image plane. This approximation represents the obvious motion of each individual pixel on the image plane, which is called optical flow. The approach in [21] is another example of image descriptors which is considered to be a global representation when space-time shapes are generated. After that, space-time interest points (STIPs) has been emerged when [74] proposed to trigger extraordinary attention to the informative interest points to establish a new local representation. The other method for human representation is local representation, for instance, visual local descriptors such as histogram of optical flow (HOF) and histogram of oriented gradients (HOG) are vastly used or extended to 3D. With camera devices development, depth image-based representations have been emerged as new research topic and have drawn growing attention in recent years.

Moving from visual representation of visual data to recognizing human activity, machine learning methods are developing various classification techniques. Basically, many classification methods were designed for domains other than HAR. For example, the first time that hidden Markov model (HMM) and dynamic time warping (DTW) were used in speech recognition. Another example is using deep learning method firstly developed for classifying large amount images. In the domain of human activity recognition, many activity datasets are collected, shaping public and transparent benchmarks for comparing different classification approaches.

As mentioned above, the second approach used in human activity recognition is sensor-based HAR, which is based on using sensor network technologies. The data generated from monitoring systems based on sensor networks is basically time series, corresponding to the changes of state and/or various parameter measurements that are usually processed through data fusion, statistical analysis, or probabilistic methods and knowledge techniques for activity recognition.

In these methods, sensors can be attached to subjects with broad names like “wearable sensors” or “smartphones”. The term “wearable sensors” usually refers

to sensors that are placed directly or indirectly on the human body, generating signals when the wearers perform actions. Consequently, wearable sensors can monitor descriptive features of the subjects' physiological state or movement. They can be embedded into eyeglasses, shoes, belts, wristwatches, clothes, mobile devices, or directly positioned on the body. They can collect various types of information from subjects, such as position, movement, pulse, temperature, and skin conductance. For instance, inertial measurement units (IMUs) and radio frequency identification (RFID) tags are used to collect behavioral information about a subject. This type of approach is potent in recognizing subjects' physical movement, for example, physical exercises. Accelerometer sensors are probably the most frequently used wearable sensor for activity monitoring. They are particularly effective in monitoring activities that involve repetitive body motions such as walking, running, sitting, standing, and climbing stairs.

Human movement involves the use of one or more parts of the body [18] and can be distinguished at different levels of granularity. The terms "action" and "activity" as components of physical movement are mainly used in activity-recognition communities. In some scenarios, they are used interchangeably, while in others, they indicate the complexity of different behaviors and durations.

Simple ambulatory behavior performed by a single subject and typically lasting for a short duration of time is referred to as "action". Examples of actions include opening a bottle, closing a drawer, opening a fridge, putting eggs into a pan, etc. In contrast, complex behaviors consisting of a sequence of actions and/or interleaving or overlapping actions are denoted by the term "activities". These can be performed by one or more people interacting with each other in a structured manner. Activities are typically characterized by much longer temporal durations, such as making a cake or two people washing dishes. As one activity can contain only one action, there is no cutoff boundary between these two behavior categories. Nevertheless, this simple categorization provides a basic conceptualization and clarity for the discussions in this research.

From the aforementioned introduction, it can be seen that computer vision and pervasive computing domains are broad and interconnected and that the relation between the two can be complementary [7]. Furthermore, human activity starts from a movement of one or more parts of the human body, and this physical motion

can be approximated using optical flow estimation.

With these facts in mind, the current research started from optical flow as a computer-vision problem. We subsequently introduced some open issues in respect to existing techniques and optical flow evaluation metrics. Then, the relationship between optical flow and human activity was examined. In the process, in order to minimize the number of sensors for human activity recognition, we decided on the use of only one sensor: a first-person camera.

In the following text, we first provide the motivation for undertaking this research. After this, we define our problem statement and research questions. Next, we clarify the contribution and the outline of the thesis.

1.1 Motivation

Nowadays, human activity recognition has drawn a lot of attention in the field of computer science due to the increase demand from many domain applications, for instance in surveillance systems, healthcare systems, robotics and in other physical and metrological applications. Motion is considered as an important cue for human activity recognition. [29, 22, 69, 7] have provided a significant evidences in their research suggests that visual data can does a vital role in activity recognition. Among the main realms that deploy sensor data in general and in human activity recognition in particular are:

- **Security and Visual Surveillance**: Visual surveillance systems are designed as - and considered to be - modular systems. They contain functional modules, such as for motion detection, that estimate depth, track objects, and analyze objects' behavior [67]. For instance, optical flow can identify moving objects, and it is an effective method to subtract foreground and background [134]. Moreover, optical flow is used extensively for tracking in visual surveillance [135, 131].

- **Activity Recognition**: Optical flow is a useful parameter for activity recognition [72], since it is invariant to appearance, even in the lake of temporal coherence, and its accuracy at boundaries is important for activity recog-

nition [103]. Some of the most important application domains of human activity recognition:

  - **Active and assisted living applications for smart homes**: Enhancing the quality of life for elderly and disabled people is a crucial and principal objective, and modern technologies offer innovative ways to achieve this. Activity recognition techniques can be used to assist and monitor individuals in order to help secure their safety and well-being [39, 97, 95].
  - **Healthcare monitoring**: The quality of patients' lives has increased considerably thanks to the development of medical science and technology, and activity recognition has become a vital component of healthcare monitoring systems. For example, it can be used in human tracking, fall detection, security alarm, and cognitive-assistance systems [50, 48, 141, 70].

- **Robot Navigation**: The process of identifying the most suitable path between a robot's start and goal locations is considered a type of navigation [23]. The robot's speed and direction can be determined by calculating optical flow from image sequences it observes from the surrounding world. Optical flow can provide the robot with information about the unknown environment, making it one of the primary techniques used for mapless robotic navigation [41]. Optical flow is also used for obstacle detection and collision avoidance.

1.2 Problem Statement

Human activity recognition can be considered a computer vision and pervasive computing problem, with the intersection between the two based on the experimental setup of the experiments. This setup defines both the wearable sensors (e.g., inertial measurement sensors, global positioning system, biosensors, and cameras) to be used in the experiment and the problem(s) the research question(s) will solve. In the present research, we started from a computer vision problem based on optical flow and then introduced open issues in respect to existing techniques and

evaluation metrics. After this, we scrutinized the relation between optical flow and human activity recognition, taking the effectiveness of using multiple wearable sensors into consideration. Lastly, we focused on using only one sensor (first-person camera) for human activity recognition in order to minimize the number of sensors needed. In this way, the reader gains a clear overview of the state of the art and the intersection between the two research themes.

1.2.1 Optical Flow

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image [54]. Optical flow can emerge from relative motion between objects and the viewer. Despite the advances in computation, optical flow estimation is still an open and active research area in computer vision. Optical flow can be considered as a variational optimization problem to find pixel correspondences between any two consecutive frames [54]. Research paradigms in this field have evolved from considering optical flow estimation as a classical problem [25], to more high-level approaches using machine learning, for example, convolutional neural networks (CNN) as a state-of-the-art method [43, 61, 125, 111].

Training convolutional neural networks (CNN) to predict generic optical flow requires a massive amount of training data, including ground truth, and involves considerable computational power, e.g. graphics processing units (GPUs). However, obtaining ground truth for realistic videos is very hard to achieve [28] and simply not available in some scenarios. To overcome this problem, unsupervised learning frameworks have been proposed. In such way, the resources of unlabeled videos can be utilized [65].

The idea behind unsupervised methods is to avoid including ground truth optical flow in training convolutional neural network, but to nonetheless use a photometric loss that measures the difference between the target image and the (inverse / forward) warped subsequent image based on a generated dense optical flow field predicted from the convolutional networks. Hence, an end-to-end convolutional neural network can be trained with any amount of unlabeled image pairs, which helps in overcoming the need for ground truth optical flow as training input. However, fully unsupervised approaches are usually harder to train and show weaker

performance than supervised approaches, despite their having access to the true data statistics during training.

Researchers have generated many pre-trained optical flow estimation models in both supervised and unsupervised ways. The amount of effort and training time required to produce such models is significant. Consequently, benefiting from the better performance of the existing supervised pre-trained optical flow models to enhance motion boundaries for specific purpose datasets will help in reducing effort and time in scenarios in which little to no training data that includes ground truth is available.

1.2.2 Optical Flow Evaluation

Optical flow estimation methods have evolved dramatically. The most common evaluation metrics for the estimated optical flow are end point error (EPE) [91] and angular error (AE) [16], noticing that AE metric is based on prior work of Fleet and Jepson [46]. Even though EPE and AE metrics are popular, it is unclear which one is better. Moreover, AE penalizes errors in regions of zero motion more than motion in smooth non-zero regions. Also, different cases exist in which EPE gives the same value between various optical flow estimation scenarios. There is a need to evaluate existing optical flow performance metrics and suggest new ones to overcome the drawbacks of those in current use.

1.2.3 Human Activity Recognition

Recognizing human activities can be based on different sensor modalities, the most common ones being visual and inertial sensing. These modalities can be used simultaneously or independently.

Inertial measurement sensors (IMUs) are devices with capabilities to measure and report a body's specific force, angular rate, and orientation. The sensor's local coordinate system contains three main measurements: *accelerometers*, which are the instantaneous acceleration for each axis; *gyroscopes*, which represent the rotational velocity of the inertial; and *magnetometers*, which exemplify the instantaneous magnetic field measured with corresponding X, Y and Z axes. One of the drawbacks of using IMUs is the high measurement of uncertainty at slow motion

and lower relative uncertainty at high velocities. On the other hand, inertial sensors are able to measure very high velocities and accelerations.

Much research work suggests combining different sensor modalities to improve human activity recognition [30, 7, 2]. For example, in addition to IMUs, using visual descriptors extracted from visual sensors can mitigate the high measurement of uncertainty at slow motion captured by IMUs and can track features very accurately invariant to appearance of the representation at low velocities. However, for real life scenarios, a realistic and compromised number of modalities should be used. Previous research has focused on recognizing activities that are distinct and independent, like lie, sit, walk, cycle, ... etc. [132] or similar ones but with different visual items such as pour bag, pour oil, stir big bowl, stir egg, ... etc. [79]. Research in recognizing complex activities for the same object (e.g. a drawer) that are similar but opposite in nature, (e.g. open, close), is limited.

1.3 Research Questions

The research interest in the current endeavor focuses on finding suitable solutions for the problem statements and research challenges delineated previously, with the organization of this dissertation corresponding to the arrangement of our research questions. Consequently, the research questions with respect to optical flow as a computer vision problem have been notated (**I.x**), and research questions with regard to human activity recognition have the prefix (**II.x**) prefix. This means the research starts from optical flow as the pattern of apparent motion, continues on through the intersection between optical flow and human activities, and, lastly, answers pervasive computing questions related to human activity recognition.

Basically, we are interested in address the following research questions based on our problem statement and research challenges.

I.1 How is it possible to benefit from existing pre-trained optical flow models without the existence of ground truth and with a limited training set?

I.2 How can optical flow performance metrics be evaluated with the existence of ground truth?

I.3 How can the best optical flow evaluation metric be determined? What are the theoretical justifications of using one metric and why?

II.1 How can optical flow influence the use of multi-sensor human activity recognition?

II.2 What features can be extracted from multi-sensor human activity recognition? And what methods can be used for human activity recognition?

Research questions **I.1**, **I.2** and **I.3** are mainly concerned with the computer vision problem based on optical flow estimation, evaluating and enhancing motion boundary. Questions **II.1**, **II.2** are related to human activity recognition.

1.4 Contributions

This section points out the scientific contributions of this thesis, which not only answers research questions, but also enriches the computer vision and pervasive computing fields.

The following summarizes the main achievements:

1. We took advantage of the existing pre-trained models for optical flow estimation to fine-tune it in an unsupervised way in the absence of ground truth and when training dataset was limited. We also designed an unsupervised loss function based on classical variational optical flow estimation methods, which resulted in training objectives to learn the dataset specific statistics. Moreover, training time per dataset was reduced immensely. Additionally, the proposed unsupervised fine-tuning concept for optical flow resulted in improvement of motion boundaries estimated by gradients in the optical flow field.

2. We have provided a theoretical justification for using different optical flow performance metrics and the reasons behind it. Moreover, we have introduced novel optical flow performance metrics and evaluated them alongside current metrics.

3. We have developed an activity extraction tool for both visual and IMUs data based on [137] annotations for the Carnegie Mellon University Multi-Modal Activity database (CMU-MMAC) [38].

4. We introduced a novel statistical feature extraction method for IMUs data based on curvature of function graph and tracking the positions of left and right hands in space.

5. We have investigated complex and similar human activities for the same object, for example (close-drawer, open-drawer).

6. We have provided experimental proof of the limitation of IMUs data to distinguish human activities and suggesting that local visual descriptors can be complementary to IMUs for activity recognition.

7. We have minimized the number of sensors and used only first person visual data for activity recognition.

1.5 Published Work

This dissertation is based on previous publications, and a major component of the effort extends the content of these published works. For instance, Part I and II were conducted under the supervision and guidance of Prof. Heiner Stuckenschmidt and Prof. Samir Brahim Belhaouari. The published content can be found in the following selected publications:

- *Alhersh, T. and Stuckenschmidt, H. (2019). Unsupervised fine-tuning of optical flow for better motion boundary estimation. In Tremeau, A., Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications : February 25-27, 2019, in Prague, Czech Republic ; Volume 5: VISAPP (S. 776-783). , SciTePress: Setúbal, Portugal.*

- *Alhersh, T. and Stuckenschmidt, H. (2019). On the combination of IMU and optical flow for action recognition. In , 2019 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom*

Workshops) : 11-15 March 2019 in Kyoto, Japan (S. 17-21). , IEEE: Piscataway, NJ.

- *Alhersh, T., Brahim Belhaouari, S. and Stuckenschmidt, H. (2019). Action recognition using local visual descriptors and inertial data. In Chatzigiannakis, I., Ambient Intelligence : 15th European Conference, AmI 2019, Rome, Italy, November 13–15, 2019, Proceedings (S. 123-138). Lecture Notes in Computer Science, Springer International Publishing: Cham.*

- *Alhersh, T., Belhaouari, S. and Stuckenschmidt, H. (2020). Metrics performance analysis of optical flow. In Braz, J., VISIGRAPP 2020 : proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Feb 27, 2020 - Feb 29, 2020, Valetta, Malta (S. 749-758). , SCITEPRESS - Science and Technology Publications: Setúbal.*

1.6 Outline

This section provides the outline of this thesis and summarizes the content of each provided chapter.

Chapter1: Introduction. This chapter introduces the human activity recognition approaches used and the intersection between computer vision and pervasive computing. It also elucidates the motivation behind the research, the problem statement, research questions, the effort's scientific contributions, and related published work.

Part I: Foundation

Chapter2: Human Activity Representation. In this chapter, a foundation of human activity representation in respect to vision-based and sensor-based approaches has been provided. This foundation is necessary for the reader to understand later chapters that discuss our approaches for feature extraction in more

detail.

Chapter3: Human Activity Recognition. This chapter also falls under the foundation section, and it is important for the reader to understand the recognition techniques used in this work. We have provided basics of machine- learning approaches for human activity recognition, including Convolutional Neural Network (CNN) and Support Vector Machine (SVM).

Part II: Optical Flow Fine-tuning and Evaluation

Chapter4: Unsupervised Optical Flow Fine-tuning. In this chapter, we exploit a well-performing pre-trained model for optical flow estimation and then fine-tune it in an unsupervised way where ground truth is not available using a classical variational optical flow estimation method and training objectives to learn the dataset specific statistics. Thus, by means of dataset training, time can be reduced tremendously. Moreover, motion boundaries estimated by gradients in the optical flow field can be improved using the proposed unsupervised fine-tuning.

Chapter5: Performance Analysis of Optical Flow. We provide theoretical justification for using optical flow performance metric and the reasons behind this approach. In practice, design choices are often made based on qualitative unmotivated criteria or by trial and error. In this chapter, novel optical flow performance metrics are proposed and evaluated alongside current metrics.

Part III: Human Activity Recognition

Chapter6: Learning Human Activities. Here, we focus on the research questions introduced with respect to human activity recognition. For this reason, we first introduce an action extraction tool and then explain feature extraction methods for both IMUs and visual features. After this, we provide recognition techniques for human activities. Lastly, results of three experiments are reported: IMUs and optical flow, IMUs and local visual descriptor features, and visual features only. The chapter concludes with a comprehensive discussion.

Part IV: Wrap-up

Chapter7: Conclusion and Future Work. This chapter concludes the work presented in this thesis, highlighting the main research questions and how the research has answered them. Regarding future work, we have outlined promising future research directions that can be used to extend or enhance this work.

Part I

Foundation

CHAPTER 2

Human Activity Representation

This chapter presents the fundamental knowledge about human activity representation approaches categorized based on computer-vision and sensor-based paradigms that are crucial for understanding the subsequent chapters.

Human behavior analysis tasks can be classified according to the following semantic degrees: motion, action, activity, and behavior [90], as shown in **Figure** 2.1. Viewed from one perspective, motion is considered the lowest semantic degree and behavior the highest. Seen in another way, motion requires the shortest period of time to be performed. To develop a behavior, however, a longer period of motion capturing is needed. Motion information over time produces action, different interactions construct an activity, and more complex activities shape a behavior.

It is also useful to distinguish human behaviors at different levels of granularity. An example is physical behaviors for which the terms "action" and "activity" are mainly used in activity recognition communities. In some scenarios, these terms are used interchangeably, whereas in others they are used to denote behaviors of different complexity and duration. In the latter cases, the term "action" is usually applied to refer to simple behavior performed by a single person that typically lasts for a short period of time. Examples of actions include closing a cupboard, opening a jar, etc. In contrast, the term "activities" in this context is used to refer to more complex behaviors shaped from a sequence of actions and/or interleaving or overlapping actions.

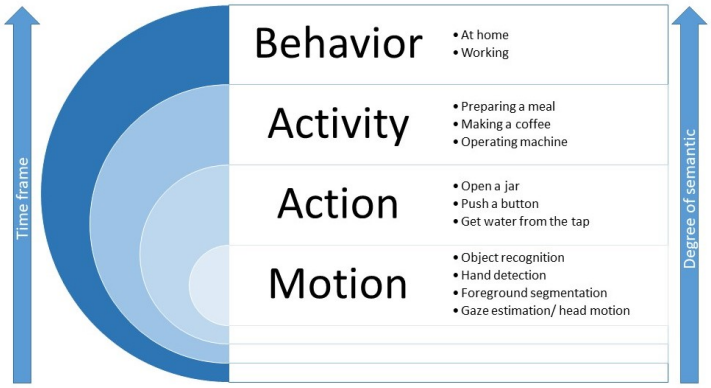


Figure 2.1: Human behavior components, starting from motion until shaping a behavior.

2.1 Vision Based Approaches

Vision-based human activity representation can be categorized based on research trajectories of global and local representations. In this context, optical flow is considered to be a global representation of human activity, while histogram of optical flow (HOF) and histogram of gradients (HOG) represent local descriptors.

2.1.1 Optical Flow

The notion of *optical flow* refers to the displacements of intensity patterns. The origin of this definition is based on the description of the physiological phenomenon when an image is formed on the retina, causing visual perception of the world. In this context, the relative motion between observer and object is the cause of optical flow. This motion only represents intensities in the image plane, but does not necessarily account for the actual 3D motion in the physical scene [120].

In computer vision practice, interest motion refers to the real displacement of objects. Hence, the projection of image plane in three-dimensional space of motion is what actually needs to be estimated; it is commonly referred to as the motion field. Another problem can arise when intensity changes due to changes in light or light reflection, which is known as scenes flow. The optical flow can only be extracted from a video frames sequence [116], and its formulation can be denoted

as described in the following paragraph.

In real world scene, the brightness reflection of a point represents the pixel intensity $I(x, y, t)$ at location (x, y) at time t in an image plane. The same point will be positioned in the image plane at location $(x + \delta x, y + \delta y)$ and time $t + \delta t$ after δt time lapse. At a short time interval, it is expected that the intensity of that point will remain unchanged and can be denoted as:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \tag{2.1}$$

The two-dimensional velocity in image plane can be represented by an optical flow vector (u, v), where, $u = \delta x/\delta t$ and $v = \delta y/\delta t$. Hence, **Equation** 2.1 is referred to as constant intensity constraint [54] and can be denoted as:

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t) \tag{2.2}$$

Equation 2.2 refers to the constant intensity constraint that is used for computing optical flow. From **Equation** 2.2, if time interval δt is considered very short, then, the computation of optical flow can be estimated by minimizing 2.3 equation. The classical work of Horn and Schunck [54] is considered the basis of the first practical model for optical flow.

$$\frac{d}{dt} I(x, y, t) = 0 \tag{2.3}$$

In the following section, we introduce the research paradigms in optical flow estimations that have evolved, from considering optical flow as a classical problem [25], to more high-level approaches using machine learning, an example of which is convolutional neural networks (CNN), a state-of-the-art method [43, 61, 125, 111].

Classical Approaches

In this section, we introduce optical flow basics for classical approaches that are not based on machine learning.

Differential Technique This technique, which uses spatial and temporal derivatives of image brightness to compute velocity, can also refer to gradient-based approaches. Relying on brightness constancy assumption leads to an aperture problem: two unknown components (vertical and horizontal displacement pixels) cannot be determined by one equation of flow field, and this creates an ill-posed problem.

In order to make the problem well-posed, another constraint needs to be incorporated with brightness consistency assumption as encoded priori information [54, 80]. Prior usually has the shape of spatial coherence exploit by global or local constraints. These constraint methods vary in their interaction with nearby pixels. Local methods use nearby pixels' intensity values as pixel flow constraint, whereas a global method uses nearby pixels' flow vectors as pixel flow constraint.

Region-based Technique The core of this technique is based on finding matched patches between two consecutive images. The highest correlation between two corresponding patches is defining optical flow, which is the shift of those patches [16, 10]. This technique is more robust to noise than differential techniques; it also works well when images are decimated or interlaced [117].

Feature-based Technique In this technique, an attempt is made to link discriminative sparse features for successive images over time [26, 127] in two main steps: features detection, followed by corresponding matching. The estimated optical flow ignores areas of ambiguity, and though the generated flow field is sparse, it is robust. The optical flow can be determined via discriminative features for edges, corners, and low-contrast features like flat regions [81]. The feature-based technique has two general drawbacks. Firstly, the estimated optical flow is very sparse if objects or background contain non-discriminative features. Secondly, selected features sometimes prove to be unreliable and disappear in subsequent frames.

Frequency-based Technique The frequency-based technique can be referred to as velocity-tuned filters in which optical flow is calculated using filters in the Fourier domain such as velocity-tuned filters. One advantage of this technique is that using mechanisms operating on spatio-temporally oriented energy in the

Fourier domain [17] makes it possible to estimate motion that cannot be estimated using matching approaches. For instance, this technique can work with random dot patterns motion. Frequency-based techniques can be classified into two groups based on velocity-tuned filters output: energy-based methods and phase-based methods.

Variational Approaches

Horn and Schunck have suggested minimizing global energy function for computing the displacement vector of each pixel. This function consists of two terms: brightness consistency and smoothness. This method plays a vital role in optical flow computation because almost all top-performing optical flow algorithms rely on variational techniques.

Variational optical flow techniques have many advantages compared to other optical flow estimation methods [26, 27, 24]:

- Different assumptions could be integrated into a one minimization framework.
- They produce a dense flow field because it has a filling-in effect, whereas numerous other methods require interpolation of the sparse flow field as post-processing step.
- The energy function could be formulated in such a way which is invariant to rotations in most cases.
- Variational optical flow techniques can be accelerated using bidirectional multigrid methods to some extent allowing for real-time performance on standard hardware [27].

The energy function [25] can be structured by combining color, gradient and smoothness terms where $I_1, I_2 : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^3$ are any two consecutive frames. Also, $x := (x, y)^T$ are the point in Ω domain, and $w := (u, v)^T$ is the optical flow field as follows:

$$E(w) = E_{color} + \gamma E_{gradient} + \alpha E_{smooth} \tag{2.4}$$

where the color energy E_{color} is an assumption that the corresponding points should have the same color:

$$E_{color}(w) = \int_{\Omega} \Psi(|I_2(x + w(x)) - I_1(x)|^2)dx \tag{2.5}$$

The gradient energy $E_{gradient}$ is a constraint that is invariant to additive brightness changes to deal with the illumination effect:

$$E_{gradient}(w) = \int_{\Omega} \Psi(|\nabla I_2 + w(x)) - \nabla I_1|^2)dx \tag{2.6}$$

Adding smoothness constraint E_{smooth} works as a regularity term for penalizing the total variation of the flow field generated from 4.2 and 4.3:

$$E_{smooth}(w) = \int_{\Omega} \Psi(|\nabla u(x)|^2) - |\nabla v(x)|^2)dx \tag{2.7}$$

Function $\Psi(s)$ allows dealing with non-Gaussian deviations corresponding to matching criteria and occlusions. It corresponds to a Laplace distribution that has longer tails than the Gaussian distribution.

Machine Learning Approaches

One of the most popular machine learning algorithms with a vast impact on almost all disciplines is Neural Networks. It has been applied decisively over time and outperforms other algorithms in speed and accuracy. Convolutional Neural Networks (CNN) is a neural networks variant used mainly in the field of computer vision. The name convolutional has been derived from hidden layers that shape the neural network and consist of convolutional layers, pooling layers, normalization layers, and fully connected layers.

CNN methods learn to extract deep features from input images. Even through optical flow estimation needs accurate per-pixel localization, it also requires finding correspondences between two consecutive input images. This, in turn, includes learning image feature representations and learning to match them at different locations in the two images [43].

This breakthrough achievement encouraged other subsequent optical flow esti-

mation techniques such as supervised, unsupervised, and semi-supervised. Optical flow estimation using CNN algorithms provides a promising alternative to variational methods. Flexibility in using image features for optical flow estimation is considered the main advantage of using CNN methods. CNN can extract more abstract, deeper, and multi-scale features using multi-layer and hierarchical architectures. Moreover, CNN can model complex, non-linear transformations between the input images and the estimated flow field. Overall, the stochastic minimization of the loss across an entire training dataset avoids some of the pitfalls of optimizing a complex energy-function on individual inputs in variational methods [85].

Optical Flow Evaluation

Evaluation procedures for optical flow estimation methods are important for the quality of the optical flow produced. Two main approaches can be used for evaluating estimated optical flow: qualitative and quantitative.

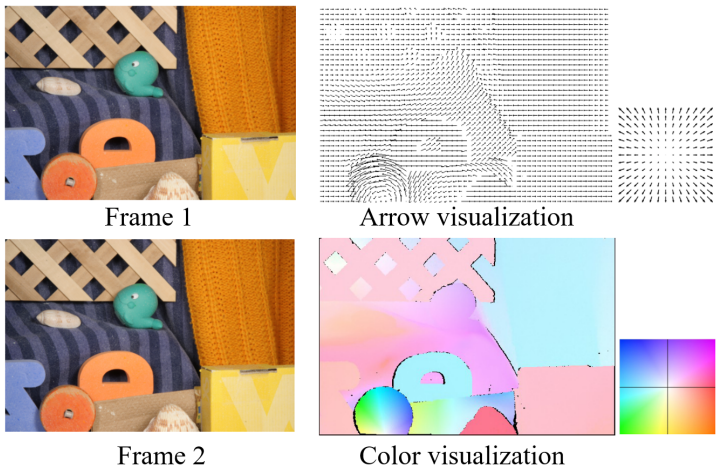


Figure 2.2: Two types of visualization of the motion field transforming Image1 in Image2.

Motion fields visualization provides qualitative intuition in regard to the accuracy of the optical flow estimation. Two main visualization techniques are presented in **Figure** 2.2. Motion vectors are directly represented by arrow visualization and provides a good intuitive understanding of physical motion. On the

counterpart, a clear illustration needs motion field under-sampling to prevent overlapping of arrows.

The other technique is color coding visualization, in which motion field vectors are associated with a color hue to indicate direction and a saturation to represent the magnitude of the vector. It allows for dense visualization of the flow field and for better visual visual perception of subtle differences between neighbor motion vectors [47].

The first quantitative evaluation metrics for optical flow were published in 1994 [91, 16] and suggested the use of end point error (EPE) [91], which can be described as the *Euclidean* distance between two vectors; it can be defined as **Equation** 5.1:

$$EPE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}. \tag{2.8}$$

and AE [16], which represents the angle between the two extended vectors $(1, u, v)$ and $(1, u_{GT}, v_{GT})$ and defined in **Equation** 5.2:

$$AE = cos^{-1}\left(\frac{uu_{GT} + vv_{GT} + 1}{\sqrt{u^2 + v^2 + 1}\sqrt{u_{GT}^2 + v_{GT}^2 + 1}}\right). \tag{2.9}$$

2.1.2 Histogram of Optical Flow (HOF)

This method is based on extracting motion features from image sequences using optical flow [94]. The main advantage of this method is that the overburden of correctly estimating motion in variable lighting conditions and mess is entirely limited to optical flow calculation.

Histogram of optical flow does not make any assumptions about the source of optical flow data; nevertheless, it can be adopted and applied in various ways. The only implicit assumption is that the sequences of images have the same frame rate and the same optical flow field dimensions. Moreover, HOF assumes that each sequence of images contains a single temporal reference, which can be used for temporal alignment, and that there exists predefined partitioning of the image into sub-regions exists.

HOF basically computes the primary motion in each of the sub-regions. For

motion, amplitude and direction are quantized via the use of 2D optical flow histograms, and therefore the dominant motion can be encoded simply by assigning a symbol to each of the histogram bins. This way, a compact representation of whole body motion, including gestures, is built. The sets of such symbol sequences are called HOF descriptors. In a real-world implementation, the descriptors can be extracted from the flow sequences immediately after the flow is obtained, therefore reducing the need for storage of original video sequences or optical flow field sequences. Observing the maximum in each histogram is an inherently noisy approach; however, due to the small number of bins, the effects of noise are small. Likewise, the lowest-velocity bin is discarded to get rid of the low-velocity noise, which inevitably appears in optical flow vectors.

First, note that the image flow induced by camera rotation (pan, tilt, roll) varies smoothly across the image irrespective of 3D depth boundaries, and in most applications it is locally essentially translational because significant camera roll is rare. Thus, any kind of local differential or difference of flow cancels out most of the effects of camera rotation. The remaining signal is due to either depth-induced motion parallax among the camera, subject, and background, or to independent motion in the scene. Differentials of parallax flows are concentrated essentially at 3D depth boundaries, while those of independent motions are largest at motion boundaries [35, 74].

2.1.3 Histogram of Gradients (HOG)

The appearance and shape of local objects inside an image can be characterized very well by the distribution or histogram of local intensity gradients or edge directions, even without accurate prior knowledge of the corresponding gradient or edge positions. Histogram of oriented gradients (HOG) is one of feature descriptors used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image-detection window, or region of interest (ROI).

A practical implementation of the HOG descriptor algorithm is described as follows:

1. Divide the image into small connected regions called cells, and for each cell

accumulating a local 1-D histogram of gradient directions or edge orientations is computed for the pixels within the cell.

2. Discretize each cell into angular bins according to the gradient orientation.
3. Each cell's pixel contributes weighted gradient to its corresponding angular bin.
4. Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.
5. For better invariance to illumination, shadowing, etc., normalization group of histograms that represent blocks can be performed. The set of these block histograms represents the descriptor.

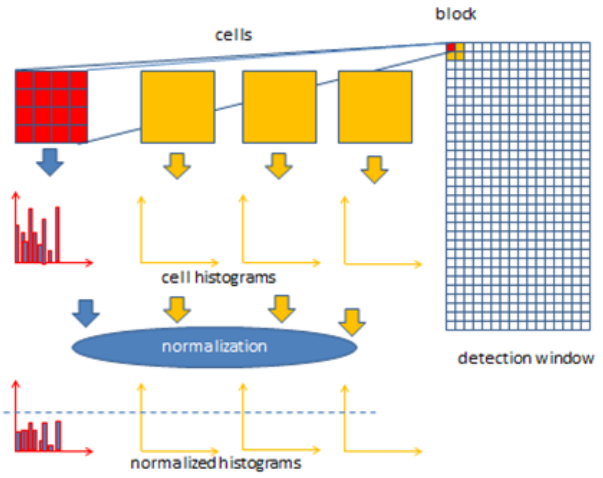


Figure 2.3: Demonstration of HOG algorithm and implementation scheme [1].

Figure 2.3 demonstrates the algorithm implementation scheme of HOG. HOG descriptor computation requires some basic configuration parameters before applying it; for instance, masks to calculate derivatives and gradients, geometry of splitting an image into cells and grouping cells into blocks, overlapping of blocks and normalization parameter [1].

2.1.4 Motion Boundaries Histogram (MBH)

Another common visual descriptor for videos is Motion Boundary Histogram (MBH), an approach proposed by [35]. One advantage of this approach is the robustness to camera and background motions. The basic idea of MBH is to represent the oriented gradients computed over the vertical and the horizontal optical flow components. In this representation, constant camera movements tend to disappear, and the description focuses on optical flow differences between frames (motions boundaries).

Motion boundaries histogram represents optical flow vertical and horizontal components separately using two scalar maps, which can be viewed as gray-scale images of the motion components. Then, histograms of oriented gradients are computed for each component of the two optical flow images using the same approach deployed for computing HOG in normal images. It should be taken into consideration that only flow differences represented by information changing in motion boundaries are kept, and the constant motion information is removed, which cancels most camera motion effects [118, 119].

2.2 Sensor Based Approaches

Currently, a vast range of sensors is available in activity monitoring. This includes sensors based on RFID, audio, accelerometers, and motion detectors to name but a few. These sensors are different in many ways, varying, for example, in type, purpose, signals generated, underlying theoretical concept, and technical hardware and infrastructure. However, they can be categorized into two main classes in terms of how they are deployed in activity recognition applications: wearable sensors and dense sensors. In this research, we are mainly focused on Inertial Measurement Units (IMUs) wearable sensors, as they are unobtrusive and considered to be complementary to visual-based sensors. In the following section, sensors considered for use in this work are only introduced as part of experiments or discussion. The three main components of IMUs - sensor accelerometer, gyroscope, and magnetometer - are introduced in greater detail in the following section.

2.2.1 Inertial Measurement Units (IMUs)

An inertial measurement units (IMUs) is a sensor that provides three-axis acceleration, angular turning rate (gyroscope), and magnetometer. The mass of the sensor has the mechanical freedom to move independently from the outer assembly. The sensor mass has a plurality of sensing and suspension elements of particular orientation on a selected plane for each axis of detection that face a corresponding set of sensing and suspension elements on the respective interior surfaces of said outer assembly [87].

Accelerometer

An accelerometer is an electromechanical sensor that can measure either static or dynamic forces of acceleration of a body. Static forces include gravity, while dynamic forces can include vibrations and movement. This reflects the change in velocity for a certain time duration.

There are many different ways to build an accelerometer. Some use the piezoelectric effect and have microscopic crystal structures that get stressed by accelerative forces, causing a voltage to be generated. Another way to make an accelerometer is via sensing changes in capacitance. For instance, if there are two microstructures next to each other, they have a certain capacitance between them. The capacitance change will cause an accelerative force by moving one of the structures [3].

Gyroscope

A gyroscope measures the angular velocity. This sensor calculates how fast an angle changes around an axle over time. A gyroscope can capture body rotation that determines the orientation. Different classes of gyroscopes exist, depending on the physical operating concepts and the technology involved. Gyroscopes can be used alone or integrated in more complex systems such as a gyrocompass [45]; Inertial Measurement Units [87], Inertial Navigation System [68] and Attitude Heading Reference System [3].

The main effect upon which a gyroscope depends is that an isolated spinning mass shows tendency to maintain its angular position with respect to an inertial

reference frame. When an external constant torque (or a constant angular speed) is applied to the mass, its rotation axis undergoes a precession motion at a constant angular speed (or a constant output torque) in a direction that is normal to the direction of the applied torque (or to the constant angular speed) [93].

Magnetometer

A magnetometer sensor is an instrument used to measure the direction and strength of a magnetic field in the vicinity of the instrument. Magnetometer location on Earth plays a role in the varieties of magnetism because of the differences in the magnetic field caused by the differing nature of rocks and the interaction between charged particles from the sun and the magnetosphere of a planet [14].

The two main components of an inertial sensor are accelerometer and gyroscope; however, magnetometer can also be considered as part of an inertial measurement units. Nevertheless, stated more precisely, magnetometer is not an inertial sensor [115]. When magnetometer is combined with accelerometer and a gyroscope, it allows the continual tracking of body orientation for all three dimensions: pitch, yaw, and roll, which are the three dimensions of movement when an object moves through a medium. In theory, accelerometer and magnetometer are sufficient to obtain those dimensions, but having a gyroscope will increase precision. For instance, magnetometer accuracy is poor for fast moving objects; on the other hand, accuracy is maintained over time. In contrast, gyroscope accuracy drops significantly over time since it reacts fast and accurately to changes. Furthermore, both accelerometer and the gyroscope need an initial orientation start because both only react to changes. Consequently, all three sensors excel at different levels, and combining them allows quick, accurate positioning and orientation of objects.

CHAPTER 3

Human Activity Recognition

This chapter introduces the fundamentals of human activity recognition in respect to machine learning. In this context, the chapter's focus is to introduce a background for underlying related topics applied in this work.

Due to the availability and accessibility of wearable sensors, human activity recognition (HAR) has become one of the trendiest and popular research topics in the last decade. HAR has gained in importance because of wide engagement in many research areas, including healthcare systems, interactive gaming and sports, and monitoring systems that seek to improve the quality of life. In HAR, many human activities are recognized, such as opening a drawer, turning a light on or off, preparing a meal, walking, sleeping, taking medicine, etc. While diverse methods can be used to recognize human activity, this research limits it focus to two main machine learning methods: convolutional neural networks (CNN) and support vector machines (SVM).

Figure 3.1: HAR generic methodology design workflow, which includes, decide human activity, define a device to capture this activity, collecting data that includes pre-processing for data and labeling, training the model and then evaluating it.

Figure 3.1 illustrates the general workflow for designing general human activity recognition methodologies [40]. Designing HAR-based application encompasses five main steps. In the first, the activity to be recognized is defined. In the

second, the type of device for sensing the designated activity is determined and data collection started. The third step is to decide what is to be done with the collected data, including data pre-processing and cleaning and annotation. In the next step, a machine learning model has to be built, which includes training, validating, and testing the model. In the fifth and final step, the accuracy of the model should be evaluated in terms of the activity recognition metrics.

3.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) were first introduced by [76] in 1989. They are considered a special type of neural network for processing data that has a predefined topology. For example, time-series data - which can be represented as a 1-D grid when sampled at a specific time interval - has been taken into consideration. Image data can be also considered as a 2-D grid of pixels. The name CNN indicates that this type of neural network employs convolution, which is a mathematical operation. Convolution is a special type of linear operation and will be discussed in the following sections. Hence, CNN is a simplified type of neural network that uses convolution operation instead of general matrix multiplication in at least one layer of the network.

3.1.1 Convolution Operation

Convolution in general is an operation on two functions of a real-valued argument, e.g. if we are tracking the location of an object with a laser sensor, and this sensor provides a single output $x(t)$ that represents the position of this object at time t. In this case, both x and t are real values, and saying this, the differences in multiple readings from the laser sensor at multiple instant times can be obtained if we assume that this laser sensor is somehow noisy. To obtain a less noisy estimate of the object's position, averaging several measurements can be performed. Relevant measurements are represented by the more recent measurements to obtain a weighted average that gives more weight to recent measurements. This can be done via a weighting function $w(a)$, where a is the age of a measurement. If we apply such a weighted average operation at every moment, we obtain a new function s

providing a smoothed estimate of the position of the object:

$$s(t) = \int x(a)w(t-a)da. \tag{3.1}$$

The latter operation is termed a *convolution*. The convolution operator can be denoted with an asterisk [51]:

$$s(t) = (x * w)(t). \tag{3.2}$$

From the previous example, and based on convolutional network terminology, the first argument represented by the function x usually represents the input of the network. The second argument, denoted by the function w, is called the kernel. The output of the network is sometimes referred to as a feature map. Usually, when dealing with computer data, time should be discretized, and the sensor mentioned in the previous example will provide data at regular intervals, for instance once per second. Index of time t can, after that, take integer values only. Assuming that x and w are only defined on integer t, the discrete convolution is as follows:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \tag{3.3}$$

In machine learning applications, the input data as shown in **Figure** 3.2 is usually a multidimensional array, and the kernel is usually defined as a multidimensional array of parameters that are adapted by the learning algorithm. Each element of the input and kernel must be explicitly stored separately, so it's usually assumed that these functions are zero everywhere but in the finite set of points for which values are stored. In other words, the infinite summation can be considered a summation over a finite number of array elements. Moreover, convolutions can be used over more than one axis at a time. For example, when using a two-dimensional image I as input, we probably also want to use a two-dimensional kernel K:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n). \tag{3.4}$$

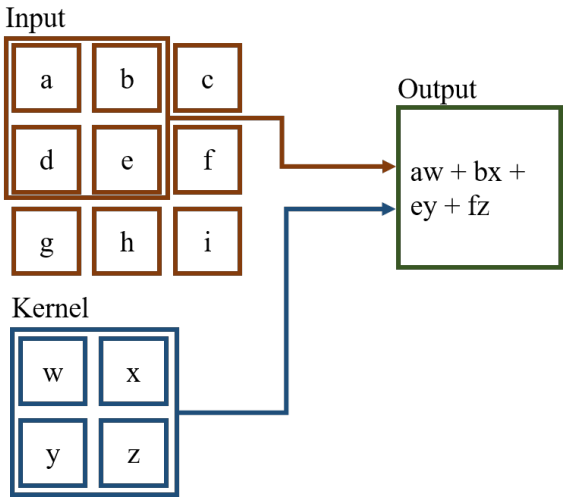


Figure 3.2: In this example, a 2-D convolution has been performed using the input data (brown outline) with 2×2 kernel (blue outline); the output is shown in the green box.

3.1.2 Layers Used to a Build CNN

A CNN is a sequence of layers; the function of every layer of a CNN transforms one volume of activations to another through a differentiable function. Three main types of layers are used to build CNN architectures: the convolutional layer, the pooling layer, and the fully connected layer. Stacking these layers will form a full CNN architecture.

Convolutional layer - This layer is counted as the core building block of any convolutional network. Most of the heavy computational processing is done in this layer. It computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. The convolutional layer parameters comprise of a set of learnable filters. Each filter is spatially small; however it extends via the full depth of the input volume.

Pooling layer - Periodic insertion of a pooling layer is considered a common practice between successive convolutional layers in network architecture. The

main goal of this layer is to reduce the number of parameters and overhead computations in the network in order to control overfitting. In other words, this layer performs a down sampling operation along the spatial dimensions.

Fully connected layer - In this layer, all neurons are fully connected to all activations in the previous layer, similar to the same concept in regular neural networks. Hence, their activations can be calculated with a matrix multiplication followed by a bias offset.

3.1.3 CNN Architecture Overview

Unlike regular neural networks, the layers shaping CNNs consist of neurons arranged in three dimensions (width, height, and depth), where the depth is referred to as the third dimension of an activation volume, and not to the depth of a full CNN. This is the case, for instance, if input images have an input volume of activations, and the volume has $32 \times 32 \times 3$ dimensions (width, height, depth, respectively). Consequently, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully connected manner. Moreover, in this example, the final output layer will have $1 \times 1 \times c$ dimensions (where c is the number of classes) because by the end of the CNN, the architecture full image is reduced into a single vector of class scores, arranged along the depth dimension as shown in **Figure** 3.3.

3.2 Support Vector Machine (SVM)

A support vector machine [53] (SVM) is a supervised learning classifier that computes a hyperplane to separate different classes. Accordingly, the purpose of this classifier is to find a hyperplane that separates data and avoids over-fitting. An SVM works by representing each data point in an n-dimensional space. Then the SVM classifier determines a hyperplane that maximizes the margin between different classes (see **Figure** 3.4). If the hyperplane fails to separate all data points, the SVM classifier transforms input data to a higher dimensional space to make them linearly separable, as shown in **Figure** 3.5. Transformation of data to a higher di-

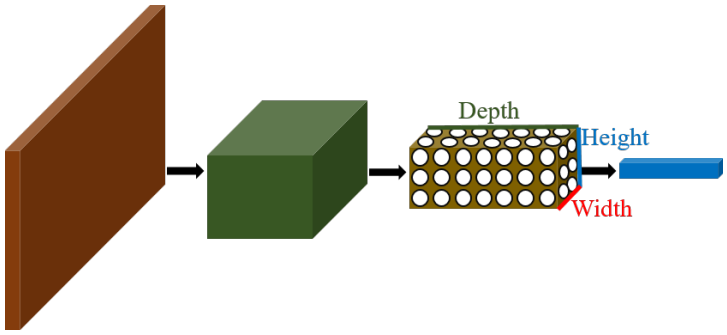


Figure 3.3: A CNN arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a CNN transforms the 3D input volume to another 3D output volume of neuron activations. In this example, the brown input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (red, green, and blue channels, assuming that the image is an RGB image).

mensional space is called a kernel trick. This helps in projecting data points with extra features to get linear classification.

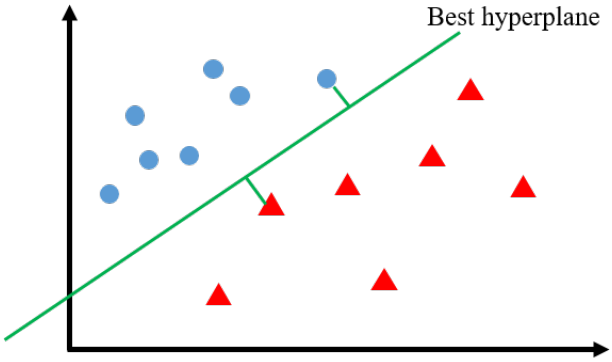


Figure 3.4: A simple example of using an SVM classifier. In this example, the SVM objective is to find an optimal separation hyperplane between the class blue circles and the class red triangles. The best hyperplane is shown in green.

Finding the maximum-margin hyperplane for separating data point classes can be considered an optimization problem. Tuning parameters can be used in SVMs to enhance the resulting hyperplane - for instance, the type of kernel function used (e.g. linear, polynomial, or Sigmoid), as well as what is termed the “gamma pa-

rameter" that affects some kernels. Also, a regularization parameter can be used to avoid misclassifying the data points. Multiclass classification problems in SVMs can be solved using one-vs-one or one-vs-the-rest [55].

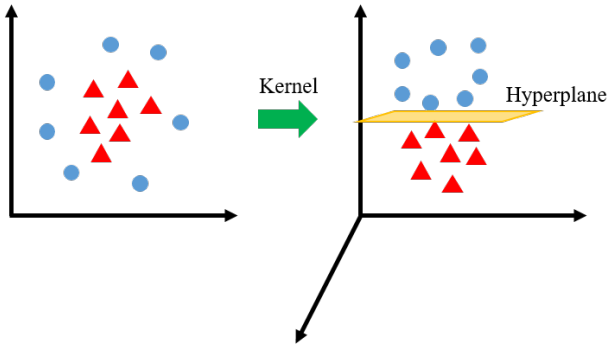


Figure 3.5: An example in which the hyperplane fails to separate all data points on the left side. In this case, the SVM classifier transforms input data to a higher dimensional space, as on the right side, to make them linearly separable.

Part II

Optical Flow Fine-tuning and Evaluation

CHAPTER 4

Unsupervised Optical Flow Fine-tuning

The notion of *Optical flow* refers to the displacements of intensity patterns. The origin of this definition is based on the description of a physiological phenomenon when an image is formed on the retina causing what is called visual perception of the world. In this context, the relative motion between observer and object causes optical flow in the scene. This motion only represents intensities in the image plane, but does not necessarily account for the actual 3D motion in the physical scene [120].

In computer vision practice, interest motion refers to the real displacement of objects. Hence, the projection of the image plane in the three-dimensional space of motion is what actually needs to be estimated - and is commonly called the motion field. Another problem can arise when intensity changes due to a change in light or light reflection, which is known as scenes flow. The optical flow can only be extracted from a video frames sequence [116].

One of the most popular machine learning algorithms, and has had vast impact on almost all disciplines, is neural networks. It has been applied significantly over time and outperforms other algorithms in speed and accuracy. Convolutional neural networks (CNN) are a variant of neural networks used mainly in the field of computer vision. The name convolutional has been derived from hidden layers that shape the neural network and consist of convolutional layers, pooling layers, normalization layers, and fully connected layers.

CNN methods learn to extract deep features from input images. Estimation of

optical flow requires accurate per-pixel localization, and it also requires that correspondences between two consecutive input images be found. This includes learning image feature representations and learning to match them at different locations in the two images [43].

Due to this breakthrough achievement, the development of supervised, unsupervised, and semi-supervised optical flow estimation techniques was subsequently encouraged. Optical flow estimation using CNN algorithms provides a promising alternative to the variational method. CNN's flexibility in using image features for optical flow estimation - it can extract more abstract, deeper, and multi-scale features using multi-layer and hierarchical architectures - is considered its main advantage. Moreover, CNN can model complex, non-linear transformations between the input images and the estimated flow field. Overall, the stochastic minimization of the loss across an entire training dataset avoids some of the pitfalls of optimizing a complex energy-function on individual inputs in variational methods [85]. Recently, CNN-based approaches have proven successful in optical flow estimation in the supervised, semi-supervised, and unsupervised training paradigms. Supervised training requires large amounts of training data with task-specific motion statistics. Usually, synthetic datasets are used for this purpose. For semi-supervised training, a combination of labeled and unlabeled data is required. Although fully unsupervised approaches have access to the true data statistics during training, they are usually harder to train and show weaker performance.

In this chapter, we exploit a well-performing pre-trained model and fine-tune it in an unsupervised way using a classical optical flow estimation method. The training objectives will help facilitate learning of the dataset-specific statistics, thus reducing per dataset training time by big margin.

4.1 Related Work

Motivated by the success of deep learning in various computer vision tasks, the era of optical flow estimation has been shifted from classical energy-based techniques to end-to-end trained models. In this chapter, we focus on these end-to-end deep learning methods used for optical flow estimation. As mentioned previously, learning processes for optical flow can be divided into supervised, semi-supervised

and unsupervised learning. Supervised and semi-supervised learning optical flow requires ground truth but in different ways (to be discussed in the following sections), whereas the unsupervised learning approach doesn't require ground truth.

4.1.1 Supervised Optical Flow Learning

This type of optical flow learning networks is based on end-to-end CNN architectures, in turn based on regression [75], which utilize CNN for the whole pipeline by acting as an approximation function to effectively learn the relationship between input images and the desired optical flow output having the labeled training dataset [59].

Training a network in a supervised way requires a huge number of image pairs, with their ground truth flow as a training dataset. In real-life scenarios, obtaining dense optical flow ground truth is challenging. To overcome the lack of appropriate training data for optical flow, especially ground truth, Dosovitskiy *et al.* [43] have created a synthetic dataset named FlyingChairs by layering natural images with rendered computer-aided design models of chairs. In order to simulate real-life motion, they have followed a special parameterized affine motion.

Dosovitskiy *et al.* [43] suggested the first end-to-end CNN, which contains two CNN networks. The first is FlowNetSimple or (FlowNetS) in which input images are stacked together and then fed through a generic network to decide how to process the image pair to extract the motion information. The second is FlowNetCorr (FlowNetC), which includes a correlation layer that performs multiplicative patch comparisons between two feature maps. Their network succeeded in predicting optical flow at up to ten image pairs per second.

However, due to the substantial differences between synthetic and real-life images, FlowNet - which was trained on the FlyingChairs dataset - unfortunately didn't generalize well to real images. Actually, the accuracy of the estimated optical flow using FlowNet, even after fine-tuning on real-world images, fell behind the results of classical energy-based models at that time. This opened a question as to whether CNN regression models can outperform classical energy-based methods. Nevertheless, FlowNet has demonstrated the possibility of using an end-to-end CNN regression model for optical flow estimation. Furthermore, FlowNet

has opened the door for other researchers to build on top of it by establishing many standard practices that are used for training optical flow models, such as learning-rate scheduling, overall network architectures, data augmentation for both image pair and ground truth flow. The last mentioned includes geometric transformations, adding Gaussian noise, changing color and brightness and had great influence on the follow-up research.

By stacking several simple FlowNet models with some modifications and the introduction of a fusion network, Ilg *et al.* [61] achieved astonishing results using FlowNet2. Regardless of the conceptual simplicity of the model, the stack of multiple FlowNet networks had a powerful impact and significantly improves the estimated optical flow accuracy by more than 50% over FlowNet.

Not only did Ilg *et al.* improve accuracy results, but they also provided many crucial practices for training networks such as using a correlation layer and pre-training and fine-tuning on synthetic datasets. FlowNet2 has also been trained on another synthetic dataset, which has 3D motion and photometric effects called FlyingThings3D [83]. They showed that using the proper training dataset can increase optical flow estimation accuracy by more than 20%.

On the other hand, PWC-Net [111], a network which is 17 times smaller than FlowNet2, was designed using three main concepts: pyramidal processing, warping, and the use of a cost volume. In this effort, they adopted DenseNet architecture named was designed using three main concepts; pyramidal processing, warping, and the use of a cost volume. Also, they adopted DenseNet architecture [56], which directly connects each layer to every other layer in a feed forward fashion. PWC-Net uses coarse-to-fine method in many pyramid levels by constructing feature pyramid using CNN for optical flow estimation. After that, PWC-Net creates a cost volume using the feature maps from both the source image and the warped target image based on the current optical flow. Next, the following CNN layers will decode optical flow outputs from the cost volume. Their results show the advantage of light weight design of the network and this reflects on shorting the training time and the fast estimation, while obtaining competitive results compared to other methods.

Ranjan and Black [96] introduced the Spatial Pyramid Network (SPyNet), in which they combined classical coarse-to-fine pyramid methods with deep learning

for optical flow estimation. It contains five pyramid levels, each of which consists of a shallow CNN that estimates optical flow between source and targeted images. Although SPyNet is outperformed by classical energy-based optical flow estimation methods, it succeeded in integrating classical methods in deep learning.

SPyNet is 96% smaller and faster than FlowNet; hence, less memory is required, which makes it promising for embedded and mobile applications. SPyNet learns to predict flow increment at each pyramid level rather than minimizing a classical objective function.

LiteFlowNet was developed by Hui *et al.* [57]; it is 30 times smaller than the model size of FlowNet2 and 1.36 times faster in execution. In the process, they drilled down the missed architectural details in FlowNet2, which involved introduction of an effective flow inference at each pyramid level through a lightweight cascaded network to improve optical flow estimation accuracy, permitting seamless incorporation of descriptor matching in the network. Moreover, a flow regularization layer was developed to ameliorate the issue of outliers and vague flow boundaries by using a feature-driven local convolution.

Hur and Roth [58] have proposed an iterative residual refinement (IRR) network, which is based on an iterative estimation scheme using weight sharing. IRR can be applicable to many backbone architectures to improve the accuracy of optical flow estimation. The basic concept of IRR is iteratively refining output from a previous pass through the network as input and using only a single network block with shared weights, which improves optical flow estimation by residually refining the previous estimate.

A Hierarchical Discrete Distribution Decomposition, known as HD^3 [136], was proposed by Yin *et. al.* for dense pixels correspondence optical flow estimation. This concept enables both optical flow estimation and the corresponding uncertainty. The main architectural design of HD^3 is based on PWC-Net, using, for instance, a multi-scale pyramid, warping, and cost volume. This work defers from the previously mentioned optical flow estimation methods in that it directly regresses optical flow with CNN. Their experimental results show clear advantages, including having state-of-the-art accuracy for optical flow estimation on established benchmark datasets and uncertainly measures.

4.1.2 Unsupervised Optical Flow Learning

In supervised methods for optical flow estimation, it is very important to have ground truth flow as part of the training dataset. In real-life scenarios, obtaining dense optical flow ground truth is challenging. To overcome the lack of appropriate training data for optical flow, especially ground truth, another research paradigm for CNNs adopts an unsupervised learning approach. Self-supervised or unsupervised learning of optical flow depends on proxy loss minimization more than optical flow estimation and should be close to some ground truth. Hence, designing the correct proxy loss is crucial to the success of unsupervised estimation of optical flow [59].

Inspired by the classical Horn and Schunck [54] optical flow estimation, Ahmadi and Patras [4] used a loss function based on the classical equation of optical flow constraint (brightness constancy assumption) to train CNN. Dealing with this unsupervised loss function as a minimization problem, the network leaned how to predict optical flow. By combining coarse-to-fine estimation, they improved optical flow estimation to be as close as FlowNet.

FlowNet was adopted while equipped with unsupervised Charbonnier loss function to minimize photometric consistency, which measures the difference between the first input image and the (inverse) warped subsequent image based on the predicted optical flow by the network [65, 98]. They have proposed using unsupervised proxy loss inspired by the Markov random field (MRF). This proxy is followed by a smoothness term. Their optical flow estimations were competitive to supervised methods and suggests that unsupervised optical flow estimation methods have potential when labels are missing for training data.

Zhu *et al.* [142] argue that using optical flow estimators to generate proxy ground truth data by means of an off-the-shelf classical energy-based method for training CNNs could help in learning to estimate motion between image pairs that is as good as using true ground truth. They demonstrated that the network backbone could be enhanced by using DenseNet [56], a dense connectivity network.In another effort, Long *et al.* [78] used the interpolation between frames to train CNNs for optical flow estimation.

In order to handle occlusion, Wang *et al.* [124] proposed an end-to-end network

consisting of two copies of FlowNetS with shared parameters. One is to produce forward optical flow, and the other generates backward warping, which is used for occlusion mask. Loss function used includes occlusion predicted by motion. To tackle large motion estimation, they introduced a histogram equalizer and an occlusion map for the warped frame.

Makansi *et al.* [82] proposed an assessment network that can learn to predict the error form generated by a set of optical flow fields with various optical flow estimation techniques. Then, the assessment network is used as a proxy ground truth generator to train FlowNet. This effort is the most closely related to our work, except that we focus on the effectiveness of implementing classical optical flow optimization objectives in CNN architecture.

Janai *et al.* [64] learned optical flow and occlusions together via modeling a temporal relationship for a three-framed window by estimating past and future optical flow. They used photometric loss function and reason explicitly about occlusions. Their extended unsupervised optical flow learning using a multi-frame setting was based on PWC-Net architecture. They used three different types of decoders: a future frame decoder that estimates optical flow from the reference frame to the future frame; past optical flow decoder; and an occlusion decoder. The results obtained by Janai *et. al.*were competitive to classical energy-based techniques.

In their work, Meister *et. al.*[85] suggested using a proxy loss function that takes occlusions into consideration. They demonstrated better accuracy than a supervised backbone such as FlowNet. Moreover, bi-directional optical flow is estimated using the same network by means of changing the order of input images and then detecting occlusions via a bi-directional consistency check. The proxy loss function is applied to non-occluded regions only since brightness constancy assumption can't hold for occluded pixels.

Meister *et. al.*proposed a higher-order smoothness term in addition to a ternary census loss [107, 138] to obtain a data term that is robust to brightness changes. This advanced proxy loss significantly improves the accuracy by halving the error compared to previous unsupervised learning approaches. SelFlow [77] suggests injecting noise into superpixels to create occlusion, and then letting one model guide another in order to learn optical flow from occluded pixels. Furthermore, they demonstrated the use of multi-frame input as an extension to improve optical flow

estimation accuracy using temporal coherence exploiting. SelFlow was evaluated on public benchmark datasets.

4.1.3 Semi-supervised Optical Flow Learning

A compromise between supervised and unsupervised learning optical flow estimation, semi-supervised methods based on Generative Adversarial Networks (GANs) [52] have also recently been proposed. For example, Lai *et. al.*[73] has proposed an adversarial loss to learn the structural pattern of the flow warp error, which allows training the network in a semi-supervised fashion. Their method is based on producing optical flow using a generator network from two given input images. They then calculated a flow warp error map using the difference in image intensity between the first image and the warped second image (using the flow output). Lastly, using a discriminator network, they distinguished between whether the warp error map is created by the generator or it is the ground truth. The purpose of the generator is to fool the discriminator network by generating optical flow whose warp error patterns are similar to the ground truth. Training such networks requires a combination of labeled and unlabeled data. Their experimental results have demonstrated benefits from supervised and unsupervised methods.

Yang *et. al.*[133] also proposed a semi-supervised network of optical flow estimation by learning a conditional prior. They argued that current learning-based approaches for optical flow estimation are not based on any explicit $regularizer$ (this refers to any prior, model, or assumption that adds any restrictions to the solution space). Hence the results obtained have a risk of overfitting while training, causing a mismatch problem when testing. To overcome this issue, Yang *et. al.*proposed a network that contains prior information of potential optical flows from input image and later used this network as a $regularizer$ for training.

First, to learn prior knowledge on potential optical flows of input image, they trained the conditional prior network in a supervised way. Next, FlowNet was trained in an unsupervised way using a regularization loss from the trained conditional prior network. Their experimental results showed the importance of using a conditional prior network to get competitive results with the usual supervised training, also demonstrating better generalization on a different dataset. This sug-

gests that semi-supervised optical flow learning can help in domain generalization by leveraging from other domain ground truth availability when labeled data is missing.

4.2 Dataset

Three well-known datasets have been used for unsupervised fine-tuning and testing predicted optical flow: **KITTI 2012** [49], **KITTI 2015** [86] and **Sintel** [28]. The decision against including the Flying Chairs dataset [43] in this work was made because the FlowNet2-SD model was trained on it and to avoid over fitting while fine-tuning.

4.2.1 KITTI

KITTI 2012 [49] is a real-world computer vision benchmark that was recorded using four video cameras with high resolution, a laser scanner, and a localization system. It contains 389 stereo image pairs and their optical flow; there are 194 training image pairs and 195 image pairs for testing purposes.

KITTI 2015 [86] is another benchmark containing 200 training scenes and 200 test scenes (four color images per scene, saved in lossless png format). Compared to the KITTI 2012 benchmark, it covers dynamic scenes for which ground truth was established in a semi-automatic process.

4.2.2 Sintel

Sintel [28] is an open source synthetic dataset extracted from animated film produced by Ton Roosendaal and the Blender Foundation. It contains 1041 image pairs for training and 552 image pairs for testing both training and testing. It comes with clean and final versions that have been used to investigate when optical flow algorithms break. This means that each frame has been rendered in different pass: the clean pass, which contains shading, but no image degradations, and the final pass, which additionally includes motion blur, defocus blur, and atmospheric effects, and corresponds to the final movie [129].

Since the Sintel training dataset provides optical flow ground truth, which can be used for validating our approach, we have divided the training dataset into training (845 training image pairs) and validation (196 testing image pairs from alley_2, ambush_5, market_2, and sleeping_1 sequences) datasets for validating our method.

4.3 Methods

Many motion boundary estimation methods depend on optical flow [92, 122, 126, 62]. Philippe Weinzaepfel *et al.* [126] suggested a learning-based method for motion boundary detection based on random forests since motion boundaries in local patch tend to have similar patterns, static appearance and temporal features, color, optical flow, image warping and backward flow errors. In their work, Li *et al.* proposed an unsupervised learning approach for edge detection. This method utilizes two types of information as input: motion information in the form of noisy semidense matches between frames, and image gradients as the knowledge for edges. The performance of motion boundary estimation is limited by several issues, such as the removal of weak image edges and label noises.

We have adopted FlowNet2-SD architecture, which is implemented in the Caffe deep learning framework and considered a subnet of Flownet2 [61]. FlowNet2-SD is a modified, deeper version of FlowNetS to deal with small displacements. FlowNet2-SD architecture is illustrated in **Figure** 4.1. We have replaced the final and intermediate losses with unsupervised losses described in the following section.

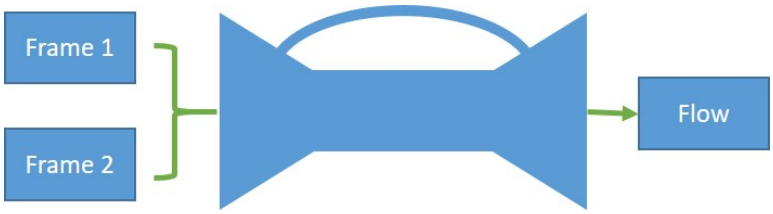


Figure 4.1: FlowNet2-SD architecture that takes two input images and produces optical flow (repainted from [61]).

4.4 Network Architecture

Figure 4.2 shows our proposed unsupervised loss function. Only one stage (resolution) is shown from multi-resolution optical flow architecture adopted from FlowNet2-SD. Stacking both input images together and feeding them to the network allows the network itself to decide how to process the image pair to extract the motion information. In each stage (resolution), the loss is constructed by calculating three main losses:

- Warp loss is calculated when the second frame is back warped with the optical flow produced and the difference between the generated warped frame and frame one is calculated.
- Gradient loss is the difference calculated between gradients of warped image and gradients of frame one.
- Smoothness loss works as a penalizing term through calculating the variation of generated flow field in the u and v directions.

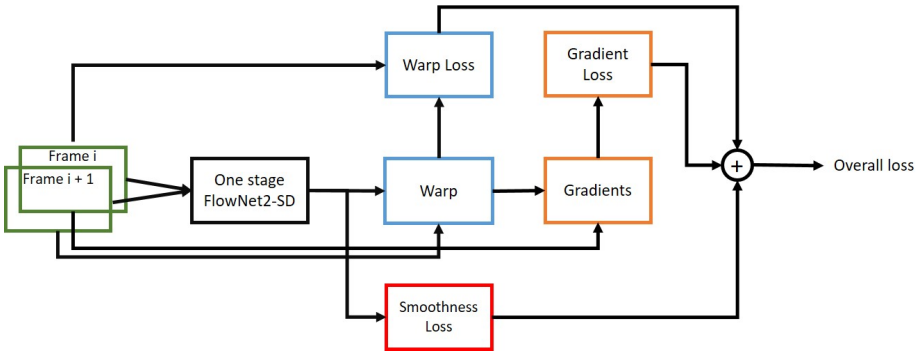


Figure 4.2: An overview of how unsupervised losses have been constructed. Only one stage (resolution) of producing flow from FlowNet2-SD is shown.

Cost Functions

The cost function can be structured by combining color, gradient and smoothness terms where $I_1, I_2 : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^3$ are any two consecutive frames. Also,

$x := (x, y)^T$ are the point in Ω domain and $w := (u, v)^T$ is the optical flow field [25] as follows:

$$E(w) = E_{color} + \gamma E_{gradient} + \alpha E_{smooth} \tag{4.1}$$

Where the color energy E_{color} is an assumption that the corresponding points should have the same color:

$$E_{color}(w) = \int_{\Omega} \Psi(|I_2(x + w(x)) - I_1(x)|^2) dx \tag{4.2}$$

The gradient energy $E_{gradient}$ is a constraint that is invariant to additive brightness changes to deal with the illumination effect:

$$E_{gradient}(w) = \int_{\Omega} \Psi(|\nabla I_2 + w(x)) - \nabla I_1|^2) dx \tag{4.3}$$

Adding the smoothness constraint E_{smooth} works as a regularity term for penalizing the total variation of the flow field generated from **Equations** 4.2 and 4.3:

$$E_{smooth}(w) = \int_{\Omega} \Psi(|\nabla u(x)|^2) - |\nabla v(x)|^2) dx \tag{4.4}$$

$\Psi(s)$ represents different metrics as follows:

$$\Psi(s) = \begin{cases} \|s\|_1, \\ s \in [E_{color}(w), E_{gradient}(w), E_{smooth}(w)] \\ \|s\|_2, \\ s \in [E_{color}(w), E_{gradient}(w), E_{smooth}(w)] \end{cases} \tag{4.5}$$

Where,

$$\|s\|_1 = \sum_{i=1}^{n} |y_i - f(x_i)| \tag{4.6}$$

and

$$\|s\|_2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{4.7}$$

Equation 4.5 shows the non-local functions used in our approach. $L_1 norm$ **Equa-**

Table 4.1: Different combinations of cost function terms from **Equation** 4.1 and their references used in this research.

Terms	Reference
$E_{\|color\|_2} + \gamma E_{\|gradient\|_1} + \alpha E_{\|smooth\|_1}$	f_1
$E_{\|color\|_2} + \gamma E_{\|gradient\|_2} + \alpha E_{\|smooth\|_1}$	f_2
$E_{\|color\|_1} + \gamma E_{\|gradient\|_1} + \alpha E_{\|smooth\|_1}$	f_3
$E_{\|color\|_2} + \alpha E_{\|smooth\|_1}$	f_4
$E_{\|color\|_2}$	f_5
$\gamma E_{\|gradient\|_1} + \alpha E_{\|smooth\|_1}$	f_6
$\gamma E_{\|gradient\|_1}$	f_7

tion 4.6 and L_2norm **Equation** 4.7 were used for different combinations using color, gradient, or smoothness terms.

4.5 Results and Discussion

4.5.1 Quantitative and Quantitative Results

Visualizations of some generated examples of optical flow are illustrated in **Figure** 4.3 for Sintel and in **Figure** 4.4 for KITTI 2012 and 2015. KITTI here represents real-world data, while Sintel exemplifies a synthetic scenario. Our method succeeded in capturing fine structure results around edges, while FlowNet2-SD shows smooth results as shown in **Figure** 4.7.

Quantitative results show that FlowNet-SD-unsup achieved good results with comparison to baseline. We are not in a situation to compete with fine-tuning in supervised way (ground truth available) and achieve better results, but to find a fast (in terms of training and execution) and reasonable method to produce competitive optical flow when ground truth is not given, i.e. real-world scenarios. Runtime for generating one optical flow file takes only $1.3e^{-4}$ seconds.

Evaluation for the validation dataset from Sintel based on different combinations of cost function described in **Table** 4.1 fine-tuned on Sintel is shown in **Table** 4.2.

Table 4.2 results show a small variation in the EPE values for different settings. For example, using L_2norm for warping and gradient functions combined

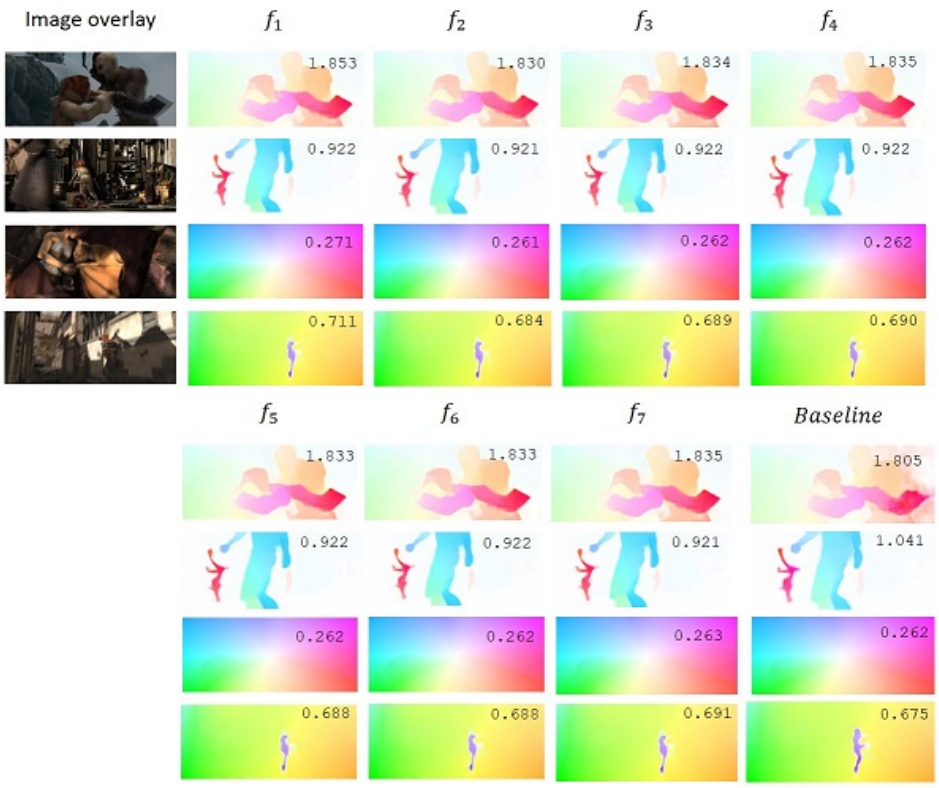


Figure 4.3: Examples of optical flow estimated using different combinations of cost functions based on **Table** 4.1 and EPE on different Sintel validation sets. f_{1-7} are the corresponding cost function line in the mentioned table.

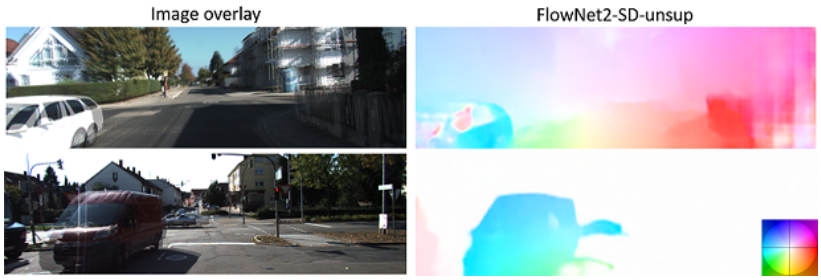


Figure 4.4: Qualitative results: Optical flow estimated on KITTI 2012 upper part, and KITTI 2015 bottom part using our method FlowNet2-SD-unsup. Right bottom corner shows optical flow color code used in this manuscript.

Table 4.2: EPE results for evaluating our method on the validation sets of Sintel training with comparison to FlowNet2-SD (Baseline).

Method	Sintel Clean	Sintel Final
FlowNet2-SD (Baseline)	4.036	4.354
FlowNet2-SD-ft-unsup	4.016	4.354

with $L_1 norm$ for smoothness achieved the best results for alley_2 clean, ambush_5 clean, market_2 clean, and sleeping_1 clean validation datasets. Baseline has outperformed our approach in some final validation sets by a small margin, but the average EPE for final validation sets from both FlowNet2-SD-unsup and FlowNet2-SD is the same.

4.5.2 Motion Boundary Evaluation

We have compared motion boundary estimations from our method (FlowNet2-SD-unsup) and the baseline (FlowNet2-SD). Our method outperforms the baseline by a large margin in Sintel clean, while it was almost the same for Sintel final, **Table** 4.3. There is also an improvement in quality of qualitative results, which is visible in **Figure** 4.5. The variations in motion in different validation sequences have produced different F-measures in **Table** 4.3. One observation is that the F-measure score is correlated with the number and magnitude of produced motion boundaries.

Defining the correct and optimum values of network parameters is crucial to obtaining good results. Therefore, we have observed that even if EPE results are minimal, good visualization for optical flow is not always obtained.

Another observation is reported in **Table** 4.4: while investigating our approach on the produced optical flow results from Sintel validation dataset, EPE results is vary among different validation sequences (alley_2, ambush_5, market_2 and sleeping_1) and in some cases inside the same sequence **Figure** 4.6.

Figure 4.6 shows two different frames from ambush_5 validation dataset, their corresponding magnitude maps for $\vec{U}$ and $\vec{V}$ and a histogram of optical flow magnitudes and EPE. The histogram of optical flow magnitudes for the above frame shows that most values have small magnitudes between -5, and 5 and the majority

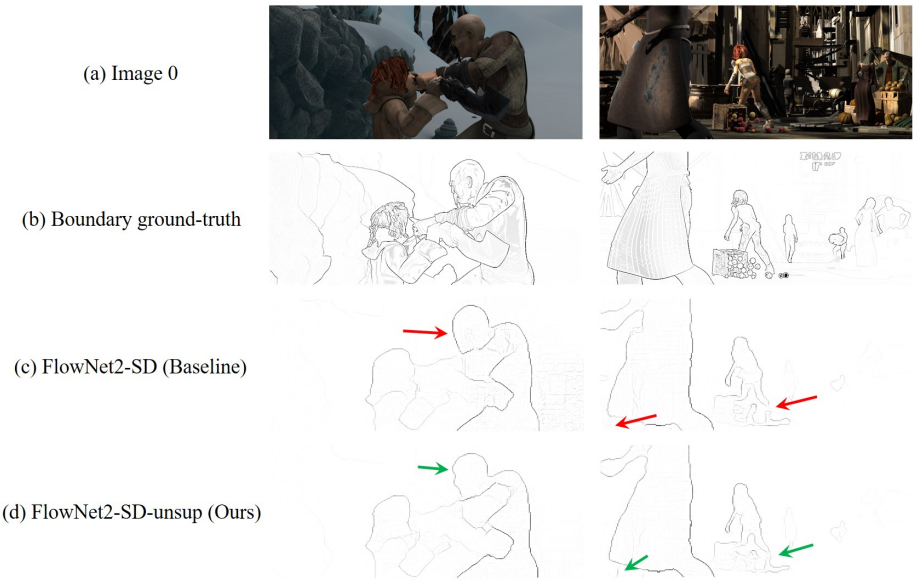


Figure 4.5: Visualization of motion boundaries from some Sintel validation sets. Our approach succeeds in detecting more fine structures (see green arrows) compared to the baseline.

Table 4.3: F-measure comparison between our motion boundary estimation generated by our method FlowNet2-SD-ft-unsup using different loss function as described in **Table** 4.1 and the baseline on the Sintel train validation dataset.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	Baseline
alley_2	Clean	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
	Final	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
ambush_5	Clean	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.46
	Final	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.49
market_2	Clean	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.67
	Final	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.72
sleeping_1	Clean	0.23	0.23	0.23	0.22	0.23	0.23	0.23	0.27
	Final	0.31	0.31	0.31	0.32	0.32	0.31	0.31	0.19
Average	Clean	0.40	**0.40**	0.40	0.40	0.40	0.40	0.40	0.34
	Final	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37

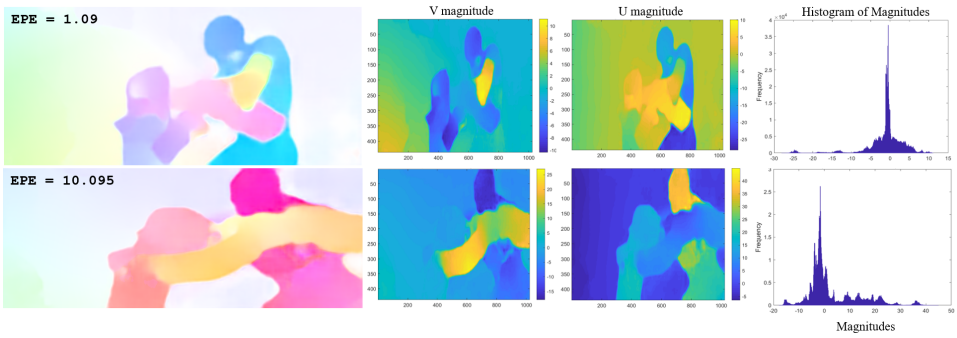


Figure 4.6: Two different images from validation sequence ambush_5 and their corresponding histograms of optical flow magnitudes and visualizations of magnitudes in U and V directions. It's obvious that higher EPE value has higher large frequencies.

Table 4.4: EPE results for optical flow generated by our method FlowNet2-SD-ft-unsup using different loss function as described in Table 4.1 and the FlowNet2-SD (Baseline) on various Sintel validation sequences.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	Baseline
alley_2	Clean	0.520	**0.517**	0.521	0.531	0.521	0.520	0.520	0.518
	Final	0.528	0.527	0.528	0.528	0.528	0.527	0.527	**0.525**
ambush_5	Clean	14.372	**14.366**	14.373	14.403	14.374	14.372	14.372	14.454
	Final	15.612	15.612	15.613	15.612	15.612	**15.611**	**15.611**	15.625
market_2	Clean	0.936	**0.935**	0.936	0.940	0.937	0.937	0.937	0.941
	Final	1.030	**1.029**	1.030	**1.029**	**1.029**	1.030	1.030	1.035
sleeping_1	Clean	0.237	**0.235**	0.238	0.246	0.237	0.237	0.237	**0.234**
	Final	0.250	0.250	0.250	0.249	0.249	0.250	0.250	**0.236**
Average	Clean	4.016	**4.013**	4.017	4.030	4.017	4.016	4.016	4.037
	Final	4.355	4.355	4.355	4.354	4.354	4.355	4.354	4.355

is around zero with EPE 1.09. On the other hand, the distribution of optical flow magnitudes in the below frame are between -10 and 10 with extended distribution to 20 with EPE 10.095. This indicates that our method is not able to capture large displacement in motion represented by high magnitude values.

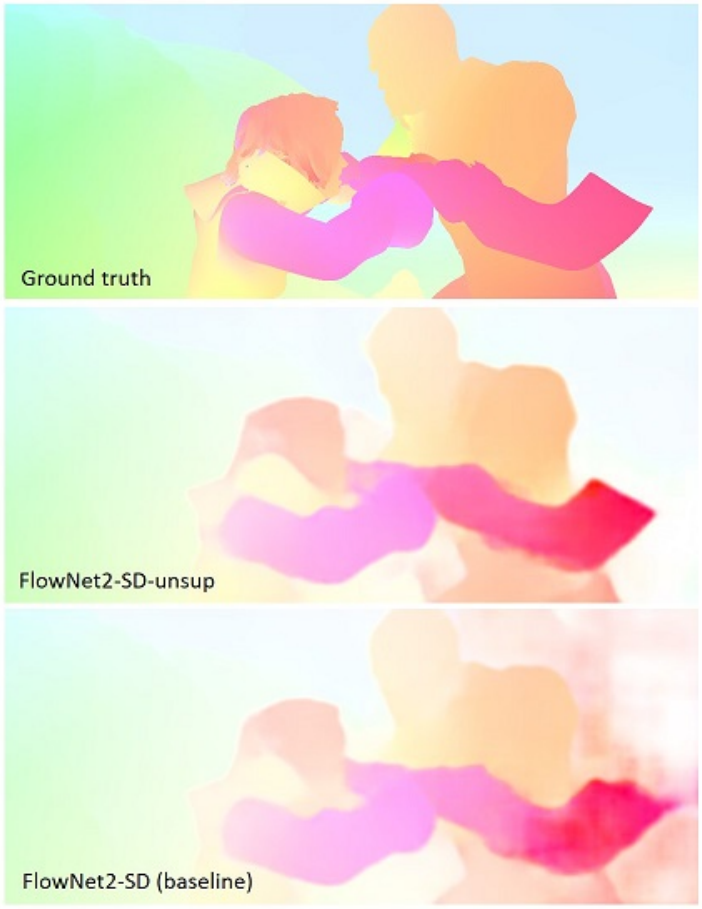


Figure 4.7: Qualitative - results: We compare our results in the second row using FlowNet2-SD-unsup with ground truth in the first row and the baseline generated from FlowNet2-SD in the third row. Our model produces better flow and captures fine structures around boundaries.

4.6 Conclusion

To conclude this chapter, we have introduced an unsupervised loss function based on classical optical flow formula using deep learning. Our approach shows potential to minimize the need of ground truth for both optical flow estimation and motion boundary detection. Moreover, we benefit from pre-trained models to reduce time via fast unsupervised fine-tuning. This work opens the opportunity to

investigate more on how to enhance the results to compete with state-of-the-art approaches.

CHAPTER 5

Performance Analysis of Optical Flow

Optical flow computation is considered a fundamental problem in computer vision. In fact, it originates from the physiological phenomenon of visual perception of the world through image formation on the retina, which is based upon the displacement of intensity patterns [47]. Thus, optical flow can be defined as the projection of velocities of 3D surface points onto the imaging plane of visual sensors [17]. However, the relative motion constructed between the observer and objects of an observed scene only represents motion of intensities in the image plane; it does not necessarily represent the actual 3D motion in reality [120]. A consequent problem emerges that intensity changes are not necessarily due to objects' displacements in the scene, but can also be caused by other circumstances such as changing light, reflection or modifications of objects' properties that affect their light emission or reflection [47]. Research paradigms in optical flow estimation have advanced from considering it as a classical problem [54, 25] to a higher-level approaches using machine learning [125, 111, 8]. For instance, convolutional neural networks (CNNs) are considered to be a state-of-the-art method for optical flow estimation.

Despite the fact that optical flow estimation methods have evolved dramatically, the most common evaluation methodologies are end point error (EPE) [91] and angular error (AE) [16], noticing that the AE metric is based on prior work by Fleet and Jepson [46]. Even though EPE and AE metrics are popular, it is unclear which one is better. Moreover, AE penalizes errors in regions of zero motion more than motion in smooth non-zero regions. Furthermore, different cases exist

(**Figure** 5.1) in which EPE gives the same value between various scenarios, which will be discussed later in this chapter. The purpose of this research is not to evaluate optical flow estimation methods, but, to evaluate the existing optical flow direct evaluation metrics and suggest new metrics compensate for drawbacks of existing ones.

5.1 Related Work

Even though many optical flow estimation algorithms have been proposed, there are few publications that analyze their performance. Two main approaches can be used for evaluating optical flow: qualitative and quantitative. Motion fields of optical flow can be visualized in either arrow or color forms (**Figure** 2.2), which provide qualitative insights on the accuracy of the estimation. Arrow visualization represents motion vectors and provides good intuition about motion. On the other hand, motion field vectors should be under-sampled to prevent arrows overlapping. The color code visualization allows for dense representation of the motion field by associating color hue to the direction and saturation to the magnitude of vectors [47]. The first direct quantitative evaluation metrics for optical flow were published in 1994 [91, 16] and suggested using EPE [91], which can be described as the *Euclidean* distance between two vectors. It is defined in **Equation** 5.1:

$$EPE = \sqrt{(u - u_G)^2 + (v - v_G)^2}. \tag{5.1}$$

and AE [16], which represents the angle between the two extended vectors $(1, u, v)$ and $(1, u_G, v_G)$; it is defined in **Equation** 5.2:

$$AE = cos^{-1}\left(\frac{uu_G + vv_G + 1}{\sqrt{u^2 + v^2 + 1}\sqrt{u_G^2 + v_G^2 + 1}}\right). \tag{5.2}$$

AE is very sensitive to small estimation errors caused by small displacements, whereas EPE hardly discriminates between close motion vectors [47]. **Figure** 5.1 illustrates four different cases in which the EPE metric gives same error value between ground truth (G) and estimated motion vector. This drawback is caused by the fact that EPE only considers the difference of vectors and ignores the magnitude

of each one.

McCane *et al.* [84] have suggested two evaluation metrics. The first, which is based on the AE metric with motion vectors normalization, is defined in **Equation** 5.3, which is based on AE metric with motion vectors normalization. Nevertheless, AE does not take the vector magnitude into consideration and uses only angles; the normalization step has no actual effect.

$$E_A = cos^{-1}(\hat{c} \cdot \hat{e}), \tag{5.3}$$

where E_A is the error measure, c is G, e is the estimated optical flow, and $\hat{}$ denotes vector normalization.

The second metric is the normalized magnitude of the vector difference between G and estimated optical flow which is defined in **Equation** 5.4.

$$E_M = \begin{cases} \frac{\|c-e\|}{\|c\|} & \text{if } \|c\| >= T, \\ |\frac{\|e\|-T}{T}| & \text{if } \|c\| < T \text{ and } \|e\| >= T, \\ 0 & \text{if } \|c\| < T \text{ and } \|e\| < T, \end{cases} \tag{5.4}$$

where E_M is the error measure.

Baker *et al.* [15] compared the performance of EPE and AE and argued that EPE should become the preferred optical flow evaluation metric based on a qualitative assessment of an estimated optical flow for Urban sequence.

Despite the fact that optical flow estimation methods have evolved dramatically, the most common evaluation methodologies are end point error (EPE) [91] and angular error (AE) [16], noting that the AE metric is based on prior work of Fleet and Jepson [46]. Even though EPE and AE metrics are popular, it is unclear which one is better. Moreover, AE penalizes errors in regions of zero motion more than motion in smooth non-zero regions.

5.2 Dataset

Since this work does not strive to evaluate optical flow estimation algorithms, we decided to use only the ground truth dataset from Baker *et. al.*[15]. The number of available ground truth files is eight, and the data description is shown in **Table** 5.1. Three G files have maximum values of more than 10^9 for a limited number of pixels: Dimetrodon, Hydrangea and RubberWhale. In order not to have a bias in the analysis results, a threshold of maximum 10^7 was set. We created different modified versions of G based on possible scenario errors with magnitude in the steps set $S = \{-30, -20, -10, 10, 20, 30\}$. For each G file, one of the following scenarios is applied:

1. Shift G horizontally by $s \in S$ and replace shifted pixels by zeros.
2. Shift G vertically by $s \in S$ and replace shifted pixels by zeros.
3. Shift G horizontally and vertically by $s \in S$ and replace shifted pixels by zeros.
4. Rotate G by $s \in S$ degrees and replace shifted pixels by zeros.
5. Magnify G by multiplying by $s \in S$.
6. Shift G horizontally and vertically and then rotate by $s \in S$ and replace shifted pixels by zeros.
7. Shift G horizontally and vertically and then rotate and magnify by $s \in S$ and replace shifted pixels by zeros.

This will allow us to have 42 different versions of each G file with total of 336 modified G files.

5.3 Methods

To overcome the drawbacks of the existing evaluation metrics for optical flow, we proposed five different metrics: $\vec{E}$ is the modified optical flow represented by (u, v), whereas $\vec{G}$ is the ground truth vector notated by (u_G, v_G).

Name	Min	Max	Std
Dimetrodon	-4.33E+00	1.67E+09	3.55E+08
Grove2	-3.31E+00	4.01E+00	3.64E+00
Grove3	-4.09E+00	1.43E+01	2.89E+00
Hydrangea	-7.02E+00	1.67E+09	4.13E+08
RubberWhale	-4.58E+00	1.67E+09	2.09E+08
Urban2	-2.13E+01	8.51E+00	7.96E+00
Urban3	-4.19E+00	1.73E+01	5.15E+00
Venus	-9.38E+00	7.00E+00	2.91E+00

Table 5.1: Used G files in our experiment.

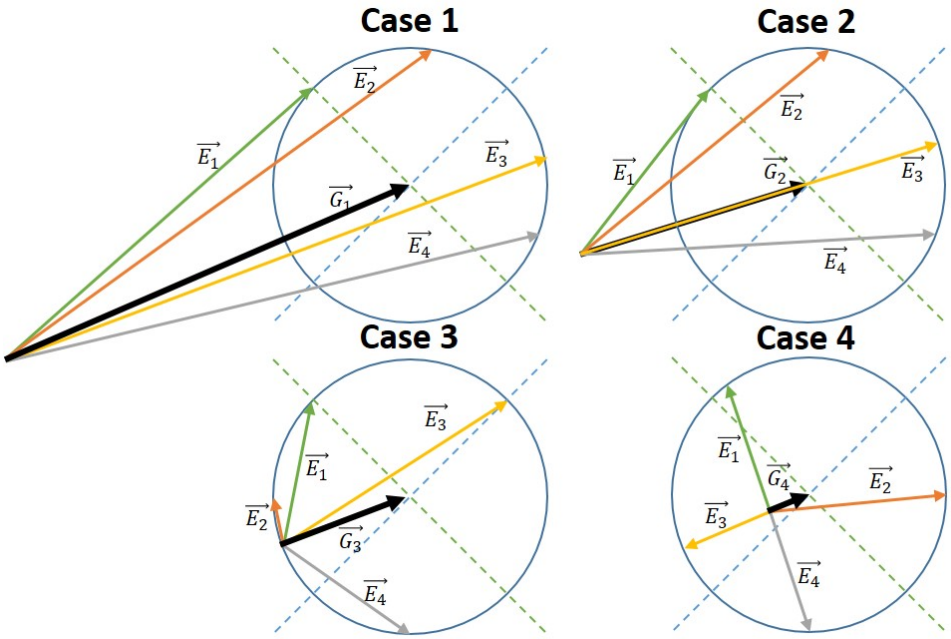


Figure 5.1: Different cases where the EPE metric gives the same error value between G represented by the black vector $\vec{G}_i, i \in (\vec{G}_1, \vec{G}_2, \vec{G}_3, \vec{G}_4)$ and other estimated optical flow vectors $\vec{E}_j, j \in (\vec{E}_1, \vec{E}_2, \vec{E}_3, \vec{E}_4)$.

5.3.1 2D-Angular Error (2DAE)

Adding a new dimension and forcing it to be equal to 1 will affect the measurement of the angle. This is an enhancement on AE, where the angle in 2D space between

(u, v) and (u_G, v_G) is considered instead of 3D space as shown in **Equation** 5.5:

$$2DAE = \begin{cases} cos^{-1}\left(\frac{u_{est}u_G + v_{est}v_G}{\sqrt{u_{est}^2 + v_{est}^2}\sqrt{u_G^2 + v_G^2}}\right), \\ \text{if } (u^2 + v^2)(u_G^2 + v_G^2) \neq 0 \\ \theta, \text{if } (u^2 + v^2)(u_G^2 + v_G^2) = 0 \end{cases} \tag{5.5}$$

for example, if we have the following two points $(0.1, 0.1)$ and $(3, 3.1)$, then AE = 1.2025, but 2DAE = 0.0164.

5.3.2 Generalized Angular Error (GAE)

This is also an enhancement on AE, where the angle in the 3D space between (α, u, v) and (β, u_G, v_G) is considered instead of 3D between $(1, u, v)$ and $(1, u_G, v_G)$ space. From Cauchy Schwarz's theory [106], we can prove the following inequalities:

$$-1 \leq \frac{\alpha\beta + (uu_G + vv_G)}{\sqrt{\alpha^2 + u^2 + v^2}\sqrt{\beta^2 + u_G^2 + v_G^2}} \leq 1 \tag{5.6}$$

The metric GAE can be defined as:

$$GAE = \begin{cases} cos^{-1}(\frac{\alpha\beta + (uu_G + vv_G)}{\sqrt{\alpha^2 + u^2 + v^2}\sqrt{\beta^2 + u_G^2 + v_G^2}}), \\ \qquad \text{if } (u^2 + v^2)(u_G^2 + v_G^2) \neq 0 \\ \theta, \quad \text{if } (u^2 + v^2)(u_G^2 + v_G^2) = 0 \end{cases} \tag{5.7}$$

where α and β can be any real numbers, for instance if $\alpha = \beta = 0$, then GAE = 2DAE. On the other hand, if $\alpha = \beta = 1$ this will lead to AE in **Equation** 5.2.

5.3.3 Joint Angular and End Point Error (JAEE)

This metric is a kind of mixture between AE and EPE, where the difference in magnitude between (u, v) and (u_G, v_G) is added to the perpendicular distance between them **Figure** 5.2. The perpendicular distance between (u, v) and (u_G, v_G) is defined as follows:

$$max\big(\|proj_{\vec{G}}\vec{E}\|, \|proj_{\vec{E}}\vec{G}\|\big) \tag{5.8}$$

where the perpendicular distance is defined as the angular distance between the two non-null vectors $\vec{E}$ and $\vec{G}$. Therefore, our metric can be defined as

$$JAEE = \begin{cases} \|\vec{G} - \vec{E}\| + max\big(\|proj_{\vec{G}}\vec{E}\|, \|proj_{\vec{E}}\vec{G}\|\big), \\ \qquad \text{if} \quad \|\vec{G}\vec{E}\| \neq 0 \\ \|\vec{G} - \vec{E}\| + max\big(\|\vec{G}\|, \|\vec{E}\|\big), \\ \qquad \text{if} \quad \|\vec{G}\vec{E}\| \neq 0 \end{cases} \tag{5.9}$$

where the projection of vector $\vec{b}$ over $\vec{a}$ is given by the following formula:

$$proj_{\vec{a}}\vec{b} = \frac{\vec{a}\vec{b}}{|\vec{a}|^2}\vec{a} \tag{5.10}$$

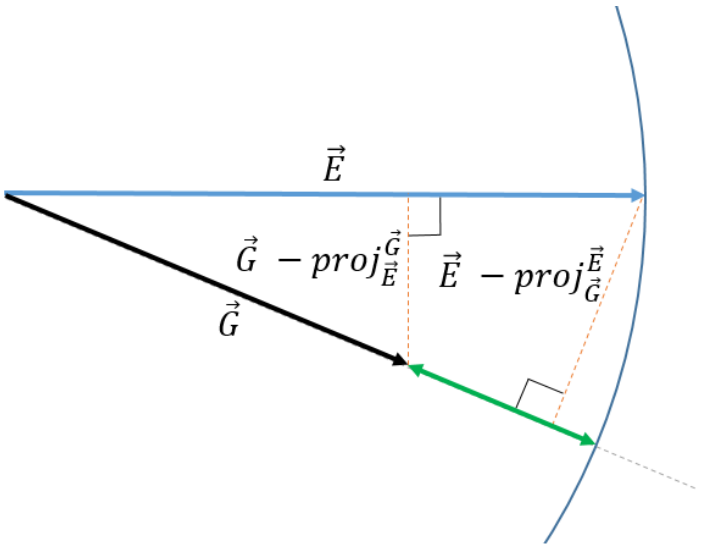


Figure 5.2: The angular distance between the two non-null vectors $\vec{E}$ and $\vec{G}$ based on the perpendicular distance between both vectors.

5.3.4 Normalized End Point Error (NEPE)

EPE metric is takes into consideration the magnitude of the difference between two vectors and ignores the magnitude of each vectors in the sense that the EPE metric gives the same value for case 1 and case 2, in which the radius of two circles is the same (refer to **Figure** 5.1). The following metric is an enhancement of the magnitude error E_M proposed by [84]:

$$NEPE = \begin{cases} \frac{\sqrt{(u-u_G)^2+(v-v_G)^2}}{min\Big((u^2+v^2),({u_G}^2+{v_G}^2)\Big)}, \\ \quad \text{if} \quad min\big((u^2+v^2),({u_G}^2+{v_G}^2)\big) > \epsilon \\ \frac{\sqrt{(u-u_G)^2+(v-v_G)^2}}{\epsilon}, \\ \quad \text{if} \quad min\big((u^2+v^2),({u_G}^2+{v_G}^2)\big) \leq 0 \end{cases} \tag{5.11}$$

where, ϵ is a threshold around 0.01.

5.3.5 Enhanced Normalized End Point Error (ENEPE)

One way to get over EPE drawbacks is to calculate the relative distance between $\vec{E}$ and $\vec{G}$ vectors and to use different normalization methods as in the following:

$$ENEPE1 = \begin{cases} \frac{\sqrt{(\|\vec{P_G}\|)^2+\tau(\|\vec{N_G}\|)^2}}{min\Big((u^2+v^2),({u_G}^2+{v_G}^2)\Big)}, \\ \text{if} \quad min\big((u^2+v^2),({u_G}^2+{v_G}^2)\big) > \epsilon \\ \frac{\sqrt{(\|\vec{P_G}\|)^2+\tau(\|\vec{N_G}\|)^2}}{\epsilon}, \\ \text{if} \quad min\big((u^2+v^2),({u_G}^2+{v_G}^2)\big) \leq 0 \end{cases} \tag{5.12}$$

If the normalization is performed by only $\vec{G}$ vector, then

$$ENEPE2 = \begin{cases} \frac{\sqrt{(\|\vec{P_G}\|)^2+\tau(\|\vec{N_G}\|)^2}}{\sqrt{u_G^2+v_G^2}}, \\ \quad \text{if} \quad (u_G^2+v_G^2) \neq 0 \\ \sqrt{u^2+v^2}, \\ \quad \text{if} \quad (u_G^2+v_G^2) = 0 \end{cases} \tag{5.13}$$

If the normalization is performed by the average of $\vec{G}$ and $\vec{E}$ vectors, then

$$ENEPE3 = \begin{cases} \frac{2\sqrt{(\|\vec{P_G}\|)^2+\tau(\|\vec{N_G}\|)^2}}{\sqrt{u_G^2+v_G^2}+\sqrt{u^2+v^2}}, \\ \quad \text{if} \quad (u_G^2+v_G^2) \neq 0 \\ \sqrt{u^2+v^2}, \\ \quad \text{if} \quad (u_G^2+v_G^2) = 0 \end{cases} \tag{5.14}$$

If normalization is ignored, then

$$ENEPE4 = \sqrt{(\|\vec{P_G}\|)^2+\tau(\|\vec{N_G}\|)^2} \tag{5.15}$$

where τ is strictly positive value, and it works as steering power for normal component $\vec{N_G}$ and $\vec{P_G}$ and $\vec{N_G}$ are defined as

$$\vec{P_G} = \frac{(uu_G+vv_G)}{(u_G^2+v_G^2)}\vec{G} - \vec{G} \tag{5.16}$$

$$\vec{N_G} = \vec{E} - \frac{(uu_G+vv_G)}{(u_G^2+v_G^2)}\vec{G} \tag{5.17}$$

5.4 Results and Discussion

Systematic experiments have been conducted to evaluate optical flow performance. As we are evaluating 10 different metrics, a total number of 3360 experiments were

Metric	Setting
GPRE	$\alpha = \beta = 0$
NEE	$\epsilon = 0.01$
ENEE1	$\epsilon = 0.01, \tau = 3$
ENEE2	$\tau = 100$
ENEE3	$\tau = 100$
ENEE4	$\tau = 5$

Table 5.2: Metric settings used in all experiments.

performed for each dataset. Behavior and sensitivity of every metric have been reported for motion variations in horizontal, vertical, rotational and magnification - or a combination. Parameter settings used in all experiments are summarized in **Table** 5.2.

As a rule of thumb, a good metric has to produce an error value proportional to the absolute values in step sequence S described in the previous section. A general overview of mean error curves for existing and proposed error metrics in $\log$ scale is illustrated in **Figure** 5.3. It is obvious that some metrics outperform others, but it is not yet clear which metrics are more suitable for optical flow performance measurement. More detailed explanations and results are reported in the following sections.

5.4.1 Metrics Evaluation of the Baker Dataset

Metrics calculate errors between G and modified G. The most general example of a modified G is when G values are shifted horizontally and vertically and then rotated after magnification by a value. For instance, **Figure** 5.4 shows mean error metric curves for the Baker dataset. $X - axis$ represents values used to shift, rotate, and magnify actual G, while $Y - axis$ is the mean error values.

Based on our rule of thumb, **Figure** 5.4 shows that LPE, ENEE4, and EPE metrics are more sensitive to motion variation when G is modified with negative values, while NEE and ENEE1 are more sensitive to motion variation when G is modified with positive values.

According to the approach used in modifying G, no motion pixels are replaced

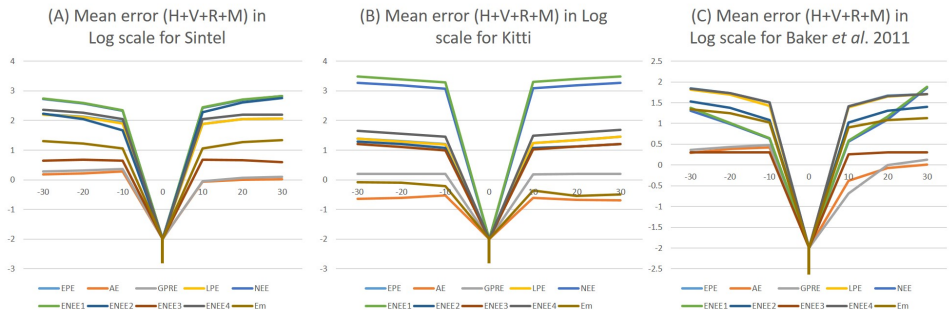


Figure 5.3: Mean error ($y - axis$) in log scale for all metrics between G and modified G in different scenarios: (I) when G are shifted horizontally(H) by the number of pixels in $x - axis$, (II) G are shifted vertically (V) by the number of pixels in $x - axis$, (III) G are magnified (M) by values in $x - axis$, (IV) G are shifted horizontally and vertically by the number of pixels in $x - axis$, (V) G are rotated (R) by the degree of the degree in $x - axis$, (VI) G are shifted horizontally and vertically, then rotated by values of $x - axis$. This applies to (A) the Sintel dataset, (B) the Kitti dataset, and (C) the Baker dataset. Note that $\log(0^+) = -\infty$, which is represented by the lowest point in the graph.

with zero values when G is rotated; hence, this will increase zero values in modified G and mean error will be biased. To overcome this issue, the third quartile of the error can be used instead of mean error. The third quartile is denoted by Q3, which is the median of the upper half of the data set. This means that about 75% of the numbers in the data set lie below Q3 and about 25% lie above Q3.

Since it is not clear from mean error which metric is better, Q3 mean error gives a clearer idea about the best metrics. **Figure** 5.5 illustrates Q3 mean error for all metrics. It is obvious that NEE and ENEE1 metrics are outperform other metrics.

The second best are ENEE4, EPE and LPE metrics, and ENEE2 and Em metrics are third.

Visualization of optical flow error for the Hydrangea sample, which is part of Baker dataset, is shown in **Figure** 5.8 and indicates that NEE and ENEE1 metrics are compromised metrics between EPE because they greatly penalize errors; AE penalizes errors less.

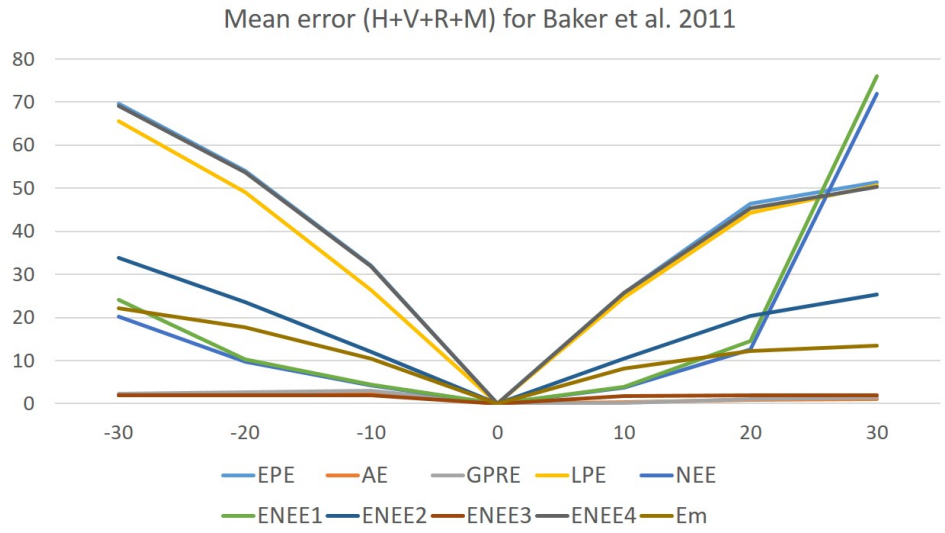


Figure 5.4: Mean error ($y - axis$) for the Baker dataset for all metric calculations between G and modified G when they are shifted horizontally and vertically and then rotated after that magnified by values of $x - axis$.

5.4.2 Metrics Evaluation on KITTI Dataset

The second evaluation was conducted on the KITTI dataset. The mean error of existing and proposed metrics are shown in **Figure** 5.6. It is clear that ENEE1 and NEE metrics are more sensitive to motion variation than others.

Optical flow error visualization for the sample image of the KITTI dataset is shown in **Figure** 5.9. A compromised visualization between EPE and AE metrics is represented by NEE and ENEE1 metrics.

5.4.3 Metric Evaluation on Sintel Dataset

The last evaluation for metrics was performed on the Sintel dataset. The mean error of all metrics is plotted in **Figure** 5.7. Based on our rule of thumb, NEE and ENEE1 metrics are producing error values more proportional to the absolute value of motion change. Hence, NEE and ENEE1 metrics are more sensitive to errors and performing better than other metrics.

Visualization of optical flow error as sample image of Sintel dataset is shown

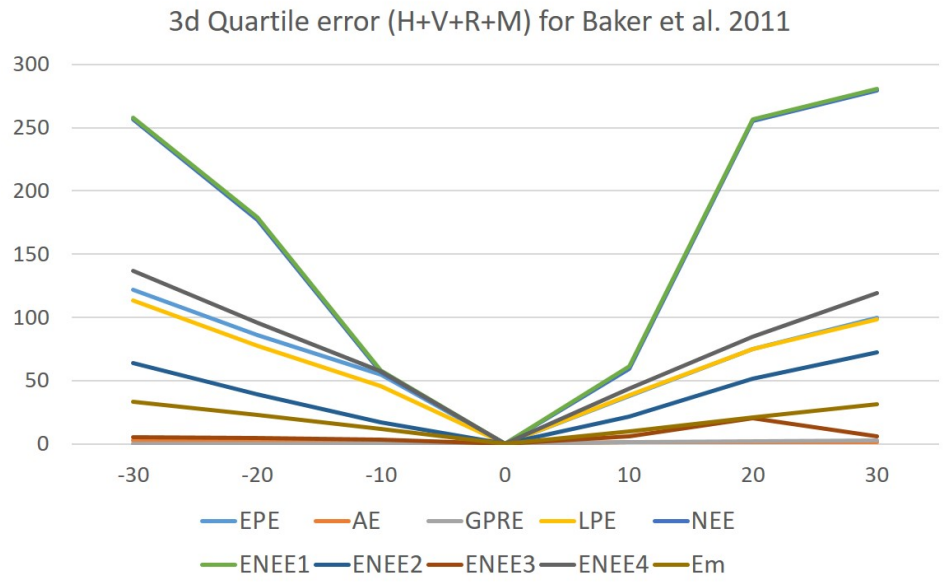


Figure 5.5: Third quartile of mean error ($y - axis$) for the Baker dataset for all metrics calculating error between G and modified G when motion is shifted horizontally and vertically then rotated magnified by values of $x - axis$.

in **Figure** 5.10. This indicates that NEE and ENEE1 metrics are moderate versions between between EPE which highly penalize errors and AE which less penalize errors.

5.4.4 Discussion

A qualitative assessment [15] was conducted on two common error metrics, EPE and AE, and suggested using EPE rather than AE based on only one sample from the Baker dataset from Urban sequence. However, there is a need for a systematic evaluation of optical flow performance; thus this experiment was conducted on three popular datasets using ten different error metrics. A good metric is considered to be more sensitive to errors, for example, producing error values proportional to the change of motion between modified-G and G.

Existing metrics such as EPE, AE and EM have sensitivity differ slightly from

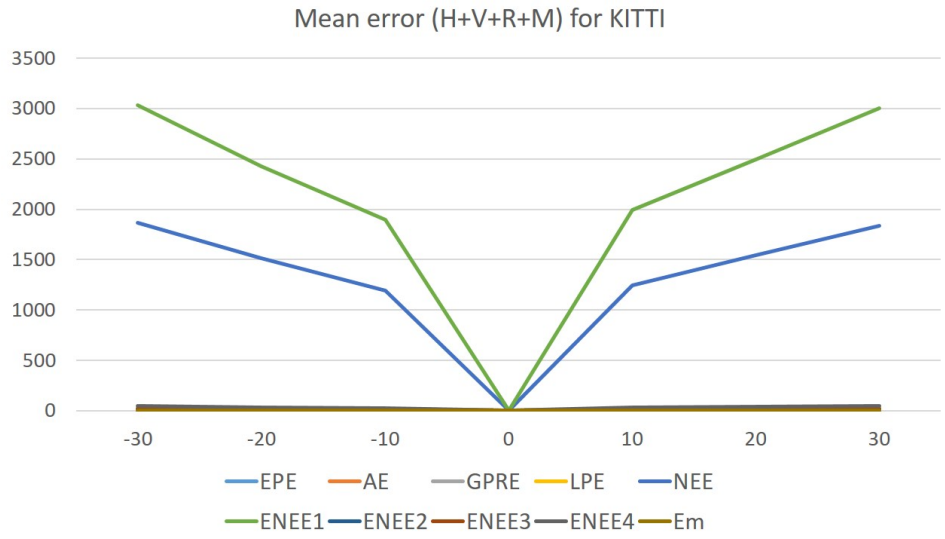


Figure 5.6: Mean error ($y - axis$) for KITTI dataset for all metric calculations between G and modified G when they are shifted horizontally and vertically then rotated after that magnified by values of $x - axis$.

one dataset to another. For instance, EPE and EM performed well on Baker, while AE and Em are less sensitive on Kitti and AE is not sensitive on Sintel. The best EPE sensitivity was on on Kitti. AE sensitivity was the worst among the three metrics.

A detailed look into metrics behavior related to motion change is illustrated in **Figure** 5.11. The following observations have been derived:

- It is observed from **Figure** 5.11 (A,B, and C) that almost all metrics except ENEE2 and Em are sensitive to horizontal, vertical, and (horizontal and vertical) motion variation, with some differences in magnitude. ENEE2 and Em metrics are more sensitive for horizontal variation **Figure** 5.11(B) and horizontal and vertical variation **Figure** 5.11(C).

- All metrics except AE, GPRE, and ENEE3 are sensitive to magnitude of motion variation. AE, GPRE, and ENEE3 metrics cannot detect variations in motion magnitude, as shown in **Figure** 5.11(E).

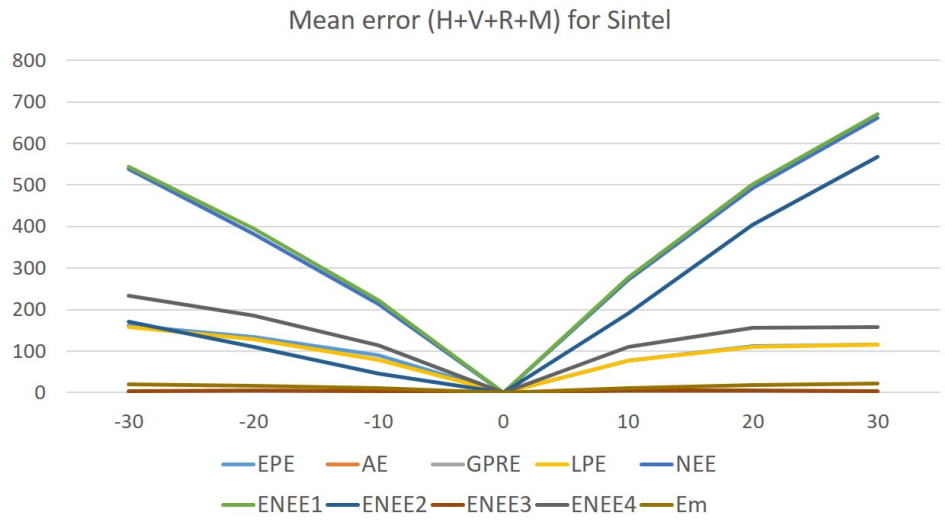


Figure 5.7: Mean error ($y - axis$) for Sintel dataset for all metric calculations between G and modified G when they are shifted horizontally and vertically then rotated after that magnified by values of $x - axis$.

- NEE and ENEE1 metrics are sensitive for angle variation as seen in **Figure** 5.11 (D, F), while AE, GPRE, and Em are sensitive only for small rotational variation.

Based on the previous observations, we conclude that all metrics are sensitive to horizontal, vertical, and (horizontal and vertical) variation. AE, GPRE, NEE and ENEE1 metrics are sensitive to rotational variations. All metrics except AE and GPRE are sensitive to magnitude changing in motion. And only NEE and ENEE1 metrics are sensitive to all horizontal, vertical, rotational, magnitude or a combination. These results are summarized in **Table** 5.3.

5.5 Conclusion

In this chapter, a novel performance measure of optical flow has been proposed. Moreover, a systematic evaluation of optical flow performance has been conducted. Drawbacks of existing performance metrics have been identified. Among the five

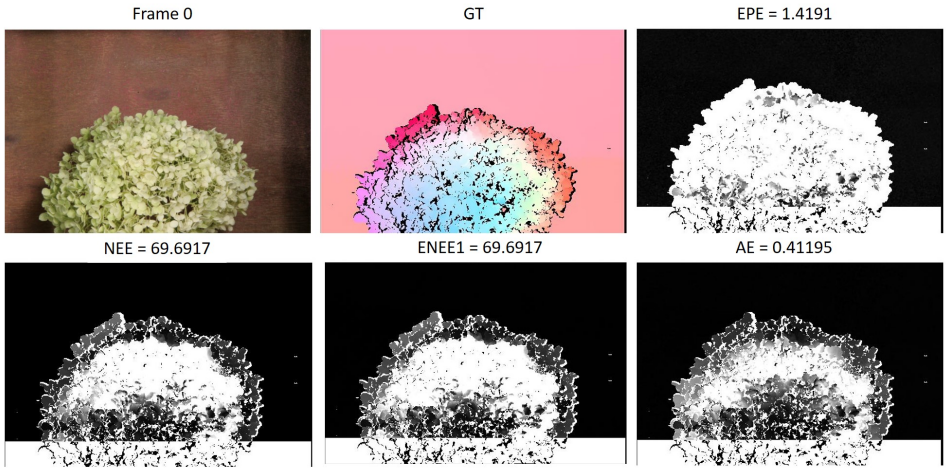


Figure 5.8: Sample image from Bakers' (Hydrangea) dataset, the corresponding ground truth and the visualization of motion error for four different error metrics (EPE, AE, NEE and ENEE1) between ground truth and modified ground truth when G pixels are shifted vertically by -50 pixels.

proposed optical flow performance metrics, NEE and ENEE1 error metrics have outperformed all others, including the existing ones. The sensitivity of NEE and ENEE1 to motion variation is very significant, indicating that the use of NEE and ENEE1 error metrics is strongly recommended for measuring the performance of estimated optical flow with regard to ground truth.

Figure 5.9: Sample image from KITTIs' dataset, the corresponding ground truth and the visualization of motion error for four different error metrics (EPE, AE, NEE and ENEE1) between ground truth and modified ground truth when G pixels are shifted vertically by -50 pixels.

	V	H	H+V	R	M	H+V +R	H+V +R+M
EPE	✓	✓	✓		✓		
AE	✓	✓	✓	✓			
GPRE	✓	✓	✓	✓			
LPE	✓	✓	✓		✓		
NEE	✓	✓	✓	✓	✓	✓	✓
ENEE1	✓	✓	✓	✓	✓	✓	✓
ENEE2	✓	✓	✓		✓		
ENEE3	✓	✓	✓		✓		
ENEE4	✓	✓	✓		✓		
Em	✓	✓	✓		✓		

Table 5.3: Summarized results for our rule-of-thumb method to choose best metric based on metric sensitivity to motion variation in horizontal (H), vertical(V), rotational(R), and magnification (M) or a combination.

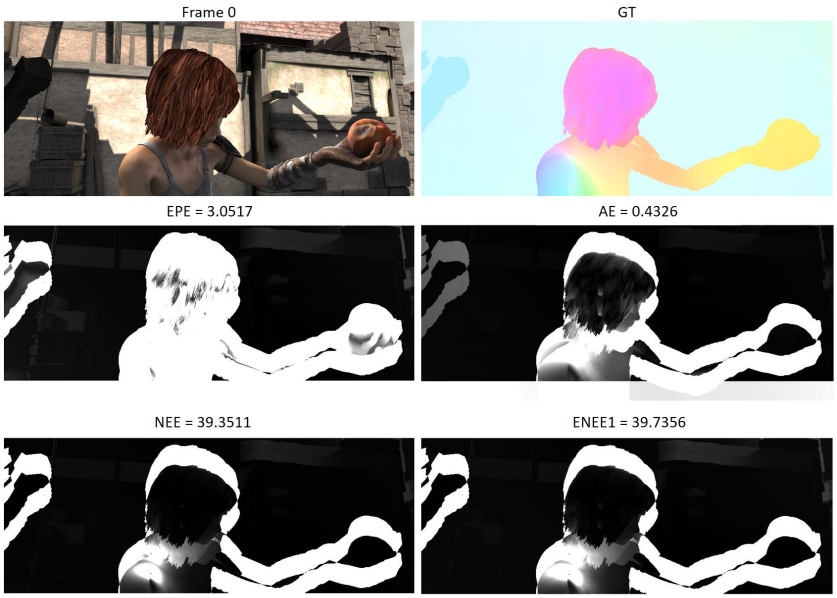


Figure 5.10: Sample image from Sintels' dataset, the corresponding ground truth and the visualization of motion error for four different error metrics (EPE, AE, NEE and ENEE1) between ground truth and modified ground truth when G pixels are shifted vertically by -50 pixels.

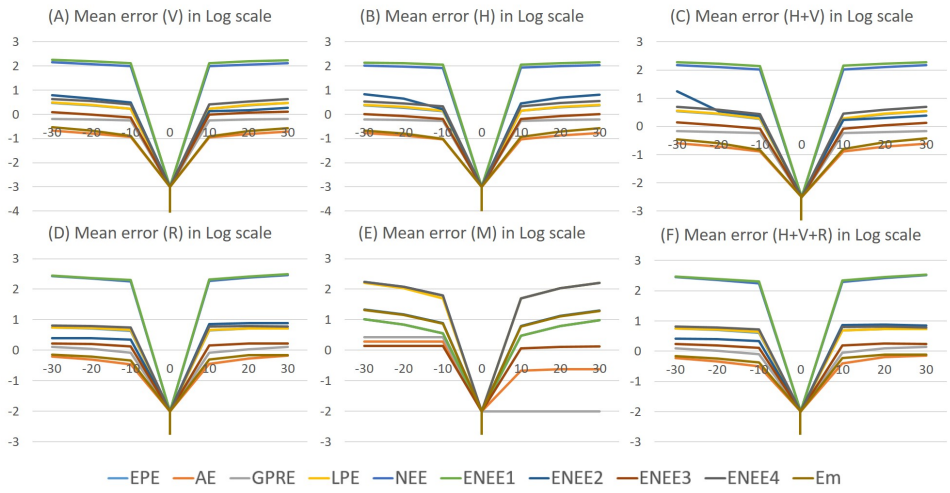


Figure 5.11: All datasets mean error ($y-axis$) in $\log$ scale for all metrics between G and modified G in different scenarios: (A) when G are shifted horizontally(H) by number of pixels in $x-axis$, ; (B) G are shifted vertically (V) by number of pixels in $x-axis$; (C) G are magnified (M) by values in $x-axis$; (D) G are shifted horizontally and vertically by number of pixels in $x-axis$; (E) G are rotated (R) by angle degree in $x-axis$; (F) G are shifted horizontally and vertically then rotated by values of $x-axis$. Note that $\log(0^+) = -\infty$, which is represented by the lowest point in the graph.

Part III

Human Activity Recognition

CHAPTER 6

Learning Human Activities

Human behavior analysis tasks are classified according to the degree of semantic as follow: motion, action, activity and behavior [90]. From one hand, motion has the lowest degree of semantic while behavior has the highest one. This is based on the fact that to move, and to capture the motion, requires the shortest period or time. To document the behavior of motion, however, a long period of motion capturing is needed. Motion information over time produces action, different interactions construct an activity, and more complex activities shape a behavior.

Recognizing human activity can be based on different sensor modalities, the most common ones are visual and inertial sensing. These modalities can be used simultaneously or independently. Inertial measurement sensors (IMUs) are devices with capabilities to measure and report the body's specific force, angular rate, and orientation. In the sensor's local coordinate system, there are three main measurement modalities: *accelerometers*, which capture instantaneous acceleration for each axis; *gyroscopes*, which represent the rotational velocity of the inertial; and *magnetometer*, which exemplify the instantaneously measured magnetic field with corresponding *X, Y* and *Z* axes. One of the drawbacks of using IMUs is the high degree of uncertainty when measuring at slower motion velocities and lower relative uncertainty at high velocities. On the other hand, inertial sensors are able to measure very high velocities and accelerations.

Depending on their methodological nature, two main approaches can be applied on egocentric activity recognition: object and motion-based approaches [19, 90]. Object-based approaches deal with various types of information about objects and their interaction with hands. A relationship between object, hand, location,

and pose will be instantiated to recognize an activity. However, motion-based approaches depend on the camera's location on the subject's body, i.e., where the camera is mounted vis-à-vis the subjects' head, shoulder, or chest.

One of the motion-based approaches is optical flow, which refers to the displacement of intensity patterns [47]. It represents the motion of visual features such as points, objects, shapes, *etc.* via continuous view of the environment to produce a relative motion representation between environment and observer [5].

Optical flow is computed from visual sensing and can be defined as the apparent motion of objects in consecutive frame pairs. It can be subcategorized into forward optical flow, when the displacement vector for each pixel of the first frame has been computed, or backward optical flow, when it is estimated from the second frame. A field of vectors will be generated in u and v directions.

Research paradigms in the optical flow field are divided into two main methods. The first considers optical flow estimation as a classical problem [25] and is considered as a variational optimization problem to find pixel correspondences between any two consecutive frames [54]. The second can be formulated as a machine learning problem, an example of which is convolutional neural networks (CNN) [43, 125, 111, 9]. Activity frames can be modeled as 3D volumes in time, and various local visual descriptors can be extracted such as histograms of oriented gradients (HOG), histograms of optical flow (HOF), and motion boundary histograms (MBH) [88, 35, 118, 119].

The recognition of human activities can be performed using classifiers trained by IMUs features alone, visual features alone, or a combination of the different sensor modalities [30, 7, 2, 42, 6]. In this research, learning human activities was conducted via two main methods: support vector machines and deep learning using features extracted from only two IMUs sensors. These were placed at the subject's left and right hands, with egocentric vision corresponding to similar, complex, and opposite human activities.

In this presentation of the sections of the research, the following contributions of the endeavor are highlighted:

- Developed an action extractor tool for both visual and IMUs data based on [137] annotations.

- The use of deep learning for feature extraction and training for visual data.
- The introduction of a novel statistical feature extraction method for IMUs data based on curvature of function graph and tracking the positions of left and right hands in space.
- The provision of an experimental proof of the limitation of IMUs data to distinguish activities and the suggestion that visual features can be complementary to IMUs in human activities recognition.
- The performance of intermediate fusion between IMUs and visual sensors in order to recognize actions using SVM classifier.

6.1 Related Work and Contribution

Different sensor modalities can be used for recognizing human activity. Two of the most common sensing modalities are visual and inertial, which can be used simultaneously or independently.

Inertial measurement sensors can be used alone for human activity recognition [132, 20, 63, 12, 112, 89, 13, 37]. IMUs can measure various body signals such as force, angular rate, and orientation. Measurement by means of IMUs suffers from a high level of uncertainty at slow motion and lower relative uncertainty at high velocities. For example, convolutional long short-term memory (LSTM) was used to solve sequential human activity recognition problems through proposing a multi-level neural network structure model based on the combination of an inception neural network and gated recurrent units (GRU) [132]. The best F-measure score achieved was 94.6% on the Opportunity dataset. Bevilacqua *et al.* [20] used convolutional neural networks (CNNs) to classify human activities. They used the Otago exercise program dataset, which contains 16 activities based on five sensors placed on subjects: two sensors placed on the distal third of each shank(left and right), two sensors centered on both left and right feet and one sensor placed on the lumbar region.

In their work, Jalloul *et al.* [63] constructed a structural connectivity network to explore the relations between the sensing modules while performing activities

based on the correlation between some wearable sensing modules. These were positioned at different parts of the body and constitute a monitoring system for four different activities (walking, standing, lying and sitting).

LSTM also was used by [12] to classify seven main activities with different motion primitives recorded using Apple watch. A hybrid deep framework based on LSTM and an extreme learning machine (ELM) was proposed by Sun *et al.* [112] to overcome the problem of sequential activity recognition. The proposed framework was composed of convolutional layers, LSTM recurrent layers, and an ELM classifier, which can automatically learn feature representations and model the temporal dependencies between features. Their framework has been evaluated on an Opportunity dataset with 17 different gestures, for which it achieved a 91.8% F_1 score with all classes, including a null class and 90.6% without using a null class.

Another work, conducted by Rueda *et al.*, deploys CNN using a multi-channel time series for activity recognition and evaluated their model on Opportunity, Pamap2, and Order Picking datasets. A data-driven architecture based on an iterative learning framework was proposed by Davila *et al.* [37] to classify human locomotion activities, such as walking, standing, lying, and sitting extracted from the Opportunity dataset using multi-class SVM classifiers. Their framework produced an average accuracy of 74.08% while using only 6.94% of the samples in the input domain for training compared to average accuracy of 81.07% obtained by the supervised method when using 80% of samples for training, and the 20% remaining samples for testing.

It should be noted that only visual data can be used in activity recognition. For instance, RGB-D data have been used in deep learning to recognize human activities [60, 33]. Sudhakaran *et al.* [109] presented a hierarchical feature lightweight aggregation scheme that can be plugged into any deep architecture with CNN backbone. At each layer, the feature from a CNN block is gated and its residual is transferred to the adjacent branch. They evaluated their proposed technique on Something-v1, EPIC-KITCHENS, and HMDB51 datasets. The results obtained show an improvement of about 24% compared to a Temporal Segment Network (TSN). To extract more precise object-related features to guide 3D CNN training, Wang *et al.* [123] introduced the Baidu-UTS object detection model that consists

of two parts: the first is a 3D CNN branch that takes sampled video clips as input and produces a clip feature. The second extracts the object-related features from the context frames. An EPIC-KITCHENS dataset was used in their research to predict verbs, nouns, and activities from the vocabulary for each video segment. Sudhakaran *et al.* [110] proposed long short-term attention (LSTA), a recurrent unit that addresses shortcomings of LSTM when the discriminative information in the input sequence can be spatially localized. Moreover, they deployed LSTA in a two-stream architecture with cross-modal fusion and evaluated their method on four datasets: GTEA 61, GTEA 71, EGTEA Gaze+, and EPIC-KITCHENS. Their network was trained for multi-task classification with verb, noun and activity supervision. Activity classifier activation was used to control the bias of verb and noun classifiers. Other information extracted from visual data can also be used for activity recognition applications.

Optical flow can be derived from visual sensing. It represents the apparent motion of objects in consecutive frame pairs. The displacement vector for each pixel of the first frame is called optical flow forward and from the second frame back to the first frame is called backward optical flow, which forms a field of vectors in u and v directions. The interaction between optical flow and activity recognition has been discussed in [103].

The success of optical flow in many activity recognition applications [113, 5, 71, 128] is not due to its temporal structure. However, optical flow [113, 5, 71, 128], deep-learned spatial descriptors [105], and dense trajectories based on motion boundary histograms [121] are considered to be invariant to appearance of the representation [103].

A significant amount of research work suggests combining different sensor modalities to improve human activity recognition [30, 7, 2]. However, for real-life scenarios, realistic and compromised number of modalities should be used. Lu and Velipasalar [79] deployed LSTM to classify activities using four IMUs sensors corresponding 36 components with egocentric video from the CMU Multimodal Activity (CMU-MMAC) database [38]. Visual and audio sensors were used by [11] for activity recognition. [2] has presented a framework for recognizing proprioceptive activities using IMUs egocentric data. They used cross-domain knowledge transfer with a CNN-LSTM to exploit discriminative characteristics of multimodal

feature groups provided by stacked spectrograms from the inertial data. Diete and Stuckenschmidt [42] have used visual features with objects information and inertial data for human activity recognition.

6.2 Dataset

The Carnegie Mellon University Multi-Modal Activity (CMU-MMAC) [38] main database has been used to train and test our activity recognition methods. This database contains human activity measures constructed from multimodal sensors mounted on subjects while performing tasks related to cooking and food preparation in Carnegie Mellon's Motion Capture Lab. More than forty subjects were involved in the preparation of recipes for five types of food:

- Brownies.
- Pizza.
- Sandwich.
- Salad.
- Scrambled eggs (Eggs).

Various modalities were used to record the following data types:

1. Video:

   Three video cameras with high recording resolution (1024 x 768) with temporal resolution (30 Hertz).

   Two video cameras with 60 Hertz temporal resolution and with spatial resolution equals 640 x 480.

   One wearable camera with 30 Hertz temporal resolution. And equipped with high spatial resolution $(800 \times 600/1024 \times 768)$.

2. Audio:

   Five microphones.

3. Motion Capture:

   Motion capturing system consists of 12 infrared MX-40 cameras. Each of which is recording images at four megapixel resolution at 120 Hertz.

4. Internal Measurement Units (IMUs):

   Wired IMUs (3DMGX).

   Bluetooth IMUs (6DOF).

5. Wearable devices:

   BodyMedia.

   eWatch.

This dataset was collected using 55 subjects, with each participating in several subexperiments.

6.2.1 Action Extraction

A Matlab tool has been developed [1] to extract the corresponding actions for both IMUs and visual data from the CMU Multi-Modal Activity database (CMU-MMAC) [38] based on [137] annotations. The action extraction tool is illustrated in **Figure** 6.1. The user provides IMUs ID, video, and subject in order to extract all actions based on the annotation file provided for each subject. Data extracted for both modalities is synchronized using start and end times that are provided. To facilitate processing extracted files; each extracted file or image name contains a prefix for subject ID, serial number, and action name.

6.2.2 CMU-MMAC Annotations [137]

The most recent set of annotations for CMU-MMAC dataset has been published by [137], showing an increased number of labels for different types of scenarios. In their paper, [137] have explained the approach used for annotating the CMU-MMAC dataset. Moreover, they have offered semantic annotations that can be used in experiments related to reasoning. The annotations mainly focus on three

[1] `https://github.com/alhersh/ActionExtractor`

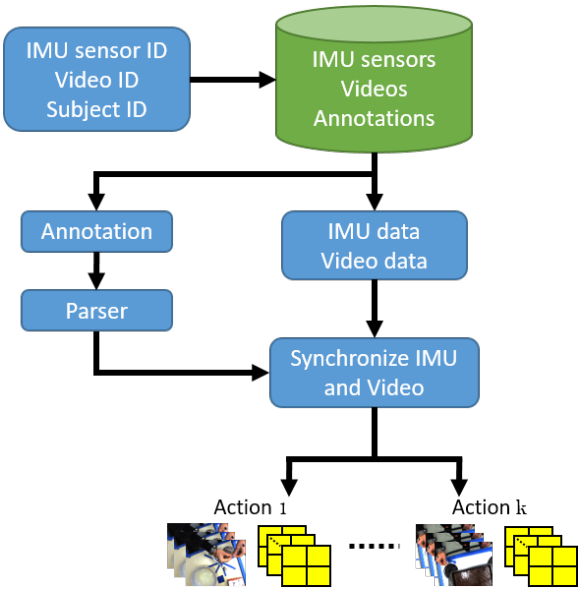


Figure 6.1: Action-extractor tool. User has to identify the following parameters to be passed to the Action-extractor tool: IDs for IMUs sensor, subject, and video to extract the corresponding images, and IMUs data for all actions provided based on an annotation file.

recipes: Brownie, Scrambled eggs (Eggs), and Sandwich, in which they used all valid subjects related to those recipes. These new annotations are based on egocentric vision recorded by the first-person camera.

Activity classes are enumerated as eleven for Brownie, eleven for Eggs, and eight for Sandwich. A derivative from these activity classes are activities based on *verb-Object1-object2-...-object_n*, in which the number of objects differs from one activity to another, for example *close_drawer* and *shake-butter_spray_can*.

6.3 Methods

This section provides information about methods used in this part of the research, starting from feature extraction methods, then moving through feature fusion, and finally on to the classification techniques used.

6.3.1 Feature Extraction

The main purpose of feature extraction is to reduce the dimension of high-dimensional raw data, hence producing more manageable groups of data for processing that consequently reduce the computational power required. The feature extraction method can select and /or combine different variables together to shape features, effectively reducing the data while nonetheless describing the original dataset in an accurate and complete way. Next, an illustration of the feature extraction methods used in this research is provided.

Deep learning

GoogLeNet [114] was used to extract features from videos. This network is a convolutional neural network that is 22 layers deep, counting only the layers with parameters. In total, about 100 layers were used to construct it. The length of the feature vector extracted is 1024 for each frame.

The convolutional network of GoogLeNet [114] was used as a feature-extraction tool by obtaining activations from inputted video frames to the network. Hence, videos were converted to feature vector sequences, in which feature vectors are basically the output of the activation function on the last pooling layer ("pool5-7x7_s1") of the GoogLeNet [114] network, as illustrated in **Figure** 6.2.

Figure 6.2: Data flow diagram that illustrates obtaining activations.

In this part of the research, we used only the Brownie, Scrambled eggs (Eggs), and Sandwich recipes. The name and distribution of classes in the Brownie dataset is shown in **Figure** 6.3.

IMUs Statistical Features

In this part of the research, we considered nine activities with three opposing pairs including

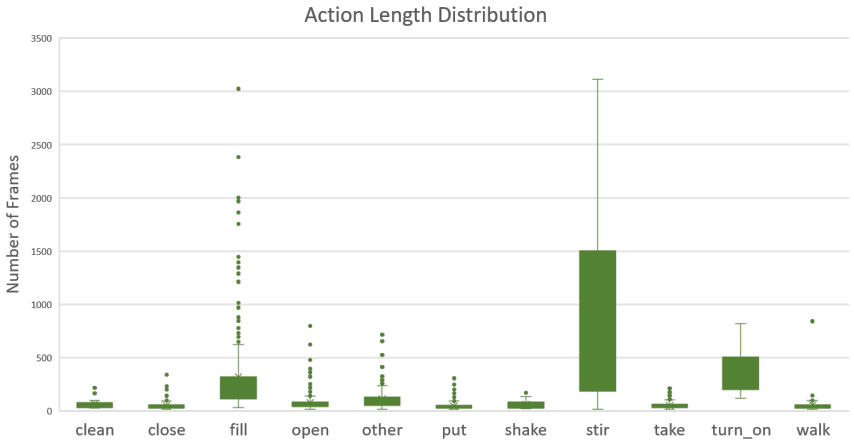


Figure 6.3: Distribution of the number of frames for activities considered from the CMUMMAC Brownie dataset. The activity name has been derived from the verb part of the annotated label of activity.

- close-bread-bag/open-bread-bag; close-drawer/open-drawer; and close-fridge/open-fridge.
- fill-oil-oil-bottle-pan.
- shake-butter-spray-can.
- stir-bowl-fork.

Statistical features for IMUs activities were extracted as shown in **Figure** 6.4. These were based on a sliding window of size 25 points, which is equal to 0.2 seconds, and with 40% overlap to produce a feature vector of 504 length as follows

1. Define left and right sensors.
2. For each left and right hand sensor, the positions of the left and right hands is tracked using [130], and the first derivative of the position for each hand is calculated.
3. The difference between the various combinations of left and right hands in three-dimensional space is calculated.

4. The curvature of (X, Y, Z) acceleration and gyro data was computed using the following formula:

$$k = \frac{|y''|}{(1 + y'^2)^{\frac{3}{2}}} \tag{6.1}$$

5. Normalize previous values using the formula:

$$f(x) = \frac{x - \mu}{\sigma} \tag{6.2}$$

, where μ is the mean and σ is the variance.

6. Mean, variance, entropy, kurtosis, moment (order 3 and 4), and the number of local maxima (peaks).

Visual Descriptors

Literally, optical flow refers to the displacement of intensity patterns [47]. Theoretically, it is the motion of visual features such as points, objects, shapes etc. through a continuous view of the environment. It represents the motion of the environment relative to an observer [5]. Optical flow generated can be processed in many methods for different applications.

This section discusses the approach aligned with this part of the research. Motion-based features of optical flow can depend on oriented histograms of various kinds of local differences or differentials. For example, the histogram of oriented gradients (HOG), histogram of optical flow (HOF), and motion-boundary histograms (MBH) [88, 35, 118, 119].

The HOG method tiles a detector window with a dense grid of cells, with each cell containing a local histogram over orientation bins. At each pixel, the image gradient vector is calculated and converted to an angle, voting into the corresponding orientation bin with a vote weighted by the gradient magnitude. Votes are accumulated over the pixels of each cell. The cells are grouped into blocks, and a robust normalization process is run on each block to provide strong illumination invariance. The normalized histograms of all of the blocks are concatenated to give the window-level visual descriptor vector for learning [35].

We adopted the method used in [118, 119] method for extracting local visual

descriptors HOF, HOG, and MBH for each activity in the video, with two main modifications; the first one is increases block size from 8 pixels to 50. According to [34, 35], the recommended values for the HOG parameters are:

- Detection-window size of 64×128
- Block size of 16×16

Since we are using a video resolution of (300×400), the detection window has been increased by a ratio of $468.75\% \times 312.5\%$, which increases the block size to 50×50 pixels. In the second, descriptors for each activity were aligned with IMUs features to produce a feature vector with a length of 144 for each activity. Also, in this part of research, the activities considered are the same as mentioned in the previous section.

6.3.2 Feature Fusion

A significant amount of research work suggests combining different sensor modalities to improve human activity recognition [30, 7, 2]. However, for real-life scenarios, a realistic and compromised number of modalities should be used.

Sensors fusion can be performed in three main levels. The first is low level, in which raw data from different sources are combined to produce new more informative data than the inputs. The second is the intermediate level, which combines various features from different sensors together to build a feature map. The last is high level, also called the decision level, which combines decisions from several experts using various methods such as voting, fuzzy logic, and statistical methods [36, 44].

The advantages provided by combining local visual descriptors and IMUs data are the complementary characteristics of visual descriptors and inertial sensors. For instance, IMUs data have large measurement uncertainty at slow motion and lower relative uncertainty at high velocities. Inertial sensors can measure very high velocities and accelerations. On the other hand, visual descriptors can track features very accurately invariant to appearance of the representation at low velocities. For high velocity, tracking is less accurate since the resolution must be reduced to obtain a larger tracking window with the same pixel size and, hence, a higher tracking

velocity [99].

For the sensor-fusing experiment, we used an intermediate level of sensor fusion since we are fusing features generated from IMUs and visual data to produce a feature vector with a length of 648 for each activity used in this experiment as shown in **Figure** 6.4. Activities used in this part of research are the same used in the previous two sections.

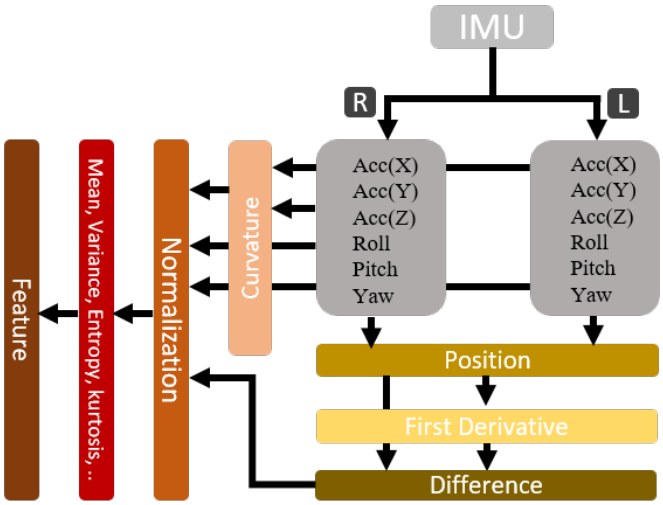


Figure 6.4: IMUs feature extraction procedure.

6.4 Classification

6.4.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression problems. It is a discriminative classifier formally defined by a separating hyperplane. For example, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts in which each class lies in either side. Nevertheless, it is mostly used in classification problems [32].

SVM algorithms use a set of mathematical functions that are defined as kernels. The function of a kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions, for example, linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. In this work we used the Cubic SVM classifier in our research. Based on the Matlab 2018b Classification Learner App, the following settings were used for the classification process:

- Kernel function - cubic polynomial kernel given the following formula:

$$k(x_1, x_2) = (x_1^T x_2 + 1)^3 \tag{6.3}$$

- Kernel scale - automatic
- Box constraint level: 1
- Multiclass method - one-vs-one
- Standardize data - true

Data fetched into the classifiers was split into 80% for training and 20% for testing. We used five-fold cross validation for training.

6.4.2 Recurrent Neural Networks (RNN)

In recurrent neural networks (RNN), previous information could be used to predict current or future information, for example, using multiple and sequential video frames to predict, recognize, or classify the whole chunk of frames.

Bidirectional LSTM (BiLSTM) is considered an extension of RNN [102] in which the layer learns bidirectionally and there are dependencies between time steps of a sequence of images, data, or time series. The significance of these dependencies is used when the network is needed to learn from the complete time series at each time step, as shown in **Figure** 6.5. For this reason, we used BiLSTM in our experiment to classify activities that are considered to be a sequence of images in time. The error backpropagated through time, but layers can be preserved in BiLSTM. Maintaining numerous constant errors allows recurrent nets to continue to learn over many time steps.

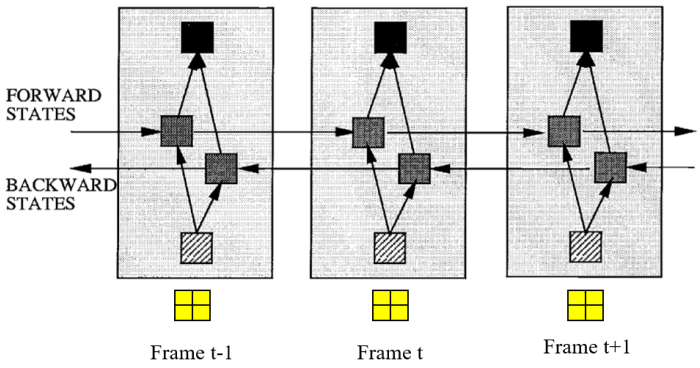


Figure 6.5: BiLSTM architecture based on [102].

An overview of the method used for classifying human activity using only visual data is shown in **Figure** 6.6. Input data is activity videos, features are extracted using GoogLeNet. Activities are extracted and then trained using a Bidirectional Long short-term memory (BiLSTM) network for human activity recognition.

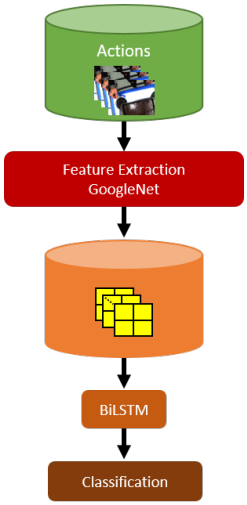


Figure 6.6: An overview of general human activities classification approach. Input data is activity videos, and features are extracted using GoogLeNet, producing a features database that is used for training a recognition model using BiLSTM.

6.5 Experiments

6.5.1 IMUs and Optical Flow

As an initial experiment, the feasibility of using visual features from optical flow and IMUs was conducted; four activities were used in this experiment. **Figure** 6.10 shows the feature vector differences between various sensors for various activities. It is obvious that the produced features can distinguish between "take-oil", "put-baking", and ("open-fridge" or "stir-egg") while there is overlapping between ("put-baking" and "stir-egg"), so we conducted a T-Test to confirm this observation, as presented in **Table** 6.2. The results of T-Test confirm our observation that the feature vectors between "put-baking" and "stir-egg" are not significant and thus hard to distinguish.

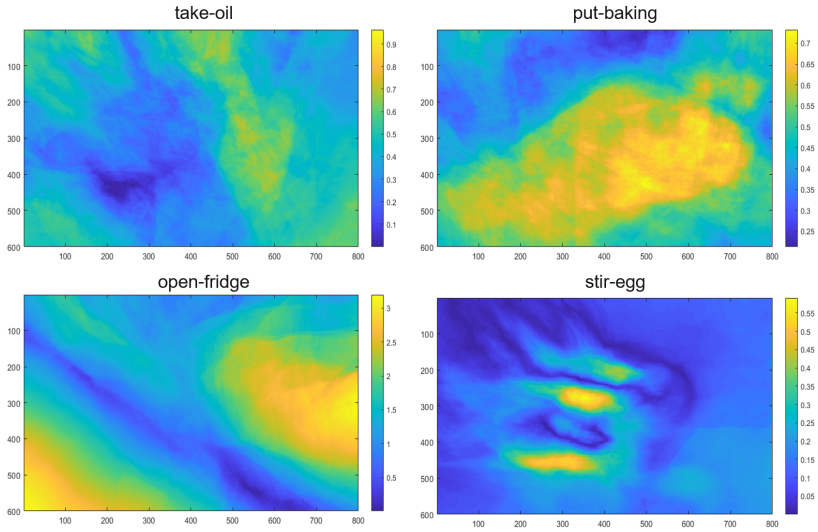


Figure 6.7: Visualization of averaged HOGs for the four activities used in this experiment.

Results for the averaged HOGs of the activities are illustrated in **Figure** 6.7. TThe visualizations of the averaged HOG for each activity show that it is easy to distinguish activities from each other. This information can be used as a complementary feature for IMU features to produce more robust feature vectors.

Distances between HOGs for different activities can be used as a quantitative

measurement to evaluate the similarities between activities. **Figure** 6.9 shows the distances in a log scale between all combinations of activities used in this research. Different metrics were used to calculate the distances between activities HOGs:

(i) **Chi Square** is a metric that can be used to compare histograms and can be defined as

$$d(x,y) = \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{(x_i + y_i)} \tag{6.4}$$

(ii) **L1** can be defined as

$$\|s\|_1 = \sum_{i=1}^{n}|y_i - y_i)| \tag{6.5}$$

(iii) **Earth Mover's Distance (EMD)** is a method to evaluate dissimilarity between two multi-dimensional distributions in some feature space where a distance measure between single features, which we call the ground distance, is given [100]. The EMD between histograms x and y is given by

$$emd(x,y) = \sum_{i=1}^{n}|cd_x(i) - cd_y(i)| \tag{6.6}$$

where,

$$cd_x(i) = \sum_{j=1}^{i} x_j \tag{6.7}$$

and,

$$cd_y(i) = \sum_{j=1}^{i} y_j \tag{6.8}$$

(iv) **Euclidean**

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6.9}$$

(v) **Squared Euclidean (SQ Euclidean)**

$$d(x,y)^2 = \sum_{i=1}^{n}(x_i - y_i)^2 \tag{6.10}$$

Real value distances for averaged HOGs for pairwise combination of activities using previously mentioned metrics are shown in **Table** 6.1. The actual differences provide more information about measurement differences inside the same metric, in which *Chi Square* metric provides the maximum difference among all pairwise activities as shown in **Figure** 6.8.

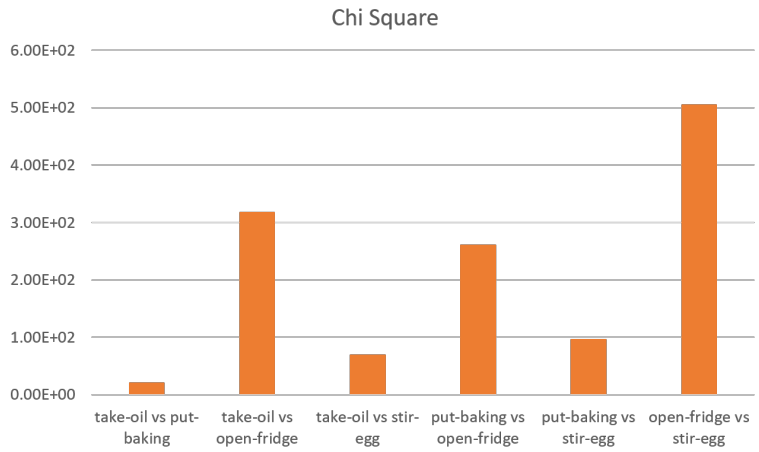


Figure 6.8: Visualization of averaged HOG for the four activities used in this experiment.

Table 6.1: Pairwise distance between averaged HOGs for different activities using Chi Square, L1, EMD, Euclidean and Squared Euclidean metrics

Activity1	**Activity2**	**Chi Square**	**L1**	**EMD**	**Euclidean**	**Squared Euclidean**
take-oil	put-baking	2.03E+01	1.29E+02	4.46E+04	5.52E+00	3.31E+01
take-oil	open-fridge	3.18E+02	9.80E+02	3.72E+05	3.88E+01	1.54E+03
take-oil	stir-egg	6.95E+01	2.24E+02	7.77E+04	8.88E+00	8.01E+01
put-baking	open-fridge	2.60E+02	8.96E+02	3.34E+05	3.64E+01	1.36E+03
put-baking	stir-egg	9.58E+01	2.96E+02	1.16E+05	1.09E+01	1.24E+02
open-fridge	stir-egg	5.05E+02	1.18E+03	4.50E+05	4.57E+01	2.12E+03

In this experiment, the preliminary investigations provide insights that combining optical flow and IMUs data can be complementary to each other and indeed a promising direction. The next section provides more details and experiments on

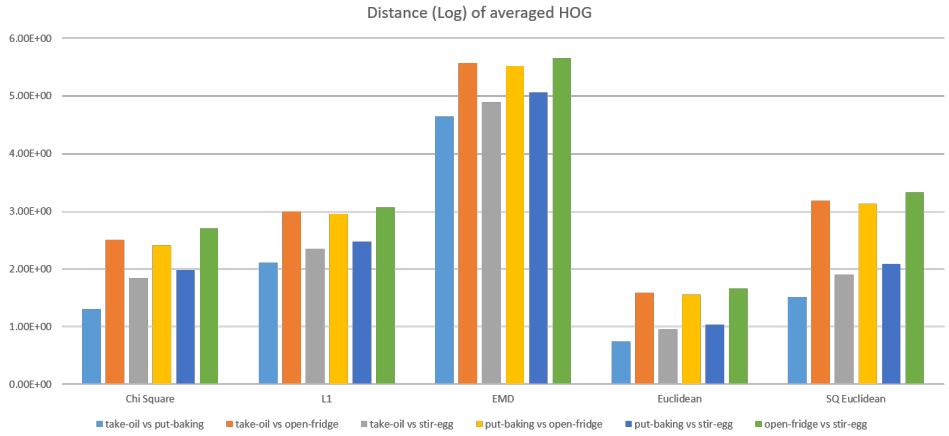


Figure 6.9: Different distance metrics (Chi Square, L1, Earth Mover's Distance (EMD), Euclidean and Squared Euclidean (SQ Euclidean)) in Log scale between different combination of activities feature vectors for HOG.

Table 6.2: P-value for T-Test using pairwise activities form IMU feature vectors for different sensors

P-Value										
Activity1	**Activity2**	**Acc-X**	**Acc-Y**	**Acc-Z**	**Roll**	**Pitch**	**Yaw**	**Mag-X**	**Mag-Y**	**Mag-Z**
take-oil	put-baking	0.0072	0.0077	0.0066	0.0181	0.0112	0.0112	0.0078	0.0069	0.0096
take-oil	open-fridge	0.0034	0.0036	0.0026	0.0013	0.001	0.0011	0.0044	0.002	0.0052
take-oil	stir-egg	0.002	0.0014	0.0013	0.0124	0.0024	0.0036	0.002	0.0009	0.0016
put-baking	open-fridge	0.1172*	0.1357*	0.099*	0.6825*	0.3141*	0.3219*	0.1321*	0.1207*	0.1947*
put-baking	stir-egg	0.2486	0.2425	0.2352	0.3618	0.2702	0.2907	0.2507	0.2234	0.2454
open-fridge	stir-egg	0.0001	0.0001	0.0001	0.0017	0.0002	0.0003	0.0001	0	0.0001
* $\times 1.0e^{-03}$										

combining IMUs and visual features.

6.5.2 IMUs and Visual Features

As mentioned in the previous chapter, researchers have used many evaluation datasets for activity recognition. In this experiment, we used the CMU Multi-Modal Activity database (CMU-MMAC) [38] since it is the only dataset that combines egocentric vision and IMUs sensors positioned on subjects left and right hands.

Based on semantic annotations of CMU-MMAC dataset proposed by [137],

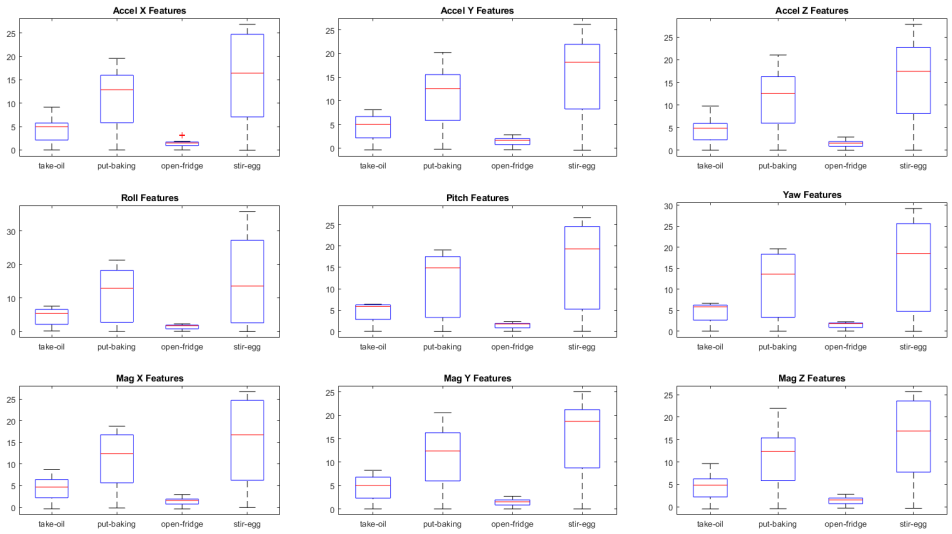


Figure 6.10: IMU features for all IMU sensors used in various activities for the experiment.

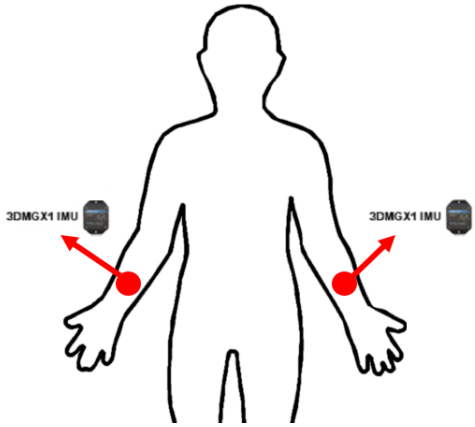


Figure 6.11: Location of IMUs sensors (3DMGX) on subjects' right and left hands.

tthree recipes were annotated: Brownie, Eggs and Sandwich. We used data extracted from the following modalities in our experiment in which a head-mounted high spatial resolution (800×600) camera at low temporal resolution (30 Hertz) was deployed.

The resolution of egocentric videos was reduced by factor of 0.5 to get a reso-

lution of (400×300). Second, two wired IMUs (3DMGX) on right and left hands, each of which had a triaxial *accelerometer*, *gyro*, and *magnetometer* sensor with a sampling rate at 125 Hz. The location of the sensors used in this experiment is plotted in **Figure** 6.11; the sample signal is shown in **Figure** 6.12.

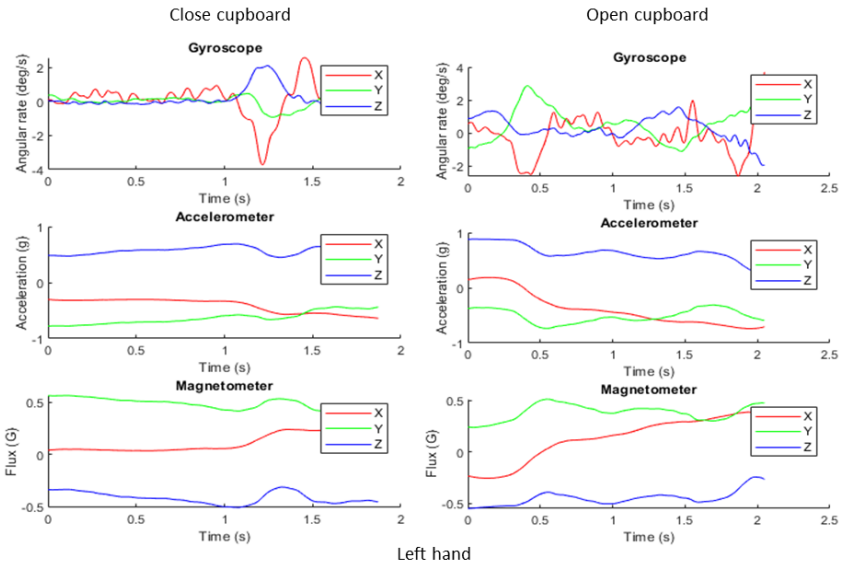
In this research, we considered 9 activities with 3 opposite pairs, including (close-bread-bag, open-bread-bag), (close-drawer,open-drawer), and (close-fridge,open-fridge), and fill-oil-oil-bottle-pan, shake-butter-spray-can and stir-bowl-fork.

The variability, both between subjects (intersubject variability) and within subjects (intrasubject variability), of the execution time for the same activity is illustrated in **Figure** 6.13. Occurrences on the X-axis represent the repetition of activity for all subjects performing the same activity, while the Y-axis represents the execution time of activity in seconds. For instance, activity "clean" can be done in less than 1 second, and it can be performed in more than 7 seconds. These scenarios increased the complexity of the problem.

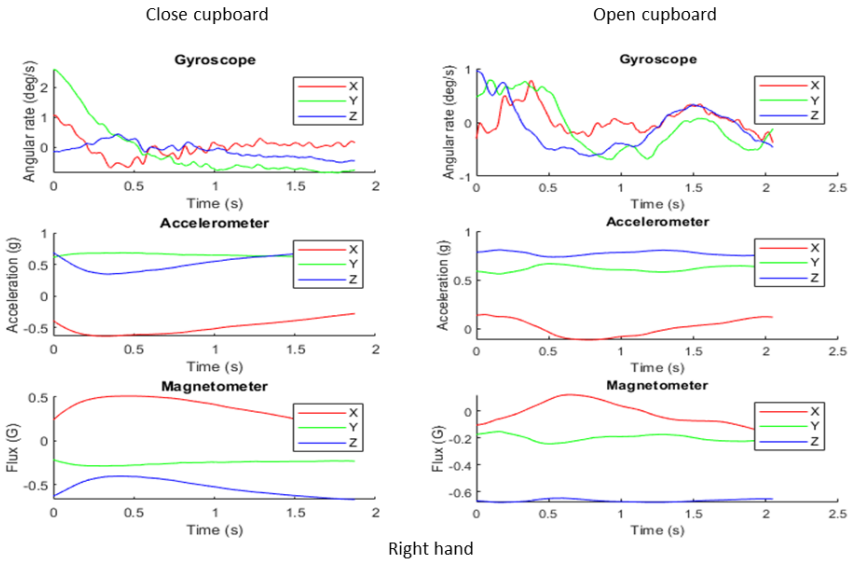
Here, we report the results from **Table** 6.3. The first part presents classification results for IMUs alone. The next section shows the results for activity recognition using visual descriptors only. The third part shows activity recognition results using fused IMUs features and visual descriptors. The last part is a general discussion about the results achieved.

Our recognition result achieved 49.89% for activity recognition using only IMUs data from left and right hands. This result is better than [79] by around 5%, and consideration should be given to the fact that we recognized 9 activities from only 2 sensors, while [79]'s method was used to recognize only 6 activities with 4 IMUs sensors. Furthermore, they used LSTM for feature extractions, whereas we used the statistical feature extraction method. In contrast, [108] achieved 62% f-measure recognition score by using only *accelerometer* data captured from IMUs sensors attached to non-human objects.

Visual activity recognition using VGG16 performed by [104] achieved 58.78% and 64.93% using CapsNet [101]. HOG and HOF were used by [119] for activity recognition, with the result that HOF produced better classification results than HOG. Our visual descriptor results are better than those of [119] and [108] using HOG and MBH descriptors, but [119] achieved better results than ours using HOF. The best classification result for recognizing all activities using visual descriptors



(a) Left hand IMU signal



(b) Right hand IMU signal

Figure 6.12: Sample signal of two opposite activities for two IMUs sensors: close cupboard and open cupboard for left (a) and right (b) hands for the same subject.

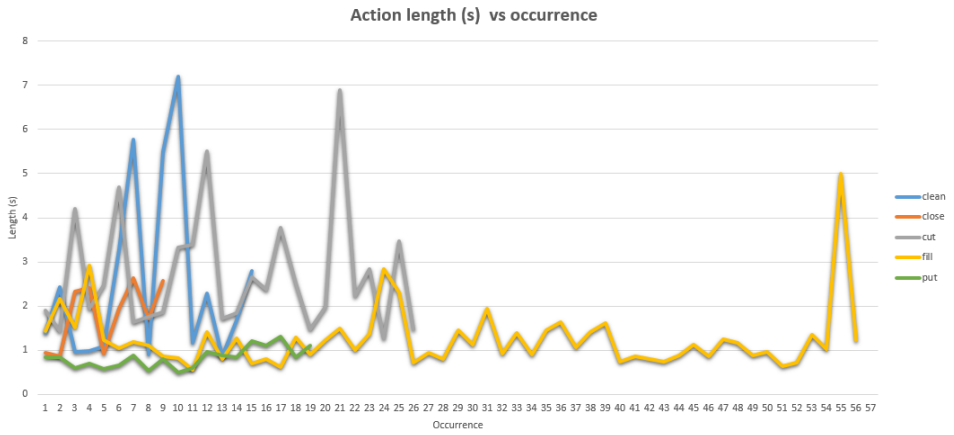


Figure 6.13: The variation of execution time for the same performed activity. Occurrences on the X-axis represent the repetition of activity for all subjects performing the same activity. While the Y-axis represents the execution time for activity in seconds.

was 86.88% using our HOG descriptor.

As the literature suggests, combing more than one sensor modality can increase the recognition accuracy. This has been proved in many previous research efforts. We worked on enhancing the recognition accuracy, and our approach was able to achieve this by a big margin compared to other methods. IMUs features are fused with the best visual descriptor (HOG) features to produce a feature vector of 648 lengths for each activity. Classifiers used in this experiment were trained to recognize 9 activities, with 3 opposite cases. The results achieved 99.61% accuracy. The confusion matrix is shown in **Figure** 6.14.

Identifying block size is a major step in local visual descriptors (HOG and HOF) calculation. As [34, 35] suggested that for a detection window of size 64×128, the recommended block size should be 16×16. [119] used a detection window of size 320×240 and block size of 8×8. And [108] used a detection window size of 640×480 with a block size of 32×32. In our settings, we used a detection window size of 300×400 with a block size of 50×50; thus our settings are best aligned with the block size suggestion by [34, 35] in regard to the detection window size, which explains the differences in activity recognition accuracy.

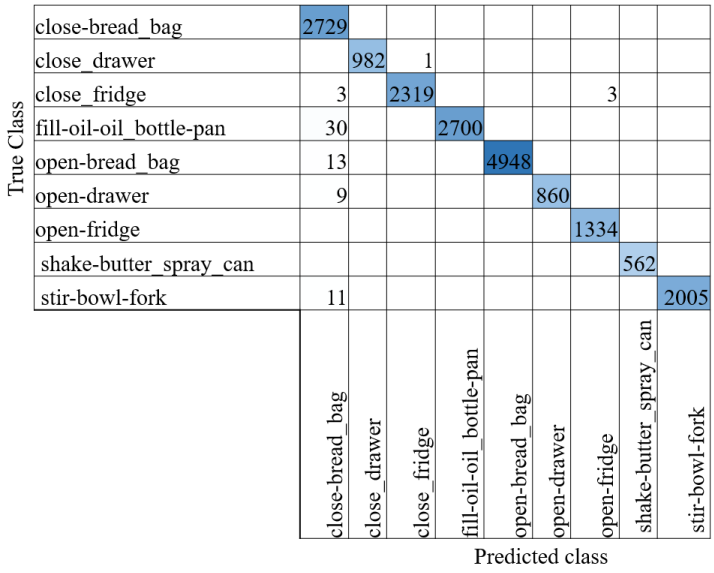


Figure 6.14: Confusion matrix for activity recognition results using IMUs and HOG fusion.

After conducting this experiment and successfully obtaining competitive results, the next step was to test a bigger dataset; however, for simulating real-life scenarios, minimizing the number of sensors should be considered [6, 42]. In the activity described in the following section, we used only and egocentric video sensor.

6.5.3 Only Visual Features

To be able to compare with state-of-the-art results, we considered the annotations of [137], which comprise a semantic annotation of the CMU-MMAC dataset on three recipes: Brownie, Eggs, and Sandwich. For human activities recognition, we used activity classes from Brownie, Eggs, and Sandwich recipes. For example, the name and distribution of classes for the Brownie recipe is shown in **Figure**6.3. For this experiment, we used only visual data with a fixed split of 80% training and 20% testing. In this experiment, we assumed that any 15 frames of the activity could be used to identify the designated human activity. We split each human

Method / Dataset	Modalities	Features	Classification Accuracy
[79] CMU-MMAC	IMUs	LSTM	45.06
	Visual	VGG16	58.78
	Visual	CapsNet	64.93
	IMUs + Visual	VGG16 + LSTM	65.60
	IMUs + Visual	CapsNet + LSTM	84.40
[119] UCF50	Visual	HOG	76.50
	Vision	HOF	79.50
[108] 50 Salads	Visual	HOG	49.00*
	Visual	HOF	47.00*
	Visual	MBH	53.00*
	IMUs**	Statistics	62.00*
	IMUs** + Visual	(HOG + HOF + MBH) + Statistics	71.00*
Our CMU-MMAC	IMUs	Statistics	49.89
	Visual	HOF	63.29
	Visual	HOG	86.88
	Visual	MBH	71.71
	IMUs + Visual	Statistics + HOG	**99.61**

* f-measure score.

** IMUs sensors were attached to objects not humans, and only *accelerometer* data was used.

Table 6.3: Comparison of recognition results using IMUs data, visual data, and fusion between IMUs and visual data between our method and three other methods.

activity video into 15 frame videos, which increased the training data on the one hand and provided the opportunity to examine our assumption that part of activity can recognize it on the other.

GoogLeNet [114] was used to extract features from videos. It is considered a convolutional neural network that contains 22 layers (counting only layers with parameters). The length of the feature vector produced for each frame was 1024.

In this experiment, the features produced were merged into recurrent neural networks (RNN). Previous information in RNNs could be used to predict current or future information, for example, using multiple and sequential video frames to predict, recognize, or classify the whole chunk of frames.

In this experiment, bidirectional LSTM (BiLSTM) was considered, which is an extension of RNN [102]. The layer learned bidirectionally where dependencies between time steps of sequence of images/data or time series exist. The signifi-

cance of these dependencies is used when the network is needed to learn from the complete time series at each time step, as shown in **Figure** 6.5. This is why we used BiLSTM in our experiment to classify activities that are considered to be a sequence of images in time. The error backpropagated through time and layers can be preserved in BiLSTM. Maintaining more constant error allows recurrent nets to continue to learn over many time steps.

The results for human activities recognition are summarized in **Table** 6.4, where we compare ours with those of [42], which is considered state of the art. The averages of precision, recall, and F1 score are reported in **Table** 6.4, where our method outperformed the state-of-the-art work of [42], with an F1 score of more than 4%.

Dataset	**Method**	**Precision**	**Recall**	**F1 Score**
Brownie	[42]	0.831	0.604	0.664
	Ours	0.701	0.717	**0.707**
Eggs	Ours	0.737	0.729	**0.730**
Sandwich	Ours	0.732	0.697	**0.702**

Table 6.4: Comparison between our method of classification and state of the art [42]. In this table averages of precision, recall, and F1 score are reported.

F1 score results of each human activity are shown separately in **Table** 6.5. The results achieved by [42] were better than ours: close class = 3%; other class = 1%; put class = 3%; take class = 3%; and Turn_on class = 1%. On the other hand, our method outperformed their results in the remaining 6 classes by an average of around 10%. For example, in the clean class, the difference is 6%, in fill it is around 10%, in open 3%, in shake 15%, in stir 5% and in walk with 22% difference.

A comparison of the results shows that ours were close to those of [42] in the close, other, put, take, and Turn_on classes, even though theirs were slightly better. However, our results outperformed theirs in the fill, shake and walk classes by a big margin although they used multi-modality (visual and IMUs) classification, while we used only visual data.

Table 6.6 and **Table** 6.7 show precision and recall results respectively com-

Class	Ours	[42]
Clean	0.634	0.571
Close	0.555	0.589
Fill	0.946	0.844
Open	0.731	0.707
Other	0.702	0.719
Put	0.567	0.595
Shake	0.571	0.426
Stir	0.980	0.939
Take	0.571	0.607
Turn_on	0.892	0.903
Walk	0.627	0.402

Table 6.5: F1 score comparison between our method of classification and state of the art [42].

Class	Ours	[42]
Clean	0.591	0.947
Close	0.616	0.764
Fill	0.944	0.748
Open	0.719	0.720
Other	0.700	0.866
Put	0.530	0.665
Shake	0.526	1.000
Stir	0.973	0.904
Take	0.633	0.620
Turn_on	0.839	0.976
Walk	0.642	0.935

Table 6.6: Precision results comparison between our method of classification and state of the art [42].

pared to [42] for each activity separately.

In this experiment, human activities recognition outperformed state-of-the-art work by [42] by an increment of more than 4% in the F1 score, even though we used only visual data features in this experiment. Moreover, in order to generalize

Class	Ours	[42]
Clean	0.684	0.409
Close	0.505	0.479
Fill	0.949	0.967
Open	0.744	0.696
Other	0.704	0.615
Put	0.610	0.537
Shake	0.625	0.270
Stir	0.987	0.977
Take	0.520	0.595
Turn_on	0.952	0.840
Walk	0.612	0.256

Table 6.7: Recall results comparison between our method of classification and state of the art [42].

our method, we further tested our method on another two CMU-MMAC datasets designated Scrambled eggs (Eggs) and Sandwich.

Activity recognition results for the Eggs dataset is shown in **Table** 6.8. In this table, precision, recall, and F1 score are reported. Our averaged F1 score results also outperformed state-of-the-art results [42] by more than 6%.

Even though the Sandwich dataset has three fewer activity classes, the averaged F1 score also outperformed the state-of-the-art results of [42] by 4%. Detailed results for precision, recall, and F1 scores are illustrated in **Table** 6.9.

6.6 Discussion

The results of the first experiment in **Section** 6.5.1 provide a clear overview that using more than one sensor for activity recognition is a promising direction to follow, even though this experiment considered only 4 activities and there was no actual sensor fusing or recognition.

Subsequently, another experiment in **Section** 6.5.2 considered IMUs data fused with visual descriptors data. Nine activities were chosen, six of which were opposite in nature, such as *close_drawer* vs. *open_drawer*, in addition to three other

Class	Precision	Recall	F1 score
Clean	0.667	0.545	0.600
Close	0.470	0.572	0.516
Fill	0.922	0.926	0.924
Open	0.654	0.671	0.662
Other	0.879	0.781	0.827
Put	0.627	0.485	0.547
Shake	0.776	0.843	0.808
Stir	0.959	0.980	0.970
Take	0.551	0.685	0.611
Turn_on	0.860	0.811	0.835
Walk	0.746	0.716	0.731
Average	0.737	0.729	0.730

Table 6.8: Precision, Recall and F1 score results for activity recognition using Eggs recipe.

Class	Precision	Recall	F1 score
Clean	0.857	1.0	0.923
Close	0.833	0.613	0.707
Fill	0.935	0.980	0.957
Open	0.869	0.628	0.729
Other	0.662	0.461	0.543
Put	0.470	0.362	0.409
Shake	0.511	0.805	0.625
Stir	0.722	0.722	0.722
Average	0.732	0.697	0.702

Table 6.9: Precision, Recall and F1 score results for activity recognition using Sandwich recipe.

activities with no opposite nature to somehow generalize the model. Estimating visual features appears to be a good aspect for consideration with statistical IMUs features. However, the calculations of visual features consume a lot of time and computational power [118]. Hence, finding other alternatives for activity recognition that demand less time and computational power is needed.

In the third experiment, in **Section** 6.5.3, only one sensor was deployed, a first-person camera, and the recognition process was accelerated. In this task, a deep learning approach was used for feature extraction and recognition, which benefitted from the speed and accuracy of deep learning. In this experiment, we used all the annotated activity classes based on [137]. The results outperformed the state-of-the-art work done by [42] for the Brownie recipe. Moreover, the model was trained and tested on the Eggs and Sandwich recipes and showed very good results. In addition, since the classified video is only 15 frames in length - corresponding to 0.5 seconds - this opens the door to near real-time activity recognition.

6.7 Conclusion

In this chapter, we have presented our action extraction tool for the CMU-MMAC dataset based on [137] annotations. Moreover, we have suggested categorizing human activities as general human and activities like (close, open) without specifying any object of interaction. This scenario increases the difficulty of recognition since it needs generalization. The second category is more specific: human activities like (close-drawer, open-drawer) were able to improve the recognition results; however, using the same object with opposite movements proved challenging. In the last category, we tried to simulate a more realistic scenario that can be used in real-life applications using only two IMUs sensors on the left and right hands with egocentric vision using opposite activities to make the experiment more realistic.

Our results for general human activities recognition outperformed the state-of-the- art work by [42] by more than a 4% increment of F1 score, even though we used only visual data features in this experiment. For more specific human activities, recognition results show that intermediate fusion between IMUs and egocentric data improves results by a big margin.

Part IV

Wrap-up

CHAPTER 7

Conclusions and Future Work

7.1 Conclusion

Nowadays, wearable technology advances are potentially transforming the quality of life, business, and the global economy. Wearable devices are electronics that consumers can wear to collect data from their bodies and the surrounding environment. Human activity recognition is one of the important research fields in wearable technology, as human activity recognition can be considered a computer vision and/or pervasive computing problem.

In this research we started from a computer vision problem based on optical flow, subsequently introduced open issues in respect to existing techniques, and then studied the relation between optical flow and human activity recognition taking the effectiveness of using multiple wearable sensors into consideration. Finally, we conducted experiments in activity recognition using different modalities. The following is a detailed summary to recap our research questions and answers.

I.1 How is it possible to benefit from existing pre-trained optical flow models without the existence of ground truth and with a limited training set?

In the literature, there are many pre-trained models for optical flow estimation. Optical flow estimation models can be trained via supervised or unsupervised training paradigms. Supervised training requires large amounts of training data with task specific motion statistics. Usually, synthetic datasets are used for this purpose. Fully unsupervised approaches are usually harder to train and show

weaker performance, although they have access to the true data statistics during training. In order to overcome this issue and benefit from pre-trained optical flow estimation models, we exploited a well-performing pre-trained model and fine-tuned it in an unsupervised way using classical optical flow training objectives to learn the dataset specific statistics. Thus, per-dataset training time can be reduced from days to less than one minute. Specifically, motion boundaries estimated by gradients in the optical flow field can be greatly improved using the proposed unsupervised fine-tuning.

I.2 How can optical flow performance metrics be evaluated with the existence of ground truth?

A significant amount of research has been conducted on optical flow estimation in previous decades. However, only a limited amount of research has been conducted on performance analysis of optical flow. These evaluations have shortcomings: the most common evaluation methodologies are end-point error (EPE) [91] and angular error (AE) [16], noting that the AE metric is based on prior work of Fleet and Jepson [46]. Even though EPE and AE metrics are popular, it is unclear which one is better. Moreover, AE penalizes errors in regions of zero motion more than motion in a smooth non-zero region, whereas EPE hardly discriminates between close motion vectors [47]. In addition, different cases exist (**Figure** 5.1) in which EPE gives same value between various scenarios. The only existing evaluation was done by Baker *et al.* [15], in which they compared the performance of EPE and AE and argued that EPE should become the preferred optical flow evaluation metric based on a qualitative assessment of an estimated optical flow for Urban sequence.

We have proposed a novel performance evaluation methodology based on using only optical flow ground truths and a modified version of ground truths in terms of shifting horizontally and vertically or magnifying by a certain value, or rotating, or a combination for evaluating performance metrics. The behavior and sensitivity of every metric have been reported for motion variations in horizontal, vertical, rotational, and magnification or a combination.

I.3 How can the best optical flow evaluation metric be determined? What are the theoretical justifications of using one metric and why?

A qualitative assessment [15] has been conducted on two common error metrics, EPE and AE, and suggested using EPE rather than using AE based on only one sample from the Baker dataset from Urban sequence. However, there is a need for a systematic evaluation of optical flow performance; hence we have conducted experiments on three popular datasets using ten different error metrics. A good metric is considered to be more sensitive to errors, for example, producing error values proportional to the change of motion between modified GT and GT. Existing metrics such as EPE, AE, and EM have sensitivity that differs slightly from one dataset to another. For instance, EPE and EM performed well on Baker, while AE and Em proved to be less sensitive on Kitti, and AE is not sensitive on Sintel. EPE's best sensitivity was on Kitti. On the other hand, AE's sensitivity was the worst among all three metrics.

As a rule of thumb, a good metric has to produce an error value proportional to the absolute values of change in optical flow with regard to ground truth. The general overview of mean error curves for existing and proposed error metrics gives a clear indication that some metrics outperform others.

Based on our observations, we concluded that all metrics are sensitive to horizontal, vertical, and (horizontal and vertical) variation. AE, GPRE, NEE, and ENEE1 metrics are sensitive to rotational variations. All metrics except AE and GPRE are sensitive to magnitude changing in motion. And only NEE and ENEE1 metrics are sensitive to all horizontal, vertical, rotational, and magnitude - or a combination.

II.1 How can optical flow influence the use of multi-sensor human activity recognition?

Optical flow can track features very accurately invariant to appearance of the representation at low velocities. For high velocity, tracking is less accurate

since the resolution must be reduced to obtain a larger tracking window with the same pixel size and, hence, a higher tracking velocity [99]. So, the advantage of combining optical flow and IMU data is the complementary characteristics of optical flow and inertial sensors since IMU data have large measurement uncertainty at slow motion and lower relative uncertainty at high velocities. Inertial sensors can measure very high velocities and accelerations. Our findings in IMUs and optical flow experiments provided obvious evidence that using a histogram of optical flow gradients can distinguish activities from each other.

II.2 What features can be extracted from multi-sensor human activity recognition? And what methods can be used for human activity recognition?

The feature extraction step is necessary to reduce the dimension of high dimensional raw data, toward the goal of producing more manageable groups of data for processing, and consequently, reducing computational power. The feature extraction method can select and /or combine different variables together to shape features, which will reduce data effectively, while describing the original dataset in an accurate and complete way. In this research, we have used three main groups of feature extraction methods; two of them are for visual data and one is for IMUs data.

First, we used local visual descriptors, such as a histogram of oriented gradients (HOG), a histogram of optical glow (HOF), and motion boundary histograms (MBH) [88, 35, 118, 119] as features for visual data. Even though the results were competitive and achieved high accuracy, it is combined with statistical features form IMUs, and local visual descriptors feature extraction needs huge computational power and a significant amount of time. Then, we used an existing deep learning model, GoogLeNet [114], to extract features from videos.

This method proved faster than local visual descriptors. We used two common methods for activity recognition: SVM for IMUs and visual descriptors and RNN, only for visual data extracted using GoogLeNet. Our recognition results outperformed the work of [42], which is considered state of the art, and we also tested our recognition method on another two CMU-MMAC recipes (Eggs and

Sandwich), which produced promising results.

7.2 Future Work

To conclude this thesis, we address several open issues, directions, and extensions for this research work and are dividing future work into the realms of computer vision and activity recognition.

7.2.1 Computer Vision

Recently, convolutional neural network (CNN)-based approaches have proven to be successful in the computer vision domain. They are being used in optical flow estimation in supervised as well as unsupervised training paradigms.

We have exploited a well-performing pre-trained model and fine-tuned it in an unsupervised way using classical optical flow in order to reduced training time and enhance motion boundaries. This work is opening the opportunity to investigate more on how to enhance estimated optical flow results to compete with state-of-the-art approaches. One component of future work is handling large displacement, which tends to be a common drawback in many optical flow estimators and reduces noise around edges.

In contrast, a significant amount of research was conducted on optical flow estimation in previous decades. However, only a limited number of research efforts have been devoted to performance analysis of optical flow. These evaluations have shortcomings and a theoretical justification for using one approach and the reasons for doing it are needed. In practice, design choices are often made based on unmotivated qualitative criteria or by trial and error. To the best of our knowledge, ours is the first experiment to provide theoretical justification for optical flow evaluation metrics.

Also, novel optical flow performance metrics have been proposed and evaluated alongside with current metrics. Our empirical findings suggest using two new optical flow performance metrics, namely Normalized Euclidean Error (NEE) and Enhanced Normalized Euclidean Error version one (ENEE1) for optical flow performance evaluation with ground truth. One of the open issues is enhancing the

rule of thumb for choosing the best metric. Moreover, testing the new metrics to evaluate some estimated optical flow is needed.

7.2.2 Human Activity Recognition

Different body sensors and modalities can be used in human activity recognition, either separately or simultaneously. Multi-modal data can be used in recognizing human activity.

Until now, we have proved that human activity recognition can even be improved by using only one sensor such as visual data. Moreover, we have investigated different feature extraction methods for both IMUs and visual data, such as statistical features for IMUs and local visual descriptors for visual data and feature extracted from deep learning. However, an open issue remains defining the best way for feature extraction, one that combines speed and best represents the activity with a minimum number of features. Exploring different ways of feature extraction for both IMUs and visual data is also needed. Minimizing the number of sensors placed on the human body for monitoring purposes is something to be considered for future smart homes and cities. In this way, human activity recognition can move from recognizing offline data to becoming a real-time recognition application.

Finally, more investigation is needed when considering opposite activities for the same object, something that has been abandoned by many research works. Also, a tradeoff among speed of recognition, accuracy, and the number of sensors used for human activity has to be considered.

Bibliography

[1] Histogram of oriented gradients (hog) descriptor | developer reference for intel® integrated performance primitives. https://software.intel.com/en-us/node/529070?language=de. (Accessed on 08/11/2020).

[2] Girmaw Abebe and Andrea Cavallaro. Inertial-vision: cross-domain knowledge transfer for wearable sensors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1392–1400, 2017.

[3] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi, and Vijayabaskar Kasi. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262, 2013.

[4] Aria Ahmadi and Ioannis Patras. Unsupervised convolutional neural networks for motion estimation. In *2016 IEEE international conference on image processing (ICIP)*, pages 1629–1633. IEEE, 2016.

[5] Samet Akpinar and Ferda Nur Alpaslan. Video action recognition using an optical flow based representation. In *IPCV*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014.

[6] Taha Alhersh, Samir Brahim Belhaouari, and Heiner Stuckenschmidt. Action recognition using local visual descriptors and inertial data. In *European Conference on Ambient Intelligence*, pages 123–138. Springer, 2019.

[7] Taha Alhersh and Heiner Stuckenschmidt. On the combination of imu and optical flow for action recognition. In *2019 IEEE International Conference*

on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2019.

[8] Taha Alhersh and Heiner Stuckenschmidt. Unsupervised fine-tuning of optical flow for better motion boundary estimation. 2019.

[9] Taha Alhersh and Heiner Stuckenschmidt. Unsupervised fine-tuning of optical flow for better motion boundary estimation. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications : February 25-27, 2019, in Prague, Czech Republic ; Volume 5: VISAPP*, pages 776–783, Setúbal, Portugal, 2019. SciTePress. Online-Ressource.

[10] Padmanabhan Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[11] Mehmet Ali Arabacı, Fatih Özkan, Elif Surer, Peter Jančovič, and Alptekin Temizel. Multi-modal egocentric activity recognition using audio-visual features. *arXiv preprint arXiv:1807.00612*, 2018.

[12] Sara Ashry, Reda Elbasiony, and Walid Gomaa. An lstm-based descriptor for human activities recognition using imu sensors. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, ICINCO*, volume 1, pages 494–501, 2018.

[13] Ferhat Attal, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. Physical human activity recognition using wearable sensors. *Sensors*, 15(12):31314–31338, 2015.

[14] Yong Bai and Qiang Bai. *Subsea engineering handbook*. Gulf Professional Publishing, 2018.

[15] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[16] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[17] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.

[18] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, 2020.

[19] Alejandro Betancourt, Pietro Morerio, Carlo S Regazzoni, and Matthias Rauterberg. The evolution of first person vision methods: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(5):744–760, 2015.

[20] Antonio Bevilacqua, Kyle MacDonald, Aamina Rangarej, Venessa Widjaya, Brian Caulfield, and Tahar Kechadi. Human activity recognition with convolutional neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 541–552. Springer, 2018.

[21] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402. IEEE, 2005.

[22] Aaron Bobick and James Davis. An appearance-based representation of action. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 307–312. IEEE, 1996.

[23] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263, 2008.

[24] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[25] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[26] Andrés Bruhn. Variational optic flow computation: accurate modelling and efficient numerics. *Department of Mathematics and Computer Science, Saarland University, Saarbrücken, Diss*, 2006.

[27] Andrés Bruhn, Joachim Weickert, Christian Feddern, Timo Kohlberger, and Christoph Schnorr. Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005.

[28] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *ECCV*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[29] João Carreira, Andrew Zisserman, and Quo Vadis. Action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2018.

[30] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3):4405–4425, 2017.

[31] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[32] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[33] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human motion analysis with deep metric learning. *arXiv preprint arXiv:1807.11176*, 2018.

[34] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[35] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006.

[36] Belur V Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.

[37] Juan Carlos Davila, Ana-Maria Cretu, and Marek Zaremba. Wearable sensor data classification for human activity recognition based on an iterative learning framework. *Sensors*, 17(6):1287, 2017.

[38] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute*, page 135, 2008.

[39] George Demiris, Brian K Hensel, Marjorie Skubic, and Marilyn Rantz. Senior residents' perceived need of and preferences for" smart home" sensor technologies. *International journal of technology assessment in health care*, 24(1):120, 2008.

[40] Florenc Demrozi, Graziano Pravadelli, Azra Bihorac, and Parisa Rashidi. Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey. *arXiv preprint arXiv:2004.08821*, 2020.

[41] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[42] Alexander Diete and Heiner Stuckenschmidt. Fusing object information and inertial data for activity recognition. *Sensors*, 19(19):4119, 2019.

[43] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the ICCV*, pages 2758–2766, 2015.

[44] Wilfried Elmenreich. An introduction to sensor fusion. *Vienna University of Technology, Austria*, 502, 2002.

[45] et Co. Kiel Anschuetz. *The Anschütz Gyro Compass*. 1910.

[46] David J Fleet and Allan D Jepson. Computation of component image velocity from local phase information. *International journal of computer vision*, 5(1):77–104, 1990.

[47] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21, 2015.

[48] Vigneswara Rao Gannapathy, AFBT Ibrahim, Zahriladha Bin Zakaria, Abdul Rani Bin Othman, and Anas Abdul Latiff. Zigbee-based smart fall detection and notification system with wearable sensor (e-safe). *Int J Res Eng Technol*, 2(8):337–344, 2013.

[49] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[50] Jack A Goldstone. The new population bomb: the four megatrends that will change the world. *Foreign Aff.*, 89:31, 2010.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative ad-

versarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[53] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[54] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[55] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[56] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[57] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018.

[58] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2019.

[59] Junhwa Hur and Stefan Roth. Optical flow estimation in the deep learning age. *arXiv preprint arXiv:2004.02853*, 2020.

[60] Earnest Paul Ijjina and Krishna Mohan Chalavadi. Human action recognition in rgb-d videos using motion sequence information and deep learning. *Pattern Recognition*, 72:504–516, 2017.

[61] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, volume 2, 2017.

[62] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for optical flow, disparity, or scene flow estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 614–630, 2018.

[63] Nahed Jalloul, Fabienne Porée, Geoffrey Viardot, Phillipe L'Hostis, and Guy Carrault. Activity recognition using complex network analysis. *IEEE journal of biomedical and health informatics*, 22(4):989–1000, 2018.

[64] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–706, 2018.

[65] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[66] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. Human activity recognition: A survey. *Procedia Computer Science*, 155:698–703, 2019.

[67] Ibrahim Kajo, Aamir Saeed Malik, and Nidal Kamel. An evaluation of optical flow algorithms for crowd analytics in surveillance system. In *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS)*, pages 1–6. IEEE, 2016.

[68] AD King. Inertial navigation-forty years of evolution. *GEC review*, 13(3):140–149, 1998.

[69] Jan Koenderink, Whitman Richards, and Andrea J van Doorn. Space-time disarray and visual awareness. *i-Perception*, 3(3):159–165, 2012.

[70] Christian Krupitzer, Timo Sztyler, Janick Edinger, Martin Breitbach, Heiner Stuckenschmidt, and Christian Becker. Beyond position-awareness—extending a self-adaptive fall detection system. *Pervasive and Mobile Computing*, 58:101026, 2019.

[71] S Santhosh Kumar and Mala John. Human activity recognition using optical flow based feature set. In *Security Technology (ICCST), 2016 IEEE International Carnahan Conference on*, pages 1–5. IEEE, 2016.

[72] Ammar Ladjailia, Imed Bouchrika, Hayet Farida Merouani, Nouzha Harrati, and Zohra Mahfouf. Human activity recognition via optical flow: decomposing activities into basic actions. *Neural Computing and Applications*, pages 1–14, 2019.

[73] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *Advances in neural information processing systems*, pages 354–364, 2017.

[74] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[75] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[76] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.

[77] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019.

[78] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016.

[79] Yantao Lu and Senem Velipasalar. Human activity classification incorporating egocentric video and inertial measurement unit data. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 429–433. IEEE, 2018.

[80] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[81] Oisin Mac Aodha, Ahmad Humayun, Marc Pollefeys, and Gabriel J Brostow. Learning a confidence measure for optical flow. *IEEE transactions on pattern analysis and machine intelligence*, 35(5):1107–1120, 2012.

[82] Osama Makansi, Eddy Ilg, and Thomas Brox. Fusionnet and augmentedflownet: Selective proxy ground truth for training on unlabeled images. *arXiv preprint arXiv:1808.06389*, 2018.

[83] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[84] Brendan McCane, Kevin Novins, D Crannitch, and Ben Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, 2001.

[85] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[86] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[87] Melvin M Morrison. Inertial measurement unit, December 8 1987. US Patent 4,711,125.

[88] Nuno Miguel Banheiro Moutinho. Video and image match searching, March 2 2017. US Patent App. 15/252,142.

[89] Fernando Moya Rueda, René Grzeszick, Gernot Fink, Sascha Feldhorst, and Michael ten Hompel. Convolutional neural networks for human activity recognition using body-worn sensors. In *Informatics*, volume 5, page 26. Multidisciplinary Digital Publishing Institute, 2018.

[90] Thi-Hoa-Cuc Nguyen, Jean-Christophe Nebel, Francisco Florez-Revuelta, et al. Recognition of activities of daily living with egocentric vision: A review. *Sensors*, 16(1):72, 2016.

[91] Michael Otte and H-H Nagel. Optical flow estimation: advances and comparisons. In *European conference on computer vision*, pages 49–60. Springer, 1994.

[92] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[93] Vittorio Passaro, Antonello Cuccovillo, Lorenzo Vaiani, Martino De Carlo, and Carlo Edoardo Campanella. Gyroscope technology and applications: A review in the industrial perspective. *Sensors*, 17(10):2284, 2017.

[94] Janez Perš, Vildana Sulić, Matej Kristan, Matej Perše, Klemen Polanec, and Stanislav Kovačič. Histograms of optical flow for efficient representation of body motion. *Pattern Recognition Letters*, 31(11):1369–1376, 2010.

[95] Suneth Ranasinghe, Fadi Al Machot, and Heinrich C Mayr. A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, 12(8):1550147716665520, 2016.

[96] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2. IEEE, 2017.

[97] Parisa Rashidi and Diane J Cook. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on systems, man, and cybernetics-part A: systems and humans*, 39(5):949–959, 2009.

[98] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, volume 3, page 7, 2017.

[99] Hugo Romero, Sergio Salazar, Rogelio Lozano, and Ryad Benosman. Fusion of optical flow and inertial sensors for four-rotor rotorcraft stabilization. *IFAC Proceedings Volumes*, 40(15):209–214, 2007.

[100] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[101] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[102] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[103] Laura Sevilla-Lara, Yiyi Liao, Fatma Guney, Varun Jampani, Andreas Geiger, and Michael J Black. On the integration of optical flow and action recognition. *arXiv preprint arXiv:1712.08416*, 2017.

[104] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[105] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2620–2628, 2016.

[106] J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

[107] Fridtjof Stein. Efficient computation of optical flow using the census transform. In *Joint Pattern Recognition Symposium*, pages 79–86. Springer, 2004.

[108] Sebastian Stein and Stephen J McKenna. Recognising complex activities with histograms of relative tracklets. *Computer Vision and Image Understanding*, 154:82–93, 2017.

[109] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Hierarchical feature aggregation networks for video action recognition. *arXiv preprint arXiv:1905.12462*, 2019.

[110] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019.

[111] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[112] Jian Sun, Yongling Fu, Shengguang Li, Jie He, Cheng Xu, and Lin Tan. Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors. *Journal of Sensors*, 2018, 2018.

[113] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, 2018.

[114] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[115] Timo Sztyler. *Sensor-based human activity recognition: Overcoming issues in a real world setting*. PhD thesis, 2019.

[116] Tianyu Tang. *Optical flow estimation based on 3-D gradient constraint*. PhD thesis, 2010.

[117] Zhigang Tu, Wei Xie, Dejun Zhang, Ronald Poppe, Remco C Veltkamp, Baoxin Li, and Junsong Yuan. A survey of variational and cnn-based optical flow techniques. *Signal Processing: Image Communication*, 72:9–24, 2019.

[118] Jasper Uijlings, Ionut Cosmin Duta, Enver Sangineto, and Nicu Sebe. Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. *International Journal of Multimedia Information Retrieval*, 4(1):33–44, 2015.

[119] Jasper RR Uijlings, Ionut C Duta, Negar Rostamzadeh, and Nicu Sebe. Realtime video classification using dense hof/hog. In *Proceedings of international conference on multimedia retrieval*, page 145. ACM, 2014.

[120] Alessandro Verri and Tomaso Poggio. Motion field and optical flow: Qualitative properties. *IEEE Transactions on pattern analysis and machine intelligence*, 11(5):490–498, 1989.

[121] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176. IEEE, 2011.

[122] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[123] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Baidu-uts submission to the epic-kitchens action recognition challenge 2019. *arXiv preprint arXiv:1906.09383*, 2019.

[124] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018.

[125] Anne S Wannenwetsch, Margret Keuper, and Stefan Roth. Probflow: Joint optical flow and uncertainty estimation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1182–1191. IEEE, 2017.

[126] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Learning to detect motion boundaries. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[127] Josh Wills and Serge Belongie. A feature-based approach for determining dense long range correspondences. In *European conference on computer vision*, pages 170–182. Springer, 2004.

[128] Marco Wrzalik and Dirk Krechel. Human action recognition using optical flow and convolutional neural networks. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, pages 801–805. IEEE, 2017.

[129] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In A. Fusiello et al. (Eds.), editor, *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Part II, LNCS 7584, pages 168–177. Springer-Verlag, October 2012.

[130] x-io Technologies Limited. x-io technologies limited. uk company.

[131] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 933–942, 2016.

[132] Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. Innohar: A deep neural network for complex human activity recognition. *IEEE Access*, 7:9893–9902, 2019.

[133] Yanchao Yang and Stefano Soatto. Conditional prior networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 271–287, 2018.

[134] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

[135] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: a comprehensive survey. *IEEE Transactions on Image Processing*, 25(6):2752–2773, 2016.

[136] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019.

[137] Kristina Yordanova and Frank Krüger. Creating and exploring semantic annotation for behaviour analysis. *Sensors*, 18(9):2778, 2018.

[138] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994.

[139] Abrar Zahin, Rose Qingyang Hu, et al. Sensor-based human activity recognition for smart healthcare: A semi-supervised machine learning. In *International Conference on Artificial Intelligence for Communications and Networks*, pages 450–472. Springer, 2019.

[140] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017, 2017.

[141] Chun Zhu and Weihua Sheng. Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3):569–573, 2011.

[142] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Guided optical flow learning. *arXiv preprint arXiv:1702.02295*, 2017.