TURUN YLIOPISTO
UNIVERSITY OF TURKU

RUNTIME RESOURCE MANAGEMENT FOR VISION-BASED APPLICATIONS IN MOBILE ROBOTS

Sherif A.S. Mohamed

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 19 | TECHNICA – INFORMATICA | TURKU 2023

TURUN YLIOPISTO
UNIVERSITY OF TURKU

RUNTIME RESOURCE MANAGEMENT FOR VISION-BASED APPLICATIONS IN MOBILE ROBOTS

Sherif A.S. Mohamed

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 19 | TECHNICA – INFORMATICA | TURKU 2023

University of Turku

Faculty of Technology
Department of Computing
Information and Communication Technology
Doctoral programme

Supervised by

Professor
Juha Plosila
University of Turku, Finland

Adjunct Professor
Mohammadhashem Haghbayan
University of Turku, Finland

Professor
Antonio Miele
Politecnico di Milano, Italy

Reviewed by

Professor
Guoying Zhao
University of Oulu, Finland

Associate Professor
Samira Ebrahimi Kahou
École de technologie supérieure
University of Quebec, Canada

Opponent

Associate Professor
Amir H. Payberah
KTH Royal Institute of Technology, Sweden

ISBN 978-951-29-9145-7 (PRINT)
ISBN 978-951-29-9146-4 (PDF)
ISSN 2736-9390 (PRINT)
ISSN 2736-9684 (ONLINE)
Painosalama, Turku, Finland, 2023

UNIVERSITY OF TURKU
Faculty of Technology
Department of Computing
Information and Communication Technology
MOHAMED, SHERIF: Runtime Resource Management for Vision-based Applications in Mobile Robots
Doctoral dissertation, 156 pp.
Doctoral programme
February 2023

ABSTRACT

Computer-vision (CV) applications are an important part of mobile robot automation, analyzing the perceived raw data from vision sensors and providing a rich amount of information on the surrounding environment. The design of a high-speed and energy-efficient CV application for a resource-constrained mobile robot, while maintaining a certain targeted level of accuracy in computation, is a challenging task. This is because such applications demand a lot of resources, e.g. computing capacity and battery energy, to run seamlessly in real time. Moreover, there is always a trade-off between accuracy, performance and energy consumption, as these factors dynamically affect each other at runtime. In this thesis, we investigate novel runtime resource management approaches to improve performance and energy efficiency of vision-based applications in mobile robots. Due to the dynamic correlation between different management objectives, such as energy consumption and execution time, both environmental and computational observations need to be dynamically updated, and the actuators are manipulated at runtime based on these observations. Algorithmic and computational parameters of a CV application (output accuracy and CPU voltage/frequency) are adjusted by measuring the key factors associated with the intensity of computations and strain on CPUs (environmental complexity and instantaneous power). Furthermore, we show how mechanical characteristics of the robot, i.e. the speed of movement in this thesis, can affect the computational behaviour. Based on this investigation, we add the speed of a robot, as an actuator, to our resource management algorithm besides the considered computational knobs (output accuracy and CPU voltage/frequency). To evaluate the proposed approach, we perform several experiments on an unmanned ground vehicle equipped with an embedded computer board and use RGB and event cameras as the vision sensors for CV applications. The obtained results show that the presented management strategy improves the performance and accuracy of vision-based applications while significantly reducing the energy consumption compared with the state-of-the-art solutions. Moreover, we demonstrate that considering simultaneously both computational and mechanical aspects in management of CV applications running on mobile robots significantly reduces the energy consumption compared with similar methods that consider these two aspects separately, oblivious to each other's outcome. **KEYWORDS:** computer vision, mobile robot, resource management, energy efficiency.

TURUN YLIOPISTO
Teknillinen tiedekunta
Tietotekniikan laitos
Informaatio- ja viestintäteknologia
MOHAMED, SHERIF: Runtime Resource Management for Vision-based Applications in Mobile Robots
Väitöskirja , 156 s.
Tohtoriohjelma
Kuukausi 2023

TIIVISTELMÄ

Tietokonenäön (CV) sovellukset ovat tärkeä osa liikkuvien robottien automatisointia, sillä ne analysoivat näköantureiden havaitsemaa raakatietoa ja tarjoavat runsaasti tietoa lähiympäristöstä. Nopean ja energiatehokkaan CV-sovelluksen suun-nittelu resursseiltaan rajalliselle liikkuvalle robotille siten, että samalla säilytetään tietty tavoiteltu laskentatarkkuus, on haastava tehtävä. Tämä johtuu siitä, että tällaiset sovellukset vaativat paljon resursseja, esimerkiksi laskentakapasiteettia ja akkuenergiaa, jotta niitä voidaan käyttää saumattomasti reaaliajassa. Lisäksi tarkkuuden, suorituskyvyn ja energiankulutuksen välillä on aina kompromissi, koska nämä tekijät vaikuttavat dynaamisesti toisiinsa suoritusaikana. Tässä tutkielmassa tutkitaan uusia resurssienhallinnan lähestymistapoja, joilla voidaan parantaa mobiilirobottien näköpohjaisten sovellusten suorituskykyä ja energiatehokkuutta. Koska eri hallintatavoitteiden, kuten energiankulutuksen ja suoritusajan, välillä on dynaaminen korrelaatio, sekä ym-päristö- että laskennalliset havainnot on päivitettävä dynaamisesti, ja toimilaitteita manipuloidaan ajonaikana näiden havaintojen perusteella. CV-sovelluksen algoritmi- ja laskentaparametreja (tulostustarkkuus ja suorittimen jännite/taajuus) säädetään mittaamalla keskeisiä tekijöitä, jotka liittyvät laskentojen intensiteettiin ja suorittimen rasitukseen (ympäristön monimutkaisuus ja hetkellinen teho). Lisäksi osoitamme, miten robotin mekaaniset ominaisuudet, eli tässä tutkielmassa liikenopeus, voivat vaikuttaa laskennalliseen käyttäytymiseen. Tämän tutkimuksen perusteella lisäämme robotin nopeussäädön resurssienhallinta-algoritmiimme tarkasteltujen laskennallisten parametrien (tulostustarkkuus ja suorittimen jännite/taajuus) lisäksi. Arvioidaksemme ehdotettua lähestymistapaa suoritamme useita kokeita miehittämättömällä maa-ajoneuvolla, joka on varustettu sulautetulla tietokoneyksiköllä ja jossa käytetään RGB- ja tapahtumakameroita CV-sovellusten näköantureina. Saadut tulokset osoittavat, että esitetty hallintastrategia parantaa näköpohjaisten sovellusten suorituskykyä ja tarkkuutta ja vähentää samalla merkittävästi energiankulutusta verrattuna uusimpiin ratkaisuihin. Lisäksi osoitamme, että sekä laskennallisten että mekaanisten näkökohtien samanaikainen huomioon ottaminen liikkuvissa roboteissa toimivien CV-sovellusten hallinnassa vähentää merkittävästi energiankulutusta verrattuna vastaaviin menetelmiin, joissa nämä kaksi näkökohtaa otetaan huomioon erikseen ilman tietoa toistensa vaikutuksesta. **ASIASANAT:** tietokonenäkö, mobiilirobotti, resurssien hallinta, energiatehokkuus.

Acknowledgements

The support and guidance I have received throughout this dissertation process has been invaluable.

Firstly, My sincere gratitude goes to my supervisors from the University of Turku, Prof. Juha Plosila and Dr. Mohammad-Hashem Haghbayan, for their expertise and ideas that helped develop the research methodology and question. I was able to improve my thinking through their insightful comments and push my work to new heights. It has been an honor to work with my supervisor Professor Antonio Miele from the University Politecnico di Milano. I appreciate your patience with me and the many opportunities you have given me to reshape my research. Dessutom, jag skulle vilja tacka min kompis Jawad Yasin för alla hjälp.

My parents Prof. Abdelmonem and Amna have been a great help, and I am grateful for their love, wise advice, and sympathetic ears. They have always been a great support to me. My siblings Dr. Mohamed, Khaled, Hend, and Omar also deserve my deepest gratitude for their support. I am grateful to my second family, my father-in-law Mamdouh and mother-in-law Mervat, for their prayers. Lastly, I couldn't have completed my dissertation without my wife Minerva and my daughter Emma. Together, they were a source of love and joy during these very intensive academic years. Thank you for your unconditional support and stimulating discussion, Minerva. You were amazing.

"A Lannister always pays his debts"

01.06.2022

Sherif A.S. Mohamed

SHERIF A.S. MOHAMED

Sherif earned his bachelor's degree in computer science from Ain Shams university, Egypt. In 2016, he earned his master's degree from Kunsan National University, South Korea. Currently, he is pursuing his doctoral degree at University of Turku, Finland. He is working as a senior AI imaging Engineer at Huawei Technologies Oy, Finland. He is interested in the following research areas: vision-based algorithms, event-based cameras, embedded systems, resource-awareness, power efficiency, and machine learning.

Table of Contents

Abbreviations

CV	Computer Vision
GPS	Global Localization System
IO	Inertial Odometry
INS	Inertial Navigation Systems
DR	Dead Reckoning
MEMS	Mirco Electro Mechanical System
EKF	Extended Kalman Filter
RO	Radar Odometry
CW	Continuous Wave
FMCW	Frequency Modulated Continuous Wave
SAR	Synthetic Aperture Radar
ICP	Iterative Closest Point
RANSAC	RANdom SAmple Consensus
LO	Laser Odometry
LiDAR	Lighting Detection and Ranging
MAV	Micro Aerial vehicle
OGM	occupancy Grid Map
SVD	Singular Value Decomposition
GICP	Generalized Iterative Closest Point
VO	Visual Odometry
DVS	Dynamic Vision Sesnor

FoV	Field of View
OF	Optical Flow
CMOS	Complementary Metal–Oxide–Semiconductor
CCD	Charge-Coupled Device
SGM	Semi-Global Matching
AER	Address Event Representation
BA	Background Activity
SNR	Signal-to-Noise Ratio
TS	TimeStamp
KNN	K-Nearest Neighbors
LoG	Laplacian of Gaussian
DVFS	Dynamic Voltage and Frequency Scaling
IRA	Internal Resource-Aware
EIRA	External and Internal Resource-Aware
SW	Software
HW	Hardware
f^b	Linear force
w^b	Angular velocity
C_b^n	Attitude
f^n	Reference Vector
P^n	Position
V^n	Velocity
I_i	Grayscale images
S_i	Radar scans
C_i^n	Radar features

W_e	Emitted wavelength
W_r	Received wavelength
PC_i	Radar pointcloud
L_i	Radar landmark
DOF	Dense Optical Flow
RGB	red green blue
RGB-D	Depth RGB
K	Intrinsic calibration matrix
R	Rotational matrix
t	Translation vector
$\mathbf{P}^{ID}$	Inverse-depth points
$\mathbf{P}^{E}$	Euclidean points
$\mathbf{w}$	Image coordinates
p_c	Camera position
$\mathbb{R}^3$	World coordinates
$\mathbb{R}^2$	Image coordinates
x_i	x-axis
y_i	y-axis
z_i	z-axis
θ_i	Elevation
ψ_i	Azimuth
(d_i)	Depth
3D	3-dimensions
2D	2-dimensions
ρ_i	Inverse depth

σ_ρ^2	Inverse-depth variance
α	Parallax angle
$H(x)$	Entropy
p	occurrence probability
n	number of pixels in an image
LC-Harris	Low-complexity Harris
R_h	Harris score
I_x	Horizontal gradient
I_y	Vertical gradient
DC	Direct current
RA	Resource awareness

List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

I **Sherif A. S. Mohamed**, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen and J. Plosila, “A Survey on Odometry for Autonomous Navigation Systems,” in IEEE Access, vol. 7, pp. 97466-97486, 2019.

II **Sherif A.S. Mohamed**, M. Haghbayan, J. Heikkonen, H. Tenhunen and J. Plosila, “Monocular Visual Odometry Based on Hybrid Parameterization”, In Proc. of the 12th International Conference on Machine Vision (ICMV), SPIE, 2019.

III **Sherif A.S. Mohamed**, M. Haghbayan, M. Rabah, J. Heikkonen, H. Tenhunen and J. Plosila, “Towards Dynamic Monocular Visual Odometry Based on an Event Camera and IMU Sensor”, In Proc. of the 3rd EAI International Conference on Intelligent Transport Systems (INTSYS), Springer, pp 249-263, 2019.

IV **Sherif A.S. Mohamed**, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and J. Plosila, ”Towards real-time edge detection for event cameras based on lifetime and dynamic slicing”, In Proc. of the International Conference on Artificial Intelligence and Computer Vision (AICV), Springer, pp 584-593, 2020.

V **Sherif A.S. Mohamed**, J. N. Yasin, Mohammad-Hashem Haghbayan, Antonio Miele, Jukka Heikkonen, Hannu Tenhunen, and J. Plosila,”Dynamic Resource-aware Corner Detection for Bio-inspired Vision Sensors”, In Proc. of The 25th international conference on pattern recognition (ICPR), 2020.

VI **Sherif A.S. Mohamed**, J. N. Yasin, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and J. Plosila, ”DBA-Filter: A Dynamic Background Activity Noise Filtering Algorithm for Event cameras”, In Proc. of the Computing Conference, 2021.

VII **Sherif A.S. Mohamed**, J. N. Yasin, Mohammad-Hashem Haghbayan, Antonio Miele, Jukka Heikkonen, Hannu Tenhunen, and J. Plosila, ”Asynchronous Corner Tracking Algorithm based on Lifetime of Events for DAVIS

Cameras", In Proc. of the 15th International Symposium on Visual Computing (ISVC), Springer, pp 530-541, 2020.

VIII **Sherif A.S. Mohamed**, Mohammad-Hashem Haghbayan, Antonio Miele, Onur Mutlu and J. Plosila, "Energy-Efficient Mobile Robot Control via run-time Monitoring of Environmental Complexity and Computing Workload", In Proc. of the International Conference on Intelligent Robots and Systems (IROS) 2021.

The list of original publications have been reproduced with the permission of the copyright holders.

1 Introduction

Computer vision, usually abbreviated as CV, enables robots and other machines to see, recognize, and analyze things in their environment the same way humans do. The concept of computer vision is based on how to teach computers to analyze and process data collected from the surrounding environment such as images to perform low-level applications including corner/edge detection [1], optical flow [2] and depth estimation [3], as well as high-level applications such as self-localization [4], object detection [5], and 3D reconstruction [6]. To do that the first phase of CV is to observe the environment and collect raw data. Different types of cameras and setups can be used, depending on the application and the domain. There are a number of cameras that are commercially available such as RGB [7], fisheye [8], omnidirectional [9], RGB-D [10] and event-based cameras [11]. Every camera has its advantages and disadvantages under various conditions such as low lighting, indoor/outdoor, textureless environment and highly dynamic scenes.

In addition to becoming increasingly popular, CV applications are gaining significant attention from a variety of fields due to the unique attributes of cameras, such as their low price, small size, and ability to capture rich information [12]. As a result, compact systems with limited resources, i.e., mobile robots, may be equipped with a set of cameras to perform CV methods. The use of this combination has a significant impact on various domains. For example, in manufacturing, CV can provide different solutions such as productivity analytics, visual inspection, and quality management [13]. In healthcare, a number of studies have demonstrated promising results in complex medical diagnostics including cancer detection [14], COVID-19 diagnosis [15], and mask detection [16]. In addition, computer vision approaches have the potential to have a huge impact to reshape and increase the productivity of the agriculture domain. Computer vision has potential to improve the overall operation of the agricultural sector, from reducing production costs with intelligent automation to boosting productivity [17]. Other domains can also benefit from the rapid evolution of CV algorithms such as sport [18] and transportation. [19].

The main challenges of vision-based applications are accuracy, performance, and efficiency [4]. These aspects are correlated to each other and there is always a trade-off among them when optimizing one aspect over the other aspects. For example in feature extraction techniques, e.g., edge detection, obtaining high accuracy is a quite important point that highly affects the outcome of the applications that are using such techniques, e.g., visual odometry or object detection. Therefore, it is vital to process all valuable information, i.e., inliers, and discard outliers in order to increase the accuracy. From the other perspective, considering and processing all the captured raw data, here we refer to them as the information, consumes lots of energy and time that in mobile robots is an important factor. These mobile robots are considered resource-constrained systems, i.e., they have limited resources such as computing units and battery [11]. Resource management in mobile robots demands processing the captured data in run-time. This is because most of the features needed for optimizing/balancing different mentioned aspects, e.g., performance and accuracy, are changing dynamically over the system's activity time. Based on this, to have an efficient CV-based application, it is necessary to develop a resource-aware management system that offers the manageability of resources of mobile robots at runtime.

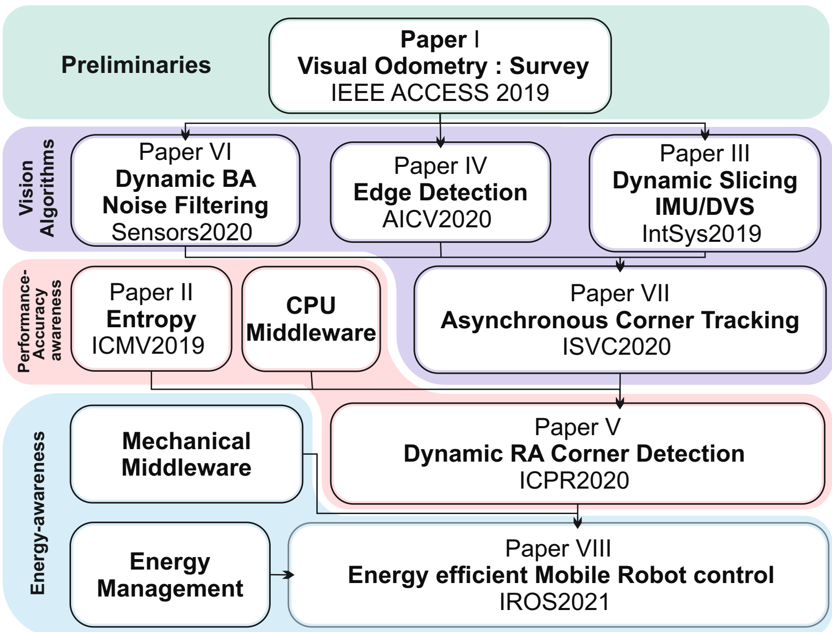


Figure 1. An overview of selected papers included in the thesis. Each category is labeled with different color.

In this thesis, the main objective is to design and implement a resource-aware system that monitors and manages the resource of a mobile robot in run-time. In this process, three key factors of computer vision, i.e., accuracy, performance, and efficiency, are considered to be optimized individually or together. The proposed management strategies continuously monitor the system's behaviour in run-time and manipulate the actuators to achieve high efficiency. Here the monitoring can be the robot's environment and/or the internal computing units while the actuation is toward the environment, e.g., mechanical units, and/or the computing units, e.g., CPU's voltage and frequency.

Figure 1 illustrates the different section of this thesis and the related publications.

Preliminaries: In the preliminaries part, a brief introduction about different aspects of the vision-based applications and resource management is discussed. Moreover, in this section, the literature review providing an up-to-date understanding of the previous research is included. For this one paper, i.e., Paper I, is published based on this thesis that provides a comprehensive categorization of GPS-denied self-localization techniques including the most recent developed vision methods.

Vision algorithms: In order to improve the vision-based application on mobile robots, one step in this thesis is focused only to improve the performance and accuracy at the pure algorithmic level. Our algorithms mainly use two different types of vision cameras including RGB and event cameras. As event-based cameras is a new vision sensor with different output, i.e., event stream, than conventional RGB cameras, i.e., intensity images. Before applying multi-objective resource management systems on application based on event camera, we develop novel methods for analyzing their output. The proposed algorithms seek to find sharp edges and unique corners in the generated event stream. Results of this phase of research are two papers i.e., Paper VII and Paper IV. Since event cameras generate an asynchronous stream of events, we develop an algorithm to represent the event stream to easily understand the amount of information in the scene. This approach is explained in more detail in Paper III. Event cameras also suffer from hardware limitations, for example shot noise in photons. Thus, denoising algorithm is needed to improve both the accuracy and computational complexity. We proposed a denoising algorithm that filters out background activity noise and trailing noise. Details of the proposed algorithm and experimental results are reported in Paper VI.

Performance/accuracy awareness: After development of efficient CV application. We implement a resource-aware system that is capable of monitoring the computational unit of a mobile robot. Based on the feedback received from the computational units in run/time, the resource management system performs adaptive filtering on the incoming data from the vision sensors as well as dynamically manipulating the frequency of the processing units to keep the performance, here frame per second, and accuracy of the application running on the robot. The proposed approach is evaluated for different situations the robot might encounter, such as low and high textured environments. Paper V is the original publication by the author based on this.

Energy-awareness: After developing the performance resource management, we extended the previous system by considering the energy consumption of the mobile robot as well as optimizing the application execution. To do that, the management algorithm factors in the power consumption of both computational parts, i.e., processing units, and mechanical parts, e.g., motors, and feeds this information back during operation. The actuator knobs for the proposed resource management are the dynamic voltage/frequency of the processing units and the speed of the robot. This approach is demonstrated in more detail in Paper VIII.

Testing and evaluation of the proposed techniques take place on a mobile robot powered by a Jetson TX2 board and a DAVIS camera. Event cameras are chosen because of their attractive features including low power consumption, high dynamic range (HDR), and ultra-high temporal resolution, making them suitable for embedded systems such as mobile robots. The output of these cameras depends on both the complexity and dynamicity of the environment. Using the experimental setup, we demonstrate the necessity of using the resource-aware system to enable mobile robots to perform complex vision applications with high performance and efficiency. Results of this study demonstrate that the proposed resource awareness RA system is able to maintain the high performance and accuracy of vision apps running on an embedded system. In addition, the total energy consumption of the robot system and in general, the energy consumed per unit distance, a metric for the locomotion cost, has decreased significantly. Experiments are conducted on various environmental complexities including low-, medium- and high-complexity. Results show that the proposed RA approach can save on average 50.5 %, 41 %, and 30% of energy, respectively.

Accordingly, the remaining chapters of the thesis are arranged in the following manner. Chapter 2 covers the fundamental concepts of self-localization using cameras, the potential and challenges of applying CV applications on mobile robots and discuss what resource management can offer to enable efficient deploy of CV application on mobile robots. The development of novel computer-vision algorithms for event-based cameras is presented in Chapter 3. Chapter 4 is devoted to highlighting the proposed performance/accuracy awareness system. In Chapter 5, we present a resource-aware system that achieves energy awareness by co-managing the computing units and the mechanical parts of the mobile robot. Chapter 6 concludes the thesis with some discussion.

2 Preliminaries

2.1 Self-Localization

Self-localization problem is one important aspect in nowadays autonomous systems design [4]. For self-localization, the autonomous system, e.g., a vehicle or robot, might use antennas to receive radio signals from satellites or observes the surrounding environment using onboard sensors to determine the location. While using GPS for localization, the map is known and the radio signals transmitted by satellites are used to locate the position of the vehicle. This technique is suitable for outdoor localization due to the limitations of radio signals in indoor environments [20]. In contrast, GPS-denied self-localization methods utilize data collected by different types of onboard sensors, e.g., cameras to obtain the position, i.e., the change in the translation vector and orientation matrix over time. Such techniques provide a more accurate and reliable location and they can operate in scenarios where the map is unknown.

The overall categorization of self-localization techniques is illustrated in Figure 2. This includes two main branches GPS and GPS-denied localization. In GPS-denied localization, vehicles/robots are equipped with different types of sensors including wheels, IMUs, ranging sensors such as radars and Light Detection and Ranging (Lidar) and vision sensors, e.g., cameras, which are used to observe the surrounding environment and capture data, which is processed to obtain the relative pose, i.e., position and orientation. As part of our research, we investigate different proposed vision approaches in the literature. There are several reasons for this, the attractive features of cameras and the importance of self-location for autonomous mobile robots. Cameras offer several benefits over other types of sensors, including their small size, energy efficiency, and the ability to capture rich and detailed images of their surroundings. As a result of these advantages, cameras are an ideal solution for platforms with limited resources, such as mobile robots. Robots equipped with cameras can perform various tasks, such as odometry and obstacle avoidance. Several visual-odometry techniques will be discussed, along with their advantages and limitations.

Visual odometry (VO) approaches process images to obtain the relative pose of the camera over time. Early use of the VO was in NASA's Mars exploration mission. The robot used a set of cameras to obtain the position on the rough terrains of Mars. Figure 3 illustrates the four main parts of VO methods based on each of these factors:

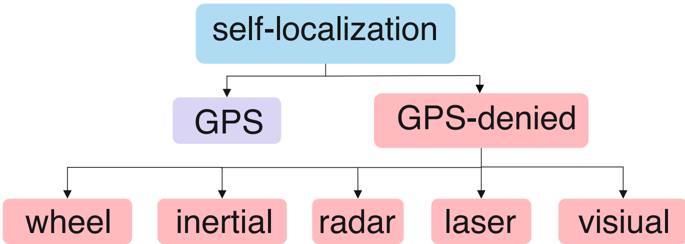


Figure 2. Self-localization strategies in the literature

camera pose, type of camera, and the number of cameras.

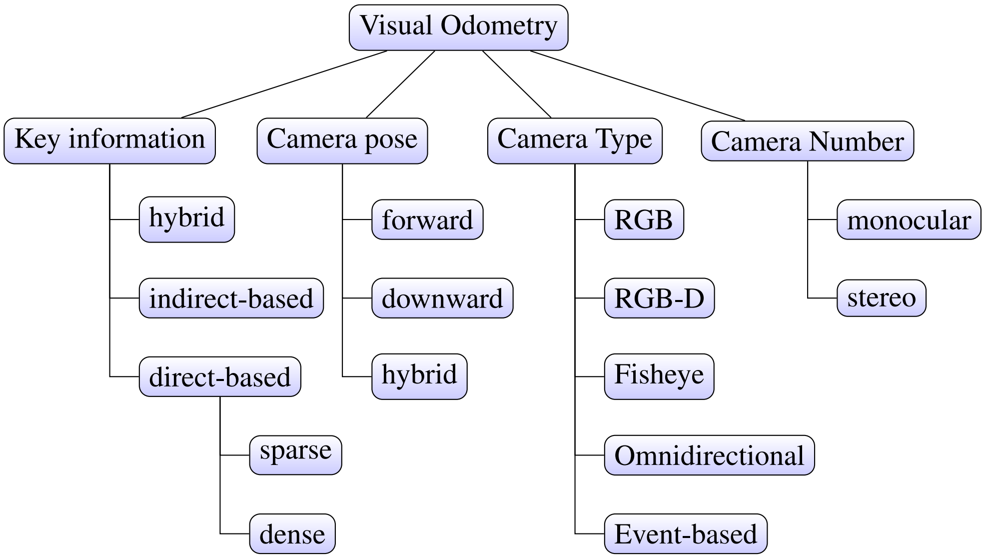


Figure 3. Overview classification of the existing visual odometry methods based on four criteria

key information: Existing approaches can be grouped based on the key information used to estimate the pose into three main groups, such as 1) feature-based approaches 2) direct-based approaches, and 3) hybrid approaches. Through feature-based approaches, or indirect approaches, interesting features such as corners or edges are extracted from images, and then these features are tracked in successive images to calculate the transformation matrix. An edge is characterized by significant changes in intensity in one dimension. To extract strong edges, existing edge detectors are either based on gradients or Laplacian [21] kernels. Corners are one of the most unique and easy-to-track features in an image because they change intensity in two dimensions. There have been several corner detectors proposed in the literature over the past 40 years. These include Harris [1], SIFT [22], FAST [23], and SURF [24]. There are pros and cons to each of these detectors; a detailed assessment of these detectors can be found in [25]. Direct-based methods analyze raw images directly to determine the camera's pose. In order to calculate the transformation matrix, two consecutive images are aligned to minimize intensity changes, i.e., residuals. Based on a global smoothness assumption, dense optical flow (OF) optimizes each pixel in images using several optimization techniques [26]. Sparse approaches solve the brightness constancy for only some pixels using a template matching algorithm [2]. By combining both indirect- and direct-based methods in a single framework, hybrid-based approaches address the limitations of each method. Feature-based approaches tend to perform poorly in low-textured environments due to the limited number of edges and corners in the scene. Direct-based methods, on the other hand, can deal with low-texture environments since they use pixels instead of features to compute the transformation matrix. Direct methods, however, require significant computing power to process all pixels in an image, while feature-based methods offer a more efficient solution by only processing certain features. In hybrid-based VO, a number of features are extracted from images, then a batch is extracted around each feature and all the pixels in the batch are used to minimize intensity changes when computing the pose. Many approaches are available to compute the pose of the camera, including [9] [27] [28] [29].

Camera pose. In the existing VO methods there are three camera setups available: forward-facing [30] [31] [32], downward-facing [33], and unidirectional [34] setup. Due to their wide field of view (FoV), forward-facing camera setups can effectively cover a large area, making them suitable for ground and aerial vehicles. However, this setup has some limitations in detecting micro-movements and shadows. A downward-facing setup is primarily used for obtaining the position and orientation in a previously explored environment. While this setup is capable of dealing with slow movement, its ability to match features between two consecutive images when the camera is moving rapidly is not so good. It has been reported that some approaches used the forwarding-facing and downward-facing setups to determine the camera's pose. High-speed localization is achieved with the forward-facing camera, while low-speed pose determination is achieved with the downward-facing camera.

Camera type. In terms of the type of camera used to observe the environment, existing VO approaches can be classified into five groups: RGB, RGB-D, fisheye, omnidirectional, and event-based. To generate color images, CMOS or CCD sensors are used to record the absolute intensity of the pixels with a fixed frame rate (e.g., 60 FPS). RGB images are colorful images in which each pixel has three channels including red, green, and blue. The majority of VO algorithms use gray-scale images with a single channel to reduce computation complexity. CCD sensors employ a global shutter technique, which means the entire frame is captured at the same time, while CMOS sensors utilize a rolling shutter technique, which means the sensor captures the brightness of each pixel sequentially from top to bottom. In comparison with a stereo RGB setup, RGB-D cameras are the best choice when it comes to providing information about the depth of an image. The stereo RGB camera setup provides the depth of an image by performing a search along epipolar lines, which is a computationally-intensive process. Similar to RGB VO methods, RGB-D approaches can be classified as indirect methods, e.g., [10] [35] and direct-based methods such as in [36]. Typically, fisheye cameras have a field of view between 100 and 180 degrees. In comparison to pinhole cameras, fisheye cameras offer a better view of the environment. Although it introduces some distortion, it can be corrected by using a special distortion model. A couple of VO algorithms that take advantage of fisheye cameras can be found in [8] [37]. Alternatively referred to as 360-degree cameras, omnidirectional cameras have an azimuth field of view (FoV) of 360 degrees and an elevation of 90 to 140 degrees. Therefore, it can estimate poses more accurately than traditional cameras with a limited field of view, because it gathers more information from the surrounding environment. The system also eliminates the inherent rotation-translation ambiguity caused by a small field of view cameras. I have listed a few approaches [9] [38] [39] that utilize omnidirectional cameras. The term ”event camera” refers to cameras that use bio-inspired sensors, such as dynamic vision sensors (DVS). It records pixel-by-pixel brightness changes and triggers asynchronous events. Some of the features that event cameras offer are including HDR, low latency and high temporal resolution. In addition, they consume less power than traditional cameras because they use energy to process only brightness changes. As a result, they offer a substantial improvement for high frequent localization on resource-constrained platforms.

Number of cameras. Visual odometry algorithms can be categorized into monocular and stereo algorithms based on cameras used to observe the scene. Monocular setups obtain the pose in terms of a 1D translation vector and 3D rotational matrix of a vehicle by processing images taken by one camera. It is not affected by the baseline issue, which is why it has gotten so much attention in recent years. Since the transformation between the first two frames are unknown, monocular-based approaches can only compute translation vectors up to a relative scale. Here are a few examples of approaches utilizing a single camera: [28] [7] [40]. Multiple cameras are used in a stereo camera setup such as [41] [42] [43] to allow easy reconstruction of 3D information from stereo image pairs. To obtain an accurate pose, key information can be extracted and tracked between two stereo pairs, and then a motion estimation algorithm is applied, e.g., Maximum-Likelihood. This is a major drawback of using a stereo camera setup to obtain the localization is that the calibration of the camera affects the pose estimation. Many factors can affect the calibration of the stereo cameras such as shocks, vibrations, etc., therefore extrinsic calibrations gradually degrade over time, and periodic re-calibration is needed to keep an accurate estimate of position. Moreover, the baseline of stereo cameras, i.e., the distance between the stereo cameras are fixed. In several scenarios, this impacts depth estimation accuracy.

2.2 Mobile Robotics

Mobile robotics is a solutions-oriented branch of industry that is rapidly developing, merging engineering sciences and information technology with such disciplines as computer vision, Artificial Intelligence and many others. It is this interdisciplinary interplay that has made it possible to master the inherent complexity of mobile robots. Mobile robots have the ability to move independently and perform specific tasks. Furthermore, they can function autonomously without human assistance, in addition to their mobility. Sensors, actuators, and control electronics form the basis for mobile robots. The purpose of sensor technology is to be able to detect and collect a wide range of information and data that can be used for odometry and movement planning. As a result of their low price, small size, and ability to capture detailed information from the environment, cameras are one of the most common sensors used in mobile robots. Utilizing images captured via the camera, mobile robots can perform different vision-based applications such as visual odometry and object detection. Implementing efficient vision-based applications on mobile robots is a challenging task since processing the data captured by vision sensors requires a lot of computational resources. Since mobile robots have limited resources such as computing power and battery capacity, operating high efficiency and performance vision-based applications in real-time is challenging. Therefore, it is crucial to develop a resource-aware system that monitors and coordinates the use of resources and optimizes the vision-based application in runtime to achieve a high efficacy and performance CV applications running on mobile robots under challenging conditions.

2.3 Resource-aware systems

Resource-aware frameworks have the ability to monitor their resource usage and dynamically manage resources according to specified constraints. Resources are CPU units and batteries. They enable complex applications to operate on resource-constrained platforms, e.g., mobile robots. Since mobile robots have limited resources of energy, i.e., battery capacity, they are equipped with microcontrollers that have limited computing capabilities. It is imperative to consider both energy and computational constraints while designing the resource-aware system. The computational constraints can be managed using a high-level approximation or filtering in the application layer and via a middleware that monitors the status of the CPU units and manipulates the frequency of the CPU accordingly. We present a solution to manage the computational constraints of a mobile robot running a complex CV application using two correlated phases 1) a three-layer filtering and 2) a middleware that monitors and dynamically changes CPU's frequencies via DVFS, more details can be found in Chapter 4. In the field of mobile robotics, energy constraints are a major problem [44; 45].

A main focus of the research has been optimizing the kinematic energy of a robot neglecting the fact that aside from kinematic energy consumption, there are many other sources of energy drain. Cyber-physical devices such as mobile robots – that is, they contain both physical parts, such as motors, and cyber parts, such as microcontrollers – it is imperative to consider both components when calculating the overall energy consumption [46]. As an example, applications involving heavy visual processing drain most of the available energy. Thus, the cyber parts of robots contribute heavily to total energy consumption and therefore it should be taken into account while optimizing their energy consumption. As a result of the motivations described above, we present a resource-aware system that can simultaneously manipulate mechanical and computational actuators to achieve the best overall energy consumption. For more information, refer to Chapter 5.

2.4 Mobile Setup

In this section, we will give details about the robot architecture, the embedded board, and vision applications. Below is a more detailed explanation of these components.

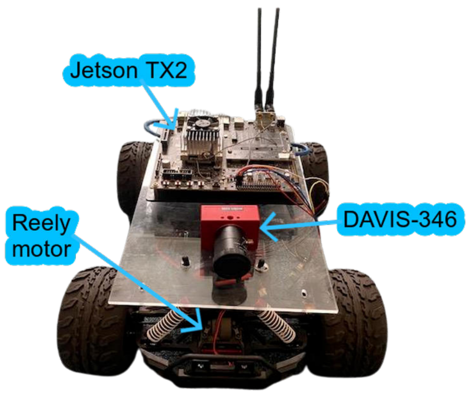


Figure 4. Platform setup.

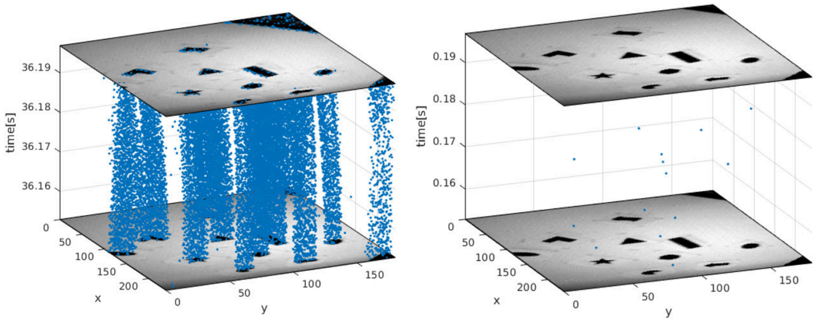
Overall robot architecture. A brushless DC motor, an event-based camera, and an embedded system board are included in the mobile robot, as seen in Figure 4. Jetson TX2 is an extremely powerful and energy-efficient embedded board with two separate CPU clusters running at maximum frequencies of 2 GHz each. We refer to these two clusters as little and big clusters. The *LITTLE.cluster* is a quad-core ARM Cortex-A57 which is designed to be energy efficient and multithreading. The *big.cluster* with a dual-core Nvidia Denver 2, is specifically designed to achieve high single-thread performance. An event camera (e.g., DAVIS-346) is mounted on the robot to generate both intensity images and asynchronous event streams [47; 48]. The robot is powered by a DC brushless motor. This motor operates at a high speed of 50000 RPM and has a power rating of 3000 kV.

The embedded board. The middleware monitors and manages the resource of the robot, i.e., CPU units. Firstly, a mapping unit assigns each task to a different core within the CPU cluster based on the desired performance. The CPU frequency is managed with runtime DVFS. The CPU cluster voltage/frequency can range from 300 kHz to 2 GHz at various intermediate steps. The middleware can identify if the application is losing data due to overload by reporting application throughput. Throughput is expressed as a number between 0 and 1, with 1 representing the full processing of all the data.

Application. Our proposed system is tested and evaluated using three applications with varying runtime complexity, namely e-frame construction and a corner detector with and without filtering. The e-frame construction application has a low runtime complexity [49] construct gray-scale images from generated event streams using a highpass filter and pixel-by-pixel comparisons. The corner detector [50] detects features, i.e., corners from the event stream. This is a complex application because it performs expensive operations to determine whether each incoming event is a corner by computing its eigenvalues. We use a three-level filtering algorithm with the corner detection application in order to reduce the computational complexity to a medium level [11].

3 Vision Algorithms

Computer vision methods in the literature are typically based on images captured by conventional cameras with CMOS vision sensors. An example of this is obtaining a robot's pose by detecting and tracking the corners of consecutive intensity images. There are several major limitations to this type of sensor that may affect the performance of CV applications. Greyscale or RGB images are typically generated by capturing the intensity at a fixed rate, for example, 60Hz. A stationary camera produces redundant information when the scene is static, which causes the computational load to increase without improving accuracy. The motion blur phenomenon occurs, however, when objects are highly dynamic in the scene, which adversely affects the vision application. Blind time, which is the time gap between two consecutive images, can also cause significant information loss. Traditional cameras are not able to cope with highly dynamic scenes, resulting in inaccurate CV applications such as object tracking, face detection and pose estimation.



(a) The camera moves quickly (b) The camera is static

Figure 5. Events generated between two consecutive images in two different scenarios.

Event cameras use different types of vision sensors e.g., ATIS [51] to capture intensity changes per pixel and publish events that represent these changes. Every time the intensity of a pixel increases/decreases by a certain amount, an event is triggered. The amount of triggered events per second depends on both the environmental complexity as well as its dynamicity. Figure 5 shows two examples of the output of the event camera. A few events are triggered by event cameras in static or textureless scenes. Millions of events are generated when the camera moves rapidly and/or the scene is highly dynamic. By transmitting only brightness changes and thereby not transmitting redundant data, event cameras consume less power than traditional cameras. Their total power consumption at the system level is less than 100 mW [52]. Event cameras have many unique characteristics that make them a perfect sensor for applications that operate in highly dynamic environments. For instance, obtaining the pose for a drone operating in complex environments, in which pose shall be updated frequently (e.g., 200 Hz) to enable the drone to navigate safely.

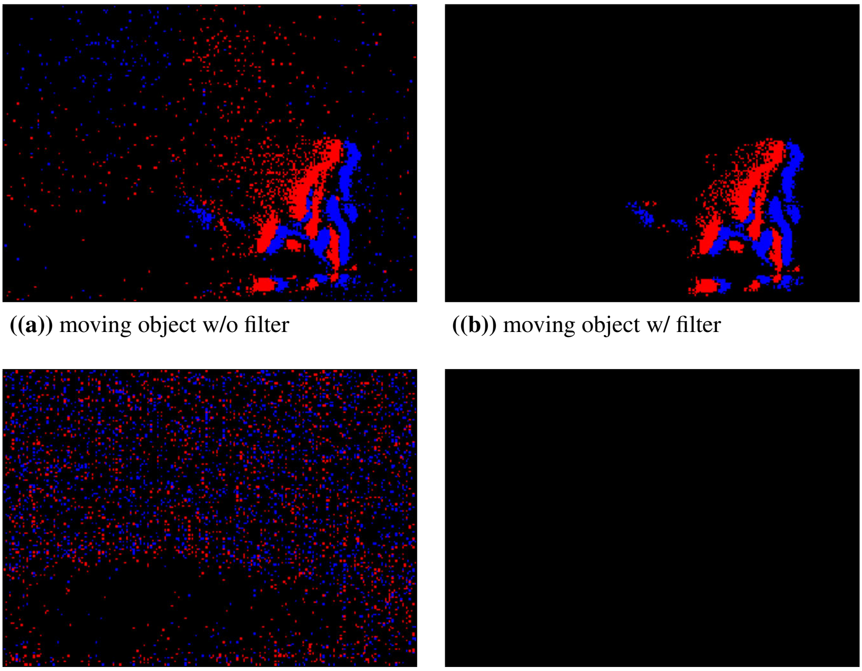
It is required to develop novel methods to process the event stream, in order to exploit events' camera potential. The vision-based algorithms that have been refined in this thesis are techniques to improve the 1) denoising 2) image construction and 3) feature extraction.

3.1 Denoising

Like all vision sensors, event cameras are noisy due to hardware limitations in photons and transistors [47]. A study of the quantization of temporal contrast in event cameras is particularly relevant because the process has not been fully described. A novel denoising method is needed to model the noise of an event stream in order to extract meaningful information from it. There are two major types of noise exits in the event camera output including background activity (BA) and hot pixel noise. There are many factors that cause the background activity, including charge injection, transistor leakage, and thermal noise. BA events location is more random than real events and occurs less frequently. A similar phenomenon exists in DVS, similar to the hot pixel in traditional image sensors. Due to its inability to reset properly, the pixel continuously emits events.

In this thesis, we present a dynamic denoising algorithm capable of achieving high signal-to-noise ratios (SNR) and running in a real-time manner on embedded systems. The proposed denoising consists of two stages 1) a timestamp(TS) filter eliminates noise using a historical map of timestamps of previous events as well as the computed optical flow. With this filter, we can remove noise that can result from sudden movements or significant contrast changes. 2) a background activity (BA) filter employs a spatial-temporal kernel using the KNN algorithm to eliminate hardware noise, such as transistor switch leakage, that causes BA noise. Results demonstrate that our approach achieves high SNR while reducing the number of events processed without sacrificing information. According to our experiments, the proposed algorithm can achieve a high SNR of 13.64 dB.

In Figure 6, we compare the generated out of a DVS camera without any filtering against our approach. A ground-mounted event camera recorded the dataset, which consisted of a person running in front of the camera. According to the results, our filter is effective in removing background activity noise and trailing noise caused by significant contrast changes, as well as hardware limitations.



((a)) moving object w/o filter ((b)) moving object w/ filter

((c)) no motion w/o filter ((d)) no motion w/ filter

Figure 6. Night run sequence. Screenshots taken at various points in time of the DVS video output.

Abbr.	Definition
$\mathbf{T_{w/o}}$	No. of True events w/o applying filtering
$\mathbf{N_{w/o}}$	No. of Noise w/o applying filtering
$\mathbf{T_f}$	No. of True events w/ filtering
$\mathbf{N_f}$	No. of Noise w/ filtering
$\mathbf{P_{TR}}$	Percentage of true events left after filtering
$\mathbf{P_{NR}}$	Percentage of noise left after filtering
$\mathbf{TNR_{w/o}}$	Ratio of true to noise w/o filtering
$\mathbf{TNR_f}$	Ratio of true to noise w/ filtering

Table 1. Quantitative metric

To quantify noise filtering performance, we present the following terms and metrics in Table 1. 1.

	shapes_6dof	night_run
$\mathbf{T_{w/o}}$	10.31×10^6	0.34×10^6
$\mathbf{N_{w/o}}$	7.62×10^6	0.455×10^6
$\mathbf{T_f}$	9.73×10^6	0.3×10^6
$\mathbf{N_f}$	0.42×10^6	0.064×10^6
$\mathbf{P_{TR}}$	94.37%	88.2%
$\mathbf{P_{NR}}$	5.24%	14.06%
$\mathbf{TNR_{w/o}}$	1.31 dB	-1.27%
$\mathbf{TNR_f}$	13.64 dB	6.709 dB

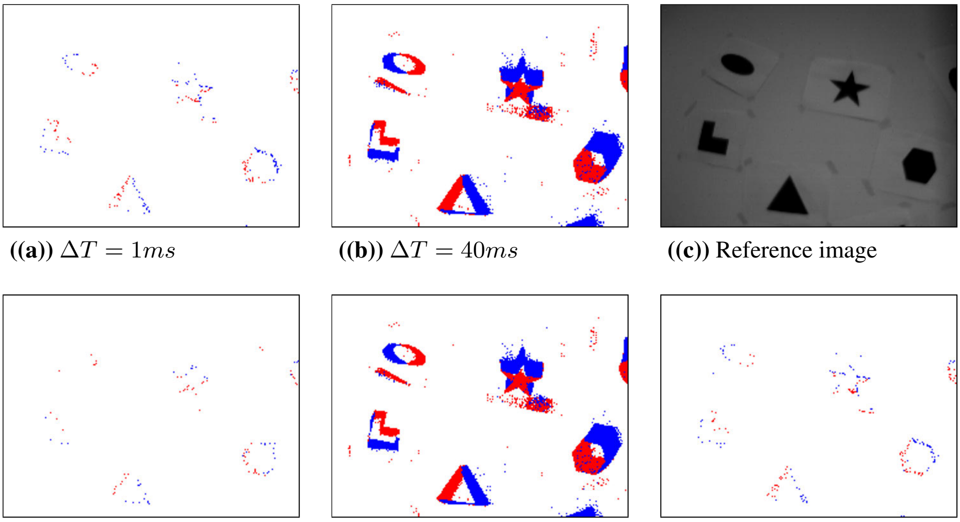
Table 2. Analyzing the quantitative performance of the proposed method.

Table 2 illustrates the quantitative results of the algorithm. Two datasets [53] were used for the evaluation with different environmental complexities. Results show that the proposed algorithm is effective in removing noise caused by sudden movements, high contrast and BA noise. Depending on the scene, the filter can maintain a high SNR (13.6 dB) during static and slow camera motions and on average 6.7 dB in dynamic scenes. The algorithm and results are reported in Paper VI.

3.2 Image construction

With their attractive capabilities, event-based cameras are well suited to obtain the pose of fast and agile robots that operate in complex environments. Because event cameras provide a stream of asynchronous events, processing the data generated by them is not straightforward. Corners and edges, for example, cannot be extracted directly from an event stream, making event-based applications difficult in practice. In order to reconstruct an event, it is hard to predict how many events are needed to construct a meaningful and sharp image from events. The primary question is, how many events must be accumulated to create a meaningful frame? Several studies have sought to solve these problems by pre-defining a number of events (N), e.g., N = 2000 event or a time window (Δ T) to generate a frame [54]. Nevertheless, this type of technique is only applicable in certain situations, for instance when the number of events for a given scene continues to remain constant, which means it cannot provide solutions for all situations. In terms of efficiency and accuracy, such techniques are limited by the camera's velocity and environmental complexity.

In this study, we propose a hybrid technique in which a frame-based camera and an IMU are used to dynamically select events required to build a sharp e-frame. The number of needed events is defined by two factors: 1) the entropy of the scene, that is, how much information is in an image, and 2) the camera velocity, which is the rate at which the environment changes, which affects the number of generated events per second. The assumption here is that the scene is static and the camera is moving.



((a)) $\Delta T = 1ms$ ((b)) $\Delta T = 40ms$ ((c)) Reference image

((d)) $\#Events = 10^2$ ((e)) $\#Events = 10^4$ ((f)) Our approach

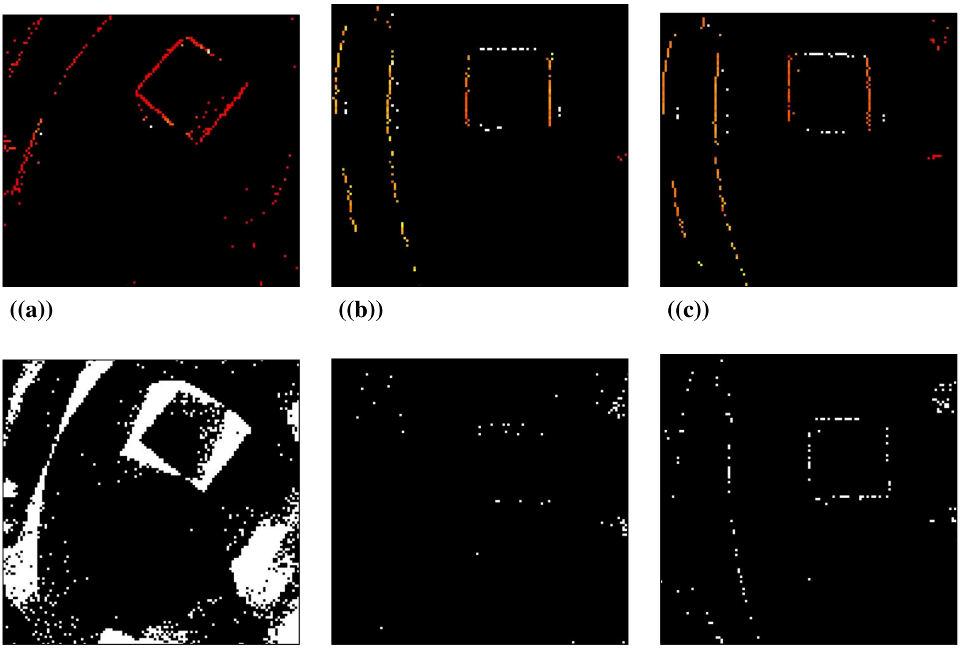
Figure 7. A comparison of different techniques used to construct images from the event stream

Two different values of Δ TS and $\#Events$ are used to reconstruct the frames in 7. This scene involves rapid camera movement, i.e., a large number of events are generated per second (up to 10x10ˆ6 e/s). Using our proposed method, we were able to construct sharp frames in all scenarios, including slow and fast camera movements. A large $/DeltaT$ and $/Events$ will produce blurry frames when the camera is moving slowly. In contrast, small ΔT and $\#Events$ result in uninformative frames. For more details, see Paper III.

3.3 Feature Detection

Feature-based methods are one of the most common approaches to designing a high-level vision-based application. A feature-based approach involves extracting edges or corners from images using different feature detectors. Event cameras generate asynchronous events. Events can be collected over a predefined period of time to detect edges [55] or by constructing an e-frame every a specified number of events [54]. As the camera's speed or the environment's complexity changes, these techniques become inefficient. The reason for this is that the number of events generated over time corresponds to environmental complexity and the speed of the camera [12]. The edge in the constructed frame becomes sparse/bleeding as the events accumulate in a fixed time epoch.

We propose a novel method of selecting events in a scene based on edges to build a sharp e-frame. The selection of events is based on their colonization and the time an event takes to move to one of the 8 neighboring pixels. In order to compute events' lifetime, a local plane is fitted to its surface, and then the velocity of the event is computed (V_x, V_y). By colonizing events, a system that processes a lot of events is able to reduce its computing penalty. This results in improved performance. Our method detects the edges of event cameras in a accurate and environment-resilient manner. Due to its performance advantages over comparable approaches, this technique can also be customized to fit robots powered by embedded boards.



((a)) **((b))** **((c))**

((d)) **((e))** **((f))**

Figure 8. Results of proposed system 8(a), 8(b), and 8(c) compared with a time fixed window algorithm(t = 30ms) 8(d), 8(e), and 8(f) in different camera speed.

The proposed approach produces high-quality edges at varying camera speeds, as illustrated in the figure 8. The speed of the camera affects the number of triggered events, and thus a fixed time interval causes blur at high speeds; our algorithm, however, shows consistent and robust behavior even at high speeds (see Figure 8(a)). At slow movements, our method was able to generate sharp edges and cope with the low number of events generated by the event camera (see Figure 8(b) and 8(e). Lastly, both algorithms performed well in an average speed test. In Paper IV , we describe and elaborate on this approach.

4 Peformance/Accuracy Awarness

It is challenging to enable vision-based applications on platforms with limited computing power since analyzing the captured data such as images and events demands a lot of computational power. For example, event-based cameras in some scenarios can generate $10*10^6$ event/s. Processing of all triggered events will dramatically increase the computing complexity. High performance and accurate implementation of such applications on a mobile robot are difficult due to the resource limitation of mobile robots and the presence of a trade-off between accuracy and performance. Therefore, it is essential to have a resource-aware (RA) system that can co-optimize both accuracy and performance to enable computationally expensive applications, such as vision-based applications on embedded platforms. The RA system will monitor and manage computing resources, i.e. CPU units as well as perform approximation/filtering to the raw data, to ensure those complex CV applications run with high performance and accuracy. By selecting and filtering the incoming events, we can improve the accuracy of the application, e.g., corner detection, and reduce computational load. There are two factors that determine the quality of applications running on embedded systems such as the accuracy and the throughput. Throughput reports the real-time performance of the running application. The throughput value ranges from 0 to 1, with 1 meaning that all captured data was processed on time.

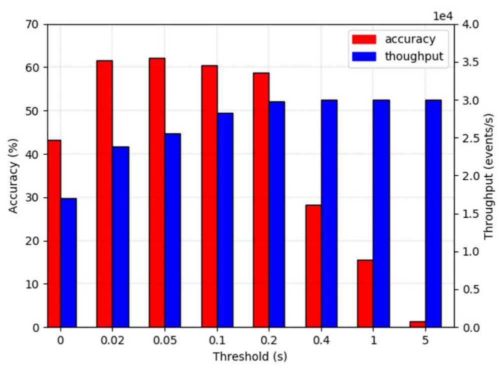


Figure 9. The relationship between accuracy and performance, i.e., throughput vs. approximation, i.e., the timestamp threshold.

Events represent the brightness change with four pieces of information, such as the location of the pixel in the sensor, the time when the change happened, and the polarity of the change [-1,+1], where -1 indicates a decrease in brightness and +1 indicates an increase in brightness. When the camera moves rapidly or there is high contrast in the scene, the event camera may trigger more than one event at the same pixel within a short time frame. Thus, the most recent timestamp does not reflect the actual time when the brightness changes occurred, which affects the performance and accuracy of the performed application. A timestamp threshold filter can be used to filter out redundant events caused by this phenomenon. Figure 9, we plot the accuracy of the application and the achieved throughput which ranges between [0, 3] against various filtering thresholds. There is no doubt, however, that both low filtering rates and over-filtering of events have an adverse impact on accuracy. In addition, the quantity of events retained determines the performance of the system and its computational load. Ultimately, we use a filtering scheme to optimize both the accuracy and throughput of the system.

A resource-aware strategy is proposed that is composed of three layers of filtering to remove noisy events from the pipeline, a monitoring unit, i.e., middleware that monitors the status of the CPU and calculates the throughput error, and the LC-Harris unit which extracts high-quality corners from the processed events. As the filtering process operates, it takes into consideration both the effect of events on accuracy and performance in terms of throughput. To ensure a robust real-time performance under various conditions, the system filters out redundant and noisy events and only passes unique events to be analyzed to perform a certain application, e.g., pose estimation. The selection technique is applied to utilize the correlation of events and their neighbors in the spatial and temporal domain, as well as the status of the computational resources, i.e., CPU.

Monitoring Unit. A monitoring unit continuously observes the internal environment of the robot, which is comprised of its computing units and calculates the throughput error. Throughput values range from 0 to 1, and they indicate how much data was processed in real-time. The timestamp threshold and the frequency of the CPU unit can be adjusted based on the calculated throughput to ensure real-time performance with high efficiency under various conditions.

Filtering Unit. The filtering unit is composed of thee filters including timestamp, arc, and lifetime filter. Additionally to reducing the computational load, the three-layer filtering also improves accuracy by removing outliers. We apply adaptive timestamp filtering in the first layer. The timestamp threshold is dynamically adjusted according to the temporal rate of change and the throughput error reported from middleware. Afterward, a pixel-to-pixel comparisons filtering, namely Arc filter is applied to retain only the corner candidates without increasing the computation load. On the third layer, the lifetime filter computes the time for a corner candidate takes to shift to one of the 8 neighboring events and eliminates all neighboring corner candidates that occur between the timestamp of this corner candidate and its lifetime.

LC-Harris. Various vision-based applications such as odometry and obstacle detection rely on extracted corners to perform the desired task. Harris [1] is one of the common methods to detect strong and resilient corners from the environment. The same techniques could be employed for event cameras, but this would lead to a heavy computational load, as computing gradients in the traditional way are computationally expensive. A new Harris algorithm is proposed that approximates the original score to reduce the computational complexity, as well as three-layer filtering to extract corners. A window size of 9x9 pixels around the incoming corner candidate is extracted and encoded with [0,1], where 1 means there is an event that exits in this pixel. Only the most recent 25 N neighbors are used to improve the runtime complexity. Then, the Harris score is calculated by computing the vertical and horizontal gradients of these local patches as follows,

$$R_h = a^{'} * c^{'} = \sum |I_x| * \sum |I_y| \tag{1}$$

where R_h denotes the Harris score, I_x and I_y are gradients in the x- and y-axis, receptively. In comparison with the traditional Harris, the proposed algorithm can select the best corner candidates, which saves 59% on average on execution time.

To evaluate our method in terms of accuracy against different state-of-the-art methods, we follow a similar evaluation metric as presented in [56]. Two cylinders are used with 3 and 5 pixels radius. Events that fall inside the small cylinder are considered true corners (TC). Events that lay between the small and the large cylinder are false corners (FC). In order to construct the two oblique cylinders, Harris [1] is used to compute the intensity corners, and KLT [2] is used to compute the tracking lines for an extracted corner. Two oblique cylinders are then created by using this line as the center. Table 3 demonstrates that our method outperforms eFAST, Arc*, and FA-Harris. It should be noted that both the eFast and the Arc* are performing only simple pixel-wise operations to improve the performance which leads to poor accuracy.

Table 3. Reported accuracy [%] of various approaches.

Approaches	shapes on wall	boxes in office	walking person	running person	normal office
eFAST[57]	56.401	48.592	51.091	55.9	54.59
eHarris[58]	57.014	49.268	69.262	62.262	61.321
Arc* [56]	55.387	49.012	52.414	53.418	51.214
FA-Harris [59]	57.668	49.662	65.329	49.666	63.664
Ours	**63.202**	**53.276**	**72.182**	**68.761**	**69.626**

Real-time performance, defined as the throughput which indicates the ability to process the raw data on time consistently, is one of the most critical aspects of embedded system applications. An overview of the average throughput of the various approaches can be found in Figure 10. In various scenarios, our method achieves the desired throughput (throughput = 30). In comparison, Arc*, eFAST, and FA_Harris are only capable of delivering the required throughput in situations with low textures (e.g., shapes_6dof and walking). Despite this, these algorithms are incapable of delivering the required throughput in complex environments (e.g., boxes_6dof). The eHarris method fails to achieve the desired throughput in most scenarios because it processes all incoming events and performs costly operations to detect corners. This approach is demonstrated and elaborated in Paper V.

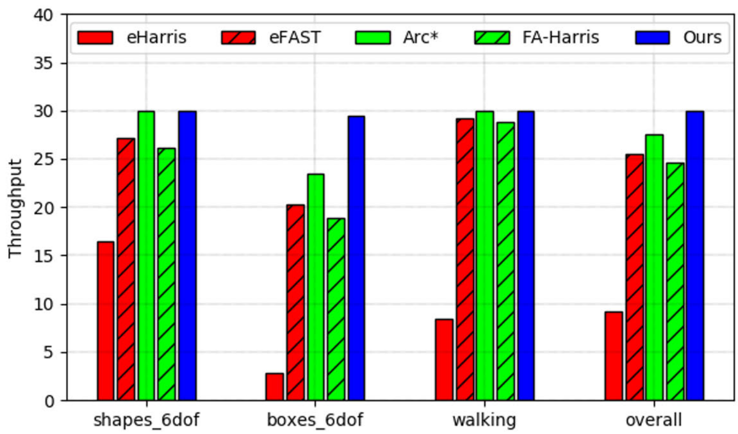


Figure 10. The throughput performance of different methods. The desired throughput is 30.

5 Energy Awareness

In the previous section, we focused on achieving real-time performance and accuracy for event-driven applications, e.g., corner detection on resource-constrained platforms. The main focus was monitoring the status of the computing units of the embedded system using middleware and adaptively tuning filtering parameters in the event-based pipeline to ensure real-time performance as well as achieving high accuracy. In this section, we are taking into account the energy constraints, i.e., the battery capacity of mobile robots. The energy-aware system manages the resource of the mobile system to find an energy-efficient solution under various conditions. Managing energy constraints is a major challenge for mobile robotics. The main focus of approaches in the state-of-the-art has been optimizing the kinematics energy, i.e., energy consumption of mechanical parts. Mobile robots are cyber-mechanical devices, i.e, they contain cyber-parts, e.g., onboard microcontrollers beside mechanical parts such as motors. Both cyber and mechanical parts contribute to the overall energy consumption. For instance, cyber-parts, e.g., onboard computing units draw a large amount of power when they execute heavy vision-based applications, therefore intelligent techniques are required to manage both cyber- and mechanical-parts power consumption.

Generally, existing techniques aim to improve the energy efficiency of the cyber-parts or the mechanical parts of mobile robots, without taking into consideration their relationships with one another. For example, the speed of a mobile robot with convectional cameras affects the quality of the captured images and hence results in a variant computing workload. On the other hand, the robot's speed and the environmental complexity affect the amount of generated events per second and consequently the computing workload. The number of edges in the scene indicates the complexity of the environment. High complexity environments have a lot of edges, resulting in numerous events. In addition, the speed of the camera or the robot affects the number of events triggered per second. Regardless of the same environmental complexity, the number of generated events per second increases significantly when an event-based camera moves rapidly.

A mobile robot is used in the experiments and the vision applications were based on data captured by an event camera. The results are reported in Figure 11 to illustrate the energy consumption per unit distance. We used three applications in this experiment, each with varying computational complexity.

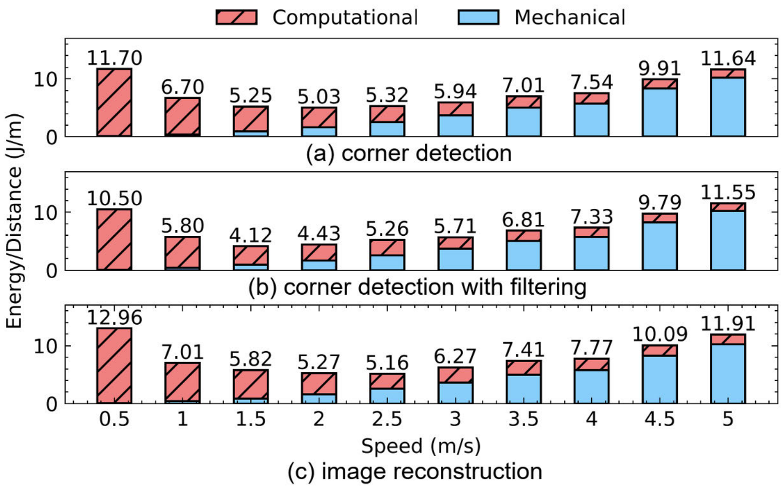
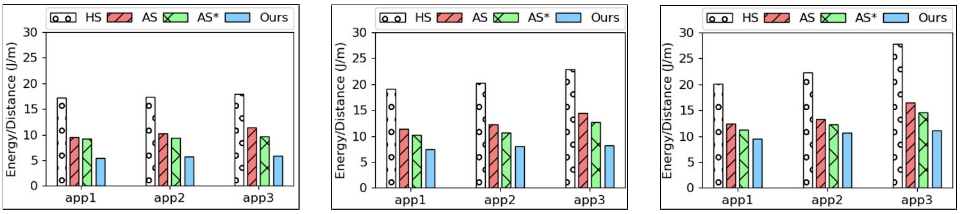


Figure 11. Comparing average power consumption of computational, i.e., cyber parts and mechanical parts of a mobile robot during three different visual applications.

According to the figure, by maximizing the speed to the maximum value, the energy consumption may not be optimized. Actually, by the speed is set to the highest values, it will increase the power consumption of both the mechanical parts and the cyber parts since the rapid movement of the event camera will generate more events. Keeping the robot's speed low is an effective way to save mechanical energy. Even though fewer events are generated, computation energy still dominates. Since the speed of the travel is reduced and the total travel time is also increased, the processing units consume most of the energy as the static energy during the waiting period. We can see that the lowest energy consumption occurs when the robot is moving at a specific speed. A third important point to bear in mind is that the most energy-efficient speed will vary depending on the application. We can see that the lowest energy consumption occurs when the robot is moving at a specific speed. To provide energy-efficient control of the robot, the computational and mechanical controllers should be tuned together at runtime. This dynamic adjustment can be achieved by adjusting the voltage/frequency of the computing units and the robot speed jointly. We present a novel control technique for manipulating the mechanical and computational actuators together in order to achieve the lowest possible energy consumption of a mobile robot. Actuators include the motor voltages to change the speed and the voltages/frequency of the CPU. Actuators are controlled by feeding back data, i.e., the power from the mechanical and computational components in real-time to the controller. To minimize the overall energy consumption, the controller performs fast hill-climbing at runtime in order to optimize the system configuration. As shown in Figure 12, On a robot equipped with an embedded board and an event camera, experiments show that the proposed control algorithm can save battery



((a)) Low-complexity ((b)) Medium-complexity ((c)) High-complexity

Figure 12. Evaluation of the average energy per distance in three different environmental complexity.

energy by an average of 50.5 percent, 41 percent, and 30 percent in three different environmental complexities including a low-complexity, a medium-complexity, and a high-complexity, respectively. An in-depth description of the implementation and the algorithm is provided in Paper VIII.

6 Discussion and Conclusion

Development of CV applications for mobile robots demand careful attention since they have limited resources, including battery capacity and embedded boards with limited computing power. Running vision applications on mobile robots will open new horizons and reshape the future of various domains such as agriculture, transportation, security and health. The runtime complexity, accuracy and performance of these applications depend on many factors, including the complexity of the environment and the speed of the robot and other objects in the scene. High-performance execution of such applications on mobile robots is difficult to achieve consistently. Resource-aware (RA) systems could be a solution to tackle these challenges since they can monitor and manage robot's resources to ensure high performance and efficient operation of complex computations at runtime. In this thesis, we analyze the accuracy, efficiency and performance of various vision-based applications running on a mobile robot under different environmental complexities. We introduce two RA systems: 1) a performance/accuracy awareness system and 2) an energy awareness system.

Designing a computing performance/accuracy RA system is a challenging process since usually there exists a trade-off between accuracy and performance. To increase the accuracy, it is important to process the majority of captured data, which increases the computational load and, consequently, decreases the computing performance. However, discarding the majority of captured data to achieve high performance will result in poor accuracy. In this thesis, we address this problem by considering different aspects of optimization together, i.e., optimizing the software algorithms, filtering the unnecessary data, and tuning the resource management. Based on this, our proposed idea targets to improve three parts of the system: 1) middleware that monitors and manages resources and calculates throughput errors, 2) the filtering phase consisting of three phases that discard events from the pipeline, and 3) vision applications such as a corner detection algorithm, i.e. LC-Harris. In addition to monitoring and managing the computing resources, i.e. CPU units, the system also performs filtering and estimation on the raw data. The middleware system monitors the status of the CPUs and calculates the throughput error at runtime. DVFS is used by the middleware to manage the CPU frequency based on this error. Because redundant events might degrade the quality of the application, e.g. the corner detection, we introduce three-layer filtering to improve both accuracy and performance. The Harris detector uses a scoring technique to detect unique corners; it employs, however, complex calculations to compute the score. We propose an approximation method to reduce the computational complexity, which is able to reduce the execution time by 59% on average. The experiments demonstrate that our method outperforms in terms of accuracy other state-of-the-art methods, such as eFAST and Arc^*, as well as achieves the desired throughput consistently under various conditions.

Mobile robots are considered cyber-mechanical devices, i.e., they contain cyber components, e.g. onboard microcontrollers, along with mechanical components such as motors. The majority of power-saving approaches in the literature aim to improve the power efficiency of the robot's mechanical or cyber parts. Our experiments show that the cyber and mechanical parts have a strong correlation, and both contribute to the overall energy consumption of the robot. For example, during slow robot motion, the cyber parts have a greater impact on the total energy consumption. In contrast, when the robot moves faster, the majority of the energy is consumed by the mechanical parts. In order to achieve a high energy efficiency of a mobile robot running vision-based applications, we propose an algorithm to co-manage both mechanical and cyber parts at runtime. The hill-climbing algorithm is used to find the local optima through tuning both the voltage of the motors to control the robot's speed and the voltage/frequency to control the CPU frequency. In three different cases of environmental complexity, including low complexity, medium complexity, and high complexity, the experiments show that our presented algorithm can save energy on average by 50.5%, 41%, and 30%, respectively.

Vision-based applications such as object tracking, optical flow estimation, and pose estimation are essential for enabling mobile robots to perform various tasks. In the past 30 years, the majority of these applications were based on analyzing images captured by traditional RGB cameras. The problem is that images captured by these sensors contain a huge amount of background information, i.e. redundant data, and they observe the scene at a fixed rate, e.g. 60 Hz. Processing and analyzing these images on a mobile robot is infeasible, which limits mobile robots' capability to perform tasks that require vision-based applications. Event cameras offer a great solution to these challenges since they observe only brightness changes and thereby avoid capturing redundant data. Moreover, event cameras require less power to run: approximately 100 mW compared with 5 W on average for embedded RGB cameras. These characteristics make them an ideal solution to enable vision-based applications on mobile robots. Event cameras, like traditional cameras, have different types of noise due to hardware limitations. Eliminating or filtering out this noise would improve both the performance and accuracy of running applications. We propose a dynamic denoising algorithm that is capable of filtering out two types of noise: 1) the background activity noise and 2) the trailing noise. The results show that the proposed method can achieve and maintain an SNR as high as 13.6 dB in static environments and as high as 6.7 dB in dynamic scenes. As we mentioned previously, event cameras only observe the brightness changes to generate a stream of asynchronous events. To process that output, we present novel methods to construct event frames and extract unique features from the event stream. The evaluation results demonstrate that the proposed image construction method can dynamically construct event frames at various speeds and in different environmental complexities using the aid of an RGB camera and an IMU. Our edge detection method is able to extract sharp edges by utilizing the lifetime of events. Lastly, we present a corner detection algorithm that uses three-stage filtering and an approximation of the Harris detector to extract unique corners from the event stream.

As future work, there are several potential ways to improve the solutions presented in this thesis. The denoising method can be improved, for instance, by implementing an AI-based method to model other sources of noise such as hot pixels, cold pixels, and flickering noise. The challenge is the lack of event-based datasets required to train the model. Ground truth for such sensors is also difficult to obtain. An option to resolve this issue is to use self-supervised models or to simulate the behaviour of an event camera by using complementary sensors (e.g. frame cameras and IMUs) and generate ground truth events, i.e. true events. The performance management system can be improved by considering in the optimization problem other parts that might contribute to the total energy consumption such as communication units. Furthermore, the performance management system can also be improved by applying AI-based optimization to find the global optima. This would raise at least the following challenging questions to address: 1) How much an AI-based model would contribute to the total energy consumption? 2) How to train a robust model? 3) Which constraints need to be considered in order to generate an accurate model?

7 Overview of Original Publications

Presented below is a short summary of the published articles related to this thesis.

7.1 Paper I: A Survey on Odometry for Autonomous Navigation Systems

The development of a navigation system is one of the major challenges in building a fully autonomous platform. Full autonomy requires a dependable navigation capability not only in a perfect situation with clear GPS signals, but also in situations where the GPS is unreliable. Therefore, self-contained odometry systems have attracted much attention recently. This paper provides a general and comprehensive overview of the state-of-the-art in the field of self-contained, i,e, GPS denied, odometry systems and identifies the out-coming challenges that demand further research in future. Self-contained odometry methods are categorized into five main types, i.e., wheel, inertial, laser, radar, and visual where such categorization is based on the type of the sensor data being used for the odometry. Most of the research in the field is focused on analyzing the sensor data exhaustively or partially to extract the vehicle pose. Different combination and fusion of sensor data in a tightly/loosely coupled manner and with filtering or optimizing fusion method have been investigated. We analyze the advantages and weaknesses of each approach in terms of different evaluation metrics such as performance, response time, energy efficiency, and accuracy that can be a useful guideline for researchers and engineers in the field. In the end, some future research challenges in the field are discussed.

- A comprehensive literature review for odometry approaches in the state-of-the-art.

Author's contribution: The author contributed by reviewing and analyzing the literature to provide a comprehensive literature review and extracting the characteristics of the state-of-the-art approaches for comparison. Additionally, he contributed by writing the manuscript.

7.2 Paper II: Monocular Visual Odometry Based on Hybrid Parameterization

Visual odometry (VO) is one of the most challenging techniques in computer vision for autonomous vehicle/vessels. In VO, the camera pose that also represents the robot pose in *ego-motion* is estimated analyzing the features and pixels extracted from the camera images. Different VO techniques mainly provide different trade-offs among the resources that are being considered for odometry, such as camera resolution, computation/communication capacity, power/energy consumption, and accuracy. In this paper, a hybrid technique is proposed for camera pose estimation by combining odometry based on triangulation using the long-term period of direct-based odometry and the short-term period of inverse depth mapping. Experimental results based on the EuRoC data set shows that the proposed technique significantly outperforms the traditional direct-based pose estimation method for Micro Aerial Vehicle (MAV), keeping its potential negative effect on performance negligible.

The primary contribution of this paper is as follows:

- A monocular visual odometry method that utilize a hybrid parameterization, i.e., long-term triangulation of direct-based odometry and short-term inverse depth mapping to achieve high accuracy of pose estimation while maintaining low computational load.

Author's contribution: The author contributed by investigating the monocular visual odometry approaches in the literature, developing the algorithm, implementing the algorithm on hardware and performing the evaluation experiments. The author contributed by writing the manuscript and presenting the work.

7.3 Paper III: Towards Dynamic Monocular Visual Odometry Based on an Event Camera and IMU Sensor

Visual odometry (VO) and visual simultaneous localization and mapping (V-SLAM) have gained a lot of attention in the field of autonomous robots due to the high amount of information per unit cost vision sensors can provide. One main problem in VO techniques is the high amount of data that a pixelated image has, affecting negatively the overall performance of such techniques. An event-based camera, as an alternative to a normal frame-based camera, is a prominent candidate to solve this problem by considering only pixel changes in consecutive events that can be observed with high time resolution. However, processing the event data that is captured by event-based cameras requires specific algorithms to extract and track features applicable for odometry. We propose a novel approach to process the data of an event-based camera and use it for odometry. It is a hybrid method that combines the abilities of event-based and frame-based cameras to reach a near-optimal solution for VO. Our approach can be split into two main contributions that are 1) using information theory and non-euclidean geometry to estimate the number of events that should be processed for efficient odometry and 2) using a normal pixelated frame to determine the location of features in an event-based camera. The obtained experimental results show that our proposed technique can significantly increase performance while keeping the accuracy of pose estimation in an acceptable range.

The primary contribution of this paper is as follows:

- An algorithm to obtain to compute the complexity of the environment, i.e., the entropy by using the aid of an RGB camera.
- A visual odometry method based on event and frame cameras.

Author's contribution: The author contributed by reviewing the model of event cameras, designing and implementing the algorithm, and performing the required experiments to properly evaluate the algorithm. In addition, writing the manuscript and presenting the work.

7.4 Paper IV: Towards real-time edge detection for event cameras based on lifetime and dynamic slicing

Retinal cameras, such as dynamic vision sensors (DVF), transmit asynchronous events with ultra-low latency ($\sim 10\mu s$) only at significant luminance changes, unlike traditional CMOS cameras which transmit the absolute brightness of all pixels including redundant backgrounds. Due to these significant characteristics, they offer great potential to obtain efficient localization of high-speed and agile platforms. Moreover, event cameras have a high dynamic range ($\sim 140dB$), which makes them suitable for platforms that operate indoors in low-lighting scenarios and in outdoor environments, where the camera might be pointing at a strong light source, e.g. the sun. In this paper, we propose an algorithm to detect edges in event streams coming from retinal cameras. To do that, an algorithm is developed to extract edges from events by augmenting a batch of events with their lifetimes. The lifetime of each event is computed using a local plane fitting technique. We use a batching technique to increase the frame rate of generated images since events with a high sample rate cause the processing of a single event to be computationally expensive. The size of the batch will be adjusted based on the mean optical flow of the previously generated batch. The obtained experimental results show that our proposed technique can significantly reduce the response time with the same sharpness in generating the edges.

The key contributions of this paper are as follows:

- An adaptive and robust edge detection algorithm for event based cameras.
- A dynamic batch-based algorithm based on the mean optical flow of previous batches.

Author's contribution: The author contributed by reviewing the literature and design the algorithm. Performing the required experiments and writing the manuscript and presenting it.

7.5 Paper V: Dynamic Resource-aware Corner Detection for Bio-inspired Vision Sensors

Event-based cameras are vision devices that transmit only brightness changes with low latency and ultra-low power consumption. Such characteristics make event-based cameras attractive in the field of localization and object tracking in resource-constrained systems. Since the number of generated events in such cameras is huge, the selection and filtering of the incoming events are beneficial from both increasing the accuracy of the features and reducing the computational load. In this paper, we present an algorithm to detect asynchronous corners form a stream of events in real-time on embedded systems. The algorithm is called the Three Layer Filtering-Harris or TLF-Harris algorithm. The algorithm is based on an events' filtering strategy whose purpose is 1) to increase the accuracy by deliberately eliminating some incoming events, i.e., noise and 2) to improve the real-time performance of the system, i.e., preserving a constant throughput in terms of input events per second, by discarding unnecessary events with a limited accuracy loss. An approximation of the Harris algorithm, in turn, is used to exploit its high-quality detection capability with a low-complexity implementation to enable seamless real-time performance on embedded computing platforms. The proposed algorithm is capable of selecting the best corner candidate among neighbors and achieves an average execution time savings of 59% compared with the conventional Harris score. Moreover, our approach outperforms the competing methods, such as eFAST, eHarris, and FA-Harris, in terms of real-time performance, and surpasses Arc* in terms of accuracy.

The key contribution of this paper is as follows:

- A resource-aware system that monitors and controls the internal resources of the embedded system to achieve the desired throughput consistently.
- A three-layer filtering which removes redundant events from the pipeline to improve both the accuracy and performance by removing redundant data.
- An approximation of Harris corner detector to improve the computational complexity of the detector and enable such algorithm to run on embedded platforms.

Author's contribution: The author contributed by reviewing state-of-the-art approaches, designing and implementing the algorithm. In addition, performing the required experiments, writing the manuscript and presenting the work.

7.6 Paper VI: DBA-Filter: A Dynamic Background Activity Noise Filtering Algorithm for Event cameras

Newly emerged dynamic vision sensors (DVS) offer a great potential over traditional sensors (e.g., CMOS) since they have a high temporal resolution in the order of μs, ultra-low power consumption and high dynamic range up to 140 dB compared to 60 dB in frame cameras. Unlike traditional cameras, the output of DVS cameras is a stream of events that encodes the location of the pixel, time, and polarity of the brightness change. An event is triggered when the change of brightness, i.e., log intensity, of a pixel exceeds a certain threshold. The output of event cameras often contains a significant amount of noise (outlier events) alongside the signal (inlier events). The main cause of that is transistor switch leakage and noise. This paper presents a dynamic background activity filtering, called DBA-filter, for event cameras based on an adaptation of the K-nearest neighbor (KNN) algorithm and the optical flow. Results show that the proposed algorithm is able to achieve a high signal to noise ratio up to 13.64 $dB.$ event cameras, background filtering, KNN, dynamic, noise

The key contribution of this paper is as follows:

- An adaptive and dynamic denoising algorithm that removes two types of noise including the background activity noise and trailing noise.

Author's contribution: The author contributed by reviewing state-of-the-art approaches and designing the algorithm. In addition, define a quantitative evaluation metric and performing the required experiments, writing the manuscript and presenting the work.

7.7 Paper VII: Asynchronous Corner Tracking Algorithm based on Lifetime of Events for DAVIS Cameras

Event cameras, i.e., the Dynamic and Active-pixel Vision Sensor (DAVIS) ones, capture the intensity changes in the scene and generates a stream of events in an asynchronous fashion. The output rate of such cameras can reach up to 10 million events per second in high dynamic environments. DAVIS cameras use novel vision sensors that mimic human eyes. Their attractive attributes, such as high output rate, High Dynamic Range (HDR), and high pixel bandwidth, make them an ideal solution for applications that require high-frequency tracking. Moreover, applications that operate in challenging lighting scenarios can exploit from the high HDR of event cameras, i.e., 140 dB compared to 60 dB of traditional cameras. In this paper, a novel asynchronous corner tracking method is proposed that uses both events and intensity images captured by a DAVIS camera. The Harris algorithm is used to extract features, i.e., frame-corners from keyframes, i.e., intensity images. Afterward, a matching algorithm is used to extract event-corners from the stream of events. Events are solely used to perform asynchronous tracking until the next keyframe is captured. Neighboring events, within a window size of 5x5 pixels around the event-corner, are used to calculate the velocity and direction of extracted event-corners by fitting the 2D planar using a randomized Hough transform algorithm. Experimental evaluation showed that our approach is able to update the location of the extracted corners up to 100 times during the blind time of traditional cameras, i.e., between two consecutive intensity images.

The main contributions of this paper are as follows:

- A matching algorithm that filters incoming events and processes only potential event corner.
- An asynchronous corner tracking algorithm which is based on the life time of events.

Author's contribution: The author contributed by reviewing the literature and designing the algorithm. In addition, performing the required experiments and writing the manuscript.

7.8 Paper VIII: Energy-Efficient Mobile Robot Control via Run-time Monitoring of Environmental Complexity and Computing Workload

We propose an energy-efficient controller to minimize the energy consumption of a mobile robot by dynamically manipulating the mechanical and computational actuators of the robot. The mobile robot performs real-time vision-based applications based on an event-based camera. The actuators of the controller are CPU voltage/frequency for the computation part and motor voltage for the mechanical part. We show that independently considering speed control of the robot and voltage/frequency control of the CPU does not necessarily result in an energy-efficient solution. In fact, to obtain the highest efficiency, the computation and mechanical parts should be controlled together in synergy. We propose a fast hill-climbing optimization algorithm to allow the controller to find the best CPU/motor configuration at run-time and whenever the mobile robot is facing a new environment during its travel. Experimental results on a robot with Brushless DC Motors, Jetson TX2 board as the computing unit, and a DAVIS-346 event-based camera show that the proposed control algorithm can save battery energy by an average of 50.5%, 41%, and 30%, in low-complexity, medium-complexity, and high-complexity environments, over baselines.

The main contribution of this paper is as follows:

- An energy-aware system that enables complex vision-based applications on mobile robots.
- An adaptive hill-climbing to co-optimize both the mechanical and computational parts of a mobile robot.

Author's contribution: The author contributed by reviewing state-of-the-art approaches, designing and implementing the algorithm on a mobile robot. In addition, performing the required experiments, writing the manuscript and presenting the work.

List of References

[1] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[2] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[3] Xiaochuan Yin, Xiangwei Wang, Xiaoguo Du, and Qijun Chen. Scale Recovery for Monocular Visual Odometry Using Depth Estimated with Deep Convolutional Neural Fields. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:5871–5879, 2017.

[4] S. A. S. Mohamed, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila. A survey on odometry for autonomous navigation systems. *IEEE Access*, pages 97466–97486, 2019.

[5] Jawad Naveed Yasin, Sherif Abdelmonem Sayed Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Muhammad Mehboob Yasin, and Juha Plosila. Energy-efficient formation morphing for collision avoidance in a swarm of drones. *IEEE Access*, 8: 170681–170695, 2020.

[6] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[7] A I Mourikis and S I Roumeliotis. A Multi-state Kalman filter for vision-aided inertial navigation. *Proceedings of the {IEEE} International Conference on Robotics and Automation ({ICRA})*, (April):10–14, 2007.

[8] L. Heng and B. Choi. Semi-direct visual odometry for a fisheye-stereo camera. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4077–4084, Oct 2016.

[9] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5):1015–1026, 2008.

[10] Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *In Proc. of the Intl. Sym. of Robot. Research*, 2011.

[11] Sherif A. S. Mohamed, Jawad N. Yasin, M. H. Haghbayan, Antonio Miele, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. Dynamic Resource-aware Corner Detection for Bio-inspired Vision Sensors. In *In Proc. Intl. Conf. of Pattern Recognition*, 2020.

[12] S. A. S. Mohamed, M. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila. Towards dynamic monocular visual odometry based on an event camera and imu sensor. In *Intelligent Transport Systems, From Research and Development to the Market Uptake (INTSYS2019)*. Springer International Publishing, 2020.

[13] Yunxia Chen, Yaohua Wu, and Han Xing. A complete solution for agv slam integrated with navigation in modern warehouse environment. In *2017 Chinese Automation Congress (CAC)*, pages 6418–6423, 2017.

[14] Gu Yunchao and Yang Jiayao. Application of computer vision and deep learning in breast cancer assisted diagnosis. pages 186–191, 01 2019. ISBN 978-1-4503-6612-0.

[15] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: a tailored deep convolutional

neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1):19549, Nov 2020.

[16] Shilpa Sethi, Mamta Kathuria, and Trilok Kaushik. Face mask detection using deep learning: An approach to reduce risk of coronavirus spread. *Journal of Biomedical Informatics*, 120:103848, 2021.

[17] Taihú Pire, Martín Mujica, Javier Civera, and Ernesto Kofman. The rosario dataset: Multisensor data for localization and mapping in agricultural environments. *Int. J. Robotics Res.*, 38(6), 2019.

[18] Jikai Lu, Jianhui Chen, and James J. Little. Pan-tilt-zoom SLAM for sports videos. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 24. BMVA Press, 2019.

[19] Chao Duan, Steffen Junginger, Jiahao Huang, Kairong Jin, and Kerstin Thurow. Deep learning for visual slam in transportation robotics: A review. *Transportation Safety and Environment*, 1: 177–184, 12 2019.

[20] P. Srinivas and A. Kumar. Overview of architecture for gps-ins integration. In *2017 Recent Developments in Control, Automation Power Engineering (RDCAPE)*, pages 433–438, 2017.

[21] Phillip A. Mlsna and Jeffrey J Rodriguez. *Gradient and Laplacian Edge Detection*, pages 495–524. Elsevier Inc., 2009.

[22] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[23] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[24] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[25] Şahin Işık. A comparative evaluation of well-known feature detectors and descriptors. *International Journal of Applied Mathematics, Electronics and Computers*, 3, 01 2014.

[26] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 (1):185 – 203, 1981.

[27] J. Feng, C. Zhang, B. Sun, and Y. Song. A fusion algorithm of visual odometry based on feature-based method and direct method. In *2017 Chinese Automation Congress (CAC)*, pages 1854–1859, Oct 2017.

[28] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 15–22, 2014.

[29] Nicola Krombach, David Droeschel, and Sven Behnke. Combining feature-based and direct methods for semi-dense real-time stereo visual odometry. *Advances in Intelligent Systems and Computing*, 531(July):855–868, 2017.

[30] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3:965–972, 2018.

[31] N. Onkarappa and A. D. Sappa. An empirical study on optical flow accuracy depending on vehicle speed. In *2012 IEEE Intelligent Vehicles Symposium*, pages 1138–1143, 2012.

[32] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa. A robust visual odometry and precipice detection system using consumer-grade monocular vision. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3421–3427, 2005.

[33] N. Nourani-Vatani, J. Roberts, and M. V. Srinivasan. Practical visual odometry for car-like vehicles. In *2009 IEEE International Conference on Robotics and Automation*, pages 3551–3557, 2009.

[34] L. Piyathilaka and Rohan Munasinghe. Multi-camera visual odometry for skid steered field robot. In *2010 Fifth International Conference on Information and Automation for Sustainability*, pages 189–194, 2010.

[35] D. Belter, M. Nowicki, and P. Skrzypczyński. Improving accuracy of feature-based rgb-d slam by modeling spatial uncertainty of point features. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1279–1284, May 2016.

[36] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849. Springer International Publishing, 2014.

[37] P. Liu, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys. Direct visual odometry for a fisheye-stereo camera. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1746–1752, Sep. 2017.

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

[39] D. Caruso, J. Engel, and D. Cremers. Large-scale direct slam for omnidirectional cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148, 2015.

[40] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014.

[41] J. Kelly and G. S. Sukhatme. An experimental study of aerial stereo visual odometry. *IFAC Proceedings Volumes*, 40(15):197 – 202, 2007.

[42] G. Dubbelman and F. C. A. Groen. Bias reduction for stereo based motion estimation with applications to large scale visual odometry. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2222–2229, 2009.

[43] Niko Sünderhauf and Peter Protzel. Towards using sparse bundle adjustment for robust stereo odometry in outdoor terrain. In *Published in Proceedings of towards autonomous robotic systems 2006 (TAROS06)*, pages 206–213, 2006.

[44] T. Kundu and I. Saha. Energy-Aware Temporal Logic Motion Planning for Mobile Robots. In *Proc. Intl. Conf. on Robotics and Automation (ICRA)*, pages 8599–8605, 2019.

[45] C. Cunningham, J. Amato, H. L. Jones, and W. L. Whittaker. Accelerating energy-aware spatiotemporal path planning for the lunar poles. In *Proc. Intl. Conf. on Robotics and Automation (ICRA)*, pages 4399–4406, 2017.

[46] Yongguo Mei, Yung-Hsiang Lu, Y. Hu, and C. Lee. A case study of mobile robot's energy consumption and conservation techniques. In *In Proc. Intl. Conf. on Advanced Robotics*, pages 492–497, 2005.

[47] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *CoRR*, abs/1904.08405, 2019.

[48] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240x180 130dB 3µs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.

[49] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time Intensity Estimation Using Event Cameras. In *Asian Conf. Comput. Vis. (ACCV)*, pages 308–324, 2018.

[50] Sherif Abdelmonem Sayed Mohamed, Jawad Naveed Yasin, Mohammad Hashem Haghbayan, Antonio Miele, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. Asynchronous corner tracking algorithm based on lifetime of events for DAVIS cameras. In *Proc. of Intl. Symp. on Advances in Visual Computing (ISVC)*, pages 530–541, 2020.

[51] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102 (10):1470–1484, 2014.

[52] J Conradt, M Cook, R Berner, P Lichtsteiner, R J Douglas, and T Delbruck. A pencil balancing robot using a pair of aer dynamic vision sensors. In *International Conference on Circuits and Systems (ISCAS)*, pages 781–784, May 2009.

[53] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, feb 2017.

[54] Henri Rebecq, Timo Horstschaefer, Guillermo Gallego, and Davide Scaramuzza. EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017.

[55] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. 09 2018.

[56] I. Alzugaray and M. Chli. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, Oct 2018.

[57] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. In *British Machine Vision Conference (BMVC)*, 2017.

[58] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4144–4149, 2016.

[59] Ruoxiang Li, Dianxi Shi, Yongjun Zhang, Kaiyue Li, and Ruihao Li. Fa-harris: A fast and asynchronous corner detector for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pages 6223–6229. IEEE, 2019.

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

Painosalama, Turku, Finland 2023

ISBN 978-951-29-9145-7 (PRINT)
ISBN 978-951-29-9146-4 (PDF)
ISSN 2736-9390 (PRINT)
ISSN 2736-9684 (ONLINE)