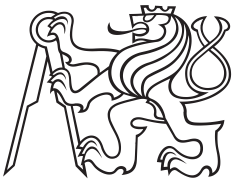
Dissertation Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Interactive Robotic Perception of Cable-Like Deformable Objects

Ondřej Holešovský

Supervisor: prof. Ing. Václav Hlaváč, CSc.
Supervisor–specialist: Mgr. Radoslav Škoviera, Ph.D.
Field of study: Cybernetics and Robotics
Subfield: -
January 2025

Acknowledgements

I am grateful to my supervisor, Václav Hlaváč, for his support, guidance, and patience. Special thanks go to my supervisor-specialist Radoslav Škoviera, who gave me valuable feedback and who collaborated with me the most on the work presented in this thesis. Besides my supervisors, I would like to thank my colleagues at CIIRC with whom I had the pleasure to work with. Karla Štěpánová proofread some of my publications which led to this thesis and suggested valuable improvements. Libor Wagner built some of the experimental hardware equipment for me. Vojtěch Vondráček developed and tested software which helped me obtain the results in the chapter "Evaluation of Event Camera Optical Flow Algorithms". Jiří Sedlář adviced me on the structure of the composite MovingCables dataset. Pavel Krsek and Vladimír Smutný helped me with practical issues regarding illuminance, camera calibration, and others.

Among external collaborators, I am grateful to Roman Vítek (University of Defence, Brno), who contributed the ballistic experimental results to the comparison of event and frame cameras in this thesis. I also thank Andrejs Zujevs (Riga Technical University) for lenting me his DVS240 event camera.

Last but not least, I am grateful to my family and friends for supporting me on this difficult and long journey.

I gratefully acknowledge the project Robotics for Industry 4.0, CZ.02.1.01/0.0/0.0/15_003/0000470 and the TAČR project FW08010076 for financially supporting my research.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu. Při tvorbě textu ani obrázků práce jsem nepoužil generativní umělou inteligenci (AI).

V Praze, 31. ledna 2025

Abstract

Manipulating cluttered cables, hoses or ropes is challenging for both robots and humans. Humans often simplify these perceptually challenging tasks by pulling or pushing tangled cables and observing the resulting motions. We propose to use a similar interactive perception principle to aid robotic cable manipulation. A fundamental building block of such an endeavor is a cable motion segmentation method that densely labels moving cable image pixels. This thesis presents MovingCables, a moving cable dataset, which we hope will motivate the development and evaluation of cable motion segmentation algorithms. The dataset consists of real-world image sequences automatically annotated with ground truth segmentation masks and optical flow. We designed a cable *motion segmentation* method and evaluated its performance on the new dataset. The motion segmentation method operates by thresholding optical flow magnitude estimated by a deep neural network. It assumes that there is only one cable moving in the scene. In order to address this shortcoming, we proposed a novel *motion correlation* method which integrates visual and proprioceptive perception. We formulated the cable interactive segmentation problem in such a way that the *motion correlation* method does not require robot arm segmentation masks. Furthermore, a novel grasp sampling method can propose new cable grasp points given a partial cable segmentation to improve the segmentation via additional cable-robot interaction. We evaluated the proposed *motion correlation* method on data sequences recorded by our physical robotic setup and showed that the method outperforms the *motion segmentation* baseline. All the proposed cable motion segmentation methods rely on traditional frame cameras. Being motivated by the fact that neuromorphic event cameras are more energy and data efficient than frame cameras when capturing sparsely changing scenes such as those with moving cables, we tried segmenting moving cables with event cameras. As of 2024, however, none of the state-of-the-art optical flow estimators for event cameras we tested was suitable for cable motion estimation. We also experimentally compared event cameras with frame cameras on high speed motion sensing tasks, such as observing flying bullets or markers rotating on a disk. The experimental results include sampling/detection rates and position estimation errors as functions of illuminance and motion speed; and the minimum pixel latency of two commercial event cameras (ATIS, DVS240). The event cameras responded more slowly to positive than to negative large and sudden contrast changes. The event cameras we tested were limited by pixel latency when tracking small objects at very high speeds, resulting in motion blur effects. Sensor bandwidth limited them when recognizing larger objects. The event camera spatial sampling density monotonically decreases with growing motion speed, which may limit the applicability of existing event-based algorithms relying on fixed-sized event batches. Both camera types provided comparable position estimation accuracy but event cameras were more bandwidth-efficient in our experiments.

Keywords: motion segmentation, dataset, cables, deformable, optical flow, robotic manipulation, interactive perception, deep learning, event cameras, computer vision, artificial intelligence

Supervisor: prof. Ing. Václav Hlaváč, CSc.
Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Jugoslávských partyzánů 1580/3, 160 00 Prague, Czech Republic

Abstrakt

Manipulace s nepřehlednými kabely, hadicemi nebo lany je náročná pro roboty i pro člověka. Člověk často zjednodušuje tyto na vnímání náročné úkoly taháním za zamotané kabely a pozorováním výsledných pohybů. Navrhujeme použít podobný princip interaktivního vnímání pro usnadnění robotické manipulace s kabely. Základním stavebním kamenem takového snažení je metoda pohybové segmentace kabelu, která umí v obrazu označit pixely pohybujícího se kabelu. Tato práce představuje MovingCables, dataset pohybujících se kabelů, který, jak doufáme, bude motivovat vývoj a testování algoritmů pro pohybovou segmentaci kabelů. Dataset se skládá z obrazových sekvencí nasmínaných v reálném světě, které jsou automaticky anotované segmentačními maskami a optickým tokem. Navrhli jsme metodu pohybové segmentace (motion segmentation) kabelu a vyhodnotili její přesnost na novém datasetu. Metoda pohybové segmentace funguje na základě prahování velikosti optického toku odhadovaného hlubokou neuronovou sítí. Předpokládá, že se ve scéně pohybuje pouze jeden kabel. Abychom toto omezení odstranili, navrhli jsme novou metodu pohybové korelace (motion correlation), která integruje vizuální a proprioceptivní vnímání. Problém interaktivní segmentace kabelu jsme formulovali tak, aby se metoda pohybové korelace obešla bez segmentačních masek ramene robota. Nově navržená metoda pro výběr úchopů (grasp sampling) navíc umí na základě částečné segmentace kabelu navrhnout nové body pro uchopení kabelu tak, aby se segmentace kabelu mohla zlepšit pomocí dodatečné interakce mezi robotem a kabelem. Navrhovanou metodu pohybové korelace jsme vyhodnotili na datových sekvencích zaznamenaných naším fyzickým robotickým pracovištěm. Metoda pohybové korelace na těchto sekvencích dosahuje vyšší přesnosti než původní metoda pohybové segmentace. Všechny námi navržené metody pohybové segmentace kabelu využívají tradiční kamery. Biologicky inspirované událostní kamery jsou energeticky a datově efektivnější než tradiční kamery při sledování řídce se měnících scén, jako jsou například scény s pohybujícími se kabely. Proto jsme ještě vyzkoušeli pohybovou segmentaci kabelů s událostními kamerami. V roce 2024 však nebyla žádná z námi testovaných metod pro odhad optického toku z událostních kamer vhodná pro odhad pohybu kabelů. Experimentálně jsme také porovnali událostní kamery s tradičními kamerami na úlohách snímání velmi rychlých pohybů, jako je pozorování letících balistických projektilů nebo sledování kontrastních značek otáčejících se na rotujícím disku. Experimentální výsledky obsahují vzorkovací/detekční rychlosti a chyby odhadu polohy jako funkce osvětlení a rychlosti pohybu; a také minimální časy odezev pixelů dvou komerčních událostních kamer (ATIS, DVS240). Událostní kamery reagovaly pomaleji na pozitivní než na negativní velké a náhlé změny kontrastu. Událostní kamery, které jsme testovali, byly omezeny odezvou pixelů při sledování malých objektů při velmi vysokých rychlostech, což způsobovalo efekt pohybového rozmazání. Datová propustnost senzorů omezovala událostní kamery při rozpoznávání větších objektů. Také jsme zjistili, že prostorová vzorkovací hustota událostní kamery monotónně klesá s rostoucí rychlostí pohybu. To může omezovat použitelnost některých stávajících algoritmů s dávkovým zpracováním událostí, které pracují s konstantním počtem událostí v každé dávce. Oba typy kamer, tradiční i událostní, poskytovaly srovnatelnou přesnost odhadu polohy objektů, ale událostní kamery v našich experimentech při stejných scénách přenášely menší objemy dat za jednotku času.

Klíčová slova: pohybová segmentace, dataset, kabely, deformovatelné, optický tok, robotická manipulace, interaktivní vnímání, hluboké učení, událostní kamery, počítačové

vidění, umělá inteligence

Překlad názvu: Interaktivní robotické vnímání deformovatelných objektů podobných kabelům

Contents

Chapter 1

Introduction

1.1 Motivation

Manipulating one-dimensional deformable objects such as cables, hoses or ropes (henceforth referred to as "cables" for brevity), especially when cluttered, is challenging both for humans and robots due to self-occlusions, high-dimensional state space, uniform visual appearance, and complex interaction dynamics. Imagine, for example, that a robot should replace a specific damaged cable in the scene shown in Fig. 1.1. There are passive computer vision methods [Choi et al., 2023, Caporali et al., 2023b, Caporali et al., 2022a] for segmenting individual cable instances. However, these methods struggle with occlusions or complex intersections of multiple cables. Novel cable segmentation methods are therefore needed.

Figure 1.1: One of the untidy cables in the scene is moving. Its motion segmentation by our MfnProb motion segmentation method is in green.

Our work is inspired by the way humans interactively discover the topology of cluttered cables when trying to untangle them. When a human finds it too hard to visually infer whether two cable segments are directly linked, she grasps and pulls or pushes one of them. The motion visually distinguishes the grasped cable from the clutter. This observation guides us to integrate perception and interaction to aid robotic cable manipulation.

1.2 Task formulation

Given the location of a short segment of an entangled cable, a single-arm perceiving robot wants to use interactive perception to find all other movable segments belonging to the same cable. Interactive perception is the exploitation of forceful robot-environment interactions to simplify and enhance perception [Bohg et al., 2017, Tsikos and Bajcsy, 1991]. Interactive perception is a research field at the intersection of computer vision and robotics, which themselves belong to the even larger research field of artificial intelligence.

For the purpose of interactive exploration and manipulation, a cable (segment) is a deformable three-dimensional object. Its length c_l is the largest dimension, measured linearly in its fully extended state. The skeleton (backbone) of the cable is a deformable space curve whose length is also equal to c_l. Cable width c_w is approximately equal to its height. The ratio of cable length and width is larger than a constant k_{lw}, $\frac{c_l}{c_w} > k_{lw}$. The allowed cable width is between minimum c_{wmin} and maximum c_{wmax} bounds, $c_{wmin} < c_w < c_{wmax}$. None of the three dimensions (length, width, height) changes significantly as a result of forces exerted

on the cable by the robot. The exception is the allowed cable deformation at the grasp point due to grasping, i.e. pressing the cable grasp point between two gripper fingers.

We formulate and address the task in two levels of complexity:

1. Segmenting a single moving cable when no other cable in the scene moves.
2. Segmenting a grasped cable when the grasped cable can perturb neighboring cables, causing multiple moving cables.

1.2.1 Single moving cable segmentation

A human or a robot arm moves only a single cable in the scene, the motion of any other cables is negligibly small. A camera records an image sequence of the cable motion. The goal of a moving cable segmentation algorithm is to segment (label) the moved cable pixels in each image of the recorded sequence. The motion segmentation methods addressing this problem and presented in this thesis assume that either the segmentation mask of the arm moving the cables is available or the arm is not visible in the image.

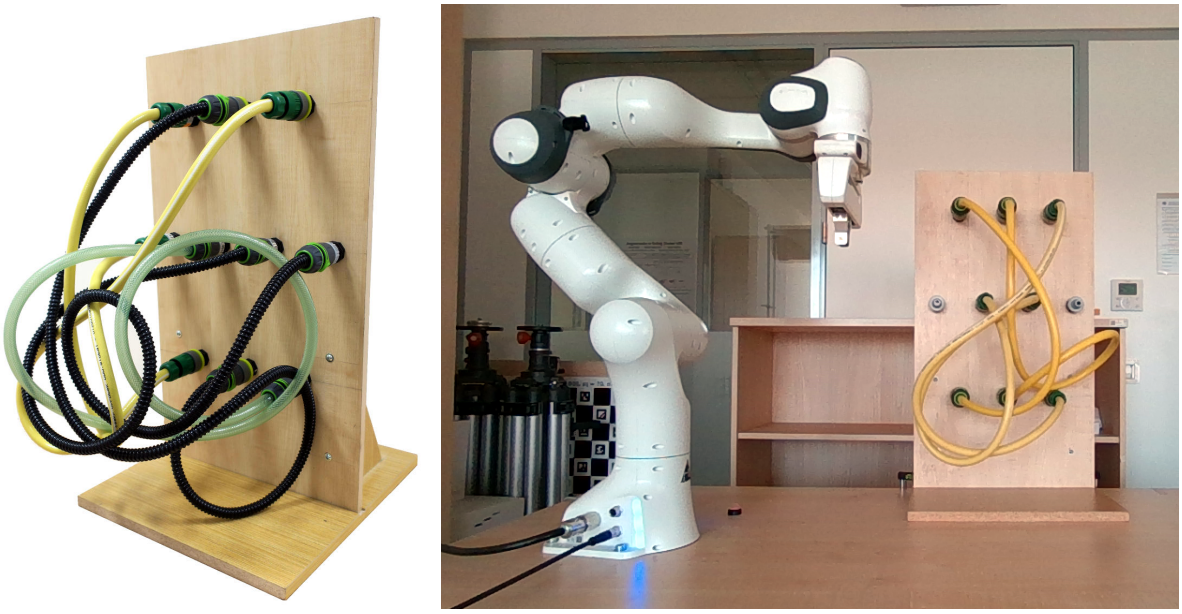
We tried to solve the single moving cable segmentation task using both traditional frame cameras (Chapters 4 and 5) and neuromorphic event cameras (only Chapter 8). We tried using event cameras as they are more energy- and data-efficient image sensors than traditional frame cameras when capturing sparsely changing scenes such as those with moving cables.

1.2.2 Interactive cable segmentation with perturbations

Our approach to solving the more general task with neighboring cable perturbations requires a robot grasping and moving the cable to be segmented. Without loss of generality, we scale the experimental workspace to the size of our robot and its gripper, see Fig. 1.2. The rigid part of the workspace consists of a horizontal table, a vertical board, and male and female garden hose connectors. Male garden hose connectors are mounted on the rigid vertical board. Each hose endpoint is fitted with a female garden hose connector which is in turn connected to one of the male connectors on the board. The hoses hang freely.

An RGB-D camera (a traditional frame camera) fixed to the world frame observes the workspace. We assume that the robot knows the transformation between the camera coordinate frame (pixel coordinates and depth) and the robot coordinate frame. We also assume that a short graspable cable segment visible in the camera image is given, so that the robot can start interacting with the cable of interest to explore it. However, the cables can occlude one another elsewhere due to their crossings, so the cables are only partially observable.

Cable exploration starts by grasping the given initial cable segment. The robot interacts with the grasped cable, moving it in various directions. These robot motions can sometimes perturb neighboring cables, e.g. by the robot or the grasped cable hitting the neighboring cables. Given the robot gripper trajectories (proprioceptive measurements) and the sequence of images captured by the RGB-D camera, the goal of an interactive motion segmentation algorithm is to label/segment the image pixels belonging to the grasped cable. The algorithm can provide these labels (motion segmentations) in any image frame but does so typically in the last captured image of the sequence. To further increase cable segmentation recall, a grasp



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

Figure 1.2: The robotic experimental workspace. (**a**) The experimental task board with garden hose connectors and tangled hoses. (**b**) Franka Emika Panda robot workspace with the task board as seen by an RGB-D camera fixed to the world frame.

segment sampling algorithm can propose another suitable cable segment from the motion segmentations. The robot grasps this proposed segment, moves the cable, and performs the motion segmentation again. Finally, the interactive motion segmentation algorithm gathers all the cable segmentations in a common image frame for use in downstream manipulation tasks.

1.3 Thesis contributions

The contributions of this thesis include:

1. MovingCables, the first moving cable segmentation dataset with optical flow and instance segmentation ground truth, is automatically generated by a novel data annotation method. (Chapter 4, published in [Holešovský et al., 2024].)
2. MfnProb, a novel cable motion segmentation algorithm based on an optical flow prediction neural network with probabilistic outputs. (Chapter 5, published in [Holešovský et al., 2024].)
3. An evaluation of five cable motion segmentation algorithms (including MfnProb) on the MovingCables dataset demonstrates how the dataset can be used. (Chapter 5, published in [Holešovský et al., 2024].)
4. A *motion correlation* method able to segment a grasped cable by moving it in a cluttered environment even when the cable or the robot sometimes perturb neighboring cables. (Chapter 6, submitted in [Holešovský et al., 2025].)

5. A grasp sampling method which can propose new cable grasp points given a partial cable segmentation to improve the cable segmentation via additional cable-robot interaction. (Chapter 6, submitted in [Holešovský et al., 2025].)

6. A formulation of the cable motion segmentation problem which does not require robot arm segmentation masks. (Chapter 6, submitted in [Holešovský et al., 2025].)

7. An evaluation of the proposed *motion segmentation/correlation* methods on data sequences recorded by our physical robotic setup. (Chapter 6, submitted in [Holešovský et al., 2025].)

8. The first experimental, explicit, high-speed motion benchmark of event- and frame-cameras. Our experiments brought several novel findings: (a) The quantification of the ON/OFF response latency difference and its impact on high-speed object detection or tracking; (b) The implications of the burst-mode event readout [Posch et al., 2011, Guo et al., 2007] operating close to its bandwidth limit; (c) The event-camera spatial sampling density monotonically decreases with growing motion speed. Therefore, fixed-sized event batches do not guarantee perfect speed-adaptive scene sampling as previously assumed [Liu and Delbruck, 2018], even if the scene appearance does not change; (d) The position estimation accuracy of frame and event-cameras is comparable when tracking small high-speed objects. (Chapter 7, published in [Holešovský et al., 2021] and [Holešovský et al., 2020].)

9. An evaluation of two state-of-the-art (as of 2024) optical flow estimators for event cameras on the task of cable motion estimation. We have found that none of them is suitable for moving cable motion estimation or segmentation. (Chapter 8, not published or submitted before.)

Chapter 2

Key Concepts

2.1 Digital camera

A digital camera is a device able to capture images and videos via a lens and a planar electronic image sensor. A camera lens focuses light rays onto the image sensor. In this thesis, we geometrically approximate cameras with lenses using the pinhole camera model that projects all light rays through a common center of projection [Szeliski, 2011]. The pinhole camera model projects a scene point with 3D coordinates (X, Y, Z) in metric units to an image point expressed in image pixel coordinates (x, y).

$$x = \frac{f_x}{Z} X + c_x, \tag{2.1}$$

$$y = \frac{f_y}{Z} Y + c_y, \tag{2.2}$$

where f_x, f_y is the camera focal length measured in horizontal and vertical pixel lengths, respectively, and (c_x, c_y) are the pixel coordinates of the optical center. In practice, $f_x \approx f_y$ tends to hold as image sensor pixels are usually square-shaped. The parameters f_x, f_y, c_x, c_y are intrinsic camera parameters and one can obtain them by camera calibration [Szeliski, 2011]. The Z axis is aligned with the camera optical axis, it points away from the camera and the center of projection (the focal point) is its origin. It is perpendicular to the image sensor plane. The Z variable is called the depth of the 3D scene point. See Fig. 2.1 for an illustration of the pinhole model.

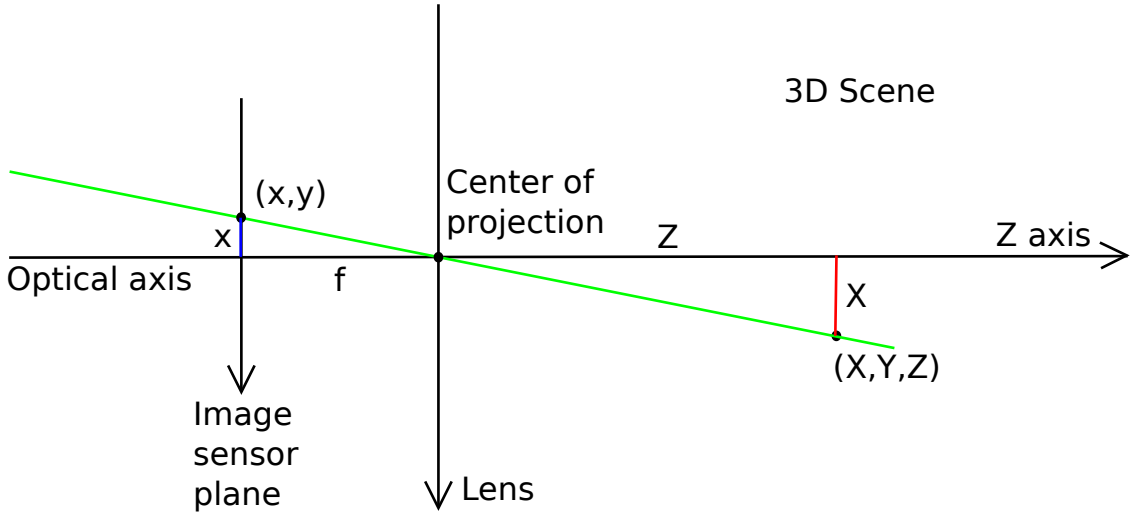


Figure 2.1: The pinhole camera model.

Image sensors of traditional frame cameras aim to capture still image frames. They (usually) do it in one of these two ways:

- *Global shutter* cameras expose all sensor pixels simultaneously to incoming light and then sequentially read out their intensity values.

- *Rolling shutter* cameras expose and read out sensor pixels sequentially by rapidly scanning the sensor pixel array horizontally or vertically, typically row-by-row.

The duration for which a pixel accumulates incoming light photons to measure the light intensity is called exposure time.

2.2 Event camera

Event-cameras, also known as Dynamic Vision Sensors (DVS), have been popular among academic researchers since approximately 2010. Independent pixels of event-cameras [Mahowald, 1992], [Lichtsteiner et al., 2008] generate asynchronous events in response to local logarithmic intensity changes. Each pixel performs level-crossing sampling of the difference of logarithmic brightness sensed by the pixel. Each time the difference passes a preset threshold, the pixel emits a change detection (CD) event and resets its brightness reference to the current brightness. A CD event is characterized by its pixel coordinates, its precise timestamp in microsecond resolution, and the polarity of the brightness change. The advantages of event cameras over traditional cameras include lower sensor latency, higher temporal resolution, higher dynamic range (120 dB+ vs. ∼60 dB of traditional cameras), implicit data compression, and lower power consumption.

2.3 Optical flow

Optical flow in traditional frame cameras is an independent per-pixel estimate of motion between two images [Szeliski, 2011], target and reference. Given the target image I_t sampled at $x_i \in \mathbb{R}^2$ discrete pixel locations, optical flow vectors $\phi_i \in \mathbb{R}^2$ estimate the location of these pixels in the reference image I_r. The optical flow minimizes the brightness or color difference between corresponding pixels summed over all the pixel locations of the target image, $\sum_i [I_r(x_i + \phi_i) - I_t(x_i)]^2$.

We distinguish between two types of optical flow: full optical flow and "normal flow". Sufficiently textured image regions such as corners allow full optical flow estimation. In contrast, when an image contains textureless cables or rods, straight or slightly curved edges are the only visual features an optical flow estimator can see. In that case, one can only estimate "normal flow", which is the normal projection of the (full) optical flow vector on the normal unit vector of the edge (e.g. the cable/rod boundary).

2.4 Convolutional neural networks

A convolutional neural network (CNN) is an artificial neural network that uses the linear mathematical operation called convolution in at least one of its layers. CNNs are designed for processing data naturally stored in a grid, such as time series or 2D images. Convolutional layers are translation (or shift) invariant and very efficient when applying the same linear transformation to each small region of a large input [Goodfellow et al., 2016].

Convolutional and other layers in a CNN contain parameters (weights) which need to be learned (estimated) on a training dataset. When a neural network learns, it first predicts an output given its current set of weights and a training input sample. In a supervised learning setting, an objective (loss) function computes the value of the training loss given the predicted output and the desired (ground truth) output from the training dataset. The goal of the training process is to minimize the training loss, such that the predicted outputs match the ground truth outputs as closely as possible. Neural networks usually minimize the training loss using the stochastic gradient descent algorithm with gradient backpropagation computing the gradient of the loss function with respect to all the network weights [Goodfellow et al., 2016].

In this thesis, we mostly use convolutional neural networks to estimate the optical flow between two input color images. Other CNNs in this thesis predict optical flow or reconstruct intensity images from change detection (CD) events.

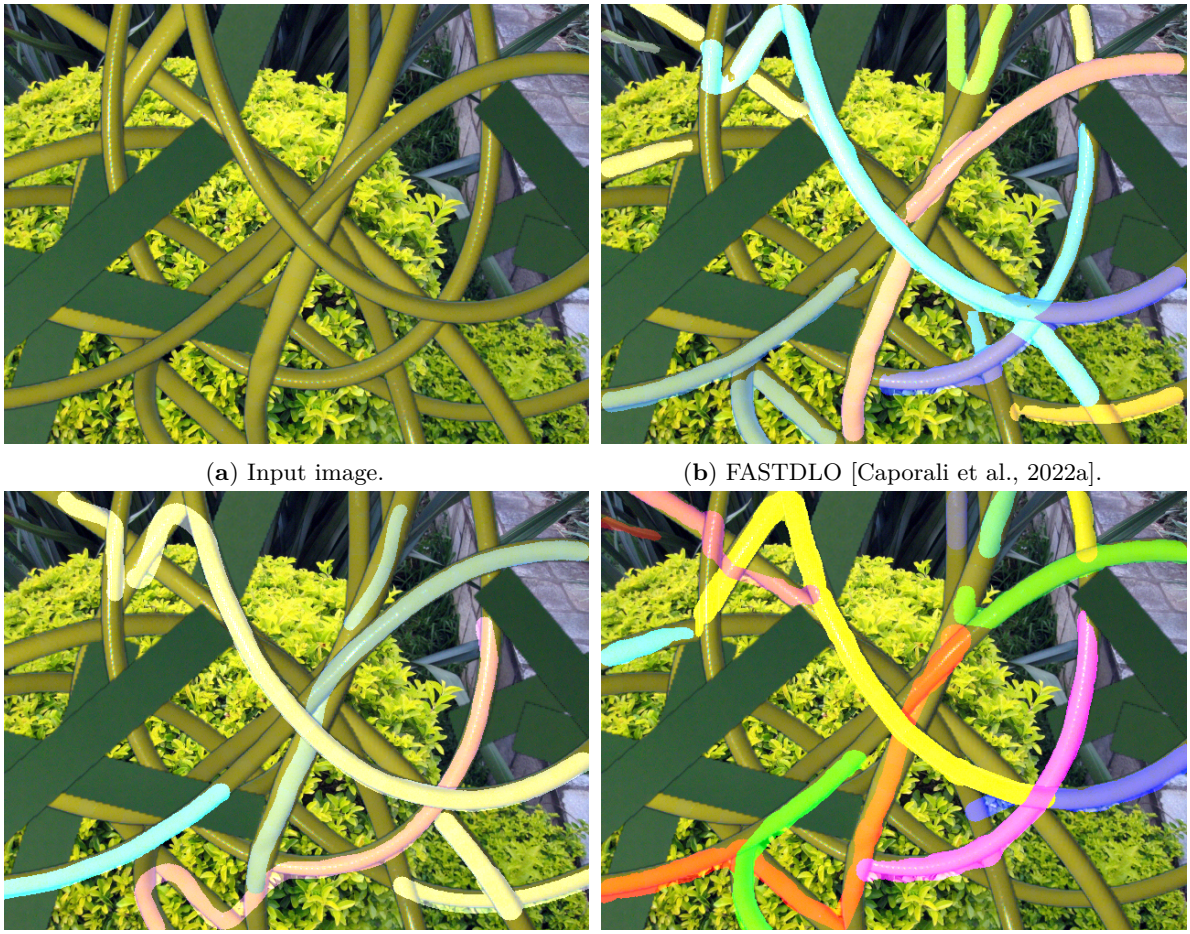
Chapter 3

State of the Art

3.1 Cable perception

Cable segmentation is generally challenging because cables are often of uniform appearance without distinctive features. Several cable detection or segmentation methods in the literature thus relied on simplifying assumptions. Some assumed a single cable was present in the scene [Yan et al., 2020, Wnuk et al., 2020], others relied on a good cable/background color contrast or on color thresholding to segment the cables from the background [Yan et al., 2020, Zhu et al., 2020, Zhu et al., 2021, Keipour et al., 2022, Viswanath et al., 2021, Shivakumar et al., 2023, Lv et al., 2023].

A DeepLabV3+ semantic segmentation neural network can segment wires in an image [Zanella et al., 2021]. Ariadne+ [Caporali et al., 2022b] segmented individual wires by processing a superpixel region adjacency graph, taking advantage of the DeepLabV3+ semantic segmentations. An additional TripleNet network predicted the superpixel connectivity scores at wire intersections.



(**a**) Input image.

(**b**) FASTDLO [Caporali et al., 2022a].

(**c**) RT-DLO [Caporali et al., 2023b].

(**d**) mBEST [Choi et al., 2023].

Figure 3.1: Instance segmentation of an image from our dataset.

FASTDLO [Caporali et al., 2022a] is a recent state-of-the-art passive wire instance segmentation method. It skeletonized each foreground segment predicted by the DeepLabV3+

network to find cable sections, intersections, and endpoints. At each intersection, a similarity neural network paired the neighboring segments with similar color, thickness, and direction estimates. The more recent RT-DLO [Caporali et al., 2023b] method replaced FASTDLO's skeletonization with a sparse graph-based approach to handle degraded foreground segments. mBEST [Choi et al., 2023] found cable instances in skeletonized foreground segments by minimizing the cumulative bending energy of the cables. FASTDLO, RT-DLO, and mBEST may struggle with multiple overlapping cables and severe occlusions, see Fig. 3.1. We note, however, that scenes involving occlusions or more than two cables at an intersection were outside the scope of mBEST [Choi et al., 2023]. Zhaole et al. showed that the semantic segmentation networks [Zanella et al., 2021, Caporali et al., 2022a] trained on wire datasets do not generalize well to cables of different textures and color patterns (e.g. ropes) [Zhaole et al., 2024]. Their combination of the Segment Anything large vision model with a post-processing method outperformed [Zanella et al., 2021, Caporali et al., 2022a] in segmenting a cable from the background. More recently, a Perceiver-inspired architecture segmented cable instances in a color image given a specific text-based prompt [Caporali et al., 2024].

Several methods can track cables across multiple video frames given an instance segmentation in the first frame [Chi and Berenson, 2019, Wang et al., 2021, Xiang et al., 2023]. DLO3DS [Caporali et al., 2023a] estimated the 3D shape of static cables from multiple views captured by an eye-in-hand 2D camera. It relied on cable instance segmentation provided by FASTDLO.

Deep networks can replace cable state estimation algorithms when task-specific human-labeled training data is available. They can propose interaction keypoints, detect endpoints, classify knots, or refine grasps. Such networks were applied to untangle a multi-cable knot [Viswanath et al., 2021], a non-planar knot [Sundaresan et al., 2021] or a long cable [Shivakumar et al., 2023]. In [Shivakumar et al., 2023], an interactive perception algorithm preferred certain manipulation primitives over others when the perception was uncertain. Nevertheless, these approaches assumed that the cables were segmentable from the background by color thresholding. A deep network also helped a robot pick a wiring harness entangled in a pile of wiring harnesses [Zhang et al., 2023]. It predicted the success probability of each available open-loop action given a grasp candidate and a depth image of the scene.

Our work exploits the motion of a cable of interest to simplify the cable perception task, even in complex scenes with multiple overlapping cables and severe occlusions.

3.2 Interactive segmentation

Interactive perception is the exploitation of forceful robot-environment interactions to simplify and enhance perception [Bohg et al., 2017, Tsikos and Bajcsy, 1991]. Interactive segmentation [Kenney et al., 2009], a more specific interactive perception skill, interacts with the environment and segments it into a set of movable objects based on the observed motion. It is computationally efficient and requires little prior knowledge about the environment.

Interactive segmentation processes a visual motion signal to segment the moving objects. Options to consider include intensity image differencing with 2D template tracking [Kenney et al., 2009], dense optical flow [Singh et al., 2021, Boerdijk et al., 2020, Patten et al., 2018, Eitel et al., 2019], sparse feature tracking [Patten et al., 2018, Price et al., 2021], object

trackers [Price et al., 2021]. Compared to optical flow, intensity change detection performs poorly when the moved object and the background are of similar color [Boerdijk et al., 2020] or when multiple objects move [Eitel et al., 2019]. Change detection used together with optical flow improves robustness under strong occlusion, where never-reappearing pixels degrade optical flow [Patten et al., 2018]. One cannot apply sparse feature tracking to most cables due to their uniform visual appearance.

There are several ways of processing the motion signals to segment the objects. Optical flow clustering [Singh et al., 2021, Eitel et al., 2019] followed by the rejection of rotating objects and the segments not overlapping with the initial gripper location [Eitel et al., 2019] relied on the rigid object assumption. Optical flow thresholding can segment a rigid or deformable moving object when nothing else moves in the scene [Boerdijk et al., 2020, Holešovský et al., 2024], e.g. when a passive segmentation method segments the robot arm. Optical flow or object trackers can also track segmentation hypotheses between image frames [Patten et al., 2018, Price et al., 2021, Fang et al., 2024, Lu et al., 2023, Qian et al., 2024]. The hypothesis representations used for that are diverse. [Patten et al., 2018] proposed a probabilistic segmentation framework to update an octree neighborhood graph, where each node represented a voxel and each edge encoded their similarity. [Price et al., 2021] sampled segmentations from a segmentation tree generated by Convolutional Oriented Boundaries. [Fang et al., 2024] generated segmentation hypotheses with confidence estimates by prompting the Segment Anything Model (SAM). [Lu et al., 2023] tracked undersegmented passive instance segmentation masks of rigid objects across multiple pushing interactions to obtain more reliable segmentations. [Qian et al., 2024] performed interactive rigid object segmentation while assuming that several points could be tracked on each object and that an initial (under)segmentation was available. Their method did not require object singulation.

We have not found any motion segmentation method tested on cluttered cables. To segment cables, we started with a method based on thresholding the magnitude of optical flow predicted by an off-the-shelf neural network [Zhao et al., 2020]. Next, we improved its results by extending it with probabilistic outputs [Gast and Roth, 2018] and by retraining it on standard optical flow datasets. Finally, we integrated visual and proprioceptive perception to aid robotic cable exploration even when the robot or the grasped cable sometimes perturb neighboring cables.

3.3 Cable datasets

We are not aware of any existing moving cable dataset. [Zanella et al., 2021] published a static cable dataset for training and evaluating segmentation methods. They took photos of wires on a monochromatic background and randomized it using the chroma key technique. In [Caporali et al., 2023c], a human labeled 3D keypoints along a real-world wire using a VR tracker pen. A camera mounted on a robotic arm took images of the wire from different viewpoints. The authors trained semantic and instance segmentation networks on dataset mixtures containing different proportions of synthetic and real-world images. They showed that adding real-world training data improved accuracy at test time.

We propose MovingCables, a novel dataset utilizing UV fluorescent markers to obtain the motion ground truth. UV fluorescence provided the ground truth in datasets for optical flow

[Baker et al., 2010] and the semantic segmentation of rigid and deformable objects [Takahashi and Yonekura, 2020, Thananjeyan et al., 2022]. [Baker et al., 2010] painted fluorescent speckles onto several objects, including clothes. They switched between visible and UV light to record images with and without the speckles. The Lucas-Kanade algorithm estimated the ground truth optical flow even for low-textured objects thanks to the speckles. Instead of relying on speckles, we opt for stripe markers to obtain uninterrupted marker trajectories extending across the entire video recording.

3.4 Comparing event cameras and frame cameras

Several authors deal with the comparison of event- and frame-cameras. [Barrios-Avilés et al., 2018] test the object detection latency of event and standard cameras. Their vision system detects a black circular dot rotating on a white disk and estimates the position of the dot for control purposes. Surprisingly, the authors report latency differences between the two cameras in the order of 100 ms, despite the frame rate of the standard camera being 64 fps at VGA image resolution. It is unlikely that such long latency would be caused by the cameras or by the object detection algorithm based on image intensity thresholding running on the standard camera frames.

Reconstruction of images from an event-camera is more complicated compared with frame-cameras. The state-of-the-art approach to cope with this task was published in [Rebecq et al., 2019a], who compare the quality of images reconstructed from events to standard camera frames. The reconstructed images better capture the dynamic range of the scene than the standard frames. The authors also compare visual-inertial odometry algorithms running on traditional camera frames and images reconstructed from events. Event-based reconstructed intensity image results are reported to be on average superior to the results of traditional frames and the state-of-the-art methods running on events directly. However, the first is no surprise as the chosen traditional camera frame rate was only 20 frames per second, and the captured frames suffered from severe motion blur, probably due to overlong exposure time.

[Boettiger, 2020] goes into great depth of analysis and comparison of the properties of both event- and frame-based cameras; the methodology relies on experimental evaluation similarly to ours. The author proposed a similar experiment of a rotating disk with a dot as used in our work. However, the frame-camera used sampled the scene with a relatively low frequency of 20 Hz. The analysis also did not include tracking objects moving at very high speed (e.g., our projectiles). Additionally, the experiments in this work were much less controlled (e.g., no control or precise ground truth for the rotational speed, no varying lighting conditions, etc.). The conclusion of the author, based on experiments, is that as of now, event-cameras are not significantly superior to frame-cameras in tracking application (at least not in general). The author concludes that further development of the event-cameras and tracking algorithms specific for asynchronous events is necessary.

[Cox et al., 2020] proposed to use Johnson's criteria [Johnson, 1958] to compare the automated target recognition performance of event- and frame-cameras. The authors modeled the bandwidth advantage of event-cameras over frame-cameras and assumed that performance is limited by sensor bandwidth. The theoretical analysis further presupposed that noise-free and highly sensitive event-cameras collect the same relevant information as the frame-cameras.

We note that some of these assumptions may not be valid in practice, which motivates us to propose the experimental benchmark and analysis.

[Censi et al., 2015] proposed a power-performance approach to comparing sensor families on a given task. They applied the approach theoretically to the comparison of event-based and periodic sampling in a single-pixel vision sensor. The pixel latency and exposure time are neglected. The overall power consumption is a cost to be minimized and assumed to be linearly proportional to the available sensor bandwidth. The mean square brightness reconstruction error measured performance. The authors found that event-based sampling dominates periodic sampling across all power levels on brightness signals driven by a Brownian process and by sharp switches between two intensity levels ("Poisson" texture). On a very noisy, slowly varying signal, periodic sampling dominates. The two sampling methods perform equally well on a piece-wise linear signal (ramps). On a mixed piece-wise linear and "Poisson" texture signal, periodic sampling is better in the low-power regime, and event-based sampling is better when more power is available.

Event cameras can perform vibration measurement or monitoring [Dorn et al., 2018, Lai et al., 2020]. [Lai et al., 2020] proposed to use a DAVIS240C event-camera in full-field structural monitoring, boundary condition identification and vibration analysis. The event-camera observed the free vibrations of a cantilever beam: the first natural frequency of the beam was approximately 10 Hz and the Photron Fastcam SA5 frame-based high-speed camera provided the ground truth data. The authors found that the advantages of the event-camera compared to the high-speed camera were the high dynamic range, the high equivalent frame rate and the absence of the blur effect.

As of 2021, we did not find any other comparison of frame- and event- cameras beyond the foregoing.

Articles introducing event-camera sensor designs usually test the sensors as well. [Lichtsteiner et al., 2008] measured the pixel transfer function and bandwidth as the mean event response to sinusoidal LED stimulation of 2:1 contrast across a range of frequencies for four DC levels of illumination. They also measured the latency of a pixel response to a positive 30% contrast step at a range of DC illuminance levels. [Posch et al., 2011] presented similar event latency measurements. They also evaluated pixel contrast sensitivity, which is the event probability due to increasing relative contrast at identical initial illuminance. [Lichtsteiner et al., 2008] estimated the standard deviation of the event contrast threshold, which is the only non-ideal behavior simulated by the open-source event-camera simulator ESIM [Rebecq et al., 2018].

[Delbruck et al., 2020] modelled contrast threshold uncertainty, pixel bandwidth, temporal noise, leak events, and hot pixels and used the model to convert videos to events. They noted that, due to the limited pixel bandwidth, a larger brightness step-change triggers a longer series of events, causing "motion blur". Thus, it might not be possible to disambiguate the blur caused by motion from that caused by the finite response time under lower illumination conditions. Based on the pixel latency measured by [Lichtsteiner et al., 2008], they discussed the DVS operation under natural lighting conditions. They recorded that negative (OFF) contrast edges cause lower latency than positive (ON) ones, but they did not quantify the difference. They did not model the finite event readout bandwidth.

Several high-speed applications using event cameras have been published. A pair of event cameras provided position feedback to keep a pencil balanced—the median rate of the feedback

loop was 4 kHz [Conradt et al., 2009]. [Delbruck and Lang, 2013] built a robotic goalkeeper using event cameras, achieving a 550 Hz median update rate. Pacoret et al. [NI et al., 2012] tracked microparticles at a frequency of several kHz by means of an event-based Hough transform. [Howell et al., 2020] applied event cameras to the detection and tracking of high-speed micrometer-sized particles in microfluidic devices.

The survey by [Gallego et al., 2019] mentions several different event-camera application areas such as real-time interaction systems, object tracking, surveillance, object recognition, depth estimation, optical flow, 3D structured light scanning, high dynamic range (HDR) imaging, video compression, visual odometry, and image deblurring. They also list properties of thirteen event-cameras. We used two of them in our experiments.

3.5 Event camera optical flow

Estimating optical flow from event cameras requires different methods than estimating optical flow from traditional image frames. Most current state-of-the-art methods are artificial neural networks trained on simulated data [Luo et al., 2023, Gehrig et al., 2024] or real-world data [Zhu et al., 2018a, Zhu et al., 2019, Gehrig et al., 2021, Paredes-Vallés et al., 2023]. Bflow [Gehrig et al., 2024] is an artificial neural network predicting Bézier curve trajectories instead of pixel displacements. It searches for pixel correspondences in multiple correlation volumes computed from voxel grids representing the events. [Gehrig et al., 2024] trained the network on their own synthetic dataset containing indepedently moving objects. Contrast maximization [Gallego et al., 2018] can estimate motion by transforming a batch of events to a common timestamp such that an objective function (contrast) of the transformed events is maximized. As a result, contrast maximization finds the sharp image of edge patterns (image of warped events (IWE)) which generated the event batch while estimating the scene or camera motion. Recent work used contrast maximization to estimate multi-scale optical flow from raw events by optimization [Shiba et al., 2024]. In addition to estimating optical flow directly, the contrast maximization objective enables unsupervised training of neural networks predicting optical flow [Zhu et al., 2019, Paredes-Vallés et al., 2023, Shiba et al., 2024].

There are two primary datasets commonly used for benchmarking event-based optical flow methods, MVSEC [Zhu et al., 2018b, Zhu et al., 2018a] and DSEC [Gehrig et al., 2021]. MVSEC contains sequences recorded indoors and outdoors using a camera rig held in a hand or mounted on a moving car, motorbike or drone. DSEC recorded sequences with event cameras mounted on a driving car. Both MVSEC and DSEC computed the ground truth optical flow as the motion field given ground truth scene depths from a LiDAR and camera translational and rotational velocities. The motion field ground truth assumed that the scene was static and only the camera moved. Therefore MVSEC does not contain independently moving objects. Although there are independently moving objects in DSEC, its optical flow benchmark does not evaluate the accuracy on them because the ground truth optical flow is not valid there.

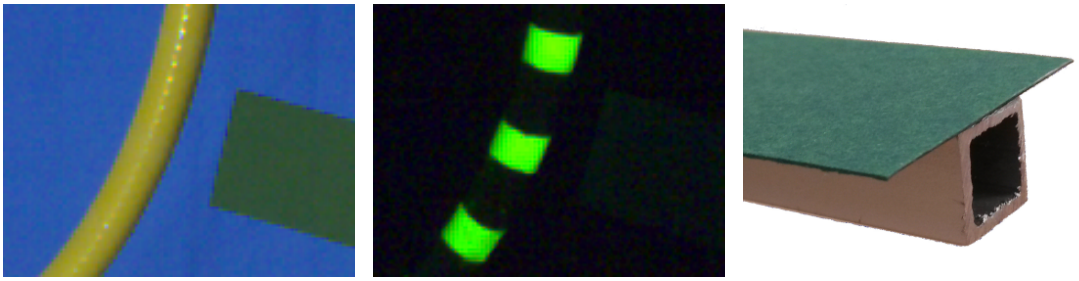
We recorded our own event sequences of moving rods and cables and evaluated the suitability of [Shiba et al., 2024] and [Gehrig et al., 2024] optical flow predictors for cable motion segmentation.

Chapter 4

MovingCables Dataset

Methods that segment moving cables are an essential building block of the eventually integrated action-perception loop. To test or train such methods, we need a suitable dataset. Creating such a dataset is challenging because we need to obtain not only the cable instance segmentation masks but also the cable motion ground truth. We created an automatically annotated moving cable dataset and a novel method able to segment moving cables.

As our robots are too large to manipulate thin cables gently, we recorded video clips featuring a garden hose being manually pushed by a poking stick. We painted UV fluorescent markers on the hose to facilitate ground truth motion estimation. The UV paint is invisible in regular white light but shines clearly in UV light, see Fig. 4.1. Marker tracking automatically estimated the ground truth optical flow and chroma key techniques generated cable and poking stick segmentation masks. Finally, we generated video clips featuring multiple overlapping hoses by compositing several single-hose video clips into one.



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

Figure 4.1: Yellow hose, dark green poking stick, blue backdrop. (**a**) No markers are visible on the hose in white lighting. (**b**) UV lighting shows the UV fluorescent markers and hides everything else. (**c**) Detail of the tip of the poking stick.

Here we present the dataset creation process, the automatic data annotation method, and the nature of the resulting data. We started by recording the video clips of a single hose with a blue screen in the background (Section 4.1). Chroma key segmentation and UV fluorescent marker tracking automatically annotated these images with optical flow and segmentation masks (Section 4.2). Finally, we composited multiple recorded single-hose clips and various background images (Section 4.2) to obtain the final composed dataset consisting of clips showing multiple overlapping hoses (Section 4.3).

4.1 Data recording

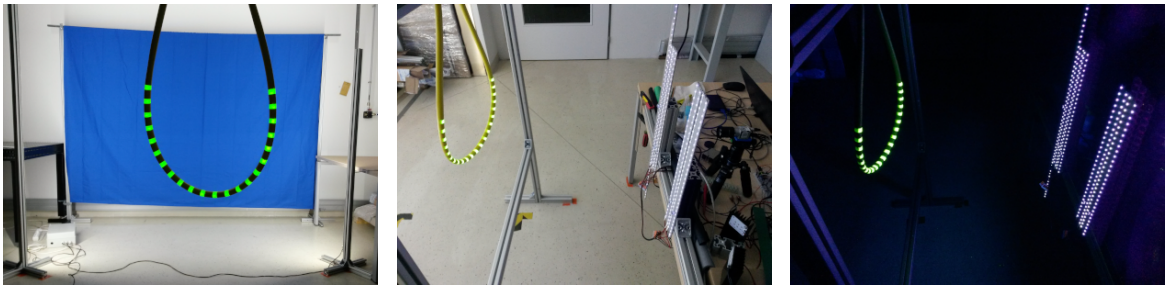
A Basler ace aCA640-750uc camera with a 6mm lens recorded the moving cable scene. A frame standing in front of the camera held the two endpoints of a plain yellow garden hose. We placed a blue screen in the background.

The poking stick, see Fig. 4.1c, was a long thin aluminum bar with dark green cardboard attached to one of its faces. We ensured the cardboard faced the camera when recording to keep the aluminum bar hidden.

The UV fluorescent stripe markers in Fig. 4.1b are cylinder shells painted on a cable in regular intervals with transparent UV fluorescent paint (UV-elements Invisible Glow Lacquer

green[1]).

White LED strips two meters tall lit the background blue screen from the sides, see Fig. 4.2a. Another set of vertical UV LED strips (370 nm wavelength) illuminated the cables, see Fig.s 4.2b, 4.2c. Solid-state relays turned the white and UV LED strips rapidly on and off. White LED strips could also illuminate the cables in the foreground with visible light. Instead, we used high-power white SMD LEDs and a custom LED driver with a digital PWM/enable control input.



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

Figure 4.2: (**a**) Vertical white lights light the blue screen background from the left and right. (**b**) Shining UV light strips with white light turned on. (**c**) Only UV lights turned on.

The camera recorded the scene at 640×480 pixels, 120 FPS. Its digital trigger output signal emitted at the start of every exposure controlled the lights. A UV-lit image followed each white-lit image taken by the camera so that the white-lit image sequence was recorded at 60 FPS. We recorded one video clip per one poking interaction. Each clip is 10 seconds long and contains ca. 1201 images. The raw recorded dataset consists of 177 clips and 212 581 images.

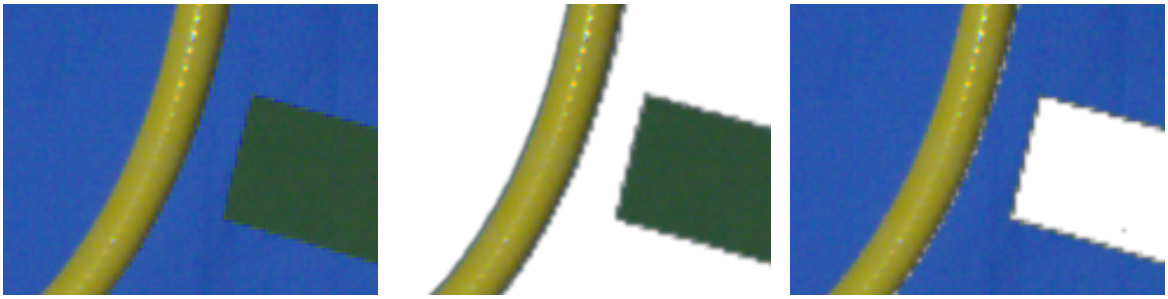
4.2 Post-processing

We post-process the recorded clips in two stages. The first stage performs chroma keying, marker detection, marker tracing, and optical flow ground truth computation. Foreground-background compositing and data augmentation run separately in the second stage.

Chroma keying. We use chroma keying to key the blue screen and the green poking stick, see Fig. 4.3. Chroma keying segments the key color image regions by thresholding the red, green and blue color channels.

Marker detection and tracing. The stripe marker detector detects the UV fluorescent markers in the images of each clip. We ensure during recording that most markers are visible in all the frames of a clip and the poking stick occludes none of them. As the detector does not measure the marker depth, the cables should ideally move in a plane parallel to the image plane. Nearest neighbor marker tracking finds the traces of individual markers. Position

[1]https://www.uv-elements.de/shop/en/Invisible-Glow-Lacquer-50ml-green

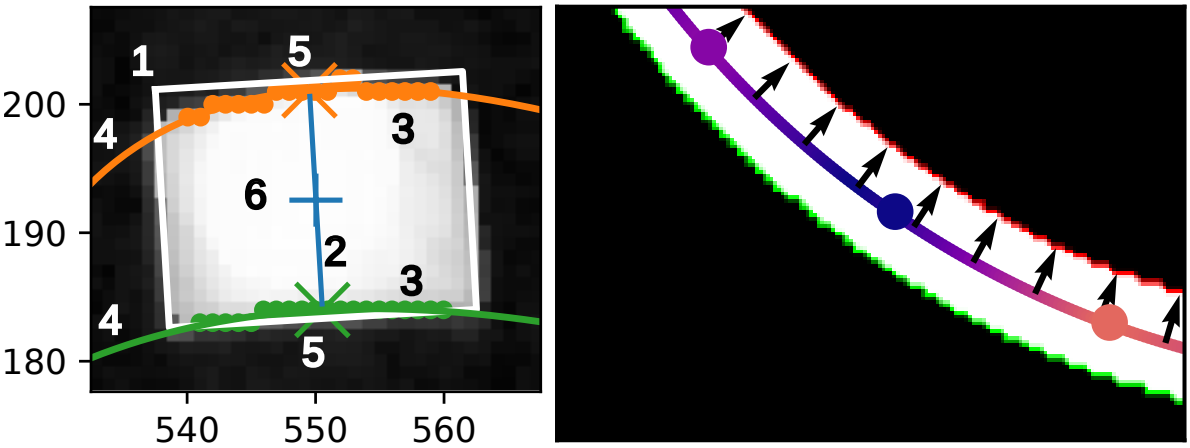


(a) (b) (c)

Figure 4.3: (a) Blue screen, yellow hose, green poking stick. (b) Transparent background. (c) Transparent poking stick.

interpolation of the traces estimates the marker position in the white-lit images. Given the complete marker traces, one can compute marker velocity or displacement for any image pair.

The marker detector extracts individual marker blobs by thresholding a UV-lit image using a fixed intensity threshold. It then locates the center point of the marker blob, see Fig. 4.4a.



(a) (b)

Figure 4.4: (a) Marker center point detection. 1) Fit the minimum area rotated bounding rectangle to the blob. 2) Rectangle (and marker) center line. 3) Scan along lines parallel to the center line. Find the endpoints of the line segments entirely within the blob. 4) Fit a parabola to each set of endpoints. Use orthogonal distance regression (ODR). 5) Intersect each parabola with the center line to estimate the central segment. 6) The central segment center is the marker center. (b) Interpolating optical flow along a cable backbone (middle curve). The cable segmentation mask is white, its contour lines are red and green. The dots represent marker centers, their colors indicate the optical flow magnitude. Black arrows show the unit normal vectors of the backbone spline.

Optical flow ground truth. Optical flow is an independent per-pixel estimate of motion between two images [Szeliski, 2011]. Given the current image I_j sampled at $x_i \in \mathbb{R}^2$ discrete pixel locations, optical flow vectors $\phi_i \in \mathbb{R}^2$ estimate the location of these pixels in a reference image I_1. The optical flow minimizes the brightness or color difference between corresponding pixels summed over all the pixel locations of the current image, $\sum_i [I_1(x_i + \phi_i) - I_j(x_i)]^2$.

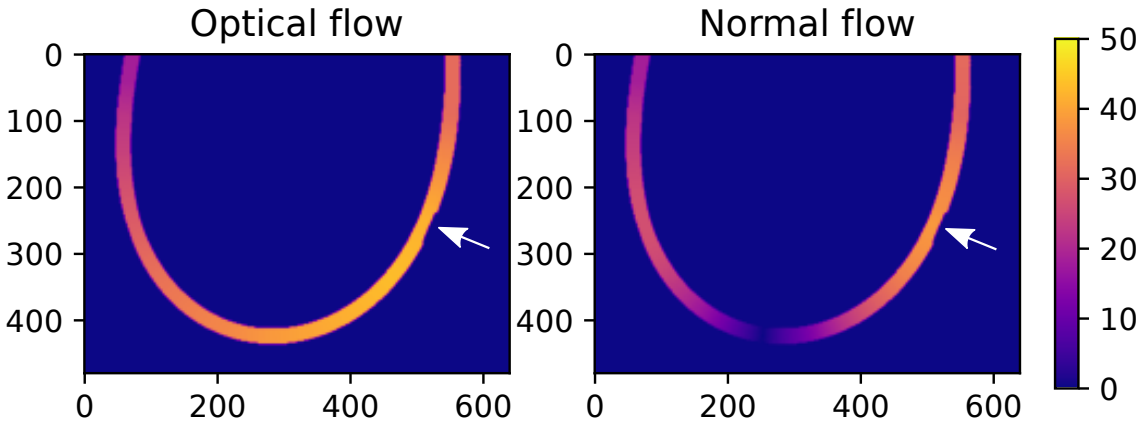


Figure 4.5: Sample ground truth optical and normal flow magnitude in pixels when poking the cable at the right side towards the left as marked by the white arrow.

We provide two types of flow ground truth: full optical flow and "normal flow". Sufficiently textured cables allow full optical flow estimation. "Normal flow" is relevant for textureless cables that only exhibit motion at their boundaries. It is the normal projection of the optical flow vector on the cable boundary normal unit vector. Both ground truths neglect motions caused by a cable rotating around its axis.

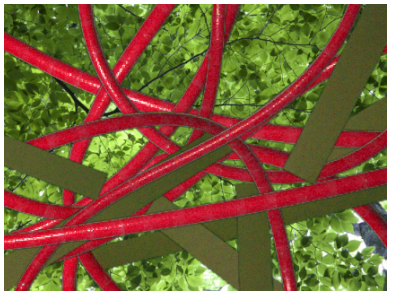
In the recordings, the cable never crosses itself and its endpoints are outside the image. Given a cable segmentation mask (a binary image) and marker traces, interpolation estimates the ground truth flow for each cable pixel.

Thresholding the background-foreground alpha matte yields the foreground mask, and thresholding the poking stick alpha matte yields the poking stick mask. We dilate the poking stick mask by two pixels to ensure that (almost) all poking stick boundary pixels are segmented. The cable segmentation mask is the foreground mask with the poking stick mask pixels removed (set to zero).

The interpolation process illustrated in Fig. 4.4b finds the longest closed contour in the cable segmentation mask, removes its points lying on the image boundary and finds the cable backbone curve by interpolating the two remaining parallel contour lines. Fitting a spline curve to the backbone points estimates the normal vectors for computing the normal flow. Linearly interpolating the displacement of the two markers closest to a backbone point yields its motion. The remaining pixels of the cable segment obtain their flow estimate from their nearest backbone point. See Fig. 4.5 for a sample visualization of the ground truth optical and normal flow magnitude during a poking action.

Compositing and data augmentation. We composite each final clip from a static background image, a moving cable clip and one or more static clips or still images extracted from moving cable clips. We keep both the moving and static poking sticks in the compositons. One can generate a semi-three-dimensional scene of cables stacked on top of each other this way, see Fig. 4.6a.

We manually downloaded CC0-licensed (public domain) background images from the internet. The search was biased towards textures, bushes or woods, and distractors (queries:



(**a**) Ten cables.

(**b**) Sample distractor backgrounds.

(**c**) Sample clutter backgrounds.

Figure 4.6: A sample composition (**a**) and backgrounds (**b**, **c**).

texture, colorful texture, fractal texture, bushes, ropes, wires, pile). We divided the images into two classes: clutter and distractors. Distractors may be confused with hoses, cables, wires, or ropes. Clutter is everything else. See Fig. 4.6.

Even though the backgrounds are often artificial textures or high-quality photographs, we wanted to reduce any JPEG artifacts and remaining sensor noise. Thus we downscaled each background image at least by a factor of two and extracted the center crop 640×480 pixels in size.

Foreground augmentation randomly alters the color of moving and static cables. It can transform hue, contrast, saturation and brightness; invert RGB colors, shuffle RGB channels, or convert to grayscale.

A sensor noise model adds artificial noise to the static background and still cable images to ensure that all image regions exhibit similar noise distributions. If we did not add noise, methods based on temporal image differencing could "segment" the moving cable by assuming that only the moving cable pixels were affected by variable sensor noise.

Sensor noise model. We use sRGBNoise [Kousha et al., 2022], a model originally trained on images taken by five different smartphones [Abdelhamed et al., 2018]. The model generates noise conditioned on a noise-free image, the camera name, and ISO value. We collected a training set with the Basler camera to train its noise model. We treated the (downscaled) background images as the noise-free input to sRGBNoise at inference time. However, the real sensor noise already corrupted the still cable images. Therefore we applied a bilateral filter to suppress the noise before feeding them to sRGBNoise.

4.3 The composed dataset

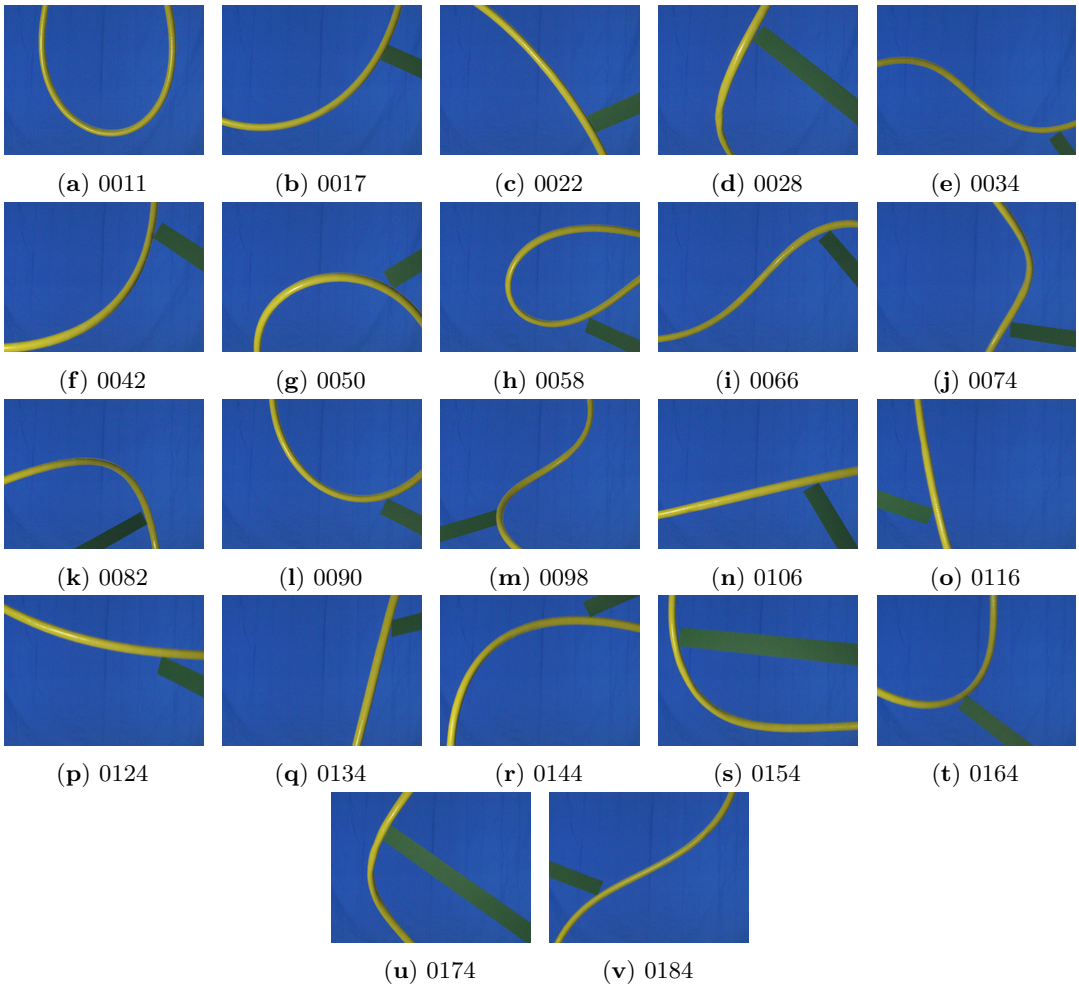


Figure 4.7: The 22 cable configurations of the recorded raw dataset.

We composed the final dataset from the 177 recorded clips (106 200 white-lit images in total). Each recorded clip shows a cable of a single configuration (i.e., a characteristic global shape), see Fig. 4.7, and a single motion class.

Dataset features. Table 4.1 summarizes the main features of the composed dataset. The motion classes are: poking the cable, pushing/pulling an endpoint, endpoint lateral motion, or static (no motion). The cable density relates to the number of cables overlaid in a composition.

Dataset splits. We composed the training, validation, and test dataset splits as follows. First, we divided the recorded clips into three mutually exclusive sets, one for each dataset split. The division satisfies the following constraints: (a) The images of each recorded clip are used in only one split. (b) In each split, each cable configuration is represented by at least one moving cable clip. (c) The number of recorded clips of each motion class in each split is specified in Table 4.2. (d) The cable density classes are represented equally.

Property	Count	Comment
Cable configurations	22	
Cable densities	2	low (4-5 cables), high (10-11 cables)
Motion classes	4	poking, push/pull, lateral, static
Background classes	4	clutter, distractor, plain original, plain transformed color

Table 4.1: The main features of the composed dataset.

Motion class	Recorded	Training	Validation	Test
Poking	104	67	11	26
Push-pull	36	12	8	16
Lateral	16	3	3	10
Static	21	0	0	21

Table 4.2: The division of the recorded clips by motion class into the three splits (training, validation, test).

We used every recorded moving cable clip to create exactly two composed clips, each with a unique background and a unique combination of cable configurations. In a subset of the compositions, we also randomly transformed the colors of the cables or the plain background.

Table 4.3 presents the numbers of images and video clips in each composed dataset split. Each video clip is ten seconds long and consists of ca. 600 white-lit images.

4.4 Conclusions

We have proposed a method to automatically annotate a real-world moving cable segmentation dataset with optical flow and segmentation masks thanks to UV fluorescent markers, controlled lighting, and chroma keying. Using the method, we have created the MovingCables dataset consisting of 312 video clips. The clips differ in their backgrounds, cable colors, numbers of overlaid cables, motion interaction types, or distinct combinations of cable configurations.

As an alternative to a real-world dataset, one could build a synthetic dataset in a simulator. For example, the Blender software can simulate chain-like rope dynamics[2]. It would likely require less manual work as one would not need to design and build any hardware setup. A simulator could simulate a cable in many different positions, such as lying on a desk or hanging freely. However, the cable appearance and the scene lighting would be synthetic. Furthermore, simulating realistic hose or cable dynamics may be more challenging than simulating a chain-like rope. Nevertheless, a synthetic moving cable dataset could complement the real-world dataset presented in this thesis.

Data availability: Code, dataset, and visualizations are available at `https://github.com/holesond/movingcables` and `https://doi.org/10.5281/zenodo.11475246`.

[2]https://blender.stackexchange.com/questions/97749/how-to-simulate-a-rope

	Train	Validation	Test	All
Clips	164	44	104	312
Images	98 399	26 407	62 381	187 187

Table 4.3: The size of the composed dataset and its splits.

Chapter 5

Single Moving Cable Segmentation

The cable motion segmentation methods presented in this chapter assume that either the segmentation mask of the arm moving the cables is available or the arm is not visible in the image. In practice, one can obtain the arm mask using, e.g., the arm CAD model and forward kinematics, model-based rigid object segmentation or pose estimation/tracking [Hodaň et al., 2024], UV fluorescent markers, or color thresholding (our approach).

5.1 MfnProb motion segmentation method

Given a sequence of color images, poking stick segmentation masks, and a motion threshold τ, a motion segmentation algorithm detects cable motion with respect to the first (reference) image I_1 of the clip. The algorithm outputs a motion mask P_m for each image. The pixels p of cable segments in image I_j shifted by more than τ pixels away from their position in the reference image I_1 should be marked as moving in the motion mask, $P_m(p) = 1$. Poking stick pixels and all other pixels p should be marked as static, $P_m(p) = 0$.

We compare five cable motion segmentation methods. The first four of them are baseline methods based on off-the-shelf optical flow predictors, namely MaskFlownet [Zhao et al., 2020], GMFlow [Xu et al., 2023], FlowFormer++ [Shi et al., 2023] and the OpenCV implementation of Farnebäck's optical flow algorithm [Farnebäck, 2003]. To compute the motion segmentation masks, we added optical flow magnitude (L2-norm) thresholding to these methods.

The fifth method is our novel proposed method, MfnProb. To create MfnProb, we added probabilistic outputs [Gast and Roth, 2018] and thresholding to the MaskFlownet deep neural network architecture. Given a pair of noisy input images and trained (certain) network weights, MfnProb predicts noisy optical flow vectors. The probability distribution of a predicted optical flow vector is assumed to be multivariate Laplacian parametrized by location μ and a diagonal covariance matrix Σ, $\sigma^2 = \operatorname{diag}\Sigma$. The network learns to predict the mean $\phi_p = \mu$ and the standard deviation $\sigma_p \in \mathbb{R}^2$ of each optical flow vector probability distribution given a pair of images.

The predicted standard deviation (or variance) has to be non-negative. To ensure that, Gast and Roth [Gast and Roth, 2018] proposed to predict the variance in log space, i.e., $\sigma^2 = \exp(\hat{\sigma}^2)$, where $\hat{\sigma}^2$ was the (log-space) output of the neural network. When we tried to train MfnProb with the exponential, the training diverged. Therefore we replaced the exponential with a softplus function, i.e. $\sigma = \ln(1 + \exp(\hat{\sigma}))$ if $\hat{\sigma} \leq 20$ and $\sigma = \hat{\sigma}$ otherwise, to ensure non-negative standard deviations.

The training loss function of a predicted optical flow vector $\phi_p \in \mathbb{R}^2$ given its ground truth $\phi_{gt} \in \mathbb{R}^2$ is

$$\left(\epsilon + \sum_{i=1}^{2} \frac{(\phi_{p,i} - \phi_{gt,i})^2}{(\sigma_{p,i})^2}\right)^{0.5} + \sum_{i=1}^{2} \ln((\sigma_{p,i})^2), \tag{5.1}$$

which is proportional to the negative log-likelihood of the multivariate Laplacian distribution. We set

$$\sigma_{p,i} = \sigma_{min} + \operatorname{softplus}(\hat{\sigma_{p,i}}). \tag{5.2}$$

The index i runs over the two flow coordinates, horizontal and vertical. $\sigma_{p,i}$ is the standard deviation predicted by the network for the flow coordinate $\phi_{p,i}$. We set $\epsilon = 10^{-8}$ and $\sigma_{min} = 10^{-2}$ to stabilize the training. We trained with the same training schedule on the same optical flow datasets as [Zhao et al., 2020], namely FlyingChairs [Dosovitskiy et al., 2015], FlyingThings3D [Mayer et al., 2016], Sintel [Butler et al., 2012], KITTI [Geiger et al., 2013], HD1K [Kondermann et al., 2015, Kondermann et al., 2016].

In addition to thresholding the optical flow magnitude, MfnProb can utilize the predicted uncertainty to reduce false positives. The segmentation labels any pixel with uncertainty magnitude $\|\sigma_p\|_2 > \sigma_t$ as static. We empirically set the uncertainty threshold σ_t on the validation set to maximize the mean segmentation intersection over union (IoU). In practice, we argue it is safer to predict a static scene when too uncertain because reliable robot's actions, such as grasping, depend on precise true positive segmentation. When a segmentation algorithm has high precision but low recall, the robot can compensate for the low recall by trying multiple different motions until the segmentation succeeds. On the other hand, compensating for low precision is harder.

5.2 Algorithm evaluation process

Given a τ value, a predicted motion mask P_m, and the ground truth optical flow, the evaluation process computes standard segmentation quality metrics, namely the mean intersection over union (IoU), precision, and recall. Our experiments show that increasing the τ threshold above 10 pixels (up to 20) leads to significant decreases in both IoU and recall on the validation set of our dataset. On the other hand, the maximum noise level of the marker detector is 0.528 pixels for static markers. Therefore the evaluation varies τ from 1 to 20 pixels on the validation set and chooses the optimal τ^* value yielding the highest validation IoU. The evaluation reports the test set results given τ^*. In practice, a robot should try to move a cable as little as possible to preserve the cable topology and avoid hitting other cables by accident.

The evaluation also reports the mean endpoint error of the predicted optical flow (EPE) in pixels.

5.3 Evaluation results

Table 5.1 shows the evaluation results of the cable motion segmentation methods on the test set of our dataset. Methods MaskFlownet FT and MfnProb FT are MaskFlownet and MfnProb fine-tuned on a mixture of Sintel, KITTI, HD1K, and the MovingCables training set. We evaluated the methods on the normal flow ground truth as the hoses in the clips have almost no texture, see Fig. 4.6a. The optimal motion threshold τ values on the validation set were 2.5 pixels for MaskFlownet, 2.0 pixels for MfnProb, 1.0 pixel for Farnebäck, 1.0 pixel for GMFlow, and 1.5 pixels for FlowFormer++. The optimal uncertainty threshold of MfnProb was positive infinity, i.e., no high-uncertainty predictions had to be suppressed to maximize the validation IoU.

MfnProb outperforms MaskFlownet in all the evaluation metrics. The probabilistic training scheme reduced the overall mean EPE by almost half. Mean segmentation recall has improved

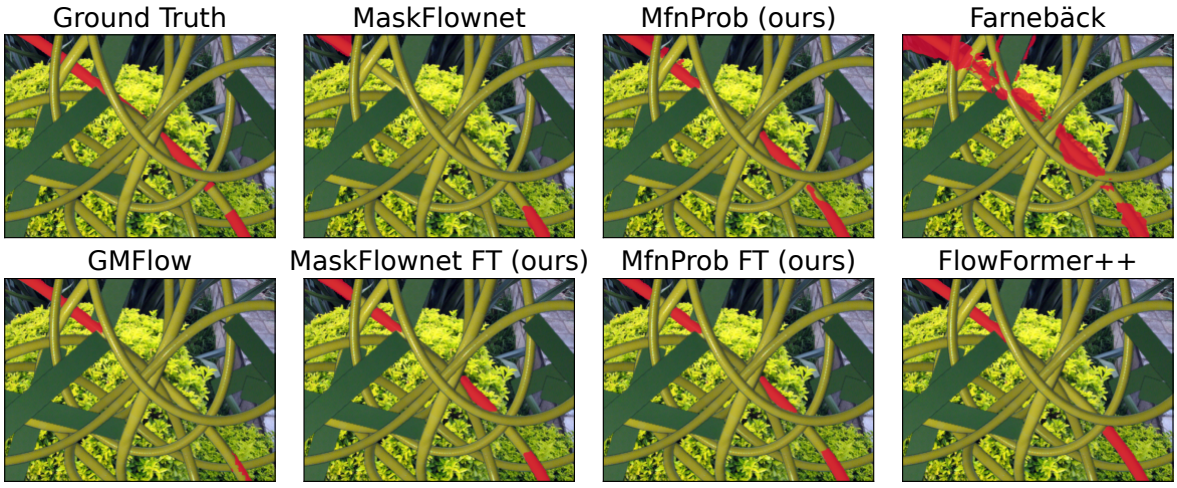


Figure 5.1: Sample motion segmentation. Estimated and ground truth moving segments (at $\tau = 2.5$ pixels) are red.

Method	Recall↑	Precision↑	IoU↑	EPE↓ (pixels)
MaskFlownet [Zhao et al., 2020]	0.6098	0.6415	0.4079	1.22
MfnProb (ours)	0.8768	0.7324	0.6560	0.65
Farnebäck [Farnebäck, 2003]	**0.8977**	0.3520	0.3335	1.56
GMFlow [Xu et al., 2023]	0.7202	0.8077	0.5925	0.45
FlowFormer++ [Shi et al., 2023]	0.7934	**0.8675**	**0.6932**	**0.44**
MaskFlownet FT (ours)	0.8286	0.8295	0.7022	0.37
MfnProb FT (ours)	**0.8762**	**0.8638**	**0.7673**	**0.32**

Table 5.1: Mean evaluation metrics on the test set. IoU – intersection over union, EPE – endpoint error of optical flow.

by 68%, precision by 25%, and IoU by 42% simultaneously. MfnProb outperforms GMFlow in terms of IoU and recall but not precision. FlowFormer++ reaches the highest IoU among the methods not fine-tuned on MovingCables. MfnProb FT achieves the highest IoU overall. Sample segmentations are in Fig. 5.1. Our additional qualitative experiments on real-world videos without chroma keying or compositing indicate that all the motion segmentation methods generalize well to different cable textures (hoses, ropes, cables) and real backgrounds.

Table 5.2 presents the runtime of each algorithm with batch size one. We obtained these results on a desktop computer with Intel Core i9-9900K CPU (3.60GHz) and NVIDIA GeForce RTX 2080 Ti GPU. The original and the probabilistic MaskFlownet networks are similarly computationally intensive, achieving runtimes around 0.040 s per image pair on the GPU and 2.6 s on the CPU. The CPU process times suggest that both networks demand approximately 48× more CPU computation than Farnebäck's algorithm. FlowFormer++ is less suitable for real-time interactive perception than MfnProb as it is 5.7× slower on a GPU.

We further evaluated the methods separately on clips with different background classes (clutter, distractor, plain), see Table 5.3. Clutter and distractor backgrounds yield comparably

Method	GPU wall (seconds)↓	CPU wall (seconds)↓	CPU process (seconds)↓
MaskFlownet [Zhao et al., 2020]	**0.039**	2.6	8.6
MfnProb (ours)	0.041	2.6	8.7
Farnebäck [Farnebäck, 2003]	N/A	**0.056**	**0.18**
GMFlow [Xu et al., 2023]	0.045	1.09	8.0
FlowFormer++ [Shi et al., 2023]	0.232	N/A	N/A

Table 5.2: Mean wall and process runtimes required to compute optical flow for a pair of RGB VGA (640 × 480 pixels) images. The CPU times apply only to algorithms running exclusively on the CPU. CPU process time is equal to user time plus system time or to the CPU wall time multiplied by the mean number of parallel threads used.

Method	Clutter↑	Distractor↑	Plain↑
MaskFlownet [Zhao et al., 2020]	0.4535	0.4739	0.2562
MfnProb (ours)	0.6889	0.7161	0.5322
Farnebäck [Farnebäck, 2003]	0.3503	0.3060	0.3343
GMFlow [Xu et al., 2023]	0.5942	0.5889	0.5927
FlowFormer++ [Shi et al., 2023]	**0.7072**	**0.7396**	**0.6199**
MaskFlownet FT (ours)	0.7367	0.7525	0.5855
MfnProb FT (ours)	**0.7803**	**0.7942**	**0.7149**

Table 5.3: Mean IoUs on the test set separately for three background types, clutter, distractor, and plain.

accurate segmentations. Plain backgrounds, however, tend to cause significantly more false positive segmentations by the neural networks in static areas, resulting in lower mean IoU. Replacing poorly textured plain backgrounds with texture-free solid colors completely confuses the neural networks, see Table 5.4. They falsely predict motion in almost the entire image. Fine-tuning MaskFlownet or MfnProb on MovingCables brings negligible improvements. By contrast, plain backgrounds do not affect Farnebäck's performance significantly. We think that the neural networks do not regularize towards the smallest flow at a pixel where many flow vectors have very similar matching costs.

5.4 Conclusions

We have tested MaskFlownet, GMFLow, and FlowFormer++ off-the-shelf optical flow neural networks on our dataset and found that they can segment moving cables from a static background. We added uncertainty outputs to the MaskFlownet architecture and retrained it with a probabilistic loss function on standard optical flow datasets. This retrained MfnProb network has significantly improved the cable motion segmentation performance over MaskFlownet on our dataset. Fine-tuning MaskFlownet and MfnProb on MovingCables further improved the accuracy. Nevertheless, we believe that optical flow estimators should work reliably on any realistic visual input without fine-tuning.

Method	Min.↑	Median↑	Mean↑	Max.↑
MaskFlownet [Zhao et al., 2020]	0.0378	0.0433	0.0451	0.0531
MfnProb (ours)	0.0379	0.0443	0.0455	0.0530
Farnebäck [Farnebäck, 2003]	**0.3977**	**0.4545**	**0.4508**	**0.4900**
GMFlow [Xu et al., 2023]	0.1033	0.1189	0.1193	0.1397
FlowFormer++ [Shi et al., 2023]	0.0573	0.0645	0.0667	0.0837
MaskFlownet FT (ours)	0.0567	0.0629	0.0631	0.0689
MfnProb FT (ours)	0.0526	0.0575	0.0605	0.0981

Table 5.4: Statistics of per-clip mean IoUs on 20 clips with various solid background colors. We kept one low density cable composition fixed for all the clips.

Limitations: We have found that all the neural networks struggle with texture-free backgrounds. Furthermore, manipulating a cable in a cluttered environment can perturb neighboring cables, causing multiple moving cables. As our methods segment motion by thresholding the flow magnitude, they segment multiple moving cables as a single cable. We address this limitation in Chapter 6.

Data availability: Code, dataset, and visualizations are available at `https://github.com/holesond/movingcables` and `https://doi.org/10.5281/zenodo.11475246`.

Chapter 6

Interactive Robotic Moving Cable Segmentation by Motion Correlation

Chapters 4 and 5 (also published in [Holešovský et al., 2024]) presented a new moving cable dataset automatically annotated with optical flow and instance segmentation masks. They also proposed several cable motion segmentation methods which assumed that only a single cable was moving in the scene and that the segmentation masks of the arm moving the cables were available.

The method presented in this chapter uses a physical robot to slightly move the grasped cable in different directions. It segments the cable by correlating the robot gripper motion with the cable motion measured by an optical flow estimator. The method uses multiple robot actions to segment the cable of interest in the last frame of a recorded image sequence. Unlike [Holešovský et al., 2024], it can segment the grasped cable from neighboring cables moving due to perturbations. We also reformulate the interactive cable segmentation problem to avoid the need for the arm segmentation masks. We are not aware of any other method addressing the same problem of interactive cable segmentation.

6.1 Methods

A robot typically performs multiple actions on a cable while exploring it and records a data sequence of color images and gripper positions. It divides the data sequence by action, such that the images $\{I_{a,k}\}$ and the gripper positions $\{g_{a,k}\}$ are indexed by the action index $a \in \{1, ..., A\}$ and the within-action sample index $k \in \{1, ..., N_a\}$, where A is the total number of actions in the data sequence and N_a is the number of data samples of action a. To ensure spatially uniform sampling, we subsample the images and gripper positions such that $||g_{a,k} - g_{a,k+1}||_2 \geq \delta_g$, where δ_g is the minimum gripper sample displacement.

Both segmentation methods segment moving cables by computing optical flow vectors $\phi_{a,k}(p) \in \mathbb{R}^2$ for all discrete image pixel locations $p \in \mathbb{R}^2$ between each image $I_{a,k}$ and the target image I_{A,N_A}, which is the last image of the last action of the recorded data sequence. The motion segmentation/correlation methods express all the flow vectors in the target image frame, such that ideally $I_{a,k}(p + \phi_{a,k}(p)) \approx I_{A,N_A}(p)$ holds for all non-occluded pixels p, where $I_{a,k}(p)$ is the image color or brightness sampled at pixel p. The gripper/arm is outside of the field of view in the target frame. The arm therefore does not occlude any cables in the target image and causes little additional optical flow consistent with the arm motion which would require arm masking.

An action with index a is a robot periodically moving a cable initially grasped at position $g_{a,1}$ between the positions $(g_{a,1} + \Delta_a)$, $g_{a,1}$, $(g_{a,1} - \Delta_a)$, and back to $g_{a,1}$. Δ_a is a constant action-specific vector that defines the direction and amplitude of the movement, 1 is the index of the first (reference) image and of the first gripper position of action a. The robot repeats the periodic motion K_p times in each action. We let the robot perform two actions for each initial grasp point $g_{a,1}$. One action moves the gripper along the cable axis projected to the camera image plane, the other perpendicularly to the cable axis. We note, however, that the robot could easily execute a more diverse set of actions specified by different action vectors Δ_a.

The segmentation methods assume that an initial cable segment to grasp is given for each cable instance which should be explored and segmented. In our experiments with the physical robot, a human provided the initial grasp point by drawing the short cable segment to grasp

into the camera image frame on a computer screen. Alternatively, passive semantic cable segmentation methods or known cable endpoint locations (connectors, sockets) could provide the initial grasp point.

To enable the robot to grasp a selected cable segment, the methods need to estimate the 3D cable center and axis given the 2D cable segment in an RGB-D image. The methods compute the segment center as the geometric median of the cable segment 3D points. Assuming that the segment is longer than it is wide, the cable axis is the principal axis computed by PCA from the set of cable segment 3D points within an inlier distance from the segment center.

Once an interactive cable segmentation method (partially) segments the initially grasped cable by moving it, a grasp sampling algorithm can propose the next suitable cable segment to grasp and move to increase the recall and/or precision of the overall interactive segmentation process. We describe our grasp sampling algorithm in Section 6.1.3.

We describe two interactive cable segmentation methods in this section. One is the *motion segmentation* baseline, an adaptation of MfnProb FT from [Holešovský et al., 2024]. The other is the novel *motion correlation* method we propose.

6.1.1 Motion Segmentation Baseline Method

The *motion segmentation* baseline method is MfnProb FT moving cable segmentation method from [Holešovský et al., 2024]. The baseline assumes that there is only one cable moving in the scene and that any perturbations of neighboring cables are negligibly small. It estimates moving cable segmentations in each image of a sequence. We adapted it in this work, such that it segments cables only in the target frame based on the motion detected at image $I_{a,k}$ using the optical flow magnitude

$$||\phi_{a,k}(p) - \phi_{a,1}(p)||_2 > \tau_m, \tag{6.1}$$

where τ_m is a motion threshold. Each pixel p where the flow magnitude is larger than τ_m adds one vote into the vote image $V_a(p)$. The algorithm outputs a motion mask M_a for each action a. It reports only those pixels p as moving, i.e. $M_a(p) = 1$, where $V_a(p) > \tau_v N_a$, where τ_v is a relative vote threshold ($0 \leq \tau_v \leq 1$) and N_a is the total number of input images participating in action a voting. It marks other pixels p as static, i.e. $M_a(p) = 0$. We can say that action a labels as moving or votes for only those pixels p where $M_a(p) = 1$.

6.1.2 Motion Correlation

To segment a grasped cable from other cables, we propose the *motion correlation* algorithm that leverages the predictable movement of the grasped cable when manipulated in multiple directions. Neighboring cables can be perturbed only if they are in contact with the grasped cable or the robot. For loosely entangled cables, repeated gripper motions in different directions typically move the grasped cable predictably while occasionally disturbing the neighboring cables. Our proposed algorithm exploits this predictability to effectively segment the grasped moving cable from its surroundings. See Figure 6.1 for a visual summary of the method.

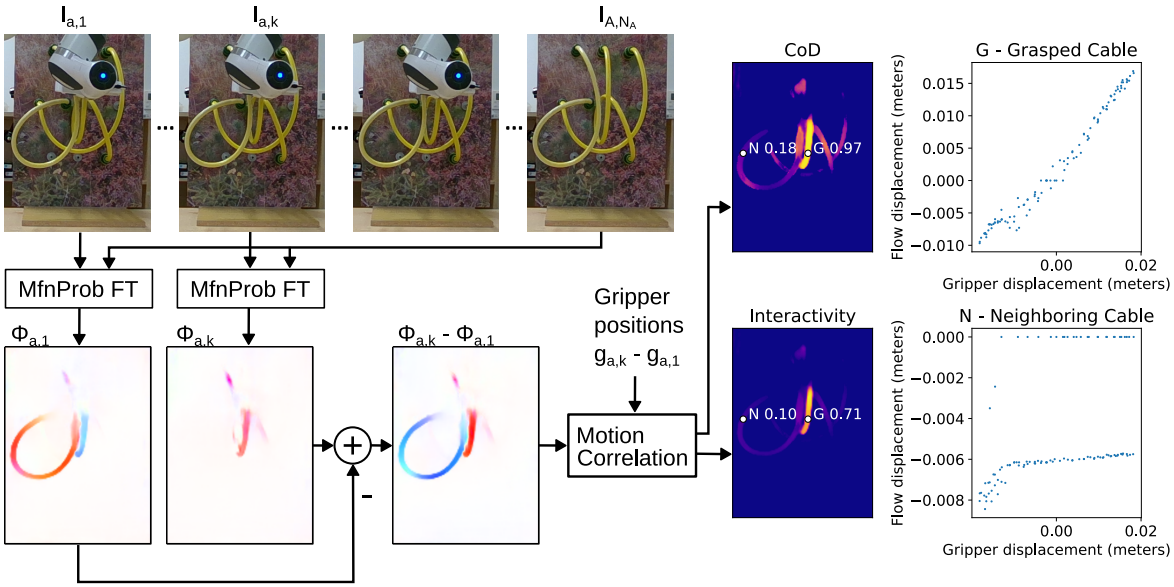


Figure 6.1: An overview of the proposed *motion correlation* method. MfnProb FT [Holešovský et al., 2024] estimates the optical flow.

For each sample (a, k), the algorithm computes the gripper displacement along the motion direction,

$$d_{g,a,k} = (g_{a,k} - g_{a,1}) \cdot \frac{\Delta_a}{||\Delta_a||_2}. \tag{6.2}$$

In each image $I_{a,k}$, the algorithm processes only the pixels p where $||\phi_{a,k}(p) - \phi_{a,1}(p)||_2 > \tau_n$, where τ_n is a noise suppression threshold. For each such pixel, it computes the optical flow displacement in meters along the principle flow direction $\psi_a(p)$ (unit vector) during action a as follows

$$d_{\phi,a,k}(p) = \frac{Z_{a,k}(p)}{f}(\phi_{a,k}(p) - \phi_{a,1}(p)) \cdot \psi_a(p), \tag{6.3}$$

where f is the camera focal length in pixels and $Z_{a,k}(p)$ is the depth of pixel p in image $I_{a,k}$ in meters. The equation assumes that the image has already been undistorted and that the camera pixel shape is (almost) square. (One could generalize to rectangular pixels by scaling the horizontal flow coordinates by f_x and the vertical ones by f_y to compute the 2D metric flow.)

The method estimates $\psi_a(p)$ from the relative flow vectors $(\phi_{a,k}(p) - \phi_{a,1}(p))$ using online PCA with online flow covariance computation.

Fitting a linear model h_a to predict flow displacement $d_{\phi,a,k}(p)$ given gripper displacement $d_{g,a,k}$,

$$d_{\phi,a,k}(p) \approx h_a(p, d_{g,a,k}) = c_{a,0}(p) d_{g,a,k} + c_{a,1}(p), \tag{6.4}$$

and computing the coefficient of determination (CoD) $R^2_a(p)$ is the last step of the algorithm. The algorithm computes the linear model parameters $c_{a,0}(p)$, $c_{a,1}(p)$ and $R^2_a(p)$ online from

the first- and second-order moments of $d_{\phi,a,k}(p)$ and $d_{g,a,k}$ to reduce its memory requirements. Pixels p where $|c_{a,0}(p)| > c_{0,min}$ (interactivity threshold) and $R_a^2(p) > R_{min}^2$ (CoD threshold) are labeled as moving, i.e. $M_a(p) = 1$, other pixels p as static, i.e. $M_a(p) = 0$. The first condition ensures that only sufficiently interacting cable segments are labeled as moving. The second condition suppresses the areas where the relationship between the gripper and flow displacements significantly differs from a linear function.

6.1.3 Grasp Sampling

We further propose a grasp sampling algorithm to enhance the recall and/or precision of the overall interactive segmentation process by identifying the next suitable cable segment to grasp and move, given a partial cable segmentation.

The proposed algorithm operates based on a set of defined sampling priorities and requirements, structured as follows:

1. First, prefer segments supported by the highest number of different actions to reduce the possibility of grasping a cable different from the one which should be explored.
2. Second, sample among segments sufficiently far from the previously grasped segments. This priority improves cable exploration efficiency.

Furthermore, each cable segment sampled by the algorithm meets several requirements:

1. A sampled grasp segment has a minimum and maximum acceptable size (image area) and a minimum acceptable eccentricity. This reduces the possibility of sampling too small noise segments or segments which are not elongated enough and thus do not resemble cables. The maximum area threshold ensures sufficiently dense sampling of the segmentation image.
2. The estimation of the 3D cable segment axis is possible using the sampled 2D segment and the depth image, i.e. there is a sufficient number of depth measurements available in the sampled cable segment area.
3. The angle between the cable segment axis and the image plane is sufficiently low. Pose and depth estimation of cable segment axes which are close to normal to the image plane is less accurate due to a lower number of pixel samples per a metric unit of cable length.
4. Do not sample (almost) the same grasp segment multiple times. Successfully sampled segments meeting all the requirements are blacklisted from further sampling.

For example, the sampler first samples all segments supported by at least two actions and only those segments which are at least G_{max} meters distant from previous grasps. If there is no such sample, the sampler gradually reduces the minimum grasp distance threshold by a G_Δ step. If there is no sample even with the minimum distance threshold of G_{min}, the sampler samples from all segments supported by at least one action which are at least G_{max} meters distant from previous grasps and so on. G_{max} should approximately correspond

to the workspace size, G_{min} sets the lower bound on the cable exploration efficiency. The G_Δ parameter sets the trade-off between sampling precision and runtime. The lower the G_Δ, the higher the precision and the runtime and vice versa.

Finally, a motion planner tries to plan a robot end effector trajectory to grasp each sampled segment. If the planning fails, e.g. due to pose unreachability or robot (self) collisions, the grasp sampler samples another grasp.

6.2 Implementation

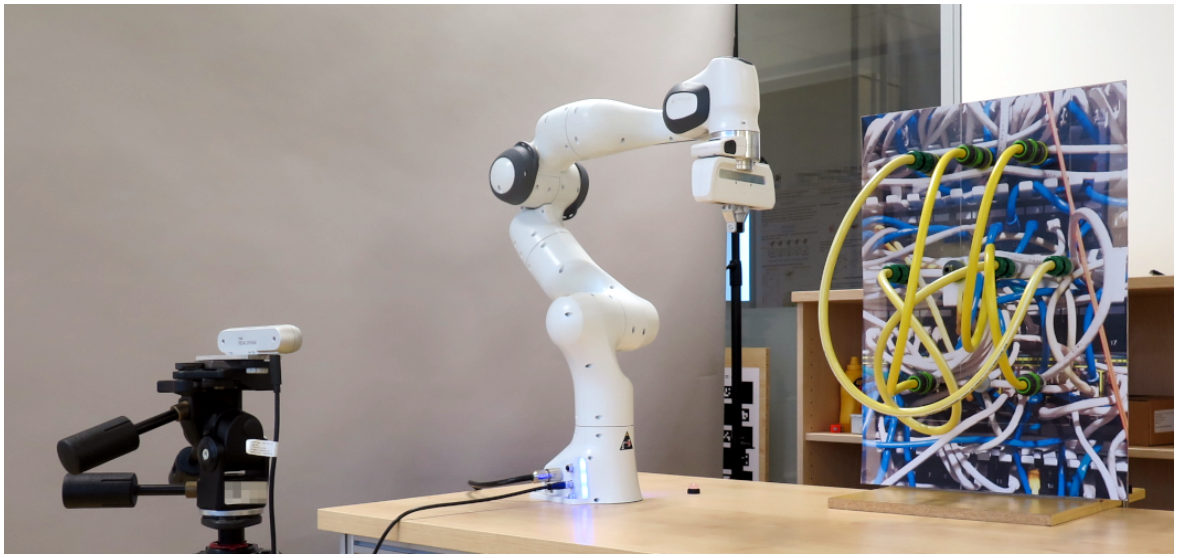


Figure 6.2: The robotic setup. An Intel RealSense D456 RGB-D camera is on the left, a Franka Emika Panda robotic arm is in the middle, the task board with tangled hoses is on the right.

We performed real-world cable manipulation experiments using a Franka Emika Panda robot controlled via MoveIt2 in ROS2. An Intel RealSense D456 RGB-D camera watched the robot and the task board mounted next to it, see Fig. 6.2. Color and depth image resolution was 1280×720 pixels at 15 FPS framerate. The 15.4 mm thick cables (hoses) were at the depth of ca. one meter. As the cable depth variability relative to the mean cable depth was low in our experiments, we used the gripper depth in the camera coordinate frame in place of $Z_{a,k}(p)$ for simplicity. We attached at least one endpoint of each hose to the task board using a garden hose connector. To obtain variously complex scene backgrounds, we mounted replaceable cardboard sheets behind the garden hose connectors. We printed a different background image on each sheet.

We used M3T multi-body tracker [Stoiber et al., 2022] for camera-robot calibration. A static robot detector (a constant transformation matrix) or a manual detector (a human selecting four keypoints on the robot arm in a color image) approximately initialized the tracker state when the Franka Emika Panda robot was in the "ready" pose. M3T then iteratively refined the camera-robot pose estimate by tracking the robot arm in the RGB-D images.

MfnProb FT [Holešovský et al., 2024] deep neural network estimated the optical flow required by the *motion correlation* and *motion segmentation* methods. It processed color images cropped to the task board area (272×368 pixels in size). We scaled up the images $1.5\times$ because MfnProb tends to ignore the motion of cables thinner than ca. ten pixels. To compute the optical flow in parallel with the motion correlation algorithm, we implemented the computation in a two-stage pipeline. The first stage computed the optical flow on a GPU (NVIDIA GeForce RTX 2080 Ti), the second stage ran the motion correlation implemented in Numpy on a CPU (Intel Core i9-9900K CPU (3.60GHz)).

MoveIt2 built an octomap from scene point clouds to avoid collisions when reaching for the grasps.

6.3 Experiments

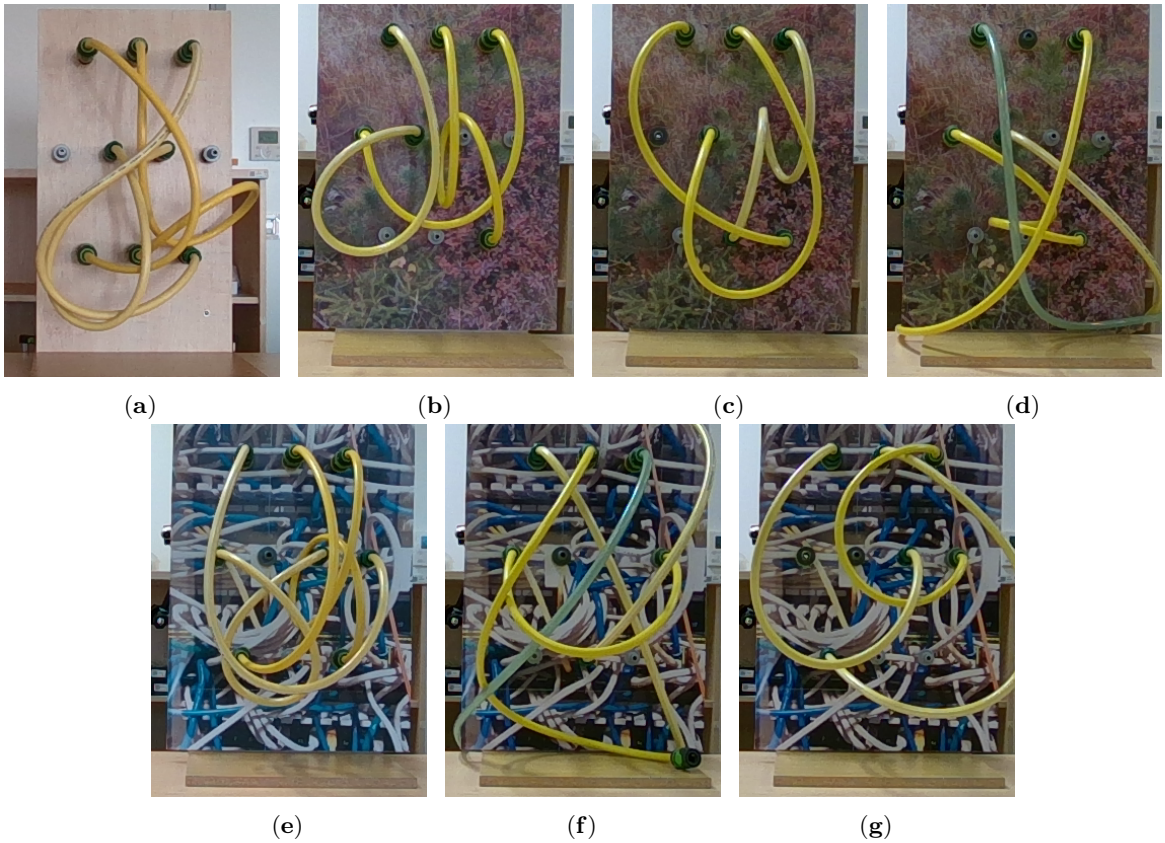


Figure 6.3: The seven cable configurations of the recorded dataset.

We evaluated the cable segmentation methods on a dataset of 50 sequences recorded with our robotic setup. We manually annotated the grasped cable ground truth segmentation in the target image of each dataset sequence. Each sequence consists of two actions performed at one grasp point. Table 6.1 summarizes the main features of the recorded dataset and Figure 6.3 shows its seven cable configurations.

High cable segmentation precision with low recall is usually more valuable in a robotic cable

Property	Count	Comment
Cable configurations	7	
Cables explored	21	
Backgrounds	3	plain wood, bushes, network cables
Validation sequences	8	2 977 images
Test sequences	42	16 088 images

Table 6.1: The main features of the dataset.

Parameter name	Symbol	Value
Minimum gripper sample displacement	δ_g	0.001 meters
Action amplitude	$\|\|\Delta_a\|\|_2$	0.02 meters
Number of motion periods	K_p	3
Motion threshold [Holešovský et al., 2024]	τ_m	2.0 pixels
Noise suppression threshold	τ_n	0.67 pixels
Relative vote threshold	τ_v	see Table 6.3
Interactivity threshold	$c_{0,min}$	see Table 6.3
Coefficient of determination threshold	R^2_{min}	see Table 6.3
Furthest grasp sampling frontier	G_{max}	0.2 meters
Grasp sampling frontier step	G_Δ	0.05 meters
Nearest grasp sampling frontier	G_{min}	0.1 meters

Table 6.2: Parameter values used in our experiments.

exploration scenario than high recall with low precision [Holešovský et al., 2024]. Precision typically determines manipulation success, e.g. grasping the correct cable, whereas recall affects task efficiency, e.g. the number of actions required to explore the cable. Therefore we prefer the F_β score over the intersection over union (IoU) as the main measure of cable segmentation performance. We use F_β with $\beta < 1$ to increase the importance of precision over recall. The lower the β, the higher is the weight of precision in the F_β computation.

Table 6.2 lists the method parameter values we used. We chose the threshold values ($c_{0,min}$, R^2_{min}, τ_v) to maximize the F_β score on the validation set for several different β values, see Table 6.3. The predicted segmentation of each sequence was the union of the segmentations from both actions at a single grasp point. All three thresholds tend to grow with decreasing β to increase the segmentation precision.

Table 6.4 presents the test set evaluation results for each β when the methods output for each sequence the union of the segmentations predicted for both actions. Table 6.5 shows similar test set results when the methods output the intersection of the segmentations. In both cases, *motion correlation* outperforms the *motion segmentation* baseline in terms of F_β and precision. On the other hand, the baseline achieves higher recall.

The "No gripper ablation" method in Tables 6.4, 6.5 is an ablation of the *motion correlation* method. It segments moving cables by thresholding the standard deviation of the relative optical flow displacement $d_{\phi,a,k}(p)$ while ignoring the gripper motion information. Its precision and F_β scores are worse than the ones of *motion correlation*, suggesting that the gripper

β	τ_v	$c_{0,min}$	R^2_{min}
1.0	0.31	0.17	0.66
0.5	0.47	0.20	0.77
0.4	0.47	0.18	0.86
0.3	0.47	0.36	0.92

Table 6.3: Parameters optimizing F_β score on the validation set for different β values.

Method	β	Recall↑	Precision↑	F_β↑	IoU↑
Motion segmentation	1.0	**0.6791**	0.3642	0.4205	0.2868
No gripper ablation	1.0	0.6065	0.4407	0.4402	0.3047
Motion correlation	1.0	0.5911	**0.4758**	**0.4971**	**0.3488**
Motion segmentation	0.5	**0.5842**	0.4153	0.4100	0.2912
No gripper ablation	0.5	0.5376	0.4800	0.4471	0.2939
Motion correlation	0.5	0.5123	**0.6117**	**0.5660**	**0.3771**
Motion segmentation	0.4	**0.5842**	0.4153	0.4102	0.2912
No gripper ablation	0.4	0.5154	0.4869	0.4566	0.2862
Motion correlation	0.4	0.4618	**0.7488**	**0.6533**	**0.3914**
Motion segmentation	0.3	**0.5842**	0.4153	0.4115	0.2912
No gripper ablation	0.3	0.5038	0.4894	0.4682	0.2818
Motion correlation	0.3	0.3430	**0.9001**	**0.7505**	**0.3281**

Table 6.4: Mean evaluation metrics on the test set of methods computing the union of segmentations from two actions of each sequence. The best results are in bold.

motion information facilitates precise moving cable segmentation.

Fig. 6.4 qualitatively compares sample segmentations computed by the two cable segmentation methods with the parameters optimizing $F_{0.4}$ on the validation set. The *motion segmentation* baseline tends to segment multiple moving cables as the grasped cable, see esp. Fig. 6.4a, 6.4b. It also sometimes wrongly segments the robot arm, which is present in most sequence images but absent in the target image (Fig. 6.4a, 6.4c). In contrast, *motion correlation* produces cable segments mostly correctly marking the grasped cable, especially when considering the intersection of the segmentations from both actions (Fig. 6.4a, two votes). Both methods can segment cables partially extending beyond the image boundary (Fig. 6.4c).

Table 6.6 compares the test set evaluation results of the *motion correlation* method and of several state-of-the-art passive segmentation methods. We used the 2D grasp point in each target image to prompt SAM 2 or to retrieve the cable instance segmentation mask closest to the grasp point in the case of FASTDLO, mBEST, RT-DLO. *Motion correlation* (optimizing $F_{0.4}$) outperforms the passive segmentation methods in terms of $F_{0.4}$ and precision. SAM 2 reaches the highest recall as it usually segments multiple or all the cables visible in the image instead of only the grasped cable. As the passive segmentation methods processed the arm-free target images, they could achieve higher recall than *motion correlation*. *Motion correlation* processes images of cables partially occluded by the robot arm, which reduces its

Method	β	Recall↑	Precision↑	F_β↑	IoU↑
Motion segmentation	1.0	**0.3512**	0.3921	0.2727	0.1714
No gripper ablation	1.0	0.2849	0.4975	0.2387	0.1361
Motion correlation	1.0	0.2343	**0.7171**	**0.3259**	**0.2068**
Motion segmentation	0.5	**0.2435**	0.3658	0.2341	0.1274
No gripper ablation	0.5	0.2100	0.4932	0.2218	0.0957
Motion correlation	0.5	0.1860	**0.8193**	**0.4310**	**0.1766**
Motion segmentation	0.4	**0.2435**	0.3658	0.2468	0.1274
No gripper ablation	0.4	0.1922	0.4894	0.2299	0.0866
Motion correlation	0.4	0.1599	**0.9449**	**0.4711**	**0.1576**
Motion segmentation	0.3	**0.2435**	0.3658	0.2644	**0.1274**
No gripper ablation	0.3	0.1837	0.4868	0.2646	0.0825
Motion correlation	0.3	0.0501	**0.9956**	**0.2916**	0.0501

Table 6.5: Mean evaluation metrics on the test set of methods computing the intersection of segmentations from two actions of each sequence. The best results are in bold.

Method	Recall↑	Precision↑	$F_{0.4}$↑	IoU↑
SAM 2 Image (L) [Ravi et al., 2024]	**0.8826**	0.4898	0.4941	**0.4058**
FASTDLO [Caporali et al., 2022a]	0.4348	0.6924	0.5791	0.3514
mBEST [Choi et al., 2023]	0.4267	0.6979	0.5431	0.3285
RT-DLO [Caporali et al., 2023b]	0.2828	0.4275	0.3011	0.2050
Motion correlation ∪	0.4618	0.7488	**0.6533**	0.3914
Motion correlation ∩	0.1599	**0.9449**	0.4711	0.1576

Table 6.6: Mean evaluation metrics on the test set of *motion correlation* and passive segmentation methods. *Motion correlation* either computes the union (∪) or the intersection (∩) of segmentations from two actions of each sequence. The best results are in bold.

maximum achievable recall and handicaps it in this comparison.

We report the mean runtime per processed image for each method and for each stage of the two-stage *motion correlation* pipeline in Table 6.7. Both methods are similarly fast and are suitable for real-time inference. We note, however, that they currently require the last (target) image of a sequence before they can start computing the optical flow. Despite running on the GPU, the optical flow estimation stage is slower than the motion correlation stage utilizing just a single CPU core.

Table 6.8 reports the success rates of the grasps selected from *motion correlation* predictions by our grasp sampling algorithm on three cable configurations. Most of the proposed grasps (96%) were on the correct cable. The robot could not reach most of the proposed grasps (59%) due to its kinematic constraints or due to other cables obstructing the way to the proposed grasp pose.

Table 6.9 presents *motion correlation* evaluation results on ten cables when segmenting a cable given one and two grasp points. We provided the first grasp point manually. The grasp sampling method automatically proposed the second grasp point for each cable. In this

Method	Runtime (seconds per image)↓
Motion segmentation baseline	0.047
Motion correlation	0.047
Stage 1: Optical flow (GPU)	0.043
Stage 2: Motion correlation (CPU)	0.039

Table 6.7: Mean method runtimes.

Grasp proposal category	Count	% of total
Total grasp proposals	54	100
Correct & reachable	20	37
Correct & unreachable	32	59
Wrong, another cable	2	4
Wrong, background	0	0

Table 6.8: Quantitative evaluation of automatically proposed grasps.

experiment, a positive segmentation from each action contributed a unit vote to a multi-grasp vote image. The *motion correlation* method used the parameters optimizing $F_{0.4}$ on the validation set. Table 6.9 reports the segmentation accuracy computed at all possible action thresholds, i.e. the numbers of actions supporting each segmented pixel. Two grasps yield higher segmentation recall than a single grasp. At the two-action threshold, the segmentation from two grasps has slightly lower precision than the single grasp segmentation but its $F_{0.4}$ score is higher. Precision increases with more actions supporting a segmented cable pixel. Figure 6.5 shows sample cable segmentation given one and two grasp points.

6.4 Discussion and Conclusions

We have proposed a *motion correlation* method which is able to segment a grasped moving cable even when the robot or the cable perturbs neighboring cables. It exploits the observation that gripper motion tends to correlate with grasped cable motion estimated in a common image frame by an optical flow predictor.

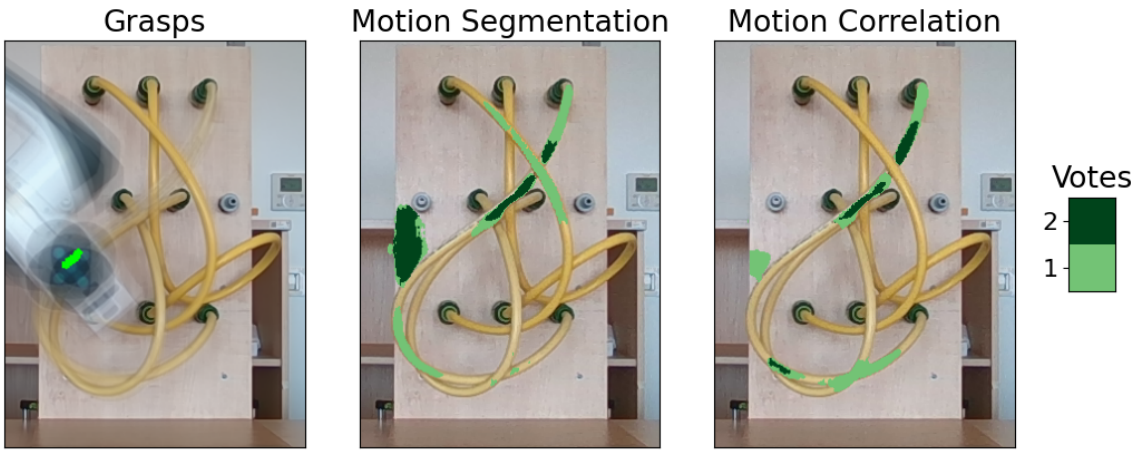
We have tested the *motion correlation* method and the *motion segmentation* baseline on data recorded with our physical robotic setup. Both quantitative and qualitative results indicate that *motion correlation* outperforms the baseline.

We have also proposed an algorithm for sampling new grasps given a partial cable segmentation. Thanks to the grasp sampler, the robot can interact with the explored cable at multiple grasp points. Our evaluation results showed that merging segmentation results from two grasps increases cable segmentation recall while preserving precision. At the same time, we have found that the robot could not reach the majority of the proposed grasps. We attribute this to the fact that the Franka Emika Panda robot we used is fairly bulky, especially its gripper is quite wide. We think that a robot with a thinner body and gripper would be more suitable for exploring cables than the Franka Emika Panda.

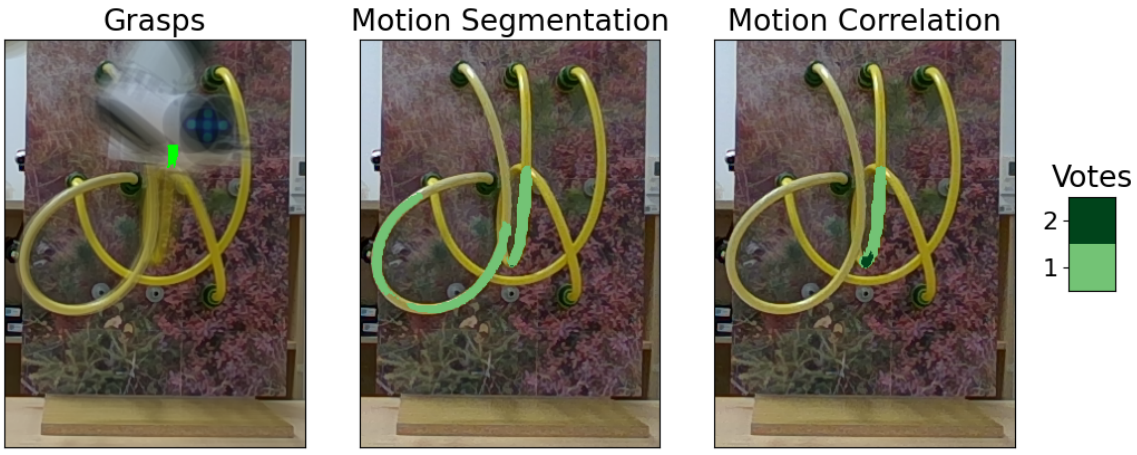
Grasps @ action thr.	Recall↑	Precision↑	$F_{0.4}$↑	IoU↑
One grasp @ 1	0.5570	**0.7882**	**0.7317**	0.4756
Two grasps @ 1	**0.6658**	0.7111	0.6934	**0.5179**
One grasp @ 2	0.2404	**0.9092**	0.6252	0.2378
Two grasps @ 2	**0.3987**	0.8838	**0.7218**	**0.3812**
Two grasps @ 3	0.1072	0.9193	0.3673	0.1065
Two grasps @ 4	0.0329	0.9797	0.1507	0.0329

Table 6.9: *Motion correlation* results with one and two grasps. A grasp consists of two actions. The number of actions supporting a segmented pixel acts as a high-level confidence threshold. The better results for each action threshold (action thr.) are in bold.

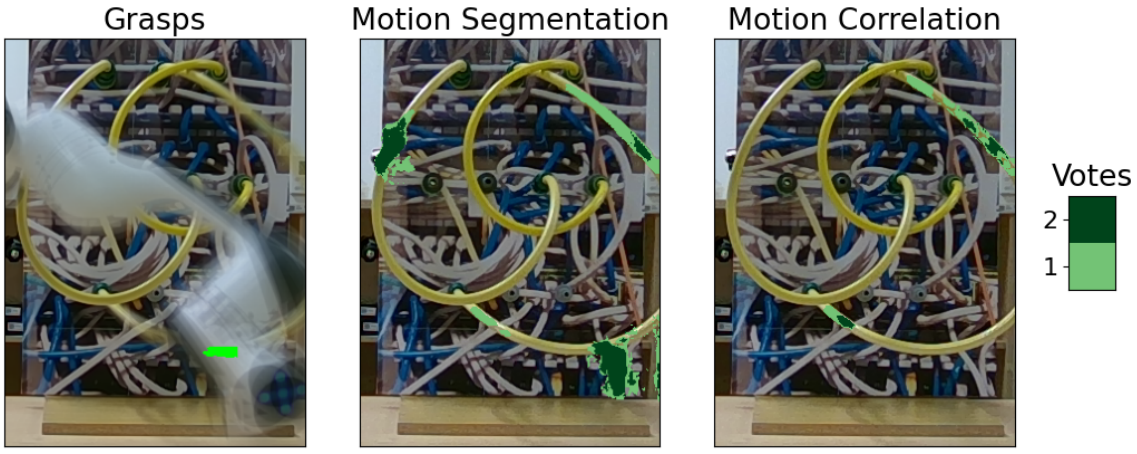
Gripper-cable motion correlation does not help in the cable exploration task when multiple cables tightly interact with each other during the gripper motion. If that happens, the robot needs to try a different action which would avoid the tight interactions with neighboring cables. Our approach uses two different actions but a robot may need a more diverse set of actions to successfully explore more complex cable configurations. When no action of a single gripper can avoid the tight interactions, one may need to use two robotic arms to verify cable segment connectivity by forceful interactions, e.g. by pulling two cable segments apart. We leave these possible extensions for future work.



(a)



(b)



(c)

Figure 6.4: Single grasp cable motion segmentation and correlation results. The robot grasped and moved the hose at a single grasp point. The shades of green in the segmentation and correlation plots indicate the number of gripper actions voting for a given pixel.

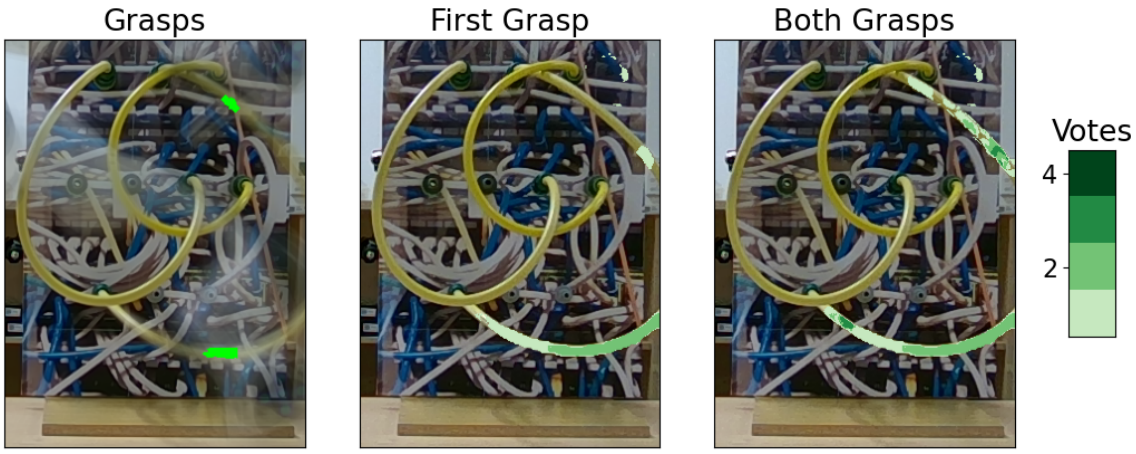


Figure 6.5: Sample *motion correlation* result given one and two grasp points. We provided the first upper grasp point manually. The grasp sampling method automatically generated the second lower grasp point. The shades of green in the segmentation and correlation plots indicate the number of gripper actions voting for a given pixel.

Chapter 7

Comparison between Event and Global Shutter Cameras

Engineers or scientists wanting to use a vision sensor usually ask which kind of vision sensor is the most suitable for the task. When considering using event-cameras, one may ask questions such as *"What is the application domain, in which an event-camera surpasses a fast frame-camera?"* or *"Which scene conditions can be detrimental to event-camera performance?"*

Here, we contribute to answering these questions by experimentally comparing two different event-cameras with fast global shutter cameras. We selected marker recognition on a rotating disk and fast object tracking as use-cases, where we expected event-cameras to triumph. We suggest that cameras performing well in our experiments will also perform well in other computer vision applications, and that these experiments will be useful when evaluating future event-cameras. The experiments' scope permits evaluation of the effects of sensor latency, temporal resolution, or bandwidth efficiency. To assess some of the event-camera performance limitations, we measure the latency of event response to different contrast stimuli at several sensor illuminance levels.

In terms of information coding efficiency, we found that an event-camera outperforms a frame-camera in all the tasks presented in this chapter. Large and sudden positive contrast changes damage the performance of current event-cameras. Too low event readout bandwidth in the best-tested event-camera limits the range of speeds at which object recognition works well. Frame- and event-cameras perform similarly well in terms of the accuracy of moving object position estimation.

We experimentally tested physical event- and frame-camera pixel arrays on scenes containing two distinct brightness levels and the transitions between them. We performed rotating disk experiments with accurate position ground truth and recorded the data for multiple distinct calibrated sensor illuminance levels. In addition to marker recognition, the cameras tracked a black dot rotating on the white disk. In our pixel latency experiments with two event-cameras, we present the latency of event response to negative and positive relative contrast steps across several contrast magnitudes and initial illuminance levels. Last but not least, we demonstrate how the cameras perform in a highly controlled ballistic experiment.

7.1 Materials and Methods

7.1.1 Methodologies, experiments

We followed two experimental methodologies commonly used when comparing devices. The first explores a controlled experiment using a deliberately simple device, cheap, and easy to duplicate. We observed contrast patterns on a rotating disk at a controlled rotational speed. This approach differs from a rather expensive, high speed pulse lighting used for comparing high-speed frame-cameras in [Manin et al., 2018].

We printed a contrast black dot 1 mm (12 pixels) in diameter or five different 3 $\times$ 3 mm (36×36 pixels) ArUco markers (squares with a black background and a white generated 4×4 binary pattern [Garrido-Jurado et al., 2014], marker generation and detection software are available in OpenCV [Aru,]), both on a white background, and pasted it to a rotating disk, at the radius of 85 mm. The disk was propelled by a BLDC servo motor D5065-270kv [Odr, a] controlled by an ODrive v3.6 controller [Odr, b]. An encoder CUI Devices (Lake Oswego, OR,

USA) AMT102 [Enc,] with 8192 counts per revolution tracked the motion of the motor shaft. An STMicroelectronics (Geneva, Switzerland) STM32 microcontroller unit (MCU) [STM,] performed the encoder readout and synchronization (see below for details). We could vary the number of revolutions in the range 0.5–40 rps, which corresponds to the dot/marker peripheral speed in the range 0.27–21 m/s and mean image speed 3.3–260 kpx/s. The scene was illuminated by an adjustable non-flickering panel light FOMEI (Prague, Czech Republic) LED WIFI-36D with color temperature set to 3700 K [Fom,].

The second methodology tested event-cameras in a demanding practical use-case. We observed a flying bullet at high-speeds shot from a ballistic test barrel under controlled lighting, chosen because we found a collaborating ballistic laboratory specialized in testing personal firearms. This allowed us to test event-cameras at their speed limits. We measured related phenomena simultaneously with an expensive high-speed frame-camera. The speed of a bullet was measured independently by a Doppler radar along the bullet trajectory and by light gates at a distance of 2.5 m from the muzzle of the barrel. These two methods provided us the ground truth for the projectile speed.

7.1.2 Materials

We tested two event-cameras: iniVation (Zurich, Switzerland) DVS240 [Cam, c] (DVS240 in short), which is an evolved version of the popular DAVIS240 [Brandli et al., 2014], and Prophesee (Paris, France) ATIS HVGA Gen3 [Cam, e] (ATIS in short). Posch et al. [Posch et al., 2011] presented an earlier generation of the ATIS sensor.

Table 7.1 in the event-camera survey [Gallego et al., 2019] compares several commercial or prototype event-cameras. Some of them have better specifications than the two event-cameras of ours. However, we constructed our benchmark experiments such that only pixel and readout design affect event-camera performance. Larger camera pixel array resolution, for example, would not affect the reported performance metrics. Given our benchmark design, the DVS240 (DAVIS240) and DAVIS346 are still the best sensors produced by the company iniVation mentioned in the table. The Samsung (Seoul, South Korea) cameras were not commercially available in 2020: the exception we found was the "Samsung SmartThings Vision" home monitoring device, which has an event-camera embedded inside. However, we did not find a simple way of connecting the embedded camera to a computer and recording the events it emits. Before buying the Prophesee ATIS camera, we briefly experimented with the CelePixel (Shanghai, China) CeleX-IV camera. Although its specifications on paper are impressive, it performed much worse in our initial test than the first iniVation product commercially available, the DVS128 from 2008. Prophesee told us in May 2020 that they planned to release their Gen 4 CD sensor in Q4 2020 or later, and so we could not test it. These findings make us believe that the Prophesee Gen3 ATIS was one of the state-of-the-art commercially available event-cameras as of 2020.

The cameras we tested have been widely used by researchers and so are relevant to a large scientific community. Event-camera users may use our benchmark to test newer cameras.

We observed the phenomena in the rotating disk experiments with a global shutter frame-camera Basler (Ahrensburg, Germany) ACE acA640-750um [Cam, a] (Basler in short). In the shooting experiment, we used the global shutter frame-camera Photron (Tokyo, Japan) Fastcam SA-Z [Cam, d] (Photron in short).

We used a ballistic Doppler radar Prototypa (Brno, Czech Republic) DRS-01 [Bal,] together with Kistler (Winterthur, Switzerland) Type 2521A [Lig,] light gates for independent measurement of the bullet speed. Furthermore, the light gates also provided the trigger signal for the synchronization of the camera records. Because the output signal from the light gates is of an irregular shape, we used the programmable triggering unit Prototypa PTU-01 [Tri,] to create a regular rectangle impulse based on the light gate output signal. The rising edge of this rectangular impulse triggered the record of the tested cameras. The Veritaslight (Pasadena, California) Constellation 120 [Ver,] LED lights and DedoLight (Munich, Germany) Dedocool [Ded,] tungsten lights illuminated the scene during the ballistic measurements.

The rotating disk experiment is illustrated in Figure 7.1. Figure 7.2 depicts the ballistic experimental setup schematically. Figure 7.3 shows the corresponding photo of the laboratory shooting range.

Table 7.1 summarizes the basic parameters of the cameras used in our experiments. The stated Basler exposure time and frame rate are the fastest possible settings used at the highest tested sensor illuminance of 1000 lx. Increasing the illuminance to 2 klx resulted in overexposure. At weaker illuminance levels, the Basler exposure time was 1270 µs at 20 lx, 274 µs at 80 lx, and 59 µs at 400 lx, yielding a mean pixel brightness value of 42.0 on the white paper surface.

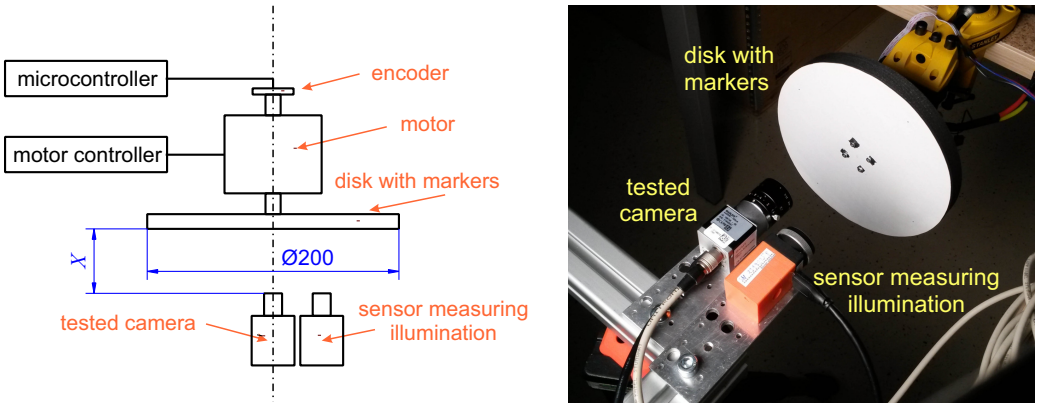


Figure 7.1: (**Left**) Schematic layout of the rotating disk experiment. The panel light is behind the cameras; (**Right**) a photograph of the rotating disk experiment.

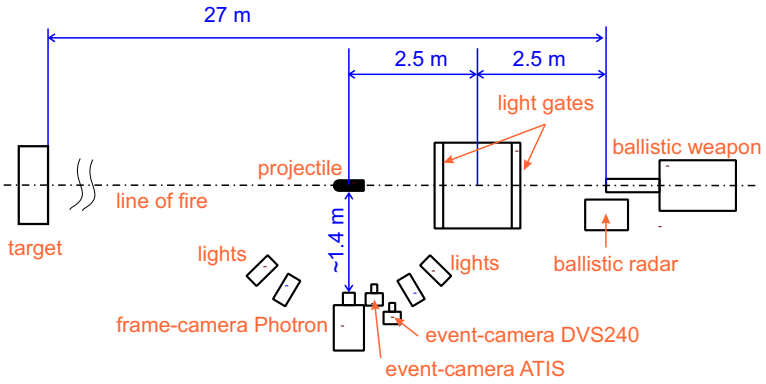


Figure 7.2: Schematic layout of the ballistic experiment.

Camera	DVS240	ATIS	Photron	Basler
resolution used [px]	240 × 180	480 × 360	640 × 280	480 × 360
pixel size [µm]	18.5 × 18.5	20 × 20	20 × 20	4.8 × 4.8
fill factor [%]	22	25	58	Unknown
PD size [μm^2]	75	100	232	<23
exposure [µs]	N/A	N/A	1	59
frame rate [FPS]	N/A	N/A	100,000	1000
max. event rate [Meps]	12	25	N/A	N/A
power [W]	0.02	0.1	230	0.6
price in 2020 [EUR]	2300	4000	~100000	335

Table 7.1: Camera parameters. Meps—million events per second, FPS—frames per second, PD—photo diode.

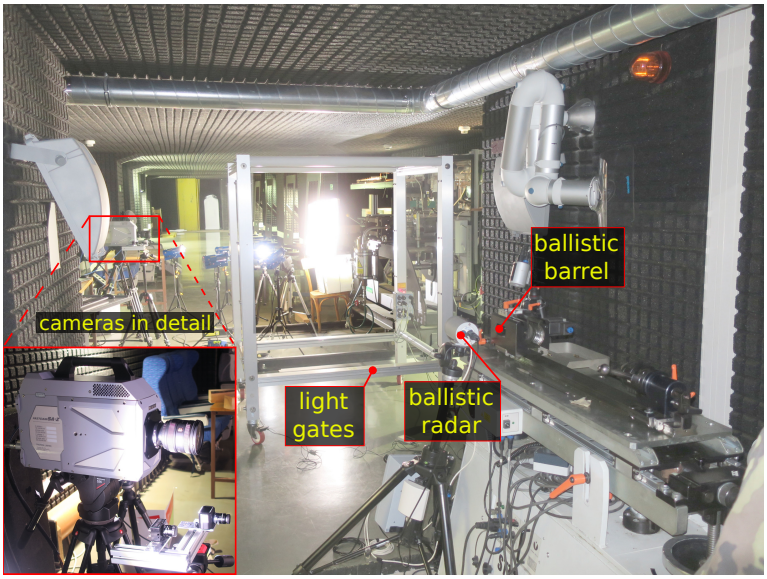


Figure 7.3: The ballistic shooting range setup.

It is important to compare the cameras on the same scenes with the same scale and sensor illuminance to avoid misleading conclusions. We observed these recommendations in all experiments with one minor exception in the ballistic experiment. The Photron camera had weaker sensor illuminance than the event-cameras.

Ideally, the pixel photodiode area in all the tested cameras should be the same. Unfortunately, it is not easy to find multiple different event- and frame-cameras with the same photodiode size.

7.1.3 Illuminance measurement

We used a Basler Dart daA2500-14um frame-camera [Cam, b] (Basler Dart in short) in tandem with a Sekonic (Tokyo, Japan) Speedmaster L-858D light meter [Sek,] (Sekonic light meter)

for measuring the sensor chip illuminance in the tested cameras.

The exposure time the Basler Dart camera needs to obtain the mean brightness value of 127 (half of the brightness range) is inversely proportional to the sensor illuminance. We calibrated the exposure times to the standard illuminance scale (in lux) using the Sekonic light meter. During calibration, both the lens-free Basler Dart camera and the light meter were pointed at an illuminated white wall from the same distance.

7.2 Event pixel response measurement

In order to separate the event pixel response limitations from the sensor readout limitations as much as possible, we performed additional event pixel response measurements.

A pre-programmed arbitrary waveform generator controlling an LED produced steps in relative contrast as the stimuli for a tested event-camera. Relative contrast change from time t_0 to time t_1 is the ratio

$$\frac{I(t_1) - I(t_0)}{I(t_0)},$$

$$t_0 < t_1, \tag{7.1}$$

where $I(t)$ is light intensity at time t. A sample relative contrast pulse train is shown in Fig. 7.4a.

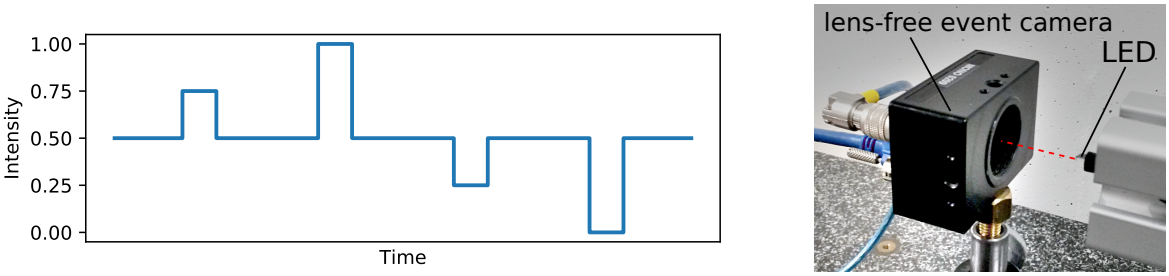
To verify the resulting LED contrast levels, we mounted a photodiode opposite the LED. Using an oscilloscope, we measured the photocurrent flowing through the photodiode, which is proportional to the illuminance.

The waveform generator was synchronized with the event-camera at each stimulus onset using external trigger signals sent to the event-camera.

We tested the camera in a completely dark room without a lens in this experiment to ensure homogeneous and fully controlled illuminance across all its pixels, see Fig. 7.4b.

To prevent event readout congestion, we activated only a small rectangular pixel region of interest (ROI) in the event-camera chip. However, to make the experimental results more externally valid, the region of interest should not be too small. Electronic noise or event readout interactions among neighboring pixels may affect the observed pixel response in a real-world scenario. We chose a 20–40 pixel large ROI so that one event may be read from each pixel in 2–4 µs at a 10 Meps event rate.

An event-camera might assign the same timestamp to events emitted within a short time period. To avoid this potential pitfall, we recorded the event-camera response to contrast stimuli of different duration (from 2 µs up to 3 ms). This way, only sufficiently fast events emitted during the stimulus presence contributed to the measured pixel response.



(a) (b)

Figure 7.4: (**a**) Sample relative contrast pulses; (**b**) the hardware setup for the pixel response measurements.

7.3 Rotating disk experiment

The rotating disk allows us to test high-speed visual sensing with a rotary encoder's accurate position ground truth. Storing the camera measurements for all rotary encoder positions at low speed gives the position ground truth for the camera measurements at higher speeds.

Data recording in general works as follows. The above-mentioned STM32 microcontroller unit (MCU) reads the encoder position at the beginning of every global shutter exposure. In the event-camera case, the MCU reads the encoder regularly (at 16 kHz at most). When the encoder is read, the MCU sends an external trigger signal to the event-camera. The event-camera captures the timestamp of the trigger signal and sends it over the USB interface together with the DVS events timestamped by the same clock.

7.3.1 Intensity Reconstruction and Marker Detection

To test all the cameras on a simple pattern recognition task, we chose to detect ArUco markers [Garrido-Jurado et al., 2014] rotating on the disk.

In the case of the event-cameras, the markers are detected in intensity images reconstructed from events. We used a state-of-the-art intensity reconstruction method called E2VID described in [Rebecq et al., 2019a, Rebecq et al., 2019b]. Code is available [Cod,].

The E2VID method uses a recurrent convolutional neural network whose architecture is similar to UNet. In each iteration, the network computes a reconstructed intensity image as a function of a batch of events and a sequence of several previously reconstructed intensity images. Rebecq et al. stored each event batch for the network input into a spatio-temporal voxel grid. The network was trained in a supervised mode on simulated event sequences and corresponding ground-truth intensity images.

A faster and smaller neural network version of E2VID was published in [Scheerlinck et al., 2020]. More recently, Ref. [Stoffregen et al., 2020] outperformed E2VID on certain event datasets by training the neural network on augmented simulated data with a wider range of event rates and contrast thresholds. We did not use [Stoffregen et al., 2020] as we found it after we had processed our experimental data using E2VID.

7.3.2 Dot Position Estimation

We need a simple but robust method to estimate the rotating dot position at wide ranges of event noise, image speed, and illuminance.

To obtain speed-invariant scene edges in a constant appearance scene, we accumulate a constant number of events for each position estimate [Liu and Delbruck, 2018]. The accumulated events should at least cover the entire dot circumference to enable reliable position estimation.

Our pixel response experimental results suggest that event-cameras respond more quickly to negative than to positive contrasts. Thus, we accumulated only the negative events in the rotating dot and ballistic experiments. Their median coordinates yielded the dot position estimate.

In the case of the Basler frame-camera, we selected the black dot pixels using global thresholding. The black dot median coordinates yielded the position estimate. The event-camera position estimates may be offset from the frame-camera estimates. This does not matter in the rotating disk experiment, where each camera generates its own position ground truth at low speeds. The offsets do not affect projectile speed estimation in the ballistic experiment. The offsets only influence the position estimation, which we discuss together with the ballistic experimental results.

7.4 Ballistic shooting range experiment

We conducted the ballistic experiment to estimate the potential of event-cameras in tracking and speed measurement of a high-speed object, in our case, the shot projectile.

We used the following cartridges in the experiments: (a) a 9 × 19 mm pistol cartridge, nominal speed $v_0 = 360$ ms^{-1}; (b) a 9 × 19 mm pistol cartridge, reduced speed $v_0 \approx 100$ ms^{-1}; (c) a 7.62 × 51 mm rifle cartridge, nominal speed $v_0 = 850$ ms^{-1}.

The cartridges were shot from ballistic test barrels. The projectile velocity has been measured using a Doppler radar and the light gates as a reference.

We compensated for lens distortion and calibrated the cameras in a single world coordinate frame using a chessboard calibration pattern to obtain comparable position estimates from all the cameras. We synchronized the cameras using the pulse signal from a projectile-detecting light gate between the gun-barrel and the camera field of view. Each projectile velocity estimate was calculated as the numerical differentiation of the projectile position.

In the frame-camera case, the projectile image coordinates were determined by global thresholding.

Accumulating a constant number of events for each position estimate resulted in significantly higher dispersion of the instantaneous velocity estimates than accumulating events for a constant time. Therefore, we estimated the projectile image coordinates as the centroid of the negative polarity events accumulated in a constant temporal interval.

7.5 Experimental results

We present the minimum measured latency across illuminance and contrast of both tested event-cameras in this section. The rotating disk results compare the sampling/detection rates and densities of the tested cameras as functions of image speed and illuminance. We report position estimation errors in the rotating dot experiment. The ballistic results include speed and position estimation errors. We support these quantitative results qualitatively by showing sample images. Finally, we compare the data efficiency of event- and frame-cameras on the rotating dot and rotating marker tasks.

7.5.1 Pixel Latency in Event-Cameras

The minimum pixel latency is the shortest stimulus duration required for an average pixel to emit one event with probability at least ≥ 0.5.

Larger relative contrast magnitude and larger illuminance cause lower minimum latency, see Figure 7.5. We observed that the latency of negative polarity events is lower than the latency of positive polarity events given the same absolute contrast. The ATIS sensor consistently outperformed the DVS240 in terms of the minimum pixel latency.

The shortest latency was 4 µs in the ATIS and 100 µs in the DVS240 sensor, both at the −100% contrast and the strongest tested reference illuminance of 2000 lx. Our ATIS result almost matches the best latency of 3 µs reported in [Posch et al., 2011]. However, we were not able to reproduce the 3 µs latency reported in [Brandli et al., 2014], which describes the DAVIS240 sensor, the preceding version of the DVS240 camera tested by us.

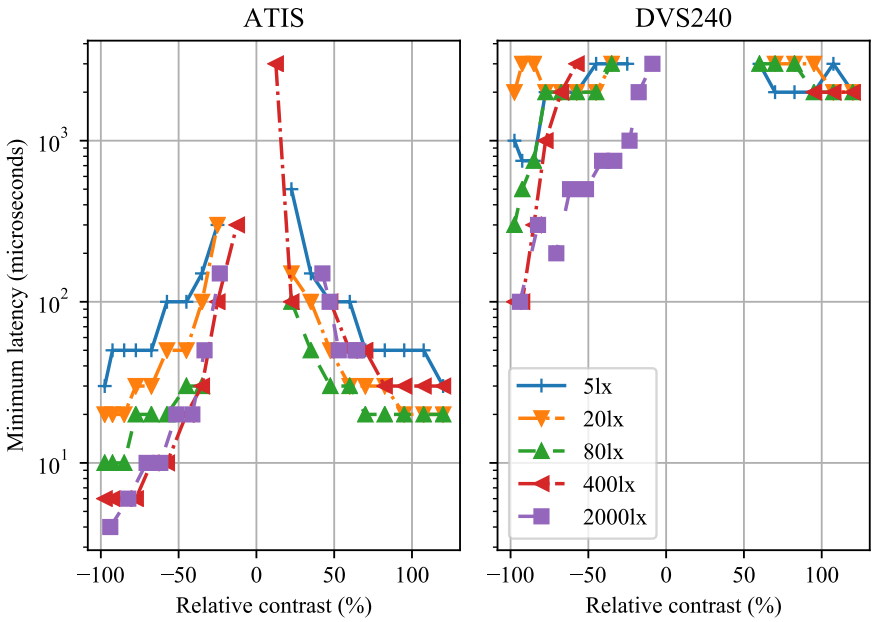


Figure 7.5: The minimum pixel latency of the ATIS and DVS240 event cameras depends on the contrast and illuminance (see the legend).

7.5.2 Pixel response

The event response of DVS pixels varies with the reference pixel illuminance and the contrast of a pulse stimulus applied. For each tested contrast level, the plots in Figure 7.6 show the distribution of the number of events emitted by a pixel across all observations (trials and enabled pixels). 250 observations were done with the ATIS pixels and 360 with the DVS240. The plots display ATIS and DVS240 results at four selected reference illuminance levels (20, 80, 400, and 2000 lx). The stimulus duration was kept constant at 3000 μs which was long enough to capture the majority of the pixel transient response of the slowest event camera and short enough to minimize the impact of noise events on the results.

As expected, larger absolute contrasts induce more events in both sensors. Somewhat surprisingly, both tested sensors exhibit stronger magnitude of event response to negative contrasts than to positive ones.

The ATIS pixel responses to a certain contrast stimulus can be quite diverse. They even range from one to more than five events for the same 70% contrast stimulus at 20 lx illuminance. This response diversity of the ATIS pixels decreases towards higher illuminance levels and lower absolute contrasts. Note that for absolute contrasts larger than $40 - 50\%$, an ATIS pixel is almost guaranteed to respond by at least one event regardless of illuminance.

In terms of response diversity, the DVS240 pixels seem to be quite the opposite of the ATIS. While most pixels respond by one to two events to a stimulus stronger than 50%, many do not respond at all. Strangely, the response diversity significantly increases at negative contrasts of the 2000 lx scenario, while almost guaranteeing at least one event per negative stimulus stronger than -25%.

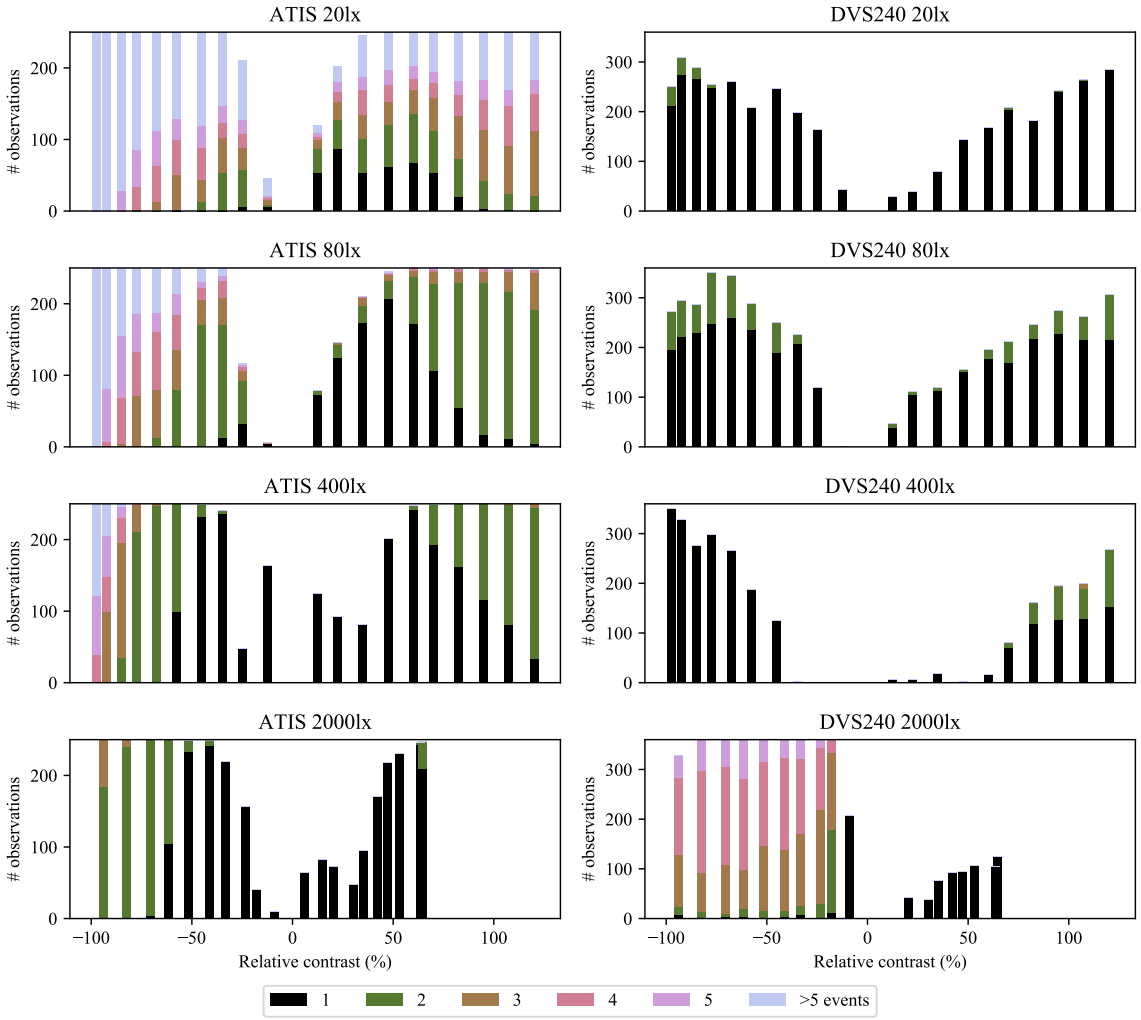


Figure 7.6: Pixel response of ATIS and DVS240 to 3000 μs long pulse stimuli at several levels of sensor illuminance and several contrasts. No experimental stimuli were emitted for contrasts above 60% in the 2000 lx case. The range of the vertical axes is always equal to the total number of observations performed.

Figure 7.7 presents the dependence of the event response on the stimulus duration, while keeping the illuminance level fixed at 400 lx.

In general, response diversity and magnitude increases in both sensors with increasing stimulus duration and absolute contrast. The ATIS pixel response distribution eventually stops changing for stimulus durations larger than 100 μs. In case of the DVS240, one has to extend the stimulus duration to at least 3000 μs for the same thing to happen (for the 3000 μs plot, see DVS240 400 lx in Figure 7.6).

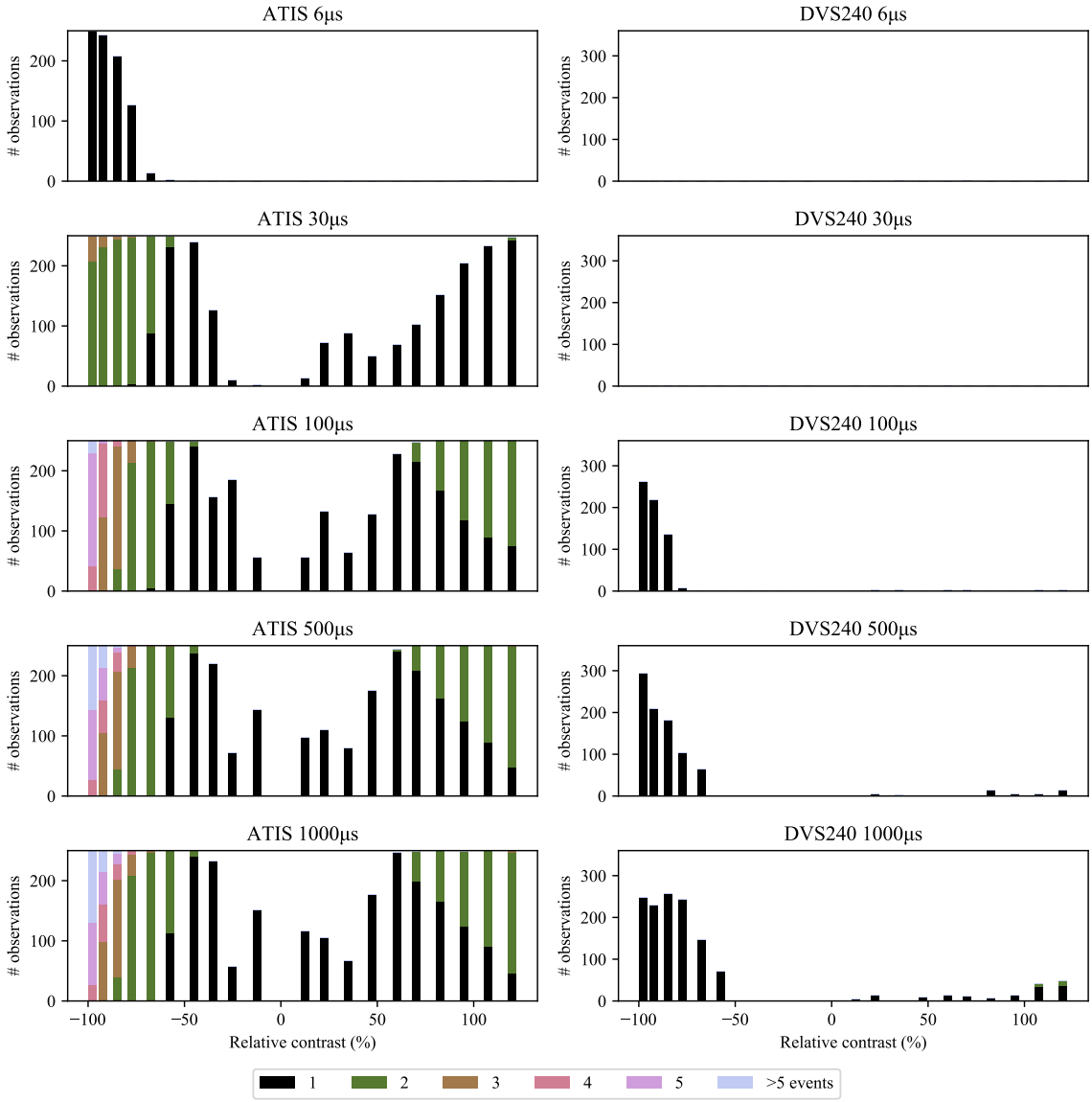


Figure 7.7: Pixel response of ATIS and DVS240 to variably long stimuli of several different contrasts at 400 lx reference sensor illuminance. The range of the vertical axes is always equal to the total number of observations performed.

7.5.3 Rotating Dot Experiment

The black dot has a diameter equal to 12 pixels in our experimental setup, cf. Figure 7.1. We assume that 19 events are needed at least to cover the half of the dot circumference. We chose to accumulate $N_c = 100$ negative polarity events to increase robustness to noise events or event readout failures.

The rotating dot position estimate error grows with increasing image speed and decreasing illuminance in general. Figure 7.8 shows the best results achieved for each camera and illuminance level.

However, we noticed several exceptions to the general rule for the DVS240 camera. Surprisingly, the position error is approximately twice as large with the 2000 lx illuminance as with the 400 lx for image speeds below 40 kilo-pixels per second (kpx/s). Furthermore, there seems to be a short speed interval around 80 kpx/s where the position estimation error plot lines of all the illuminance levels intersect.

The sharp error peaks at around 150 kpx/s in the ATIS and Basler subplot of Figure 7.8 were caused by mechanical resonance of the rotating disk at the respective rotational velocity of ca. 22 revolutions per second.

The dot position estimation error with the Basler camera at the strongest illuminance is comparable to the ATIS result, whereas it is worse at weaker illuminance (Figure 7.8). However, we cannot conclude that event cameras are inherently better than global shutter cameras at weaker illuminance because the photodiode area is at least four times smaller in the Basler than in the ATIS pixels, see Table 7.1.

We needed to set a higher ATIS pixel sensitivity $s = 50$ at lower illuminance levels (20 and 80 lx) than at higher illuminance levels (400 and 2000 lx, sensitivity $s = 40$) in order to obtain the lowest position errors presented. In contrast, we obtained the lowest DVS240 errors with the same bias setting.

With the ATIS sensor, events spread more in space due to increasing speed or decreasing illuminance level, see sample ATIS images in Figure 7.9. The prolonged and more uncertain edges of the event-images resemble the effects of motion blur in images from the Basler global shutter camera in Figure 7.10. The positive polarity edges tend to be blurred more than the negative ones, especially at higher speeds and lower illuminance levels.

All the DVS240 samples in Figure 7.9 contain a long tail of positive polarity events. Higher speeds often cause captured events to spread across a couple of pixel rows. At speeds around 200 kpx/s, a distinct but blurred group of negative polarity events was seen only at the highest illuminance level.

As the position estimation method is event-based, the effective temporal sampling rate automatically increases with image speed in both event-cameras, see Figure 7.11. The Basler global shutter sampling rate remains fixed but adapted to the exposure time of each illuminance level. The lower sensitivity setting of 40 in the ATIS causes a lower sampling rate, as it takes more time to collect a constant number of events with lower sensitivity than with a higher one. Note the general trend of higher sampling rates at weaker illuminance levels in both cameras. Finally, when the pixels gradually stop detecting contrast at high speeds, the sampling rate decays.

Although the event-camera temporal sampling rate increases in general with image speed, the increase is too slow to preserve a constant spatial sampling density. The spatial sampling density monotonically decreases with increasing speed, see Figure 7.12 showing the number of independent samples obtained per 100 pixels of distance travelled by the rotating disk surface. This may be related to the observation of the area occupied by a fixed number of events increasing with increasing speed in the images of Figure 7.9.

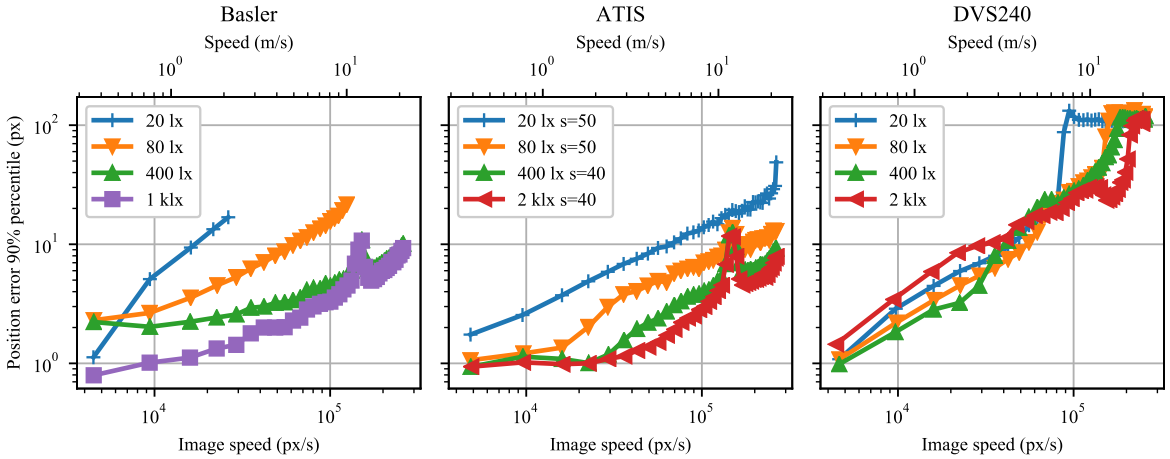


Figure 7.8: 90% percentile of the position estimation error as a function of dot image speed for Basler, ATIS, and DVS240 cameras. Data were measured for four background scene illuminance levels at the sensor plane.

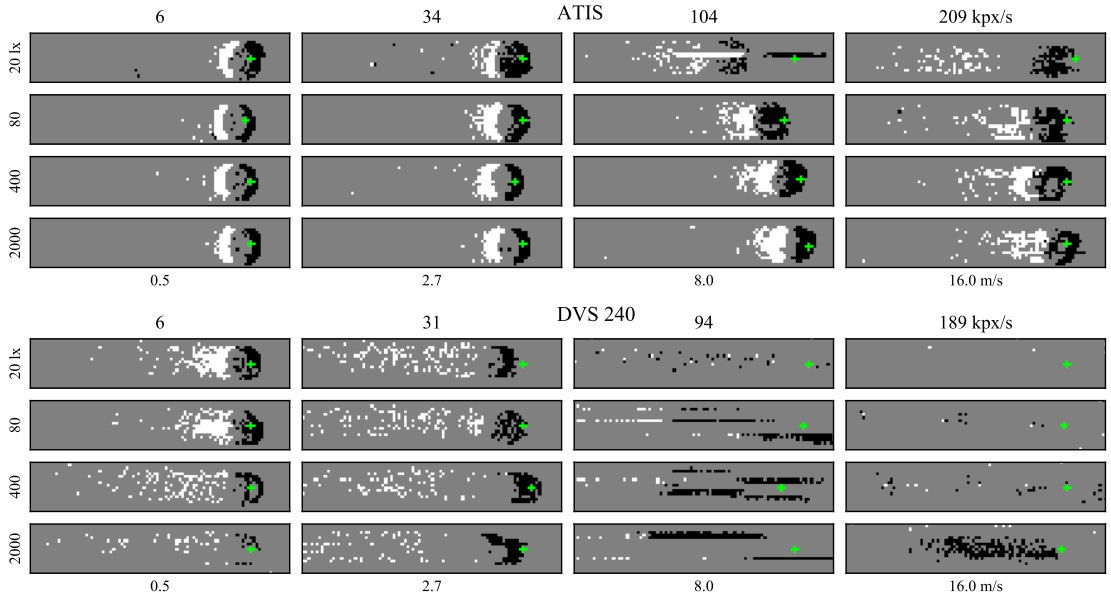


Figure 7.9: Sample ATIS (**top**) and DVS240 (**bottom**) event images of the black rotating dot moving left to right on white background at four different speeds (horizontal axis) and four different sensor illuminance levels (vertical axis). All images are cropped to 95×15 pixels from the original sensor plane with 200 events. Positive polarity events are white, negative black, and the background is grey. The green crosses indicate the ground truth median coordinates of the negative polarity events.

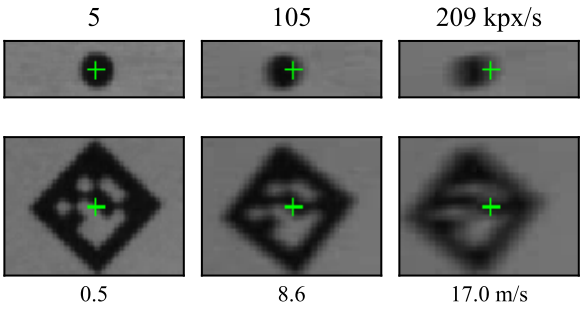


Figure 7.10: Sample images from the global shutter Basler camera. The green crosses show the ground truth position of the dot or marker center at the start of exposure. Exposure time 59 µs, sensor illuminance 1000 lx.

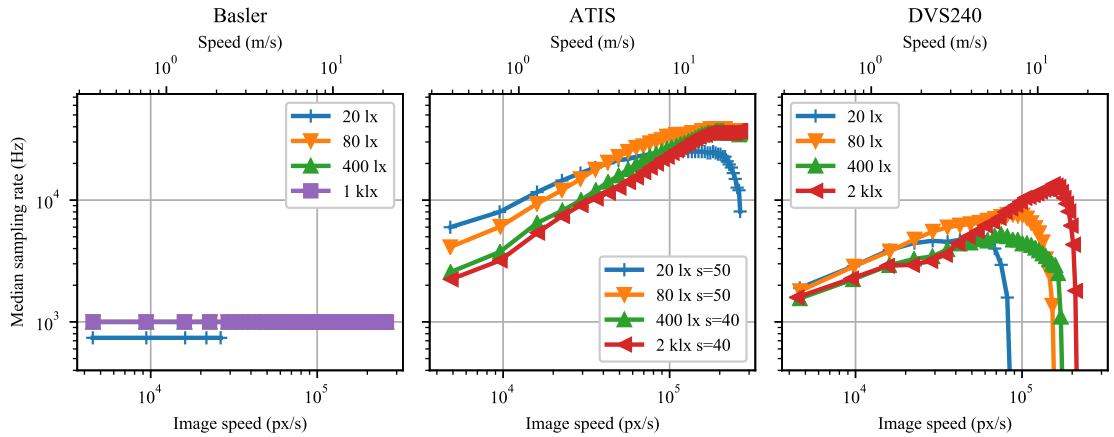


Figure 7.11: The effective median position sampling rate as a function of dot image speed for Basler, ATIS, and DVS240 cameras when computing each position estimate from 100 negative polarity events. Data were measured for four background scene illuminance levels at the sensor plane.

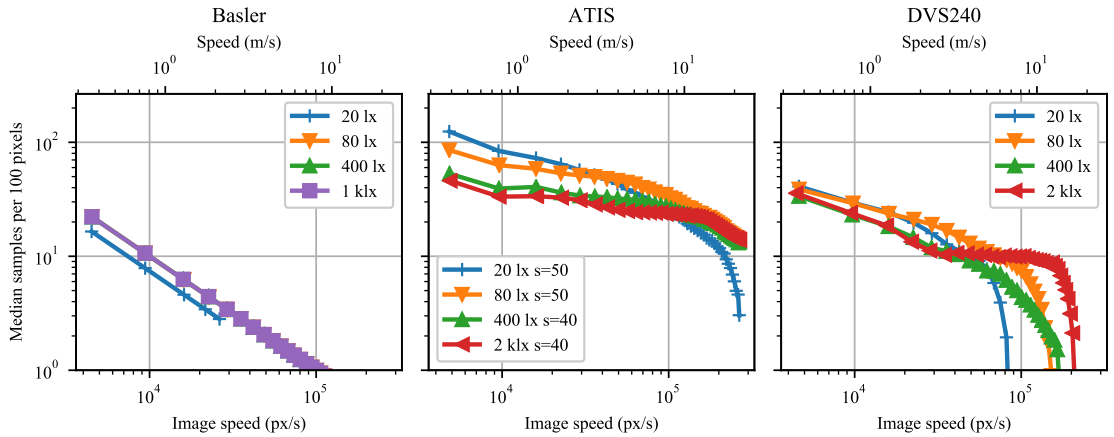


Figure 7.12: The median spatial sampling density in position estimates per 100 pixels distance as a function of dot image speed for Basler, ATIS, and DVS240 cameras. Each position estimate is computed from 100 negative polarity events. Data were measured for four background scene illuminance levels at the sensor plane.

7.5.4 Rotating Marker Experiment

To maximize marker detection density, we need to optimize the event batch size in the E2VID method used for reconstructing the intensity frames. The smaller the batch size, the more markers per disk revolution can be detected at lower speeds. However, larger batches provide for a more reliable intensity reconstruction at higher speeds. Of the batch sizes tested (270, 540, 1080, 2160, 4320 and 8640), we chose a size of 2160 events for the ATIS and 540 events for the 20 and 80 lx DVS240 recordings. These sizes enable marker detection at the highest speeds while maximizing detection density at lower speeds. We had to use 1080-sized event batches to achieve reliable low-speed marker detection in the 400 and 2000 lx DVS240 recordings.

The rotating marker results mostly resemble the rotating dot results. Spatial marker detection density in Figure 7.13 and the spatial dot sampling density in Figure 7.12 both decay with growing speed.

Likewise, the detection frequency of the event-cameras in Figure 7.14 grows with growing speed until reaching a maximum. Further increase in speed causes lower detection frequency. Compare with Figure 7.11.

We note that illuminance's impact on the ATIS detection performance is much lower than in the rotating dot experiment. This suggests that different factors limit the ATIS performance in each task.

The sample images in Figures 7.15 and 7.16 confirm these observations. The DVS240 image quality degrades gradually across the speed range. In contrast, the ATIS samples are almost perfect for most of the speed range, degrading sharply around 40 kpx/s. Furthermore, the ATIS image quality is less affected by illuminance than the one of the DVS240.

Marker detection is possible in (almost) every frame captured by the global shutter Basler camera, yielding a constant detection frequency equal to the frame rate, see Figure 7.14. Decreasing sensor illuminance demands longer exposure time, which causes stronger motion blur at a given speed, eventually resulting in marker detection failure. Furthermore, the exposure time restricts from below the interval between subsequent frames, limiting the frame rate at weaker illuminance.

Both event-cameras outperform the 1000 Hz Basler camera in the maximum achieved detection frequency. However, the Basler camera keeps detecting some markers at approximately four times higher speeds than the best event camera tested (ATIS).

Figure 7.10 shows sample Basler dot and marker images at the strongest illuminance level for three speeds. The stronger motion blur at higher speeds is clearly visible. For example, the marker moving at 209 kpx/s translated by approximately twelve pixels (1/3 of the length of its side) during the 59 µs exposure time. Nevertheless, it can still sometimes be detected and read by the detection algorithm.

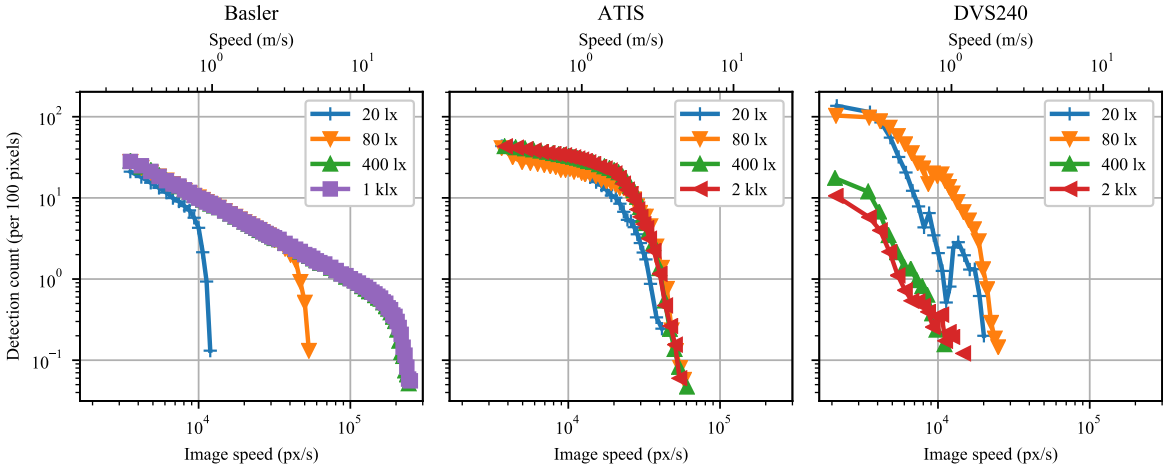


Figure 7.13: Mean number of marker detections per 100 pixels of marker trajectory as a function of marker speed for Basler, ATIS, and DVS240 cameras. Data were measured for four background scene illuminance levels at the sensor plane.

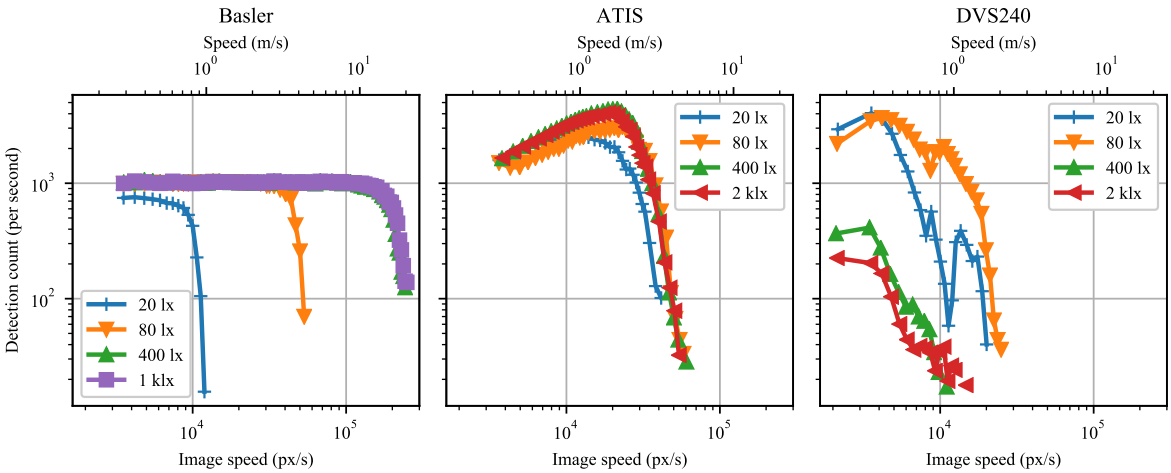


Figure 7.14: Mean number of marker detections per second as a function of marker speed for Basler, ATIS, and DVS240 cameras. Data were measured for four background scene illuminance levels at the sensor plane.

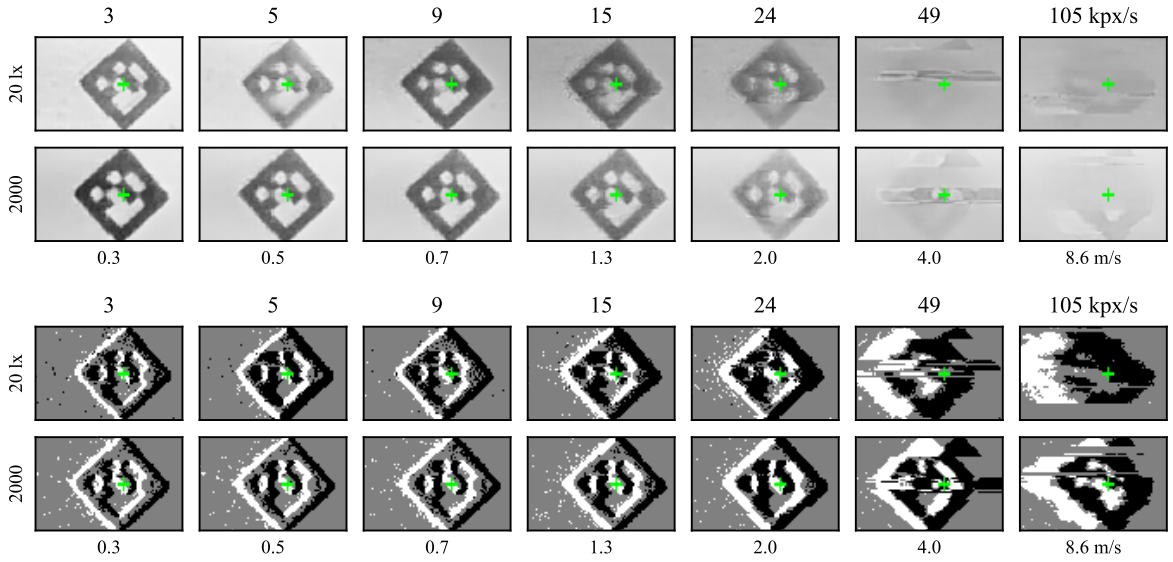


Figure 7.15: Sample reconstructed intensity and event marker images from the ATIS sensor. The images are shown for two illuminance levels at the sensor plane and seven image (**top**)/metric (**bottom**) speeds. The green crosses indicate the ground truth position of the marker center.

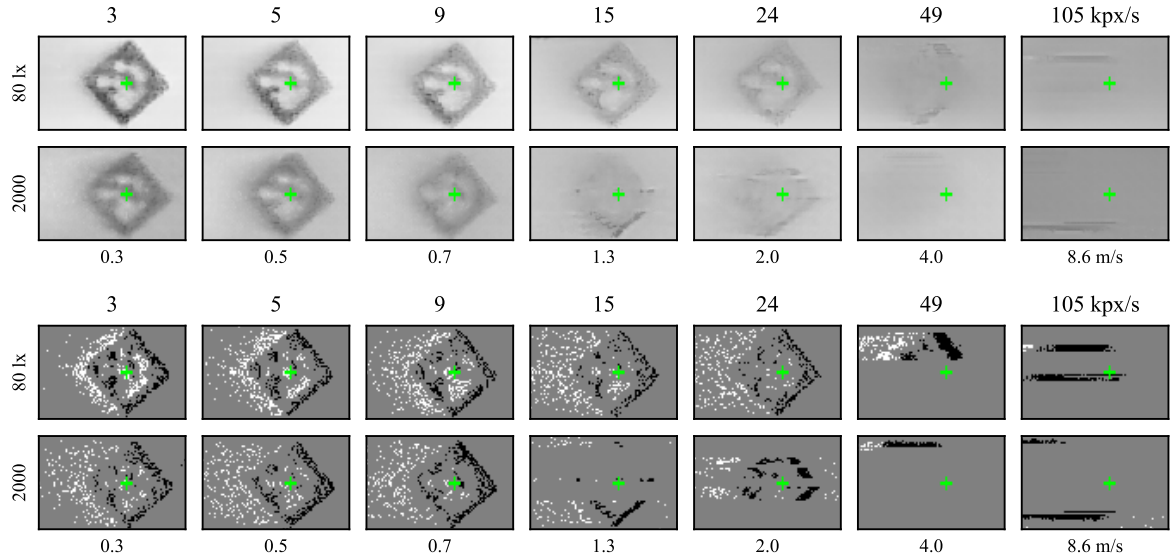


Figure 7.16: Sample reconstructed intensity and event marker images from the DVS240 sensor. The images are shown for two illuminance levels at the sensor plane and seven image (**top**)/metric (**bottom**) speeds. The green crosses indicate the ground truth position of the marker center.

7.5.5 Ballistic Experiment

The ATIS and DVS240 sensor illuminance was ca. 4500 lx in the ballistic experiment. Table 7.2 details the metric and image dimensions of the two projectile types we used. Sample images recorded by the Photron, ATIS, and DVS240 cameras at three projectile speeds are shown in Figure 7.17. The projectiles flew from the right to the left. While the effects of increasing speed are not visible in the 1 µs exposure images from the Photron camera, the leading negative

	Projectile 9 mm FMJ	Projectile 7.62 mm M80
World Diameter [mm]	9.03	7.83
Image Diameter PHOTRON [pixel]	7.50	6.50
Image Diameter ATIS [pixel]	7.92	6.89
World Length [mm]	15.80	29.46
Image Length PHOTRON [pixel]	13.14	24.50
Image Length ATIS [pixel]	13.83	25.79

Table 7.2: Dimensions of projectiles in the world and image units.

polarity edges in the ATIS images become more imprecise with increasing projectile speed. In addition, the trailing positive polarity events extend over more pixels at a higher speed. In our study, the DVS240 camera could not capture the projectile's position or appearance even at the lowest tested speed, where the 10 µs long event window contained events spread across a single pixel row.

The projectile position estimates from the ATIS event camera are very close to the ground truth Photron estimates, see Figure 7.18a. In the 365 m/s or slower recordings, the relative position variability is below 0.7% of the trajectory length. This is 3 mm or 2.5 px, and the distance traveled by the projectile in 8 µs. In the slowest 100 m/s shot, the events are accumulated mostly on the projectile's frontal edge during the 10 µs interval; see the sample ATIS event image in Figure 7.17. The thinner edge probably causes the systematic position shift ahead of the Photron estimates. In the 850 m/s experiment, the ATIS position estimates systematically lag behind the Photron estimates by ca. five millimeters.

The other contribution to the systematic shift between the projectile positions estimated from the Photron and the ATIS record could be the difference between the calibration plane and the real plane of the projectile motion. This difference is caused by the projectile trajectory dispersion.

Because numerical differentiation is very sensitive to noise, the relatively low noise in the determination of the projectile position results in large uncertainty of the estimated immediate projectile velocity, as shown in Figures 7.18b and 7.19. The ATIS immediate speed estimates exhibit significantly larger uncertainty at all the three tested projectile velocities than the Photron estimates.

However, Table 7.3 shows that the mean projectile velocity values determined using the event-camera and the Photron camera records agree. The uncertainty reported for both cameras in each of the three shots is the standard deviation of the immediate speed estimates from the mean speed. The highest relative difference between the measurements was 1.6% in the 850 m/s shot. Furthermore, both camera estimates of the mean projectile velocity correspond very well with the ballistic Doppler radar measurements.

We observe that, to our knowledge, these reports of using an event-camera in practical ballistic measurements are the first.

Mean Horizontal Speed (Meters per Second)		
Radar	**Photron**	**ATIS**
102.8	103.8 ± 9.2	103.0 ± 47.2
364.5	368.4 ± 7.0	363.7 ± 30.2
850.5	859.6 ± 10.3	846.5 ± 137.6

Table 7.3: Mean horizontal projectile speed estimates.

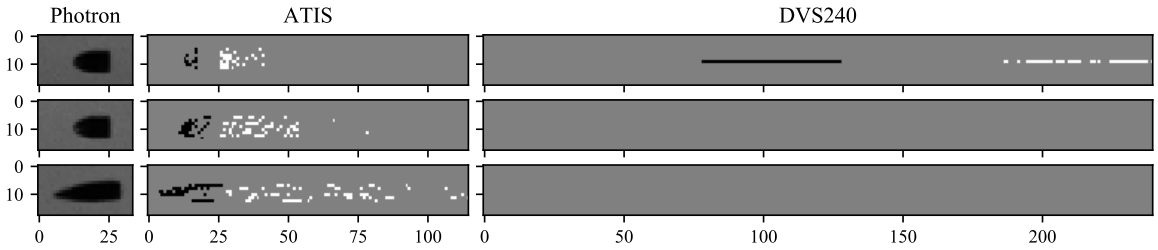
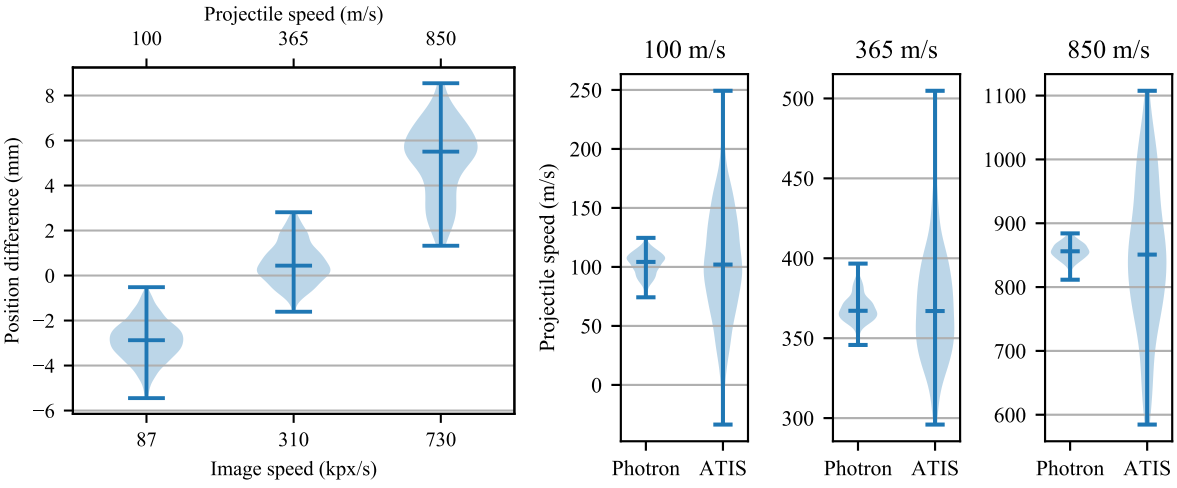


Figure 7.17: Sample images of the projectiles as seen by the Photron, ATIS, and DVS240 cameras. The event camera images display 10 µs of events, white are positive and black are negative polarity. The Photron exposure time was 1 µs. The projectile speed increased from top to bottom, from 100 through 365 to 850 m/s. (Image speed 87, 310 and 730 kpx/s.)



(a) (b)

Figure 7.18: (**a**) Distributions of the differences between the ATIS and Photron estimates of horizontal projectile position. Data were measured for three projectile speeds along a trajectory segment 440 mm long. A positive difference means that the ATIS estimate lags in time behind the Photron estimate. The horizontal lines of each violin show the maximum, median, and minimum position differences; (**b**) distributions of the ATIS and Photron projectile speed estimates along a trajectory segment 440 mm long.

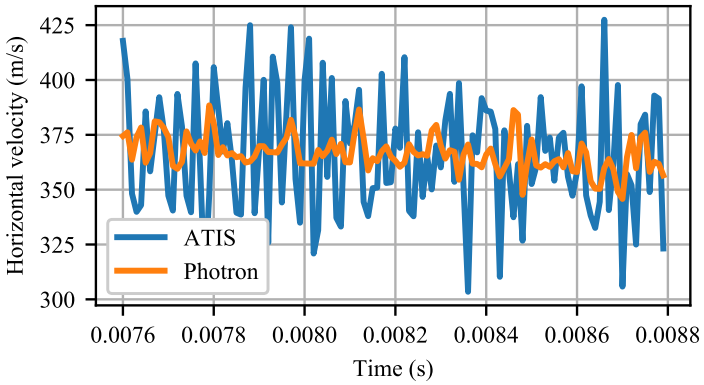


Figure 7.19: Horizontal projectile velocity estimated by the Photron and ATIS cameras along a 440 mm long trajectory segment at a constant 100 kHz sampling rate.

7.5.6 Data Efficiency

We compared the data efficiency of the ATIS event-camera and an idealized frame-camera of the same resolution in Figure 7.20. The information of interest, a detected marker or a dot position, is produced at the output sampling rate. The output rate is a function of sensor bandwidth, i.e., the amount of data per second sent by a sensor to the computer for processing. This function is task-independent for the frame-camera as each frame yields one output sample in the tasks we assumed.

We supposed that one event can be stored in approximately 27 bits. To encode the pixel coordinates, one needs $\lceil \log_2(N_{pixels}) \rceil = \lceil \log_2(480 \times 360) \rceil = 18$ bits. 1 bit is required to store the polarity and 8 bits for a 0–255 µs timestamp. (To keep track of time in longer recordings, a 32 bit long timestamp prefix can be sent every 256 µs, for example, increasing the sensor bandwidth only by 125 kbits/s.) If the entire camera and computer system are neuromorphic, there is no need to assign timestamps to the events [Mahowald, 1992].

Under these conditions, the ATIS event-camera was more data-efficient than the frame-camera, by two orders of magnitude in the rotating dot task and by one order of magnitude in the marker detection task. The ATIS marker detection curve in Figure 7.20 starts to decrease around 270 Mbits/s or 10 million events per second, probably due to hardware bandwidth constraints of the ATIS sensor.

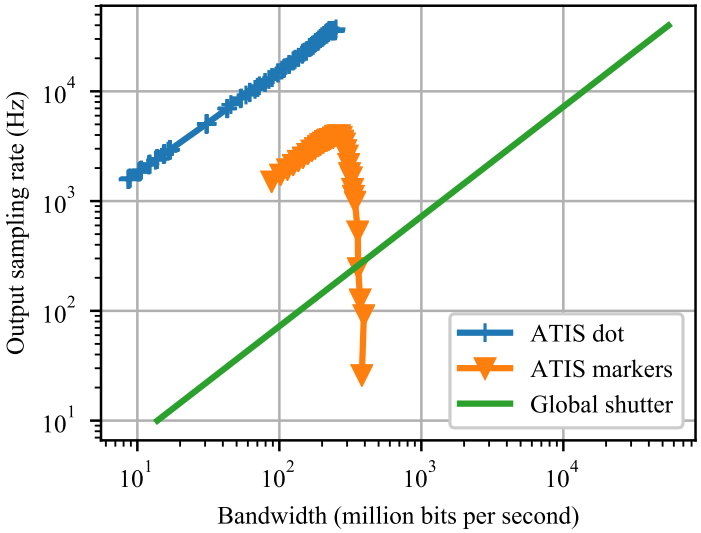


Figure 7.20: A bandwidth-performance plot shows how much data an event-camera (ATIS) and a global shutter frame-camera generate to reach a certain output sampling rate. The output sampling rate captures the number of marker detections or the number of rotating dot position estimates, both per second—assuming the strongest lighting tested (ATIS), 27 bits per event on average, negligible frame-camera exposure time, and sensor resolution of 480×360 pixels.

7.6 Discussion

Compared to an event-camera, a global shutter frame-camera could be more efficient in more complex scenes than in the simple scenes we tested. In the following illustration, we assume the sensor pixel resolution to be at most 512×512 pixels, requiring 27 bits per event on average and eight bits per one intensity sample. A larger resolution favors the frame-camera, as the events need more bits to encode the pixel coordinates. Once the number of instantaneously sampled pixels required to produce one output sample exceeds $8/27 = 30\%$ of the entire pixel array, the frame-based data encoding scheme becomes more efficient. When all pixels need to emit one event, e.g., when the global illuminance changes, the global shutter will be $3.4\times$ more efficient. Additionally, when multiple events per pixel need to be captured to recognize multiple contrast levels or large and sudden illuminance changes, frame-based sampling will be even more suitable.

Event readout aggregated by pixel rows ("burst-mode" arbiter [Posch et al., 2011, Guo et al., 2007]) trades lower bandwidth for lower timestamp accuracy when many pixels in the same row emit events approximately at the same time. In that case, the sensor sends the pixel column coordinate and the event polarity of each event. The common row coordinate and the timestamp need to be sent only once per row.

We observed the readout aggregation in action at large image speeds when the sensor readout capacity limits were reached. Both the ATIS and DVS240 event-cameras read multiple events from a single pixel row at the same time. See, for example, the fastest dots and the slowest ballistic projectile recorded by the DVS240 in Figures 7.9 and 7.17 or the fastest markers recorded by the ATIS in Figure 7.15. In [Holešovský et al., 2020], we observed that these readout limitations prevent accurate tracking of ballistic projectiles flying along pixel

columns instead of pixel rows of the ATIS camera.

The event-based readout can be both a blessing and a curse. On the one hand, it adapts the event camera to the scene dynamics. On the other hand, the camera cannot collect a brief, complete, on-demand snapshot of a rapidly changing large scene. Future event-cameras with extended readout bandwidth and sufficiently large on-chip memory buffers could alleviate this limitation.

Position estimation error tends to grow with increasing speed in the rotating dot experiment, even though a constant number of events is accumulated to obtain each estimate. This suggests that perhaps the error could be reduced in the event-camera records by explicit edge sharpening such as event lifetime estimation [Mueggler et al., 2015, Lee et al., 2019]. At lower speeds and stronger illuminance levels (up to ca. 40 kpx/s at 2000 lx), the ATIS negative event timestamps consistently increased from the leading to the trailing edge of the dot, suggesting that edge sharpening could be applied. However, only the estimation accuracy at the higher speeds and/or lower illuminance scenarios could significantly benefit from the edge sharpening. Unfortunately, we did not notice a systematic pattern in the event timestamps recorded at the higher speed and/or lower illuminance scenarios. We interpret this observation as event-camera "motion blur".

The event-camera was slow at detecting large positive relative contrasts, i.e., transitions from dark to bright illuminance, for three reasons. First, such contrasts can be almost infinite, demanding an infinite event response. Second, the minimum event pixel latency is longer at lower illuminance levels, prolonging the total event response duration. Third, in general, the minimum pixel latency is longer for the positive than for the negative contrast stimuli.

As expected, keeping the scene appearance fixed, the fixed-size event packet measurement mode increased the output sampling rate with increasing stimulus speed. At the same time, however, the spatial sampling density monotonically decreased with increasing speed. This implies fixed-sized event batches surprisingly do not guarantee perfect speed-adaptive scene sampling, as assumed, for example, by [Liu and Delbruck, 2018].

The dot position estimation error increases with growing image speed in all the cameras tested. This may be related to the spatial sampling density decreasing with the growing speed in all the cameras.

The ATIS event camera measured a ballistic projectile's mean velocity at an accuracy comparable to the very high-speed 100 kHz global shutter Photron camera. However, the ATIS was significantly worse than the Photron at measuring the instantaneous projectile velocity.

Our ballistic experiment validated the rotating disk experiment results. These experiments do not differ qualitatively. We performed the ballistic experiment to test quantitatively higher image speed, which is 730 kpx/s compared to at most 300 kpx/s in the rotating disk experiment.

7.7 Conclusions

Event-cameras' performance was limited by pixel latency when tracking small objects and by readout bandwidth in object recognition. When comparing the event- to frame-cameras, we

saw analogies between event pixel latency and exposure time and event readout bandwidth and frame rate.

We provide an answer to the question "What is the application domain, in which an event-camera surpasses a fast frame-camera?". In this study, the event-camera surpassed the fast frame-camera in terms of information coding efficiency in scenes with significant changes restricted to less than 30% of the field of view within the sampling period of interest. As the event pixel latency is significantly lower for negative than for positive contrast changes, the fastest scene changes should ideally be restricted to the negative contrast.

On the other hand, the coding efficiency analysis suggests that a frame-camera may outperform event-cameras in cluttered scenes. Assuming the readout bandwidth is fixed, the event-camera output sampling rate decreases as the scene complexity/clutter increases, whereas the frame-camera output sampling rate remains constant. We answer the question "Which scene conditions can be detrimental to event-camera performance?": highly cluttered scenes or scenes with sharp and strong positive contrast edges can be detrimental to event-camera performance.

Although a frame-camera is more limited by exposure time than bandwidth in high-speed object recognition tasks, the better event-camera we tested (ATIS) was limited more by its readout bandwidth. Future event-cameras with extended readout bandwidth and sufficiently large on-chip buffers could alleviate this limitation. Nevertheless, the potential and future perspective of the event-based principle of sensing is obvious and is emphasized, e.g., by a related DARPA Fast Event-based Neuromorphic Camera and Electronics (FENCE) program from December 2020 [DAR, 2020].

Surprisingly, the event-camera spatial sampling density monotonically decreases with growing motion speed. This fact may limit the applicability of existing event-based algorithms relying on fixed-sized event batches.

Finally, our results showed that frame- and event-cameras perform similarly well in moving object position estimation accuracy.

Data availability: The data presented in this chapter are freely available at `https://purl.org/holesond/cvut-ciirc-dvs-benchmark-2021`. The event-camera bias configurations used are included in the dataset package.

Chapter 8

Evaluation of Event Camera Optical Flow Algorithms

We tested two recent state-of-the-art (as of 2024) optical flow estimation algorithms for event cameras, [Shiba et al., 2024] and bflow [Gehrig et al., 2024], to evaluate their suitability for cable motion segmentation. [Shiba et al., 2024] is a multi-scale optimization method with a multi-reference focus loss function based on the contrast maximization framework. Bflow [Gehrig et al., 2024] is an artificial neural network predicting Bézier curve trajectories instead of pixel displacements. It searches for pixel correspondences in multiple correlation volumes computed from voxel grids representing the events. [Gehrig et al., 2024] trained the network on their own synthetic dataset containing indepedently moving objects.

We performed two sets of real-world experiments to record the event data for the evaluation: rolling aluminium rods on a table and moving freely hanging garden hoses. Rolling rods move at locally constant velocities and enable manual ground truth motion and flow estimation directly in the image frame thanks to their rigidity and simple shape. Moving hoses by hand does not guarantee constant velocities or enable simple ground truth flow estimation. However, such data can qualitatively demonstrate how the flow predictors perform on deformable cables.

8.1 Rolling rods

Fig. 8.1, 8.2, 8.3 show sample event-based optical flow estimation on a single rolling rod, two rods rolling opposite each other, and two rods rolling in the same direction at different speeds. The figures show sample event-based optical flow estimation using [Shiba et al., 2024] (without time-aware flow) and bflow [Gehrig et al., 2024] compared to manually obtained ground truth flow. These flow event images are binary images of warped events (IWE). The flow predicted by each method transformed the recorded events to a common timestamp. We plotted the transformed events to the binary IWEs, dropping the common (zero) timestamp and the event polarity. The color (hue, saturation) of each event corresponds to its optical flow vector (direction, magnitude). The labels at the sample points in each flow image show sample optical flow vector values in pixels. The RGB image and the flow predicted by MfnProb FT from RGB image pairs are for reference only as they did not have the same motion as the event data. We did not use the time-aware flow option of [Shiba et al., 2024] as it spreads many events into the static background space, cluttering the otherwise mostly sharp IWEs.

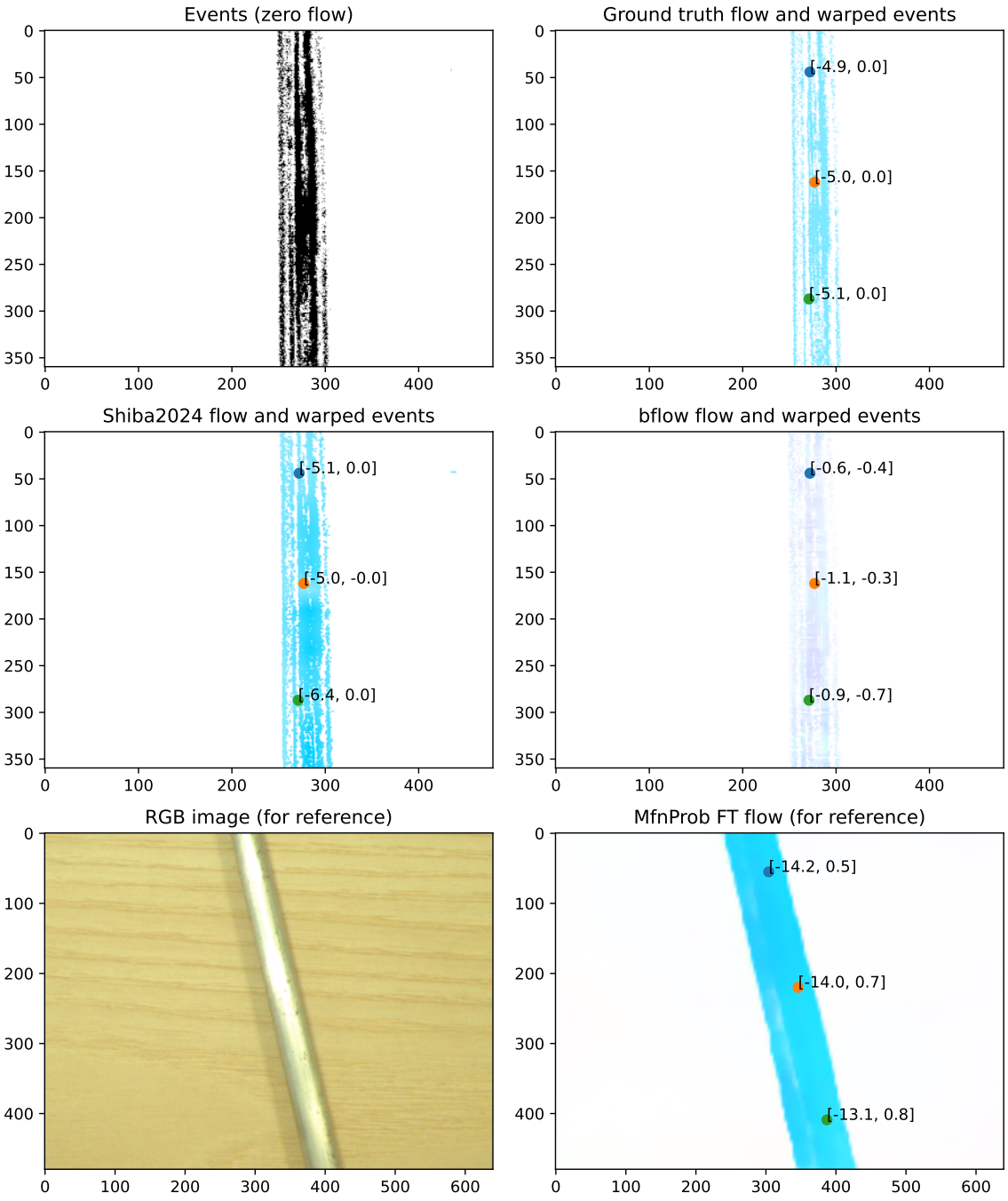


Figure 8.1: One rolling rod. The input was an event batch 0.0134 seconds long containing 15000 events.

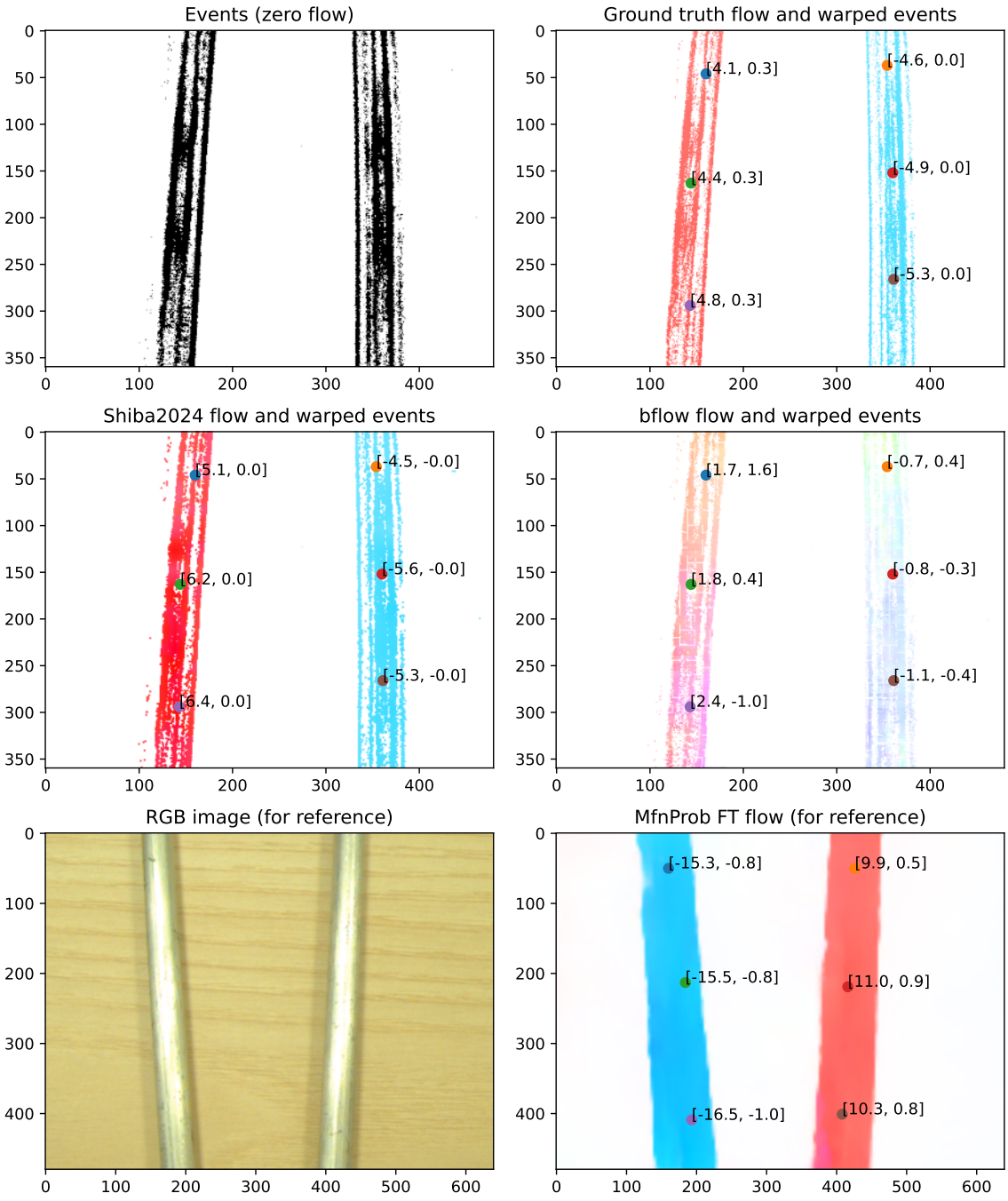


Figure 8.2: Two rods rolling opposite each other. The input was an event batch 0.0183 seconds long containing 30000 events.

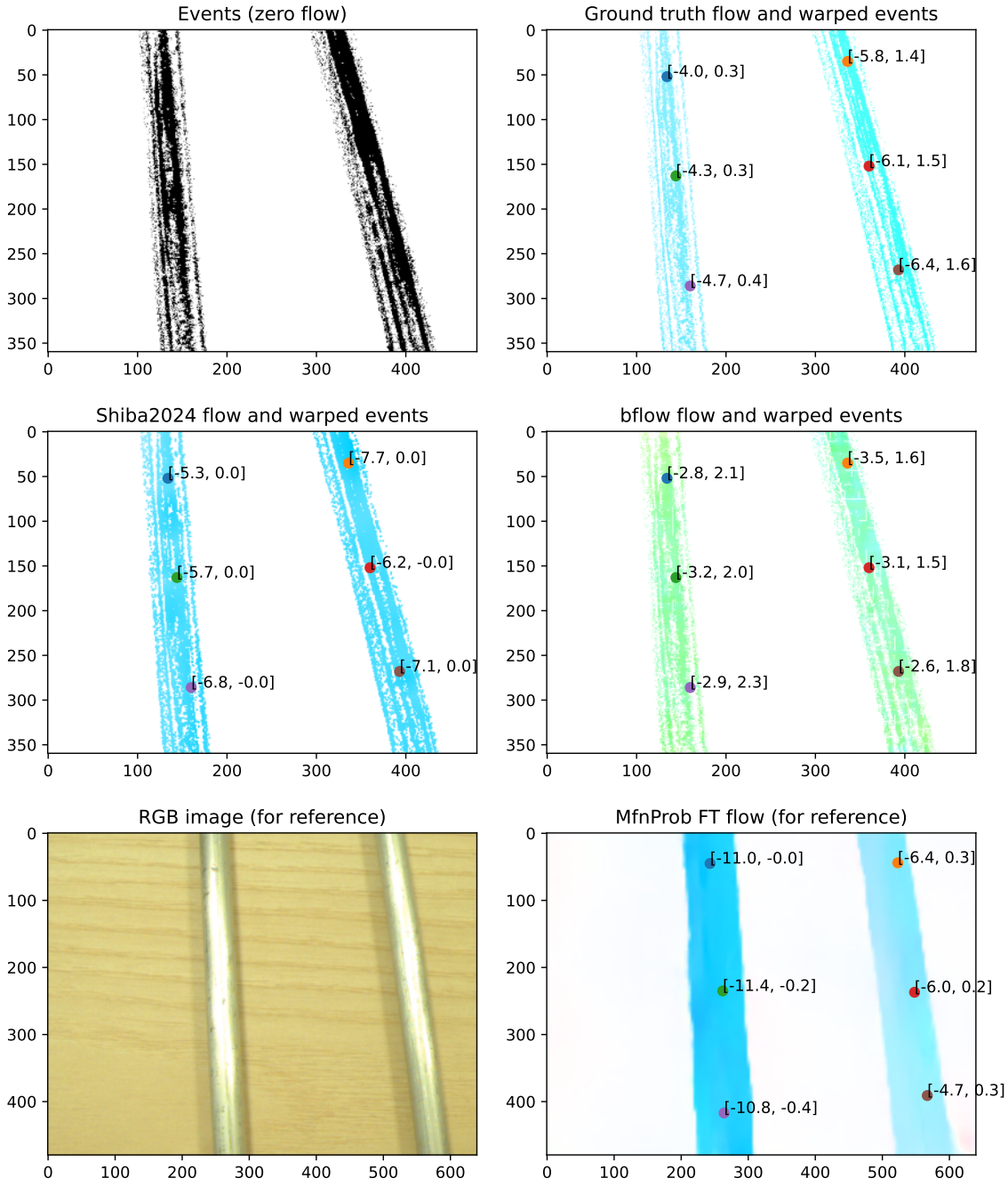


Figure 8.3: Two rods rolling in the same direction at different speeds. The input was an event batch 0.0142 seconds long containing 30000 events.

We note that the manually obtained ground truth flow usually yields better (sharper) IWEs than [Shiba et al., 2024] or bflow [Gehrig et al., 2024]. Bflow seems to be fairly inaccurate in all three rolling rod experiments. It usually predicts too low flow magnitude. [Shiba et al., 2024] predicts the flow accurately in many cases, such as for the single rolling rod in Fig. 8.1 or the right rod in Fig. 8.2. We see lower accuracy e.g. for the left rod in Fig. 8.2 or the two

rods moving in the same direction at different speeds in Fig. 8.3, where [Shiba et al., 2024] predicts very similar flow vectors for both rods.

8.2 Moving hoses

Fig. 8.4, 8.5 show sample event-based optical flow estimation on one and two moving hoses. The RGB image and the negative flow predicted by MfnProb FT from RGB image pairs are for reference only as they did not have the same motion magnitude as the event data. However, they had approximately the same motion direction as we captured the RGB images before and after the motion which generated the events. We additionally plotted the grayscale IWE computed directly by [Shiba et al., 2024] over the flow-colored binary event image for higher presentation clarity.

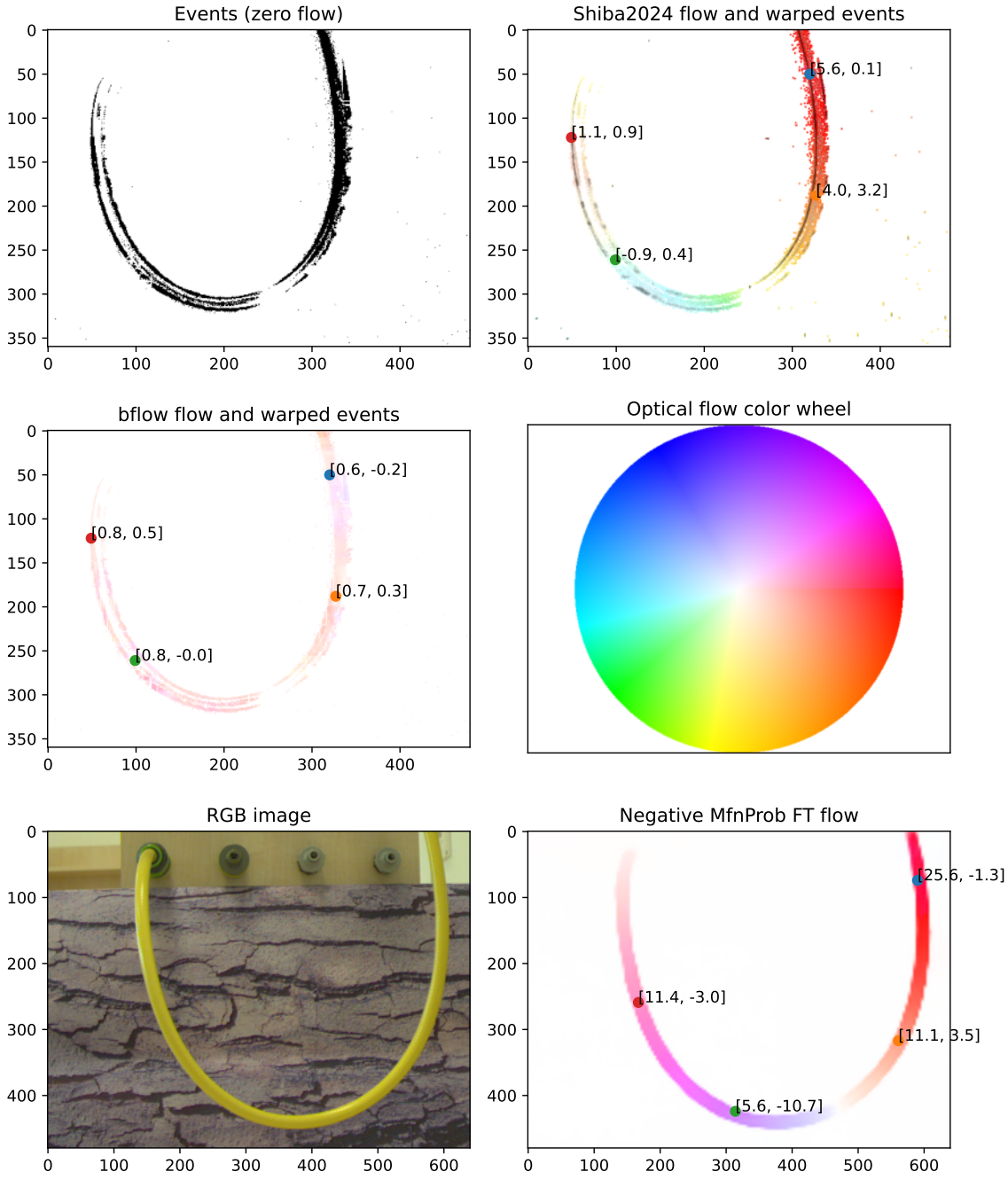


Figure 8.4: One hose moving to the right. The input was an event batch 0.0770 seconds long containing 15000 events.

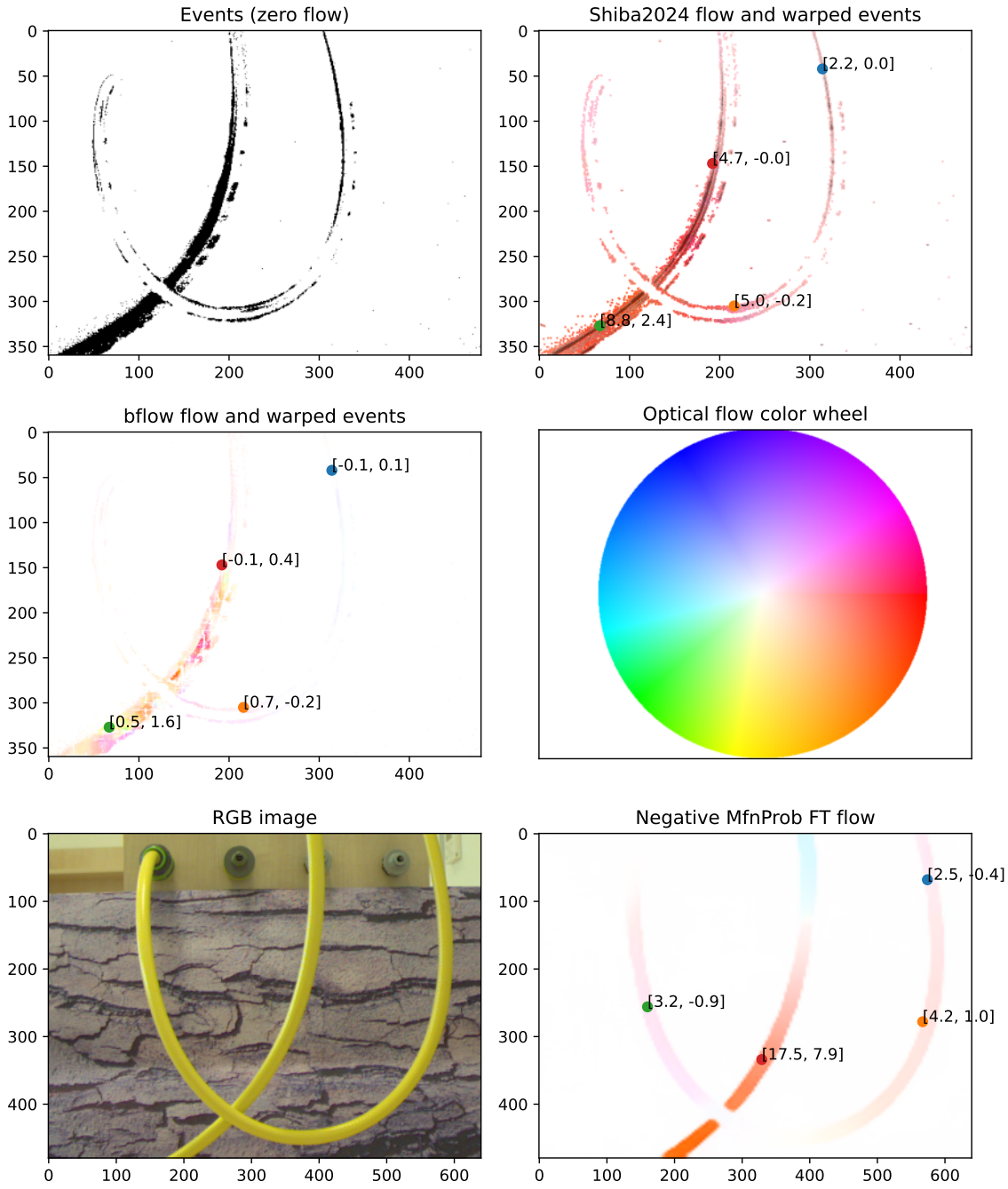


Figure 8.5: Two crossed moving hoses, one of them is moving faster than the other. The input was an event batch 0.0539 seconds long containing 15000 events.

Similarly to the rolling rod experiments, bflow estimates inaccurate flow in both hose experiments. [Shiba et al., 2024] estimates mostly accurate flow only for the right half of the U-shaped hose in Fig. 8.4. The warped events of the left half are not sufficiently motion-compensated and therefore blurry, likely due to too small predicted flow magnitude and incorrect flow direction. We think that the true flow magnitude of the left half is approximately

two times smaller than the flow of the right half, similarly to the flow estimated by MfnProb FT. [Shiba et al., 2024] predicted mostly correct flow for the diagonal hose in Fig. 8.5. It seems, however, that this flow propagated also to the other slower moving hose, causing there inaccurate flow estimates and making the two hoses almost indistinguishable from each other based on the predicted flow alone.

8.3 Discussion and conclusions

As the [Shiba et al., 2024] optical flow method encourages flow smoothness, it does not predict sharp motion boundaries between two overlapping differently moving hoses. Furthermore, the flow smoothness regularization sometimes seems to distribute the motion dominant in the scene onto neighboring objects, preventing their individual motion segmentation. The flow magnitude differences between rods or hoses rolling in the same direction at different speeds tend to disappear. At the same time, we have found that [Shiba et al., 2024] can predict the motion of a single rigid moving object (rolling rod) fairly accurately. [Shiba et al., 2024] is a non-realtime method. Estimating flow for 30000 events (around 0.015 seconds interval) in an 480×360 pixel image takes 30 seconds on an NVIDIA GeForce RTX 2080 Ti and Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz.

We have found that the optical flow estimated by the bflow [Gehrig et al., 2024] method is surprisingly inaccurate on rolling rods and moving hoses. The estimated flow is typically smaller than the true one and has significant local variability even on rigid moving objects. Although bflow is two orders of magnitude faster than [Shiba et al., 2024], it is still a non-realtime method. Estimating flow for 30000 events (around 0.015 seconds interval, 2 million events per second event rate) in an 480×360 pixel image takes 0.14 seconds on an NVIDIA GeForce RTX 2080 Ti and Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz.

The flow predicted by both bflow and [Shiba et al., 2024] is typically non-zero in the whole image, which means that static background (noise) events cannot be removed based on their flow magnitude alone. However, we note that [Shiba et al., 2024] proposed an event denoising approach based on thresholding IWEs, which could be used for background-foreground segmentation.

To conclude the observations presented above, we think that none of the two event-based optical flow estimators we tested is suitable for moving cable motion estimation or segmentation.

Chapter 9

Thesis Conclusions

This thesis proposed and evaluated several methods for segmenting a cable in cable clutter by deliberately moving the cable of interest and detecting the cable motion in a recorded image sequence. This chapter summarizes the main contributions of the thesis.

Chapter 4 proposed a method to automatically annotate a real-world moving cable segmentation dataset with optical flow and segmentation masks thanks to UV fluorescent markers, controlled lighting, and chroma keying. Using the method, we have created the MovingCables dataset consisting of 312 video clips. The clips differ in their backgrounds, cable colors, numbers of overlaid cables, motion interaction types, or distinct combinations of cable configurations.

In Chapter 5, we have tested MaskFlownet, GMFlow, and FlowFormer++ off-the-shelf optical flow neural networks on our dataset and found that they can segment moving cables from a static background. We added uncertainty outputs to the MaskFlownet architecture and retrained it with a probabilistic loss function on standard optical flow datasets. This retrained MfnProb network has significantly improved the cable motion segmentation performance over MaskFlownet on our dataset. Fine-tuning MaskFlownet and MfnProb on MovingCables further improved the accuracy.

Chapter 6 proposed a *motion correlation* method which, unlike the motion segmentation methods from Chapter 5, is able to segment a grasped moving cable even when the robot or the cable sometimes perturbs neighboring cables and even when robot arm segmentation masks are not available. It exploits the observation that gripper motion tends to correlate with grasped cable motion estimated in a common image frame by an optical flow estimator. We have evaluated the *motion correlation* method and the *motion segmentation* baseline on data recorded with our physical robotic setup.

Chapter 6 also proposed an algorithm for sampling new grasps given a partial cable segmentation. Our evaluation results showed that merging segmentation results from two grasps increases cable segmentation recall while preserving precision.

Chapter 7 experimentally compared two event cameras to two global shutter cameras. Event cameras' performance was limited by pixel latency when tracking small objects and by readout bandwidth when recognizing larger objects. When comparing the event cameras to frame cameras, we saw analogies between event pixel latency and exposure time and event readout bandwidth and frame rate. In our experiments, the event camera surpassed the fast frame camera in terms of information coding efficiency in scenes with significant changes restricted

to less than 30% of the field of view within the sampling period of interest. As the event pixel latency is significantly lower for negative than for positive contrast changes, the fastest scene changes should ideally be restricted to the negative contrast. Highly cluttered (changing) scenes or scenes with sharp and strong positive contrast edges can be detrimental to event camera performance. Surprisingly, the event camera spatial sampling density monotonically decreases with growing motion speed. This fact may limit the applicability of existing event-based algorithms relying on fixed-sized event batches. Frame and event cameras achieve similar moving object position estimation accuracy.

Chapter 8 tested two event camera optical flow estimators ([Shiba et al., 2024] and [Gehrig et al., 2024]). It found that none of them is suitable for moving cable motion estimation or segmentation. The methods struggle with predicting sharp motion boundaries between two differently moving objects. Sometimes they predict almost the same optical flow for two differently moving objects which are close to each other, especially when the objects move in similar directions at different speeds. [Shiba et al., 2024] estimated the optical flow on our data (moving rods or hoses) more accurately than [Gehrig et al., 2024], especially when only one object was moving in the scene. The flow predicted by both estimators is typically non-zero in the whole image, which means that static background (noise) events cannot be removed based on their flow magnitude alone. However, we note that [Shiba et al., 2024] proposed an event de-noising approach based on thresholding images of warped events (IWEs), which could be used for background-foreground segmentation.

Manipulating cluttered cables, hoses or ropes is challenging for both robots and humans. This thesis contributed to solving the problem of segmenting a cable in cable clutter which is hard to solve without interactive perception. First, we approached the problem by creating the MovingCables dataset of cable image sequences automatically annotated with optical flow and cable instance segmentation ground truth. Second, we designed motion segmentation methods able to segment a single moving cable from a static background and tested them on the MovingCables dataset. Third, to address the more general task of segmenting a grasped moving cable even when the robot or the cable sometimes perturbs neighboring cables, we proposed the *motion correlation* method which integrates visual and proprioceptive perception. *Motion correlation* outperformed the *motion segmentation* baseline on real-world robotic data we recorded. Being motivated by the fact that neuromorphic event cameras are more energy and data efficient than frame cameras when capturing sparsely changing scenes such as those with moving cables, we tried segmenting moving cables with event cameras. We started by comparing event cameras to frame cameras to better understand their strengths and weaknesses. Finally, we tested two event camera optical flow estimators on moving hoses and rolling rods. Unfortunately, we found that none of them is sufficiently accurate for moving cable motion estimation or segmentation.

9.1 Limitations

We have found that all the neural networks for estimating optical flow from image pairs struggle with texture-free backgrounds, where they predict non-zero optical flow even though the background is static.

Gripper-cable motion correlation does not help in the cable exploration task when multiple

cables tightly interact with each other during the gripper motion. If that happens, the robot needs to try a different action which would avoid the tight interactions with neighboring cables. Our approach uses two different actions but a robot may need a more diverse set of actions to successfully explore more complex cable configurations. When no action of a single gripper can avoid the tight interactions, one may need to use two robotic arms to verify cable segment connectivity by forceful interactions, e.g. by pulling two cable segments apart.

At the same time, we have found that the robot could not reach the majority of the proposed grasps. We attribute this to the fact that the Franka Emika Panda robot we used is fairly bulky, especially its gripper is quite wide. We think that a robot with a thinner body and gripper would be more suitable for exploring cables than the Franka Emika Panda.

9.2 Ideas for future work

- Improve state-of-the-art frame-based optical flow predictors so that they do not get confused by texture-free backgrounds.
- Use more than two different actions when moving a grasped cable to segment it by motion correlation.
- Design an algorithm able to find the optimal gripper motion to segment the grasped cable most accurately or efficiently.
- Integrate gripper interaction force measurements into the interactive cable segmentation system to detect undesirable tight interactions between neighboring cables and the grasped cable or the robot.
- It may happen that no cable moving action can avoid tight interactions with other cables. To address that scenario, extend the robotic setup to two robotic arms to verify cable segment connectivity by forceful interactions, e.g. by pulling two cable segments apart.
- Use cable motion segmentation to create automatically labeled image datasets for training passive cable segmentation methods.
- Utilize the interactive cable exploration method in downstream manipulation tasks such as cable untangling or replacement.
- Research event/frame-camera performance in high dynamic range scenes.
- Use event-cameras in robotic perception tasks requiring fast feedback loops.

Appendix A

Author's Publications

Citation counts are from Google Scholar without auto-citations.

A.1 Publications related to the thesis

A.1.1 Impacted Journal Articles

- Holešovský, O., Škoviera, R., and Hlaváč, V. (submitted 2025). Interactive Robotic Moving Cable Segmentation by Motion Correlation. IEEE Robotics and Automation Letters. [Holešovský et al., 2025] [**70%**-20%-10%]
- Holešovský, O., Škoviera, R., and Hlaváč, V. (2024). MovingCables: Moving Cable Segmentation Method and Dataset. IEEE Robotics and Automation Letters, 9(8):6991–6998. [Holešovský et al., 2024] [**60%**-30%-10%]
- Holešovský, O., Škoviera, R., Hlaváč, V., and Vítek, R. (2021). Experimental Comparison between Event and Global Shutter Cameras. Sensors, 21(4):1137. [Holešovský et al., 2021] [**40%**-20%-20%-20%] (26 citations)

A.1.2 Other conference publications

- Holešovský, O., Škoviera, R., Hlaváč, V., and Vítek, R. (2020). Practical high-speed motion sensing: event cameras vs. global shutter. In Computer Vision Winter Workshop 2020. [Holešovský et al., 2020] [**40%**-20%-20%-20%] (3 citations)

A.2 Publications not related to the thesis

A.2.1 Other conference publications

- Holešovský, O., Maki, A. (2018). Compact ConvNets with Ternary Weights and Binary Activations. In Computer Vision Winter Workshop 2018. [**60%**-40%] (1 citation)

Appendix B

Bibliography

[Enc,] Amt10 series datasheet - modular | incremental | cui devices. https://web.archive.org/web/20201223104425/https://www.cuidevices.com/product/resource/amt10.pdf. Accessed: 2020-12-20.

[Bal,] Ballistic doppler radar drs-01. https://web.archive.org/web/20200129204018/http://www.prototypa.com/drs-1-doppler-radar-system-1. Accessed: 2020-12-21.

[Cam, a] Camera basler aca640-750um. https://web.archive.org/web/20201223101740/https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca640-750um/. Accessed: 2020-12-20.

[Cam, b] Camera basler daa2500-14um. https://web.archive.org/web/20201223101938/https://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/daa2500-14um-cs-mount/. Accessed: 2020-12-20.

[Cam, c] Camera dvs240, specification sheet. https://web.archive.org/web/20201223102302/https://inivation.com/wp-content/uploads/2020/04/DVS240.pdf. Accessed: 2020-12-20.

[Cam, d] Camera photron fastcam sa-z technical sheet. https://web.archive.org/web/20180516140756/http://photron.com/wp-content/uploads/2016/11/SA-Z-REV16.10.27_LowRes.pdf. Accessed: 2020-12-20.

[Ver,] Constellation 120. https://web.archive.org/web/20201223110421/https://veritaslight.com/docs/constellation120_spec_sheet.pdf. Accessed: 2020-12-21.

[Ded,] Dedocool. https://web.archive.org/web/20201223110637/https://www.dedoweigertfilm.de/dwf-en/brands/dedolight_overview.php. Accessed: 2020-12-21.

[Odr, a] Dual shaft motor - d5065 270kv — odrive. https://web.archive.org/web/20201223103239/https://odriverobotics.com/shop/odrive-custom-motor-d5065. Accessed: 2020-12-20.

[Cam, e] Event-based evaluation kit atis hvga gen3. https://web.archive.org/web/20201022114024/https://www.prophesee.ai/event-based-evk/. Accessed: 2020-12-20.

[Fom,] Fomei led wifi-36d, panel light. https://web.archive.org/web/20201223103951/https://www.fomei.com/en/products-fomei-led-wifi-36d-panel-light-detail-239983?tabs=Technical+specification. Accessed: 2020-12-20.

[Cod,] High speed and high dynamic range video with an event camera. https://github.com/uzh-rpg/rpg_e2vid. Accessed: 2020-12-20.

[Lig,] Light screen kislter type 2521a. https://web.archive.org/web/20201223105157/https://www.kistler.com/files/download/400-336e.pdf. Accessed: 2020-12-21.

[Odr, b] Odrive v3.6 — odrive. https://web.archive.org/web/20201223103437/https://odriverobotics.com/shop/odrive-v36. Accessed: 2020-12-20.

[Aru,] Opencv, detection of aruco markers. https://web.archive.org/web/20200721095314/https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html. Accessed: 2020-12-20.

[Tri,] Ptu-1 programmable trigger unit. https://web.archive.org/web/20201223105242/http://www.prototypa.com/ptu-1-programmable-trigger-unit-1. Accessed: 2020-12-21.

[Sek,] Sekonic l-858d speedmaster light meter. https://web.archive.org/web/20201223104206/https://www.sekonic.com/ca/intl/exposure-meters/l858d. Accessed: 2020-12-20.

[STM,] Stm32f103c8 - mainstream performance line, arm cortex-m3 mcu with 64 kbytes of flash memory, 72 mhz cpu, motor control, usb and can - stmicroelectronics. https://web.archive.org/web/20201223104554/https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html. Accessed: 2020-12-20.

[DAR, 2020] (2020). Darpa, call for proposals, fast event-based neuromorphic camera and electronics (fence). https://tinyurl.com/yd824vak. Accessed: 2020-12-22.

[Abdelhamed et al., 2018] Abdelhamed, A., Lin, S., and Brown, M. S. (2018). A High-Quality Denoising Dataset for Smartphone Cameras. In *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. IEEE.

[Baker et al., 2010] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2010). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

[Barrios-Avilés et al., 2018] Barrios-Avilés, J., Iakymchuk, T., Samaniego, J., Medus, L., and Rosado-Muñoz, A. (2018). Movement Detection with Event-Based Cameras: Comparison with Frame-Based Cameras in Robot Object Tracking Using Powerlink Communication. *Electronics*, 7(11):304.

[Boerdijk et al., 2020] Boerdijk, W., Sundermeyer, M., Durner, M., and Triebel, R. (2020). Self-Supervised Object-in-Gripper Segmentation from Robotic Motions. In *4th Conf. on Robot Learning (CoRL 2020)*.

[Boettiger, 2020] Boettiger, J. P. (2020). A comparative evaluation of the detection and tracking capability between novel event-based and conventional frame-based sensors. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, USA.

[Bohg et al., 2017] Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., and Sukhatme, G. S. (2017). Interactive Perception: Leveraging Action in Perception and Perception in Action. *IEEE Trans. Robot.*, 33(6):1273–1291.

[Brandli et al., 2014] Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014). A 240 X 180 130 dB 3 μs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.

[Butler et al., 2012] Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *Computer Vision – ECCV 2012*, pages 611–625. Springer Berlin Heidelberg.

[Caporali et al., 2023a] Caporali, A., Galassi, K., and Palli, G. (2023a). Deformable Linear Objects 3D Shape Estimation and Tracking From Multiple 2D Views. *IEEE Robot. Autom. Lett.*, 8(6):3852–3859.

[Caporali et al., 2024] Caporali, A., Galassi, K., and Palli, G. (2024). Dlo perceiver: Grounding large language model for deformable linear objects perception. *IEEE Robotics and Automation Letters*, 9(12):11385–11392.

[Caporali et al., 2023b] Caporali, A., Galassi, K., Žagar, B. L., Zanella, R., Palli, G., and Knoll, A. C. (2023b). RT-DLO: Real-time deformable linear objects instance segmentation. *IEEE Trans. Ind. Informat.*, 19(11):11333–11342.

[Caporali et al., 2022a] Caporali, A., Galassi, K., Zanella, R., and Palli, G. (2022a). FASTDLO: Fast Deformable Linear Objects Instance Segmentation. *IEEE Robot. Autom. Lett.*, 7(4):9075–9082.

[Caporali et al., 2023c] Caporali, A., Pantano, M., Janisch, L., Regulin, D., Palli, G., and Lee, D. (2023c). A Weakly Supervised Semi-Automatic Image Labeling Approach for Deformable Linear Objects. *IEEE Robot. Autom. Lett.*, 8(2):1013–1020.

[Caporali et al., 2022b] Caporali, A., Zanella, R., Greogrio, D. D., and Palli, G. (2022b). Ariadne+: Deep Learning–Based Augmented Framework for the Instance Segmentation of Wires. *IEEE Trans. Ind. Informat.*, 18(12):8607–8617.

[Censi et al., 2015] Censi, A., Mueller, E., Frazzoli, E., and Soatto, S. (2015). A Power-Performance Approach to Comparing Sensor Families, with application to comparing neuromorphic to traditional vision sensors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3319–3326.

[Chi and Berenson, 2019] Chi, C. and Berenson, D. (2019). Occlusion-robust Deformable Object Tracking without Physics Simulation. In *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Choi et al., 2023] Choi, A., Tong, D., Park, B., Terzopoulos, D., Joo, J., and Jawed, M. K. (2023). mBEST: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals. *IEEE Robot. Autom. Lett.*, 8(8):4863–4870.

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

[Cox et al., 2020] Cox, J., Ashok, A., and Morley, N. (2020). An analysis framework for event-based sensor performance. In *Unconventional Imaging and Adaptive Optics 2020*, volume 11508, page 115080R. International Society for Optics and Photonics.

[Delbruck et al., 2020] Delbruck, T., Hu, Y., and He, Z. (2020). V2E: From video frames to realistic DVS event camera streams. *arxiv*.

[Delbruck and Lang, 2013] Delbruck, T. and Lang, M. (2013). Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor. *Frontiers in Neuroscience*, 7:223.

[Dorn et al., 2018] Dorn, C., Dasari, S., Yang, Y., Farrar, C., Kenyon, G., Welch, P., and Mascareñas, D. (2018). Efficient full-field vibration measurements and operational modal analysis using neuromorphic event-based imaging. *Journal of Engineering Mechanics*, 144(7).

[Dosovitskiy et al., 2015] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *2015 IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE.

[Eitel et al., 2019] Eitel, A., Hauff, N., and Burgard, W. (2019). Self-supervised Transfer Learning for Instance Segmentation through Physical Interaction. In *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Fang et al., 2024] Fang, X., Kaelbling, L. P., and Lozano-Pérez, T. (2024). Embodied Uncertainty-Aware Object Segmentation. In *2024 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Farnebäck, 2003] Farnebäck, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In *Image Analysis*, pages 363–370. Springer Berlin Heidelberg.

[Gallego et al., 2019] Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D. (2019). Event-based vision: A survey. *CoRR*, abs/1904.08405.

[Gallego et al., 2018] Gallego, G., Rebecq, H., and Scaramuzza, D. (2018). A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3867–3876. IEEE.

[Garrido-Jurado et al., 2014] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion. *Pattern Recognition*, 47(6):2280 – 2292.

[Gast and Roth, 2018] Gast, J. and Roth, S. (2018). Lightweight probabilistic deep networks. In *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. IEEE.

[Gehrig et al., 2021] Gehrig, M., Millhausler, M., Gehrig, D., and Scaramuzza, D. (2021). E-RAFT: Dense Optical Flow from Event Cameras. In *2021 International Conference on 3D Vision (3DV)*, pages 197–206. IEEE.

[Gehrig et al., 2024] Gehrig, M., Muglikar, M., and Scaramuzza, D. (2024). Dense Continuous-Time Optical Flow From Event Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4736–4746.

[Geiger et al., 2013] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Guo et al., 2007] Guo, X., Qi, X., and Harris, J. G. (2007). A Time-to-First-Spike CMOS Image Sensor. *IEEE Sensors Journal*, 7(8):1165–1175.

[Hodaň et al., 2024] Hodaň, T., Sundermeyer, M., Labbé, Y., Nguyen, V. N., Wang, G., Brachmann, E., Drost, B., Lepetit, V., Rother, C., and Matas, J. (2024). BOP challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[Holešovský et al., 2024] Holešovský, O., Škoviera, R., and Hlaváč, V. (2024). MovingCables: Moving Cable Segmentation Method and Dataset. *IEEE Robotics and Automation Letters*, 9(8):6991–6998.

[Holešovský et al., 2025] Holešovský, O., Škoviera, R., and Hlaváč, V. (submitted 2025). Interactive Robotic Moving Cable Segmentation by Motion Correlation. *IEEE Robotics and Automation Letters*.

[Holešovský et al., 2020] Holešovský, O., Škoviera, R., Hlaváč, V., and Vítek, R. (2020). Practical high-speed motion sensing: event cameras vs. global shutter. In *Computer Vision Winter Workshop 2020*.

[Holešovský et al., 2021] Holešovský, O., Škoviera, R., Hlaváč, V., and Vítek, R. (2021). Experimental Comparison between Event and Global Shutter Cameras. *Sensors*, 21(4):1137.

[Howell et al., 2020] Howell, J., Hammarton, T. C., Altmann, Y., and Jimenez, M. (2020). High-speed particle detection and tracking in microfluidic devices using event-based sensing. *Lab Chip*, 20:3024–3035.

[Johnson, 1958] Johnson, J. (1958). Analysis of image forming systems. In *Image Intensifier Symposium*, volume AD 220160, page 244–273, Ft. Belvoir, Va., USA. Warfare Electrical Engineering Department, U.S. Army Research and Development Laboratories.

[Keipour et al., 2022] Keipour, A., Bandari, M., and Schaal, S. (2022). Deformable One-Dimensional Object Detection for Routing and Manipulation. *IEEE Robot. Autom. Lett.*, 7(2):4329–4336.

[Kenney et al., 2009] Kenney, J., Buckley, T., and Brock, O. (2009). Interactive segmentation for manipulation in unstructured environments. In *2009 IEEE Int. Conf. on Robotics and Automation*. IEEE.

[Kondermann et al., 2016] Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Gussefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., and Jahne, B. (2016). The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *2016 IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE.

[Kondermann et al., 2015] Kondermann, D., Nair, R., Meister, S., Mischler, W., Güssefeld, B., Honauer, K., Hofmann, S., Brenner, C., and Jähne, B. (2015). Stereo ground truth with error bars. In *Computer Vision – ACCV 2014*, pages 595–610. Springer International Publishing.

[Kousha et al., 2022] Kousha, S., Maleky, A., Brown, M. S., and Brubaker, M. A. (2022). Modeling sRGB Camera Noise with Normalizing Flows. In *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Lai et al., 2020] Lai, Z., Alzugaray, I., Chli, M., and Chatzi, E. (2020). Full-field structural monitoring using event cameras and physics-informed sparse identification. *Mechanical Systems and Signal Processing*, 145:106905.

[Lee et al., 2019] Lee, S., Kim, H., and Kim, H. J. (2019). Edge Detection for Event Cameras using Intra-pixel-area Events. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 181. BMVA Press.

[Lichtsteiner et al., 2008] Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576.

[Liu and Delbruck, 2018] Liu, M. and Delbruck, T. (2018). Adaptive Time-Slice Block-Matching Optical Flow Algorithm for Dynamic Vision Sensors. In *BMVC 2018*.

[Lu et al., 2023] Lu, Y., Khargonkar, N., Xu, Z., Averill, C., Palanisamy, K., Hang, K., Guo, Y., Ruozzi, N., and Xiang, Y. (2023). Self-Supervised Unseen Object Instance Segmentation via Long-Term Robot Interaction. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation.

[Luo et al., 2023] Luo, X., Luo, K., Luo, A., Wang, Z., Tan, P., and Liu, S. (2023). Learning optical flow from event camera with rendered dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9847–9857.

[Lv et al., 2023] Lv, K., Yu, M., Pu, Y., Jiang, X., Huang, G., and Li, X. (2023). Learning to Estimate 3-D States of Deformable Linear Objects from Single-Frame Occluded Point Clouds. In *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE.

[Mahowald, 1992] Mahowald, M. (1992). *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, California Institute of Technology Pasadena.

[Manin et al., 2018] Manin, J., Skeen, S. A., and Pickett, L. M. (2018). Performance comparison of state-of-the-art high-speed video cameras for scientific applications. *Optical Engineering*, 57(12):1–14.

[Mayer et al., 2016] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Mueggler et al., 2015] Mueggler, E., Forster, C., Baumli, N., Gallego, G., and Scaramuzza, D. (2015). Lifetime estimation of events from Dynamic Vision Sensors. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):4874–4881.

[NI et al., 2012] NI, Z., PACORET, C., BENOSMAN, R., IENG, S., and RÉGNIER*, S. (2012). Asynchronous event-based high speed vision for microparticle tracking. *Journal of Microscopy*, 245(3):236–244.

[Paredes-Vallés et al., 2023] Paredes-Vallés, F., Scheper, K. Y. W., De Wagter, C., and De Croon, G. C. H. E. (2023). Taming Contrast Maximization for Learning Sequential, Low-latency, Event-based Optical Flow. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9661–9671. IEEE.

[Patten et al., 2018] Patten, T., Zillich, M., and Vincze, M. (2018). Action Selection for Interactive Object Segmentation in Clutter. In *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Posch et al., 2011] Posch, C., Matolin, D., and Wohlgenannt, R. (2011). A QVGA 143 dB Dynamic Range Frame-free PWM Image Sensor with Lossless Pixel-level Video Compression and Time-Domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275.

[Price et al., 2021] Price, A., Huang, K., and Berenson, D. (2021). Fusing RGBD Tracking and Segmentation Tree Sampling for Multi-Hypothesis Volumetric Segmentation. In *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE.

[Qian et al., 2024] Qian, H. H., Lu, Y., Ren, K., Wang, G., Khargonkar, N., Xiang, Y., and Hang, K. (2024). RISeg: Robot Interactive Object Segmentation via Body Frame-Invariant Features. In *2024 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 13954–13960. IEEE.

[Ravi et al., 2024] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. (2024). SAM 2: Segment Anything in Images and Videos. *arXiv*.

[Rebecq et al., 2018] Rebecq, H., Gehrig, D., and Scaramuzza, D. (2018). Esim: an open event camera simulator. In Billard, A., Dragan, A., Peters, J., and Morimoto, J., editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 969–982. PMLR.

[Rebecq et al., 2019a] Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D. (2019a). Events-to-Video: Bringing Modern Computer Vision to Event Cameras. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*.

[Rebecq et al., 2019b] Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D. (2019b). High Speed and High Dynamic Range Video with an Event Camera. *arXiv e-prints*.

[Scheerlinck et al., 2020] Scheerlinck, C., Rebecq, H., Gehrig, D., Barnes, N., Mahony, R., and Scaramuzza, D. (2020). Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

[Shi et al., 2023] Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K. C., See, S., Qin, H., Dai, J., and Li, H. (2023). FlowFormer++: Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Shiba et al., 2024] Shiba, S., Klose, Y., Aoki, Y., and Gallego, G. (2024). Secrets of Event-based Optical Flow, Depth and Ego-motion Estimation by Contrast Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18.

[Shivakumar et al., 2023] Shivakumar, K., Viswanath, V., Gu, A., Avigal, Y., Kerr, J., Ichnowski, J., Cheng, R., Kollar, T., and Goldberg, K. (2023). SGTM 2.0: Autonomously untangling long cables using interactive perception. In *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE.

[Singh et al., 2021] Singh, C. D., Sanket, N. J., Parameshwara, C. M., Fermuller, C., and Aloimonos, Y. (2021). NudgeSeg: Zero-Shot Object Segmentation by Repeated Physical Interaction. In *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Stoffregen et al., 2020] Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., and Mahony, R. (2020). Reducing the sim-to-real gap for event cameras. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 534–549, Cham. Springer International Publishing.

[Stoiber et al., 2022] Stoiber, M., Sundermeyer, M., Boerdijk, W., and Triebel, R. (2022). A multi-body tracking framework - from rigid objects to kinematic structures.

[Sundaresan et al., 2021] Sundaresan, P., Grannen, J., Thananjeyan, B., Balakrishna, A., Ichnowski, J., Novoseller, E., Hwang, M., Laskey, M., Gonzalez, J., and Goldberg, K. (2021). Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation.

[Szeliski, 2011] Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer London.

[Takahashi and Yonekura, 2020] Takahashi, K. and Yonekura, K. (2020). Invisible marker: Automatic annotation of segmentation masks for object manipulation. In *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Thananjeyan et al., 2022] Thananjeyan, B., Kerr, J., Huang, H., Gonzalez, J. E., and Goldberg, K. (2022). All You Need is LUV: Unsupervised Collection of Labeled Images Using UV-Fluorescent Markings. In *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Tsikos and Bajcsy, 1991] Tsikos, C. J. and Bajcsy, R. K. (1991). Segmentation via manipulation. *IEEE Trans. Robot. Autom.*, 7(3):306–319.

[Viswanath et al., 2021] Viswanath, V., Grannen, J., Sundaresan, P., Thananjeyan, B., Balakrishna, A., Novoseller, E., Ichnowski, J., Laskey, M., Gonzalez, J. E., and Goldberg, K. (2021). Disentangling Dense Multi-Cable Knots. In *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Wang et al., 2021] Wang, Y., McConachie, D., and Berenson, D. (2021). Tracking Partially-Occluded Deformable Objects while Enforcing Geometric Constraints. In *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE.

[Wnuk et al., 2020] Wnuk, M., Hinze, C., Lechler, A., and Verl, A. (2020). Kinematic Multibody Model Generation of Deformable Linear Objects from Point Clouds. In *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE.

[Xiang et al., 2023] Xiang, J., Dinkel, H., Zhao, H., Gao, N., Coltin, B., Smith, T., and Bretl, T. (2023). TrackDLO: Tracking Deformable Linear Objects Under Occlusion With Motion Coherence. *IEEE Robot. Autom. Lett.*, 8(10):6179–6186.

[Xu et al., 2023] Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., and Geiger, A. (2023). Unifying Flow, Stereo and Depth Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958.

[Yan et al., 2020] Yan, M., Zhu, Y., Jin, N., and Bohg, J. (2020). Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects. *IEEE Robot. Autom. Lett.*, 5(2):2372–2379.

[Zanella et al., 2021] Zanella, R., Caporali, A., Tadaka, K., Gregorio, D. D., and Palli, G. (2021). Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence. In *2021 Int. Conf. on Computer, Control and Robotics (ICCCR)*. IEEE.

[Zhang et al., 2023] Zhang, X., Domae, Y., Wan, W., and Harada, K. (2023). Learning Efficient Policies for Picking Entangled Wire Harnesses: An Approach to Industrial Bin Picking. *IEEE Robot. Autom. Lett.*, 8(1):73–80.

[Zhao et al., 2020] Zhao, S., Sheng, Y., Dong, Y., Chang, E., and Xu, Y. (2020). MaskFlownet: Asymmetric Feature Matching with Learnable Occlusion Mask. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Zhaole et al., 2024] Zhaole, S., Zhou, H., Nanbo, L., Chen, L., Zhu, J., and Fisher, R. B. (2024). A Robust Deformable Linear Object Perception Pipeline in 3D: From Segmentation to Reconstruction. *IEEE Robotics and Automation Letters*, 9(1):843–850.

[Zhu et al., 2018a] Zhu, A., Yuan, L., Chaney, K., and Daniilidis, K. (2018a). EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *Robotics: Science and Systems XIV*, RSS2018. Robotics: Science and Systems Foundation.

[Zhu et al., 2018b] Zhu, A. Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., and Daniilidis, K. (2018b). The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039.

[Zhu et al., 2019] Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2019). Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Zhu et al., 2020] Zhu, J., Navarro, B., Passama, R., Fraisse, P., Crosnier, A., and Cherubini, A. (2020). Robotic manipulation planning for shaping deformable linear objects with environmental contacts. *IEEE Robot. Autom. Lett.*, 5(1):16–23.

[Zhu et al., 2021] Zhu, J., Navarro-Alarcon, D., Passama, R., and Cherubini, A. (2021). Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours. *Robotics and Autonomous Systems*, 142:103798.