LEARNING AND EXPLOITING TEMPORAL DEPENDENCIES IN THE SYNTHESIS AND ANALYSIS OF VIDEO SIGNALS

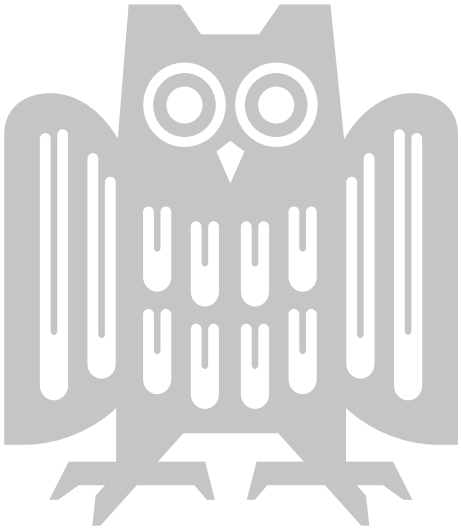
GEREON FOX

Dissertation zur Erlangung des Grades des

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken, 2025



Date of Colloquium:	August 26, 2025
Dean of the Faculty:	Prof. Dr. Roland Speicher
Chair of the Committee:	Prof. Dr. Martina Maggio
Reviewers:	Prof. Dr. Christian Theobalt
	Prof. Dr. Thorsten Herfet
Academic Assistant:	Dr. Lin Geng Foo

Für Papa, der mich auf die Computerei gebracht hat.

Und Opa, von dem ich das Basteln geerbt habe.

Abstract

The acquisition, reproduction, analysis and modification of visual information are important in all parts of human life – even more so since the advent of sufficiently capable computers. Especially the computational treatment of the temporal dimension is challenging, but also beneficial for many applications. This thesis explores the temporal dimension in three different contexts:

For the *detection of semantically relevant manipulations*, it demonstrates that previous detection methods can be fooled by the same improvements to the manipulation technique that would fool human observers. New methods are presented to nevertheless achieve high detection accuracy, and especially temporal dependencies are shown to help generalise to unseen manipulation methods.

For the *synthesis of new video signals*, previous work has constructed models that entangle spatial and temporal features. This thesis separates these features, reducing memory demand and computation time, as well as the amount of data necessary for training.

For the *reconstruction of video signals from event data*, a data modality for which training data is scarce, the thesis presents a method to turn event data into watchable signals, without using any training data at all, but outperforming previous methods that do so.

In each of these contexts, the thesis highlights the degree to which solutions depend on training sets of different sizes, and the impact this has on performance and computational cost.

Zusammenfassung

Erfassung, Reproduktion, Analyse und Modifikation visueller Informationen sind wichtig für alle Bereiche menschlichen Lebens – insbesondere seit der Verfügbarkeit leistungsfähiger Rechner. Vor allem die Zeit-Dimension ist informatisch herausfordernd, aber auch lohnenswert für viele Anwendungen. Die vorliegende Arbeit untersucht diese Dimension in drei verschiedenen Kontexten:

Für die *Erkennung semantisch relevanter Manipulationen* wird gezeigt, dass Manipulationen, die menschliche Betrachter zuverlässig täuschen, auch die bisherigen maschinellen Erkenner in die Irre führen. Neue Erkenner werden eingeführt, denen die Modellierung zeitlicher Abhängigkeiten zu erhöhter Robustheit gegenüber ungesehenen Manipulationen verhilft.

Bei der *Synthese neuer Videosignale* haben vorherige Arbeiten räumliche und zeitliche Zusammenhänge ineinander verwoben modelliert. Die vorliegende Arbeit trennt diese Dimensionen und reduziert so Speicherbedarf, Rechenzeit und Bedarf an Trainingsdaten.

Für die *Rekonstruktion von Videosignalen aus Event-Daten* sind Trainingsdaten nur schwer zu beschaffen. Die Arbeit rekonstruiert Videosignale aus Event-Daten besser als vorherige Methoden, ohne Trainingsdaten zu benötigen.

Für alle drei Aufgaben beleuchtet die Arbeit den Bedarf an Trainings-Datensätzen verschiedener Größen, sowie den daraus resultierenden Einfluss auf Ausgabequalität und Ressourcenverbrauch.

Acknowledgements

It is unthinkable that I could have completed this thesis without the support and motivation that were supplied by a great number of friends and colleagues, knowingly and unknowingly.

First and foremost I want to thank Christian Theobalt for the spectacular opportunity given to me when I needed it, as well as for his continued support and unshakable (and at times astonishing) optimism throughout my time at MPI. I am sure that I have put his patience to the test more than once.

The work presented here would have been impossible without my closest collaborators Mohamed Elgharib, Ayush Tewari and Xingang Pan, who invested countless hours into discussing research questions and working through nights. A special thank you goes to Xingang for hosting me in Singapore, which gave me many lasting memories.

I am grateful to my other collaborators and co-authors: In particular, Wentao Liu, Jalees Nehvi, Pramod Rao and Kartik Teotia have helped me out with experiments and evaluations under immense time pressure. Viktor Rudnev not only shared some of the data he recorded, but also helped me understand event cameras.

I thank Congyi Zhang for our very interesting collaboration on teeth reconstruction. I also thank Zhening Xing and Yanhong Zeng for putting up with the difficulties we navigated together.

I thank my proofreaders Franziska Müller, Hamza Pehlivan, Kartik Teotia, Navami Kairanda and Wanyue Zhang for their helpful feedback on the manuscript. Navami kindly helped me find examples of video diffusion model outputs.

I am very grateful to all members of GVV and the departments D4+6, present and previous, that I have worked with during my time at MPI. Without all the fun we shared there would have been much less reason to get out of bed in the morning! I especially thank my office mates Ikhsanul Habibie, Marc Habermann, Heming Zhu and Hamza Pehlivan. Further thanks go to Abhimitra Meka, who was my first advisor at MPI, and Dushyant Mehta, whose being a human encyclopedia of neural network architectures was most helpful in the early days. I also thank Mohit Mendiratta for helping me help someone else.

Speaking of helping, I would like to thank all our administrative assistants, especially Ellen Fries and Sabine Budde, whose pictures should be printed in dictionaries next to the word “helpful”.

Hans-Peter Seidel I thank for giving me the most satisfying job I have

ever had, namely that of department IT administrator in MPI. I enjoyed this job so much because it is all about people, who I could usually help by handing them the resources they needed. This was possible because our IST department, who actually run and maintain all those resources day in and day out, made my life very easy by being the most service-oriented and intrinsically motivated IT staff that I have ever seen. In my opinion we should just give them the digital infrastructure of all of Germany to run, and all would be well. Mediating between users and IST, however, translating their different languages into each other, is still a time-consuming assignment, and so I am grateful to Jozef Hladký and Hyeongwoo Kim, who trained me in this, as well as Bin Chen, Pramod Rao and Felix Mujkanovic, who I had the pleasure of sharing the task with. I also thank our facilities team, who provide the bedrock of it all.

Among the community of Saarland Informatics Campus I would like to especially thank Holger Hermanns, Sebastian Hack and Gert Smolka, who taught me things that were *very* helpful in my PhD work, and that I am secretly sneaking into workflows at MPI. Furthermore I thank Michelle Carnell, Erich Reindel and Felix Freiberger, as representatives of all the amazing people that make Saarland Informatics Campus a uniquely permeable playground for crazy people like me. This environment has made it hard to even imagine a better one. Having made extensive use of it through all levels of my computer science education, I appreciate it more than most people probably can. Should anything need "burning down", call me.

Two valuable friends with which I have shared this environment are Marie Mathieu and Franziska Müller, who made fundamental contributions to keeping me sane (sort of). I thank the Hirsch family and especially Anke for sharing joy.

Last, but of course not least, I thank my sister Rebecca and my parents Elisabeth and Markus Fox, who never questioned my ability to complete this piece of work.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Vision is undoubtedly among the most important of the five human senses, because it quickly gives us very detailed information about our surroundings, even at a distance. We use vision to find out where we are, to locate objects, to navigate the world, to avoid danger, to recognise other human beings, and to interact with them. Given this importance, it is not surprising that from the very beginning of human history we can find attempts to synthesize, capture, conserve and reproduce visual impressions. Examples range from the earliest cave paintings, over devices like the camera obscura, to modern-day photography. Starting in the twentieth century, with the advent of sufficiently capable computers, the analysis and synthesis of visual data has become an important field of research, fuelling technological progress in a large number of areas, including arts and entertainment, robotics, self-driving cars, medical imaging and others.

Notably visual impressions, as a way of capturing information about the world around us, do not only comprise two or three *spatial* dimensions, but usually also a *temporal* dimension, that can contain very important information: The *presence* of a car can be determined in the spatial dimensions, but the danger it poses can only be assessed by estimating the speed at which it *moves*, which we can only perceive by observing the situation over a non-zero amount of time. In general, almost any visual impression can be made more informative by adding a temporal dimension to the spatial ones, as is obvious in many areas of everyday life. In contrast to *still* imagery, however, technology only relatively recently has allowed mankind to synthesize, capture, conserve and reproduce the *temporal* aspects of visual impressions. One of the earliest and a notable example are the experiments by Eadweard Muybridge in the nineteenth century (see Figure 1.1), in which he captured motion as a sequence of visual samples along the temporal axis.

Evolving the *capture and reproduction* of still images into recording "motion pictures" required "only" the ability to capture and reproduce still frames at a more rapid rate than before. But the *editing* of visual impres-

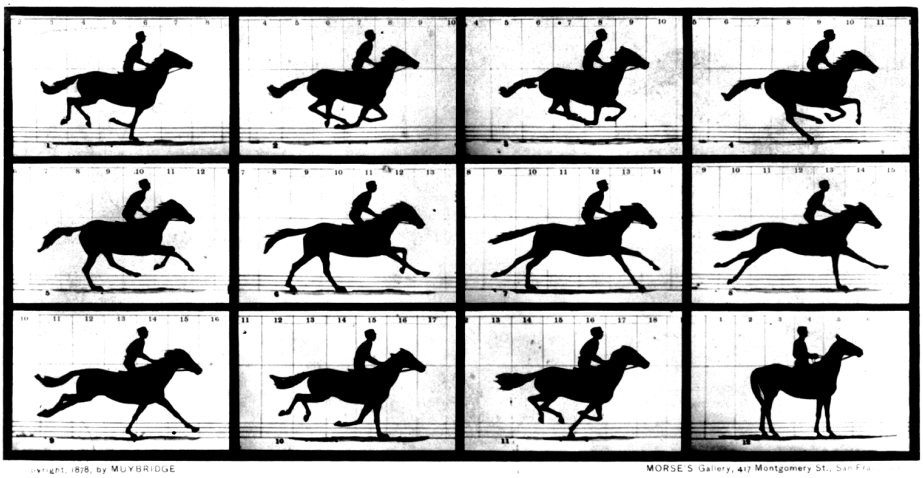


Figure 1.1: *The Horse in Motion*, captured by Eadweard Muybridge in 1878, is a series of short-exposure photographs forming one of the first "videos" ever recorded.

sions comprising a temporal aspect poses a whole new kind of challenge:

A whole sequence of images that were recorded at a fast temporal rate, i.e. a **video**, usually exhibits **temporal dependencies**: Information about one of the images is likely to admit conclusions about other images in the sequence. Viewers, based on their world knowledge, expect certain forms of these conclusions to be valid, i.e. a video that appears to violate the expected temporal dependencies would be perceived as "flawed". The same expectation, however, might lead viewers to mistake synthetic videos as an accurate depiction of the world, provided the synthesis process successfully emulates the expected dependencies.

The importance of temporal dependencies makes *editing* a video much harder than merely editing each individual frame (i.e. each image of the sequence): The editor must ensure that the modifications made to a particular frame are consistent with the content of and the modifications to at least the immediately neighbouring frames, but in general to *all* other frames in the video. This requires a new capability on the part of the editor, that was not necessary for single image editing. One obvious consequence of not fully taking this aspect into account can be observed in the film *Loving Vincent* (Kobiela et al., 2017, Figure 1.2): Its frames are the result of a manual "style transfer", where each previously photographed live action frame was turned into an oil painting by hand. While each oil painting on its own can be considered a convincing rendition of the style of Vincent van Gogh and is consistent with the original live action frame, displaying the oil paintings in rapid succession exhibits a very noticeable "flickering" effect. This phenomenon can certainly be seen as

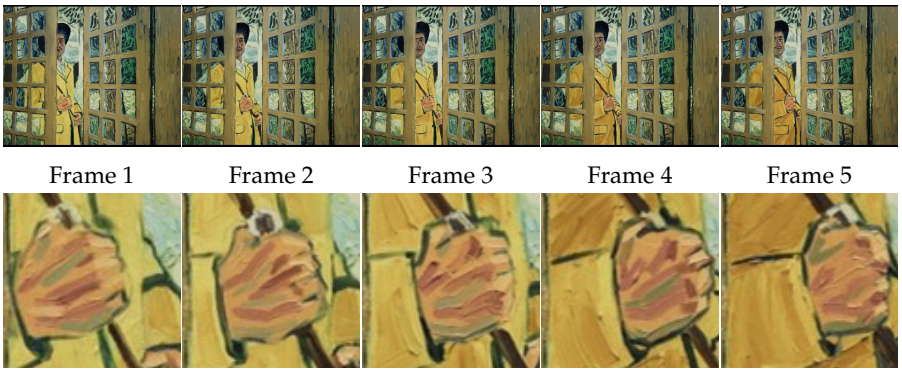


Figure 1.2: The frames of *Loving Vincent* (Kobiela et al., 2017), a 94 minute film, have been hand-painted in oil. Each painting is consistent with the underlying live action frame and convincingly imitates the style of Vincent van Gogh, but playback as a film makes the inconsistencies in the textures very apparent to the viewer: The consecutive frames in this figure (starting at about 22min45sec into the DVD version) differ strongly in the texture that was applied to the right hand of the actor.

an interesting artistic aspect of the work, but it appears plausible that the sheer number of oil paintings that had to be manufactured for 94 minutes of film did likely not even leave the option of paying meticulous attention to the temporal consistency of textures between subsequent frames.

The added difficulty of taking temporal dependencies into account carries over to the computer-based editing of videos: Especially when the goal is a photorealistic output, the complexity of temporal dependencies, arising from the physical properties of the world around us, often makes them very expensive to compute and together with their diversity across a large set of scenarios can turn consciously modelling them into a challenge beyond feasibility. Data-driven machine learning and the use of deep neural networks promise a way out of this problem, because they provide mathematical tools via which the machine can "discover" dependencies on its own, without too much human intervention. This way, not only complex, but also rather subtle dependencies can be incorporated into a model, which a human might not have realised help solve the task at hand more accurately and more efficiently than with the crudely engineered mechanisms said human could have devised in a reasonable time frame.

There are many caveats to the usage of machine learning or neural networks, in particular arising from the dependence on large amounts of training data. In this respect, too, adding the temporal dimension to two or more spatial ones multiplies the amount of data that is required, and thus the difficulty: Acquiring the data in the first place is often a very labo-

rious task, subject to legal restrictions such as the right to privacy and/or copyright. Storing, preprocessing and transmitting the acquired data tends to turn engineering challenges from "negligible" into proper budget items. Actually training a neural network on all the data not only means processing *many* videos, but often also requires the processor to operate on items that occupy large amounts of working memory, necessitating the use of state of the art hardware accelerators. Because of these considerable resource penalties in dealing with large training corpora, developers are motivated to limit the amount of data collected to the necessary minimum. This ambition can easily lead to (advertently or inadvertently) biasing the distribution of the collected data towards a particular subclass of the inputs of interest. A model trained on such biased data will likely inherit the bias, leading to lacking predictions, or (which is worse) incorrect predictions when it is applied to inputs outside the subclass. In addition, it can be challenging to properly delineate said subclass, because its nature results from conscious or unconscious decisions that the acquisition of the training data was subject to.

While the previous paragraphs mostly focused on *editing* videos, the importance of modelling temporal dependenciess also extends to other tasks: For example, even a partial failure of the editor to take temporal dependencies into account will give the viewer an opportunity to *detect* that the video was modified in a way that may alter its semantics, i.e. the conclusions to be drawn from it. The detection of manipulations is a highly desirable capability in certain settings. Just like editing it requires knowledge about temporal dependencies, leading to similar challenges if they are to be discovered in large training corpora.

Another task for which temporal dependencies are of paramount importance is the reconstruction of a denser video signal from a sparser one: Given sparse/partial observations of a signal, missing information is to be filled in to obtain a more "complete" signal. This task can be considered a hybrid of analysis and synthesis, and since the problem is under-constrained in the mathematical sense (i.e. many different ground truth signals could explain the exact same set of observations), temporal dependencies are a valuable tool in separating more likely solutions from less likely ones.

Given the importance of temporal dependencies in video signals, as well as the challenges arising from treating them computationally, this thesis aims at furthering our understanding of these dependencies. It does so by investigating temporal aspects of three different and partially related tasks surrounding video signals:

1. Detection of semantically relevant manipulations in signals

2. Synthesis of new signals

3. Reconstruction of signals from partial observations

Each of these tasks is approached with the goal of highlighting the importance of temporal dependencies, contributing new ways of capturing them computationally and demonstrating the impact of varying degrees of dependency on training data.

1.2 Overview

For the first task, the **detection of semantically relevant manipulations** (Chapter 3), this thesis focuses on face videos, i.e. recordings of a single person talking into the camera. Humans naturally use their faces to convey their presence and to accompany their speech, which is why face videos are a ubiquitous form of communication. As such, they are are also an attractive target for the malicious modification of signals, for example to give the impression that a specific person made a statement they would never voluntarily make. Manipulations like these can be achieved by transferring facial expressions or the facial appearance of a person from a source video to a target video. They can have drastic consequences in high-stakes contexts, such as during a coup d'état, in political campaigning, or in other forms of political propaganda. But also on a smaller scale, any person for which sufficient amounts of training data are available, for example on social media, can be the target of a malicious forgery of this kind, embedding their faces in disadvantageous contexts. Especially since deep neural networks have made such editing approaches very capable of achieving convincing results (H. Kim et al., 2018; Kowalski, 2018; torzdf et al., 2020), there has been increasing concern about their misuse. This prompted the research community to employ neural networks for the *detection* of manipulations as well: The task is to decide, given an input video, if this video is the result of a mere recording of a person, or if the video contains traces of modifications that make the signal semantically different from the person's physical performance in front of the camera. Prior to the work presented in Chapter 3, the community had focused on detecting manipulations that humans could anyway spot with the naked eye, because synthesis methods were not yet robust enough to create large training corpora of manipulated content without human intervention. Assuming that synthesis would improve in this respect, Chapter 3 investigates the hypothesis that manipulating videos in a way that reliably fools humans would also make automatic detectors struggle. For the purpose of this investigation, the dataset VIDEOFORENSICSHQ is presented, comprising both authentic and manipulated videos. The quality of the manipulated content in VIDEOFORENSICSHQ is unprecedented for a dataset of comparable size, making it a valuable proving ground for the investigation of said hypothesis. As a second contribution

it is then demonstrated that despite the high quality of the manipulated videos in VIDEOFORENSICSHQ, it is still very well possible to detect them, by introducing a family of neural network architectures based on suitable preprocessing of the input video. These detectors focus not only on the original colour frames, but also on spatially and temporally filtered versions. In particular the temporal component is shown to potentially improve not only the accuracy of the detector, but also its generalization to unseen methods of manipulation. Given that obtaining manipulation datasets of sufficient size is difficult and that generalization to unseen manipulation methods is an important capability in the "arms race" between attackers and defenders, these findings show how important it is for both sides to model temporal dependencies well and that neglecting them can easily lead to defeat.

After this study in *analysing* signals, Chapter 4 turns to *synthesising* them, presenting STYLEVIDEOGAN, a **generative model for face videos**. Previous work on video generation often involved training neural networks for the entangled representation of spatial and temporal components of the video. At the very least, it required rendering spatial content (i.e. colour frames) during training on temporal data. STYLEVIDEOGAN is a neural-network-based synthesis method that does not materialise actual colour frames at training time. Instead, the training data is first embedded into the parameter space of a generative model for still images, STYLEGAN (Karras et al., 2020). Since this space has much fewer dimensions than the space of colour frames, the embedding drastically reduces the amount of memory needed during training of the temporal generator and also speeds up the training process, making the discovery of more complex patterns of motion tractable. By exploiting separability properties of STYLEGAN's parameter space, STYLEVIDEOGAN also requires less training data than most previous methods, by orders of magnitude. For example, a single 10 minute training video from only one single subject is enough to synthesize long-duration motion for a large, dense set of unseen subjects. This means that in contrast to previous work, which required large, diverse training corpora, with all the difficulties that come with them, the data collection for a STYLEVIDEOGAN model can be finished within minutes. Although the results shown in Chapter 4 are mostly on human faces, the concepts introduced here are not limited to faces, as STYLEVIDEOGAN does not make any strictly face-specific assumptions.

The third task is the **reconstruction of signals from event streams** (Chapter 5): An **event camera** records not absolute brightness values for its pixels, but merely reports the times at which brightness deviates from a reference value by a sufficient margin. This is done for each pixel independently, i.e. there is no notion of "frame", that would temporally synchronise brightness measurements across different pixels. Event cameras are useful for recording fast motion under low-light conditions. Compared

to "classical" RGB cameras they can have advantages in terms of power consumption, bandwidth, and storage, because they only record sparse observations about the signal. Because of the semantics of the event model that is usually assumed, and because of the ways in which the physical camera deviates from the assumptions of that model, it is in principle not possible to fully reconstruct the input signal: For every recorded event stream there can be many different input signals that would all explain this event stream. It is thus challenging to turn a recorded event stream into a plausible RGB video signal to be watched by a human viewer. Event cameras are a relatively recent development, and, as was once the case for RGB cameras, are not manufactured in large numbers (yet). While the technology is promising for mobile applications with limited storage capacity and power supply, it is is still rather expensive and not widespread among consumers, making the acquisition of training data even more challenging than for the RGB-based tasks of the previous chapters. This is especially true if paired data (events plus ground truth signals) is needed. Nevertheless previous work has found ways to model event-based video reconstruction: On the one hand, there are methods trained on real or (partially) synthetic data, and on the other hand there are methods solely based on hand-engineered algorithms. The method presented in this thesis belongs to the second category, avoiding the pitfalls of training data bias and contributing novel solutions to dealing with noise in the event stream. The method is based on a principled application of the physical/mathematical assumptions of the event camera model. Of course the method is still "biased" in the sense that it contains certain design decisions that could have been made in a different way. But in contrast to a trained model, these design decisions can fully be described in a concise form and they do not favour one particular semantic domain, which is an advantage over previous work. In contrast to the two previous tasks, that either analyse a *completely given* signal, or synthesize a *completely new* signal, the presented solution to the third adheres to *partial* information reported from the camera.

1.3 Structure

This thesis comprises six chapters:

- Chapter 1 gives the motivation for the work presented in this thesis, provides an overview of its structure and lists the main contributions as well as the publications they have first been presented in.

- Chapter 2 defines basic notions and notations to be used throughout the thesis.

- Chapter 3 presents a way of simulating a very advanced manipulation and synthesis algorithm for face videos, in order to create a large dataset of authentic and manipulated videos. In addition, Chapter 3 introduces a family of detectors, that, when trained on the presented and previous datasets, outperform previous detection methods, in particular because of their use of temporal information.
- Chapter 4 describes a new approach to synthesizing nearly photorealistic face videos based on a prior of still images and a very small set of motion data.
- Chapter 5 describes an algorithm for the reconstruction of video signals from event streams, that does not require any training data at all and outperforms previous work in terms of reconstruction quality.
- Chapter 6 summarises important insights of this thesis and discusses avenues for future work.

1.4 Contributions

The contributions of Chapter 3, published as Fox et al., 2021a, are:

- The dataset VIDEOFORENSICSHQ, consisting of high-quality face videos, both authentic and manipulated, with a special focus on temporal consistency of the synthetic videos. Only with this dataset was it possible to evaluate the performance of state of the art forgery detection algorithms on the kind of material that would reliably fool the human observer, demonstrating that their detection performance leaves room for improvement.
- A novel family of learning-based detectors that use combinations of colour, low-level noise and temporal dependencies for the detection of forgeries. These detectors are shown to perform better than previous methods on high-quality manipulations. Especially their temporal component helps generalise to unseen manipulation methods.

The contributions of Chapter 4, published as Fox et al., 2021b, are:

- A novel approach for unconditional video generation that is supervised in the latent space of a pretrained image generator, STYLEGAN (Karras et al., 2020), without having to render frames at training time, which leads to large savings in computational resources.
- A demonstration of how the properties of STYLEGAN's $\mathcal{W}^+$ space can be used to greatly reduce the amount of training data needed to train a generative video model.

- A novel gradient angle penalty loss that helps generate videos of arbitrary duration.

The contributions of Chapter 5, published as Fox et al., 2024, are:

- A new optimization-based method to reconstruct high-frequency brightness signals that explain a stream of input events and a sequence of input frames with long exposure time. In contrast to previous methods it does not require any training data and hence avoids biases that would arise from difficulties in collecting such data.

- Per-event confidence weights, regularised by a special loss term, serve to account for noise in the event data.

- Additional signal control points in-between events help produce smooth signals.

- Bézier interpolation in-between signal control points leads to improved reconstruction accuracy.

1.5 Publications

The work documented in this thesis has been presented in the following publications:

- Gereon Fox et al. **(2021a)**: "VideoForensicsHQ: Detecting High-quality Manipulated Face Videos". In: *2021 IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5-9, 2021*. IEEE, pp. 1–6

  Supported by ERC Consolidator Grant 4DReply (770784).

  Video presentation including results under

  https://vcai.mpi-inf.mpg.de/projects/VForensicsHQ

- Gereon Fox et al. **(2021b)**: "StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN". in: *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*. BMVA Press, p. 220

  Supported by the ERC Consolidator Grant 4DReply (770784).

  Video presentation including results under

  https://vcai.mpi-inf.mpg.de/projects/stylevideogan

- Gereon Fox et al. **(2024)**: "Unsupervised Event-Based Video Reconstruction". In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, pp. 4167–4176

  Video presentation including results under

  https://vcai.mpi-inf.mpg.de/projects/colibri

Furthermore, contributions were made to the following publications, not part of this thesis:

- Congyi Zhang et al. **(2022)**: "An Implicit Parametric Morphable Dental Model". In: *ACM Trans. Graph.* 41.6, 217:1–217:13
- Pramod Rao et al. **(2022)**: "VoRF: Volumetric Relightable Faces". In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, p. 708
- Pramod Rao et al. **(2024b)**: "A Deeper Analysis of Volumetric Relightable Faces". In: *Int. J. Comput. Vis.* 132.4, pp. 1148–1166
- Pramod Rao et al. **(2024a)**: "Lite2Relight: 3D-aware Single Image Portrait Relighting". In: *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024*. Ed. by Andres Burbano et al. ACM, p. 41
- Barbod Pajoum et al. **(2024)**: "Adaptive Grids for Neural Scene Representation". In: *29th International Symposium on Vision, Modeling, and Visualization, VMV 2024, Munich, Germany, September 10-13, 2024*. Ed. by Lars Linsen et al. Eurographics Association
- Zhening Xing et al. **(2024)**: "Live2Diff: Live Stream Translation via Uni-directional Attention in Video Diffusion Models". In: *CoRR* abs/2407.08701. arXiv: 2407.08701
- Viktor Rudnev et al. **(2024)**: "Dynamic EventNeRF: Reconstructing General Dynamic Scenes from Multi-view Event Cameras". In: *CoRR* abs/2412.06770. arXiv: 2412.06770

Background 2

Videos are discrete representations of visual signals. Chapters 3 to 5 are concerned with the nature of this representation: Chapter 3 is about detecting transformations of video signals that alter the conclusions an observer would draw from them. Chapter 4 aims at generating videos that are not derived from visual signals, but look as if they might have been. Chapter 5 attempts to approximate the visual signal from which a given video has been derived, with the help of event data.

This chapter defines some basic principles and notations and establishes the notions of authenticity, generation and reconstruction used in this thesis.

2.1 Basics

$\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}$ the set of real numbers. The thesis uses the notions of

- **discrete ranges** $\mathbb{N}_{<n} := \{i \in \mathbb{N} \mid 0 \leq i < n\}$
- **dense intervals** $[a; b] := \{x \in \mathbb{R} \mid a \leq x \leq b\}$

Besides mathematical foundations including probability theory, the thesis assumes a basic understanding of neural networks and gradient-descent-based optimization.

2.2 Temporal Dependence

Given a probability measure P, two probabilistic events A and B are called **independent**, if and only if

$$P(A \cap B) = P(A) \cdot P(B) \tag{2.1}$$

Two random variables $X, Y \in \mathbb{R}$ are called **independent**, if and only if for all $x, y \in \mathbb{R}$, the events $\{X < x\}$ and $\{Y < y\}$ are independent.

Informally, independence of two variables X and Y means that no amount of information about the value of X can yield any information about the value of Y, and vice versa. The above definitions show that independence of events and independence of variables/quantities are closely related.

Two events/variables are called **dependent**, or exhibiting a **dependency**, if and only if they are not called independent. Informally, this means that it is possible for information about one of the events/variables to admit conclusions about the other event/variable, even if these conclusion may be of stochastic nature only, e.g. "Based on X having the value x, it is now very likely that Y has value...". The more the knowledge about X reduces the uncertainty about the value of Y, the "stronger" one considers the dependence of Y on X. This notion is formally captured by the concept of **mutual information** (Cover et al., 1991).

The events/variables/quantities considered in this thesis range from simple observations like physical signal values or colours of (sub-)pixels of a video, to all sorts of variables/quantities/predicates that can be defined on them. Informal examples are "Brightness value of pixel $(123, 456)$ at time 0.543s", or "Center point of the left pupil of the dog in frame 42". The thesis does not explicitly formulate such random variables. Instead, it postulates that the probability distributions and signals considered in Chapters 3 to 5 admit a great multitude of possible random variables and dependencies between them. In Chapters 3 and 4 it is up to the neural networks to discover and utilise them. Chapter 5 takes a different approach, by explicitly stating relationships between a ground truth signal and the information reported by a sensor, to model the signal.

The emphasis of this thesis is on *temporal* dependencies. A **temporal dependency** is one between random variables X and Y that are defined based on different subranges of the temporal dimension: While two pixels of the same video frame may or may not be **spatially dependent** (for example because they are direct neighbours and thus likely to have similar values), two pixels from *different* frames may or may not be temporally dependent, for example because they are part of the same feature of a moving object. Temporal dependencies allow one to use information from one temporal region to draw conclusions about another temporal region.

2.3 Visual Signals and Videos

A **visual signal** is a function $V : \mathbb{R}^4 \to \mathbb{R}^+$ that maps coordinate tuples $(t, \alpha, \beta, \lambda)$ to values $V(t, \alpha, \beta, \lambda)$. Visual signals in the real world originate from physical light transport, at the end of which light of wavelength λ is received at a point of a two-dimensional surface (parametrised by α and β), at some time t. Physically speaking, a value $V(t, \alpha, \beta, \lambda)$ quantifies

the **spectral irradiance** at surface position (α, β) at time t, resulting from light of wavelength λ, i.e. spectral irradiance is an amount of physical energy per time, per area and per wavelength, with standard unit $1\frac{\mathrm{J}}{\mathrm{m}^3 \cdot \mathrm{s}}$ (ISO 9288:2022(en), 2022).

A **frame signal** is a function $B : \mathbb{R} \to \mathbb{N}_{<W} \times \mathbb{N}_{<H} \times \mathbb{N}_{<C} \to \mathbb{R}^+$ where (W, H) is called the **spatial resolution** and C is called the **number of channels** of the signal. Spatial resolution will often be denoted $W \times H$, or (for $W = H = A$) as A^2 in the remainder of the thesis, when confusion with the Cartesian product over sets can be ruled out. In keeping with previous literature (e.g. Stepan Tulyakov et al., 2022; X. Zhang et al., 2022) values of B will often be called **brightness**, although this term is actually a perceptual one, instead of a physical quantity.

In the physical world, spectral irradiance is captured by a **sensor**, that consists of finitely many (sub-)pixels. To model this capture process, a visual signal V can be turned into a frame signal B_V by discretization of surface and wavelength: For each $(x, y, z) \in \mathbb{N}_{<W} \times \mathbb{N}_{<H} \times \mathbb{N}_{<C}$ the set $\mathrm{pixel}(x, y, z)$ comprises all spatiospectral coordinates (α, β, λ) that contribute to the (sub-)pixel with the address (x, y, z). B_V is defined by integration:

$$B_V(t)(x, y, z) := \int\limits_{(\alpha,\beta,\lambda)\in \mathrm{pixel}(x,y,z)} \mathcal{S}(t, \alpha, \beta, \lambda, V(t, \alpha, \beta, \lambda))\, d\alpha\, d\beta\, d\lambda \tag{2.2}$$

where $\mathcal{S}$ models physical properties of the sensor. By varying spatiotemporally, $\mathcal{S}$ can model all kinds of sensor noise.

A **frame sequence** or **video** is a function

$$F : \mathbb{N}_{<K} \to \mathbb{N}_{<W} \times \mathbb{N}_{<H} \times \mathbb{N}_{<C} \to \mathbb{R}^+ \tag{2.3}$$

where K is called the **temporal resolution**, (W, H) is called the **spatial resolution** and C is called the **number of channels** of the sequence.

The physical quantity corresponding to the values of F is radiant flux (SI unit $1\frac{\mathrm{J}}{\mathrm{s}}$). The pixels of a physical sensor need to be exposed to radiant flux for a non-zero amount of time, in order to accumulate electric charge in their circuitry, that can be measured and quantised to obtain digital data. This **exposure process** can be modelled by integration over time: Given a frame signal B, the video F_B is defined as

$$F_B(i)(x, y, z) := \mathrm{CRF}\left(i, x, y, z, \int\limits_{t_i^{\mathrm{open}}}^{t_i^{\mathrm{close}}} B(t)(x, y, z)\, dt\right) \tag{2.4}$$

where the intervals $[t_i^{\text{open}}; t_i^{\text{close}}]$ are called **exposure intervals**, **exposure periods**, or simply **exposures**, and the **camera response function** $\text{CRF} : \mathbb{N}_{<K} \to \mathbb{N}_{<W} \times \mathbb{N}_{<H} \times \mathbb{N}_{<C} \times \mathbb{R}^+ \to \mathbb{N}$ defines the mapping from accumulated radiant flux to recorded values. The longer the temporal interval over which Equation 2.4 integrates, the stronger the **motion blur** effect becomes, i.e. moving objects appear faint and elongated along the direction of their movement.

Equation 2.4 allows only **global shutter** recording, i.e. t_i^{open} and t_i^{close} are the same for all pixels. Not all cameras use a global shutter, but the DAVIS 346C in Chapter 5 does, while no assumptions are made about the shutter mode in Chapters 3 and 4.

While the exposures of a video are pairwise disjoint, they do not form a partitioning of sequence time, i.e.

$$\bigcup_{i=0}^{K-1} [t_i^{\text{open}}; t_i^{\text{close}}] \subsetneq [t_0^{\text{open}}; t_{K-1}^{\text{close}}] \tag{2.5}$$

This is because the electric charges that have accumulated in the sensor during exposure need to be read out and the pixels need to be refreshed, which can take non-negligible amounts of time, during which no exposure is possible. Chapter 5 refers to the intervals between exposures as **exposure gaps**.

Unless specified otherwise, a **frame** is a video F of temporal resolution 1, in which case the frame index argument may be omitted, i.e. $F(x, y, z) := F(0)(x, y, z)$.

2.4 Authenticity

A **proposition** p is a statement about the world. It is not useful to try and further formalise the notion of "statement about the world", other than to establish that for a statement to be called a proposition it must be considered binary, i.e. it is either **true**, or **false**. $\mathbb{P}$ denotes the set of all propositions. The **world** can then be modelled as the set

$$\mathbb{W} := \{p \in \mathbb{P} \mid p \text{ is true}\} \tag{2.6}$$

The world is consistent and complete, i.e. $\forall p \in \mathbb{P} : p \in \mathbb{W} \leftrightarrow \neg p \notin \mathbb{W}$.

Chapter 3 axiomatically assumes the existence of an **observer** function $O : \mathbb{V} \to \mathbb{P} \to \{\top, \bot, \Omega\}$, that interprets any given visual signal $V \in \mathbb{V}$ as information about the world, i.e. for all $p \in \mathbb{P}$, it holds that

$$(O(V)(p) = \top \to p \in \mathbb{W}) \ \wedge \ (O(V)(p) = \bot \to p \notin \mathbb{W}) \tag{2.7}$$

while $O(V)(p) = \Omega$ denotes that the observer cannot determine if p is true or false. This means: While the visual signal is giving only *incomplete*

information about the world, it is assumed that all the information it does give is *consistent* with the world.

As a second axiom, Chapter 3 assumes that sensors are well-behaved, i.e. O is defined on videos as well and it is consistent with the visual signal V a video was derived from:

$$\forall p \in \mathbb{P} : O(F_{B_V})(p) \neq \Omega \rightarrow O(F_{B_V})(p) = O(V)(p) \tag{2.8}$$

Note that this constraint does allow information from the signal to be lost in the capture process (Equations 2.2 and 2.4).

Chapter 3 aims at deciding whether a video is "real" or "fake". In this context, a video F is called **real**, or **authentic** if and only if there is a visual signal V, such that

$$\forall p \in \mathbb{P} : O(F)(p) \neq \Omega \rightarrow O(F)(p) = O(F_{B_V})(p) \tag{2.9}$$

This means that every authentic video must only allow conclusions that could as well be drawn from recording a visual signal with a well-behaved sensor (see Section 2.3). All other videos are called **synthetic** or **fake**.

This is clearly not the only way in which authenticity could be defined. For example it could be defined directly with respect to the world, circumventing the notions of visual signals or well-behaved sensors. This however, could all too easily admit synthetically generated videos to be called authentic, provided that they look artificial enough for the observer to conclude that they tell nothing about the world (i.e. $\forall p \in \mathbb{P} : O(F(p)) = \Omega$). Inconveniently, making videos look *less* realistic would then make them *more* likely to be called authentic, which is why it is important to tie the definition of authenticity to the recording process.

It is beyond the scope of this thesis to comprehensively explore the space of possible observer functions. Instead, the detectors in Chapter 3 learn to distinguish real videos from fake videos by training neural networks to tell videos recorded by well-behaved sensors apart from videos that are of different origin.

2.5 Generative Models

Let $\mathbb{F}$ be the set of all videos and P a probability measure on $\mathbb{F}$. Depending on the application, $P(A) = 0$ for large subsets $A \subseteq \mathbb{F}$. For example, P could assign each set of videos a quantity proportional to the number of elements in it that a majority of humans would identify as depicting a cat. A **generative model** for P is a pair (Z, G) of a random variable $Z \in \mathbb{R}^d$ and a function $G : \mathbb{R}^k \rightarrow \mathbb{R}^d \rightarrow \mathbb{F}$ for some $k, d \in \mathbb{N}$, that takes a parameter $\theta \in \mathbb{R}^k$ and a value $z \in \mathbb{R}^d$ and maps them to a video. Usually

Z is normally distributed, i.e. $z \sim \mathcal{N}(0;1)^d$. P_θ denotes the probability measure of the random variable $G(\theta)(Z)$. In this context, values z of Z may be referred to as **latent codes** and subsets of $\mathbb{R}^d$ as **latent spaces**. The goal is to design G and optimise θ such that P_θ becomes as similar as possible to P. This is desirable because it will enable G to produce samples of cat videos.

To achieve this goal in a principled way, the notion of "similarity" of two probability measures needs to be defined. A prominent choice for a similarity measure is to minimise Jensen-Shannon divergence, defined as

$$\mathrm{JS}(P, P_\theta) := \frac{1}{2}\,\mathrm{KL}\left(P, \frac{P + P_\theta}{2}\right) + \frac{1}{2}\,\mathrm{KL}\left(P_\theta, \frac{P + P_\theta}{2}\right) \tag{2.10}$$

where the nonsymmetric Kullback-Leibler divergence for two probability measures P_A and P_B is defined as

$$\mathrm{KL}(P_A, P_B) := \int P_A(x) \log\left(\frac{P_A(x)}{P_B(x)}\right) dx \tag{2.11}$$

Training a **Generative Adversarial Network** (GAN) is a way of approximating the minimum of $\mathrm{JS}(P, P_\theta)$: In addition to the generator G, the GAN comprises a **discriminator** $D : \mathbb{R}^l \to \mathbb{F} \to [0;1]$. GAN training then approximates a solution to

$$\min_\theta \max_\phi \left(\mathop{\mathbb{E}}_{F \sim P} \log D(\phi)(F)\right) + \left(\mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,1)^d} \log(1 - D(\phi)(G(\theta)(z)))\right) \tag{2.12}$$

where the term that is minimised is equal to $\mathrm{JS}(P, P_\theta)$ up to a constant (Goodfellow et al., 2014), provided that G and D have sufficient capacity. In practise the architectures of the neural networks constraining G and D may prevent the theoretical optimum from being reached. Chapter 3 trains GANs by gradient descent in order to synthesize fake videos.

However, not only in practice can training a GAN to convergence be difficult, but also, as Arjovsky et al., 2017, show, $\mathrm{JS}(P, P_\theta)$ can be discontinuous, making its gradient with respect to θ unsuitable for gradient-descent-based optimization. This suggests that the GAN objective (Equation 2.12) makes training unnecessarily difficult. Instead, Arjovsky et al., 2017, propose to minimise the **Wasserstein distance**, or **Earth Mover Distance**, that for probability distributions over videos can be defined as

$$\mathrm{EM}(P, P_\theta) := \inf_{\gamma \in \Pi(P, P_\theta)} \mathop{\mathbb{E}}_{(F_1, F_2) \sim \gamma} \|F_1 - F_2\| \tag{2.13}$$

where $\Pi(P, P_\theta)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are P and P_θ and for videos F_1, F_2 that are of equal temporal resolution

K and spatial resolution (W, H) one defines

$$\|F_1 - F_2\| := \sqrt{\sum_{\substack{0 \le i < K \\ (0,0,0) \le (x,y,C) < (W,H,C)}} (F_1(i)(x,y,z) - F_2(i)(x,y,z))^2} \tag{2.14}$$

If resolutions do not match, the smaller dimensions can be filled in with zeros.

Arjovsky et al., 2017, show that if G can be represented by a neural network with the usual building blocks (affine transformations, convolutions and non-linearities, that are continuous in θ and component-wise Lipschitz-continuous) then there exists a function f that solves

$$\max_{f \text{ is 1-Lipschitz}} \left(\mathop{\mathbb{E}}_{F \sim P} f(F) \right) - \left(\mathop{\mathbb{E}}_{F \sim P_\theta} f(F) \right) \tag{2.15}$$

By defining a neural network C_w with weights w, and training it by gradient descent, one can make C_w approximate this f. Once such an approximation is found, one can use it to compute

$$\nabla_\theta \operatorname{EM}(P, P_\theta) = - \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,1)^d} \nabla_\theta C_w(G(\theta)(z)) \tag{2.16}$$

which is what is needed to minimise $\operatorname{EM}(P, P_\theta)$ by gradient descent.

This framework of training the generator G with the help of a so-called **critic** C_w is called a **Wasserstein GAN**. Chapter 4 uses this approach to generate videos. The critic has a role similar to that of the discriminator in a GAN, but instead of classifying samples its purpose is to establish a well-behaved gradient between real videos and generated videos: While the maximization in Equation 2.15 drives the average values of real and generated videos under the critic as far apart as possible, thereby increasing the average critic gradient between them, the Lipschitz constraint on the critic prevents the gradient from exceeding certain limits. This means that in contrast to a GAN discriminator, whose gradients can become very staircase-like if it overfits to the current state of the generator, a Wasserstein GAN critic cannot be "over-trained".

To enforce the Lipschitz constraint of Equation 2.15, Arjovsky et al., 2017, propose to bound w to a compact range, which suffices by the definitions of the usual neural network building blocks in the critic's architecture. However, based on the findings by Gulrajani et al., 2017, and the outcomes of early experiments, Chapter 4 instead uses the gradient penalty proposed by Gulrajani et al., 2017: The L2 norm of the gradient of C_w with respect to its input is averaged over a number of samples, to penalise deviations of this norm from 1. This soft constraint has been found to lead to smoother convergence, allowing higher learning rates. However, the gradient penalty objective does not allow batch normalization in the critic (Gulrajani et al., 2017).

2.6 Event Streams

An **event stream** is a finite set E of tuples

$$e_j = (t_j, x_j, y_j, z_j, p_j) \in [t_{\text{start}}; t_{\text{end}}] \times [0 : W] \times [0 : H] \times [0 : C] \times \{-1, +1\}$$

for some temporal interval $[t_{\text{start}}; t_{\text{end}}]$ and $W, H, C \in \mathbb{N}$ that satisfies

$$\forall t, x, y, z : |\{e \in E \mid \exists p : e = (t, x, y, z, p)\}| \leq 1 \tag{2.17}$$

The elements of E are called **events**, with t_j the **time**, (x_j, y_j, z_j) the **address** and p_j the **polarity** of the event e_j. In the remainder of the thesis, the word "event" will usually stand for the events defined here, not for the concept from probability theory. Two events $e_j, e_{j'} \in E$ are called **consecutive** if and only if $t_j < t_{j'}$ and

$$\exists x, y, z : \{(t, x, y, z, p) \in E \mid t_j \leq t \leq t_{j'}\} = \{e_j, e_{j'}\} \tag{2.18}$$

In this case, e_j is called **predecessor** of $e_{j'}$ and $e_{j'}$ is called **successor** of e_j. The predecessor and successor of any event are unique if they exist.

Event streams are usually interpreted as information about a frame signal B: It is assumed that there exist polarity-dependent **logarithmic brightness thresholds** $c_{+1}, c_{-1} \in \mathbb{R}^+$, by which the logarithm of B within one pixel must deviate from a reference value in order to trigger an event. This reference value is usually the logarithm of B at the predecessor event.
An **ideal event stream** for B is any event stream E that satisfies the following rules for all x, y, z: Let **logarithmic brightness** be defined as $\tilde{B}(t) := \log(B(t)(x, y, z) + \epsilon)$ for a small positive constant ϵ and let $\text{next}(t) := \min\{t' \mid t < t' \wedge \exists p : (t', x, y, z, p) \in E\}$ for any time $t_{\text{start}} \leq t < t_{\text{end}}$. Then all t with $t = t_{\text{start}} \vee \exists p : (t, x, y, z, p) \in E$ must satisfy:

1. $t_{\text{start}} \leq t \leq t_{\text{end}}$
2. For $(t', x, y, z, p') \in E$ with $t' = \text{next}(t)$:

$$p' \cdot (\tilde{B}(t') - \tilde{B}(t)) \geq c_{p'} \tag{2.19}$$

3. For all $t' \in]t; t_{\text{end}}]$ and $p' \in \{-1, +1\}$:

$$p' \cdot (\tilde{B}(t') - \tilde{B}(t)) \geq c_{p'} \;\rightarrow\; \text{next}(t) \leq t' \tag{2.20}$$

The event model defined here is very similar to those used in previous literature (e.g. Pan et al., 2019; Rudnev et al., 2021; X. Zhang et al., 2022), although many of them assume $c_{+1} = c_{-1}$.

Assuming $t_{\text{start}}, t_{\text{end}}, \epsilon, c_{+1}, c_{-1}$ are fixed globally, there is exactly one ideal event stream for every B: For any given pixel, if one starts at $t = t_{\text{start}}$ and sweeps across the interval $[t_{\text{start}}; t_{\text{end}}]$ applying Equation 2.20 always as early as B allows, one can add events to E that also satisfy Equation 2.19, showing that an ideal event stream can exist. Let now E, E' be two ideal, but strictly different event streams. For any given pixel, if E contains no events, then Equation 2.20 makes sure that Equation 2.19 is unsatisfiable for any event, so E' could not contain events either. It can thus be assumed w.l.o.g. that t' is the time of the earliest event in $E \setminus E'$, and t the time of the predecessor event (contained in both), or $t = t_{\text{start}}$ if no predecessor exists. Then by Equation 2.19 the premise of Equation 2.20 is fulfilled for E', but the conclusion would not be, violating the assumption that E' is ideal.

The converse is not true: For a given ideal event stream E there are in general infinitely many different frame signals B, for the following reasons:

1. Initial brightness $B(t_{\text{start}})$ is not known.

2. Equation 2.20 allows logarithmic brightness to fluctuate in-between events as long as it does not leave the corridor defined by c_{+1}, c_{-1}. Because of the logarithm the corridor for the linear brightness becomes larger the brighter the signal at the predecessor event is (see also Figure 5.12).

3. In practise c_{+1}, c_{-1} are often unknown.

Nevertheless, event streams have an important advantage over videos as sources of information about B: While a video averages the frame signal over relatively long temporal *intervals* (Equation 2.4), an event stream reports information about discrete *points* in time. This makes event streams an attractive data modality for applications in which temporal precision is of interest.

While the **reconstruction** of B from E is highly ambiguous already in theory, it is made even harder by the circuitry of the **event camera**, i.e. the electronic device that converts spectral irradiance into event streams: The logarithmic thresholds not only depend on polarity, but also vary spatially and temporally. The circuits take time to recover from events, i.e. the next event can only be triggered after a certain **refractory period**. This leads to a considerable **event latency**, i.e. the event time stamps t_j can be off by thousands of microseconds (McMahon-Crabtree et al., 2023; Serrano-Gotarredona et al., 2013; Y. Yang et al., 2024). In addition the circuitry tends to cause significant numbers of spurious events that violate Equation 2.19, leading to errors that accumulate over the course of the signal. While each of these effects can be amplified or reduced by tuning

the hardware configuration, they interact with each other, e.g. reducing the refractory period can greatly increase the number of spurious events.

Many approaches to event-based reconstruction (see Section 5.2) address these theoretical and practical ambiguities by learning prior assumptions from data. Most of them use **event binning**: They partition sequence time into a finite set of intervals and for each interval and pixel record only the total sum of the event polarities. This turns the event data into a format that allows pointer arithmetic (which is not possible on an event stream, because different pixels usually contain very different numbers of events for the same time interval) and it averages out noise. However, binning also discards most information about the distribution of the events inside the interval, effectively undoing some of the advantage that event streams have over videos in terms of temporal resolution.

In contrast, Chapter 5 uses hand-crafted priors to avoid laborious and bias-prone data collection, does not require pointer arithmetic and accounts for imperfections of the event camera by admitting confidence values for events, alleviating the need for event binning. The event camera model used in Chapter 5 is the DAVIS 346C (see Section 5.3), which records long-exposure frames along with events through the same pixel matrix, which further helps to disambiguate the reconstruction problem to some degree.

Detecting High-Quality Face Video Fakes

3

This chapter investigates the relationship between a human's ability to rate the authenticity of a video and a machine's ability to do the same (published as Fox et al., 2021a). Since the learning-based synthesis of photos and videos has reached quality levels that make the output hard to discern from authentic footage, the scientific community has begun to address concerns about the misuse of this technology. One branch of these efforts is the development of learning-based forgery detection algorithms, that are trained and evaluated on benchmark datasets. This chapter examines if the performance of such detectors depends on the presence of artefacts that the human eye can see. As a test bed for this purpose, it introduces a new benchmark dataset for face video forgery detection, VIDEOFORENSICSHQ, of unprecedented visual quality. This dataset allows to demonstrate that previous detection techniques have difficulties detecting fakes that reliably fool the human eye, especially because they neglect the temporal dimension. The chapter introduces a new family of detectors that examine combinations of spatial and temporal features, to outperform existing approaches both in terms of detection accuracy and generalization to unseen manipulation methods.

3.1 Introduction

Methods for face video synthesis have reached high levels of visual realism. Some allow facial expressions to be modified or transferred (H. Kim et al., 2018; Thies et al., 2020, 2016), while others implement face swapping, i.e. replacing the face interior with a different face identity (Garrido et al., 2014). Reacting to concerns that these could be misused to modify videos in unethical ways, the research community has developed techniques to detect forgeries for generic content (Afchar et al., 2018; Bayar et al., 2016, 2018; S. Wang et al., 2020) as well as specifically for faces (Agarwal et al., 2019; Rössler et al., 2019; Sabir et al., 2019).

In order to compare the effectiveness of forgery detection methods it is vital to evaluate them on benchmark datasets. As one example, FACEFORENSICS++ (Rössler et al., 2019) contains internet videos modified

by several face synthesis techniques (Dufour et al., 2019; Kowalski, 2018; Thies et al., 2020, 2016; torzdf et al., 2020) and demonstrates that an off-the-shelf image classifier, XCEPTIONNET (Chollet, 2017), outperforms methods specifically designed for fake detection. However, whenever a forgery detector achieves a high detection accuracy on a dataset, one must wonder: Does this mean that the detector is very good, or does it mean that the fakes in the dataset are just too easy to detect? The observation that the fakes in existing benchmark datasets seem easy to spot for the *human* eye (Figure 3.2) gives rise to the following hypothesis:

Hypothesis (H): *The accuracy of existing face video forgery detection methods depends on visual artefacts that humans would be able to spot with the naked eye. As soon as fakes are missing such artefacts, detector performance will drop.*

The relevant artefacts include implausible lighting, unnatural smoothness and splicing boundaries occurring as part of the synthesis process. In addition the synthesis methods often produce *temporal* artefacts, that previously were neglected by many forgery detectors, but can easily be seen by humans. In the course of investigating **H** this chapter makes two main contributions:

First, it presents VIDEOFORENSICSHQ, a benchmark dataset of high quality face video manipulations, designed to *not* include said artefacts (Figure 3.1). A user study (Section 3.4) shows that humans find the fakes in it considerably harder to detect than in previous datasets. Only VIDEOFORENSICSHQ allows to investigate **H**, by evaluating existing detectors on it, showing that their performance leaves room for improvement.

Second, making use of this room, the chapter presents a novel family of learning-based detectors that examine combinations of colour, low-level noise and temporal dependencies. These detectors are found to perform better than previous methods on high-quality fakes. Especially their temporal component shows improved generalization to unseen synthesis methods.

3.2 Related Work

3.2.1 Face Editing & Reenactment

The basis for the manipulation of facial imagery are methods to reconstruct, edit and "reenact" face images or videos, where **reenactment** means to combine the expressions and poses of a source actor with the appearance of a target actor. Zollhöfer et al., 2018, give a good overview of the state of the art of such techniques at the time, while this section focuses on select methods with more direct relevance to the creation of "fakes".

Many facial editing methods first fit a face model to the input data. A prominent such model is the pioneering PCA model by Blanz et al., 1999,

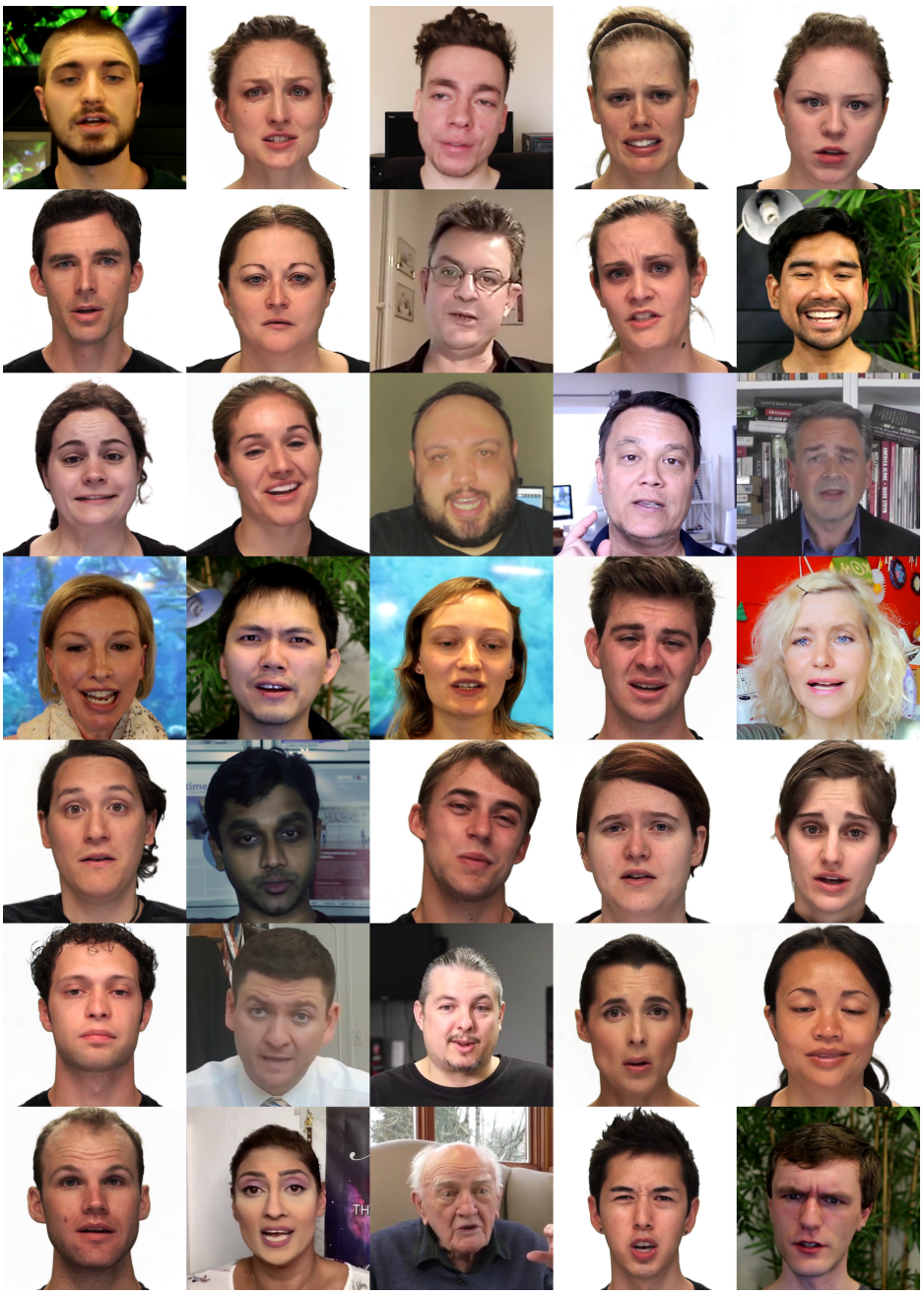
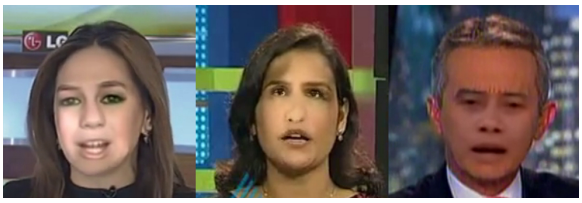


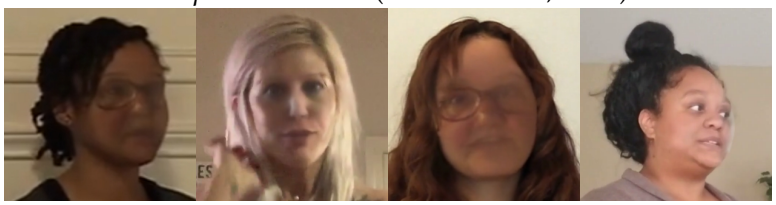
Figure 3.1: Frames from fake videos in VIDEOFORENSICSHQ: Each frame is the output from the synthesis pipeline in Figure 3.3. While this pipeline does produce artefacts typical for GAN-generated images, a human can hardly spot any flaws. To examine motion, see the project video (Section 1.5).



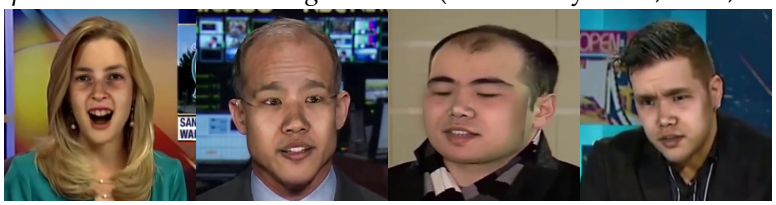
FACEFORENSICS++ subsets FS, DF and F2F (Rössler et al., 2019)

FACEFORENSICS++ subset NEURALTEXTURES (Rössler et al., 2019)

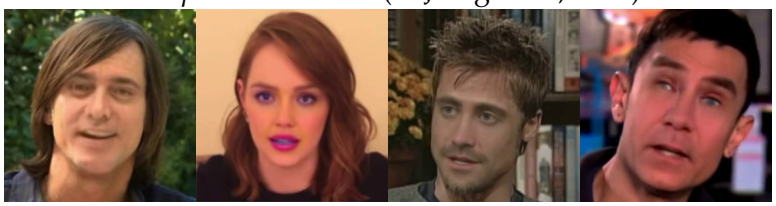
Deep Fake Dataset (Dufour et al., 2019)



Deep Fake Detection Challenge Dataset (Dolhansky et al., 2020, 2019)



DeeperForensics-1.0 (L. Jiang et al., 2020)



CELEB-DF (Y. Li et al., 2020)

Figure 3.2: Previous face video manipulation datasets contain many artefacts that humans can spot easily. NEURALTEXTURES shows good spatial quality, but playing its fakes back as videos allows humans to spot them, see Section 3.4.

but different sets of features, such as facial contours (W. Wu et al., 2018), so-called neural textures (Thies et al., 2019), and/or correspondences to reference images (Nagano et al., 2018) can also be used to model at least parts of the information about the input face. Generative Adversarial Networks (GANs; Goodfellow et al., 2014) are capable of learning large and complex probability distributions, and so are a component of many methods discussed in the following. Many GANs use U-NET-inspired architectures (Isola et al., 2017; Ronneberger et al., 2015), that because of their suitability for image-to-image transformation have been reused even when generative capabilities are only of secondary importance.

FACE2FACE, by Thies et al., 2016, was the first approach for real-time facial reenactment on monocular RGB input. It tracks both source and target actor by fitting a blend-shape model (Blanz et al., 1999) to the input frames and then solves an optimization problem to transform source expressions such that they fit into the distribution of target expressions. Wiles et al., 2018, take several images of the target actor as input and compute correspondences of these image pixels with one common reference image of the actor. They then reenact this actor given a driving sequence, that can be given as video or audio data. This method does not model temporal dependencies. The method by W. Wu et al., 2018, trains subject-specific autoencoders, that represent the state of a face by the shape of its contours. By encoding a source face, transforming the resulting boundaries with a CYCLEGAN-inspired approach (Jun-Yan Zhu et al., 2017) and then decoding it with the decoder learned for a different identity, facial expressions can be transferred between subjects. The method does not model temporal dependencies and thus exhibits a very noticeable flickering. Nagano et al., 2018, take one single, neutral-expression input image of the target actor, and drive it based on a video of the source actor. Notably there do not appear temporal smoothness or consistency flaws in the results, but spatially there are issues with lighting that is baked into textures and with unnatural deformations around the eyes and mouth interior that give away the artificiality of the results. *Deep Video Portraits* (DVP), by H. Kim et al., 2018, for the first time showed space-time coherent realistic global pose and expression editing in videos using a GAN. H. Kim et al., 2019, even extended this method to one that preserves the personal speaking style of the target actor. Depending on the amount of training material that is available for the target actor, results can look flawless to human viewers, even in the temporal domain, in which the network is actively encouraged to learn dependencies. In NEURALTEXTURES Thies et al., 2019, reconstruct input videos by fitting a 3D geometry model to their frames. This geometry model is given a feature texture, that can be projected into the image plane, and then turned into a photorealistic re-rendering of the original frame, via a so-called *deferred neural renderer*. Both the renderer and the texture are optimised to reconstruct the input video well. By re-

constructing two videos in this way and then combining the texture from one video with the geometry from the other, the appearance of a target actor can be applied to the performance of a source actor. Both spatial and temporal quality often appear flawless, but there do occur occasional flickering artefacts. FACESWAP (Kowalski, 2018) and DEEP FAKES (torzdf et al., 2020) at the time were very popular face editing implementations on GitHub, but at least at the degree of automation that was necessary to create FACEFORENSICS++ (Rössler et al., 2019) (see Section 3.2.2), both their spatial and temporal quality are noticeably flawed. The method by Zakharov et al., 2020, turns the input face image into an "avatar" that can be reenacted at more than 20FPS on a smartphone, using facial landmark vectors as a driving sequence. However, objects occluding the face (glasses or hats) are not handled well, and identities are not preserved well enough to reliably fool human viewers. Many of the aforementioned methods, despite their decent spatial output quality, model temporal dependencies either not at all, or only partially and so do exhibit noticeable flaws when their output is played as a video. A good example for this observation is NEURALTEXTURES (see Figure 3.2), where spatial artefacts are often small, but temporal flaws could be spotted in the user study (Section 3.4).

3.2.2 Face Manipulation Datasets

At the time the work presented in this chapter was conducted, there already had appeared, and were appearing, several datasets of manipulated images (Guan et al., 2019; Zhou et al., 2017) or videos (Dolhansky et al., 2019; Guan et al., 2019; Korshunov et al., 2019; Rössler et al., 2019):

Zhou et al., 2017, provide 2010 manipulated images generated with a popular smartphone app and with the method by Kowalski, 2018. Guan et al., 2019, provide large numbers of manipulated images and videos, but they cover general content, not just faces, and only contain 2340 images and 118 videos resulting from GAN-based manipulations. Korshunov et al., 2019, provide one of the first face *video* datasets using GAN-based "deep fake" algorithms. It contains 620 manipulated videos of 43 subjects, but the resolution of the face region is at most 128^2 pixels. The FACEFORENSICS++ dataset (Rössler et al., 2019) contains 1000 authentic videos. Each of them was manipulated with 4 different editing techniques: DEEP FAKES (torzdf et al., 2020), FACESWAP (Kowalski, 2018), FACE2FACE (Thies et al., 2016), and NEURALTEXTURES (Thies et al., 2019). As already stated in Section 3.2.1, DEEP FAKES and FACESWAP exhibit very noticeable artefacts, while FACE2FACE and NEURALTEXTURES produced significantly better quality, but neither reliably fool human viewers (see also Figure 3.2 and Section 3.4). Google released the *Deep Fake Detection Challenge Dataset* (Dufour et al., 2019). It contains over 3000 manipulated videos, with many exhibiting visual artefacts that humans can spot.

Facebook released the *Deep Fake Dataset* (Dolhansky et al., 2020, 2019) of manipulated face videos of varying quality, with face resolution often much less than 299^2. Fakes are often easy to spot for the human eye. The dataset DEEPERFORENSICS 1.0 (L. Jiang et al., 2020), provides, augmentations aside, 1000 forgeries derived from FACEFORENSICS++. Artefacts are more subtle than in previous datasets, but lighting baked into textures and temporal flickering often give fakes away. CELEB-DF (Y. Li et al., 2020), derives more than 5639 fake videos from 590 authentic YouTube videos. Artefacts detectable by humans are rare, but do include spatial artefacts (see Figure 3.2) as well as temporal flickering, especially at blending boundaries.

As can be seen in Figure 3.2, the user study in Section 3.4, and the evaluation in Section 3.6, none of these datasets meets the requirement of containing a large number of high resolution manipulated face videos that reliably fool human viewers and machines.

3.2.3 Detection of Manipulated Visual Content

Already before the advent of deep learning, the detection of manipulations in digital images and videos was an active area of research. This section focuses on the more recent efforts that either constituted or were directly related to the state of the art in the detection of forged facial footage. More comprehensive surveys are provided by Tolosana et al., 2020, and Verdoliva, 2020.

Many detection techniques do not make any strictly face-specific assumptions. A foundational work in this category is the one by Fridrich et al., 2012. It introduced convolutional kernels designed for steganalysis, that several later works built on: Cozzolino et al., 2014, combined some of these kernels with an SVM-based classifier, while Cozzolino et al., 2017, initialise a convolutional neural network (CNN) with the kernel weights and fine-tune to further improve detection performance. Bayar et al., 2016, 2018, constrain the convolutional layers of a CNN in such a way as to limit their output to high spatial frequency content. This helps the CNN focus on those spatial frequencies that typically contain traces of manipulations. With a similar motivation Qian et al., 2020, apply the discrete cosine transform (DCT) to input images before further processing. Zhou et al., 2018, use a two-stream network to detect edited images. One stream processes the image content while the other stream focuses on high spatial frequencies, based on insights from Fridrich et al., 2012. NOISEPRINT (Cozzolino et al., 2020) localises edited regions of an image by extracting patterns that are characteristic for a particular model of camera and detecting inconsistencies in these patterns. S. Wang et al., 2020, showed that a standard image classifier trained on one CNN generator can generalise well to data produced by unseen generators. The classifier

is trained on a large volume of data with careful pre- and post-processing and data augmentation. Results show high classification accuracy on a variety of unseen synthesis methods, including architectures for face generation. None of the aforementioned detection techniques take the temporal dimension into account.

Face-specific forgery detection techniques can be classified into single-image-based (Afchar et al., 2018; Agarwal et al., 2019; Durall et al., 2019; L. Li et al., 2020; Raghavendra et al., 2017; Rössler et al., 2019; S. Wang et al., 2019, 2020; Zhou et al., 2017) and multi-image-based (Agarwal et al., 2019; Sabir et al., 2019) approaches. Zhou et al., 2017 (similar to their later work; Zhou et al., 2018) use a two-stream network for the detection of manipulations, but the focus is on faces and the spatial high frequency stream refines the features from Fridrich et al., 2012, by means of a triplet loss. Raghavendra et al., 2017, detect the rather specific manipulation of morphing two faces into one. Their input is assumed to have undergone physical printing and scanning. The detector by Rahmouni et al., 2017, learns to distinguish authentic photos from computer-generated content, but the synthetic images in their training data are easily identified as such by humans. MESOINC-4 (Afchar et al., 2018) is an inception-inspired (Szegedy et al., 2017) CNN with a small number of layers. It stacks the output of several convolutional layers with different kernel shapes, to learn at which level of granularity the input should be analysed. S. Wang et al., 2019, sampled the space of manipulations that Adobe Photoshop's "Face-Aware Liquify" tool has to offer, in order to create a training set for their detector. Remarkably, this detector generalises to a test set of of manipulated faces that an artist created with the same tool and a more general version of it. Durall et al., 2019, classify the Discrete Fourier Transform (DFT) of images using support vector machines, logistic regression and k-means. L. Li et al., 2020, aim at detecting face images that result from smooth blending of two source images, such that a blending boundary can be extracted. Smooth blending is a component of many manipulation techniques and the authors show that the model generalises to unseen manipulations as long as they contain this component. XCEPTIONNET (Chollet, 2017) is a deep neural network designed for general image classification. Rössler et al., 2019, used it for the detection of manipulations in face images. Their evaluation has it outperform a number of other models that were specifically designed for the fake detection task (Afchar et al., 2018; Bayar et al., 2016, 2018; Cozzolino et al., 2014, 2017; Fridrich et al., 2012; Rahmouni et al., 2017).

One of the few works that study temporal dependencies, by Y. Li et al., 2018, learns to detect manipulated content by unnatural eye blinking. This demonstrates the utility of temporal signals for forgery detection, but is of course a rather constrained cue that attackers can easily avoid. Agarwal et al., 2019, assume that the input video to be classified shows a known

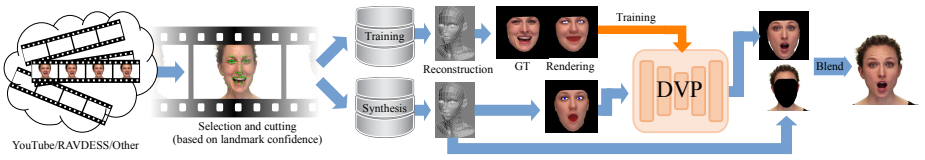


Figure 3.3: Every video in the "Synthesis" set undergoes monocular reconstruction, to obtain facial parameters that are close to the training distribution. DVP (H. Kim et al., 2018) turns these into photorealistic videos.

person of interest (POI) for which sufficient authentic training material is available. Idiosyncratic patterns of facial and head movements, so-called "action units" are learned for the POI, such that a support vector machine (SVM) can distinguish their action units from those of other people. This approach is shown to work well for both comedic impersonations of the POI, as well as reenactment manipulations where a source actor is providing the driving motion. However, the authors admit that the approach is not robust to context changes, i.e. the POI facing slightly off-camera instead of looking directly into the camera can compromise detection performance. Sabir et al., 2019, present a detector based on a recurrent neural network (RNN) that processes the frames of a video. It improves detection accuracy by 4.55% over state of the art detectors. The publication measures this improvement on the strongly compressed version of the FACEFORENSICS++ dataset. However, the evaluation only covers those subsets that contain strong human-visible artefacts even under strong compression (namely F2F, FS and DF), while a quantitative analysis on the NT subset is not provided. This may be explained by the NT subset having been added only to a later version of FACEFORENSICS++.

The detectors presented in Section 3.5 are based on XCEPTIONNET, but they combine multiple input modes including temporally preprocessed inputs. Section 3.6 shows that these detectors perform better than previous ones on a dataset that contains very high quality face manipulations (Section 3.3).

3.3 The VideoForensicsHQ Dataset

Investigating **H** (see Section 3.1) requires a benchmark dataset that contains *many* fakes of high quality: In order for humans to be unable to spot fakes, the dataset should avoid artefacts such as temporal jitter, unnatural movement, implausible lighting, unusual smoothness, or strong blending boundaries. Before the creation of VIDEOFORENSICSHQ there were state-of-the-art synthesis techniques that achieved such quality under ideal conditions, but no *large-scale* benchmark dataset aggregating *many* such

high quality results. The user study in Section 3.4 confirms that the fakes in VIDEOFORENSICSHQ fool humans significantly more often than those in other datasets. Since it was by no means the goal of this work to find a novel synthesis method, the GAN-based method *Deep Video Portraits* (DVP, H. Kim et al., 2018) is adapted for *large-scale* fake creation. DVP is a *conditional* GAN, i.e. instead of sampling from a learned distribution completely randomly (which would make targeted fakes rather difficult to construct), the attacker can make DVP's output follow a desired "driving sequence".

While DVP can transfer performances from a source person to a *different* target person, this mode can lead to artefacts if the distribution of facial expressions differs a lot between source and target. Not even the "style-preserving" variant (H. Kim et al., 2019) avoids glitches as reliably as necessary. VIDEOFORENSICSHQ is thus limited to "intra-person" transfers (i.e. source and target are the same person). Previous works (Fried et al., 2019; Suwajanakorn et al., 2017) show this to be a very relevant threat scenario. Since DVP is trained on a set of frames that is disjoint from the source/target sequence, it has not seen the expressions to synthesize in advance and must still generate the typical GAN artefacts that are common with synthesis methods, but typically go unnoticed by humans.

3.3.1 Synopsis

VIDEOFORENSICSHQ contains 1737 videos of talking faces (43% male, 57% female), with 8 different emotions. Figure 3.1 shows example fakes from the dataset. Most videos have resolution 1280×720. They amount to $1{,}666{,}816$ frames with average resolution 968^2 and the average face covering 487^2 pixels. There are three different subsets: Group #1 was mined from the data used by H. Kim et al., 2019, Group #2 from RAVDESS (Livingstone et al., 2018), and Group #3 from YOUTUBE. Table 3.1 lists the sizes of these subsets. In total, VIDEOFORENSICSHQ contains $326{,}973$ fake frames, comparable to the NEURALTEXTURES (Thies et al., 2019) part of FACEFORENSICS++. While their fakes are the ones that come closest to VIDEOFORENSICSHQ in terms of visual quality (see Figure 3.2), Section 3.4 shows that fakes in the latter are much harder to detect for humans: 65.8% of VIDEOFORENSICSHQ fakes are mistaken as reals, while only 14.3% of the NEURALTEXTURES fakes pass this test.

3.3.2 Production Process

Mining real videos as the basis for fakes is challenging because jump-cuts, animations and unusual face poses need to be circumvented automatically, especially for YOUTUBE. To synthesize video of an identity, DVP requires about 5 to 10 minutes of training material, with all frames showing the

same face at roughly the same distance, in a near-frontal pose. To find such material, a facial landmark tracker (Saragih et al., 2011) runs over all frames of each source video F, obtaining 66 landmark positions for every frame $F(i)$, and one confidence value in the range $[0; 1]$ for every landmark position. Three metrics are computed on this basis:

1. c_i: average landmark confidence for frame $F(i)$
2. d_i: average offset between landmark positions in $F(i)$ and $F(i-1)$, divided by face size
3. mean and standard deviation of the c_i's and d_i's

A frame is regarded unsuitable in any of the following cases:

a) $c_i < 0.2$

b) $d_i > 0.1$

c) $c_i < 0.6$ deviates from the confidence mean by more than 110% of the standard deviation (in negative direction)

d) $d_i > 0.025$ deviates from the displacement mean by more than 110% of the standard deviation (in positive direction)

If none of these apply, the frame is added to the current segment of suitable frames. A **segment** here is a contiguous set of frames, with no scene cuts. The longest good segments are added to the **training set** for the respective identity until 5000 to 6000 frames are reached. All good segments beyond that make up the disjoint **synthesis set** for this identity. While the training set is used for training DVP, the synthesis set will be used as the basis for the actual fakes.

The training set is processed with a monocular face reconstruction approach (Garrido et al., 2016), that encodes the facial performance as a sequence of parameter vectors. The vectors are then rendered to obtain the conditioning input that DVP learns to turn into RGB output again (Figure 3.3). This way, for each identity, one obtains one DVP model that can render facial performances at photorealistic quality. The input to such a model can be any arbitrary facial performance, also given as a sequence of parameter vectors. But to reliably avoid strong artefacts, one should give facial performances as input that are close to the distribution that DVP saw during training (without, of course, using any of the training data!). One can simulate an attacker that is able to synthesize such parameter sequences, by applying monocular reconstruction to the synthesis set as well, thereby obtaining parameters that have the necessary properties. This is why VIDEOFORENSICSHQ mostly avoids visible glitches, but still preserves the less noticeable artefacts that every GAN-based synthesis method inevitably exhibits.

Subset	Source	Real frames	Fake frames
Group #1	H. Kim et al., 2019	119,992	60,058
Group #2	RAVDESS	190,259	74,765
Group #3	YouTube	1,029,592	192,150
Total		1,339,843	326,973

Table 3.1: VideoForensicsHQ consists of three subsets. The authentic frames for all subsets were mined from different sources.

3.3.3 Detailed Usage of DVP

The original version of DVP (H. Kim et al., 2018) cannot handle dynamic backgrounds and works at a fixed resolution of 256^2. Training frames are thus cropped around the face and the background is masked out (see Figure 3.3), to make the network focus all its capacity on the face region. The resulting square images are scaled to resolution 256^2. Instead of a separate conditioning image for the eye gaze (like in DVP), gaze renderings are overlayed on the face rendering. Importantly, DVP allows the supervision of temporal dependencies, by means of "temporal windows", as proposed by H. Kim et al., 2018: The discriminator sees temporal volumes of 5 frames for most of Group #2. Since they were found to not improve subjective quality considerably, Group #1 and Group #3 were synthesized with window size 1. DVP was trained for up to 200 epochs, estimating the mean squared photometric error against ground truth on the validation set. The model with the smallest error was used for synthesis. Since the facial performances rendered at synthesis time have been reconstructed from real footage, the coordinates of the face region in that footage are known. This allows alpha-blending the DVP output into those original frames.

3.4 User Study

To compare the quality of fakes in VideoForensicsHQ to FaceForensics++ (Rössler et al., 2019), 13 manipulated videos were randomly selected from VideoForensicsHQ and the NeuralTextures subset of FaceForensics++ (Rössler et al., 2019; Thies et al., 2019) respectively. Other approaches in FaceForensics++ produce fakes with much more visible artefacts (see Figure 3.2). In addition, 6 unmodified videos from VideoForensicsHQ and 7 from FaceForensics++ were selected randomly. In total the study contained 39 videos, randomly shuffled for each participant.

For each video, participants had to answer the question "Does the video look real or fake?". Most participants were computer scientists, with little-to-no knowledge of face manipulation techniques. 61 subjects participated

Video set	Rated "fake"	Rated "real"
Real videos	15.0%	85.0%
NEURALTEXTURES fakes	85.7%	14.3%
VIDEOFORENSICSHQ fakes	34.2%	65.8%

Table 3.2: Study participants were given 39 videos from VIDEOFORENSICSHQ and the NEURALTEXTURES subset of FACEFORENSICS++, both authentic and manipulated, and had to rate them as "real" or "fake". Fakes from VIDEOFORENSICSHQ fooled the participants considerably more often than NEURALTEXTURES.

in the study. On average, fakes from VIDEOFORENSICSHQ were rated real 65.8% of the time, and fakes from FACEFORENSICS++ were rated real only 14.3% of the time (see Section 3.4). Authentic videos were rated fake 15% of the time, which reflects a baseline error level in human detection performance. Participants were also asked what made them flag videos as fake. Some of the most common responses were:

1. Various visual artefacts, especially in the mouth interior
2. Non-natural eye movement
3. Body movements or hand gestures not matching speech
4. Non-natural mouth-related movements e.g. lips being tight when they should not be, deforming/dislodging jaw, etc..
5. Incorrect audio-lip synchronization
6. A single glitch occurring over 2-3 seconds
7. Spoken language not matching language of written text

Many of these observations can only be made on videos, not on images, i.e. when a temporal component is present.

3.5 Detecting High-Quality Face Manipulations

XCEPTIONNET (Chollet, 2017) was ranked best in FACEFORENSICS++ (Rössler et al., 2019). If **H** is true, XCEPTIONNET should perform worse on VIDEOFORENSICSHQ than it does on FACEFORENSICS++. This expectation is justified because XCEPTIONNET is a generic image classifier that has not been designed for fake detection and thus should look for clearly visible artefacts in the image space. The goal of this section is to enhance its ability to detect seemingly flawless fakes, without compromising its ability to detect strong artefacts. To reach this goal, this section presents a

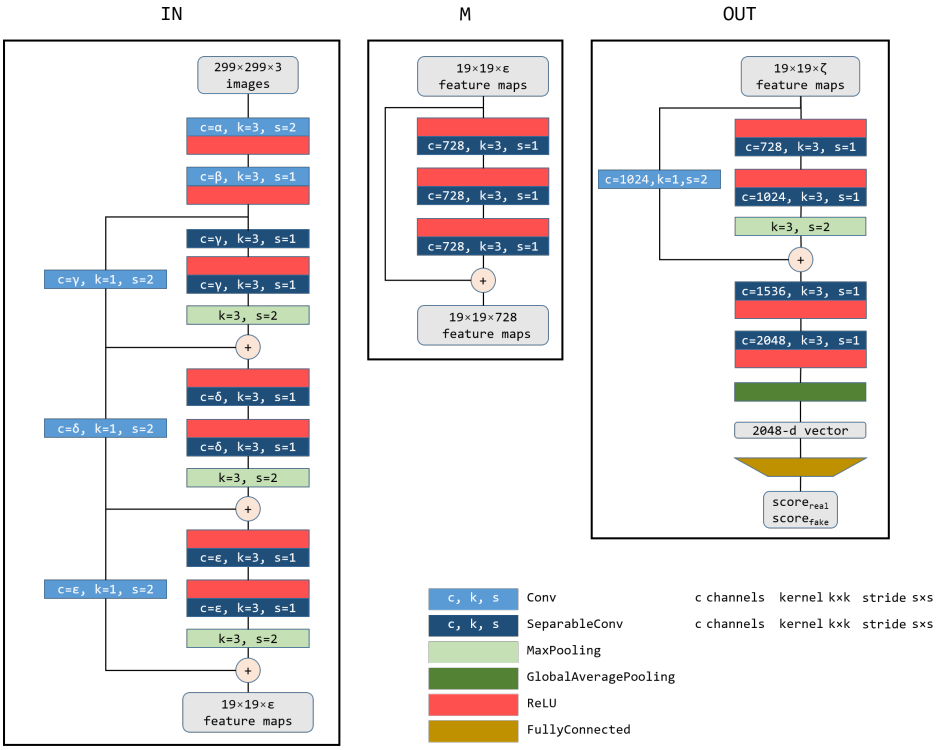


Figure 3.4: The building blocks of XCEPTIONNET (Chollet, 2017) are the basis for the multi-stream detectors presented in this section (see Figure 3.6). In order to trade memory capacity between multiple streams and fuse them at the right locations, the numbers of features in each layer vary in the different detectors. ϵ in M and ζ in Out are determined by the number of output feature in the preceding block. All "Conv" and "SepConv" layers are followed by batch normalization (Ioffe et al., 2015).

novel family of detectors (Figure 3.6) that examine combinations of multiple cues: the original RGB values, low-level spatial noise, and temporal dependencies (see Figure 3.7).

XCEPTIONNET can be modelled as a function

$$(\mathbb{N}_{<W} \times \mathbb{N}_{<H} \times \mathbb{N}_{<C} \to \mathbb{R}^{+}) \to [0;1]^2 \tag{3.1}$$

that maps RGB frames F to scores for the two classes "real" and "fake", with $W = H = 299$ and $C = 3$. XCEPTIONNET consists of an entry flow $\mathrm{In}_{C\alpha\beta\gamma\delta\epsilon}$, a middle flow M, and an exit flow Out, see Figure 3.4. Parameters α, β, γ, δ, and ϵ specify the number of features per convolutional layer. One can denote XCEPTIONNET as

$$\mathrm{X_C} := \mathrm{In}_{3,32,64,128,256,728} \circ \mathrm{M}^8 \circ \mathrm{Out} \tag{3.2}$$

In the following, leading or trailing zeros in the indices of $\mathrm{In}_{C,\alpha,\beta,\gamma,\delta,\epsilon}$ disable the respective layers.

Repetitions of M drive up memory consumption and training overhead. To test whether 8 repetitions are actually necessary for forgery detection, M can be omitted entirely:

$$\mathrm{X_B} := \mathrm{In}_{3,32,64,128,256,728} \circ \mathrm{M}^0 \circ \mathrm{Out} \tag{3.3}$$

Since VIDEOFORENSICSHQ contains very few strong visual artefacts that $\mathrm{X_C}$ or $\mathrm{X_B}$ could easily pick up, the network $\mathrm{X_S}$ is defined to not classify frames F themselves, but their spatially high-pass-filtered versions $\frac{1}{2} \cdot (F - g * F) + \frac{1}{2}$, where $g * F$ is the convolution of F with a Gaussian kernel of size 5 and standard deviation $\sigma = 1.1$. The architecture of $\mathrm{X_S}$ is that of $\mathrm{X_B}$.

The combination of $\mathrm{X_C}$ and $\mathrm{X_S}$,

$$\mathrm{X_{CS}} := \mathrm{concat}(\mathrm{In}_{3,32,64,128,256,364}, \mathrm{In}_{3,32,64,128,256,364}) \circ \mathrm{M}^2 \circ \mathrm{Out} \tag{3.4}$$

receives the same inputs as $\mathrm{X_C}$ and $\mathrm{X_S}$ and concatenates the colour and noise features just before entering M^2, where the combined receptive field of the convolutional kernels has size 17×17 (see Figure 3.6). $\mathrm{X_{CS}}$ can be extended to

$$\mathrm{PP_{ct}} := \mathrm{concat}(\mathrm{In}_{3,32,64,0,0,0}, \mathrm{In}_{3,8,8,0,0,0}) \circ \mathrm{In}_{0,0,72,128,256,512} \tag{3.5}$$

$$\mathrm{PP_{cts}} := \mathrm{concat}(\mathrm{PP_{ct}}, \mathrm{In}_{3,16,32,64,128,256}) \tag{3.6}$$

$$\mathrm{X_{CST}} := \mathrm{PP_{cts}} \circ \mathrm{M}^1 \circ \mathrm{Out} \tag{3.7}$$

which receives temporal features F_T as a third input (Figures 3.6 and 3.7). F_T is extracted from a temporal slice of the input video as follows:

1. Let F be the input video.
2. Obtain F' by convolving all frames of F with a spatial 2D Gaussian kernel (size 49, $\sigma = 7.7$), suppressing high spatial frequencies that motion would turn into temporal ones (e.g. an edge sweeping over a pixel).
3. Temporal high-pass filtering:

$$F''(i) := -\frac{1}{4}F'(i-1) + \frac{1}{2}F'(i) + -\frac{1}{4}F'(i+1) \tag{3.8}$$

4. Batch normalization:

$$F'''(i)(x,y,z) := \gamma_z \cdot F''(i)(x,y,z) + \beta_z \tag{3.9}$$

where γ_z and β_z are per-channel parameters of an affine transformation that are tuned during training to bring the average frame $F'''(i)$ to mean 0 and variance 1.

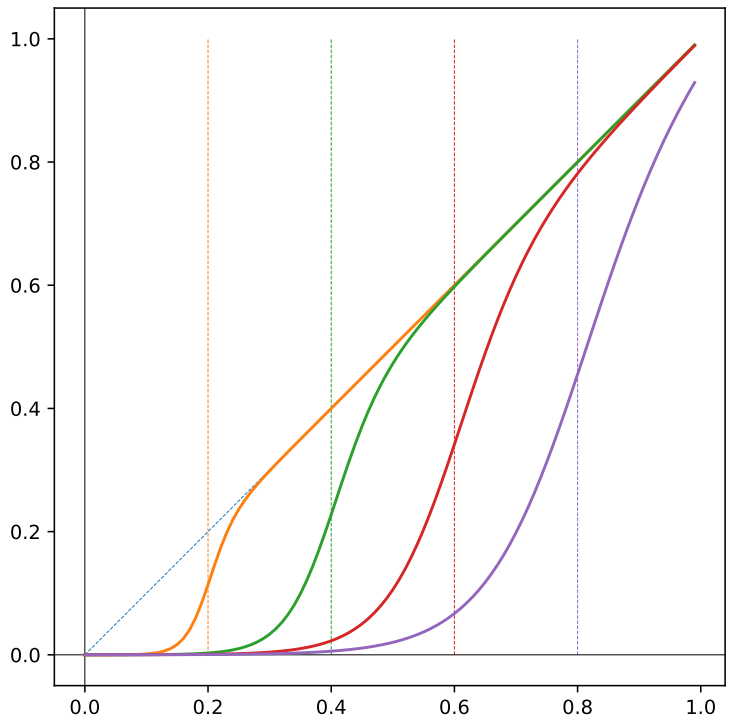


Figure 3.5: Graphs of cliff_t, for $t \in \{0.2, 0.4, 0.6, 0.8\}$. The function is used in Step 5 to suppress small temporal amplitudes, that are assumed to be of natural origin. cliff_t is differentiable in t, allowing the optimal cut-off amplitude to be learned.

5. Amplitudes smaller than t are dampened by computing

$$A(i) := \mathrm{cliff}_t(F'''(i)) - \mathrm{cliff}_t(-F'''(i)) \tag{3.10}$$

   where the cliff function

$$\mathrm{cliff}_t(f) := \frac{t}{10} \cdot (\log(1 + \exp(\chi(f))) + 10 \cdot \mathrm{sigmoid}(\chi(f))) \tag{3.11}$$

   with $\chi(f) := \frac{10}{t} \cdot (f - t)$ is smooth and differentiable in t (see Figure 3.5.

6. Computation of temporal gradients: $G(i) := |A(i) - A(i-1)|$.

7. Temporal low-pass filtering with kernel $(\frac{1}{32}, \frac{1}{8}, \frac{3}{16}, \frac{1}{8}, \frac{1}{32})$.

This process emphasizes unnaturally fast motions, often observed in forgeries. With the exception of Step 2, all operations are pixel-wise. Step 3 suppresses low temporal frequencies, which are likely of natural origin. Steps 4 and 5 filter the high frequency spikes for those of a certain minimum amplitude, which are most likely artificial: cliff_t is supposed

to dampen low amplitudes of a signal. The parameter t is a threshold specifying which amplitudes are to be dampened. Figure 3.5 plots cliff_t for different values of F'''. The function is differentiable in t, such that the training process can automatically determine a good position and shape for the "cliff". Step 6 turns oscillations between + and - into large positive values. Step 7 stabilises the resulting signals, such that more output frames exhibit bright regions that the classifier can detect.

The fact that Steps 1 to 7 use only very few trainable parameters (namely the batch statistics in Step 4 and t), helps prevent them from overfitting to the training distribution. The evaluation in Section 3.6.2 shows that $\mathrm{X_{CST}}$ generalises better to unseen manipulation methods than other detectors.

The number of repetitions of M and the points at which feature streams are fused were chosen empirically to maximise detection accuracy while not exceeding the $11\mathrm{GB}$ of GPU memory in an NVIDIA 1080Ti. The resulting trade-offs can lead to $\mathrm{X_S}$ performing slightly better than $\mathrm{X_{CS}}$ and $\mathrm{X_{CST}}$ on videos in which colour and temporal information do not give a benefit over spatial noise, because the latter two cannot dedicate as much memory to spatial noise as $\mathrm{X_S}$ (Section 3.6). On the other hand, spatial noise cues alone do not generalise as well as combinations with other types of information (Section 3.6.2).

3.6 Results

VIDEOFORENSICSHQ and the family of detectors presented in Section 3.5 allow the investigation of **H**:

State of the Art Detectors The detectors of Section 3.5 are compared to a number of previous approaches: Detector $\mathrm{X_C}$, i.e. XCEPTIONNET (Chollet, 2017), performs best in the FACEFORENSICS++ evaluation (Rössler et al., 2019), that also includes MESOINC-4 (Afchar et al., 2018) and MISLNET (Bayar et al., 2016, 2018). SIMPLEFEATURES (Durall et al., 2019) specifically analyses footage in the frequency domain. S. Wang et al., 2020, have trained an instance of RESNET-50 to generalise to the detection of unseen synthesis methods, to be referred to in the following as EASYSPOT.

Preprocessing and training All training and test data for all detectors was preprocessed by the same pipeline: Face bounding boxes were computed using DLIB (King, 2009), with temporal smoothing of their coordinates. Constant-size square bounding boxes were extracted and scaled to resolution 299^2 (exception MESOINC-4: 256^2). All videos were resampled at $25\mathrm{Hz}$. Frames for which no face bounding box could be found were omitted. For SIMPLEFEATURES, 209-dimensional feature vectors were computed as specified by Durall et al., 2019.

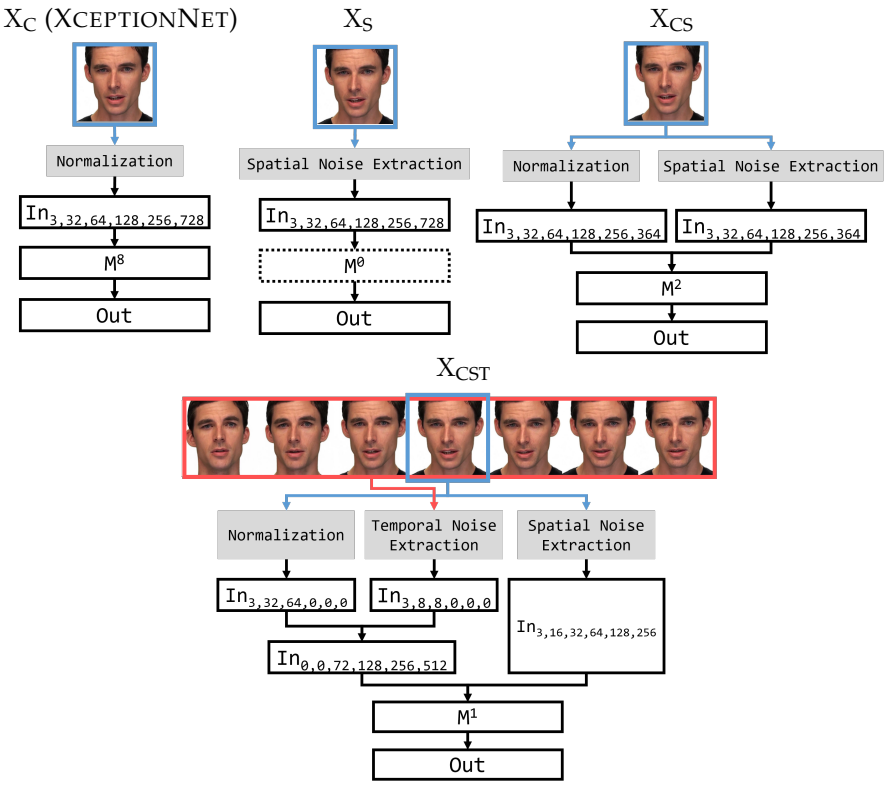
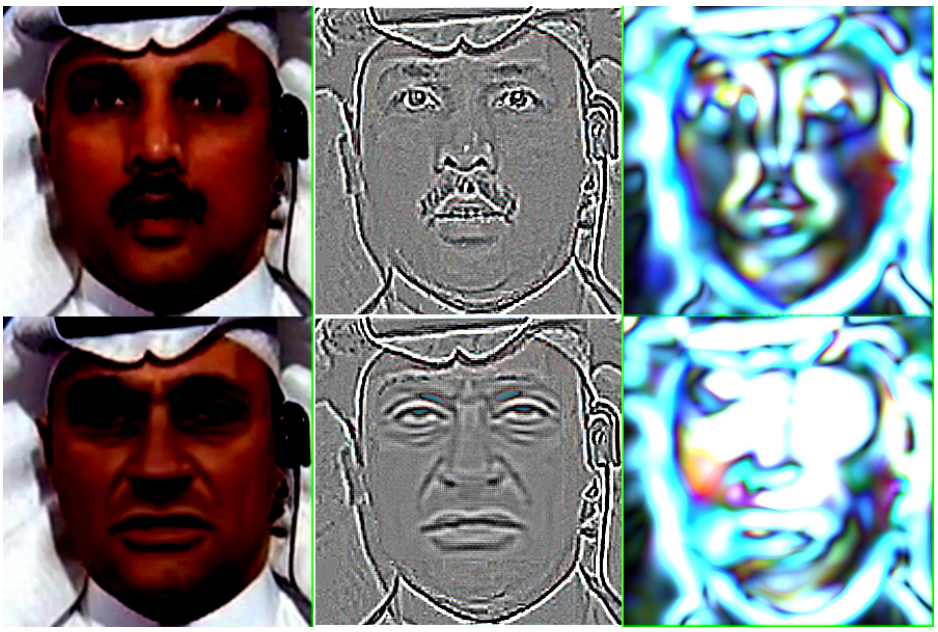


Figure 3.6: The detectors presented in this chapter extend XCEPTIONNET (Chollet, 2017, see Figure 3.4) to a multi-stream classifier for combinations of colour, spatial noise and temporal features (see also Figure 3.7).



Normalised RGB Spatial noise Temporal noise

Figure 3.7: The top row was extracted from an authentic video, the bottom row from a FACEFORENSICS++ fake. The X detectors (see Figure 3.6) process normalised colour (left), spatial high frequency information (centre) and the aggregation of temporal information defined in Steps 1 to 7 of Section 3.5 (right). The latter focuses on high frequency flickering, that is usually of unauthentic origin.

The Xception-based detectors X_C, X_B, X_S, X_{CS} and X_{CST} are all trained with stochastic gradient descent (momentum 0.9, weight decay 10^{-5}), multiplying the initial learning rate of 0.03 with factor $0.97^{0.1}$ per epoch. For X_{CST} the threshold t is initially set to $\frac{1}{40}$. Previous detectors are trained as specified in their publications, except for EASYSPOT and SIMPLEFEATURES: EASYSPOT is claimed to generalise well to unseen rendering methods and so this evaluation uses its pretrained weights and merely optimises a threshold on its singular output value, based on the ROC curve over the samples that were seen within one epoch of training. This optimization is performed for 5 epochs, averaging the 5 resulting thresholds. For every training batch of SIMPLEFEATURES a new SVM is optimised on the Fourier features. At validation and test time the predictions of all SVM models obtained in this way are averaged.

All detectors are trained with batch size 24, except for MESOINC-4 (512), SIMPLEFEATURES (512) and MISLNET (256). Except for EASYSPOT, all methods are trained with a hard limit of 100 epochs. Training stops earlier if 5 epochs with a validation accuracy of more than 99% have been seen (not necessarily consecutively). The model with maximal validation accuracy is used at test time.

To account for imbalances in the datasets (e.g. the different subsets of FACEFORENSICS++ contain different numbers of frames), 10% of the training frames and 20% of the validation frames are sampled in every epoch as follows: First a class is sampled ("real"/"fake"), then a subset (which is relevant for VIDEOFORENSICSHQ because it consists of three different groups), then a subject and then one of the sequences for this subject. Frames are sampled uniformly from sequences. Since SIMPLEFEATURES is not designed for the amounts of data resulting from the aforementioned sampling rates, they are lowered to 0.5% training and 1% validation samples for this method. This training procedure is the reason why the accuracies reported in Table 3.3 for MESOINC-4 and MISLNET are slightly lower than those reported by Rössler et al., 2019.

At test time, *all* frames of the test sets are used, but per-frame predictions are weighed by the probability of a frame being sampled according to above sampling process.

3.6.1 Detecting Highly Photorealistic Manipulations

To test **H**, detectors were trained on FACEFORENSICS++ (Rössler et al., 2019) and DEEPERFORENSICS 1.0 (L. Jiang et al., 2020), which both contain strong visual artefacts (Figure 3.2), as well as on VIDEOFORENSICSHQ, which does not (Figure 3.1).

Table 3.3 confirms **H**: With the exception of the detectors developed in Section 3.5 all accuracies are considerably lower on VIDEOFORENSICSHQ, than on previous datasets. In fact, XCEPTIONNET (X_C), the best detector

Detector	FF++	Deeper Forensics	VFHQ
X_C	99.23%	**98.64%**	88.59%
X_B	99.34%	98.09%	91.95%
X_S	**99.38%**	98.21%	**99.45%**
X_{CS}	99.25%	98.32%	97.12%
X_{CST}	99.35%	98.34%	97.78%
MESOINC-4	92.44%	97.25%	76.73%
EASYSPOT	75.55%	61.05%	56.44%
SIMPLEFEATURES	56.69%	62.88%	61.98%
MISLNET	95.82%	97.89%	74.65%

Table 3.3: All detectors were trained on the training+validation portions of the respective datasets and then tested on their test sets. Percentages are test accuracies averaged over two runs of the experiment. X_B, X_S, X_{CS} and X_{CST} consistently achieve high accuracies on all three datasets, whereas previous methods perform worse especially on VIDEOFORENSICSHQ. X_{CS} and X_{CST} do not benefit from analysing more clues than X_S, because VIDEOFORENSICSHQ does not have strong artefacts in these channels and the previous datasets contain strong artefacts that are easily detected.

in FACEFORENSICS++, drops by more than 10% and is even outperformed by X_B, which is a reduced version of X_C. The detectors X_S, X_{CS} and X_{CST} on the other hand perform well on all three datasets. The table shows X_{CS} and X_{CST} perform not quite as well as X_S. This is because they had to sacrifice some of the GPU memory that X_S can dedicate to spatial noise, in order to handle colour and temporal features (Section 3.5). Since VIDEOFORENSICSHQ does not contain strong visual or temporal artefacts, this sacrifice does not pay off.

Only VIDEOFORENSICSHQ is able to differentiate the best detectors from one another, while on previous datasets many detectors achieve close to 100% accuracy.

3.6.2 Generalization across Manipulation Techniques

Since the genesis of a fake is often unknown, detectors should generalise to unseen synthesis methods.

To evaluate this ability, detectors were trained on the FACEFORENSICS++ subsets FS ∪ NT (created with FACESWAP, Kowalski, 2018 and NEURALTEXTURES, Thies et al., 2019) and F2F ∪ DF (created with FACE2FACE, Thies et al., 2016 and DEEP FAKES, torzdf et al., 2020). Then they were tested on the subset they were *not* trained on. FACEFORENSICS++ is better suited for this experiment than VIDEOFORENSICSHQ, because the latter contains only one manipulation technique and differs from other datasets

in more respects than just the synthesis method (higher resolution faces, no visible artefacts, etc.).

Table 3.4 shows that training the X-detectors on FS ∪ NT makes them generalise well to F2F ∪ DF, where they outperform previous methods. X_{CS} ranking higher than X_S and X_C suggests that *combining* colour and spatial noise can help generalization.

The opposite experiment, i.e. training on F2F ∪ DF, followed by testing on FS, gives low accuracies for *all* detectors, suggesting that FS contains artefacts not seen in F2F ∪ DF. Surprisingly, although the subset NT is of significantly higher *spatial* quality than the others (see Figure 3.2), the X-based detectors manage to generalise to this manipulation method. It is here, in the absence of strong spatial artefacts, where especially temporal dependencies help X_{CST} perform significantly better than all the other detectors. Although detectors like X_C or MISLNET are theoretically able to learn the spatial filtering hard-coded in X_S, they perform considerably worse than X_S in Tables 3.3 and 3.4.

3.6.3 Importance of Temporal Features

It could be suspected that the only reason why X_{CST} generalises better than X_{CS} in Table 3.4 is that it simply dedicates fewer neurons to colour and spatial noise (see Equations 3.4 and 3.7), which serves as a kind of regularization. This is why Table 3.4 also contains the architecture $X_{CST\setminus T}$, that is exactly the same as X_{CST}, but with temporal noise extraction replaced by a layer that produces an all-zero image. This means that $X_{CST\setminus T}$ sees no actual temporal information, but dedicates exactly as many neurons to colour and spatial noise as X_{CS} does. Table 3.4 shows that this ablation leads to a significant drop in accuracy compared to X_{CST} and X_{CS}, especially on NT, where temporal features are most important for detecting fakes. This demonstrates that the temporal component of X_{CST} does meaningfully contribute to its performance and generalization ability. Temporal features more than compensate for the reduced number of neurons for the colour and noise streams.

3.6.4 Training on a Union of Datasets

Since a single dataset can hardly cover all variations of forged video content (synthesis methods, image qualities, lighting conditions, camera angles, etc.), a robust detector should be trained on a *union* of datasets.

To evaluate how well detectors handle such a scenario they were trained on the union of FACEFORENSICS++, VIDEOFORENSICSHQ and DFDC (preview) (Dolhansky et al., 2019), to be tested on FACEFORENSICS++ and VIDEOFORENSICSHQ (see Table 3.5). (Unfortunately, the test accuracies on DFDC (preview) were only about 80% for all detectors,

Detector	Train: FS ∪ NT		Train: F2F ∪ DF	
	Acc.F2F	Acc.DF	Acc.NT	Acc.FS
X_C	82.25%	92.91%	58.40%	50.09%
X_B	90.40%	95.15%	66.03%	50.27%
X_S	98.46%	94.93%	85.21%	55.19%
X_{CS}	**99.46%**	98.74%	86.74%	51.78%
X_{CST}	98.91%	99.03%	**90.65%**	56.77%
$X_{CST\setminus T}$	99.21%	**99.18%**	85.42%	57.65%
MESOINC-4	89.65%	71.94%	73.10%	49.76%
EASYSPOT	77.91%	80.03%	84.40%	**58.89%**
SIMPLEFEATURES	55.87%	55.47%	53.68%	54.19%
MISLNET	64.22%	94.53%	64.04%	50.02%

Table 3.4: All detectors were trained on the training+validation portions of two out of four FACEFORENSICS++ subsets and then tested on the test portions of the other two. Percentages are test accuracies averaged over three runs of the experiment. Especially X_{CST}, which exploits temporal dependencies, achieves high accuracies on the unseen manipulation types. This effect is strongest on the NT, that has the best spatial quality of the four subsets (see Figure 3.2), so the temporal dependencies pay off the most here. $X_{CST\setminus T}$ is an ablated version of X_{CST} (see Section 3.6.2). The FS subset appears to be very hard to generalise to for all detectors.

because the very challenging perspectives and lighting conditions, as well as the fast motion, made the common preprocessing pipeline struggle to the point that it became the bottleneck for accuracy.)

Compared to training on only one single dataset (see Table 3.3), the task is now hard enough to also differentiate the X detectors from one another: X_B again performs better than X_C. X_S and X_{CS} are on par. X_{CST} can once more demonstrate the benefit of temporal information, ranking highest on both test sets.

3.6.5 Impact of Training Corpus Size

VIDEOFORENSICSHQ contains only 45 identities, while FACEFORENSICS++ contains 1000 identities. This raises the question if the number of identities in VIDEOFORENSICSHQ is sufficient to train a good detector.

To answer this question, small training sets were randomly sampled from VIDEOFORENSICSHQ, with different numbers of identities. Detectors were trained on these subsets and then tested on random test sets of 15 identities each (disjoint from the training sets). For each number of training identities, the experiment was repeated 3 to 5 times, and the accuracies were averaged, resulting in the curves in Figure 3.8.

The best detectors achieve close to 100% test accuracy already for train-

Arch.	Test accuracy	
	FACEFORENSICS++	VIDEOFORENSICSHQ
X_C	95.90%	78.99%
X_B	96.16%	80.02%
X_S	97.69%	88.94%
X_{CS}	98.01%	87.91%
X_{CST}	**98.67%**	**90.63%**
MESOINC-4	74.34%	76.65%
EASYSPOT	75.88%	56.75%
SIMPLEFEATURES	54.20%	55.51%
MISLNET	90.02%	78.85%

Table 3.5: Detectors were trained on the union of the training+validation portions of FACEFORENSICS++, VIDEOFORENSICSHQ and DFDC (preview) (Dolhansky et al., 2019). Percentages are detection accuracies on the test portions. Compared to Table 3.3 all detectors perform less well, but the detectors presented in this chapter suffer less than previous ones. X_{CST} handles the diversity of manipulation methods best.

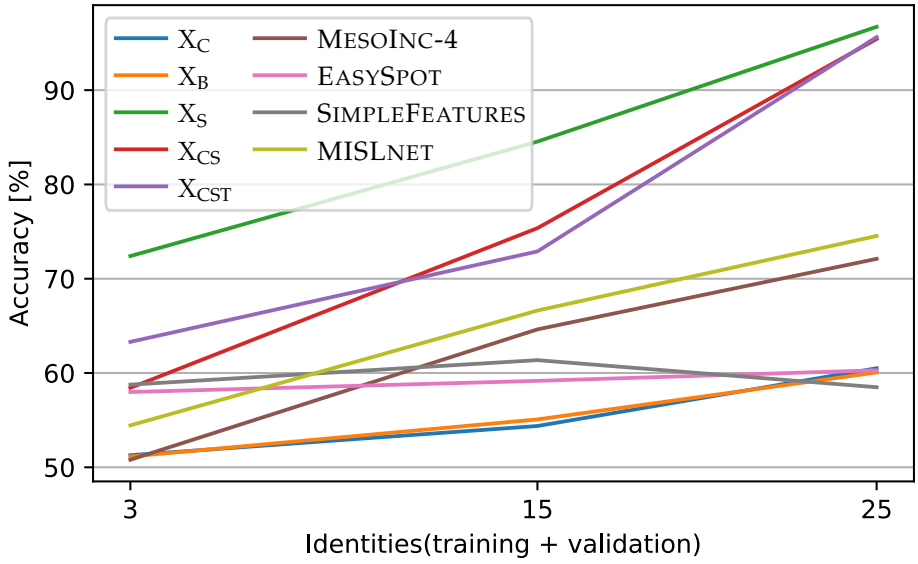


Figure 3.8: Detectors were trained on subsets of VIDEOFORENSICSHQ that contained different numbers of identities (see Section 3.6.5). The resulting test accuracies indicate that X_S, X_{CS} and X_{CST}, i.e. those detectors that look at something other than plain RGB, need much fewer training identities than previous detectors to achieve a high accuracy on the test set.

ing corpora of only 25 identities (training + validation), which is much fewer than the total number of identities in VIDEOFORENSICSHQ, providing evidence that VIDEOFORENSICSHQ is sufficient to generalise to unseen identities, and that the X detectors do *not* overfit to the training identities.

3.7 Limitations

While X_{CST} is one of the first fake detectors to take temporal dependencies into account, its temporal processing (Steps 1 to 7) is still rather primitive: It is designed to detect temporal high-frequency "flickering" and its only learnable parameter is the cut-off amplitude t. However, temporal dependencies of real-world videos are likely much more sophisticated than merely ruling out high-frequency pixel changes. While X_{CST} was sufficient to detect state of the art fakes at the time, it may not be general enough for more recent manipulation methods.

VIDEOFORENSICSHQ is the first dataset of high-quality manipulated face videos, but all its fakes were computed using only one method, DVP, which is based on a convolutional generator. Today's fakes, however, are often the result of diffusion-based approaches (Ho et al., 2020; Rombach et al., 2022; Sohl-Dickstein et al., 2015), that can be controlled (L. Zhang et al., 2023) without alpha blending as a post-processing step. In this sense VIDEOFORENSICSHQ would benefit from extension to further manipulation methods.

3.8 Conclusions

This chapter has introduced VIDEOFORENSICSHQ, the first benchmark for face video detection that provides a large number of manipulations a human would not be able to spot. Only with this dataset was it possible to investigate whether previous approaches to face video forgery detection are ready for the advent of synthesis methods that produce seemingly "perfect" results. This investigation confirmed hypothesis **H**, i.e. it showed that previous detectors struggle to detect fakes that a human would not be able to spot either, although these fakes still do contain traces of artificial genesis.

To compensate for the shortcomings of existing detection approaches in this scenario, Section 3.5 introduced a novel family of detectors that combine spatial and temporal information in a way that has not been used in the area of face video forgery detection before. Section 3.6 showed these detectors to outperform related methods both on previous datasets and on VIDEOFORENSICSHQ.

While at first sight one might mistake the "intra-person" expression transfers in VIDEOFORENSICSHQ to be harmless, there is work (Fried et al., 2019; Suwajanakorn et al., 2017) that demonstrates even slight manipulations of this kind to have dramatic consequences. The absence of human-detectable artefacts in VIDEOFORENSICSHQ has the advantage of preventing detectors from learning to rely on their presence. This suggests that VIDEOFORENSICSHQ can enrich any detector training set.

While this chapter emphasized *analysis* of video footage, and only "simulated" a capable attacker in order to obtain training data of sufficient quality and quantity, the next chapter focuses on the *synthesis* of videos using very small training sets.

Temporal Generation using a Pretrained StyleGAN

4

The previous chapter used existing Generative Adversarial Networks (GANs) to produce training data for fake detectors. In this chapter, the goal is to advance video generation itself: Most generators require an extensive training dataset to learn temporal dependencies, while many of them remain rather limited in the resolution and visual quality of their output. This chapter presents a novel approach to the video synthesis problem (published as Fox et al., 2021b) that helps improve visual quality and drastically reduces the amount of training data and resources necessary for generating videos. Its formulation separates the *spatial* domain, in which individual frames are synthesized, from the *temporal* domain, in which motion is generated. The spatial domain is covered by a pretrained StyleGAN network (Karras et al., 2021, 2019, 2020), the latent space of which allows control over the properties of the objects it was trained for. The expressive power of this model allows for training videos to be embedded in its latent space. The temporal architecture can thus be trained not on sequences of video frames, but on sequences of StyleGAN latent codes. The advantageous properties of the StyleGAN space simplify the discovery of temporal dependencies. It suffices to train the temporal architecture on only 10 minutes of footage of 1 subject. After training, the model can not only generate new portrait videos for the training subject, but also for *any* subject that can be embedded in the StyleGAN space.

4.1 Introduction

Generative Adversarial Networks (GANs; see Section 2.5) have achieved unprecedented levels of output quality for the generation of complex probability distributions. This is especially true for images of human faces, where StyleGAN (Karras et al., 2021, 2019, 2020) can produce photorealistic images of high resolution (e.g. 1024^2). These capabilities, however, did not automatically carry over to the domain of *videos*: While several methods for video generation show promising results in modelling content and motion (Muñoz et al., 2021; Saito et al., 2020; Tian et al., 2021; Sergey Tulyakov et al., 2018; Weissenborn et al., 2020), they usually are

Figure 4.1: Training StyleVideoGAN on one single, short training video (less than 10 minutes) of a human face, makes it learn temporal dependencies that allow the generation of motion for random new subjects, sampled from the latent space of StyleGAN. Each row shows a different new subject, but all frames were generated from the same model. More results in the project video (Section 1.5).

subject to at least a subset of the following limitations: small spatial resolution ($\leq 128^2$); spatial artefacts; constrained motion; necessity of large amounts of training data; large computational cost for training (memory, time; see Table 4.4 and Section 4.4.4) .

This chapter presents StyleVideoGAN, an approach to unconditional video generation that addresses these problems: The goal is to learn a generator for nearly photorealistic, high-resolution videos (up to 1024^2), by training on a video dataset that contains no more than 10 minutes of footage. In addition, although the training footage depicts only a single subject, the trained model should be able to generate motion not only for the training subject, but for many different (random) subjects. The generation of portrait videos is a good proving ground for this method, because portraits are an attractive target for animation and because high-quality training data and StyleGAN models for this domain are readily available. To demonstrate that the method is also applicable to other domains, very different from portraits, Sections 4.4.6 and 4.4.7 show that it can be applied to other object classes as well.

The key idea of StyleVideoGAN is to embed the training video into the latent space of a pretrained StyleGAN model: StyleGAN is a generator for images (see Section 2.5), i.e. it maps latent codes z that are usually obtained as samples of a multivariate Gaussian to frames F. In the course of this computation, the generator first maps z to a new latent

code $w \in \mathcal{W} \subseteq \mathbb{R}^{512}$, that is then copied 18 times, to be consumed by the layers of StyleGAN's main network. Since encoding a given image as a $w \in \mathcal{W}$ has been observed to give rather imprecise reconstructions, one instead lifts the "18 copies" constraint and computes $w \in \mathcal{W}^+ \subseteq \mathbb{R}^{18\times512}$, i.e. the 18 sub-vectors can now differ from one another. Previous works (Abdal et al., 2019; Richardson et al., 2021) have shown this to lead to more precise reconstructions.

Embedding videos in $\mathcal{W}^+$ turns them from sequences of RGB frames into sequences of $\mathcal{W}^+$ vectors. While an RGB frame has $1024 \cdot 1024 \cdot 3 = 3{,}145{,}728$ dimensions, a $\mathcal{W}^+$ code only has $18 \cdot 512 = 9216$ dimensions. This means that the embedding allows the temporal generator to be supervised in a much lower-dimensional space, which simplifies the discovery of temporal dependencies. Also, this transformation completely eliminates the necessity to actually render any video frames at training time, which greatly reduces the amounts of memory and time required to train the model.

There is another major advantage of the embedding approach, that allows the model to generate motion for a great multitude of subjects, even though it has seen only one subject at training time: The linear separability properties of $\mathcal{W}^+$ space (Nitzan et al., 2020; Shen et al., 2020; Tewari et al., 2020a) allow for motion to be transferred from one subject to another, unseen subject, which the chapter will refer to as the "**offset trick**".

Using these ideas, it is possible to train a Wasserstein GAN (Arjovsky et al., 2017; see Section 2.5) for the generation of high resolution videos using a minimal amount of training data and computational resources. The temporal windows seen at training time are already longer (25 time steps) than those possible in many previous methods, but it is desirable to be able to generate videos that are considerably longer at test time. This can be achieved by making the generator a recurrent neural network (RNN). As previous work (Tian et al., 2021) has pointed out, a standard RNN may tend to produce "looping" motion, i.e. repeat the same motion pattern over and over. StyleVideoGAN addresses this problem with a novel **gradient angle penalty**. In summary, this chapter comprises the following contributions:

- A novel approach to unconditional video generation that is supervised in the latent space of a pretrained image generator, without having to render video frames at training time, leading to large savings in computational resources.

- The first video generation approach to exploit the properties of StyleGAN's $\mathcal{W}^+$ space, greatly reducing the demand for training data.

- A novel gradient angle penalty loss that helps generate videos that are longer than the temporal windows seen at training time.

4.2 Related Work

4.2.1 Generative Models for Videos

Blattmann et al., 2023a,b, provide a comprehensive overview of previous works on video generation. This section focuses on the state of the art up to (and including) 2021.

Recurrent neural networks (RNNs), which the GRU-based generator in this thesis is an example of, have been been used in several earlier works, to model the temporal dimension: Babaeizadeh et al., 2018 and E. Denton et al., 2018, presented early methods of modelling videos by predicting an entire distribution of possible futures given a suffix of the past. It was shown that acknowledging this stochastic nature of videos (i.e. the same limited information about the past admits many different futures of different probability) improves the quality of the output. Castrejón et al., 2019, showed that the performance of these and other previous methods is limited by model capacity and introduced a hierarchical variational model to improve the representation of distributions. Franceschi et al., 2020, aim at separating static information about the video from temporally variable information, but, like other methods mentioned in this section, supervise their model in the image domain, which is computationally expensive and thus limits them to much smaller resolutions than the method presented in this thesis. This constraint is shared by notable methods based on normalizing flows (Blattmann et al., 2021; Dorkenwald et al., 2021), which require large video datasets and generate output only at low resolutions.

Weissenborn et al., 2020, generalised transformers (Vaswani et al., 2017) to a three-dimensional self-attention mechanism, in order to construct an autoregressive video generation model. Inspired by the work of Menick et al., 2019, they produce videos as sequences of lower-resolution slices, to reduce computational complexity. Nevertheless, spatial resolution is only 64^2 and sampling a video of only 30 frames takes about 2 minutes according to the authors.

Most related to the work presented in this thesis are **GAN-based approaches** (Goodfellow et al., 2014) to video generation. Many of these are based on the Wasserstein GAN architecture (Arjovsky et al., 2017; Gulrajani et al., 2017), that has been shown to be able to prevent mode collapse. Villegas et al., 2017, aim to disentangle the content of videos from their motion, by training separate encoders for content and motion respectively. The output from the encoders is combined to predict the next frame following a sequence of input frames. The output resolutions are again rather small. This is also the case for the closely related work

by E. L. Denton et al., 2017, who introduce an adversarial loss that makes an encoder separate information about frames into a "content" part, that should distinguish a given video from other videos, and a "pose" part, that cannot distinguish different videos (because the same poses can occur in different videos). The approach by Saito et al., 2017, is based on a Wasserstein GAN with a novel parameter clipping method. The architecture uses a shared image generator for each frame, but it is not pretrained, and supervision is defined on the image domain. Output resolution is fixed to 64^2. Sergey Tulyakov et al., 2018, presented MoCoGAN, decomposing the generation of videos into a motion part, the state of which is evolved by an RNN throughout the sequence, and a content part, that remains fixed over the sequence. MoCoGAN is trained using separate discriminators for content and motion. Yushchenko et al., 2019, formulated video generation by means of Markov Decision Processes (MDP), extending the MoCoGAN framework. They identify "video freezing" and "video looping" as important flaws in generated videos. J. Wu et al., 2019, starting at small spatiotemporal resolution, progressively grow their GAN model (Karras et al., 2018). This process enables, as one of the first methods ever, the generation of video distributions at resolution 256^2. Separating video content into appearance and motion, Y. Wang et al., 2020, present a GAN with a 3-stream convolutional generator that receives dedicated latent codes for appearance and motion. Its discriminator architecture is similar to that of MoCoGAN. The generated videos are limited to a fixed duration (default 16 frames) and their resolutions are small (64^2). Ye et al., 2020, present a method for the unconditioned generation of face videos. They emphasize the generation of long-duration videos, but spatial resolutions are small (128^2). Kahembwe et al., 2020, show that three-dimensional convolution kernels in a video discriminator make the loss landscape unnecessarily hard to optimise in. Instead they propose a family of lower-dimensional kernels and apply them to the discriminators of MoCoGAN and the work of Saito et al., 2017, which improves their ability to model the training distribution. The method is the first to generate videos at spatial resolution 512^2. Saito et al., 2020, decompose the generation problem into multiple generators, where earlier generators "receive high-frame-rate, low-resolution videos" as input and later generators receive "low-frame-rate, high-resolution videos" as input. This makes computational complexity linear in the resolution and thus allows the authors to show resolutions of up to 256^2. Concurrently to the method presented in this thesis, Yao et al., 2021, introduced a method for the editing of human faces in videos, also exploiting the properties of the StyleGAN latent space. However, this approach merely preserves the motion given in the input, performing spatial edits in a temporally smooth way, but does not learn to generate new motion. K. Hong et al., 2021, train their discriminator to infer the "arrow of time" as an auxiliary

task, i.e. to determine whether the frames of a video are in forward, or backward temporal order. The generator learns to produce frames in forward order, but the gradient of the arrow of time estimated by the discriminator serves as an additional corrective, forcing the generator to model temporal dependencies more accurately. The authors show output resolutions of up to 256^2. Muñoz et al., 2021, similar to StyleVideoGAN, model videos as paths through a latent space, with each point in the latent space representing one frame. Their generator consists of a sequence generator and a frame generator. Supervision by the discriminator happens on the basis of RGB frames and resolutions up to 192^2 are shown.

Summary All methods listed in this section, and especially the GAN-based approaches, have in common that models are supervised in the image domain. This means that at training time, full frames for all the videos in a training batch need to be synthesized, which costs memory and computation time. Furthermore they need to be analysed by the discriminator, which again is very expensive both in terms of time and space. Especially the space cost is a limiting factor here, as the available GPU(s) will set a limit on the amount of memory that can be used for a training iteration and thus constrains the maximum video resolution that models can be trained for: Previous methods produce resolutions of 64^2 (Saito et al., 2017; Sergey Tulyakov et al., 2018), 128^2 (Ye et al., 2020), 192^2 (Muñoz et al., 2021), 256^2 (K. Hong et al., 2021; Saito et al., 2020; J. Wu et al., 2019) or 512^2 (Kahembwe et al., 2020). Furthermore, many approaches struggle with generating realistic videos of longer durations: Often the number of generated frames is fixed at training time, to numbers as low as 16 (Muñoz et al., 2021; Saito et al., 2017, 2020; Sergey Tulyakov et al., 2018), or 32 (J. Wu et al., 2019) frames.

It is possible to offload some of the memory allocations to the machine's main memory, but this makes implementations more complex and there does not seem to be any previous work that makes use of this possibility, one likely reason being that this would only soften the space constraints, but not reduce the runtime cost. The biggest difference of StyleVideoGAN to all previously mentioned methods is that at training time, no frames need to be rendered, which drastically reduces both space and time demands and thus allows for much larger resolutions of 1024^2.

Tian et al., 2021, presented the only method that can generate videos at this same resolution, 1024^2. The authors formulated video generation as the problem of finding a suitable trajectory through the latent space of a pretrained and fixed image generator, such as StyleGAN. Despite this commonality and their ability to produce high-resolution output, there are a number of important differences between their approach and the work presented in this chapter:

- Their discriminator supervises the generator in the image domain, which is a much higher-dimensional and more redundant domain than $\mathcal{W}^+$.

- Their design inherently relies on the image generator being available for forward and backward passes at training time, which increases the required amounts of GPU memory and computation time immensely, compared to StyleVideoGAN.

- Their method requires a diverse training set. When trained on a single video, their results show very limited motion, as demonstrated in Section 4.4.

4.2.2 StyleGAN Inversion & Latent Editing

StyleGAN (Karras et al., 2018, 2019, 2020) is a GAN-based image generator that at the time the work presented in this chapter was conducted represented the state of the art in image generation, due to its unprecedented spatial resolution and photorealism, especially for human faces. Apart from the outstanding quality of the images it generates, multiple works (Härkönen et al., 2020; Nitzan et al., 2020; Shen et al., 2020; Tewari et al., 2020a) have investigated the properties of its latent space and found it to be very suitable for editing. This is because for many "one-dimensional" properties, one can find a hyperplane (Shen et al., 2020) in the StyleGAN latent space that is (approximately) perpendicular to the direction along which this property changes. For example, in the case of human faces, if the property is "age", one can find a direction in the latent space along which faces get older, whereas they get younger in the opposite direction. It is this property that StyleVideoGAN uses in order to generalise motion from the single training identity to new, unseen identities.

However, to be able to train StyleVideoGAN, one must be able to embed the training footage into the latent space of the StyleGAN model, frame by frame. Various methods (Abdal et al., 2019, 2020; Pidhorskyi et al., 2020; Richardson et al., 2021; Tewari et al., 2020b; Jiapeng Zhu et al., 2020) exist for this embedding. They can broadly be divided into two categories, namely those that find latent codes by optimization (Abdal et al., 2019, 2020; Tewari et al., 2020b) and those that learn to *encode* the input image as a latent code with one single forward pass (Pidhorskyi et al., 2020; Richardson et al., 2021; Jiapeng Zhu et al., 2020). The latter have the advantage of being faster, albeit at the expense of some precision. In this category, pSp (Richardson et al., 2021) maps input images to $\mathcal{W}^+$. StyleVideoGAN uses a pretrained, fixed instance of pSp, that leads to temporally smooth & consistent reconstructions of videos even though it is applied to each frame in isolation.

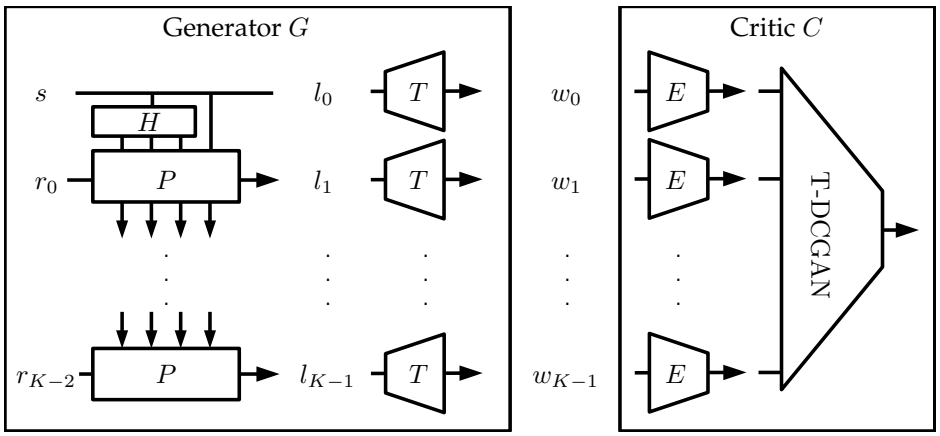


Figure 4.2: StyleVideoGAN consists of a generator G and a critic C. G contains a recurrent producer based on GRU cells, and a translator that maps motion latent codes to StyleGAN's $\mathcal{W}^+$ codes (Section 4.3.2). C is only needed at training time and so does not need to be recurrent (Section 4.3.3).

4.3 Method

StyleVideoGAN (Figure 4.2) is a Wasserstein GAN (Arjovsky et al., 2017; Section 2.5), consisting of a generator G (Section 4.3.2) and a critic C (Section 4.3.3).

To generate a video of K frames, G receives a pair (s, r) as input, with $s \sim \mathcal{N}(0;1)^{32}$ and $r \sim \mathcal{N}(0;1)^{32\times(K-1)}$. At training time $K := 25$, but since the generator is an RNN, K can take different values at test time. The output of G is a sequence of K latent codes $w_i \in \mathcal{W}^+$, with $0 \leq i < K$. Before training, pSp (Richardson et al., 2021), an encoder-based inversion method for StyleGAN, embeds the training video in $\mathcal{W}^+$ space. This embedding is the source of "real" samples for the critic to distinguish from the generator's output. No frames are rendered during training; StyleGAN is absent. This leads to considerable savings in training time and memory consumption, in particular compared to the method by Tian et al., 2021 (see Section 4.4). Only at test time is the output of G fed into a StyleGAN instance.

Although the most extensive results in this chapter are for portrait videos, no part of StyleVideoGAN other than the preprocessing step (Section 4.3.1) is inherently face-specific: Sections 4.4.6 and 4.4.7 show its application to other object classes.

Figure 4.3: Projecting face videos into $\mathcal{W}^+$ with the help of pSp reasonably maintains the identity of the actor and leads to temporally smooth results. The input video is from the YouTube subset of VideoForensicsHQ.

4.3.1 Data Preprocessing

Training data is derived from 1 single video of 1 to 10 minutes duration. Before embedding its frames in $\mathcal{W}^+$ via pSp, they are preprocessed in a similar way that the training data for the respective StyleGAN model has been preprocessed. In the case of faces this means computing face crops like for the FFHQ dataset (Karras et al., 2019). However, since FFHQ is an image dataset without a temporal dimension, no mouth keypoints (which move too much) but only eye corners are used for alignment, and temporal low-pass filters are applied to the rotation and scale of the face bounding boxes. Applying pSp to each frame of the training video reliably leads to sequences of $\mathcal{W}^+$ codes that, when rendered with StyleGAN, give a temporally smooth video again (Figure 4.3). The identity of the training subject is not always preserved perfectly, but this is negligible if the goal is to generate motion for a great multitude of generated identities.

4.3.2 Generator

The generator contains a **producer stack** P of 4 GRU cells (K. Cho et al., 2014) that process the **per-time-step randomness** r_i. To initialise the GRU memory, the MLP H "hallucinates" some memory contents for the first three cells, whereas the last is initialised with the **initial randomness** s:

$$(h_{0,0}, h_{0,1}, h_{0,2}) := H(s) \qquad h_{0,3} := s$$

After this initialization, P can produce a sequence of low-dimensional latent codes $l_i \in \mathbb{R}^{32}$ according to the following recurrence:

$$((h_{i+1,0}, \ldots, h_{i+1,3}), l_{i+1}) := P(r_i, (h_{i,0}, \ldots, h_{i,3}))$$

for $0 \leq i < K$. The **translator** $T : \mathbb{R}^{32} \to \mathbb{R}^{512}$ (also an MLP; reminiscent of StyleGAN's mapping network) maps these latent codes from the space in which motion is generated to a higher-dimensional one. A set of learned affine transformations then maps to $\mathcal{W}^+ \subseteq \mathbb{R}^{18\times512}$, which gives the final output $w_0, \ldots, w_{K-1} \in \mathcal{W}^+$ of G.

The design of G is not particularly novel: Previous works (e.g. Muñoz et al., 2021; Tian et al., 2021; Sergey Tulyakov et al., 2018) have presented similar designs. The novelty of StyleVideoGAN lies in the ideas outlined in Section 4.1, i.e. supervision in $\mathcal{W}^+$ instead of image space, the "offset trick", the loss functions, and the resulting massive reduction in the amount of data and resources required for training: Only at test time, not at training time, are $w_0, \ldots, w_{K-1}$ forwarded to StyleGAN for rendering of actual video frames. K can then be considerably larger than the 25 time steps used at training time. Section 4.4 reports results for $K = 250$.

4.3.3 Critic

In contrast to G, the critic C is only used at training time and can therefore rely on a fixed K: A 6-layer **extractor** MLP $E : \mathcal{W}^+ \to \mathbb{R}^{32}$ maps $\mathcal{W}^+$ codes to a learned space of relevant features, that are then fed into a temporally convolutional network. The convolutional part is derived from DCGAN (Radford et al., 2016), turning spatial convolutions into temporal ones and eliminating batch normalization layers. The latter is required for the WGAN-GP objective (also known as "gradient penalty") (Gulrajani et al., 2017) that enforces the Lipschitz constraints of the Wasserstein GAN (Section 2.5).

4.3.4 Loss Terms & Training

Training minimises the loss term

$$\mathcal{L} = \mathcal{L}_{\text{WGAN}} + \lambda_{\text{GP}}\mathcal{L}_{\text{GP}} + \lambda_{\text{GAP}}\mathcal{L}_{\text{GAP}} \tag{4.1}$$

where $\mathcal{L}_{\text{WGAN}} + \lambda_{\text{GP}}\mathcal{L}_{\text{GP}}$ is the WGAN-GP loss (Gulrajani et al., 2017; $\lambda_{\text{GP}} = 50$) and $\mathcal{L}_{\text{GAP}}$ is a novel **gradient angle penalty** (with $\lambda_{\text{GAP}} = 100$):

Training StyleVideoGAN only with the WGAN-GP loss (see ablation study in Section 4.4.5) shows that synthesizing videos for $K > 25$ can lead to outputs that seem to be **looping**, i.e. the same motion pattern is repeated over and over. As observed in previous work (Tian et al., 2021), P learns to simply ignore the per-time-step randomness r_i and to rely exclusively on $(h_{i,0}, \ldots, h_{i,3})$, without modifying it much in the course of the sequence. This means that there is a tendency to make s determine the entire output, which makes looping very likely, because the information contained in s has only ever been supervised in the first K frames generated at training time and therefore may suffice only for

that much unique output. To counteract this, the gradient angle penalty makes sure that the gradient of the producer output with respect to r_i is at least a certain fraction of the gradient with respect to s:

$$\mathcal{L}_{\text{GAP}} := \max\left(0, \frac{\pi}{4} - \varphi\right)^2 \text{ for } \varphi := \arctan\left(\frac{\left|\left|\left[\frac{\partial d}{\partial r_0}, \ldots, \frac{\partial d}{\partial r_{K-2}}\right]\right|\right|}{||\frac{\partial d}{\partial s}||}\right) \quad (4.2)$$

where $d := \text{dnorm}(l_{K-1} - l_0)$ is the per-dimension-normalised difference between the last time step and the first time step generated by P: dnorm normalises each component of its input vector independently from the other components, according to running statistics that are tracked during training, such that its output can be expected to have mean 0 and variance 1 in each component.

Unless stated otherwise, models were trained using the ADAM optimiser (Kingma et al., 2015) for 350 epochs, with exponentially averaging the weights of the generator throughout training using a momentum of 0.995.

4.3.5 The Offset Trick

Although STYLEVIDEOGAN is trained on only 1 single subject, it should be able to generate motion for a large set of randomly sampled actors. This can be achieved by making use of the advantageous properties of STYLEGAN's $\mathcal{W}^+$ space, that have been used for face editing before (Härkönen et al., 2020; Shen et al., 2020; Tewari et al., 2020a): Given a point in $\mathcal{W}^+$, the directions into which one would need to shift this point in order to change the identity of the subject are mostly orthogonal to those directions that would change the pose/expression/articulation. It should thus be possible to first generate a motion trajectory for the training subject and then shift this trajectory along a direction that is orthogonal to those latter directions, to transfer it to a different subject that also exists in $\mathcal{W}^+$.

To find the directions responsible for pose/expression/articulation it suffices to conduct a Principal Component Analysis (PCA) of the $\mathcal{W}^+$ embedding of the frames. This yields the 32 directions in which the point cloud representing the training frames extends the furthest. Since these training frames span the relevant range of motion states but always show the same subject, it can be assumed that shifting points in these directions changes the state, but not the identity of the face. Given the PCA basis and having sampled a motion trajectory $w_0, \ldots, w_{K-1} \in \mathcal{W}^+$ for the training subject, one can randomly sample a point from STYLEGAN's $\mathcal{Z}$ space, render it using STYLEGAN and then embed it in $\mathcal{W}^+$ using PSP, obtaining w_{new}. This new point shows a random new subject, that already is in a particular (likely non-neutral) state: For example, in the case of faces, w_{new} might correspond to a person with their mouth closed. One must

Model	Reference	FID (↓)		FVD (↓)	
		Short	Long	Short	Long
StyleVideoGAN	Original	54.1 ±0.1	**54.2 ±1.2**	627.6 ±25.5	**629.1 ±24.7**
	$\mathcal{W}^+$	**1.1 ±0.1**	**3.9 ±1.5**	42.9 ±12.9	**84.0 ±17.7**
StyleVideoGAN $\setminus \mathcal{L}_{\text{GAP}}$	Original	**53.9 ±0.5**	58.8 ±4.8	**603.7 ±39.4**	727.1 ±159.1
	$\mathcal{W}^+$	**1.1 ±0.0**	7.0 ±4.2	**33.2 ±3.7**	178.4 ±94.4
Tian et al., 2021	$\mathcal{W}$	4.07	97.9	706.3	2130.3
Sergey Tulyakov et al., 2018	Original	87.9	87.6	2849.3	2845.1
Saito et al., 2020	Original	108.8	169.0	1211.4	2339.2
Muñoz et al., 2021	Original	75.5	-	755.4	-

Table 4.1: Models were trained on footage of subject #1, to generate a set of "short" videos and a set of "long" videos (see Section 4.4.1). FID and FVD scores compare the generated output distributions to the training distribution. In the case of StyleVideoGAN and the work by Tian et al., 2021, StyleGAN re-renderings (from $\mathcal{W}$ or $\mathcal{W}^+$) are used for computing scores, to factor out the imperfection of pSp embedding. The scores for StyleVideoGAN are averages and standard deviations for 5 repetitions of the experiment. Tian et al., 2021, kindly trained their model on the data that was sent to them.

not naively use this point as the starting point for a "transferred" motion trajectory, because the motion generated for the training actor might start with a mouth-closing motion. Applying this motion to a mouth that is already closed would likely lead to strong artefacts. Instead one projects w_{new} onto the PCA basis, resulting in w'_{new}. The point w'_{new} represents the training actor in the same state as the new actor. The difference $\Delta := w_{\text{new}} - w'_{\text{new}}$ is the exact offset by which to shift the motion trajectory, i.e. the new trajectory is $w_0 + \Delta, w_1 + \Delta, \ldots, w_{K-1} + \Delta$.

As illustrated in Figure 4.4, thanks to the disentangled representation of images in $\mathcal{W}^+$, this simple offset operation is sufficient to transfer motion generated for the training subject to new random subjects. The project video (see Section 1.5) shows that not embedding the new actor with pSp or naively offsetting the sequence without using the PCA basis leads to much stronger artefacts.

4.4 Results

4.4.1 Training Data & Metrics

For ablation studies and comparisons to previous work, videos of subjects speaking into a commodity RGB camera were used, all less than 10 minutes long. Some of these are depicted in Figure 4.5. For quantitative evaluation of trained models, **Fréchet Inception Distance** (FID; Heusel et al., 2017) rates *spatial* quality, while **Fréchet Video Distance**

Figure 4.4: First row: motion trajectory generated for the training subject. **Second row**: Result of naively shifting this trajectory to a point sampled from $\mathcal{W}$. Since the structure of $\mathcal{W}$ differs from that of $\mathcal{W}^+$, the transfer result looks very unnatural. **Third row**: First rendering the $\mathcal{W}$ code with StyleGAN and then embedding it as $w_{\text{new}} \in \mathcal{W}^+$ with pSp before shifting the trajectory to the result gives a much more natural face. **Fourth row**: Adjusting the direction of the shift by projecting w_{new} onto the PCA basis (see Section 4.3.5) leads to better alignment of the target actor with the source actor.

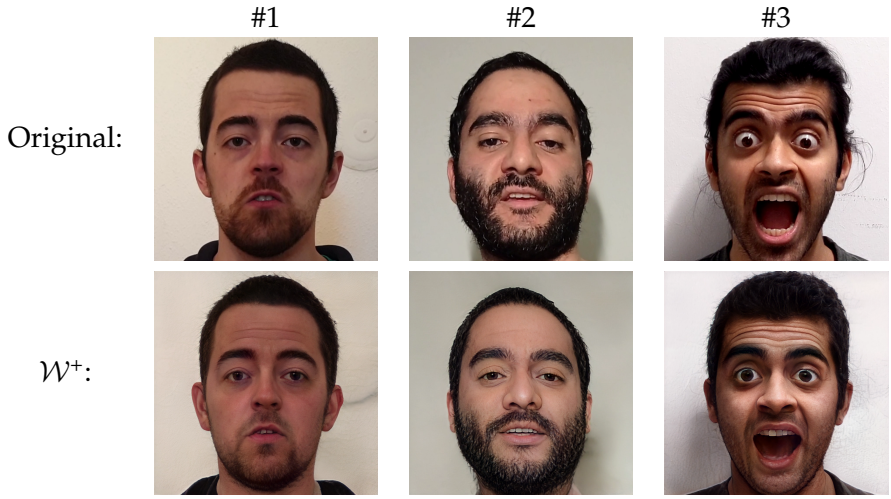


Figure 4.5: Footage of the subjects #1, #2 and #3 was used for the quantitative evaluation. The top row shows aligned crops from the original RGB frames, while the bottom row shows their $\mathcal{W}^+$ embeddings (see Section 4.3.1).

Model	Reference	FID (↓)		FVD (↓)	
		Short	Long	Short	Long
StyleVideoGAN	Original	62.9 ±0.2	65.1 ±0.8	846.7 ±19.1	944.1 ±51.7
	$\mathcal{W}^+$	**0.6 ±0.0**	**2.0 ±0.2**	**39.2 ±19.2**	**51.8 ±18.1**
StyleVideoGAN \ $\mathcal{L}_{\text{GAP}}$	Original	62.9 ±0.1	64.8 ±3.0	848.8 ±8.0	926.2 ±58.5
	$\mathcal{W}^+$	0.7 ±0.0	4.3 ±2.8	57.0 ±93.1	110.7 ±74.9
Sergey Tulyakov et al., 2018	Original	76.5	77.8	1318.7	1338.7
Saito et al., 2020	Original	41.5	51.6	640.0	935.2
Muñoz et al., 2021	Original	46.4	-	578.3	-

Table 4.2: The same experiment as in Table 4.1, but for subject #2.

Model	Reference	FID (↓)		FVD (↓)	
		Short	Long	Short	Long
StyleVideoGAN	Original	52.7 ±0.2	53.7 ±0.5	589.2 ±9.7	625.3 ±3.1
	$\mathcal{W}^+$	3.7 ±0.2	**4.8 ±0.4**	61.4 ±4.2	**98.6 ±11.3**
StyleVideoGAN \ $\mathcal{L}_{\text{GAP}}$	Original	52.3 ±0.5	55.2 ±2.1	590.9 ±15.2	679.0 ±28.9
	$\mathcal{W}^+$	**3.4 ±0.1**	8.2 ±2.0	**54.0 ±3.6**	133.5 ±28.5
Sergey Tulyakov et al., 2018	Original	123.0	141.1	1163.3	1500.3
Saito et al., 2020	Original	82.1	270.3	823.9	2090.8
Muñoz et al., 2021	Original	83.3	-	1037.0	-

Table 4.3: The same experiment as in Table 4.1, but for subject #3. This subject was not talking, but instead performing some simple face motions in a random order (like smiling or acting surprised). The training video is only 1 minute and 20 seconds in length.

(FVD; Unterthiner et al., 2018) rates the quality of motion. The reference sets for all methods are their training datasets, preprocessed as required by the particular method. For StyleVideoGAN both the original RGB frames, as well as the StyleGAN output for the $\mathcal{W}^+$ embedding are used as references in Tables 4.1 to 4.3.

Each method was trained on the training video, depicting only one subject. Then two sets of videos were sampled from each model: The "Short" set consists of 2048 videos that are as long as the temporal window the respective method saw at training time (see column "K" in Table 4.4). The "Long" set consists of 128 videos that all have at least 128 frames. The technique by Muñoz et al., 2021, is not able to produce samples longer than its training window, which is why Tables 4.1 to 4.3 contain no numbers for this set. FID scores are computed on 8000 frames randomly sampled from the reference and generated sets. FVD scores are computed on 2048 videos from each of the two sets, with the duration of the videos again equal to the default temporal window length of each method.

The temporal consistency of facial identity, i.e. the question whether different frames of the generated video depict the same person, was assessed using a variant of the **Average Content Distance** (**ACD**; (Sergey Tulyakov et al., 2018)): For each generated frame, identity features are extracted with a popular facial recognition library (Geitgey et al., 2020) and the average L2-distance between all pairs of frames in a video constitutes the ACD score of that video.

4.4.2 Training Details

All methods were trained with their default hyper-parameters, except for that by Saito et al., 2020, where batch size was set to 2 and `clstm channels = 512`. Tian et al., 2021, kindly trained their technique on the training data sent to them. All methods were trained with at least the

computational resources that were available to StyleVideoGAN, but usually for much longer training time.

Each training video contains 1 single actor/object. For the proof of concept on hands, all training data was recorded from one actor. Training used a batch size of 128, learning rate 0.005, and exponential weight averaging with momentum 0.997 in all experiments. All StyleVideoGAN models were trained for 350 epochs.

4.4.3 Evaluation Details

FID scores were computed by sampling 8000 frames from both the training set (as preprocessed for the respective method) and the set of generated videos.

FVD scores were computed by sampling 2048 video slices from both the training set (as preprocessed for the particular method) and the set of generated videos. For each method, regardless of the length of the generated samples ("short" versus "long"), the slices used for score computation were always 25 frames long for StyleVideoGAN and 16 frames long for the previous methods.

ACD scores were computed always on 128 "long" samples generated by the trained models. For StyleVideoGAN these long samples were 400 frames long. For Tian et al., 2021, they were 128 frames long. Achieving a good ACD score becomes harder as sequences grow longer.

4.4.4 Video Generation

Figure 4.6 shows sequences generated by three different models, each for the respective training identity. As shown in Figure 4.1 however, the "offset trick" allows StyleVideoGAN to generate motion for randomly sampled identities as well, despite the training set always containing only 1 actor. All videos are synthesized at a resolution of 1024^2 and even though StyleVideoGAN was trained only on a temporal window of 25 frames, it can easily generate videos that are much longer, e.g. 1500 frames.

Comparison to Previous Methods StyleVideoGAN is compared to previous approaches by training them all on the training set and evaluating the metrics described above:

The method by Tian et al., 2021, also generates a trajectory in the StyleGAN latent space, making it the most related to StyleVideoGAN. The model evaluated here was kindly trained by the authors. It does, by default, not generate videos of the training identity, but instead samples random identities from StyleGAN's $\mathcal{W}$ space (*not* $\mathcal{W}^+$!). This is a problem for the computation of FID and FVD scores, which always compare

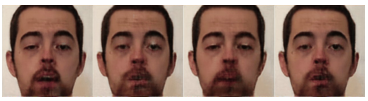
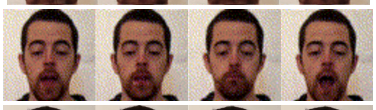
Method	Sample	Res.	K
StyleVideoGAN		1024^2	25
Tian et al., 2021		1024^2	16
Muñoz et al., 2021		128^2	16
Sergey Tulyakov et al., 2018		64^2	16
Saito et al., 2020		192^2	16

Table 4.4: All methods were trained on subject #1 (see Figure 4.5) and a sample was generated from the trained model. StyleVideoGAN and the method by Tian et al., 2021, can synthesize motion for unseen identities, while the other methods are not capable of such generalization. StyleVideoGAN is supervised only in $\mathcal{W}^+$ and not, like all other methods, in RGB space, and thus reduces its resource demands such that it supports the highest spatial and temporal resolutions at training time and picks up more characteristic motion. According to their publications, Muñoz et al., 2021, support spatial resolution 192^2 and Saito et al., 2020, support 256^2, but the resolutions given in the table were the ones used for the experiment. Temporal quality can only be judged by the project video (see Section 1.5).

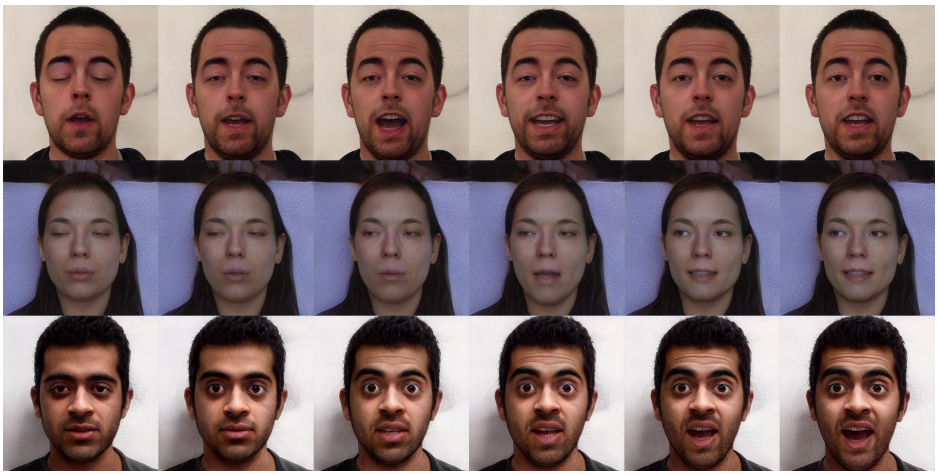


Figure 4.6: After training StyleVideoGAN on one single, short training video (less than 10 minutes) it has learned temporal dependencies that enable it to generate convincing output videos. Each row shows consecutive output frames after training on one single subject.

the generated distribution (random identities) to the training distribution (fixed training identity). The model thus needed to be forced to generate samples depicting the training identity, as otherwise it would have been impossible to compute fair scores. This is achieved by sampling random frames from the training video and projecting (Karras et al., 2019) them to $\mathcal{W}$. The resulting $\mathcal{W}$ points are then "injected" as the initial code for the model to condition its generated sequences on. As mentioned in previous work (Richardson et al., 2021; Shen et al., 2020), $\mathcal{W}$ cannot represent real images as faithfully as $\mathcal{W}^+$, which is why instead of the training video, its re-rendering from the $\mathcal{W}$ codes was used as the reference set for score computation. While the method is able to produce videos at resolution 1024^2, the model the authors trained on the given data does not generate a lot of facial motion, i.e. while the camera is panning, the facial expression is very static. This is reflected in Table 4.1. Training this method on larger video datasets requires, according to the publication (Tian et al., 2021), about 5 days on 8 Quadro RTX 8000 GPUs for a resolution of 1024^2, i.e. 40 GPU days in total. StyleVideoGAN is trained on a single Quadro RTX 8000 GPU in around 6 hours. While these numbers are not directly comparable (large video dataset for Tian et al., 2021, versus a singleton training set for StyleVideoGAN), the much larger memory demand (8 GPUs versus 1 GPU), and the limited variety of motion despite much more diverse training material (see also project video; Section 1.5) do show that StyleVideoGAN is computationally more efficient.

The methods by Saito et al., 2020, and Sergey Tulyakov et al., 2018, generate realistic motion, but are limited in terms of spatial resolution

(256^2 and 64^2 respectively). As shown in the project video (see Section 1.5), the results of the method by Muñoz et al., 2021, include strong structural artefacts. All three methods are not capable of generalizing their output to random identities after being trained on only one subject, i.e. their training set would have to be much larger to make them generate the diversity of output identities that Tian et al., 2021, and StyleVideoGAN achieve.

Tables 4.1 to 4.3 report Fréchet distances for all methods. In addition, ACD scores were computed for StyleVideoGAN (random actors) and for the method by Tian et al., 2021: While StyleVideoGAN averages at 5.68 (over 5 models) Tian et al., 2021, achieve a score of **0.54**. This large difference can be explained by the very limited facial motion generated by Tian et al., 2021, that of course makes it much easier to preserve the identity across frames. For a visual impression of this observation see the project video (Section 1.5).

4.4.5 Evaluation of the Gradient Angle Penalty

Tables 4.1 to 4.3 show that while disabling $\mathcal{L}_{\text{GAP}}$ slightly improves the scores for "short" samples, it can considerably increase the FVD scores for "long" samples. However, since the primary purpose of $\mathcal{L}_{\text{GAP}}$ is to prevent looping, which may not be effectively captured by FVD, a very short training training sequence (one single sentence, spoken three times, 20 seconds in total) was recorded, that provoked strong looping artefacts in 19 out of 20 independently trained models if $\mathcal{L}_{\text{GAP}}$ was absent. The same sequence led to looping only in 6 out of 20 models that were trained with $\mathcal{L}_{\text{GAP}}$ enabled. This suggests that $\mathcal{L}_{\text{GAP}}$ is indeed making looping artefacts much less likely.

4.4.6 Proof of Concept: Hands

To demonstrate that StyleVideoGAN can in principle be applied to content categories other than talking faces, a proof-of-concept experiment for hands was conducted: The right hand of a subject was recorded for 1 hour, performing various types of motions (like showing numerals or performing a set of gestures), resulting in a dataset of around 100k frames. The only constraint was for the hand to always turn the palm to the camera and to never leave the recording space. This dataset was used for training a StyleGAN model and the corresponding pSp inverter, both for resolution 256^2. With these models available, StyleVideoGAN could be trained with a temporal window of 75 time steps, on several test sequences (each about 8000 frames). Generated frames are shown in Figure 4.7 and in the project video (see Section 1.5).

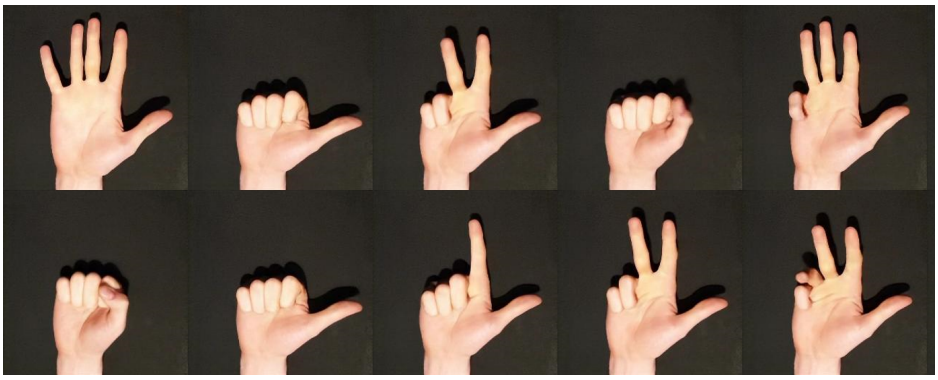


Figure 4.7: After recording enough footage of hand motions to train StyleGAN and pSp on it, StyleVideoGAN can be trained to generate new sequences of the hand moving. Temporal quality can be assessed in the project video (see Section 1.5).

4.4.7 Proof of Concept: Cars

As a third domain, StyleVideoGAN was applied to the category of cars: Using the official StyleGAN 2 checkpoint for the LSUN-Car dataset (F. Yu et al., 2015), pSp was trained from scratch. LSUN-Car is a much more challenging dataset than FFHQ, because the data is not as "clean" and because different angles of cars cannot be aligned with each other. pSp did thus not converge to the same level of precision as the official FFHQ checkpoints and the $\mathcal{W}^+$ embeddings contained clearly visible artefacts, with the identity of the car drifting depending on the orientation. In order to nevertheless demonstrate the feasibility of the core concept of StyleVideoGAN, pSp was thus trained a second time, on the frames of the car recordings. This easily achieves decent embeddings, showcased in Figure 4.8 and in the project video (see Section 1.5). The disadvantage of this approach is that it cannot demonstrate the offset trick, as the pSp model has only ever seen the training car and cannot embed cars randomly sampled from StyleGAN's latent space. Of course it would have been preferable to use an object category (other than faces) for which there is a *general* high-quality, temporally stable embedding method. However, to the best of the author's knowledge, nobody had demonstrated such a method at the time StyleVideoGAN was developed.

4.5 Limitations

Even though StyleVideoGAN improves the state of the art in video generation, in particular with respect to the amounts of computational resources and training data necessary to generate a large amount of di-

Figure 4.8: To show that StyleVideoGAN is not face-specific, a video of a car (top) was recorded. Its frames were embedded into $\mathcal{W}^+$ (centre) using a pSp model trained on the recorded video (because a more general pSp model of sufficient quality was not available). StyleVideoGAN was then trained on this embedding, allowing the generation of new motion around the car (bottom).

BIGGAN category	YOUTUBE key
indigo bird (014)	DZOiCmxSU2k
green mamba (064)	hG4WvpOU18A
bison (347)	L4eOhuLDfeU
gazelle (353)	jMIiB9DnRXg

Table 4.5: The sequences embedded into the BIGGAN latent space were excerpts of YOUTUBE videos.

verse videos, it has several limitations: One is that the quality of generated videos strictly depends on the quality of the underlying STYLEGAN model and its corresponding PSP inverter. For example, in the case of faces, non-trivial video backgrounds tend to not be represented in a temporally stable way. The importance of temporally stable embedding is also underlined by an experiment in which STYLEGAN was replaced by a BIGGAN model (Brock et al., 2019): Several short videos (see Table 4.5) were embedded using a state-of-the-art optimization-based method (Huh et al., 2020), as encoder-based BIGGAN inversion methods did not seem to exist Xia et al., 2021. The embeddings contain strong temporal noise, making training STYLEVIDEOGAN pointless. Visual results of this experiment are available as part of the supplemental material on the project page (see Section 1.5).

Another limitation is the fact that while STYLEVIDEOGAN does not contain any inherently face-specific components (see proof-of-concept for animating hands and cars; Sections 4.4.6 and 4.4.7), it is unclear whether all the advantageous properties of STYLEGAN's $\mathcal{W}^+$ space can be made use of in any arbitrary domain, e.g. if the offset trick will work there.

4.6 Conclusions

This chapter has presented STYLEVIDEOGAN, a temporal Wasserstein GAN for the unconditional generation of high resolution videos. STYLEVIDEOGAN shows that it is possible to learn the generation of motion for only a single training subject, and then transfer it to a great multitude of other subjects. Embedding the spatial information (i.e. the individual frames) of the training set into the latent space of a time-agnostic model, and then training on temporal sequences of such embeddings has been shown to make training efficient enough to allow for large spatial and temporal output resolutions. The diversity of the generated motion can be ensured by forcing the model to make its output depend on per-time-step randomness.

Like the previous chapter, this chapter was investigating temporal dependencies "inside" videos. As a consequence, STYLEVIDEOGAN gen-

erates videos with some spatial, but (apart from the training set itself) no temporal conditioning, i.e. the user has little control over the generated content. The next chapter, however, sets out to exploit dependencies between videos and the frame signals they originate from (see Equation 2.4). This means that the output of the method developed in the next chapter is very much conditioned on so-called *event input*, which helps to approximate the original frame signal.

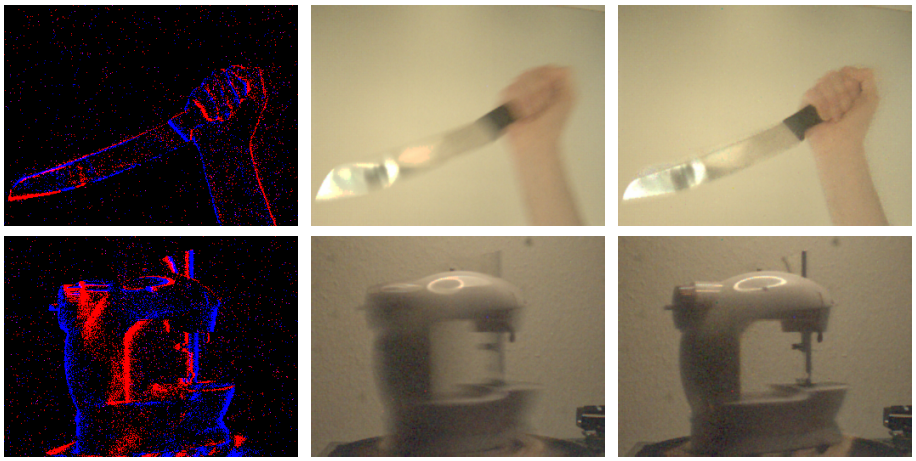
Untrained Event-based Video Reconstruction

5

While Chapters 3 and 4 have considered distributions of videos as ground truth and investigated dependencies *within* one video, this chapter will instead explore dependencies between frame signals (see Section 2.3) and the information recorded about them by a camera. Since a standard camera typically approximates a temporal integration of the frame signal (see Equation 2.4) that cannot easily be inverted, this chapter revolves around a special type of camera, the *event camera* (see Section 2.6).

An **event camera** records the times at which individual pixels change brightness, generating a stream of so-called events. The discrete and asynchronous nature of events (see Section 2.6) makes reconstructing a frame signal from event information a challenging task, even if conventional video frames are recorded along with the events: The recorded information under-constrains the frame signal, so spatiotemporal priors need to be exploited in order to achieve good results. In addition, event data tends to contain noise, i.e. time stamps can be imprecise, events may be omitted and spurious events might be reported.

Previous works have addressed these problems with neural networks, which learn spatiotemporal priors and smooth out noise, but tend to be biased towards their training distributions. This chapter introduces COLIBRI, a new approach to event-based reconstruction (published as Fox et al., 2024). Instead of relying on *learning* spatiotemporal dependencies, it models especially temporal dependencies explicitly, based on the event semantics. To deal with noise, each event is assigned an explicit confidence weight, accounting for the uncertainty arising from noise. A novel loss term balances these confidences against each other. The chapter also shows that brightness interpolation between events can benefit from the use of Bézier curves and that allowing brightness changes in exposure gaps can improve reconstruction quality. These ideas are shown to improve the state of the art in the tasks of event-based video deblurring and event-based frame interpolation.



Input events Input RGB Output RGB

Figure 5.1: Given an event stream and a sequence of long-exposure RGB frames, the goal is to model a continuous brightness signal that plausibly explains this input. The model should be queryable at arbitrary exposure times, for example to obtain deblurred output RGB frames. More results in the project video (Section 1.5).

5.1 Introduction

Classic frame-based cameras synchronously expose their pixels to incoming brightness, for a non-zero exposure duration, as modelled in Equation 2.4. The resulting video frames indicate the average brightness (or, more precisely, the average amount of energy) flowing into each camera pixel during exposure. This averaging is problematic for fast motion: Either a high-end camera with very short exposure time and high framerate is used, which requires large amounts of memory for storing frames and consumes considerable amounts of power, or a low-end camera with rather long exposure time and low framerate is used, where the averaging leads to motion blur in the frames.

Event cameras (Leñero-Bardallo et al., 2011; Lichtsteiner et al., 2008; Serrano-Gotarredona et al., 2013) have their pixels *asynchronously* report so-called "events": A pixel emits an event as soon as the brightness it measures deviates from a reference value by a sufficient margin c. The reference value is usually the level of brightness measured at the previous event. For a formal model of events see Section 2.6. Event cameras measure brightness at a far higher rate than conventional cameras, which greatly reduces motion blur and also makes them suitable for low-light conditions. In addition, the fact that pixels produce data only when they measure a *change* in brightness makes event cameras encode sequences in a much more compact format than frame-based cameras.

Events can be considered an encoding of temporal dependencies as defined in Section 2.2: The *presence* of two consecutive events at times t_1, t_2 allows to turn information about pixel brightness at t_1 into information about pixel brightness at t_2 (see Equation 2.19). The *absence* of events between t_1 and t_2 informs about brightness at all times in the interval (Equation 2.20).

This chapter focuses on event cameras that not only record events, but also low-framerate, long exposure video frames through the same pixel matrix, see Figure 5.1. In this context, there are dependencies between the video frames and the frame signal (Equation 2.4), as well as dependencies between the events and the frame signal (Equations 2.19 and 2.20).

Given a recording of such a camera it is desirable to reconstruct a frame signal that could explain both the events and the frames, which will be referred to as **event-based video reconstruction**. In doing so, one can obtain deblurred versions of the recorded frames, or interpolate frames in-between exposures. This allows one to record sequences at lower memory bandwidth and power consumption than with a classic high-framerate camera, while capturing more temporal detail than a low-framerate camera would.

Many approaches to the task of event-based video reconstruction (Z. Jiang et al., 2020; Lin et al., 2020; Paredes-Vallés et al., 2021; Rebecq et al., 2019, 2021; Scheerlinck et al., 2020; Stoffregen et al., 2020; Stepan Tulyakov et al., 2021; B. Wang et al., 2020; Z. W. Wang et al., 2019; X. Zhang et al., 2022) are based on neural networks, which learn priors that enable them to account for noise in the input. However, training these models requires the collection of a sufficiently large and diverse training dataset, which is difficult because event cameras are still relatively exotic and pricey sensors, not at all comparable to the ubiquity of the classic **active pixel sensor**s (**APS**) found in every smartphone. Datasets are thus often limited to a certain domain. *Synthetic* data, i.e. event streams that are derived (for example with ESIM; Rebecq et al., 2018) from high-framerate RGB frames, is easier to obtain, but still expensive in the sense that large numbers of frames need to be stored and used for supervision, with the additional problem that the derived events contain assumptions made by the event simulator.

In contrast, other methods (Pan et al., 2022, 2019; Z. Wang et al., 2021) do not rely on dataset-based learning, but exploit the semantic properties of events in a principled way, for example by solving an optimization problem at test time. The method presented in this chapter, COLIBRI, belongs into this category: Given events and long-exposure frames as input, COLIBRI defines a family of frame signals (see Equation 2.2) that incorporate the events by construction. This is achieved by exploiting the temporal dependencies encoded by the events (see Equations 2.19 and 2.20) in such a way that any frame signal admitted by the model

will adhere to them. An energy term measures the compatibility of the frame signals with the APS frames recorded by the camera. This energy is minimised by gradient descent, in order to find a signal in the family that optimally reproduces the frames. COLIBRI shares a loss term with mEDI (Pan et al., 2022), but the model itself and the optimization strategy are completely different. Beyond that, the model treats the problem of spurious events in a novel way, by assigning each event a "confidence weight" that can modulate its influence. Additional degrees of freedom allow the model to reproduce brightness frames in regions in which brightness changes are too small to trigger events. Lastly, frame signals are constructed as piecewise Bézier functions, instead of piecewise-constant functions, which is shown to improve reconstruction accuracy.

In summary, the contributions of this chapter are:

- A new method to reconstruct continuous frame signals that explain a stream of input events and a sequence of input frames with long exposure time.
- The method does not require any training and hence no training *data* that it would be biased towards.
- Per-event confidence weights, regularised by a novel loss term, are adjusted during optimization.
- Exposure-based control points help produce smooth signals when brightness changes did not trigger events.
- Bézier interpolation in-between events leads to higher reconstruction accuracy.

5.2 Related Work

Because of their advantages over classical frame-based cameras, event cameras have been covered by a great number of works in computer vision. A number of comprehensive surveys (Chakravarthi et al., 2024; Gallego et al., 2022; Zheng et al., 2023) are available. This section focuses only on those related to frame interpolation, deblurring and video reconstruction.

5.2.1 Event-based Frame Interpolation

Given a sequence of short-exposure frames, along with an event stream, **frame interpolation** is the task of computing additional frames that lie temporally in-between the given ones.

Early approaches (Brandli et al., 2014; Scheerlinck et al., 2018) to this task are engineered for online processing and use frames to prevent drift arising from event noise.

More recent approaches (Chen et al., 2023; Gao et al., 2023; Han et al., 2021; He et al., 2022; Kiliç et al., 2023; T. Kim et al., 2023; Paikin et al., 2021; Stepan Tulyakov et al., 2022, 2021; S. Wu et al., 2022; Zhiyang Yu et al., 2021) are **learning-based**. TIME LENS (Stepan Tulyakov et al., 2021) is one of the most prominent examples: Events are aggregated in a voxel grid, that is used for two estimates of the interpolated frame: The "synthesis-based" estimate is obtained by directly combining the grid with the input frames, and the "warping-based" estimate is obtained by deriving an optical flow estimate from the voxel grid and warping the input frames according to this flow. Both interpolation estimates and the flow are combined and refined, in order for an attention mechanism to blend them into one final output frame. This architecture allows TIME LENS to have the warping-based estimate compensate for failures of the synthesis-based one, and vice-versa. Like other methods in this category, TIME LENS assumes the input frames to be the result of very short exposure, containing as little motion blur as possible.

Several of the learning-based methods (Gao et al., 2023; Kiliç et al., 2023; T. Kim et al., 2023; Stepan Tulyakov et al., 2022, 2021) collect training data with custom hardware setups consisting of an event camera and a short-exposure, high-framerate camera, that either have a very small baseline between them, or receive their input through a beam splitter. This makes data acquisition cumbersome and the setups used in real applications are likely to differ. Other methods (Chen et al., 2023; Han et al., 2021; He et al., 2022; S. Wu et al., 2022; Zhiyang Yu et al., 2021) use (predominantly) synthetic events derived from high-framerate footage, which differs even more from concrete use cases. Any model trained on such a dataset will thus exhibit a bias towards not only the type of content present in the dataset, but also the lighting conditions, motion types (slow/fast, linear/non-linear), image resolutions, framerates and even the presence or absence of colour.

In contrast, COLIBRI is not learning-based. While its design decisions nevertheless bias it in some way, its optimization problem is only solved once the input is known, as opposed to neural networks, that optimise their losses *before* the input is known. Furthermore, most methods in this section assume the input frames to be free of motion blur, while COLIBRI is specifically designed for resolving strong motion blur.

5.2.2 Event-based Deblurring

Given a sequence of *long*-exposure frames, along with an event stream, **deblurring** is the task of computing "latent" frames that lie inside the exposure periods of the input frames. The input frames are expected to contain strong motion blur. The output frames are supposed to be "sharp" frames with an infinitesimally short exposure.

A work in this category that many later works refer to is the *(Multiple) Event-based double integral model* (EDI/mEDI) by Pan et al., 2022, 2019: It is not learning-based, but describes the relationship between blurry frames and event streams that originate from the same frame signal: Integrating (i.e. counting) events gives an estimate of the signal, and integrating this estimated signal over the exposure period gives the blurry frame. The model assumes the frame signal to be piece-wise constant between events. Since solving the equations of the EDI model frame by frame tends to produce temporal flickering (due to noise in the event stream), the mEDI equations are solved for all frames at once. COLIBRI is also based on a global optimization problem, but it admits confidence values for events and allows non-linear signal pieces in-between control points. Furthermore, it treats the non-negligible "exposure gaps" in-between frames. The advantages of these improvements are evaluated in Section 5.5.

The great majority of later methods (H. Cho et al., 2023; Z. Jiang et al., 2020; T. Kim et al., 2022; Lin et al., 2020; Shang et al., 2021; L. Sun et al., 2022; F. Xu et al., 2021; H. Zhang et al., 2023) are **learning-based**. Lin et al., 2020, treat the spatiotemporally varying event thresholds explicitly, by using dynamic filter layers (Jia et al., 2016) in the CNN architecture of their method LEDVDI. In contrast to most previous methods, the CNN processes more than 2 subsequent blurry frames at once, allowing to account for more far-ranging temporal dependencies. A side effect of this design is that the factor by which temporal resolution is increased in the deblurring process must be fixed at training time. The method uses event binning (see Section 2.6).

As in Section 5.2.1, the learning-based methods in this section need to account for the scarcity of real training data by using (partially) synthetic data (L. Sun et al., 2022; F. Xu et al., 2021), or custom hardware for data collection (H. Cho et al., 2023; L. Sun et al., 2022), giving rise to the aforementioned domain gap issues. In contrast to COLIBRI, many methods in this section (e.g. those by H. Cho et al., 2023; T. Kim et al., 2022; Lin et al., 2020; L. Sun et al., 2022; F. Xu et al., 2021; H. Zhang et al., 2023, judging by architectures and shown results) do not allow the temporal output resolution to be chosen freely after training. Many methods (Z. Jiang et al., 2020; Shang et al., 2021; L. Sun et al., 2022; F. Xu et al., 2021) use binning of events (see Section 2.6), compromising temporal accuracy.

5.2.3 Event-based Video Reconstruction

Video reconstruction, in the context of this chapter, is the task of estimating a frame signal from an event stream (see also Section 2.6). The literature contains methods that use only event data as input, as well as methods that use relatively long-exposure frames along with the events:

5.2.3.1 From Events Only

Early methods formulate the task as a pure optimization problem, with optical flow as an auxiliary output (Bardow et al., 2016), or by estimating intensity directly from the surface of latest event times (Munda et al., 2018; Reinbacher et al., 2016). In contrast to COLIBRI, they are designed for reconstruction in real time.

More recent methods (Barua et al., 2016; Cadena et al., 2021; Scheerlinck et al., 2020; L. Wang et al., 2019; Weng et al., 2021; Z. Zhang et al., 2023; L. Zhu et al., 2022; Zou et al., 2021) are **learning-based**: Rebecq et al., 2019, 2021, presented an RNN-based model that is trained on synthetic data (Rebecq et al., 2018) and processes its input at real-time rates. As in other methods that use only events, a still background cannot be reconstructed: The camera must be moved slightly in the beginning of the sequence in order to have background information in the event stream. Although maintaining this background afterwards would be simple (by copying pixel values), the network did not learn to do so reliably from the synthetic training data. Stoffregen et al., 2020, introduce the *High Quality Frames* (HQF) dataset, supposed to help bridge the domain gap between real and synthetic event data. The authors identify the logarithmic brightness threshold of the event camera (see Section 2.6) to be an important parameter contributing to the size of the domain gap, and report it to vary significantly even within single datasets, presenting another opportunity for training data bias. Zou et al., 2021, train a convolutional recurrent network on paired data. They collect real data through a custom beam splitter setup at 2000Hz. The method uses event binning. In contrast to other methods, that derive frames from events, this one derives events from the estimated frames and penalises deviations from the input events. Z. Zhang et al., 2023, cast the reconstruction problem as a "linear inverse" one, that, given an optical flow estimate, can be solved using learning-based regularisers. Since the authors do not explicitly model temporal consistency, "two consecutively reconstructed images may change appearance considerably".

Many of the methods in this category (e.g. Paredes-Vallés et al., 2021; Rebecq et al., 2019, 2021; Scheerlinck et al., 2020; L. Wang et al., 2019; Weng et al., 2021; Zou et al., 2021) use event binning (see Section 2.6). Several (L. Wang et al., 2019; Z. Zhang et al., 2023) do not explicitly model temporal dependencies, which can lead to output discontinuities. By definition none of the methods in this category use frames recorded alongside the events.

5.2.3.2 From Events and Frames

If the event camera, such as the DAVIS 346C, records frames through the same pixel matrix as the events, these can help constrain the reconstruction

problem.

The *Asynchronous Kalman Filter* (AKF, Z. Wang et al., 2021) is a rare example of a method not based on learning: It reconstructs an HDR video from blurry LDR input frames plus events. The method operates online, i.e. it must produce outputs before knowing inputs that lie in the future. The authors specifically model the noise in the event data as a sum of three Gaussian processes. In contrast to many other methods the authors model the refractory period of the event pixels (see Section 2.6). The performance of AKF is evaluated in Section 5.5.

Most other methods (Song et al., 2022; L. Sun et al., 2023; B. Wang et al., 2020; Z. W. Wang et al., 2019; Weng et al., 2023; X. Zhang et al., 2022) in this category are **learning-based**: X. Zhang et al., 2022, presented *Event-based Video Deblurring and Interpolation* (EVDI). In this model, the double integral from Pan et al., 2022, 2019, is not computed precisely, but predicted from the event stream by a neural network, that thus has the opportunity to compensate for noise in the events. Event data is partitioned and "reversed" in a way similar to that by B. Wang et al., 2020. It also undergoes binning. This way, for each target time the model can make a forward prediction derived from the previous brightness frame and a backward prediction derived from the following brightness frame. Self-supervision encourages these predictions to be as similar as possible, and a fusion module learns to combine them. Like the method presented in this thesis, the EVDI supervision also computes the deviation of the integral under the estimated brightness signal from the blurry input frames. However, this deviation is approximated numerically, and thus expensive and/or imprecise, whereas the method presented here computes the integral analytically. The compatibility between event stream and estimated brightness signal is fulfilled by construction in COLIBRI, but X. Zhang et al., 2022, need to supervise it explicitly. Section 5.5 evaluates EVDI. Song et al., 2022, model the derivative of the brightness signal of each pixel as a temporal interpolation between polynomial functions whose parameters are regressed from events by a neural network (which requires event binning). This is comparable to COLIBRI, but that method represents the *integral* under the brightness signal and obtains the signal itself by analytic derivation, while Song et al., 2022, obtain the signal from its derivative via integration. A recurrent refinement module is trained to reduce artefacts in the output frames. Both the U-NET predicting polynomial coefficients and the refinement module are trained on synthetic data (Rebecq et al., 2018). Weng et al., 2023, assume that exposure durations of the input frames are unknown and that the shutter remains closed for non-negligible amounts of time between exposures. The method is related to EVDI, but generalises it both by the exposure estimation and by abstracting several equations into neural networks. Supervision requires sharp frames and blurry inputs of varying exposure durations, so training

relies on synthetic data. The event data is binned in order to feed it into the neural networks. COLIBRI also handles shutter closures of non-negligible duration, but uses the known exposure time stamps as reported by the DAVIS 346C.

Almost all methods in this category need training data. *Real* data is difficult to acquire and thus limited in size, leading to biases towards the training distribution. *Synthetic* data is easier to acquire and the datasets can therefore be larger and content-wise more diverse, but incorporate assumptions made by event simulators like ESIM (Rebecq et al., 2018).

In contrast, COLIBRI solves an optimization problem at test time, without any pre-training. This completely eliminates the need for data collection and training. This advantage comes at the cost of *hand-made* biases that result from the design decisions of the method. However, the fact that energy is minimised only once input data is known can still be an advantage over the pretrained methods, which apply their priors to the input without any corrective. As Section 5.5 shows, this advantage can be big enough to compensate for the lack of priors learned from large data distributions. Another advantage of COLIBRI is that it does not require event binning (like Song et al., 2022; L. Sun et al., 2023; B. Wang et al., 2020; Z. W. Wang et al., 2019; Weng et al., 2023; X. Zhang et al., 2022) but incorporates each and every event at its precise time stamp. This makes pointer arithmetic impossible (e.g. because different pixels contain different numbers of events in the same temporal interval) but a combination of sorting, index tensors and binary search still allows for a reasonably fast implementation.

5.3 The DAVIS 346C Event Camera

The DAVIS 346C is a "hybrid" event camera, i.e. it records both a frame signal and an event stream through the same pixel matrix, making complex hardware setups involving a beam splitter unnecessary. The camera uses a Bayer filter, which especially for convolution-based methods makes it advisable to rearrange the brightness values as explained in Section 5.3.1. Sequences are recorded in global shutter mode. Since the gaps between exposures (see Section 2.3) cannot be reduced arbitrarily as exposure frequency increases, higher framerates can result in *less* information about the scene (see Section 5.3.2). COLIBRI is therefore designed for rather long exposures (such as 0.2s).

For the experiments in Section 5.5, the DAVIS 346C was used with the default parameters. FPGA filtering was enabled. Some sequences were recorded with the background activity filter. While the rates of events differ noticeably depending on the usage of the background activity filter, no difference in reconstruction results was noticeable.

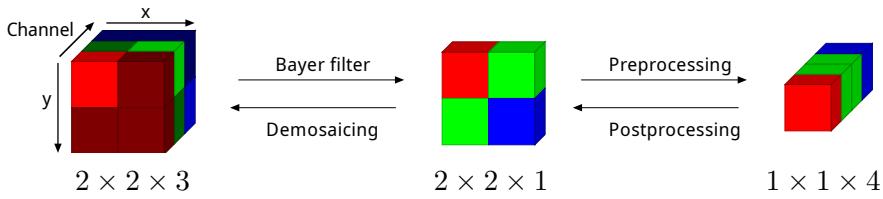


Figure 5.2: The DAVIS 346C partitions its pixel matrix into 2×2 tiles. **Left**: Each of the 4 pixels in a tile is sensitive (brighter blocks) to only one of three wavelet ranges, with with 25% of the pixels seeing red, 25% of the pixels seeing blue and 50% of pixels seeing green. This principle was used for synthetic data as well. **Centre**: The Bayer filter thus gives only one colour value per pixel, not three. **Demosaicing** can interpolate "missing" colour values (darker blocks on the left), which was done for all visual results in this chapter. **Right**: Preprocessing rearranges each $2 \times 2 \times 1$ tile to a $1 \times 1 \times 4$ pixel, to obtain contiguous colour planes. This is the representation that all methods in Section 5.5 operate on.

5.3.1 Frame Dimensions and Colour

The pixel matrix of the DAVIS 346C, of resolution 346×260, is equipped with a Bayer filter, which means that each pixel records brightness in only one of three wavelength ranges (red, green and blue). The pixel matrix is thus partitioned into 2×2 squares, that are treated as 1 pixel of depth 4 (red, green, green, blue), see Figure 5.2. The brightness frames therefore have the dimensions $W \times H \times C = 173 \times 130 \times 4$. In post-processing this transformation was undone followed by Bayer demosaicing and sRGB gamma correction to obtain the results shown in Section 5.5.

5.3.2 Exposure Gaps

Long exposure times are more desirable than high framerates for the DAVIS 346C. This is because as one shortens exposure times to obtain higher framerates, the average "gap time" between exposures does not fall below a certain minimum, as can be observed in Figure 5.3. The mapping from exposure times to framerates is thus not at all linear and the **exposure coverage**, i.e. the percentage of the sequence time during which the shutter is actually open decreases as exposure time is reduced (see Figure 5.4). This means that the shorter one makes exposures (e.g. to reduce motion blur), the less information about the course of the sequence can actually be represented by the frames. As a simple example, consider a laser pointer moving non-linearly on a wall: A long exposure frame representing this motion will be blurry, but it will inform the viewer about the exact trace of the laser point. Two short exposure frames with a considerable exposure gap between them will only give 2 positions

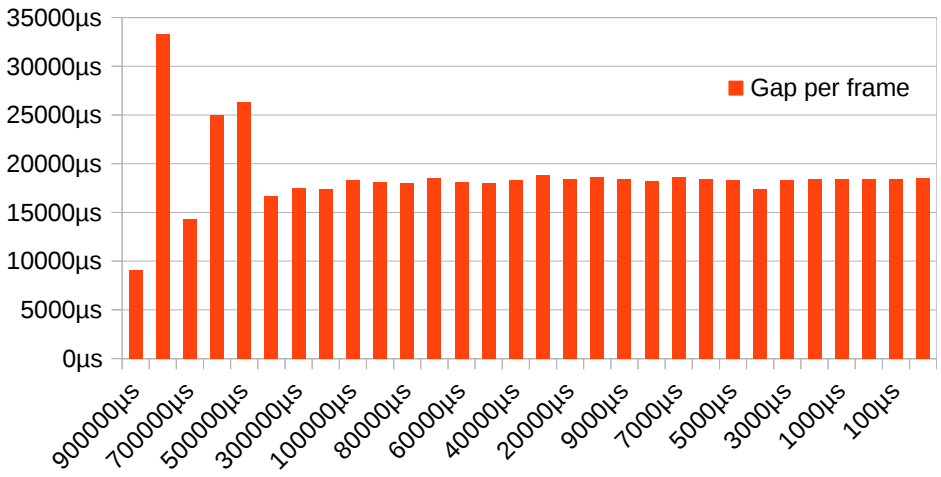


Figure 5.3: A visualization of the "Gap per frame" column of Table B.1, mapping exposure times to exposure "gaps". As exposures become shorter, gap times stay at around 18ms.

of the points and not inform the viewer about the path the point was following between those positions. It is for this reason that one should configure the camera with long exposures. For detailed measurements of the relationship between exposure time and framerate see Appendix B.

5.4 Method

As input, an event stream E and a frame sequence F are given, both interpreted to have been derived from the same frame signal B (see Sections 2.3 and 2.6). As explained in Section 5.3.2, F was captured with relatively long exposure durations. The goal is to find a frame signal B^* that is as compatible as possible with the input. B^* is modelled by directly incorporating the events into the construction of the signal, whereas the frames are used to formulate a loss term. This loss is minimised by gradient descent, to find values for the free parameters of the model. Once B^* has been found, one can compute integrals over much shorter exposures than were used for F, to re-render the sequence at arbitrary temporal resolution.

The model treats each pixel (x, y, z) in isolation. To simplify notation, much of the remainder of this section will assume one arbitrary but fixed pixel identity (x, y, z) and omit arguments and tuple components x, y, z.

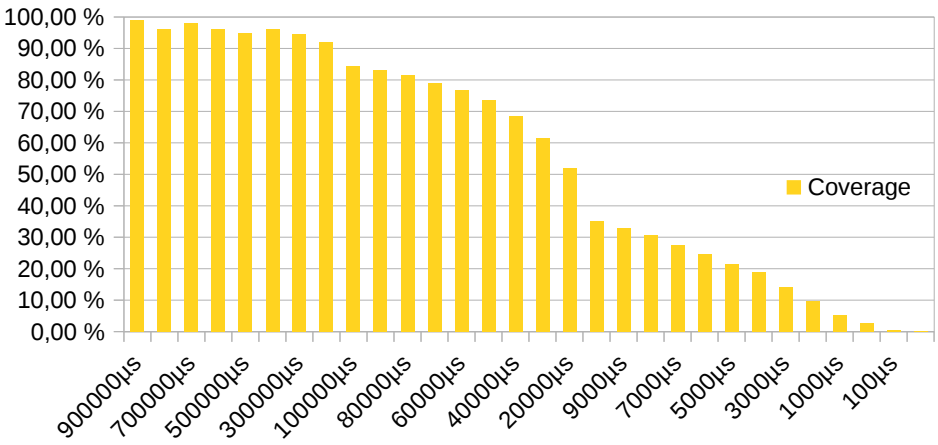


Figure 5.4: A visualization of the "Coverage" column of Table B.1, mapping exposure times to the percentage of sequence time during which the shutter is open. Shorter exposure times lead to less coverage and therefore less of the sequence time being observed in the frame signal.

5.4.1 Model

The model $M(t)$ designed here does not represent $B^*(t)$, but its integral:

$$B^*(t) := M'(t) \tag{5.1}$$

This formulation is convenient because the main loss term (Equation 5.8) is expressed in terms of integrals under B^*: Equation 5.1 allows to compute integrals by evaluating M (see Equation 5.10), while B^* can be computed accurately and efficiently by automatic differentiation.

Equations 2.19 and 2.20 constitute a strong prior on the set of admissible functions B^*, that can be exploited by representing each pixel signal of $M(t)$ as an interpolation between carefully defined **control point**s (**CP**, see Figure 5.5) $P_k = (t_k, y_k, g_k, w_k^{\text{left}}, w_k^{\text{right}}) \in \mathbb{R}^5$, that enforce, for all k:

$$M(t_k) = y_k \qquad M'(t_k) = g_k \tag{5.2}$$

while $w_k^{\text{left}}, w_k^{\text{right}}$ govern the interpolation between P_k and its neighbours. The P_k are subject to the following rules:

- Each P_k belongs to one of two types:

  An **event-based CP** represents an event and its t_k is fixed to the time of the event. An **exposure-based CP** represents the transition from one brightness frame $F(i)$ to its successor with $t_k := \frac{1}{2}\left(t_i^{\text{close}} + t_{i+1}^{\text{open}}\right)$

- $y_k \geq 0$ is a free parameter of the model.

- g_k is defined by the event semantics (see below).
- The **gradient weights** $w_k^{\text{left}}, w_k^{\text{right}}$ are free parameters that determine how quickly $M'(t_k)$ turns into $M'(t_{k+1})$.

To make the model compatible with the events by construction, the gradient parameters g_k need to be defined in accordance with Equations 2.19 and 2.20. The pixel signal is thus equipped with one **confidence weight** γ_j per event and one overall parameter $\bar{B}$. γ_j represents the confidence the model has in the validity of event j. Modelling such confidence is necessary because the assumption of only two threshold values c_{+1}, c_{-1} is a strong simplification: The physical properties of the camera circuit make the thresholds rather fuzzy, leading to an entire distribution of thresholds that could have caused a particular event. The confidence weights account for this uncertainty. The parameter $\bar{B}$ is left free and represents the average brightness assigned to the pixel over the entire sequence duration (see Figure 5.5). The confidence weights are transformed and multiplied with the thresholds, to obtain the **effective (logarithmic) thresholds** c_j for all events:

$$c_j := c_{p_j} \cdot \text{sigmoid}(\gamma_j \cdot \omega_{p_j} + \beta_{p_j}) \tag{5.3}$$

where the scales $\omega_p \in \{\omega_{+1}, \omega_{-1}\}$ and biases $\beta_p \in \{\beta_{+1}, \beta_{-1}\}$ are shared by all pixels. While the raw confidence weights are unbounded and will be scaled and shifted according to $\omega_{p_j}, \beta_{p_j}$, the sigmoid function makes sure that c_j is in $[0; c_{p_j}]$. This allows events to be "weakened", to compensate for spurious events, or strengthened, to compensate for missed events. As one configures the camera to use smaller logarithmic thresholds, to increase precision, the frequency of spurious events increases, due to imperfections of the circuitry.

Given $\bar{B}$, chaining Equation 2.19 in the form $p_{j+1} \cdot (\tilde{B}^*(t_{j+1}) - \tilde{B}^*(t_j)) = c_j$ admits only one possible valuation for those g_k that are *event*-based, if one assumes that brightness is constant between events (see Figure 5.5). The computation of the *event*-based control points is described in Section 5.4.2. For the remaining, *exposure*-based control points P_k, one considers the latest event j that occurs before t_k: Since there exists an **event-based control point** $P_{k'}$ with gradient $g_{k'}$ for this event, one can set

$$g_k := \exp\left(\delta_k \cdot c_{\text{sign}(\delta_k)} \cdot \frac{c_j}{c_{p_j}}\right) \cdot g_{k'} \tag{5.4}$$

where $\delta_k \in [-1; 1]$ is a free model parameter, allowing log-brightness values at exposure-based control points to deviate from the value at the beginning of the event interval they lie in by at most c_{+1} or c_{-1}, satisfying Equation 2.20.

One has now determined a set of control points P_k that make M consistent with Equations 2.19 and 2.20, on the basis of the parameter $\bar{B}$, the

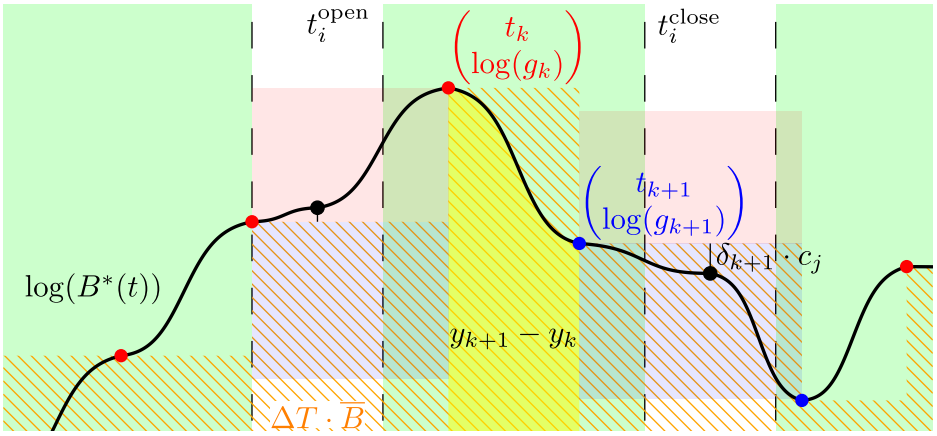


Figure 5.5: B^* visualised in the $\log$ domain: Green rectangles represent input exposures, with significant gaps in-between. The orange-hatched area ($\Delta T := t_{\text{end}} - t_{\text{start}}$) is defined by $\bar{B}$ and the event times and polarities, determining the brightness levels at each event (red for positive polarity, blue for negative). Black points are exposure-based and must lie within either the reddish or blueish rectangle that they are depicted in here (Equation 5.4). Based on the g_k, y_k and other control point parameters, M is constructed as a piece-wise Bézier curve (*not* depicted here!) and thus $M' = B^*$, depicted as the black curve.

parameters γ_j for all events and the parameters δ_k for all exposure-based control points k. Each pixel has its own set of these parameters. The only parameters shared between pixels are $\omega_{+1}, \omega_{-1}, \beta_{+1}, \beta_{-1}$, and c_{+1}, c_{-1}.

To define M *between* the control points P_k and P_{k+1}, one could use straight lines (which by Equation 5.1 would translate into piecewise-constant brightness signals) or parabolas (leading to piecewise-linear signals), but these methods would impose further constraints on the modelled signal, because they cannot comply with arbitrary combinations of control point parameters. Not even cubic interpolation along the temporal axis is sufficient, because it takes as parameters only start and end points of a signal, plus the slopes in those points, amounting to 6 degrees of freedom. However, since the gradient weights $w_k^{\text{right}}, w_{k+1}^{\text{left}}$ are supposed to control how quickly the slope of B^* transitions from g_k to g_{k+1}, a total of 8 degrees of freedom is required, necessitating the use of Bézier curves. As Section 5.5.4 shows, Bézier curves lead to higher accuracy than straight lines or parabolas.

5.4.2 Computation of Control Point Gradients

To compute the gradients of **event-based control points**, i.e. to determine the exact shape of the orange-hatched area in Figure 5.5, the events in a

given pixel are sorted by time and numbered as (t_j, p_j), with $j \in \{1, ..., n-1\}$. The **artificial event** $(t_0, p_0) := (t_{\text{start}}, \Omega)$ is added to simplify notation, where Ω denotes some fixed, but undefined polarity that will never be used. There are now n events in total.

By chained application of Equation 2.19 to the effective thresholds c_j, for all events from the first one (artificial, at time t_{start}) up to event j, one can compute the factor ${f_j}^+$ with which an initial brightness level must be multiplied in order to obtain the correct brightness value after event j:

$$ {f_j}^+ := \exp\left(\sum_{l=1}^{j} p_l \cdot c_l\right) \tag{5.5} $$

Based on the average brightness parameter $\bar{B}$ one can now define brightness values (i.e. gradients) that M should have between events j and $j+1$:

$$ g_j := \frac{\bar{B} \cdot \Delta T}{\sum_{l=0}^{n-2} {f_l}^+ \cdot \Delta t_l} \cdot {f_j}^+ \tag{5.6} $$

where $\Delta t_j := t_{j+1} - t_j$ and $\Delta T := t_{\text{end}} - t_{\text{start}}$. These are the exact values of B^* at the red and blue points of Figure 5.5, determining the shape of the orange-hatched region and making its size exactly $\Delta T \cdot \bar{B}$.

(The publication Fox et al., 2024, defined an additional ${f_j}^-$ and argued that it was needed for gradient backpropagation. That assertion has been found to be an error in the write-up. All results in the publication and this chapter were computed with a computation that does not use ${f_j}^-$).

For each control point P_k that is **event-based** and thus represents exactly one event j (where $j = 0$ represents the first, artificial event at t_{start}), one can now set $g_k := g_j$. The remaining, exposure-based g_k are defined by interpolation between the **event-based** ones, described in the next section.

5.4.3 Bézier Construction

Equation 5.2 requires a curve between two given points, (t_k, y_k) and (t_{k+1}, y_{k+1}), the derivative of which takes specific values at times t_k, t_{k+1}. A Bézier curve between the points must therefore be at least a cubic one, which means that two additional helper points are needed. With $d_{\text{A}} := w_k^{\text{right}} \cdot (t_{k+1} - t_k)$ and $d_{\text{B}} := w_{k+1}^{\text{left}} \cdot (t_{k+1} - t_k)$ the helper points are chosen as

$$ \begin{aligned} P_{k,k+1}^{\text{A}} &:= P_k + \begin{pmatrix} d_{\text{A}} \\ g_k d_{\text{A}} \end{pmatrix} \\ P_{k,k+1}^{\text{B}} &:= P_{k+1} - \begin{pmatrix} d_{\text{B}} \\ g_{k+1} d_{\text{B}} \end{pmatrix} \end{aligned} \tag{5.7} $$

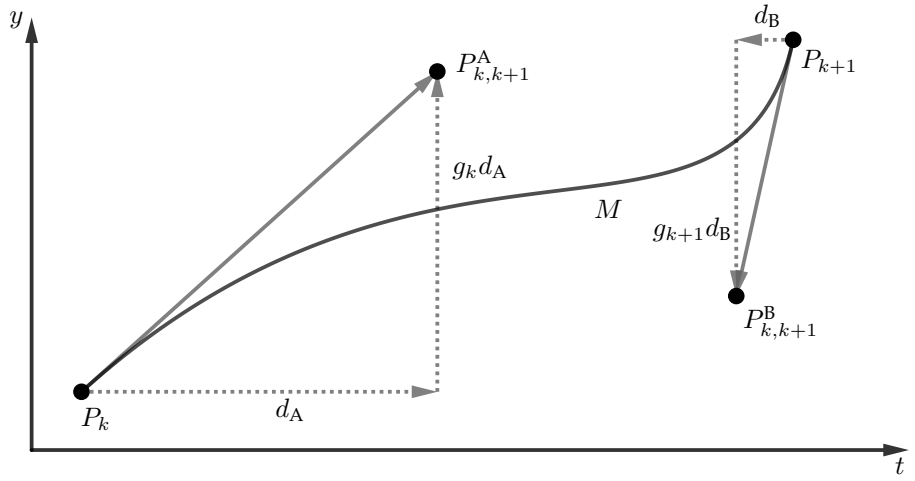


Figure 5.6: Between control points P_k and P_{k+1} the model M is defined as a cubic Bézier curve, which requires helper points $P^A_{k,k+1}$ and $P^B_{k,k+1}$ (see Section 5.4.3). Note that the curve does not represent brightness, but the integral under the brightness signal (see Equation 5.1).

Using the gradient weights w_k^{right} and w_{k+1}^{right} in the definition of the helper points makes them determine the curvature of the Bézier curve, as illustrated in Figure 5.6. The curve completely determines M between P_k and P_{k+1}.

5.4.4 Optimization

M is differentiable with respect to its parameters, so the following losses can be minimised via gradient descent:

The **exposure loss** forces the model to reproduce the input frames:

$$\mathcal{L}_{\text{exposure}} := \sum_{\forall i} \frac{\text{err}_i \left(\int_{t_i^{\text{open}}}^{t_i^{\text{close}}} B^*(t)\, dt \right)^2}{t_i^{\text{close}} - t_i^{\text{open}}} \tag{5.8}$$

where the error for frame i in saturated pixels is zero if B^* saturates the pixel as well:

$$\text{err}_i(G) := \begin{cases} F(i) - G : F(i) < 1 \\ \max(0, 1 - G) : F(i) = 1 \end{cases} \tag{5.9}$$

According to Equation 5.1 the integral can be computed as

$$\int_{t_i^{\text{open}}}^{t_i^{\text{close}}} B^*(t)\, dt = M(t_i^{\text{close}}) - M(t_i^{\text{open}}) \tag{5.10}$$

$\mathcal{L}_{\text{exposure}}$ is strictly necessary because it is the only loss that informs the model about absolute levels of brightness recorded by the camera:

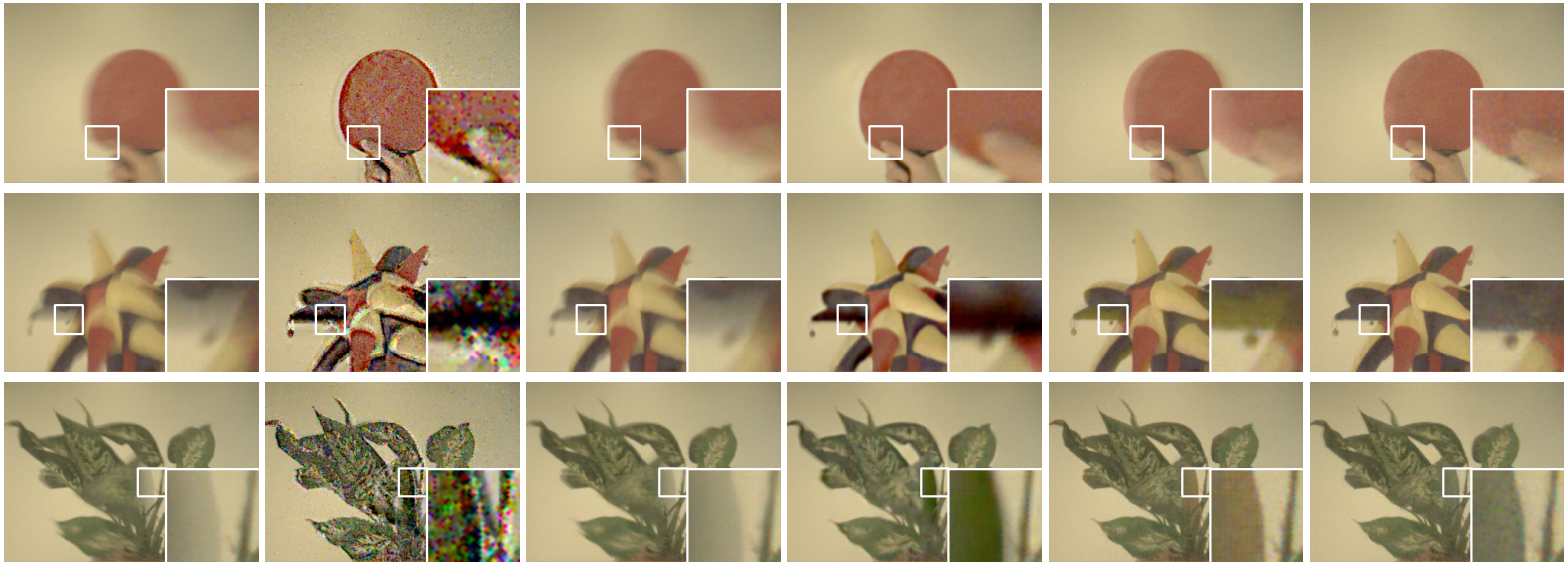


Figure 5.7: Comparison to AKF (Z. Wang et al., 2021), EVDI (X. Zhang et al., 2022), LEDVDI (Lin et al., 2020) and mEDI (Pan et al., 2022). Event input is not shown. Input exposure time was approximately 0.2s, but output exposure time for all methods was 0.002s. COLIBRI reduces motion blur most effectively.

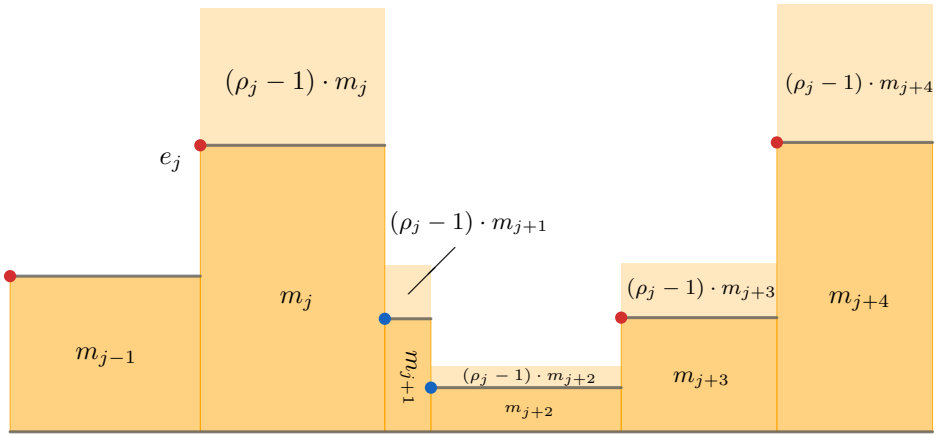


Figure 5.8: $\mathcal{L}_{\text{confidence}}$ (Equation 5.11) penalises integral mass that is missing because a confidence value is less than 1: The x axis is time, the y axis is brightness. Red control points represent positive events, blue control points represent negative events. Dark orange rectangles show the integral mass under the idealised brightness curve (see orange-hatched area in Figure 5.5). Especially e_j has $c_j \ll c_{+1}$. Increasing its confidence s.t. $c_j = c_{+1}$ would add *all* of the light orange areas to the integral under the idealised brightness curve, amounting to ${m_j}^+$.

Without it the model (in particular the parameters $\bar{B}$) could converge to arbitrary multiples of the brightness values recorded in the frames and each pixel could do so independently from the others. In addition, this term helps suppress noise that may be present in the event data.

The **confidence loss** uses the helper variables ${m_j}^+$ and ${m_j}^-$ to drive all confidence weights γ_j up, such that the $\mathrm{sigmoid}$ term in Equation 5.3 approaches 1:

$$\mathcal{L}_{\text{confidence}} := \sum_{e_j \in E} \frac{\left(\frac{1}{2}{m_j}^+ + \frac{1}{2}{m_j}^-\right)^2}{\Delta T} \tag{5.11}$$

where ${m_j}^+$ and ${m_j}^-$ are penalties on the the amount of integral mass under the brightness signal (see Figure 5.8) that would be gained/lost by making the $\mathrm{sigmoid}$ term in Equation 5.3 equal to 1:

Let $e_j \in E$ with time stamp t_j and polarity p_j. The "stop-gradient" function SG denotes the identity function, but silences gradients in back-propagation, to make $\mathcal{L}_{\text{confidence}}$ affect no parameters other than the confidence weights. For the l-th event in the pixel, there exists the **event-based control point** P_k. The **idealised integral mass** m_l under the brightness signal between the events l and $l+1$ is defined as

$$m_l := \mathrm{SG}(g_k) \cdot \Delta t_l \tag{5.12}$$

The number ρ_j is the factor by which the brightness level between events j and $j+1$ would change if the confidence for event j could make the sigmoid term in Equation 5.3 equal to 1:

$$\rho_j := \exp\left(\mathrm{SG}(c_{p_j}) \cdot \left(1 - \mathrm{sigmoid}(\gamma_j \cdot \omega_{p_j} + \beta_{p_j})\right)\right) \tag{5.13}$$

ρ_j is also the factor by which all later event intervals would increase their m_l, which is why ${m_j}^+$ equals the absolute amount of integral mass that these intervals would gain:

$${m_j}^+ := (\rho_j - 1) \cdot \sum_{l=j}^{n-1} m_l \tag{5.14}$$

Penalizing only *later* intervals would make it very cheap for the last intervals to have low confidence. This is why ${m_j}^-$ penalises *earlier* intervals:

$${m_j}^- := \left(1 - \frac{1}{\rho_j}\right) \cdot \sum_{l=0}^{j-1} m_l \tag{5.15}$$

$\mathcal{L}_{\text{confidence}}$ encourages all events to be taken as "serious" as possible. Without it some events may be needlessly assigned a low confidence, leading to high-frequency information being ignored, and thus more motion blur.

The **linearity regulariser** encourages Bézier curves to have linear derivatives and thus B^* to be piecewise linear in areas where other losses do not determine a specific shape. This is achieved between P_k and P_{k+1} by penalizing the surface area A_k of the triangle between the points (t_k, g_k), (t_{k+1}, g_{k+1}) and

$$\begin{pmatrix} t_{k,k+1} \\ g_{k,k+1} \end{pmatrix} := \begin{pmatrix} \frac{1}{2}(t_k + t_{k+1}) \\ B^*(\frac{1}{2}(t_k + t_{k+1})) \end{pmatrix}$$

resulting in the loss formulation

$$\mathcal{L}_{\text{linearity}} := \sum_{P_k, P_{k+1} \in \mathrm{CP}} \frac{A_k^2}{t_{k+1} - t_k} \tag{5.16}$$

where CP is the set of all control points for the pixel. Section 5.5.4 shows that this loss gives better results than enforcing linearity by construction.

The overall loss

$$\mathcal{L} := 1 \cdot \mathcal{L}_{\text{exposure}} + 0.2 \cdot \mathcal{L}_{\text{confidence}} + 0.1 \cdot \mathcal{L}_{\text{linearity}}$$

is minimised by gradient descent, updating the model parameters $\bar{B}$ for all pixels, the parameters γ_j for all events, the parameters y_k, w_k^{left}, and w_k^{right} for all control points, the parameters δ_k for all **exposure-based control points**, and the global parameters c_{+1}, c_{-1}, ω_{+1}, ω_{-1}, β_{+1}, β_{-1}.

5.4.5 Implementation Details

Gradient descent The method was implemented using PyTorch and its Adam optimiser (Kingma et al., 2015) for 1000 iterations. The learning rate r_i for iteration i is defined by setting $\alpha_i := (1 - \frac{i-100}{900})^{1.5}$ and specifying:

$$r_i := \begin{cases} 10^{-2} & : i = 0 \\ \frac{i}{100} \cdot r_0 & : 0 < i < 100 \\ (r_0 - r_{999}) \cdot \alpha_i + r_{999} & : 100 \leq i < 999 \\ 10^{-3} & : i = 999 \end{cases} \tag{5.17}$$

The arrays representing model parameters are usually in the range $[-1; 1]$ (even though the range of the mathematical variables they represent may be different!), but some of them, like the confidence weights for example, have a far smaller range, which effectively increases their learning rate.

Initialization and invariants The model parameters are initialised and bounded as follows:

1. Thresholds are always initialised as $c_{+1} = c_{-1} = 0.15$ and kept in the interval $[0.01; 1.8]$.
2. For each pixel, $\bar{B}$ can easily be initialised to be the average brightness level of this pixel over all input frames. It is only bounded to be non-negative.
3. The y_k and δ_k are initialised such that the orange-hatched area in Figure 5.5 satisfies $\mathcal{L}_{\text{exposure}}$ reasonably well. To do so, exposure gaps need to be taken into account. Since the total amount of physical energy received by a pixel up to some time t is monotonically increasing, the function M should be monotonically increasing as well, requiring $\forall k : y_k \leq y_{k+1}$. The δ_k are kept in the range $[-1; 1]$.
4. The confidence weights γ_j are initialised with 1 and not bounded at all, because the sigmoid term in Equation 5.3 properly bounds the effective thresholds based on these weights. For the same reason, ω_{+1}, ω_{-1} (initialised to 1) and β_{+1}, β_{-1} (initialised to 0) are left unbounded.
5. The gradient weights $w_k^{\text{left}}, w_k^{\text{right}}$ are initialised to 0.45 and restricted to the range $[0.05; 0.45]$, to avoid numerical issues that might arise in the computation of Bézier interpolation.

Bézier evaluation The usage of Bézier curves (8 degrees of freedom, see Figure 5.6), as opposed to cubic interpolation (6 degrees of freedom) has one subtle disadvantage: Bézier curves are not parametrised in the

x coordinate of the coordinate system they are defined over, but in some curve parameter t (inconveniently denoted in the literature by the same symbol that this chapter uses for its temporal axis). This means that one cannot straightforwardly map an x coordinate to a point (x, y) on the curve. Instead, one must first map x to the curve parameter t, that in a second step will give a point (x_t, y_t) on the curve. To find t such that $x_t = x$, one must find a root of a polynomial of degree 3. Fortunately, this polynomial is so well-behaved that a small number of Newton steps (fewer than 10) reliably approximates the solution very precisely. This is much easier to implement than the analytical solution (e.g. Cardano's formula).

Performance All input data (frames and events) is loaded into the GPU once before training begins, so there is no overhead for data loading. Every forward and backward pass for optimization is using all the weights of the model and is supervised by the entirety of the input data. Computational performance varies greatly with the number of events and their distribution. Recordings that were processed for this chapter were typically 5 - 15 seconds long. None of them requires more than 48GB of GPU memory, most of them require less than 24GB. Some sequences are processed within tens of minutes, others might take 2 - 4 hours, due to different numbers of events.

5.5 Results

5.5.1 Input Data

Recordings for evaluation were captured with a DAVIS 346C, at exposure 0.2s, resulting in 4.5FPS - 5.0FPS, due to the shutter remaining closed for significant durations between frames. For the DAVIS 346C, the duration of the exposure gaps remains constant as exposure time is decreased, leading to exposures covering less and less sequence time (see Figures 5.3 and 5.4). The long exposure time of 0.2s was chosen to keep this coverage high (92%). For more information see Section 5.3.2. Ground truth signals for quantitative evaluation were obtained in 2 ways: First, dropping every other frame in the recordings allows the evaluation of how well a reconstruction method is able to compute the missing long-exposure frames. Second, similarly to previous work (X. Zhang et al., 2022), events were synthesized from high-framerate RGB sequences: 30 REDS sequences (Nah et al., 2019) were temporally upsampled to 800Hz using FILM (F. Reda et al., 2022; F. A. Reda et al., 2022). Synthetic events were then computed using ESIM (Gehrig et al., 2020; threshold $c = 0.2$). A 10Hz, long-exposure frame sequence was derived from the 800Hz version, without exposure gaps, to serve as input to the evaluated reconstruction methods. The

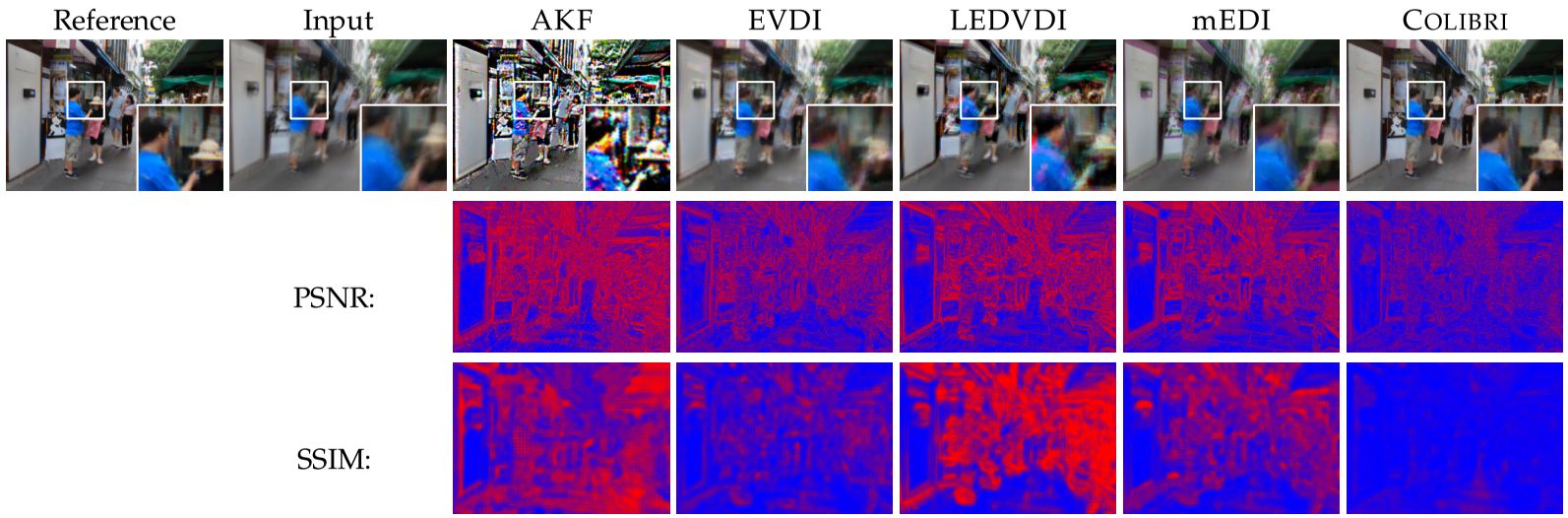


Figure 5.9: On synthetic data, outputs can be compared to a pseudo-ground truth reference. Input exposure time was 0.1s, output exposure time was 0.002s. SSIM in particular often shows the superiority of COLIBRI.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
AKF	14.42dB	0.4861	0.3675
EVDI (pretrained)	23.32dB	0.7356	0.2504
LEDVDI (pretrained)	20.66dB	0.6491	0.2000
mEDI	24.68dB	0.7888	0.1831
COLIBRI	**29.69dB**	**0.9039**	**0.0739**

Table 5.1: All methods were evaluated quantitatively on a synthetic dataset based on REDS (Nah et al., 2019). The scores in this table are averaged over the 30 sequences in the REDS validation set.

sequences of the *Color Event Camera Dataset* (Scheerlinck et al., 2019) could not be used because they lack exposure time information.

5.5.2 Comparison to Previous Work

All previous methods that COLIBRI is compared to take events and long-exposure brightness frames as input: EVDI (X. Zhang et al., 2022) is a learning-based method that is self-supervised, but needs to be pretrained. Likewise, LEDVDI (Lin et al., 2020) is learning-based, but needs ground-truth supervision and thus is trained on synthetic data. It increases temporal frequency by a factor fixed at training time. Since COLIBRI does not require any pre-training at all, the comparison uses the official checkpoints of EVDI ("GoPro" checkpoint) and LEDVDI (frequency factor 6). Since EVDI does not require ground truth for supervision, it is overfit to the input for 10 epochs, helping overcome the domain gap between the input and the training data. Like COLIBRI, AKF (Z. Wang et al., 2021) and mEDI (Pan et al., 2022) do not require pre-training.

TIME LENS (Stepan Tulyakov et al., 2022, 2021) excels at frame interpolation, but requires short-exposure frames that are free of blur. Section 5.5.3 demonstrates that it cannot deal with long-exposure frames. For more information on these previous methods, see Section 5.2.

Each method is made to produce output sequences of latent images (i.e. virtual exposure time 0) at 500FPS, which for most of them requires some linear interpolation of the output frames, because they either produce non-constant framerates or only a fixed multiple of the input framerate (LEDVDI). All previous methods were extended to process coloured data, by applying them to each colour channel individually. This is necessary because official checkpoints have been trained on single channel data only.

Figure 5.7 shows a comparison on multiple recordings: Methods are expected to turn inputs with exposure 0.2s into output with exposure 0.002s, which should reduce motion blur. AKF produces strong spatial noise. Both EVDI and mEDI give results with considerable motion blur. Surpris-

Method	PSNR ↑	SSIM ↑	LPIPS ↓
AKF	22.64dB	0.5033	0.4505
EVDI (pretrained)	33.92dB	**0.9498**	**0.0651**
LEDVDI (pretrained)	30.74dB	0.9120	0.0739
mEDI	34.91dB	0.9114	0.0876
COLIBRI	**37.91dB**	0.9251	0.0882

Table 5.2: Evaluation of the frame drop experiment (see Section 5.5.2). Exposure for both reference and input was 0.2s. Spurious events in the scene background make it hard for COLIBRI to keep background brightness constant, hence its scores do not beat those of EVDI.

Variant	Reference exposure 0.1s			Reference exposure 0.002s		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Linear interpol.	40.72dB	0.9866	0.0112	29.50dB	0.9006	**0.0624**
Parabolic interpol.	39.21dB	0.9795	0.0174	28.95dB	0.8889	0.0793
No confidences	43.96dB	0.9933	0.0047	**30.25dB**	**0.9105**	0.0647
No exp.-based CP	43.69dB	0.9929	0.0052	29.52dB	0.9007	0.0755
Without $\mathcal{L}_{\text{confidence}}$	**45.80dB**	**0.9957**	**0.0032**	27.70dB	0.8677	0.1153
Without $\mathcal{L}_{\text{linearity}}$	42.61dB	0.9915	0.0096	20.92dB	0.6678	0.2613
COLIBRI (full)	45.17dB	0.9949	0.0035	29.69dB	0.9039	0.0739

Table 5.3: Ablation study on synthetic data, comparing outputs to input frames (exposure 0.1s) and to pseudo ground truth (exposure 0.002s). Since this dataset does not contain real-world noise, the event confidences, as well as $\mathcal{L}_{\text{confidence}}$ are not improving performance. However, the full method ranks second best more often than any other method ranks best. Both linear interpolation and parabolic interpolation lead to the input frames being reproduced far less faithfully.

ingly, LEDVDI, which can only produce output at exposure time 0.033s, manages to reduce blur considerably for the racket, but suffers from a "pulsing" artefact (best observed in the project video, see Section 1.5). COLIBRI deblurs best, as can be seen in the third row for example, where EVDI and mEDI struggle with clearly resolving the right edge of the foreground leaf.

Methods were evaluated on synthetic data: Figure 5.9 confirms many observations from the recordings, with the exception of LEDVDI, which now also gives blurry results, possibly due to a bias towards its training distribution. In fact, Table 5.1 lists rather weak scores for LEDVDI and AKF, while COLIBRI consistently outperforms the others.

For quantitative comparison on real inputs, every second frame of the real recordings was dropped, to serve as a blurry long-exposure reference

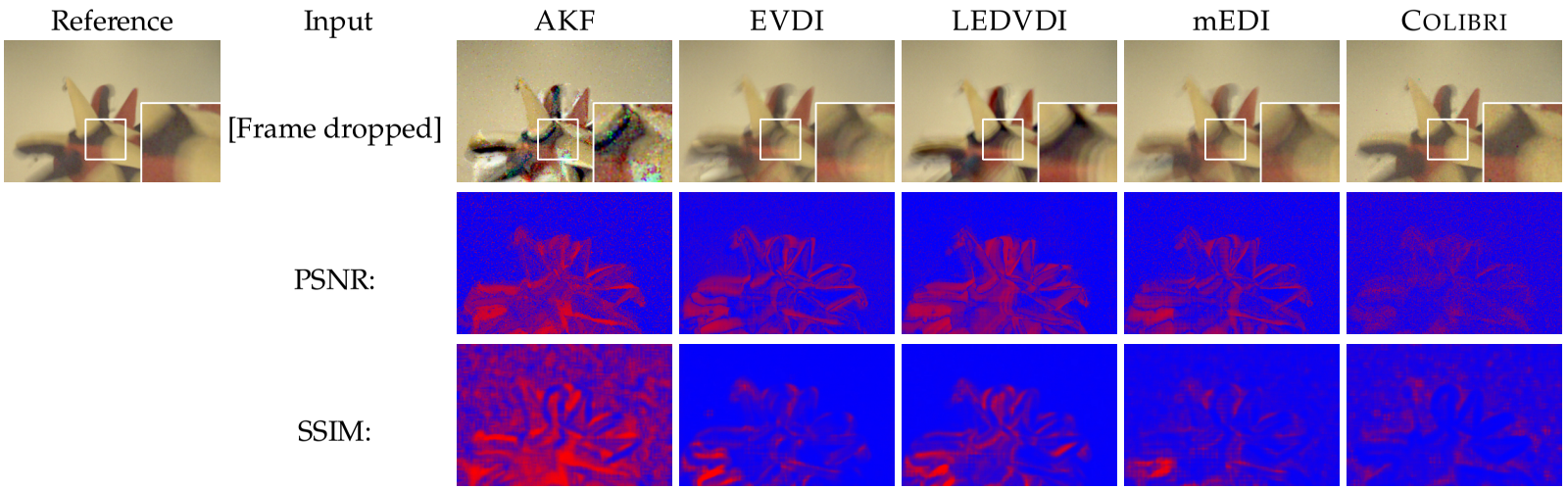


Figure 5.10: After dropping every second recorded frame (Section 5.5), the remaining frames and the events can be used as input, while the original recording can serve as a long-exposure reference. All methods struggle in this setting, but COLIBRI performs best.

(Section 5.5 and Figure 5.10). All methods struggle in this setting, because no frame data is available for more than the typical exposure period. As expected, those frames that were still remaining in the input were reproduced more faithfully than those that were dropped, confirming that event-based reconstruction greatly benefits from the availability of long-exposure brightness frames. Table 5.2 averages scores over a number of recordings: In both SSIM and LPIPS, COLIBRI is outperformed by EVDI. The reason seems to be that the spurious events reported for the scene background (i.e. exactly in those areas where *no* motion is happening) make it hard for COLIBRI to keep the brightness of the background pixels constant. Normally $\mathcal{L}_{\text{exposure}}$ (see Section 5.4.4) would correct that, but without the dropped frames COLIBRI has no way of knowing whether these events are legitimate or not. It is here, where the learned priors of EVDI prove to be an advantage. Error maps and scores are shown in the project video (see Section 1.5).

5.5.3 Time Lens is Out Of Scope

Comparing COLIBRI to TIME LENS (Stepan Tulyakov et al., 2022, 2021) would be unfair to TIME LENS. This is because Section 5.4 investigates the setting of long exposure RGB frames, which contain a lot of motion blur. However, TIME LENS is designed for interpolation between short-exposure frames, that are expected to be free of blur. To demonstrate that TIME LENS is not suitable in the long-exposure setting, it was nevertheless applied to some real recordings and synthetic data: Figure 5.11 shows that TIME LENS is unable to compensate the motion blur present in the long exposure input. Quantitative evaluation on 10Hz synthetic data (see Table 5.1) gave a PSNR score of only 22.51dB, confirming that TIME LENS is the wrong tool for the setting.

5.5.4 Ablation Study

Bézier interpolation Section 5.4 specifies that the interpolation between control points is computed in the form of Bézier splines, which means that the function M, i.e. the integral under the brightness signal (*not* the brightness signal itself!) consists of Bézier curves. Since $\mathcal{L}_{\text{linearity}}$ regularises these curves to approximate parabolas, encouraging B^* to become piecewise linear, it has to be investigated if these choices really give better results than making the interpolation use parabolas already by definition, or even linear functions (in which case the brightness signal would be piecewise constant). However, both these "simplifications" require the addition of further constraints to the model, because piecewise parabolas or lines cannot satisfy Equation 5.2 in all cases, as they admit fewer degrees of freedom than Bézier curves do. Technically this means

Input RGB | TIME LENS output | COLIBRI output

Figure 5.11: A comparison to TIME LENS (Stepan Tulyakov et al., 2022, 2021), at output exposure time 0.002s. The top two rows show results on recordings (see Figure 5.7), while the two lower rows show results on synthetic data. Since TIME LENS was neither designed nor trained for long exposure frames, it fails to resolve the motion blur present in the input.

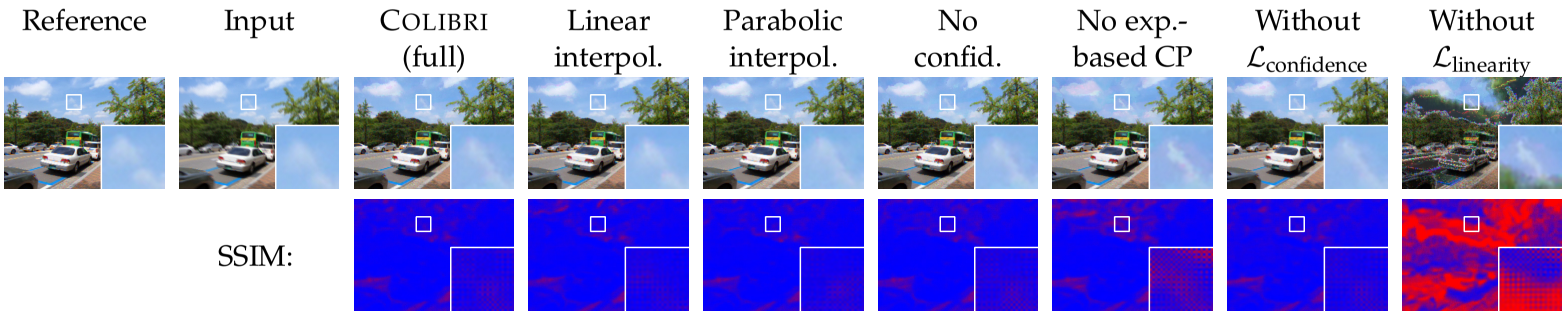


Figure 5.12: In this ablation study on synthetic data, the sky region leads to significant SSIM error if no exposure-based control points are used, because the pixels there are so bright that the *linear* brightness thresholds are relatively large. Enabling exposure-based control points allows $\mathcal{L}_{\text{exposure}}$ to correct the resulting brightness errors. The chequerboard patterns visible in the error maps are due to the Bayer filter of the camera and must appear regardless of the method used.

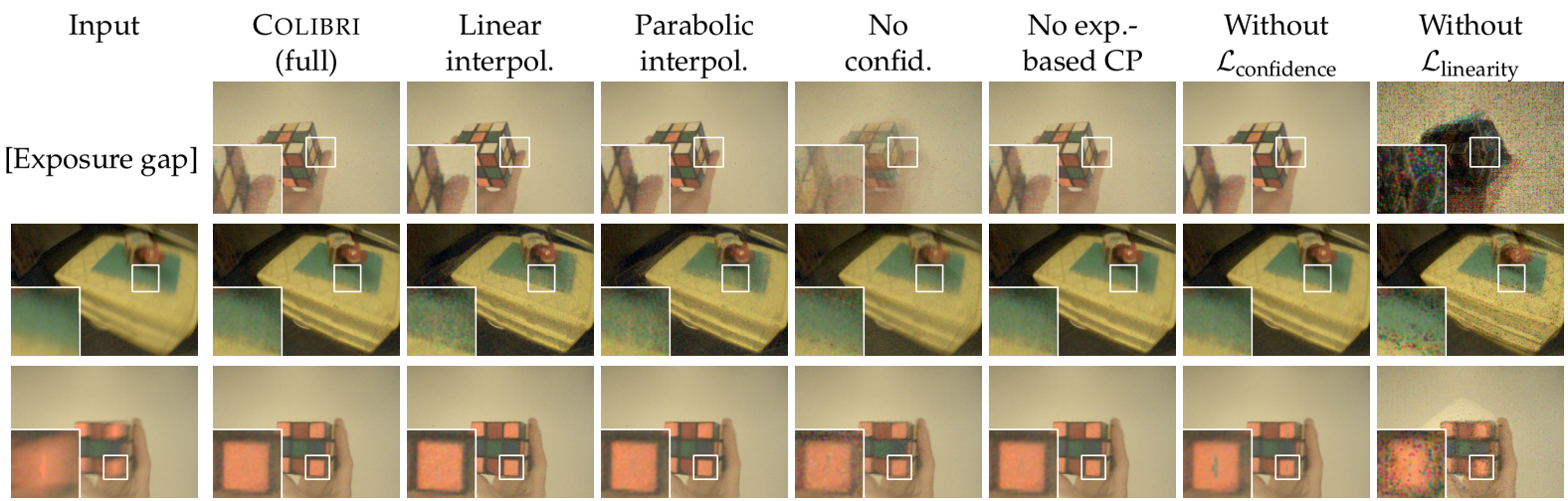


Figure 5.13: Qualitative ablation on real recordings, at output exposure time 0.002s. The first row was captured in-between two input exposure periods, not covered by any input frame. In the second row, both linear and parabolic interpolation lead to increased spatial noise in the green mat on the yellow box. In all three rows, the result without confidences shows significantly more blur and the third one shows strong artefacts in the orange tiles of the cube if $\mathcal{L}_{\text{confidence}}$ is not used. Once more, disabling $\mathcal{L}_{\text{linearity}}$ proves harmful.

that one cannot leave the y coordinates y_k free, and instead has to compute them from other parameters in a way that constrain the model to parabolas or lines. These additional constraints change the dynamics of the signal representation during optimization. Table 5.3 shows that they lead to significant degradations in quality, especially with regard to the ability of COLIBRI to reconstruct the input frames. Furthermore, the second row of Figure 5.13 shows more spatial noise for the simplified interpolation methods.

Exposure-based control points Similarly, Table 5.3 shows that omitting exposure-based control points harms the fidelity with which input frames are reconstructed. Figure 5.12 gives a possible explanation: Pixels in the sky region change their brightness only very slightly as the camera is panning. The logarithmic brightness threshold of 0.2 that was used to synthesize events for the REDS sequences is apparently too large to trigger frequent events for such subtle changes. Therefore, the control points in these pixels are very sparse and cannot satisfy $\mathcal{L}_{\text{exposure}}$, which leads to the sky region having bad SSIM scores. Only exposure-based control points add the necessary degrees of freedom here.

Confidence weights Table 5.3 shows the full method outperform all ablations except those that are missing confidences or $\mathcal{L}_{\text{confidence}}$. The reason why omitting these seems to even slightly improve the metrics in this ablation study is that since no event camera noise was simulated, almost all events in the synthetic dataset are legitimate (except for corner cases due to quantisation), so allowing confidences to deviate from 1 cannot do much good. The PSNR metric (given only for comparison with previous works that use it) is unreliable in the case of Table 5.3: Not only is SSIM a more advanced metric for perceived quality, but also does PSNR diverge towards infinity as images become more similar to the reference, making PSNR differences less and less significant. This suggests that one should focus on SSIM values especially in the first half of Table 5.3, where similarity is very high. On real data however (Figure 5.13), many events are not legitimate at all and especially in-between exposure periods the lack of confidence weights can lead to severe artefacts.

Confidence loss SSIM scores in Table 5.3 show that omission of $\mathcal{L}_{\text{confidence}}$ leads to a significant loss of quality on synthetic data at short exposure times, suggesting that confidences tend to needlessly deviate from 1. The third row of Figure 5.13 confirms this as well, with the omission of $\mathcal{L}_{\text{confidence}}$ leading to strong artefacts in the orange tiles of the cube.

Linearity regulariser Using Bézier interpolation without having $\mathcal{L}_{\text{linearity}}$ regularise those parameters that are not strictly derived from the event data leads to very strong artefacts, as indicated by the poor scores for short exposures in Table 5.3 and the visual results in Figures 5.12 and 5.13.

5.6 Limitations

The design choices of COLIBRI make it avoid many of the limitations of previous methods. For example, since it exploits the event semantics in a principled way, it does not need any pre-training and thus no training data. As a second example, the fact that pixels are treated rather independently from each other makes it easy to handle complex lighting interactions (like transparency in glass and water) non-linear motion and dis-occlusions, which are more challenging for methods that use optical flow for example (see Section 5.2). However, these same design choices of course give rise to limitations as well:

The importance of $\mathcal{L}_{\text{exposure}}$ makes it strictly necessary to have exact time stamps for the brightness frame exposures, i.e. to know the times at which the shutter opens and closes. Some existing datasets, such as the *Color Event Camera Dataset* (Scheerlinck et al., 2019) do not provide this information. In addition, methods like TIME LENS (Stepan Tulyakov et al., 2022, 2021) typically work on input frames that were recorded with minimal exposure time, which, if used as input to COLIBRI, is likely to give noisy results because short exposure times reduce the impact of $\mathcal{L}_{\text{exposure}}$, allowing noise in the event data to become more visible in the output.

Furthermore, COLIBRI uses GPU RAM rather generously: Representing the entire sequence in memory requires significant GPU capacity, especially for large numbers of events. It is possible to apply COLIBRI only to pairs of consecutive brightness frames, to stitch the results together. While this was found to lead to qualitatively comparable results with less memory demand, it does take more computation time, because frame pairs need to overlap (i.e. every frame is treated twice). Valuing computation time higher than memory consumption this chapter reported results from global optimization only.

Although COLIBRI produces estimates for the logarithmic brightness thresholds c_{+1}, c_{-1}, it is not clear how close these estimates are to the actual ground truth values: The event confidences allow not only to "tone down" individual events, but also can reduce the impact of *all* events as a whole (counter-acted by their regulariser). The latter can be compensated by c_{+1}, c_{-1}, which means that the same model signal can be represented by different values of the thresholds (compensating different levels of average event confidence). Not many methods estimate the thresholds accurately (one exception being mEDI; Pan et al., 2022), but knowledge

about the thresholds for a particular sequence is often a valuable piece of information for further processing.

5.7 Conclusions

This chapter has presented COLIBRI, a method for event-based video reconstruction. COLIBRI does not rely on sophisticated priors picked up from a large training corpus. Instead, it exploits the (idealised) temporal dependencies encoded by an event stream, along with classic brightness frames. Nevertheless COLIBRI is able to compete with previous works, even ones based on learning.

This is because COLIBRI solves an optimization problem at *test* time, instead of forcing a prior on input data that may or may not fit a training distribution. Furthermore, instead of using event binning, as most previous methods do (see Section 5.2), the novel per-event confidence weights, together with their regulariser, allow COLIBRI to effectively deal with spurious events without discarding the precise event time stamps. In addition, equipping the brightness signal with new degrees of freedom in-between exposures and even, by the use of Bézier interpolation, in-between single events, helps improve output quality compared to methods that assume brightness to be piecewise constant.

These contributions enable COLIBRI to outperform previous approaches at a temporal resolution 100 times as high as that of the input. Despite remaining limitations, such as high memory consumption and the requirement for the frames to cover as much as possible of the sequence time (i.e. COLIBRI is designed for long exposures), the performance of the presented method suggests that training-time learning should be combined with test-time optimization in order to alleviate domain gap issues.

Remarkably, while COLIBRI takes great care to precisely exploit temporal dependencies encoded by events, it only very weakly connects pixels *spatially*: Only a handful of global parameters are shared across pixels (see end of Section 5.4.4). This means that the model is not actively enforcing (detailed) spatial dependencies. Nevertheless, the supervision by noisy events and long-exposure brightness frames is enough to achieve spatially coherent output frames. If one were instead building a model that actively constructed spatial dependencies, e.g. by spatial (de-)convolutions, but neglected the representation of temporal dependencies (e.g. could not use knowledge about a previous frame when constructing the next one), the viewer would not be equally satisfied, because such models are known to lead to noticeable high frequency flickering (as acknowledged by L. Wang et al., 2019; Z. Zhang et al., 2023). In this sense, temporal dependencies can be considered even more important than spatial ones.

6 Conclusions

This thesis has explored the nature of temporal dependencies in video signals through the lenses of several concrete computational tasks in which they play important roles.

Chapter 3 has constructed a fake detection benchmark dataset of unprecedented visual quality, VIDEOFORENSICSHQ. This dataset helped to show that previous fake detection methods rely to a great extent on manipulation artefacts that a human could spot with the naked eye. The detectors presented in the chapter use not only spatial filters to extract fake detection clues, but also filter along the temporal axis, which is shown to improve robustness against unseen manipulation methods.

The focus of Chapter 4 has been not the analysis, but the *synthesis* of videos. The presented method, STYLEVIDEOGAN, learns the temporal dependencies to preserve from a very small training set of only 10 minutes of footage. STYLEVIDEOGAN's architecture, which separates spatial components from temporal components, reduces computational cost compared to previous approaches. It generates motion trajectories that can be transferred to a great multitude of unseen subjects. Its novel gradient angle penalty helps generate very long videos without looping artefacts.

In contrast to Chapters 3 and 4, which only captured dependencies *within* videos, Chapter 5 has also investigated the relationship between a video and the frame signal that it originates from (see Section 2.3): The chapter presented COLIBRI, a new method to estimate said frame signal from a combination of the video with an event stream. The method does not require any training data, but is able to compete with learning-based methods because it is not biased towards any training distribution. Instead of optimizing weights in a training phase, it optimises parameters only once the input is known. Furthermore it exploits the temporal dependencies encoded by events with greater precision than previous works, by using per-event confidences instead of event binning.

6.1 Insights

Apart from the contributions that this thesis makes to the individual tasks treated in Chapters 3 to 5, it also provides evidence to support a number of overarching insights:

Importance of temporal dependencies It did not take this thesis to understand that temporal dependencies are an important aspect of visual information. However, the thesis gives concrete evidence for their importance in several specific areas:

In the authenticity context, the arms race between attackers and defenders still has significant room to develop in the temporal dimension, because both had previously often neglected it (see Section 5.2). The relatively simple temporal filter in X_{CST} helped detect forged videos from distributions that were not seen in training. This points at temporal dependencies being a valuable dimension to invest in for fake detection, especially since the spatial dimension seems to have been mastered increasingly by the attackers (in the form of StyleGAN and diffusion models; Ho et al., 2020; Rombach et al., 2022; Sohl-Dickstein et al., 2015; L. Zhang et al., 2023).

The comparison of StyleVideoGAN with the work by Tian et al., 2021, (Section 4.2.1) shows that temporal dependencies deserve to be disentangled from spatial ones *explicitly* in the supervision of neural networks: Tian et al., 2021, have their generator traverse a PCA space that has been found *without* temporal data, and they supervise it by spatiotemporal patches. While in theory this might suffice for the generator to learn convincing motion patterns, the motion generated by Tian et al., 2021, is not as characteristic of the training material as that of StyleVideoGAN. StyleVideoGAN derives already its PCA basis from temporal trajectories. It does not even render spatial outputs at training time, such that supervision is more explicitly focused on temporal dependencies. This leads, for example, to the generated faces actually talking, instead of merely being panned around (see project video, Section 1.5).

Finally, Colibri, presented in Chapter 5, treats temporal information more accurately than previous works, by incorporating precise event time stamps in its model, instead of resorting to event binning. Event binning prevents the exploitation of any information about the distribution of events inside a bin and thus limits the possible temporal resolution of the output. This is one of the reasons why Colibri can compete with learning-based approaches. Colibri adds strong *temporal* dependencies to its model, but only rather weak spatial ones, nevertheless producing spatially coherent output pixels. Other works (L. Wang et al., 2019; Z. Zhang et al., 2023), that explicitly model spatial dependencies, but not temporal ones, do show noticeable temporal artefacts. In this sense, temporal

dependencies can be considered even more important than spatial ones.

Management of combinatoric blow-up Taking temporal dependencies into account tends to make every aspect of an algorithm, be it training data acquisition, storage, inference, or supervision, much more expensive. This is because the temporal dimension usually acts as an additional combinatoric factor that inflates resource usage. This indeed posed severe engineering challenges in the development of VIDEOFORENSICSHQ and it is the main reason for the large memory consumption of COLIBRI. But as STYLEVIDEOGAN shows, it is possible to keep resource demand at bay if one is able to properly disentangle the temporal axis from the spatial ones: The problem of generating videos did not need to be solved all at once. Instead a spatial model (STYLEGAN) was trained without a temporal dimension first. Then videos were embedded in STYLEGAN's latent space, where temporal supervision could be done on a compact representation, without even materializing actual video frames. In fact, recent work (Choi et al., 2024) uses a similar principle for the detection of fake videos (by having PSP map video frames to STYLEGAN latent codes).

Constraints to learning While it is possible to discover and exploit temporal dependencies in an automated, i.e. learning-based way, there can be advantages to avoiding learning:

If the temporal component of X_{CST} (Section 3.5) involved more learning, it might be susceptible to overfitting to the training distribution. Additional regularisers would be needed in this case to maintain generalization ability to unseen manipulation methods. Likewise, COLIBRI is a completely handcrafted method building on explicit physical priors, that prevent it from learning dependencies that would only appear to exist due to insufficient breadth of the training set. The method also satisfies the semantic constraints of events (Equations 2.19 and 2.20) *by construction* and does not need any "warm-up" before it somewhat approximates them. The fact that the method solves an optimization problem at *test* time helps it compete with methods that stoically apply learned priors to any given input, without even assessing if this input fits their training distribution or not. STYLEVIDEOGAN does learn temporal dependencies in a fully automated way, but the fact that it only sees STYLEGAN latent codes at training time prevents it from picking up too strong of a bias towards the single training subject, that would not carry over to other subjects.

In all these cases, preventing models from relying on "false" dependencies that only exist in the *available* data (but not in all the *possible* data) helps solve the problem more accurately and/or efficiently than less constrained machine learning could.

6.2 Outlook

Although Chapters 3 to 5 contribute valuable ideas to their respective areas, these tasks are still far from solved. Later works need to and partially already have put forth advanced solutions:

6.2.1 Face Video Forgery Detection

Pei et al., 2024, provide a survey of recent developments concerning face forgeries and their detection.

Among these, StyleGAN (Karras et al., 2021, 2019, 2020) was already emerging as a significant improvement to the spatial quality of face image generation at the time the work in Chapter 3 was being conducted. However, it became relevant to the detection task mostly after the properties of its latent space had been identified as being very advantageous for editing (e.g. Bounareli et al., 2023; Hou et al., 2022; D. Lee et al., 2024; Z. Liu et al., 2023; Oorloff et al., 2023 and Section 4.2). Thanks to StyleGAN, artefacts in the spatial domain have become much more subtle, and so the temporal domain is becoming relatively more important for spotting deviations from natural signals.

A second major development has been the widespread use of (latent) diffusion models (Ho et al., 2020; Rombach et al., 2022; Sohl-Dickstein et al., 2015; L. Zhang et al., 2023). Many diffusion-based models, specifically for the generation of talking faces (conditioned on audio; Du et al., 2023; G. Kim et al., 2023; Stypulkowski et al., 2024; S. Xu et al., 2024; Zhentao Yu et al., 2023), avoid the artefacts that used to be common in previous methods. For example, the blending step, where an edited/synthesised region is merged into an unedited background, is less common in diffusion models. This suggests that detectors looking for blending boundaries would work less well here. On the other hand diffusion models, despite their photorealism, still introduce visible artefacts, especially when generating videos, see Figure 6.1.

While StyleGAN and diffusion models enabled profound improvements in the synthesis of face videos, there does not seem to have been an equally fundamental change in the way that detection methods operate. There are, however, methods that use spatiotemporal attention (Vaswani et al., 2017), such as those by Z. Yang et al., 2023, and Yin et al., 2023. An increasing number of works (Choi et al., 2024; Gu et al., 2022; J. Wang et al., 2022; Z. Yang et al., 2023) is putting explicit emphasis on the temporal dimension.

In light of these recent works, it must be assumed that many new detection methods already supersede those in Section 3.5, not because they exploit a fundamentally new kind of clue, but because they refine especially the treatment of temporal dependencies, which the detectors

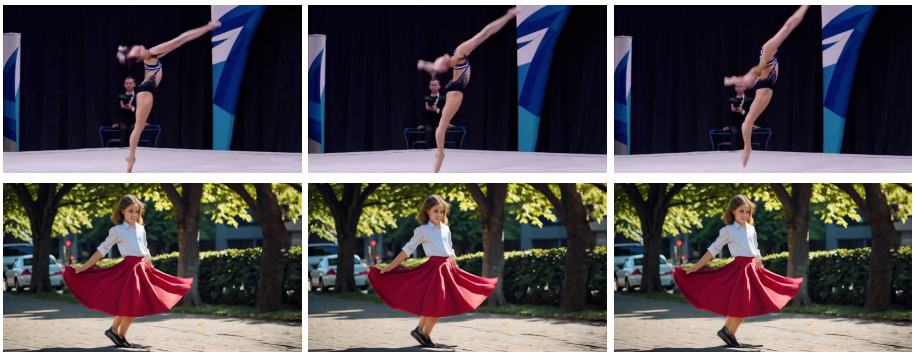


Figure 6.1: Each row shows three subsequent video frames generated by diffusion models (top: SORA, Edwards, 2024; bottom: STABLE VIDEO DIFFUSION, Blattmann et al., 2023a). These models deliver unprecedented quality, resolution and variety, in addition to being controllable by text prompts. However, the frames often violate important temporal dependencies that real footage would satisfy: The gymnast's arms morph into additional legs, while her head is unusually mobile. The girl's skirt takes a physically implausible shape, while her legs seem to fuse into one.

presented in this thesis were among the first to do (see Section 3.2.3).

Since diffusion models have superseded GANs as the state of the art in generative modelling, it seems worthwhile to explore them them for detection as well: For example, a dataset like VIDEOFORENSICSHQ, containing fake recreations of real videos, could be used to train a diffusion model to undo distortions introduced by the fake creation process (e.g. with a technique similar to SDEDIT, Meng et al., 2022). Such a model could serve both as a generator ("perfecting" fakes) and as a detector (if one learns to classify pairs of videos and their "perfected" versions).

While Chapter 3 showed that temporal dependencies can help generalise to unseen manipulation methods, the problem of generalization is still far from solved. One way of making progress here could be "multimodal" fake detection, i.e. the combined evaluation of additional information channels, such as audio (e.g. Haliassos et al., 2022).

6.2.2 Temporal Generative Models

After STYLEVIDEOGAN, several GAN-based video generators have been presented (e.g. Brooks et al., 2022; Skorokhodov et al., 2022; S. Yu et al., 2022). More recently, transformer- and diffusion-based methods have had a great impact on generative modelling. Surveys are only beginning to emerge (Lei et al., 2024; C. Li et al., 2024; R. Sun et al., 2024), but Blattmann et al., 2023a, provide a good overview in their supplemental material. Notable examples include early transformer-based methods (Ge et al.,

2022; W. Hong et al., 2023; C. Wu et al., 2022), as well as the diffusion-based IMAGEN VIDEO (Ho et al., 2022a), VIDEO LDM (Blattmann et al., 2023b), STABLE VIDEO DIFFUSION (Blattmann et al., 2023a), ANIMATEDIFF (Y. Guo et al., 2024) and SORA (Brooks et al., 2024).

While diffusion-based models supersede GANs in terms of output resolution and quality, they also have several disadvantages: Training sets are immense (e.g. WEBVID, Bain et al., 2021: originally 2, later 10 million video-text pairs, or LAION-5B, Schuhmann et al., 2022: over 5 billion image-text pairs) and inference takes more memory and time than that of GANs. Various techniques exist for reducing the computational cost, such as mixed image-and-video training (Ho et al., 2022b; Singer et al., 2023), extending spatial models by temporal dimensions (Blattmann et al., 2023b; K. Guo et al., 2024), low-rank decomposition of the weight matrices (Hu et al., 2022), or distillation (Luo et al., 2023; Meng et al., 2023). However, supervision still often happens in a spatial latent space that appears relatively costly: The widely-used STABLE DIFFUSION (Rombach et al., 2022), uses $4{\cdot}64{\cdot}64 = 16{,}384$ coefficients for each image, compared to $18 \cdot 512 = 9216$ dimensions for STYLEGAN's $\mathcal{W}^+$ space. Furthermore, the latent space is still rather isomorphic to image space. Future work could aim at further narrowing down its dimensionality and/or transforming it into a space that is easier to supervise in, in the same way that supervision in $\mathcal{W}^+$ space is cheaper than in image space.

Given the aforementioned drawbacks of diffusion models, there can still be scenarios in which GANs are preferable, due to restricted computational capacity. To this end, STYLEVIDEOGAN could be improved in several ways: PSP inversion tends to make skin appear wax-like, such that outputs appear less photorealistic than the underlying STYLEGAN model allows. Pushing the $\mathcal{W}^+$ vectors closer to the actual STYLEGAN latent space $\mathcal{W}$ (without compromising temporal stability or altering identities too much) would help reduce this problem. Such an attempt should probably be made on the basis of the more recent STYLEGAN 3 (Karras et al., 2021), which does not require training data to be aligned, but also has been found (Alaluf et al., 2022) to have a more entangled latent space. Furthermore, while the "offset trick" enables the transfer of motion to an arbitrary conditioning identity, the motion itself is being generated without any conditioning. Conditioning would be necessary for applications like temporal infilling. Investigating the latent space of STYLEVIDEOGAN and adjusting it for such uses could make it a viable competitor against diffusion-based models, in scenarios in which computational resources are not abundantly available.

6.2.3 Event-based Video Reconstruction

COLIBRI could be refined in several ways: An important property of event cameras that Chapter 5 does not treat explicitly is event latency (see Section 2.6). There are very recent learning-based works (H. Liu et al., 2024; Y. Yang et al., 2024) that explicitly model this effect. While COLIBRI does estimate logarithmic thresholds, Section 5.5 is lacking an evaluation of how well these estimates correspond to the (average) logarithmic threshold actually used by the camera: In theory, reconstruction quality could be good despite the threshold estimates being far off, because the confidence weights can compensate for such errors. Future work could explore more elaborate regularisers that rule out this ambiguity and so force the threshold estimates to be more accurate.

Several works for event-based frame interpolation (H. Cho et al., 2024; Y. Liu et al., 2024; Y. Yang et al., 2024), deblurring (T. Kim et al., 2024) and video reconstruction (Ercan et al., 2024; H. Liu et al., 2024; Y. Yang et al., 2024; X. Zhang et al., 2023) have appeared concurrently with or after COLIBRI, all learning-based. Several problems remain open:

First, there emerges a spectrum ranging from completely hand-crafted methods, that are informed by physical principles (like EDI or COLIBRI) and learning-based methods, that leave many physical details for neural networks to learn (like the work by Weng et al., 2023). It appears, however, that especially the noise characteristics of events are usually modelled only very coarsely, either by *averaging noise away* in an event binning approach, or by leaving event semantics completely for neural networks to learn. This gives ample opportunity to pick up domain-specific biases, although the core of the event semantics could be modelled explicitly (Equations 2.19 and 2.20). Instead, COLIBRI presents a novel way of addressing the noise problem, that avoids binning and training bias, by using the idealised semantics, but allowing events to be "softened" by confidence values. Future work could seek to import the confidence approach into learning-based approaches, e.g. by leaving only the confidences to be the output of a learned component, but explicitly using Equations 2.19 and 2.20.

Another strength of COLIBRI is the fact that optimization happens only once the input data is known. This is common for hand-crafted methods like EDI, but surprisingly the literature review in Section 5.2 did not encounter many learning-based methods that fine-tune their neural components at test time. Only very recent work (H. Cho et al., 2024) seems to be going into this direction, which appears to be a promising route towards bridging the domain gap between training distribution and test distribution.

Detailed architecture of StyleVideoGAN

Tables A.1 to A.4 give architecture details for Figure 4.2.

Input	Module	Outputs
s	4 layers (FC, LeakyReLU)	m (3×32)
m	BNorm(affine=True)	$(h_{0,0}, h_{0,1}, h_{0,2})$ (3×32)

Table A.1: The hallucinator H initialises the GRU memory: The first three stacked GRU cells receive vectors of length 32, the fourth receives s.

Input	Module	Outputs
$r_0, (h_{0,0}, \ldots, h_{0,3})$	GRU (4 stacked)	$(h_{1,0}, \ldots, h_{1,3}), l_1$ ($3 \times 32, 32$)
$r_1, (h_{1,0}, \ldots, h_{1,3})$	GRU (4 stacked)	$(h_{2,0}, \ldots, h_{2,3}), l_2$ ($3 \times 32, 32$)
$r_2, (h_{2,0}, \ldots, h_{2,3})$	GRU (4 stacked)	$(h_{3,0}, \ldots, h_{3,3}), l_3$ ($3 \times 32, 32$)
$\ldots$	$\ldots$	$\ldots$

Table A.2: The producer P consists of four stacked GRU cells. Its hidden state is initialised by H (Table A.1) and it recurrently turns per-time-step randomness r_i into intermediate latent codes l_{i+1} for $0 \leq i < K - 1$.

Input	Module	Outputs
l_i	BNorm(affine=True), PixelNorm	l'_i (512)
l'_i	4 layers (FC + LeakyReLU)	v'_i (512)
v'_i	BNorm(affine=True)	v_i(512)
v_i	18 parallel layers (FC + LeakyReLU + BNorm)	w_i (18×512)

Table A.3: The translator T transforms the outputs l_k of P into StyleGAN latent codes $w_k \in \mathcal{W}^+$: After a 4-layer MLP widened the dimensionality from 32 to 512, features undergo 18 independent fully connected layers (LeakyReLU activation), similar to how StyleGAN broadcasts its $\mathcal{W}$ vectors to its 18 style layers. The PixelNorm module divides input tensors by the square root of their average squared entries (Seonghyeon et al., 2020).

Input	Module	Outputs
w_i	FC + LeakyReLU (2 layers)	e'_i (512)
e'_i	FC + LeakyReLU (4 layers)	e_i (32)
$e_0, \ldots, e_{K-1}$	1D-version of the DCGAN critic (Radford et al., 2016), with 32 input channels	Critic Output (1)

Table A.4: The latent critic C takes a sequence $w_0, \ldots, w_{K-1}$ of StyleGAN latent codes as input and produces a scalar output.

DAVIS 346C Exposure Coverage

B

Table B.1 lists the framerates obtained for different exposure durations in the DAVIS 346C. This data is visualised in Figures 5.3 and 5.4.

Exposure	FPS	Gap per frame	Coverage
900,000µs	1.1Hz	9090µs	99.00%
800,000µs	1.2Hz	33,333µs	96.00%
700,000µs	1.4Hz	14,285µs	98.00%
600,000µs	1.6Hz	25,000µs	96.00%
500,000µs	1.9Hz	26,315µs	95.00%
400,000µs	2.4Hz	16,666µs	96.00%
300,000µs	3.15Hz	17,460µs	94.50%
200,000µs	4.6Hz	17,391µs	92.00%
100,000µs	8.45Hz	18,343µs	84.50%
90,000µs	9.25Hz	18,108µs	83.25%
80,000µs	10.2Hz	18,039µs	81.60%
70,000µs	11.3Hz	18,495µs	79.10%
60,000µs	12.8Hz	18,125µs	76.80%
50,000µs	14.7Hz	18,027µs	73.50%
40,000µs	17.15Hz	18,309µs	68.60%
30,000µs	20.5Hz	18,780µs	61.50%
20,000µs	26.0Hz	18,461µs	52.00%
10,000µs	35.0Hz	18,571µs	35.00%
9000µs	36.5Hz	18,397µs	32.85%
8000µs	38.1Hz	18,246µs	30.48%
7000µs	39.0Hz	18,641µs	27.30%
6000µs	41.0Hz	18,390µs	24.60%
5000µs	43.0Hz	18,255µs	21.50%
4000µs	46.8Hz	17,367µs	18.72%
3000µs	47.0Hz	18,276µs	14.10%
2000µs	49.0Hz	18,408µs	9.80%
1000µs	51.6Hz	18,379µs	5.16%
500µs	53.0Hz	18,367µs	2.65%
100µs	54.0Hz	18,418µs	0.54%
10µs	54.0Hz	18,508µs	0.05%

Table B.1: Different exposure durations were set for the DAVIS 346C, measuring the resulting framerates. This reveals that the exposure gaps cannot be pushed under a non-negligible minimum of about 18ms. As the gaps become more frequent with higher framerates, the percentage of sequence time that is actually covered by exposures decreases.

References

Abdal, Rameen, Yipeng Qin, and Peter Wonka **(2019)**: "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?" In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, pp. 4431–4440.

— **(2020)**: "Image2StyleGAN++: How to Edit the Embedded Images?" In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 8293–8302.

Afchar, Darius, Vincent Nozick, Junichi Yamagishi, and Isao Echizen **(2018)**: "MesoNet: a Compact Facial Video Forgery Detection Network". In: *2018 IEEE International Workshop on Information Forensics and Security, WIFS 2018, Hong Kong, China, December 11-13, 2018*. IEEE, pp. 1–7.

Agarwal, Shruti, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li **(2019)**: "Protecting World Leaders Against Deep Fakes". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 38–45.

Alaluf, Yuval, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or **(2022)**: "Third Time's the Charm? Image and Video Editing with StyleGAN3". In: *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part II*. Ed. by Leonid Karlinsky, Tomer Michaeli, and Ko Nishino. Vol. 13802. Lecture Notes in Computer Science. Springer, pp. 204–220.

Arjovsky, Martín, Soumith Chintala, and Léon Bottou **(2017)**: "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 214–223.

Babaeizadeh, Mohammad, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine **(2018)**: "Stochastic Variational Video Prediction". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Bain, Max, Arsha Nagrani, Gül Varol, and Andrew Zisserman **(2021)**: "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval". In: *2021 IEEE/CVF International Conference on Computer*

Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. IEEE, pp. 1708–1718.

Bardow, Patrick, Andrew J. Davison, and Stefan Leutenegger **(2016)**: "Simultaneous Optical Flow and Intensity Estimation from an Event Camera". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 884–892.

Barua, Souptik, Yoshitaka Miyatani, and Ashok Veeraraghavan **(2016)**: "Direct face detection and video reconstruction from event cameras". In: *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7-10, 2016*. IEEE Computer Society, pp. 1–9.

Bayar, Belhassen and Matthew C. Stamm **(2016)**: "A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer". In: *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2016, Vigo, Galicia, Spain, June 20-22, 2016*. Ed. by Fernando Pérez-González, Patrick Bas, Tanya Ignatenko, and François Cayre. ACM, pp. 5–10.

— **(2018)**: "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection". In: *IEEE Trans. Inf. Forensics Secur.* 13.11, pp. 2691–2706.

Blanz, Volker and Thomas Vetter **(1999)**: "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*. Ed. by Warren N. Waggenspack. ACM, pp. 187–194.

Blattmann, Andreas, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach **(2023a)**: "Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets". In: *CoRR* abs/2311.15127. arXiv: 2311.15127.

Blattmann, Andreas, Timo Milbich, Michael Dorkenwald, and Björn Ommer **(2021)**: "iPOKE: Poking a Still Image for Controlled Stochastic Video Synthesis". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 14687–14697.

Blattmann, Andreas, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis **(2023b)**: "Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 22563–22575.

Bounareli, Stella, Christos Tzelepis, Vasileios Argyriou, Ioannis Patras, and Georgios Tzimiropoulos **(2023)**: "StyleMask: Disentangling the

Style Space of StyleGAN2 for Neural Face Reenactment". In: *17th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2023, Waikoloa Beach, HI, USA, January 5-8, 2023*. IEEE, pp. 1–8.

Brandli, Christian, Lorenz Müller, and Tobi Delbrück **(2014)**: "Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor". In: *IEEE International Symposium on Circuits and Systemss, ISCAS 2014, Melbourne, Victoria, Australia, June 1-5, 2014*. IEEE, pp. 686–689.

Brock, Andrew, Jeff Donahue, and Karen Simonyan **(2019)**: "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Brooks, Tim, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A. Efros, and Tero Karras **(2022)**: "Generating Long Videos of Dynamic Scenes". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh.

Brooks, Tim, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh **(2024)**: *Video generation models as world simulators*. https://openai.com/research/video-generation-models-as-world-simulators. Accessed: April 11, 2025.

Cadena, Pablo Rodrigo Gantier, Yeqiang Qian, Chunxiang Wang, and Ming Yang **(2021)**: "SPADE-E2VID: Spatially-Adaptive Denormalization for Event-Based Video Reconstruction". In: *IEEE Trans. Image Process.* 30, pp. 2488–2500.

Castrejón, Lluís, Nicolas Ballas, and Aaron C. Courville **(2019)**: "Improved Conditional VRNNs for Video Prediction". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, pp. 7607–7616.

Chakravarthi, Bharatesh, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermüller, and Yezhou Yang **(2024)**: "Recent Event Camera Innovations: A Survey". In: *CoRR* abs/2408.13627. arXiv: 2408.13627.

Chen, Jiaben, Yichen Zhu, Dongze Lian, Jiaqi Yang, Yifu Wang, Renrui Zhang, Xinhang Liu, Shenhan Qian, Laurent Kneip, and Shenghua Gao **(2023)**: "Revisiting Event-Based Video Frame Interpolation". In: *IROS*, pp. 1292–1299.

Cho, Hoonhee, Yuhwan Jeong, Taewoo Kim, and Kuk-Jin Yoon **(2023)**: "Non-Coaxial Event-guided Motion Deblurring with Spatial Alignment". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 12458–12469.

Cho, Hoonhee, Taewoo Kim, Yuhwan Jeong, and Kuk-Jin Yoon **(2024)**: "TTA-EVF: Test-Time Adaptation for Event-based Video Frame Interpolation via Reliable Pixel and Sample Estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 25701–25711.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio **(2014)**: "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, pp. 1724–1734.

Choi, Jongwook, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi **(2024)**: "Exploiting Style Latent Flows for Generalizing Deepfake Video Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 1133–1143.

Chollet, François **(2017)**: "Xception: Deep Learning with Depthwise Separable Convolutions". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 1800–1807.

Cover, Thomas M. and Joy A. Thomas **(1991)**: *Elements of Information Theory*. Ed. by Donald L. Schilling. New York: Wiley-Interscience. ISBN: 0-471-20061-1.

Cozzolino, Davide, Diego Gragnaniello, and Luisa Verdoliva **(2014)**: "Image forgery detection through residual-based local descriptors and block-matching". In: *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*. IEEE, pp. 5297–5301.

Cozzolino, Davide, Giovanni Poggi, and Luisa Verdoliva **(2017)**: "Recasting Residual-based Local Descriptors as Convolutional Neural Networks: an Application to Image Forgery Detection". In: *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2017, Philadelphia, PA, USA, June 20-22, 2017*. Ed. by Matthew C. Stamm, Matthias Kirchner, and Sviatoslav Voloshynovskiy. ACM, pp. 159–164.

Cozzolino, Davide and Luisa Verdoliva **(2020)**: "Noiseprint: A CNN-Based Camera Model Fingerprint". In: *IEEE Trans. Inf. Forensics Secur.* 15, pp. 144–159.

Denton, Emily and Rob Fergus **(2018)**: "Stochastic Video Generation with a Learned Prior". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden,*

July 10-15, 2018. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1182–1191.

Denton, Emily L. and Vighnesh Birodkar **(2017)**: "Unsupervised Learning of Disentangled Representations from Video". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 4414–4423.

Dolhansky, Brian, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton-Ferrer **(2020)**: "The DeepFake Detection Challenge Dataset". In: *CoRR* abs/2006.07397. arXiv: 2006.07397.

Dolhansky, Brian, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton-Ferrer **(2019)**: "The Deepfake Detection Challenge (DFDC) Preview Dataset". In: *CoRR* abs/1910.08854. arXiv: 1910.08854.

Dorkenwald, Michael, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G. Derpanis, and Björn Ommer **(2021)**: "Stochastic Image-to-Video Synthesis Using cINNs". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 3742–3753.

Du, Chenpeng, Qi Chen, Tianyu He, Xu Tan, Xie Chen, Kai Yu, Sheng Zhao, and Jiang Bian **(2023)**: "DAE-Talker: High Fidelity Speech-Driven Talking Face Generation with Diffusion Autoencoder". In: *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*. Ed. by Abdulmotaleb El-Saddik, Tao Mei, Rita Cucchiara, Marco Bertini, Diana Patricia Tobon Vallejo, Pradeep K. Atrey, and M. Shamim Hossain. ACM, pp. 4281–4289.

Dufour, Nick and Andrew Gully **(2019)**: *Contributing Data to Deepfake Detection Research. Google Blog.*

Durall, Ricard, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper **(2019)**: "Unmasking DeepFakes with simple Features". In: *CoRR* abs/1911.00686. arXiv: 1911.00686.

Edwards, Benj **(2024)**: *Twirling body horror in gymnastics video exposes AI's flaws*. https://arstechnica.com/information-technology/2024/12/twirling-body-horror-in-gymnastics-video-exposes-ais-flaws/. Accessed: 2025-03-18.

Ercan, Burak, Onur Eker, Canberk Saglam, Aykut Erdem, and Erkut Erdem **(2024)**: "HyperE2VID: Improving Event-Based Video Reconstruction via Hypernetworks". In: *IEEE Trans. Image Process.* 33, pp. 1826–1837.

Fox, Gereon, Wentao Liu, Hyeongwoo Kim, Hans-Peter Seidel, Mohamed Elgharib, and Christian Theobalt **(2021a)**: "VideoForensicsHQ: Detect-

ing High-quality Manipulated Face Videos". In: *2021 IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5-9, 2021*. IEEE, pp. 1–6.

Fox, Gereon, Xingang Pan, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt **(2024)**: "Unsupervised Event-Based Video Reconstruction". In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, pp. 4167–4176.

Fox, Gereon, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt **(2021b)**: "StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN". In: *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*. BMVA Press, p. 220.

Franceschi, Jean-Yves, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari **(2020)**: "Stochastic Latent Residual Video Prediction". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 3233–3246.

Fridrich, Jessica J. and Jan Kodovský **(2012)**: "Rich Models for Steganalysis of Digital Images". In: *IEEE Trans. Inf. Forensics Secur.* 7.3, pp. 868–882.

Fried, Ohad, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B. Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala **(2019)**: "Text-based editing of talking-head video". In: *ACM Trans. Graph.* 38.4, 68:1–68:14.

Gallego, Guillermo, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza **(2022)**: "Event-Based Vision: A Survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.1, pp. 154–180.

Gao, Yue, Siqi Li, Yipeng Li, Yandong Guo, and Qionghai Dai **(2023)**: "SuperFast: 200× Video Frame Interpolation via Event Camera". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.6, pp. 7764–7780.

Garrido, Pablo, Levi Valgaerts, Ole Rehmsen, Thorsten Thormählen, Patrick Pérez, and Christian Theobalt **(2014)**: "Automatic Face Reenactment". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, pp. 4217–4224.

Garrido, Pablo, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt **(2016)**: "Reconstruction of Personalized 3D Face Rigs from Monocular Video". In: *ACM Trans. Graph.* 35.3, 28:1–28:15.

Ge, Songwei, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh **(2022)**: "Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Is-*

rael, October 23-27, 2022, Proceedings, Part XVII. Ed. by Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13677. Lecture Notes in Computer Science. Springer, pp. 102–118.

Gehrig, Daniel, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza **(2020)**: "Video to Events: Recycling Video Datasets for Event Cameras". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 3583–3592.

Geitgey, Adam, Anand Baburajan, Enric Moreu, Tommy in Tongji, Santiago Castro, Yumi Lee, Abdolkarim Saeedi, sgc097, Konstantin Krestnikov, Aref Ariyapour, Tejas Shah, Marcus Medley, Ellie Kang, et al. **(2020)**: *face_recognition*. https://github.com/ageitgey/face_recognition. Accessed: November 19, 2020.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio **(2014)**: "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, pp. 2672–2680.

Gu, Zhihao, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, and Lizhuang Ma **(2022)**: "Delving into the Local: Dynamic Inconsistency Learning for DeepFake Video Detection". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, pp. 744–752.

Guan, Haiying, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N. Yates, Andrew Delgado, Daniel Zhou, Timothée Kheyrkhah, Jeff Smith, and Jonathan G. Fiscus **(2019)**: "MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation". In: *IEEE Winter Applications of Computer Vision Workshops, WACV Workshops 2019, Waikoloa Village, HI, USA, January 7-11, 2019*. IEEE, pp. 63–72.

Gulrajani, Ishaan, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville **(2017)**: "Improved Training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 5767–5777.

Guo, Kun, Haochen Zhu, and Gang Cao **(2024)**: "Effective Image Tampering Localization Via Enhanced Transformer and Co-Attention Fusion". In: *IEEE International Conference on Acoustics, Speech and Signal Pro-*

cessing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024. IEEE, pp. 4895–4899.

Guo, Yuwei, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai **(2024)**: "AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning". In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Haliassos, Alexandros, Rodrigo Mira, Stavros Petridis, and Maja Pantic **(2022)**: "Leveraging Real Talking Faces via Self-Supervision for Robust Forgery Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 14930–14942.

Han, Jin, Yixin Yang, Chu Zhou, Chao Xu, and Boxin Shi **(2021)**: "EvIntSR-Net: Event Guided Multiple Latent Frames Reconstruction and Super-resolution". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 4862–4871.

Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris **(2020)**: "GANSpace: Discovering Interpretable GAN Controls". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin.

He, Weihua, Kaichao You, Zhendong Qiao, Xu Jia, Ziyang Zhang, Wenhui Wang, Huchuan Lu, Yaoyuan Wang, and Jianxing Liao **(2022)**: "TimeReplayer: Unlocking the Potential of Event Cameras for Video Interpolation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 17783–17792.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter **(2017)**: "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 6626–6637.

Ho, Jonathan, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans **(2022a)**: "Imagen Video: High Definition Video Generation with Diffusion Models". In: *CoRR* abs/2210.02303. arXiv: 2210.02303.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel **(2020)**: "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin.

Ho, Jonathan, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet **(2022b)**: "Video Diffusion Models". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh.

Hong, Kibeom, Youngjung Uh, and Hyeran Byun **(2021)**: "ArrowGAN : Learning to generate videos by learning Arrow of Time". In: *Neurocomputing* 438, pp. 223–234.

Hong, Wenyi, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang **(2023)**: "CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hou, Xianxu, Xiaokang Zhang, Hanbang Liang, Linlin Shen, Zhihui Lai, and Jun Wan **(2022)**: "GuidedStyle: Attribute knowledge guided style manipulation for semantic face editing". In: *Neural Networks* 145, pp. 209–220.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen **(2022)**: "LoRA: Low-Rank Adaptation of Large Language Models". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Huh, Minyoung, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann **(2020)**: "Transforming and Projecting Images into Class-Conditional Generative Networks". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12347. Lecture Notes in Computer Science. Springer, pp. 17–34.

Ioffe, Sergey and Christian Szegedy **(2015)**: "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 448–456.

ISO 9288:2022(en) **(Aug. 2022)**: *Thermal insulation – Heat transfer by radiation – Vocabulary*. Standard. International Organization for Standardization.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros **(2017)**: "Image-to-Image Translation with Conditional Adversarial Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 5967–5976.

Jia, Xu, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool **(2016)**: "Dynamic Filter Networks". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, pp. 667–675.

Jiang, Liming, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy **(2020)**: "DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 2886–2895.

Jiang, Zhe, Yu Zhang, Dongqing Zou, Jimmy S. J. Ren, Jiancheng Lv, and Yebin Liu **(2020)**: "Learning Event-Based Motion Deblurring". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 3317–3326.

Kahembwe, Emmanuel and Subramanian Ramamoorthy **(2020)**: "Lower dimensional kernels for video discriminators". In: *Neural Networks* 132, pp. 506–520.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen **(2018)**: "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Karras, Tero, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila **(2021)**: "Alias-Free Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, pp. 852–863.

Karras, Tero, Samuli Laine, and Timo Aila **(2019)**: "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 4401–4410.

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila **(2020)**: "Analyzing and Improving the Image Quality of StyleGAN". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 8107–8116.

Kiliç, Onur Selim, Ahmet Akman, and A. Aydin Alatan **(2023)**: "E-VFIA: Event-Based Video Frame Interpolation with Attention". In: *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*. IEEE, pp. 8284–8290.

Kim, Gyeongman, Hajin Shim, Hyunsu Kim, Yunjey Choi, Junho Kim, and Eunho Yang **(2023)**: "Diffusion Video Autoencoders: Toward Temporally Consistent Face Video Editing via Disentangled Video Encoding". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 6091–6100.

Kim, Hyeongwoo, Mohamed Elgharib, Michael Zollhöfer, Hans-Peter Seidel, Thabo Beeler, Christian Richardt, and Christian Theobalt **(2019)**: "Neural style-preserving visual dubbing". In: *ACM Trans. Graph.* 38.6, 178:1–178:13.

Kim, Hyeongwoo, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt **(2018)**: "Deep video portraits". In: *ACM Trans. Graph.* 37.4, p. 163.

Kim, Taewoo, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon **(2023)**: "Event-based Video Frame Interpolation with Cross-Modal Asymmetric Bidirectional Motion Fields". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 18032–18042.

Kim, Taewoo, Hoonhee Cho, and Kuk-Jin Yoon **(2024)**: "Frequency-Aware Event-Based Video Deblurring for Real-World Motion Blur". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 24966–24976.

Kim, Taewoo, Jeongmin Lee, Lin Wang, and Kuk-Jin Yoon **(2022)**: "Event-guided Deblurring of Unknown Exposure Time Videos". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVIII*. Ed. by Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13678. Lecture Notes in Computer Science. Springer, pp. 519–538.

King, Davis E. **(2009)**: "Dlib-ml: A Machine Learning Toolkit". In: *J. Mach. Learn. Res.* 10, pp. 1755–1758.

Kingma, Diederik P. and Jimmy Ba **(2015)**: "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Kobiela, Dorota, Hugh Welchman, Sean Bobbit, and Ivan Mactaggart **(2017)**: *Loving Vincent*. Poland and United Kingdom: BreakThru Films and Trademark Films.

Korshunov, Pavel and Sébastien Marcel **(2019)**: "Vulnerability assessment and detection of Deepfake videos". In: *2019 International Conference on Biometrics, ICB 2019, Crete, Greece, June 4-7, 2019*. IEEE, pp. 1–6.

Kowalski, Marek **(2018)**: *FaceSwap*. https://github.com/MarekKowalski/FaceSwap. Accessed: February 15, 2018.

Lee, Dongyeun, Chaewon Kim, Sangjoon Yu, Jaejun Yoo, and Gyeong-Moon Park **(2024)**: "RADIO: Reference-Agnostic Dubbing Video Synthesis". In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, pp. 4156–4166.

Lei, Wentao, Jinting Wang, Fengji Ma, Guanjie Huang, and Li Liu **(2024)**: "A Comprehensive Survey on Human Video Generation: Challenges, Methods, and Insights". In: *CoRR* abs/2407.08428. arXiv: 2407.08428.

Leñero-Bardallo, Juan Antonio, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco **(2011)**: "A 3.6 μ s Latency Asynchronous Frame-Free Event-Driven Dynamic-Vision-Sensor". In: *IEEE Journal of Solid-State Circuits* 46.6, pp. 1443–1455.

Li, Chengxuan, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai **(2024)**: "A Survey on Long Video Generation: Challenges, Methods, and Prospects". In: *CoRR* abs/2403.16407. arXiv: 2403.16407.

Li, Lingzhi, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo **(2020)**: "Face X-Ray for More General Face Forgery Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 5000–5009.

Li, Yuezun, Ming-Ching Chang, and Siwei Lyu **(2018)**: "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking". In: *2018 IEEE International Workshop on Information Forensics and Security, WIFS 2018, Hong Kong, China, December 11-13, 2018*. IEEE, pp. 1–7.

Li, Yuezun, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu **(2020)**: "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 3204–3213.

Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbrück **(2008)**: "A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE J. Solid State Circuits* 43.2, pp. 566–576.

Lin, Songnan, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy S. J. Ren **(2020)**: "Learning Event-Driven Video Deblurring and Interpolation". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28,*

2020, Proceedings, Part VIII. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12353. Lecture Notes in Computer Science. Springer, pp. 695–710.

Liu, Haoyue, Shihan Peng, Lin Zhu, Yi Chang, Hanyu Zhou, and Luxin Yan **(2024)**: "Seeing Motion at Nighttime with an Event Camera". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 25648–25658.

Liu, Yuhan, Yongjian Deng, Hao Chen, and Zhen Yang **(2024)**: "Video Frame Interpolation via Direct Synthesis with the Event-based Reference". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 8477–8487.

Liu, Zhian, Maomao Li, Yong Zhang, Cairong Wang, Qi Zhang, Jue Wang, and Yongwei Nie **(2023)**: "Fine-Grained Face Swapping Via Regional GAN Inversion". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 8578–8587.

Livingstone, Steven R. and Frank A. Russo **(May 2018)**: "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PLOS ONE* 13.5, pp. 1–35.

Luo, Simian, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao **(2023)**: "Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference". In: *CoRR* abs/2310.04378. arXiv: 2310.04378.

McMahon-Crabtree, Peter N., Lucas Kulesza, Brian J. McReynolds, Daniel S. O'Keefe, Anirvin Puttur, Diana Maestas, Christian P. Morath, and Matthew G. McHarg **(2023)**: "Event-based camera refractory period characterization and initial clock drift evaluation". In: *Unconventional Imaging, Sensing, and Adaptive Optics 2023*. Ed. by Jean J. Dolne, Mark F. Spencer, and Santasri R. Bose-Pillai. Vol. 12693. International Society for Optics and Photonics. SPIE, p. 126930V.

Meng, Chenlin, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon **(2022)**: "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Meng, Chenlin, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans **(2023)**: "On Distillation of Guided Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 14297–14306.

Menick, Jacob and Nal Kalchbrenner **(2019)**: "Generating High fidelity Images with subscale pixel Networks and Multidimensional Upscaling".

In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Munda, Gottfried, Christian Reinbacher, and Thomas Pock **(2018)**: "Real-Time Intensity-Image Reconstruction for Event Cameras Using Manifold Regularisation". In: *Int. J. Comput. Vis.* 126.12, pp. 1381–1393.

Muñoz, Andrés, Mohammadreza Zolfaghari, Max Argus, and Thomas Brox **(2021)**: "Temporal Shift GAN for Large Scale Video Generation". In: *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, pp. 3178–3187.

Nagano, Koki, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li **(2018)**: "paGAN: real-time avatars using dynamic textures". In: *ACM Trans. Graph.* 37.6, p. 258.

Nah, Seungjun, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee **(2019)**: "NTIRE 2019 Challenge on Video Deblurring and Super-Resolution: Dataset and Study". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 1996–2005.

Nitzan, Yotam, Amit Bermano, Yangyan Li, and Daniel Cohen-Or **(2020)**: "Face identity disentanglement via latent space mapping". In: *ACM Trans. Graph.* 39.6, 225:1–225:14.

Oorloff, Trevine and Yaser Yacoob **(2023)**: "Robust One-Shot Face Video Reenactment using Hybrid Latent Spaces of StyleGAN2". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 20890–20900.

Paikin, Genady, Yotam Ater, Roy Shaul, and Evgeny Soloveichik **(2021)**: "EFI-Net: Video Frame Interpolation From Fusion of Events and Frames". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 1291–1301.

Pajoum, Barbod, Gereon Fox, Mohamed Elgharib, Marc Habermann, and Christian Theobalt **(2024)**: "Adaptive Grids for Neural Scene Representation". In: *29th International Symposium on Vision, Modeling, and Visualization, VMV 2024, Munich, Germany, September 10-13, 2024*. Ed. by Lars Linsen and Justus Thies. Eurographics Association.

Pan, Liyuan, Richard Hartley, Cedric Scheerlinck, Miaomiao Liu, Xin Yu, and Yuchao Dai **(2022)**: "High Frame Rate Video Reconstruction Based on an Event Camera". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.5, pp. 2519–2533.

Pan, Liyuan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai **(2019)**: "Bringing a Blurry Frame Alive at High Frame-Rate With an Event Camera". In: *IEEE Conference on Computer Vision*

and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, pp. 6820–6829.

Paredes-Vallés, Federico and Guido C. H. E. de Croon **(2021)**: "Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 3446–3455.

Pei, Gan, Jiangning Zhang, Menghan Hu, Guangtao Zhai, Chengjie Wang, Zhenyu Zhang, Jian Yang, Chunhua Shen, and Dacheng Tao **(2024)**: "Deepfake Generation and Detection: A Benchmark and Survey". In: *CoRR* abs/2403.17881. arXiv: 2403.17881.

Pidhorskyi, Stanislav, Donald A. Adjeroh, and Gianfranco Doretto **(2020)**: "Adversarial Latent Autoencoders". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 14092–14101.

Qian, Yuyang, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao **(2020)**: "Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12357. Lecture Notes in Computer Science. Springer, pp. 86–103.

Radford, Alec, Luke Metz, and Soumith Chintala **(2016)**: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Raghavendra, Ramachandra, Kiran B. Raja, Sushma Venkatesh, and Christoph Busch **(2017)**: "Transferable Deep-CNN Features for Detecting Digital and Print-Scanned Morphed Face Images". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 1822–1830.

Rahmouni, Nicolas, Vincent Nozick, Junichi Yamagishi, and Isao Echizen **(2017)**: "Distinguishing computer graphics from natural images using convolution neural networks". In: *2017 IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 4-7, 2017*. IEEE, pp. 1–6.

Rao, Pramod, Mallikarjun B R, Gereon Fox, Tim Weyrich, Bernd Bickel, Hanspeter Pfister, Wojciech Matusik, Ayush Tewari, Christian Theobalt, and Mohamed Elgharib **(2022)**: "VoRF: Volumetric Relightable Faces". In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, p. 708.

Rao, Pramod, Gereon Fox, Abhimitra Meka, Mallikarjun B. R., Fangneng Zhan, Tim Weyrich, Bernd Bickel, Hanspeter Pfister, Wojciech Matusik, Mohamed Elgharib, and Christian Theobalt **(2024a)**: "Lite2Relight: 3D-aware Single Image Portrait Relighting". In: *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024*. Ed. by Andres Burbano, Denis Zorin, and Wojciech Jarosz. ACM, p. 41.

Rao, Pramod, Mallikarjun B. R., Gereon Fox, Tim Weyrich, Bernd Bickel, Hanspeter Pfister, Wojciech Matusik, Fangneng Zhan, Ayush Tewari, Christian Theobalt, and Mohamed Elgharib **(2024b)**: "A Deeper Analysis of Volumetric Relightable Faces". In: *Int. J. Comput. Vis.* 132.4, pp. 1148–1166.

Rebecq, Henri, Daniel Gehrig, and Davide Scaramuzza **(2018)**: "ESIM: an Open Event Camera Simulator". In: *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*. Vol. 87. Proceedings of Machine Learning Research. PMLR, pp. 969–982.

Rebecq, Henri, René Ranftl, Vladlen Koltun, and Davide Scaramuzza **(2019)**: "Events-To-Video: Bringing Modern Computer Vision to Event Cameras". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 3857–3866.

— **(2021)**: "High Speed and High Dynamic Range Video with an Event Camera". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 43.6, pp. 1964–1980.

Reda, Fitsum, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless **(2022)**: *Tensorflow 2 Implementation of "FILM: Frame Interpolation for Large Motion"*. https://github.com/google-research/frame-interpolation. Accessed: January 14, 2023.

Reda, Fitsum A., Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless **(2022)**: "FILM: Frame Interpolation for Large Motion". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VII*. Ed. by Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13667. Lecture Notes in Computer Science. Springer, pp. 250–266.

Reinbacher, Christian, Gottfried Graber, and Thomas Pock **(2016)**: "Real-Time Intensity-Image Reconstruction for Event Cameras Using Manifold Regularisation". In: *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. Ed. by Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith. BMVA Press.

Richardson, Elad, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or **(2021)**: "Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation". In: *IEEE Conference*

on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, pp. 2287–2296.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer **(2022)**: "High-Resolution Image Synthesis with Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 10674–10685.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox **(2015)**: "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi. Vol. 9351. Lecture Notes in Computer Science. Springer, pp. 234–241.

Rössler, Andreas, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner **(2019)**: "FaceForensics++: Learning to Detect Manipulated Facial Images". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, pp. 1–11.

Rudnev, Viktor, Gereon Fox, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik **(2024)**: "Dynamic EventNeRF: Reconstructing General Dynamic Scenes from Multi-view Event Cameras". In: *CoRR* abs/2412.06770. arXiv: 2412.06770.

Rudnev, Viktor, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt **(2021)**: "EventHands: Real-Time Neural 3D Hand Pose Estimation from an Event Stream". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 12365–12375.

Sabir, Ekraam, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan **(2019)**: "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 80–87.

Saito, Masaki, Eiichi Matsumoto, and Shunta Saito **(2017)**: "Temporal Generative Adversarial Nets with Singular Value Clipping". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 2849–2858.

Saito, Masaki, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi **(2020)**: "Train Sparsely, Generate Densely: Memory-Efficient Unsupervised Training of High-Resolution Temporal GAN". In: *Int. J. Comput. Vis.* 128.10, pp. 2586–2606.

Saragih, Jason M., Simon Lucey, and Jeffrey F. Cohn **(2011)**: "Deformable Model Fitting by Regularized Landmark Mean-Shift". In: *Int. J. Comput. Vis.* 91.2, pp. 200–215.

Scheerlinck, Cedric, Nick Barnes, and Robert E. Mahony **(2018)**: "Continuous-Time Intensity Estimation Using Event Cameras". In: *Computer Vision - ACCV 2018 - 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part V*. Ed. by C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler. Vol. 11365. Lecture Notes in Computer Science. Springer, pp. 308–324.

Scheerlinck, Cedric, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert E. Mahony, and Davide Scaramuzza **(2020)**: "Fast Image Reconstruction with an Event Camera". In: *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*. IEEE, pp. 156–163.

Scheerlinck, Cedric, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert E. Mahony, and Davide Scaramuzza **(2019)**: "CED: Color Event Camera Dataset". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 1684–1693.

Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev **(2022)**: "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh.

Seonghyeon, Kim, Levin Dabhi, Jackerz312, woctezuma, nivha, Christian Clauss, Terence Broad, matanby, and onion **(2020)**: *StyleGAN 2 in PyTorch*. https://github.com/rosinality/stylegan2-pytorch. Accessed: September 24, 2020.

Serrano-Gotarredona, Teresa and Bernabé Linares-Barranco **(2013)**: "A 128×128 1.5% Contrast Sensitivity 0.9% FPN 3 μs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers". In: *IEEE J. Solid State Circuits* 48.3, pp. 827–838.

Shang, Wei, Dongwei Ren, Dongqing Zou, Jimmy S. Ren, Ping Luo, and Wangmeng Zuo **(2021)**: "Bringing Events into Video Deblurring with Non-consecutively Blurry Frames". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 4511–4520.

Shen, Yujun, Jinjin Gu, Xiaoou Tang, and Bolei Zhou **(2020)**: "Interpreting the Latent Space of GANs for Semantic Face Editing". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 9240–9249.

Singer, Uriel, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman **(2023)**: "Make-A-Video: Text-to-Video Generation without Text-Video Data". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Skorokhodov, Ivan, Sergey Tulyakov, and Mohamed Elhoseiny **(2022)**: "StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 3616–3626.

Sohl-Dickstein, Jascha, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli **(2015)**: "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 2256–2265.

Song, Chen, Qixing Huang, and Chandrajit Bajaj **(2022)**: "E-CIR: Event-Enhanced Continuous Intensity Recovery". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 7793–7802.

Stoffregen, Timo, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert E. Mahony **(2020)**: "Reducing the Sim-to-Real Gap for Event Cameras". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVII*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12372. Lecture Notes in Computer Science. Springer, pp. 534–549.

Stypulkowski, Michal, Konstantinos Vougioukas, Sen He, Maciej Zieba, Stavros Petridis, and Maja Pantic **(2024)**: "Diffused Heads: Diffusion Models Beat GANs on Talking-Face Generation". In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, pp. 5089–5098.

Sun, Lei, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool **(2022)**: "Event-Based Fusion for Motion Deblurring with Cross-modal Attention". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVIII*. Ed. by Shai Avidan, Gabriel J.

Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13678. Lecture Notes in Computer Science. Springer, pp. 412–428.

Sun, Lei, Christos Sakaridis, Jingyun Liang, Peng Sun, Kai Zhang, Jiezhang Cao, Qi Jiang, Kaiwei Wang, and Luc Van Gool **(2023)**: "Event-Based Frame Interpolation with Ad-hoc Deblurring". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 18043–18052.

Sun, Rui, Yumin Zhang, Tejal Shah, Jiahao Sun, Shuoying Zhang, Wenqi Li, Haoran Duan, Bo Wei, and Rajiv Ranjan **(2024)**: "From Sora What We Can See: A Survey of Text-to-Video Generation". In: *CoRR* abs/2405.10674. arXiv: 2405.10674.

Suwajanakorn, Supasorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman **(2017)**: "Synthesizing Obama: learning lip sync from audio". In: *ACM Trans. Graph.* 36.4, 95:1–95:13.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi **(2017)**: "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, pp. 4278–4284.

Tewari, Ayush, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt **(2020a)**: "StyleRig: Rigging StyleGAN for 3D Control Over Portrait Images". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 6141–6150.

Tewari, Ayush, Mohamed Elgharib, Mallikarjun B. R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt **(2020b)**: "PIE: portrait image embedding for semantic control". In: *ACM Trans. Graph.* 39.6, 223:1–223:14.

Thies, Justus, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner **(2020)**: "Neural Voice Puppetry: Audio-Driven Facial Reenactment". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12361. Lecture Notes in Computer Science. Springer, pp. 716–731.

Thies, Justus, Michael Zollhöfer, and Matthias Nießner **(2019)**: "Deferred neural rendering: image synthesis using neural textures". In: *ACM Trans. Graph.* 38.4, 66:1–66:12.

Thies, Justus, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner **(2016)**: "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos". In: *2016 IEEE Conference on Computer*

Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, pp. 2387–2395.

Tian, Yu, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov **(2021)**: "A Good Image Generator Is What You Need for High-Resolution Video Synthesis". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tolosana, Ruben, Rubén Vera-Rodríguez, Julian Fiérrez, Aythami Morales, and Javier Ortega-Garcia **(2020)**: "Deepfakes and beyond: A Survey of face manipulation and fake detection". In: *Inf. Fusion* 64, pp. 131–148.

torzdf, kvrooman, kilroythethird, Clorr, Artem Ivanov, Gareth Dunstone, Bryan Lyon, Hidde Jansen, deepfakes, iperov, Othniel Cundangan, Andy Kim, Tim van den Essen, Lev Velykoivanenko, Willisseck Édouard, joshua-wu, JayantPythonLover, coldstacks, Lorjuo, leftler, dfaker, GSonderling, et al. **(2020)**: *deepfakes_faceswap*. https://github.com/deepfakes/faceswap. Accessed: July 25, 2020.

Tulyakov, Sergey, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz **(2018)**: "MoCoGAN: Decomposing Motion and Content for Video Generation". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 1526–1535.

Tulyakov, Stepan, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza **(2022)**: "Time Lens++: Event-based Frame Interpolation with Parametric Nonlinear Flow and Multi-scale Fusion". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 17734–17743.

Tulyakov, Stepan, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza **(2021)**: "Time Lens: Event-Based Video Frame Interpolation". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 16155–16164.

Unterthiner, Thomas, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly **(2018)**: "Towards Accurate Generative Models of Video: A New Metric & Challenges". In: *CoRR* abs/1812.01717. arXiv: 1812.01717.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin **(2017)**: "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 5998–6008.

Verdoliva, Luisa **(2020)**: "Media Forensics and DeepFakes: An Overview". In: *IEEE J. Sel. Top. Signal Process.* 14.5, pp. 910–932.

Villegas, Ruben, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee **(2017)**: "Decomposing Motion and Content for Natural Video Sequence Prediction". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Wang, Bishan, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang **(2020)**: "Event Enhanced High-Quality Image Recovery". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIII*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12358. Lecture Notes in Computer Science. Springer, pp. 155–171.

Wang, Junke, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yu-Gang Jiang, and Ser-Nam Lim **(2022)**: "M2TR: Multi-modal Multi-scale Transformers for Deepfake Detection". In: *ICMR '22: International Conference on Multimedia Retrieval, Newark, NJ, USA, June 27 - 30, 2022*. Ed. by Vincent Oria, Maria Luisa Sapino, Shin'ichi Satoh, Brigitte Kerhervé, Wen-Huang Cheng, Ichiro Ide, and Vivek K. Singh. ACM, pp. 615–623.

Wang, Lin, S. Mohammad Mostafavi I., Yo-Sung Ho, and Kuk-Jin Yoon **(2019)**: "Event-Based High Dynamic Range Image and Very High Frame Rate Video Generation Using Conditional Generative Adversarial Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 10081–10090.

Wang, Sheng-Yu, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros **(2019)**: "Detecting Photoshopped Faces by Scripting Photoshop". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, pp. 10071–10080.

— **(2020)**: "CNN-Generated Images Are Surprisingly Easy to Spot... for Now". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 8692–8701.

Wang, Yaohui, Piotr Bilinski, François Brémond, and Antitza Dantcheva **(2020)**: "G3AN: Disentangling Appearance and Motion for Video Generation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 5263–5272.

Wang, Zihao W., Weixin Jiang, Kuan He, Boxin Shi, Aggelos K. Katsaggelos, and Oliver Cossairt **(2019)**: "Event-Driven Video Frame Synthesis". In: *2019 IEEE/CVF International Conference on Computer Vision*

Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019. IEEE, pp. 4320–4329.

Wang, Ziwei, Yonhon Ng, Cedric Scheerlinck, and Robert E. Mahony **(2021)**: "An Asynchronous Kalman Filter for Hybrid Event Cameras". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 438–447.

Weissenborn, Dirk, Oscar Täckström, and Jakob Uszkoreit **(2020)**: "Scaling Autoregressive Video Models". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Weng, Wenming, Yueyi Zhang, and Zhiwei Xiong **(2021)**: "Event-based Video Reconstruction Using Transformer". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 2543–2552.

— **(2023)**: "Event-based Blurry Frame Interpolation under Blind Exposure". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 1588–1598.

Wiles, Olivia, A. Sophia Koepke, and Andrew Zisserman **(2018)**: "X2Face: A Network for Controlling Face Generation Using Images, Audio, and Pose Codes". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Vol. 11217. Lecture Notes in Computer Science. Springer, pp. 690–706.

Wu, Chenfei, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan **(2022)**: "NÜWA: Visual Synthesis Pre-training for Neural visUal World creAtion". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVI*. Ed. by Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13676. Lecture Notes in Computer Science. Springer, pp. 720–736.

Wu, Jiqing, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool **(2019)**: "Sliced Wasserstein Generative Models". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 3713–3722.

Wu, Song, Kaichao You, Weihua He, Chen Yang, Yang Tian, Yaoyuan Wang, Ziyang Zhang, and Jianxing Liao **(2022)**: "Video Interpolation by Event-Driven Anisotropic Adjustment of Optical Flow". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VII*. Ed. by Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Vol. 13667. Lecture Notes in Computer Science. Springer, pp. 267–283.

Wu, Wayne, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy **(2018)**: "ReenactGAN: Learning to Reenact Faces via Boundary Transfer". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Vol. 11205. Lecture Notes in Computer Science. Springer, pp. 622–638.

Xia, Weihao, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang **(2021)**: "GAN Inversion: A Survey". In: *CoRR* abs/2101.05278. arXiv: 2101.05278.

Xing, Zhening, Gereon Fox, Yanhong Zeng, Xingang Pan, Mohamed Elgharib, Christian Theobalt, and Kai Chen **(2024)**: "Live2Diff: Live Stream Translation via Uni-directional Attention in Video Diffusion Models". In: *CoRR* abs/2407.08701. arXiv: 2407.08701.

Xu, Fang, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu **(2021)**: "Motion Deblurring with Real Events". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 2563–2572.

Xu, Sicheng, Guojun Chen, Yuxiao Guo, Jiaolong Yang, Chong Li, Zhenyu Zang, Yizhong Zhang, Xin Tong, and Baining Guo **(2024)**: "VASA-1: Lifelike Audio-Driven Talking Faces Generated in Real Time". In: *CoRR* abs/2404.10667. arXiv: 2404.10667.

Yang, Yixin, Jinxiu Liang, Bohan Yu, Yan Chen, Jimmy S. Ren, and Boxin Shi **(2024)**: "Latency Correction for Event-Guided Deblurring and Frame Interpolation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, pp. 24977–24986.

Yang, Ziming, Jian Liang, Yuting Xu, Xiaoyu Zhang, and Ran He **(2023)**: "Masked Relation Learning for DeepFake Detection". In: *IEEE Trans. Inf. Forensics Secur.* 18, pp. 1696–1708.

Yao, Xu, Alasdair Newson, Yann Gousseau, and Pierre Hellier **(2021)**: "A Latent Transformer for Disentangled Face Editing in Images and Videos". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 13769–13778.

Ye, Shuquan, Chu Han, Jiaying Lin, Guoqiang Han, and Shengfeng He **(2020)**: "Coherence and Identity Learning for Arbitrary-length Face Video Generation". In: *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*. IEEE, pp. 915–922.

Yin, Qilin, Wei Lu, Bin Li, and Jiwu Huang **(2023)**: "Dynamic Difference Learning With Spatio-Temporal Correlation for Deepfake Video Detection". In: *IEEE Trans. Inf. Forensics Secur.* 18, pp. 4046–4058.

Yu, Fisher, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao **(2015)**: "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop". In: *CoRR* abs/1506.03365. arXiv: 1506.03365.

Yu, Sihyun, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin **(2022)**: "Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yu, Zhentao, Zixin Yin, Deyu Zhou, Duomin Wang, Finn Wong, and Baoyuan Wang **(2023)**: "Talking Head Generation with Probabilistic Audio-to-Visual Diffusion Priors". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 7611–7621.

Yu, Zhiyang, Yu Zhang, Deyuan Liu, Dongqing Zou, Xijun Chen, Yebin Liu, and Jimmy S. Ren **(2021)**: "Training Weakly Supervised Video Frame Interpolation with Events". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 14569–14578.

Yushchenko, Vladyslav, Nikita Araslanov, and Stefan Roth **(2019)**: "Markov Decision Process for Video Generation". In: *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*. IEEE, pp. 1523–1532.

Zakharov, Egor, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor S. Lempitsky **(2020)**: "Fast Bi-Layer Neural Synthesis of One-Shot Realistic Head Avatars". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12357. Lecture Notes in Computer Science. Springer, pp. 524–540.

Zhang, Congyi, Mohamed Elgharib, Gereon Fox, Min Gu, Christian Theobalt, and Wenping Wang **(2022)**: "An Implicit Parametric Morphable Dental Model". In: *ACM Trans. Graph.* 41.6, 217:1–217:13.

Zhang, Hongguang, Limeng Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz **(2023)**: "Event-guided Multi-patch Network with Self-supervision for Non-uniform Motion Deblurring". In: *Int. J. Comput. Vis.* 131.2, pp. 453–470.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala **(2023)**: "Adding Conditional Control to Text-to-Image Diffusion Models". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 3813–3824.

Zhang, Xiang and Lei Yu **(2022)**: "Unifying Motion Deblurring and Frame Interpolation with Events". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 17744–17753.

Zhang, Xiang, Lei Yu, Wen Yang, Jianzhuang Liu, and Gui-Song Xia **(2023)**: "Generalizing Event-Based Motion Deblurring in Real-World Scenarios". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 10700–10710.

Zhang, Zelin, Anthony J. Yezzi, and Guillermo Gallego **(2023)**: "Formulating Event-Based Image Reconstruction as a Linear Inverse Problem With Deep Regularization Using Optical Flow". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.7, pp. 8372–8389.

Zheng, Xu, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang **(2023)**: "Deep Learning for Event-based Vision: A Comprehensive Survey and Benchmarks". In: *CoRR* abs/2302.08890. arXiv: 2302.08890.

Zhou, Peng, Xintong Han, Vlad I. Morariu, and Larry S. Davis **(2017)**: "Two-Stream Neural Networks for Tampered Face Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 1831–1839.

— **(2018)**: "Learning Rich Features for Image Manipulation Detection". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 1053–1061.

Zhu, Jiapeng, Yujun Shen, Deli Zhao, and Bolei Zhou **(2020)**: "In-Domain GAN Inversion for Real Image Editing". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12362. Lecture Notes in Computer Science. Springer, pp. 592–608.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros **(2017)**: "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 2242–2251.

Zhu, Lin, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian **(2022)**: "Event-based Video Reconstruction via Potential-assisted Spiking Neural Network". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 3584–3594.

Zollhöfer, Michael, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt **(2018)**: "State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications". In: *Comput. Graph. Forum* 37.2, pp. 523–550.

Zou, Yunhao, Yinqiang Zheng, Tsuyoshi Takatani, and Ying Fu **(2021)**: "Learning To Reconstruct High Speed and High Dynamic Range

Videos From Events". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 2024–2033.

Alphabetical Index