D5.2

Initial report on Multimedia Production

Project Title	AI4Media – A European Excellence Centre for Media, Society and Democracy
Contract No.	951911
Instrument	Research and Innovation Action
Thematic Priority	H2020-EU.2.1.1. - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (ICT) / ICT-48-2020 - Towards a vibrant European network of AI excellence centres
Start of Project	1 September 2020
Duration	48 months

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911

info@ai4media.eu www.ai4media.eu

Deliverable title	Initial report on Multimedia Production
Deliverable number	D5.2
Deliverable version	1.0
Previous version(s)	-
Contractual date of delivery	February 28, 2022
Actual date of delivery	March 10, 2022
Deliverable filename	AI4Media_D5.2.pdf
Nature of deliverable	Report
Dissemination level	Public
Number of pages	76
Work Package	WP5
Task(s)	T5.2
Parner responsible	UNIFI
Author(s)	Marco Bertini, Alberto Del Bimbo, Lorenzo Seidenari (UNIFI), Ioannis Mademlis (AUTH), Remy Mignot, Lenny Renault (IRCAM), Daniele Gravina (MODL), Elisa Ricci, Enver Sangineto, Nicu Sebe, Kasim Sinan Yildirim (UNITN), Lucille Sassatelli (UCA-3IA), David Melhart (UM)
Editor	Lorenzo Seidenari (UNIFI)
Officer	Evangelia Markidou

Abstract	This deliverable presents the initial outcomes of AI4Media research activities in the context of Task 5.2 “Media content production”. The document presents research advances on the topics of image and video quality enhancement, GAN-based generation of human parts and body scenes, playable video generation, procedural terrain generation, automated cinematography, interactive content improvement for gaming, generation of synthetic musical sound mixes, and enhancement of 360°videos using user attention prediction. We also present relevant publications, links to software and relevance with AI4media use case requirements. Finally, we discuss the plans for ongoing activities and future research.
Keywords	Multimedia, content production, content enhancement, procedural content generation, playable video, procedural terrain, automated cinematography, UAVs, user attention prediction, 360°video, player modelling, piano sound synthesis

Copyright

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911

www.ai4media.eu

info@ai4media.eu

Contributors

NAME	ORGANIZATION
Lorenzo Seidenari	UNIFI
Marco Bertini	UNIFI
Alberto Del Bimbo	UNIFI
Elisa Ricci	UNITN
Enver Sangineto	UNITN
Nicu Sebe	UNITN
Kasim Sinan Yildirim	UNITN
Sotirios Papadopoulos	AUTH
Ioannis Mademlis	AUTH
Lucille Sassatelli	UCA-3IA
David Melhart	UM
Daniele Gravina	MODL
Remy Mignot	IRCAM
Lenny Renault	IRCAM

Peer Reviews

NAME	ORGANIZATION
Antonios Liapis	UM
Mike Matton	VRT

Revision History

Version	Date	Reviewer	Modifications
0.1	09/02/2022	Lorenzo Seidenari (UNIFI)	Initial version, Table of contents and sections
0.2	03/03/2022	Lorenzo Seidenari (UNIFI)	Pre-final version ready for internal review
0.3	09/03/2022	Antonios Liapis (UM), Mike Matton (VRT)	Internal review
1.0	10/03/2022	Lorenzo Seidenari (UNIFI)	Final version ready for submission

Table of Abbreviations and Acronyms

Abbreviation	Meaning
2D	Two-Dimensional
3D	Three-Dimensional
ACC	Accuracy
ADD	Average Detection Distance
AI	Artificial Intelligence
AMT	Amazon Mechanical Turk
API	Application Programming Interface
C2GAN	Cycle In Cycle Generative Adversarial Network
CADDY	Clusteringfor Action Decomposition and DiscoverY
CART	Classification And Regression Tree
CMT	Camera Motion Type
CNN	Convolutional Neural Network
CRF	Constant Rate Factor
DDPG	Deep Deterministic Policy Gradient
DDSP	Differentiable Digital Signal Processing
DNN	Deep Neural Network
DoA	Description of Action
DoF	Degrees of Freedom
DQN	Deep Q Learning
DRL	Deep Reinforcement Learning
ESPCN	Efficient Sub-Pixel Convolutional Neural Network
FID	Frechét Inception Distance
FOV	Field of View
FST	Framing Shot Type
FVD	Frechét Video Distance
G2GAN	Geometry-Guided Generative Adversarial Network
G2R	fraction of Generated images identified as Real
GAN	Generative Adversarial Network
GC	Guidance Cycle-consistency loss
GRU	Gated Recurrent Unit
GT	Ground Truth
HD	High Definition
HSV	Hue Saturation Value
IC	Image Cycle-consistency loss
InfoGAN	Information Maximizing GAN
IQA	Image Quality Assessment
IS	Inception Score

Abbreviation	Meaning
ISF	implicit style function
LiDAR	Light Detection and Ranging
LIQA	Language based Image Quality Assessment
LPIPS	Learned Perceptual Image Patch Similarity
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MDR	Missing Detection Rate
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
ML	Machine Learning
MoCoGAN	Motion and Content decomposed Generative Adversarial Network
MOS	Mean Opinion Score
MPC	Model Predictive Control
MSE	Mean Squared Error
NHI	Normalized Histogram Intersection
PCG	Procedural Content Generation
PDF	Probability Density Function
Pix2Pix	Pixel to Pixel
PL	Preference Learning
PM	Player Modelling
PoI	Point of Interest
POMDP	Partially Observable Markov Decision Process
PSNR	Peak Signal-to-Noise Ratio
PVG	Playable Video Generation
R-CNN	Region-based Convolutional Neural Network
R2G	fraction of Real images identified as Generated
RAI	RAdiotelevisione Italiana
RF	Random Forest
RGB	Red Green Blue
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RoI	Region of Interest
SAVP	Stochastic Adversarial Video Prediction
SDT	Self Determination Theory
SR	Super Resolution
SRVP	Stochastic Latent Residual Video Prediction
SSAA	Super Sampling Anti-Aliasing
SSIM	Structural Similarity Index Measure
SSQP	Structural and Statistical Quality Predictor
TD3	Twin Delayed Deep Deterministic Policy Gradient

Abbreviation	Meaning
UAV	Unmanned Aerial Vehicle
UPEQ	Ubisoft Perceived Experience Questionnaire
VMAF	Video Multi-Method Assessment Fusion
VOD	Video on Demand
VR	Virtual Reality
VRNN	Variational Recurrent Neural Network
WP	Work Package

Contents

List of Tables

List of Figures

1. Executive Summary

This deliverable presents the research outcomes of Task 5.2 (Media Content Production) during the first 18 months of the AI4Media project. We present in detail the motivation, the developed methods, and the obtained results, making explicit references to publications authored by partners, software developed by them and relevant contributions to the WP8 use cases.

Research activities in T5.2 cover a wide range of topics relevant to multimedia content production, including content enhancement techniques for images and video, generation of playable video, automated cinematography, generation of synthetic musical mixes, and more. The results of these research activities have been successfully published in top journals and conferences.

More specifically, the following research outcomes have been produced, roughly classified in three categories, i.e. content generation, content enhancement and content generation automation:

- **Content generation**: Several methodologies and tools have been developed for content generation, providing solutions for creating new content in several domains with a wide range of applications such as 3D terrain generation, conditional video and audio generation. Ongoing work includes: (i) a novel attention-guided discriminator which only considers attended regions for guided image generation; (ii) the extension of an audio synthesis model allowing to generate realistic symbolic piano performances without aligned music scores; (iii) a GAN extension based on an implicit style function that enables cost-effective multi-modal unsupervised image-to-image translations.

- **Content enhancement**: We also deal with content enhancement techniques to improve content delivery and user experience, specifically for standard and 360°video streams and gaming. The tools developed provide solutions for accurately predicting user attention in 360°video so to better allocate bandwidth selectively, for improving compressed video with a fast network able to run also on low-end devices, for evaluating the quality of restored media and for modelling the video game player experience. Ongoing work includes: (i) the application of memory based multimodal trajectory prediction networks to head motion prediction; and (ii) the extension of video enhancement approaches to include the capability to incorporate sporadic high quality content.

- **Automation of content generation process**: Finally, we propose new methods for the full automation of UAVs use in the footage shooting process. The tools developed provide solutions that allow to learn agents able to plan valid combinations of Camera Motion Types and Framing Shot Types. Among ongoing work, we have the extension of the developed Deep Reinforcement Learning methods to all target-tracking Camera Motion Types.

The goal in the near future is to extend these research outcomes, integrate them in WP8 use case demonstrators where possible, and also foster joint research that can result in developments that are mutually beneficial to the cooperating partners.

2. Introduction

Task 5.2 “Media content production” of AI4Media investigates multiple aspects of (semi-)automatic media content production, focusing on the creation, adaptation and enhancement of media content. The task examines both the pure synthesis of media content exploiting computational methods such as Deep Generative Models as well as methodologies that help in the acquisition and streaming of such content to end-user devices.

According to the AI4Media Description of Action (DoA), research activities in T5.2 cover a wide range of topics relevant to content production, including: cinematography planning, with emphasis on UAV media production; novel enhancement techniques for videos beyond the visible domain, exploiting multimodality; procedural content generation; sound synthesis of musical instruments based on synthetic music sounds; adaptation of AR/VR content to specific users and environments aiming to address the problem of efficiently streaming such heavy contents.

During the first 18 months of the project, research focused on the following aspects of multimedia content production:

- **Novel content generation, specifically in the 3D, video and audio domain.** In section 3.4, a novel terrain generation algorithm, based on GANs has been proposed. Generating realistic and varied 3D meshes of terrain is critical for VR applications such as games and simulations. In sections 3.3 and 3.2, two generative approaches have been proposed, able to create new content from conditioning variables. In section 3.3, video generation is performed in a controlled manner. More specifically, by learning a set of distinct actions it is possible to offer users the possibility to interactively generate new videos. In section 3.2, two generators are learned: an image generator in the pixel domain and a guidance generator in a possibly different modality used for guidance. This way it is possible, for example, to better generate faces, by conditioning with landmark positions. Finally, in section 3.8, an audio generation approach is proposed. Audio generation is a key problem in game, documentary and movie production. The proposed approach deals with piano sound synthesis from a symbolic representation handling polyphonic MIDI and also modelling particular properties of the piano sound, such as partials inharmonicity, partials beating, and noise of the hammers, the keys and the pedals.

- **Content enhancement and delivery.** A key aspect of content delivery is the possible degradation that happens caused by compression. Heavy content such as video is often compressed. This is especially true when dealing with 360°videos, for example. In section 3.1, a novel super-resolution and artefact removal network is proposed, specifically designed to be efficient in order to run on end-user devices. One key aspect of this work is also the capability of such an architecture to be fine-tuned on specific content so to further improve the per-video enhancement. When it comes to serving 360°videos we have a different challenge: codecs must attempt to predict user attention and exploit this information to selectively code video. To decrease the bandwidth usage, it is common to stream in high resolution only the portion of the sphere in the user Field of View. In section 3.6, a new Deep Learning based architecture is proposed, yielding state-of-the art results in head motion prediction in 360°videos. Finally, when dealing with content enhancement it is also important to design novel methods to evaluate content quality. Currently, approaches are either tuned on specific datasets or distortions, or fail to correctly score high quality images derived from GAN based enhancers. In section 3.1.2, we present a work that exploits a multimodal task (image captioning) to estimate image quality. The proposed method is extremely simple but powerful and versatile, showing many advantages over other image similarity metrics. Finally, in section 3.7, a set of ML techniques to model player experience in videogames is presented. The main idea is to identify player populations so to understand and enhance different aspects of player experience. This is a prerequisite for the design of entirely new and engaging gameplay via game content generation.

- **Automated Cinematography.** While many approaches in T5.2 deal with the direct generation of content exploiting Deep Generative Models, we must keep in mind that the main source of high-resolution high-quality content is provided by shooting footage directly. To lower production costs and to allow filming crews to deploy UAVs easier, automatic cinematography methods are sought in section 3.5. We propose to adapt a Deep Deterministic Policy Gradient (DDPG) agent training a task-specific reward function; moreover, a novel on-line distillation policy is developed from a Model Predictive Control teacher network.

All these works are described in section 3. For each one of them, we provide links to relevant publications and open software developed by the AI4Media partners, while also explaining how they connect with AI4Media uses cases and specific user requirements. Section 4 briefly summarises the AI4Media use cases and the corresponding requirements (Epics) that the techniques described in this deliverable contribute to. Finally, section 5 concludes the deliverable, describing also plans for future extensions and developments in T5.2 by the different contributing partners.

3. Content Production(Task 5.2)

Contributing partners: UNIFI, UNITN, AUTH, 3IA-UCA, MODL, IRCAM

In this section, we present the research outcomes of T5.2 during the first 18 months of the project, addressing multiple aspects of (semi-)automatic media content production, including novel content generation, specifically in the 3D, video and audio domain, content enhancement and delivery, and automated cinematography. Most of the presented works have already been published in journals or conference proceedings, contributing to the advancement of the state-of-the-art in their specific domains. Relevant papers, open software and connection with AI4Media use cases are also presented.

3.1. Media Quality Enhancement and Assessment

Contributing partners: UNIFI

In this section, we report two contributions on image quality enhancement and image quality assessment developed by UNIFI. The former [5] allows to perform super-resolution and compression artefact removal with real-time performance. The latter [6] addresses the issue of obtaining a reliable and general image quality estimator, also for images enhanced via GANs.

3.1.1. Fast Video Visual Quality and Resolution Improvement using SR-UNet

Video streaming has become the major source of Internet traffic in the last years, over desktop and mobile platforms, either for work or entertainment. Videoconferencing has become an important form of communication, especially after the emergence of the COVID-19 pandemic, and video on demand (VOD) streaming services like Netflix, Amazon Prime Video and Disney+ provide an alternative to cable or satellite TV, offering movies and shows along with broadcasters that offer their live programmes through streaming apps. All of these applications require video compression algorithms like H.264, or the more recent H.265, to optimize the available bandwidth and reduce transmission costs. However, these compression algorithms are usually lossy and they introduce visual artifacts like blocking, mosquito noise, posterization, etc. that may hinder the user experience.

In this work, we propose to use a novel neural network, called SR-UNet, to improve the visual quality of the decoded video on the device of the end user in real-time, without requiring any change in the video compression and delivery pipeline. This network is designed to improve the visual quality while reducing the bandwidth required to stream a video; it does so by performing both super resolution, i.e. reconstructing high resolution frames from a low resolution stream, and reducing video compression artifacts. Considering the visual quality improved by these operations, in order to reduce bandwidth consumption, videos may be streamed at lower resolution or with a higher compression factor.

The main contributions of this work are:

- a novel network that extends the UNet architecture by reducing its size and computational cost;
- a loss that combines signal-based and perceptual-based losses within a Generative Adversarial Network (GAN) framework.
- Furthermore, we show how the proposed network can be tailored on specific video clips, to further improve the performance of the base SR-UNet model.

This operating context is particularly relevant for VOD services. They commonly aim to reduce the bandwidth required to transmit videos, for instance, by looking for the best encoding parameters of each video. Extensive experiments on a standard video dataset encoded with H.265 show that the proposed network outperforms baselines and other state-of-the-art approaches; objective and subjective evaluations show that the network is able to improve the visual appearance of videos at different compression rates.

Our test set is composed from clips downloaded from 14 non-compressed clips of the *Derf's Collection* [7], that is commonly used to evaluate video coding and streaming [8], super resolution [9], compression artifact reduction [10] and visual quality improvement tasks [11]. The clips were compressed from 1080p to 540p with the H.265 codec with CRF 23, preparing an analogue setting as the train set. The clips are: *Ducks take-off*, *Crowd run*, *Controlled burn*, *Aspen*, *Snow mountain*, *Touchdown pass*, *Station 2*, *Rush hour*, *Blue sky*, *Riverbed*, *Old town cross*, *Rush field*, *Into tree* and *Sun flower*.

In the first experiment, we compare our SR-UNet with a UNet baseline to assess the performance of our proposed changes and, with a H.265+bicubic interpolation to assess the improvement of the methods. We compare our method also with two other competing approaches. In particular, both the base UNet, and a 6-layer ESPCN [12] network implement Rep-VGG residual layers calibrated for processing at the same frame-rate; the last competing state-of-the-art approach is an 8-layer SR-ResNet [13], which is much slower than the other architectures. All the models have been trained with the same methodology; frames are rescaled from 540p to 1080p (Full-HD).

Table 2 reports the results in terms of Structural Similarity Index Measure (SSIM), Learned Perceptual Image Patch Similarity (LPIPS) and Video Multi-Method Assessment Fusion (VMAF), reporting also the frames per second processed by each method, as obtained on a NVIDIA GTX 1080Ti. We notice that our architecture largely improves the perceptual metric (LPIPS) over H.265, thanks to the compression artifacts reduction and to the increase of frequencies in the high-frequency spectrum, and the quality improvement is also notable by the increase in VMAF. The SSIM metric, although being more perception-oriented than PSNR (Peak Signal-to-Noise Ratio), is still based on the original signal, thus it is somehow predictable how its score is reduced by the adversarial and perceptual-driven training, as reported also in [14]. SR-UNet obtains a large speed-up over the SR-ResNet while maintaining the same quality. This is obtained since our model optimizes residual layers, compresses the up-scaling layer, and removes non-useful batch-normalizations, exploiting the particular U-Net architecture.

Table 2. Comparison between models performances. ↑ indicates that higher values are better, ↓ indicates that lower values are better. Best results are highlighted in ***bold****, second best are underlined.*

Architecture	SSIM ↑	LPIPS ↓	VMAF ↑	FPS ↑
SR-UNet (ours)	0.7190	**0.2067**	84.30	**46.1**
UNet	0.7273	0.2193	**85.32**	45.4
SR-ResNet-8 [13]	**0.7278**	0.2130	84.24	6.4
ESPCN-6 [12]	0.7159	0.2125	82.29	45.0
H.265 + bicubic	0.7209	0.2821	79.15	-

In the second experiment, we evaluate rate/distortion at different CRF values, comparing the results of using our SR-Unet to scale from a 720p (HD) source to 1080p (Full-HD), with that of the source 720p resolution and the target 1080p resolution. In this case, the source 720p is upscaled by SR-UNet to 1440p, then bicubic sampling is used to downscale to 1080p; this approach is similar in spirit to that of supersampling anti-aliasing (SSAA), used in computer graphics to improve the visual quality of renderings. To further reduce the size and computational cost of the network, we reduce filters and layers by $1/4$, resulting in a network size of only 1.1MB. Fig. 1 reports the distortion in terms of VMAF, while Fig. 2 reports distortion in terms of LPIPS. Table 3 reports a selection of visual quality metrics and bitrate for some CRF values. Observing the curves in Fig. 1 and 2 shows that using SR-UNet results in a visual quality that is similar to, or better than, that of the 1080p resolution but at a much lower bitrate (20%-33% less bandwidth for the same quality). This is clearly visible in the table: the bitrate is the same of the 720p, since the network is applied as a filter before showing the frame to the user. There is no change in the video stream that is transmitted, but visual quality in terms of VMAF and LPIPS is typically better. The table shows, again, that SSIM score is penalized by the generative approach of our method.

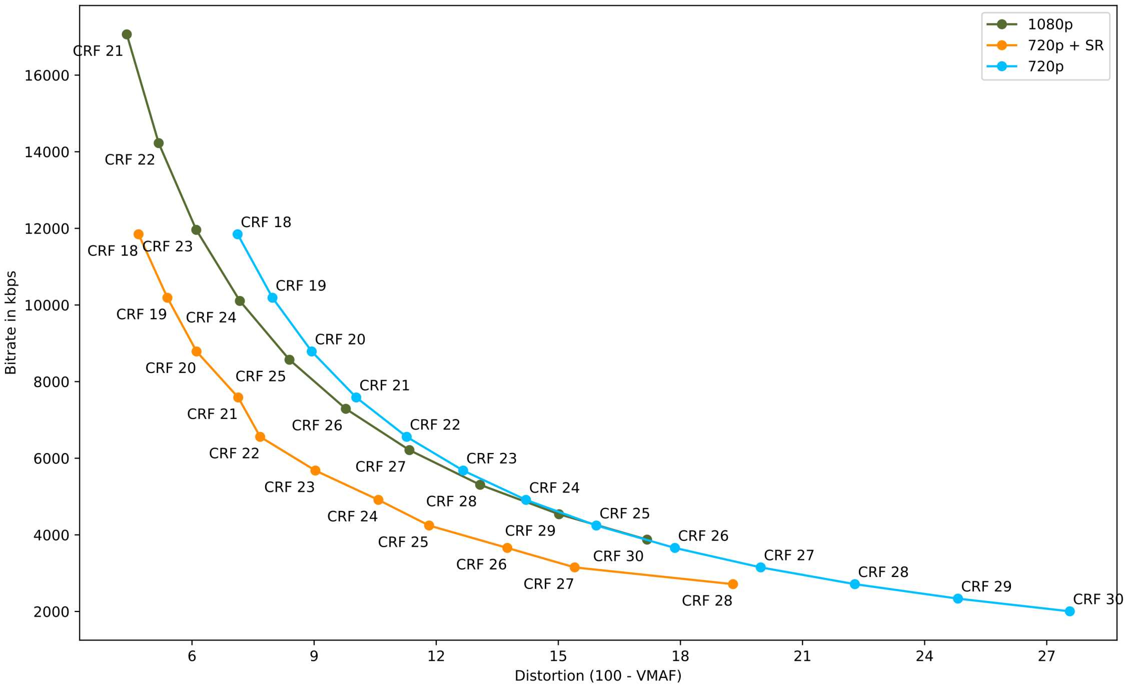


Figure 1. Rate/VMAF-distortion curve the varying of the compression CRF. We observe that after CRF 23-24, the 720p and 1080p curves merge. In this zone, the lower resolution of 720p is compensated from the lesser presence of compression artifacts, which instead may be visible on the original resolution.

Table 3. Comparison of encoding methods performances. Each triplet compares (from top to bottom): the super-resolved video metrics, a 1080p reference of similar quality and the low-quality video fed to the network. After enhancement, the video has similar perceptual quality to the higher resolution one, while preserving the bitrate for the low resolution.

Method	SSIM ↑	LPIPS ↓	VMAF ↑	Bitrate (*kb/s*) ↓
720p CRF 18 + SR-UNet	0.7611	0.1251	95.3133	**11,846**
1080p CRF 22	0.8181	0.1237	94.8185	14,225
720p CRF 18	0.7987	0.1835	92.8792	11,846
720p CRF 21 + SR-UNet	0.7711	0.1494	92.864	**7,585**
1080p CRF 24	0.8016	0.1440	92.824	10,105
720p CRF 21	0.7793	0.2024	89.964	7,585
720p CRF 23 + SR-UNet	0.7611	0.16790	90.9682	**5,678**
1080p CRF 26	0.7836	0.1634	90.2208	7,290
720p CRF 23	0.7639	0.2161	87.33	5,678
720p CRF 25 + SR-UNet	0.7402	0.1798	88.174	**4,243**
1080p CRF 28	0.7737	0.1834	86.920	5,306
720p CRF 25	0.7461	0.2312	84.065	4,243

3.1.2. Language Based Image Quality Assessment

In the last years, models able to generate novel images by implicit sampling from the data distribution have been proposed [15]. While these models are extremely appealing, generating for example photo realistic

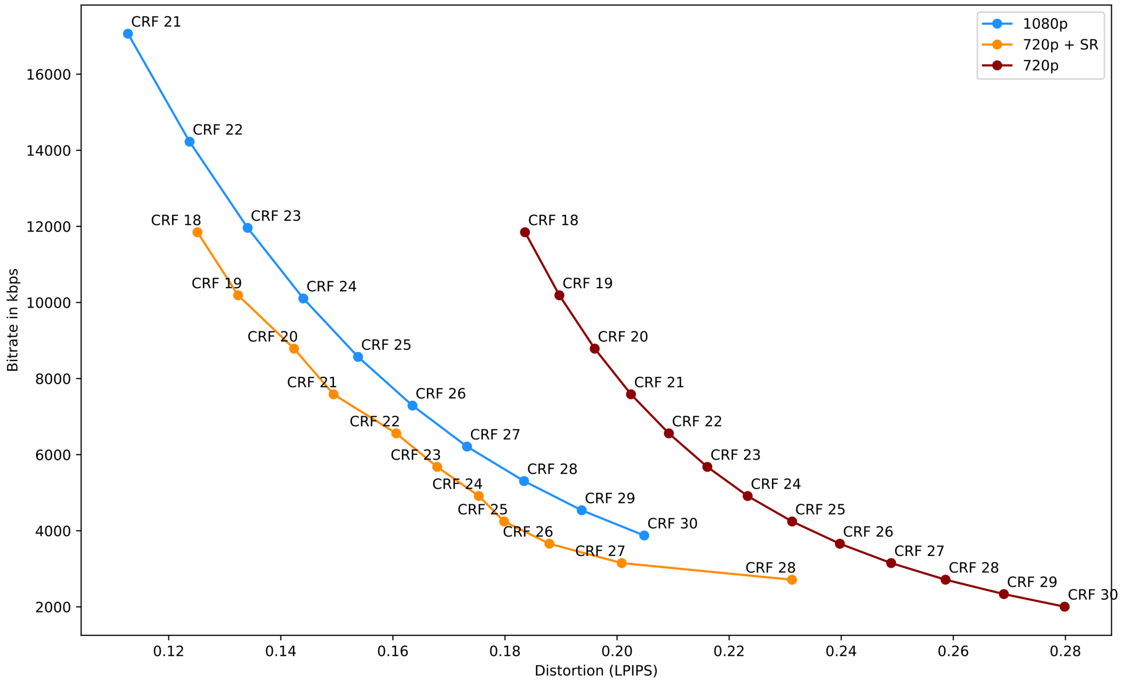


Figure 2. Rate/LPIPS-distortion curve at the varying of the compression CRF. The LPIPS distance, is way more sensitive to the higher frequencies sampled at higher resolution, however has a tendency to ignore some types of artifacts.

faces [16] or landscapes [17], they are hard to evaluate. Often anecdotal qualitative examples are presented to the reader with little quantitative and objective evidence, and evaluation of generative models is still undergoing a debate regarding how to perform it. The idea of using a computer vision classifier to evaluate the veracity of generated images was first proposed in [18]. The authors propose the Inception Score (IS), which is obtained by applying the Inception model [19] to every generated image in order to obtain the conditional label distribution $p(y|x)$. Realistic images should contain one or few well defined objects therefore leading to a low entropy in the conditional label distribution $p(y|x)$. An improved evaluation metric, named Frechét Inception Distance (FID), has been proposed by [20]. The authors show that FID is more consistent than Inception Score with increasing disturbances and human judgment. FID performs better as an evaluation metric since it also exploits the statistics of the real images.

Recently [21, 22, 14] have specifically addressed methods to evaluate GANs. In [21] two methods have been proposed that evaluate diversity and quality of generated images using classifiers trained and tested on generated images. In [23] an IQA model is trained with generated images. In [22] a discussion of 24 quantitative and 5 qualitative measures for evaluating generative models is provided, including IS and FID, image retrieval and classification performance. In [14] it is observed that many existing image quality algorithms do not assess correctly GAN generated content, especially when considering textured regions; this is due to the fact that although GANs generate very realistic images that may look like the original one, they match them poorly when considering pixel-based metrics. The proposed metric, called SSQP (Structural and Statistical Quality Predictor), is based on the "naturalness" of the image.

The main contributions of this work are the following:

- We propose an image quality assessment method based on language models, so called Language Based Image Quality Assessment (LIQA). To the best of our knowledge, language has never been used to evaluate the quality of images.

- Our evaluation protocol shows consistency across different captioning algorithms [24, 25] and language similarity metrics. Interestingly, improving the language generation model also improves the correlation between our score and MOS (Mean Opinion Score).
- Experiments show that our approach does not suffer from drawbacks of common full-reference and no-reference metrics when evaluating GAN enhanced images and keeps a high accordance with human scores for compressed images and for images restored via deep learning.

In Table 4 we report results using various captioning metrics. Interestingly all metrics show that captions over reconstructed images (REC rows) are better with respect to caption computed over compressed images (JPEG rows). This shows that image details that are compromised by the strong compression induce errors in the captioning algorithm. On the other hand, the GAN approach is able to recover an image which is not only pleasant to the human eye but recovers details which are also semantically relevant to an algorithm.

Table 4. Evaluation of image restoration over compression artifacts using GAN and captioning as a semantic task (best results highlighted in bold). Captions created from reconstructed images obtain a better score for every metric.

QUALITY	BLEU_1↑	METEOR↑	ROUGE↑	CIDEr↑	SPICE↑
JPEG 10	0.589	0.173	0.427	0.496	0.103
REC 10	**0.730**	**0.253**	**0.527**	**1.032**	**0.189**
JPEG 20	0.709	0.241	0.513	0.937	0.174
REC 20	**0.751**	**0.266**	**0.543**	**1.105**	**0.201**
JPEG 30	0.740	0.258	0.535	1.054	0.194
REC 30	**0.757**	**0.269**	**0.549**	**1.133**	**0.205**
JPEG 40	0.748	0.263	0.542	1.087	0.200
REC 40	**0.758**	**0.270**	**0.549**	**1.132**	**0.206**
JPEG 60	0.755	0.267	0.546	1.117	0.204
REC 60	**0.760**	**0.270**	**0.550**	**1.137**	**0.207**
ORIGINAL	0.766	0.274	0.556	1.166	0.211

In this work, we propose a new idea to evaluate image enhancement methods. Existing metrics based on the comparison of the restored image with an undistorted version may give counter-intuitive results. On the other hand, the use of naturalness based scores may in certain cases rank restored images higher than original ones.

We have shown that instead of using signal based metrics, semantic computer vision tasks can be used to evaluate results of image enhancement methods. Our claim is that a fine grained semantic computer vision task can be a great proxy for human level image judgement.

We show that employing algorithms mapping input images to a finer output label space, such as captioning, leads to more discriminative metrics. Future work will regard the evaluation of captions provided by humans over compressed and restored images. Moreover, we will take into account the accuracy of captions as a further metric to optimize.

3.1.3. Relevant publications

- Vaccaro, F., Bertini, M., Uricchio, T., & Del Bimbo, A. (2021, October). Fast Video Visual Quality and Resolution Improvement using SR-UNet. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 1221-1229). [5].
  Zenodo record: https://zenodo.org/record/5545598#.YgOC7b9Khrk.
- Leonardo Galteri, Lorenzo Seidenari, Pietro Bongini, Marco Bertini, and Alberto Del Bimbo. 2021. Language Based Image Quality Assessment. In ACM Multimedia Asia (MMAsia '21). Association for Computing Machinery, New York, NY, USA, Article 25, 1–7. DOI: https://doi.org/10.1145/3469877.3490605[26].
  Zenodo record: https://zenodo.org/record/6334282#.YiXnor_MJrk.

3.1.4. Relevant software and/or external resources

The Pytorch implementation of our work "Fast Video Visual Quality and Resolution Improvement using SR-UNet" can be found at https://github.com/fede-vaccaro/fast-sr-unet.

3.1.5. Relevant WP8 Use Cases

3C2-9 Management of contribution under bandwidth constraints. Fast SR-UNET can be deployed for real-time image enhancement when bandwidth constraints limit video quality. 3B2-1 (Video super resolution). Our SR-UNET can provide super resolution also for archives. Our LIQA approach can help in evaluating the quality of restored content and to score existing content quality.

3.2. Generative Adversarial Networks for Generating Human Faces, Hands, Bodies, and Natural Scenes

Contributing partners: UNITN

In this work, we focus on how to generate a target image given an input image. This has many application scenarios such as human-computer interaction, entertainment, virtual reality, and data augmentation. However, this task is challenging since it needs a high-level semantic understanding of the image mapping between the input and the output domains. Recently, Generative Adversarial Networks (GANs) [15] have shown the potential to solve this challenging task. GANs have produced promising results in many tasks such as image generation [27, 28], image inpainting [29, 30], video captioning [31], and cross-modal retrieval/matching [32, 33].

Recent works have developed powerful image-to-image translation systems, e.g., Pix2pix [34], Pix2pixHD [35], and GauGAN [36] in supervised settings, and CycleGAN [37] and DualGAN [38] in unsupervised settings. However, these methods are tailored to merely two domains at a time and scaling them to more domains requires a quadratic number of models to be trained. For instance, with m different image domains, CycleGAN and Pix2pix need to train $m(m-1)/2$ and $m(m-1)$ models, respectively. To overcome this, Choi et al. propose StarGAN [39], in which a single generator/discriminator performs image-to-image translation for multiple domains. However, StarGAN is not effective in handling some specific image-to-image translation tasks such as human pose generation [1, 2], hand gesture generation [40], and cross-view image translation [41], in which image generation could involve infinite image domains since human body, hand gestures, and natural scenes in the wild can have arbitrary poses, sizes, appearances, locations, and viewpoints.

To address these limitations, many methods have been proposed to generate images based on an extra semantic guidance, such as object keypoints [42, 43, 44, 1], human skeletons [2, 40], or segmentation maps [41, 45, 46, 36, 47, 48]. For instance, Song et al. [44] propose a G2GAN framework for facial expression synthesis based on facial landmarks. Siarohin et al. [2] introduce a PoseGAN model for pose-based human image generation conditioned on human body skeletons. Regmi and Borji [41] propose both X-Fork and X-Seq for cross-view image translation conditioned on segmentation maps. However, the current state-of-the-art guided image-to-image translation methods such as PG2 [1], PoseGAN [2], X-Fork [41], and X-Seq [41] have two main issues: 1) they directly transfer a source image and the target guidance to the target domain (i.e., $[I_x, L_y] \overset{G_i}{\to} I_y^{'}$ in Fig. 3), without considering the mutual translation between each other, while the translation across different image and guidance modalities in an unified framework would bring rich cross-modal information; 2) they simply employ the guidance data as input to guide the generation process, without involving the generated guidance as supervisory signals to further improve the network optimization. Both issues lead to unsatisfactory results.

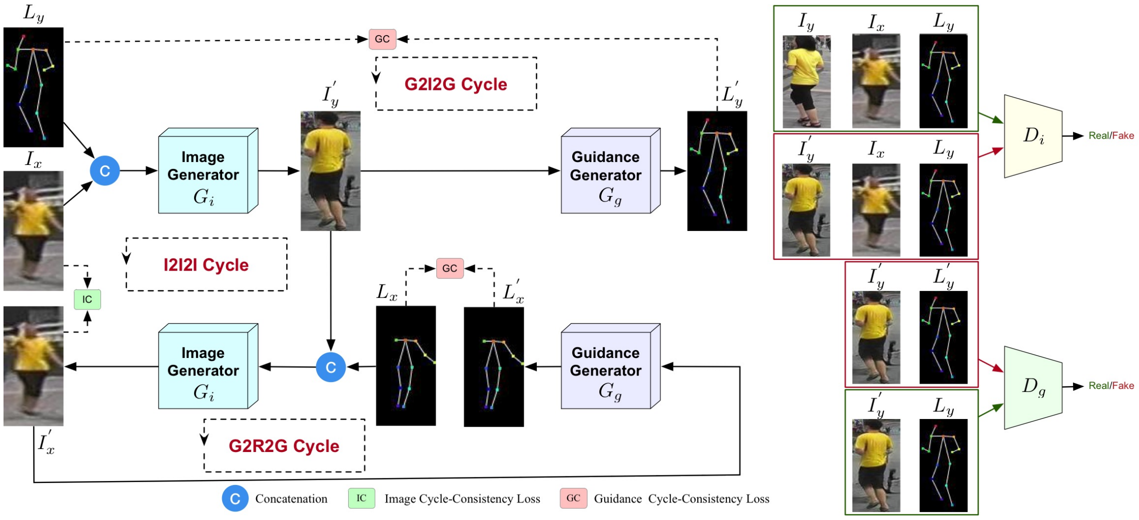


Figure 3. Overview of the proposed C2GAN, which consists of two types of generators, i.e., image generator G_i and guidance generator G_g. Parameter-sharing strategies can be used in between the image or the guidance generators to reduce the model capacity. During the training stage, two generators G_i and G_g are explicitly connected and trained by three cycles, i.e., the image cycle I2I2I: $[I_x, L_y] \overset{G_i}{\to} [I'_y, L_x] \overset{G_i}{\to} I'_x$ and two guidance cycles G2I2G: $[I_x, L_y] \overset{G_i}{\to} I'_y \overset{G_g}{\to} L'_y$, G2R2G: $[I'_y, L_x] \overset{G_i}{\to} I'_x \overset{G_g}{\to} L'_x$. The right side of the figure shows the cross-modal discriminators (i.e., D_i and D_g) for a better network optimization.

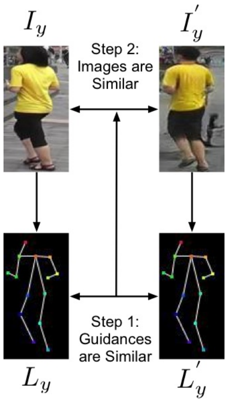


Figure 4. The motivation of the guidance cycle. If the generated guidance L'_y is close to the real guidance L_y, then the corresponding images (i.e., I'_y and I_y) should be similar.

To address these issues, we propose a novel and unified Cycle In Cycle Generative Adversarial Network (C2GAN), in which three cycled sub-nets are explicitly formed to learn both image and guidance modalities in a joint model. The framework of the proposed C2GAN is shown in Fig. 3. Specifically, to address the first limitation, C2GAN contains an image cycle, i.e., I2I2I ($[I_x, L_y] \overset{G_i}{\to} [I'_y, L_x] \overset{G_i}{\to} I'_x$), which aims at reconstructing the input and further refines the generated images I'_y. To address the second limitation, the guidance information (such as the human body skeleton) in C2GAN is not only utilized as input but also acts as output, meaning that the guidance is also a generative objective. The input and output of the guidance are connected by two novel guidance cycles, i.e., G2I2G ($[I_x, L_y] \overset{G_i}{\to} I'_y \overset{G_g}{\to} L'_y$) and G2R2G ($[I'_y, L_x] \overset{G_i}{\to} I'_x \overset{G_g}{\to} L'_x$), where G_i and G_g denote an image and a guidance generator, respectively. In this way, guidance cycles can provide weak supervision to the generated images I'_y. The intuition behind the guidance cycles

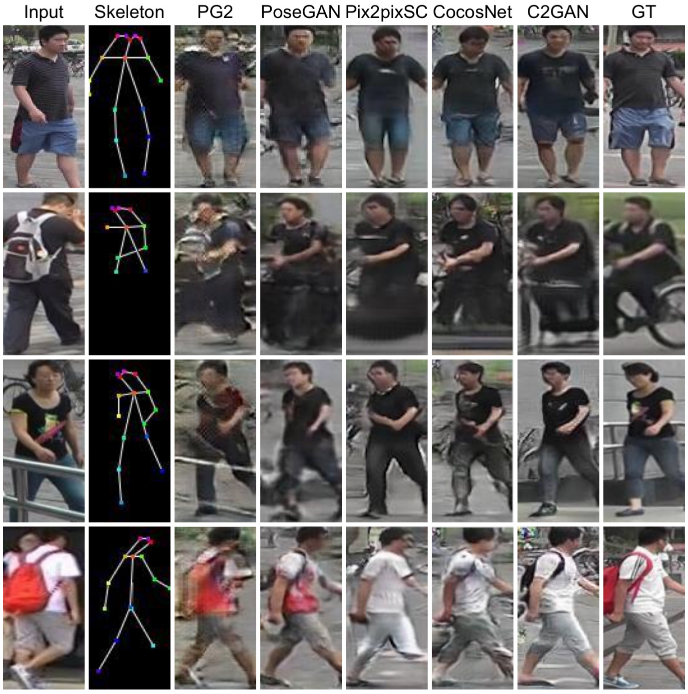


Figure 5. Qualitative comparison of person image generation on the Market-1501 dataset. From left to right: Input, Body Skeleton, PG2 [1], PoseGAN [2], Pix2pixSC [3], CocosNet [4], C2GAN (Ours), and Ground Truth (GT).

is that if the generated guidance is very close to the real guidance, then the corresponding images should be similar (see Fig. 4). In other words, a better guidance generation will boost the performance of image generation, and conversely the improved image generation will further facilitate the guidance generation. The proposed three cycles inherently constraint each other in an end-to-end training fashion. Moreover, for a better optimization of the proposed three cycles, we further propose two novel cycle losses, i.e., Image Cycle-consistency loss (IC) and Guidance Cycle-consistency loss (GC). With both cycle losses, each cycle can benefit from each other in a joint learning way. We also propose two cross-modal discriminators corresponding to the generators.

Our contributions can be summarized as follows:

- We propose C2GAN, a novel and unified cross-modal generative adversarial network for guided image-to-image translation tasks, which organizes the guidance and image data in an interactive manner, instead of using as input only the guidance information.

- The proposed cycle in cycle network structure is a new design which explores the effective use of cross-modal information for guided image-to-image translation tasks. The designed cycled subnetworks connect different modalities and implicitly constrain each other, leading to extra supervision signals for better image generation. We also investigate cross-modal discriminators and cycle losses for a more robust network optimization.

We conducted extensive experiments on generating a large variety of content including persons, facial expressions, hand gestures, and natural scenes. For brevity, here we present only the person image generation results and we refer the interested reader to [49].

For person image generation, we employed the Market-1501 dataset [50]. This dataset is a challenging person-reID dataset containing 32,668 images of 1,501 persons collected from six surveillance cameras.

Table 5. Quantitative comparison of person image generation on the Market-1501 dataset. For all the metrics, higher is better.

Model	AMT (R2G) ↑	AMT (G2R) ↑	SSIM ↑	IS ↑	Mask-SSIM ↑	Mask-IS ↑
PG2 [1]	11.2	5.5	0.253	3.460	0.792	3.435
DPIG [53]	-	-	0.099	**3.483**	0.614	3.491
PoseGAN [2]	22.7	**50.2**	**0.290**	3.185	0.805	3.502
Pix2pixSC [3]	18.6	41.5	0.275	3.141	0.790	3.468
CocosNet [4]	20.1	45.7	0.280	3.275	0.801	3.514
C2GAN (Ours)	**23.8**	47.3	0.285	3.362	**0.813**	**3.526**
Real Data	-	-	1.000	3.860	1.000	3.360

We adopted the training and testing splits used in [2] and obtained 263,631 and 12,000 pairs for the training and testing subset. We followed [2, 1] and adopted Inception Score (IS) [51], Structural Similarity (SSIM) [52], and their masked versions Mask-SSIM and Mask-IS as our evaluation metrics. Moreover, the AMT perceptual user study has been adopted to evaluate the generated images by different models.

We compared C2GAN with PG2 [1], DPIG [53], PoseGAN [2], Pix2pixSC [3], and CocosNet [4]. Different from these models which focus on person image generation, our method is a general framework and learns image and guidance generation simultaneously in a joint network. Quantitative results are shown in Table 5. C2GAN achieves better results than PG2, DPIG, Pix2pixSC, and CocosNet. Moreover, compared to PoseGAN [2], C2GAN yields better results on most metrics, i.e., AMT (R2G), IS, mask-SSIM, and mask-IS. Qualitative comparison results compared with PG2 and PoseGAN are shown in Fig. 5. C2GAN can generate more clear and visually plausible person images than both leading methods, validating the effectiveness of C2GAN. Moreover, our generated images are more similar to the ground truth.

3.2.1. Relevant publications

Hao Tang and Nicu Sebe, "Total Generate: Cycle in Cycle Generative Adversarial Networks for Generating Human Faces, Hands, Bodies, and Natural Scenes", IEEE Transactions on Multimedia, DOI: 10.1109/TMM.2021.3091847, 2022
https://ieeexplore.ieee.org/document/9464730 [49]

3.2.2. Relevant software and/or external resources

The Pytorch implementation of our work "Total Generate: Cycle in Cycle Generative Adversarial Networks for Generating Human Faces, Hands, Bodies, and Natural Scenes" can be found at https://github.com/Ha0Tang/C2GAN.

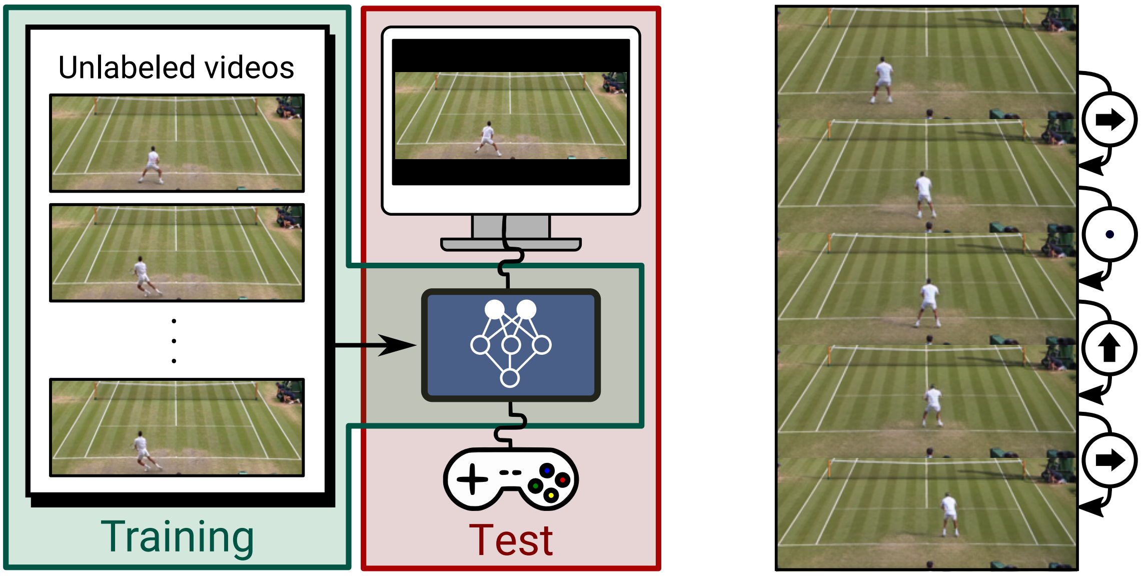
3.2.3. Relevant WP8 Use Cases

3C2-8 (Synthetic Video Generation from Single Semantic Label Map). C2GAN is a generic tool capable of generating photo-realistic images from a wide variety of domains with convincing details.

3.3. Playable Video Generation

Contributing partners: UNITN

Humans at a very early age can identify key objects and how each object can interact with its environment. This ability is particularly notable when watching videos of sports or games. In tennis and football,



(a) Playable Video Generation (b) Our results

Figure 6. We introduce the task of playable video generation in an unsupervised setting (left). Given a set of unlabeled video sequences, a set of discrete actions are learned in order to condition video generation. At test time, using our method, named CADDY, the user can control the generated video on-the-fly providing action labels.

for example, the skill is taken to the extreme. Spectators and sportscasters often argue which action or movement the player should have performed in the field. We can understand and anticipate actions in videos despite never being given an explicit list of plausible actions. We develop this skill in an unsupervised manner as we see actions live and on the screen. We can further analyze the technique with which an action is performed as well as the "amount" of action, i.e. how much to the left. Furthermore, we can reason about what happens if the player took a different action and how this would change the video.

From this observation, we propose a new task, Playable Video Generation (PVG) illustrated in Fig 6a. In PVG, the goal is to learn a set of distinct actions from real-world video clips in an unsupervised manner (green block) in order to offer the user the possibility to interactively generate new videos (red block). As shown in Fig 6b, at test time, the user provides a discrete action label at every time step and can see live its impact on the generated video, similarly to video games. Introducing this novel problem paves the way toward methods that can automatically simulate real-world environments and provide a gaming-like experience.

PVG is related to the future frame prediction problem [54, 55, 56, 57, 58, 59] and in particular to methods that condition future frames on action labels [60, 61, 62]. Given one or few initial frames and the labels of the performed actions, such systems aim at predicting what happens next in the video. For example, this framework can be used to imitate a desired video game using a neural network with a remarkable generation quality [60, 62]. However, at training time, these methods require videos with their corresponding frame-level action at every time step. Consequently, these methods are limited to video game environments [60, 62] or robotic data [61] and cannot be employed in real-world environments. As an alternative, the annotation effort required to address real-world videos can be reduced using a single action label to condition the whole video [63], but it limits interactivity since the user cannot control the generated video on-the-fly. Conversely, in PVG, the user can control the generation process by providing an action at every time-step after observing the last generated frame.

This research addresses these limitations introducing a novel framework for PVG named *Clustering for Action Decomposition and DiscoverY* (CADDY). Our approach discovers a set of distinct actions after watching multiple videos through a clustering procedure blended with the generation process that, at inference time, outputs playable videos. We adopt an encoder-decoder architecture where a discrete bottleneck layer is employed to obtain a discrete representation of the transitions between consecutive frames. A reconstruction loss is used as main driving loss, avoiding the need for neither action label supervision nor even the precise number of actions. A major difficulty in PVG is that discrete action labels cannot capture the stochasticity typical of real-world videos. To address this difficulty, we introduce an action network that estimates the action label posterior distribution by decomposing actions in a discrete label and a continuous component. While the discrete action label captures the main semantics of the performed action, the continuous component captures how the action is performed. At test time, the user provides only the discrete actions to interact with the generation process.

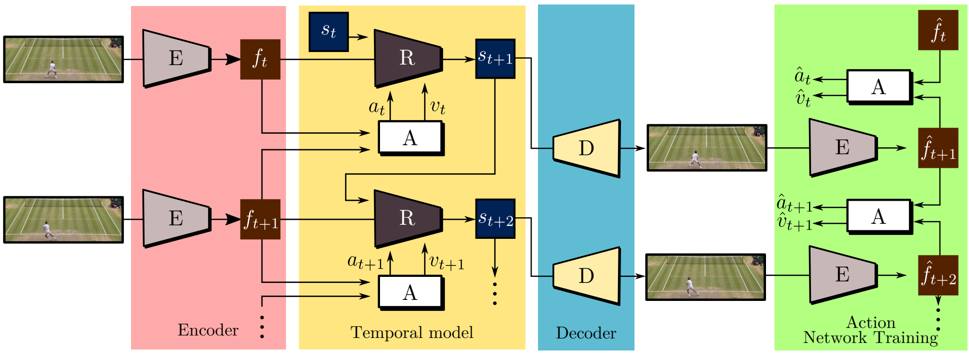


Figure 7. CADDY's training procedure for unsupervised playable video generation. An encoder E extracts frame representations from the input sequence. A temporal model estimates the successive states using a recurrent dynamics network R and an action network A which predicts the action label corresponding to the current action performed in the input sequence. Finally, a decoder D reconstructs the input frames. The model is trained using reconstruction as the main driving loss.

In our framework, a user can interactively control the video generation process by selecting an action at every time step among a set of K discrete actions. Our method is trained on a dataset of unannotated videos. We only assume that the video sequences depict a single agent acting in an environment. No action labels are required.

Inspired by Reinforcement Learning (RL) literature [64], the object in the scene is modeled as an agent interacting with its environment by performing an action at every time step. Differently from RL, our goal is to jointly learn the action space, the state representation that describes the state of the agent and its environment, and a decoder that reconstructs the observations (i.e., frames) from the state. CADDY is articulated into four main modules illustrated in Fig. 7: (i) an encoder employs a network E to extract frame representations; (ii) a temporal model estimates the label corresponding to the action performed in the current frame and predicts a state s_{t+1} that describes the environment at the next time step, after performing the detected action. The action label is predicted via a network A that receives the frame representations from the current and next frames. To predict the next frame environment state s_{t+1}, we employ a recurrent neural network R that we refer to as *Dynamics network*. (iii) a decoder module employs a network D to reconstruct each frame from the frame embedding predicted by the temporal model. (iv) the reconstructed frames are fed to the encoder to assess the quality of the estimated action labels. The overall pipeline is trained in an end-to-end fashion using as main driving loss a reconstruction loss on the output frames. The key idea of our approach is that the action network A needs to predict consistent action labels in order to correctly estimate the next frame embeddings s_{t+1}, and then, accurately reconstruct the input frames.

We evaluate our method on three video datasets: (1) *BAIR* robot pushing dataset [65]. We employ a version of the dataset in 256x256 resolution, composed of about 44K videos of 30 frames. Ground-truth robotic arm positions are available but are only used for evaluation purposes; (2) *Atari Breakout* dataset. We collect a dataset using a Rainbow DQN agent [66] trained on the Atari Breakout video game environment. We collect 1,407 sequences of about 32 frames with resolution 160x210 (358 for training, 546 sequences for validation and 503 for testing); (3) *Tennis* dataset. We collect Youtube videos corresponding to two tennis matches from which we extract about 900 videos with resolution 256x96. To respect the single agent assumption, we consider only the lower part of the field.

We propose to evaluate both action and video generation qualities by comparing the models on the video reconstruction task. A test sequence is considered and the action network is used to extract the sequence of learned discrete actions characterizing the input sequence. Starting from the initial frame, the extracted actions are used to reconstruct the remaining of the sequence. The evaluation is completed with a user-study that directly assesses the quality of the learned set of discrete actions. We adopt a large set of metrics.

Video quality metrics: (1) *LPIPS* [67]. We report the average *LPIPS* computed on corresponding frames of input and reconstructed sequences; (2) *FID* [68]. We report the average *FID* between the original and reconstructed frames; (3) *FVD* [69]. We compute the *FVD* between the original and reconstructed videos.

Action space metrics. We also introduce two metrics that measure the quality of the action space. Both metrics use additional knowledge (i.e., ground-truth information or an externally trained detector) to measure motion consistency among frames where the same action is performed. Assuming two consecutive frames, we measure the displacement Δ of a reference point on the object of interest. On the *Tennis* dataset, we employ FasterRCNN [70] to detect the player and use the bounding box center as the reference point. On *Atari Breakout*, we employ a simple pixel matching search to detect the rectangular platform and use its center as the reference point. Finally, on *BAIR*, we use the ground-truth location information of the robotic arm. The key idea of the two action quality metrics is to assess whether the predicted action labels and the displacement Δ are consistent by trying to predict one from the other: (1) Δ *Mean Squared Error* (Δ*-MSE)*. This metric measures how Δ can be regressed from the action label. For each action, we estimate the average displacement Δ, which is the optimal estimator for Δ (in terms of MSE). We report the MSE to evaluate regression quality. To facilitate comparison among datasets, we normalize the MSE by the variance of Δ over the dataset; (2) Δ*-based Action Accuracy* (Δ*-Acc)*. This metric measures how the predicted action can be predicted from the displacement Δ. To this aim, we train a linear classifier and report the action-accuracy measured on the test set.

Action-conditioning metrics. Finally, we consider two metrics that measure how the action label conditions the generated video. Therefore, it evaluates both generation quality and the learned action space: (1) *Average Detection Distance (ADD)*. Similarly to [71], we report the Euclidean distance between the reference object keypoints in the original and reconstructed frames. We employ the same detector as for the action-quality metrics to estimate the reference points. (2) *Missing Detection Rate (MDR)*. We report the percentage of detections that are successful in the input sequence, but not in the reconstructed one. This represents the proportion of frames where the object of interest is missing.

Since we present the first method for unsupervised PVG, we compare the proposed model against a selection of baselines, adapting existing methods to this new setting. We consider two criteria in the selection of the baselines: the architecture should be adaptable to the new setting without deep modifications, and the source code must be available to include these modifications. Comparing existing video prediction methods in terms of FVD shows that SAVP (Stochastic Adversarial Video Prediction) [55] is the best performing method with the code publicly available. Only the methods described in [72, 73, 74] are marginally outperforming SAVP but their code is not available.

Furthermore, SAVP [55] is widely used and features an architecture prone to be adapted to the current setting. During training, SAVP learns an encoder E that encodes information relative to the transition between successive frames which is used by the generator in the synthesis of the next frame. After training the SAVP model, we cluster the latent space learned by the encoder network on the training sequences using K-means. This procedure produces a set of K centroids that we use as action labels. During evaluation, we compute the latent representations for the input sequence and assign them the index of the nearest centroid. As a second baseline, we choose MoCoGAN [58]. This method possesses favorable characteristics for adaptation: it separates the content from the motion space and it employs an InfoGAN [75] loss to

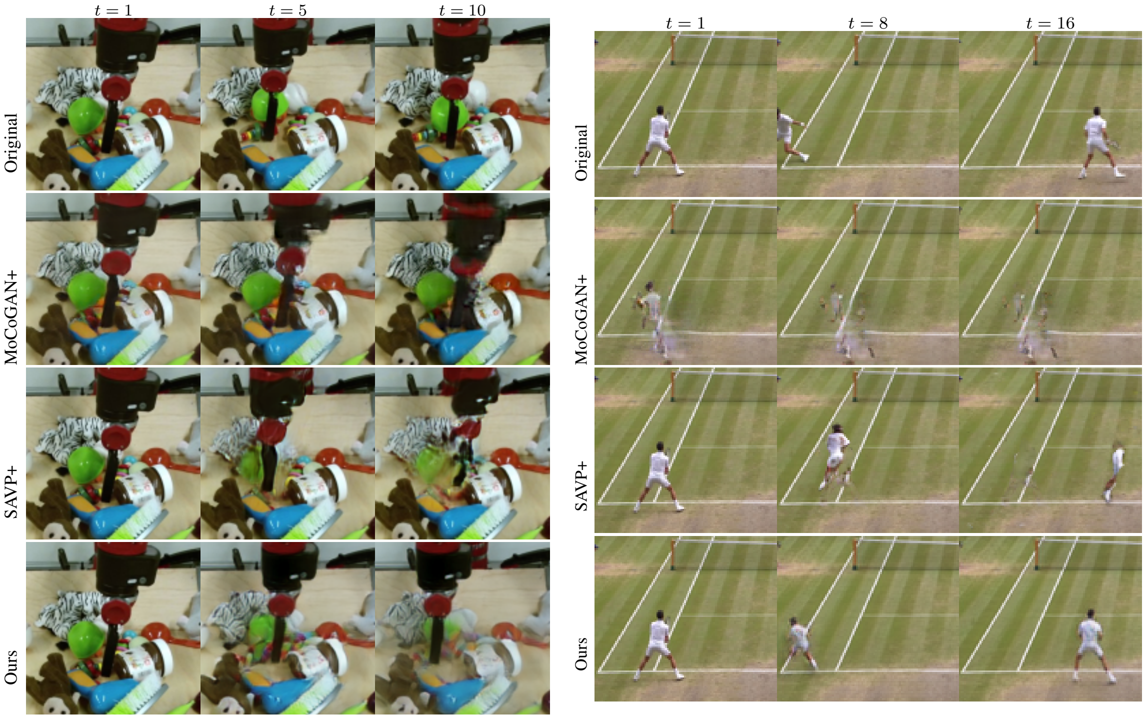


Figure 8. Reconstruction results on the BAIR *(left) and* Tennis *(right) datasets. We zoom in for better visualization.*

learn discrete video categories. We consider 6-frame short videos and assume a constant action over them. We employ an action discriminator that predicts the action performed in the input video sequence. This predicted action is used as an auxiliary input to the generator and the InfoGAN loss is used to learn the action space. Finally, we also include SRVP [54] in our comparison. SRVP is modified similarly to SAVP [55] to handle PVG. However, due to the poor empirical results and the high training costs, we consider this baseline only on the *BAIR* dataset. Several other works have been considered, but not adopted [76, 77] since they would require too important modifications to handle the PVG problem.

SAVP [55], MoCoGAN [58] and SRVP [54] are designed for video generation at resolutions lower than our datasets. Therefore, we generate videos at the resolution of 64x80, 64x64 and 128x48 respectively for the *Atari Breakout*, *BAIR* and *Tennis* datasets, and then upscale to full resolution for evaluation. In addition, we create two other baselines, referred to as SAVP+ and MoCoGAN+, obtained by increasing the capacity of the respective networks to generate videos at full resolution. Due to the high memory requirements of SRVP [54], it was not feasible to produce a version operating at full resolution.

The evaluation results on the *Tennis* dataset are reported in Tab. 6. With the exception of FVD, where the result is close to SAVP+, our method obtains the best performance in all the metrics. A Δ-*MSE* of 72.2% shows that actions with a consistent associated movement are learned. On the other hand, the Δ-*MSE* scores of the other baselines show that the movements associated with each action do not present a consistent meaning. Moreover, our method features the lowest ADD and a significantly lower MDR of 1.01%, which indicate that our method consistently generates players and moves them accurately on the field.

In Fig. 8, we show examples of reconstructed sequences on the *BAIR* and *Tennis* datasets. Our method achieves a precise placement of the object of interest with respect to the input sequence.

To complete our evaluation, we performed a user study on the *Tennis* dataset. We sample 23 random frames that we use as initial frames. For each of them, we generate K continuations of the sequences, one

Table 6. Comparison with baselines on the Tennis dataset. Δ-MSE*,* Δ-Acc *and MDR in %, ADD in pixels.*

	LPIPS↓	FID↓	FVD↓	Δ-*MSE*↓	Δ-*Acc*↑	(ADD, MDR)↓
MoCoGAN [58]	0.266	132	3400	101	26.4	28.5, 20.2
MoCoGAN+	0.166	56.8	1410	103	28.3	48.2, 27.0
SAVP [55]	0.245	156	3270	112	19.6	10.7, 19.7
SAVP+	0.104	25.2	**223**	116	33.1	13.4, 19.2
CADDY (Ours)	**0.102**	**13.7**	239	**72.2**	**45.5**	**8.85**, **1.01**

Table 7. User study results on the Tennis dataset.

	Agreement	Diversity	*Other* votes
MoCoGAN [58]	-3.15e-3	1.58	1.80%
MoCoGAN+	-2.84e-3	1.51	28.0%
SAVP [55]	0.0718	1.69	7.14%
SAVP+	-1.97e-3	**1.80**	5.40%
CADDY (Ours)	**0.469**	1.65	1.61%

for each action, each produced by repeatedly selecting the corresponding action. For each sequence, users are asked to select the performed action among a predefined set (*Left*, *Right*, *Forward*, *Backward*, *Hit the ball* or *Stay*). An additional option *Other* is provided in case the user cannot recognize any action. We measure user agreement using the Fleiss' kappa measure [78] that is commonly used to evaluate agreement between categorical ratings [79]. In addition, to validate that all the actions can be generated (i.e., detecting mode collapse), we compute the diversity of the action space, expressed as the entropy of the user-selected actions. Results are shown in Tab. 7. While all methods capture actions with high diversity, our method shows a higher agreement, indicating that users consistently associate the same action label to each learned action. Looking at the details of the votes for our method, we observe that most disagreements are due to cases where the player hits the ball while moving. In contrast, the other methods do not obtain high user agreement, with actions assuming different meanings, depending on the particular frame used as context.

3.3.1. Relevant publications

W. Menapace, S. Lathuiliere, S. Tulyakov, A.Siarohin and E. Ricci, "Playable Video Generation", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021
https://zenodo.org/record/5014666 [80]

3.3.2. Relevant software and/or external resources

The Pytorch implementation of our work ”Playable Video Generation” can be found at https://github.com/willi-menapace/PlayableVideoGeneration.

3.3.3. Relevant WP8 Use Cases

3C2-8 (Synthetic Video Generation from Single Semantic Label Map). Our approach presents a solution to the challenging task of unsupervised learning of playable video generation.

3.4. Procedural Terrain Generation

Contributing partners: AUTH

Synthetic terrain realism is critical in VR applications based on computer graphics (e.g., games, simulations). However, manually-created virtual terrains are still superior in quality compared to the ones derived with automated means, at the cost of significant labour and time expenses. The complexity of the real world (rocks, grass, trees, mountains) renders the creation of plausible, original terrain content still a challenging task. This issue can be bypassed using Procedural Content Generation (PCG) methods for (semi-)automatically creating new content on-the-fly, and thus replacing the artistic part of content generation with a choice of tweakable parameters and random elements. PCG algorithms can be used for on-the-fly creating 2D *terrain images* that encode 3D characteristics (e.g., altitude); this terrain image can then be transformed into a 3D terrain mesh at a final post-processing step.

Typical noise-based terrain generators (e.g., Worley [81], simplex [82], Perlin [82], value [83] or diamond-square [84]) suffer with regard to memory/computational requirements and/or output quality. More recent PCG approaches that have been applied for terrain generation, such as Software Agents [85], Erosion Modeling [86] and Evolutionary Algorithms [87] also typically require significant manual post-processing (e.g., applying an image overlay to achieve a realistic look) and/or extensive manual parameter tuning. Thus, Deep Neural Networks (DNNs) such as Generative Adversarial Networks (GANs) [88] have been alternatively explored for visual content generation. In [89] a GAN-based method is presented for multi-scale terrain texturing with reduced tiling artifacts. It involves training a GAN to upsample and texture map a low-resolution terrain input. Thus, during the inference stage, low-resolution terrain images can be translated on-the-fly to high-resolution ones; thus, the terrain is needed upfront as input to be up-scaled. Other GAN-based methods [90] [91] create mountain-like 3D terrains, using information extracted from training height map data. Acquiring height maps is not trivial, while the generated results need to be heavily post-processed, since they are missing textures and realistic visual features (e.g., grass, rivers, forests, etc.).

In comparison, AUTH developed a novel GAN-based method for procedural terrain generation with significantly more relaxed input data requirements (very loose constraints are only imposed upon the input data) and a higher diversity of terrain results. We call this proposed method *GAN-terrain*. Unlike other GAN-based terrain generation methods, it does not require sophisticated input data types (e.g., height maps). Thus, after training, it only incurs minimal manual supervision, since its required input simply consists of easily constructed (in a matter of seconds), rough 2D scatter plots of desired Points-of-Interest (PoIs) in image form; we call such a plot an "altitude image". The output is a 2D textured terrain resembling a satellite image, with colour encoding height and/or geomorphological properties (e.g., snow, water-body, forest, etc.), so that it can then be trivially post-processed and converted into a semantically annotated 3D terrain mesh. During training, the model learns to extract altitude/spatial information from colour density/distances of input PoIs.

GANs can easily learn complex real-word semantic content, like mountains, sea, deserts, islands, or flora, in a way that follows natural spatial alignment constraints (e.g., no jungles depicted in frozen Arctic regions, no rivers flowing uphill, etc.). However, simply training a GAN on a large set of ground-truth terrain images does not guarantee that the Generator will learn to produce complex content that obeys similar restrictions. Therefore, we opted for an Image-to-Image Translation GAN, training it using geographic coordinates and altitude information from a dataset of neighbouring landmarks, paired with the corresponding satellite image of their region.

The main advantage of GAN-terrain lies in its novel input strategy that simplifies the actual use of the deployed DNN model on the field: new inputs for the trained network, i.e., novel altitude images at the inference stage, can be trivially created in a matter of seconds with any image processing software. In fact, although the training/evaluation dataset for this paper was constructed using real geographic data, we have successfully tested the trained GAN-terrain model with arbitrary input images; the Generator still predicts relatively realistic terrain images.

The only existing methods partly similar to GAN-terrain are [92] and [93]. However, the first one also requires height maps for training, while both of them rely on unconditional GANs for 2D terrain image/texture generation. In contrast, GAN-terrain does not require height maps and is built upon the Image-to-Image Translation framework for increased robustness.

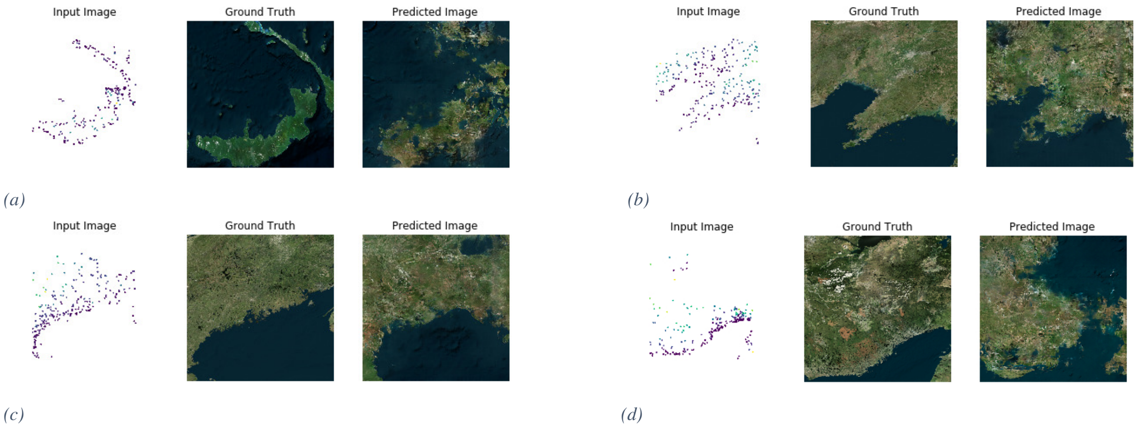
GAN-terrain. Conditional GANs for Image-to-Image Translation [94] are employed as the primary tool for completing the procedural terrain generation task. The proposed *GAN-terrain* method consists in training a conditional GAN for image synthesis so that it learns to map rough 2D Point-of-Interest (PoI) scatter maps (so-called *altitude images*) into realistic satellite terrain images containing geomorphological details. In the inference/deployment stage, after training has been completed, a similar altitude image can be easily crafted at minimal labour and time expense (within seconds), in order to be fed to the trained model as observed input image $\mathbf{X}$. The corresponding model output $\mathbf{Y}$ will be a procedurally generated 2D terrain image with rich, color-coded geomorphology that typically does not violate spatial intuitions.

To train the desired conditional GAN model under this framework, we initially collect a set of N earth surface PoIs $\mathbf{p}_i = [\lambda_i, \phi_i, R_i]^T \in \mathbb{R}^3$, $i = 1, ..., N$, composed of longitude λ_i, latitude ϕ_i and altitude R_i components. The altitude is rescaled and quantized to integer interval $[0, 253]$, assuming the height of the mount Everest (8.848m) is the maximum possible value. These N vectors can be grouped into *geographic patches*, i.e., rectangle-shaped earth regions defined from 4 PoIs. Subsequently, this set is uniformly sampled to select a set of M geographic patches, so that most earth region terrain variations are represented on the training dataset. Such a representation of all earth terrain variations is essential for high-quality, diverse content generation. Finally, for each of the M geographic patches, we collect a random number of PoIs falling geographically within it, as well as a satellite image of the patch. Patch PoIs $\mathbf{p}_{ji}$,$i = 1, ...N$ are employed to construct a 2D altitude image (λ_j, ϕ_j), of patch $j = 1, ...M$ where the horizontal/vertical coordinate corresponds to PoI latitude/longitude$(\lambda_{ji}, \phi_{ji})$, respectively, while the luminance of each point encodes PoI normalized altitude R_{ji}. Such altitude images are very sparse, since typically we sample only few Earth surface points p_{ji},$j = 1, ...M$ per patch. All other altitude image pixels have a value of 255 (white on grayscale) or (255,255,255) (white on RGB) and are excluded from altitude evaluation. This 2D altitude image, converted into image form, is an observed input image $\mathbf{X}_j$, $j = 1, ...M$. The corresponding satellite image (depicting actual geomorphology of the patch region) is employed as ground-truth output image $\mathbf{Y}_j$, $j = 1, ...M$. Thus, the training dataset is constructed by pairs $\{\mathbf{X}_j \ , \mathbf{Y}_j\}$, $j = 1, ...M$.

Each 2D altitude image $\mathbf{X}_j$ can be constructed in two slightly different ways: a) a grayscale one-channel image can be derived by encoding normalized altitude R_{ji} per-PoI as a pixel luminance value. Alternatively, a linear color palette can be used to convert normalized altitude R_{ji} into RGB color values, in order to finally obtain a three-channel colored image (e.g., one from blue to yellow, where the deepest blue/yellow denotes sea level/highest mountain peak level, respectively). Both approaches were implemented and compared in the context of this work.

As shown in Figure 9, the visual properties of the generated content are correlated with the color-coded altitude of the input PoIs; in all other respects GAN-terrain has realistically filled-in the generated terrain details fully autonomously. At model deployment-time, random input altitude images can be constructed very rapidly on-the-fly in an automated manner, thanks to the very minimal amount of required information. Even manually drawn, swiftly sketched arbitrary images can be utilized as inputs; a trained GAN-terrain model will successfully interpret them as altitude images.

In general, output diversity is an important property of a successful PCG system. In the GAN-terrain case, the purpose of the final trained GAN model during system deployment is not to precisely translate the input altitude image into an actual satellite image, but to procedurally generate a new, realistic but imaginary terrain, which may be only vaguely based on the given input. Thus, in order to enhance trained model output diversity, we optionally perform random rotations and/or flipping of each $\mathbf{X}_j$, $j = 1, ...M$



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

Figure 9. Four examples of GAN-terrain input/ground-truth/prediction triplets, having NHI similarity scores: a) 0.8393, b) 0.8824, c) 0.9482, d) 0.7944.

to augment the training dataset, without changing the corresponding $\mathbf{Y}_j$, $j = 1, ...M$. Below, we refer to GAN-terrain models trained with/without this optional augmentation strategy as "Augmented"/"Non-augmented", respectively.

This training set augmentation strategy allows the final GAN to synthesize terrain images of greater apparent diversity, by forcing it to ignore input orientation during training. Thus, during deployment of the trained model, small rotations to the input altitude image may produce arbitrarily large rotations to the output, since output orientation is in fact arbitrarily "decided" by the model and not constrained by input orientation. Thus, the Non-augmented model is forced more intensely to mimic ground-truth, while the Augmented one typically provides a more diverse result.

Evaluation. Publicly available geographical data [95] were employed in order to construct the training and testing sets for GAN-terrain. We initially collected $N = 11.2$ million world PoIs, which were utilized to create $M = 4,300$ geographic patches and attach their corresponding satellite images (of 512×512 pixels resolution) using the Microsoft Bing Maps API.

The employed GAN architecture was based on the Pix2Pix Network [94]. The network was trained using 3,000 input/output patch pairs $\{\mathbf{X}_j \ , \ \mathbf{Y}_j\}$ and was evaluated using a test set of 1,300 input/output patch pairs. Color and grayscale variants of the dataset were used for training separate GAN-terrain models. Color 2D altitude images resulted in predicted network outputs with a higher level of detail than the ones obtained using grayscale inputs, thus GAN-terrain evaluation proceeded with the color variant only. The results were impressive, as GAN-terrain successfully created highly realistic complex terrain images from very simplistic inputs.

A simple objective evaluation approach was selected, exploiting the fact that pixel color in the output image encodes semantic information. Thus, we measured the Normalized Histogram Intersection similarity (NHI) [96] between the 64-bin joint HSV color histograms [97] of each GAN-terrain prediction and its corresponding ground-truth image from the test set. Minimum/maximum NHI similarity values are $0.0/1.0$, respectively. High NHI similarity between the ground-truth and predicted image histograms can be interpreted as high *semantic concordance* among them, with regard to the distribution of visible geomorphological details (water bodies, forest, snow, mountains, etc.).

Quantitative results indicate that the mean NHI similarity between 1,300 ground-truth and predicted images is indeed relatively high (0.7665). This implies that, when the trained GAN model is given a previously unseen 2D altitude image, it synthesizes a highly similar terrain image in terms of semantic concordance. Although NHI similarity does not capture differences between the two terrain images in terms of the the exact landmass/coastline shape/orientation, this is rather irrelevant to the terrain image

Table 8. Evaluation results of the Non-augmented GAN-terrain model. Correspondence and plausibility are scored using a scale in $[1, 5]$*, while NHI similarity is a percentage. In all cases higher is better.*

Type	Mean NHI Similarity	Mean Correspondence	Mean Plausibility
Predicted Image	0.7665	4.6633	4.4682
Ground Truth	N/A	4.6138	4.5955

Table 9. Predicted images diversity comparison between the Non-augmented and Augmented model, using GIST descriptors and total variance.

Measure	Non-augmented	Augmented
Trace of GIST cov. matrix	0.18995	0.23454
Mean of the main diagonal of GIST cov. matrix	0.000207	0.000244
Variance of the main diagonal of GIST cov. matrix	3.33e-8	3.55e-8
Mean of NHI similarities of joint HSV histograms between ground-truth and predictions	0.7665	0.74172

generation task, since our goal is not to replicate the ground-truth terrain. Examples of altitude image, ground-truth terrain image and predicted output image triplets are presented in Figure 9. NHI similarity for each triplet is included for visual inspection purposes.

Additionally, we performed a subjective evaluation of generated terrain images, using 40 terrain images from our test set and 10 observers. The goal of the subjective evaluation was to let observers deduce in a systematic manner: a) whether the predicted terrain images resemble a real satellite terrain image ("plausibility"), and b) the spatial correspondence between the input 2D altitude image PoIs and the predicted terrain image ("correspondence"). We employed 20 predicted generated terrain images shuffled with 20 ground-truth terrain images for control purposes, totalling 40 terrain images. The participating subjects did not know whether each terrain image they saw was a ground-truth or a predicted one. For each image, they recorded two integer score values in the range $[1, 5]$ for plausibility and correspondence evaluation, respectively.

Subjective evaluation results, shown in Table 8, indicate that ground-truth and predicted images are nearly indistinguishable by human subjects: mean correspondence for predicted/ground-truth images was 4.6633/4.6138, respectively, while mean plausibility for predicted/ground-truth images was 4.4682/4.5955, respectively. In fact, artificial GAN-terrain images performed even better than the real ones.

Subjective evaluation was necessarily performed with a GAN model trained using the non-augmented training dataset variant of the proposed method, due to the nature of the employed "correspondence" qualitative metric. Disabling the proposed training data augmentation strategy imposes shape/orientation constraints to be learned by GAN-terrain. Thus, absence of training dataset augmentation may reduce the diversity of GAN-terrain outputs during deployment. To quantify this possibility, we trained a second GAN-terrain model using training data augmentation and then compared the predictions of the two GAN-terrain models on the test set. Evaluation consisted in calculating a GIST global image description vector [98] for each predicted terrain image in the test set, once for the Non-augmented and once for the Augmented model, and subsequently computing the mean global dispersion of these descriptors. This can be measured by averaging over the total variance (i.e., trace of the covariance matrix) of the 1,300 960-dimensional GIST vectors $\mathbf{f}_i$,$i = 1, ..., 1300$, separately for the two models.

The results, shown in Table 9, indicate that the mean global dispersion/total variance of test set predictions is significantly greater on the Augmented model variant, where our input augmentation strategy was enabled during training: it is 0.23454/0.18995 for the Augmented/Non-Augmented variant, respectively. To grasp a sense of the significance of this difference in total variance magnitude, we report that the mean/variance of the main diagonal of GIST covariance matrix in the Augmented variant is 0.000244/3.55e-8, respectively. On the other hand, mean NHI similarity of joint HSV histograms between ground-truth and predictions is slightly higher for the Non-augmented GAN-terrain model: it is 0.7665, versus 0.74172 for the Augmented case. This indicates a slight trade-off between semantic concordance and output diversity.

3.4.1. Relevant publications

- G. Voulgaris, I. Mademlis and I.Pitas, ”Procedural Terrain Generation Using Generative Adversarial Networks”, Proceedings of the EURASIP European Conference on Signal Processing (EUSIPCO), 2021. [99].
  Zenodo record: `https://zenodo.org/record/5718545#.Yg0PRpaxVEY`.

3.4.2. Relevant WP8 Use Cases

3C2 Just-in-time content creation adaptation. GAN-terrain can be employed for fast generation of synthetic terrain images with realistic appearance.

3.5. Automated Cinematography

Contributing partners: AUTH, RAI

The rapid popularization of commercial, battery-powered, camera-equipped Unmanned Aerial Vehicles (UAVs, or “drones”) during the past decade has deeply affected media production. UAVs have proven to be an affordable, flexible means for swiftly acquiring impressive aerial footage in diverse scenarios. They are suitable for movie/TV filming, outdoor event coverage for live or delayed broadcast, advertising or newsgathering, partially replacing dollies and helicopters. They offer fast and adaptive shot setup, the ability to hover above a point of interest, access to narrow spaces, as well as the possibility for novel aerial shot types not easily achievable otherwise, at a minimal cost [100].

Given the taxonomy in [100], UAV shots consist in valid combinations of Camera Motion Types (CMTs) and Framing Shot Types (FSTs). The most interesting among them involve a still or moving target/subject. Professional UAV filming requires specialised personnel for flight and filming control, i.e., separate drone and camera operators. That renders the possibility of fully autonomous drones highly attractive, since it would reduce the need for human operators and the time overhead that comes with executing demanding CMTs, such as “Orbit”. The ultimate goal would be a fully autonomous agent that can smoothly control the UAV on-the-fly in order to execute the desired CMT, while taking into consideration basic navigation and cinematography requirements.

3.5.1. Autonomous execution of Camera Motion Types using MPC distillation

A class of very popular methods for achieving autonomy in complex unknown and stochastic environments is Deep Reinforcement Learning (DRL), which relies on Deep Neural Networks (DNNs). Classical control algorithms such as Model Predictive Control (MPC) become highly complex in large scale problems and require sensor information that in most cases is not available, or is very expensive to acquire. DRL agents on the other hand can learn to navigate in stochastic environments with limited sensor information [101][102].

In the context of T5.2, AUTH tried to leverage the benefits of both worlds, with as little overhead as possible: after the proposed novel DRL-based agent has been trained, it constantly and autonomously adjusts the UAV's trajectory so that a desired CMT is executed, without relying on detailed 3D maps or on knowledge of the 3D position of the target being filmed. It only requires RGB video input from the UAV-mounted camera. However, during training, it exploits knowledge conveyed by a suitable MPC controller, which requires 3D target position information in order to function. The training process is complemented by collision and occlusion avoidance objectives. In the end, the proposed MPC-distilled DRL agent exhibits superior performance in comparison to the baseline one, while simultaneously requiring minimal sensor input in comparison to the MPC controller during the test stage.

Unlike DRL for generic robotic and UAV control, DRL specifically for UAV cinematography tasks has not been explored in depth. A few relevant algorithms have been presented for frontal view person shooting [103], for object tracking using multiple drones, or for combining trajectory optimisation and DRL in order to automate artistic choices [104]. In contrast, the purpose of the proposed method is to present a novel, purely visual-based, small-sized and quickly trainable DRL agent for UAV path planning in the context of CMT execution. There is no need to rely on complex and time consuming trajectory optimisation, pose estimation, 3D map reconstruction or state estimation methods. Just one compact Convolutional Neural Network (CNN) takes a current RGB image as input and returns the optimal control signal for the UAV, at each time step.

Technical Background. Key ideas used in the proposed method are reviewed below. The formulation consists of a semi-expert MPC controller, able to autonomously orbit around a target and avoid obstacles, and a CNN-based actor trained by DDPG with policy distillation.

1. Model Predictive Control. MPC is a discrete time control technique that provides a solution to the finite horizon, constrained optimal control problem:

$$\begin{aligned} \min_{u(\cdot)} \quad & \Phi(\mathbf{x}(t_f)) + \int_0^{t_f} l(\mathbf{x},\mathbf{u},t)\,dt \\ \textit{subject to} \quad & \mathbf{x}' = f(\mathbf{x},\mathbf{u},t),\ \mathbf{x}(0) = \mathbf{x}_0, \\ & g(\mathbf{x},\mathbf{u},t) = 0, \\ & h(\mathbf{x},\mathbf{u},t) \geq 0, \end{aligned} \tag{1}$$

where t_f is the time horizon, $\mathbf{x}_0$ a given initial state, $\Phi(\cdot)$ the final cost and $l(\cdot)$ the intermediate cost function. $f(\cdot), g(\cdot)$ and $h(\cdot)$ are time-dependent vector fields defining the system dynamics, the equality constraints, and the inequality constraints, respectively [105]. The problem's associated optimal value function V (cost-to-go) is defined as:

$$V(t,\mathbf{x}) = \min_{\mathbf{u}} \ \Phi(\mathbf{x}(t_f)) + \int_0^{t_f} l(\mathbf{x},\mathbf{u},t)\,dt. \tag{2}$$

MPC aims to solve the finite horizon, constrained optimal control problem, with explicit knowledge of the system's state and dynamics being required. In the examined task, partial dynamics knowledge is available since only the drone kinematics are known a priori; not the environment dynamics. Additionally, the only information available to the controller are the low-frequency RGB video frame it receives at the sampling rate of the action. That makes the choice of sampling rate vital, since high sampling frequencies result in more accurate control policies, but require faster inference times, which is not always feasible on embedded/on-drone computational hardware. Thus, a sampling period of $T = 0.4$ seconds was chosen.

2. Partially Observable Markov Decision Processes. Partially Observable Markov Decision Processes (POMDPs) are extensions to Markov Decision Processes (MDPs) with partially observable states, allowing for decision making under uncertain conditions. A MDP is composed of a state space S, an initial state space S_0 having an initial state distribution $p(s_0)$, an action space A, a state transition probability distribution $p(s_{t+1}|s_t,\ a_t)$ satisfying the Markov property, and a reward function $r(s_t,\ a_t) : S\ x\ A \rightarrow R$ characterizing the environment's feedback when action a_t is executed at state s_t. The state space and the action space of a MDP could be either discrete or continuous. Since UAV control signals are continuous, MDPs and POMDPs with continuous state and action spaces are described below.

MDPs are generally addressed via Reinforcement Learning (RL), i.e., by training a model which learns an optimal policy π through repeated trial-and-error sessions. The policy could be either deterministic or stochastic, where a deterministic policy, denoted as $\mu(s) : S \rightarrow A$, projects states to actions, and a stochastic policy, denoted as $a \quad \pi(\cdot|s) : S \rightarrow P(A)$, where $P(A)$ is the set of probability measures on A, returns the probability density of available state and action pairs (s, a). If the action space is continuous, the stochastic (or deterministic) policy is parameterized as a function $a \quad \pi(\cdot|s, \theta)$ (or $a = \mu(s, \theta)$), where θ refers to the parameters of the function. To keep the notation simple, policy π (or μ) is assumed to be a function of θ and all the gradients are assumed to be with respect to θ.

RL methods estimate value functions of states and action-value functions of state-action pairs. For stochastic policies, the value function is defined as:

$$V_\pi(s_t) \;=\; \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+1}, a_{t+1})|s_t\right], \tag{3}$$

where γ is a real discount factor ranging from 0 to 1. The action-value function is:

$$Q_\pi(s_t, a_t) \;=\; \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+1}, a_{t+1})|s_t, a_t\right]. \tag{4}$$

A stochastic policy is optimal if an agent receives the maximum expected future discounted reward (which is also referred to as target function) by executing it:

$$J(\pi) \;=\; \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]. \tag{5}$$

For a deterministic policy, the value functions, action-value functions and target function can be obtained by replacing $a_t = \pi(\cdot|s_t)$ with $a_t = \mu(s_t)$ in (3), (4), (5).

A MDP becomes a POMDP when an agent cannot observe state s_t, but instead receives an observation o_t, having a distribution $p(o_t|s_t)$. The observation sequence no longer satisfies the Markov property: $p(o_{t+1}|a_t, o_t, a_{t-1}, o_{t-1}, \cdots, o_0) \neq p(o_{t+1}|o_t, a_t)$. As a consequence, an agent needs access to the entire history trajectory $h_t = (o_t, a_{t-1}, o_{t-1}, \cdots, o_0)$ to infer the current state s_t and make decisions based on it.

The goal of RL in partially observable settings is thus to learn an optimal policy $a_t \quad \pi(\cdot|h_t)$ that projects history trajectories to action distributions by maximizing (5).

3. Deep Reinforcement Learning and DDPG. DDPG is a model-free DRL algorithm for continuous action control. It is a significant improvement in comparison to Deterministic Policy Gradient due to three novelties: the utilisation of Deep Neural Networks (DNNs) for the functional approximation of the actor and the critic, the use of target networks and the use of experience replay [106].

The main idea of actor-critic methods is to model the actor with a DNN and find the synaptic weights that encode the optimal policy π^*. This is achieved by updating its weights in the direction of the gradients of the expected future reward (5), using error back-propagation and a form of gradient descent. Because that gradient needs an unbiased estimator of the Q function (4) in order to guide the weights of the actor towards the optimal policy, the critic network is first updated towards the direction of the gradient of the temporal difference at each training iteration.

Target networks are networks of identical structure as the actor and critic, but their weights are updated in such a way to slowly follow the weights of the actual actor and critic. This way, stability in the learning process of the actor and the critic is improved. The concepts of target networks and the experience replay have been introduced in Deep Q-Learning [107] and have become common in DRL algorithms.

Actor-critic methods can either be on-policy or off-policy, meaning that the action taken at every step for exploring the environment is sampled from the actor network or from another policy, respectively. DDPG is off-policy, in the sense that the action used at every step is the sum of the actor policy at the current state of the environment plus an appropriate exploratory noise. In this paper, the Ornstein-Uhlenbeck process is used as the additive exploration noise.

4. Policy Distillation. Policy distillation is the transfer of knowledge from a teacher actor T to a student actor S. This is traditionally achieved by generating a dataset $D_T = \{s_i, q_i\}_0^N$ [108] using the teacher and, subsequently, exploiting it as ground-truth to train the student Q-network through regression.

One core novelty of the proposed method is to exploit a suitable MPC controller as teacher under a policy distillation framework, in order to augment the training process of a neural DRL agent. Thus, policy distillation is modified in this paper in order to account for the semi-expert nature of the employed teacher. The aim is to transfer knowledge to the student, while the latter becomes more robust and learns according to rewards that cannot be incorporated in the MPC setting. The processes of knowledge transfer and DRL training have to be simultaneous, leading to *on-line policy distillation*.

Proposed Method. The proposed method consists of a properly adapted form of a DDPG agent, trained using a novel, task-specific reward function for autonomous UAV cinematography, and the novel on-line policy distillation term from a MPC teacher. Autonomous execution of the “Orbit” CMT has been selected below as the desired cinematography task, due to its challenging nature (it is difficult to execute manually) and its high aesthetic appeal. However, the proposed method may alternatively be employed for executing *any* UAV cinematography CMT, provided that a proper MPC controller can be designed.

1. MPC for Executing Orbit CMTs. Holding the assumption that the UAV is velocity-controlled based only on its kinematics, the motion equations are:

$$\begin{aligned} \dot{x} &= a_x \\ \dot{y} &= a_y \\ \dot{z} &= a_z, \end{aligned} \tag{6}$$

where a_i is the velocity set-point along direction i and the vector $\mathbf{a} = [a_x, a_y, a_z]^T$ represents the action. We define $\mathbf{q} \in \mathbb{R}^3$ as the actual position of the UAV in the environment and $\mathbf{q}_d \in \mathbb{R}^3$ as the orbital position where, ideally, the UAV should currently be in if executing a proper “Orbit” CMT.

The MPC controller’s task is to track the orbital trajectory in 3D space, which is computed according to the “Orbit” equation in [100]. This computation requires the 3D position of both the drone and the filmed target/subject to be known. Furthermore, obstacle avoidance is implemented by surrounding all obstacles by spheres and then using the distance of the UAV from the sphere as a constraint. Obviously, this also requires the 3D positions in space of all obstacles to be known. The inequality constraints that arise are the following ones:

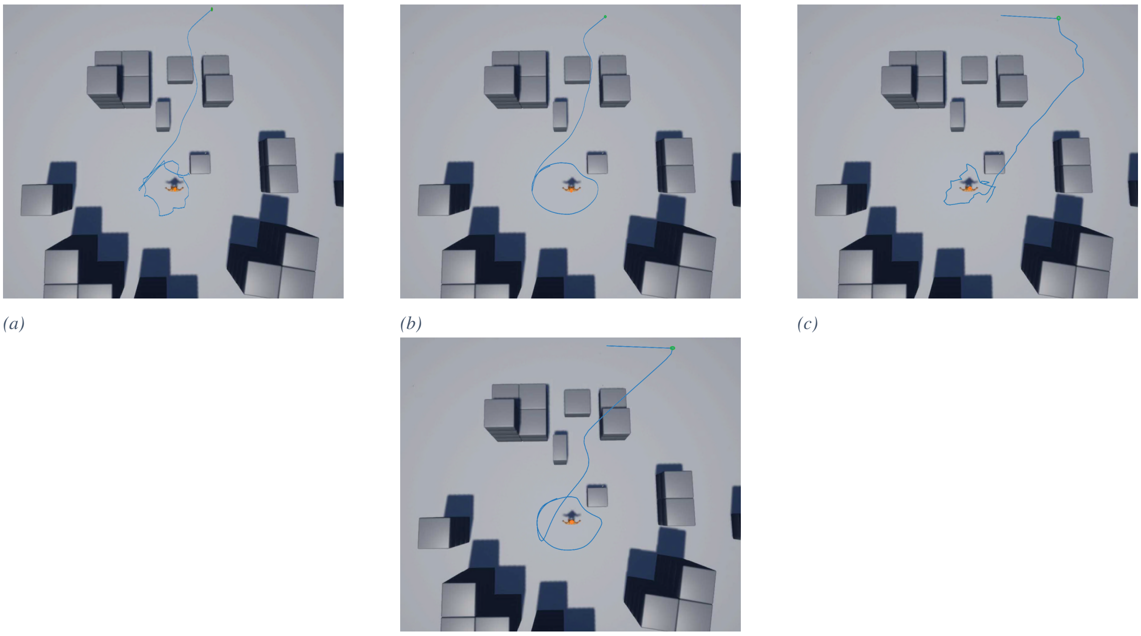
$$d_i = ||\mathbf{q} - \mathbf{c}_i||^2 - r_i^2, \; for \; i = 1, 2, .., M, \tag{7}$$

where M is the number of spheres and c_i/r_i are the center/radius of the i-th sphere, respectively. Thus, the $M + 1$ constraints are:

$$\begin{aligned} d_i &\geq 0, \; for \; i = 1, 2, ..., M \\ z &> 0. \end{aligned} \tag{8}$$

Function $l(\cdot)$ is constructed as the sum of the distance between the UAV and the ideal orbital position at that time instant, the square of the action vector and a term for creating smooth action signals:

$$l_t = ||\mathbf{q}_t - \mathbf{q}_{d_t}||^2 + ||\mathbf{a}_t||^2 + \mathbf{a}_{t-1}^T \mathbf{a}_t. \tag{9}$$



(a)

(b)

(c)

(d)

Figure 10. Top-down view of the $x-y$ plane trajectories outputted by the proposed MPC-distilled DDPG agent ((a) and (c)) and the semi-expert MPC controller ((b) and (d)). In (a),(b) and (c),(d) the starting UAV position is $[0,0]$ and $[0,20]$, respectively.

2. MPC-distilled DDPG. After the task-specific MPC controller has been designed, the proposed purely-vision based DRL agent can be trained using on-line policy distillation.

The main idea is to leverage the useful properties MPC carries in a DRL framework. Since the task of autonomous "Orbit" execution from visual input only is highly complex, solving it with a baseline, pure DRL method would result in a prohibitively high required training time and in a DNN size too large for embedded computers. Moreover, accurate reward shaping is crucial, but rather difficult in large-scale problems such as this one, since any defects in its design would likely lead to even higher training times, instabilities and sub-optimal solutions.

All these issues are significantly ameliorated by introducing a loss term that penalizes distance between the student's action and the MPC-teacher's action, at each step of the training process. This serves as a way to exploit the known UAV kinematics model while training the DRL agent, in order to improve its performance. The proposed distillation loss term is incorporated into the gradient of the cost J [106], in order to guide the policy being learnt near to the MPC controller's policy. Notably, the MPC controller is not a perfect expert (as is typically the case in imitation learning) but simply an automated algorithm with access to additional information (i.e., the target's 3D position at each time step). The gradient that is used to update the weights of the actor DNN is:

$$\begin{aligned}\nabla_{\theta^\mu} J = \mathbf{E}_{s_t\ \rho^\beta}\left[\nabla_{\theta^\mu} Q(s,\mu(s)|\theta^Q)|_{s=s_t} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_t}\right. \\ +\ \nabla_{\theta^\mu} l(\mu(s|\theta^\mu), a_{MPC}(s))|_{s=s_t}\end{aligned} \tag{10}$$

In the problem tackled, the state $s_t \in \mathbf{R}^{84x84x3}$ is the RGB image that the drone camera captures in time t, $\mathbf{a}_t \in \mathbf{R}^3$ the action vector as defined in (6) and $\mathbf{a}_{MPC} \in \mathbf{R}^3$ the corresponding MPC action given UAV position $\mathbf{q}_t$.

4. Reward shaping for the DDPG agent. In order to train a suitable DDPG agent, 4 individual rewards were linearly combined to form the final complete reward function. These rewards are presented below.

First, the UAV must learn how to orbit the target. To achieve this, the next orbital trajectory waypoint $\mathbf{q}_d$ is computed based on the "Orbit" equations from [100] and the following error is defined:

$$e_d = ||\mathbf{q} - \mathbf{q}_d||^2. \tag{11}$$

Next, an error threshold is defined, above which the orbital reward becomes zero [103]. Eventually, the reward for the orbital trajectory is:

$$r_d = \begin{cases} 0 & , \ if\ e \geq e_{th}, \\ 1 - \frac{e_d}{e_{th}} & , \ else \end{cases} \tag{12}$$

Additionally, a reward that punishes low or high altitudes is defined. This is motivated by the fact that the footage needs to be captured from an altitude not much greater than that of the filmed target, thus avoiding too large a gimbal pitch angle which would corrupt the CMT. Two options are viable to achieve this: a) explicitly reward small pitch angles, or b) specify a band of altitudes within which the reward is zero, but outside of it, it is -0.5. Option b) was chosen for this paper:

$$r_z = \begin{cases} -0.5 & , \ if\ z \geq z_{max}\ or\ z \leq z_{min}, \\ 0 & , \ else \end{cases} \tag{13}$$

Moreover, obstacle avoidance is an essential factor for robotic navigation within complex environments. Although obstacle avoidance is implemented by the MPC controller and the student DDPG actor will distill that knowledge, an optional obstacle avoidance reward term was still defined for DRL training. This is significantly more accurate than the bounding sphere approximations internally computed by the MPC controller. An on-board LiDAR is used to sense obstacles located within a range of up to d_{max} meters from the UAV, along a FoV of 360^o around it. Then, the minimum value $d_{lidar_{min}}$ of the LiDAR output is used to form the reward:

$$r_{obs} = -e^{-\alpha \frac{d_{lidar_{min}}}{d_{max}}}. \tag{14}$$

Table 10. Accumulated rewards for the test cases shown in Fig. 10, using the fully trained agents. Evidently, the MPC-distilled DDPG agent is able to reap higher/lower rewards when the target is visible/invisible at the very first steps, respectively. The baseline DDPG agent completely fails to achieve the task and crashes.

Accumulated rewards for the cases of Fig. 10		
Algorithm	**Initial Position** $[0, 0]$	**Initial Position** $[0, 20]$
MPC	0.72595	0.69523
MPC-distilled DDPG	0.74524	0.50416
Baseline DDPG	0.10650	0.39576

Lastly, a visual occlusion avoidance reward term is defined, since the UAV is expected to capture clear images of the target being filmed while executing the "Orbit" CMT. This reward term was borrowed from [104], where a semantic segmentation map is used during training to find the on-frame surface of the bounding box surrounding the filmed target. The UAV is set to various positions inside the environment, where the target/subject is not occluded from the camera's FoV. Then, the various measured distances between the UAV and the bounding box of the target/subject are used as data pairs for curve fitting a polynomial $\mathbf{p}(d)$. The actual reward term is:

$$r_{occ} = e^{-\beta(S_{actual} - S_{measured})^2}, \tag{15}$$

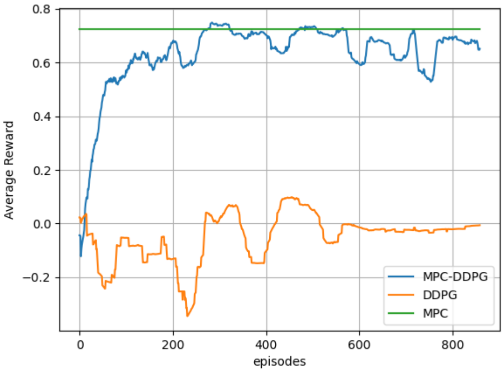


Figure 11. Average reward for the MPC-distilled DDPG and the baseline DDPG, with both of them trained using the proposed reward function. Evidently, the proposed distillation loss term significantly speeds up learning. The corresponding MPC controller's reward is shown in green, for comparison purposes.

where $S_{actual}/S_{measured}$ is the estimated actual/measured bounding box surface, respectively.

All the individual reward terms described above are combined into the final reward function:

$$r = \sigma_d r_d + \sigma_z r_z + \sigma_{obs} r_{obs} + \sigma_{occ} r_{occ}, \tag{16}$$

where σ_i are the reward coefficients.

Evaluation. The method was evaluated by training an MPC-Distilled DDPG agent in a virtual environment from the AirSim simulator [109]. The environment consists of obstacles in the form of grey boxes and a still target around which an "Orbit" CMT must be performed. The target is orange in color, so that it is clearly distinguishable in the RGB images the UAV gets. The environment is visible in Fig. 10.

The employed hyperparameter values were the following ones: the action sampling time was set to 0.4 seconds, the MPC horizon to 10 time steps, the reward coefficients to $\sigma_d = 1$, $\sigma_z = 1$, $\sigma_{obs} = 0.7$, $\sigma_{occ} = 0.6$ and the learning rates for the actor and the critic subnetwork to 0.0001 and 0.0002, respectively. Identical CNNs were used for the actor and critic subnetworks.

The average rewards for MPC-Distilled DDPG and the baseline DDPG, along with comparisons against the MPC controller, are presented in Fig. 11. Evidently, while the baseline DDPG is unable to learn in 1,000 episodes a good policy, MPC distillation allows the proposed augmented DDPG agent to rapidly learn the desired task in just 200 episodes. This is extremely fast for such a difficult task in such a complex environment. On the other hand, unlike the MPC controller, the MPC-Distilled DDPG can operate after training without access to the target's 3D position at each time step.

Fig. 10 depicts a top-down view of the trajectories that the MPC-distilled DDPG agent and the MPC agent have followed. In Fig. 10a,b the agent's starting position is $[0, 0]$ while in Fig. 10c,d it is $[0, 20]$. Evidently, the two trajectories look similar when there is no occlusion, or when the distance from the obstacles is large enough, but with the proposed MPC-distilled DDPG agent's trajectory being a lot less smooth once the UAV reaches the target's vicinity. However, the proposed agent chose to follow from the start a trajectory that provides the minimum occlusion, while the MPC controller does not account at all for visual occlusions.

The above observations illustrate the key idea of the proposed method: it leverages semi-expert MPC knowledge to reduce DNN size/complexity and significantly lower required training time, while allowing the agent to learn to behave according to the rewards designed for the DDPG algorithm. Thus, the proposed agent is able to choose trajectories satisfying requirements that the semi-expert policy does not account for (i.e., visual occlusion avoidance, improved obstacle avoidance). Although the final behaviour of the trained

MPC-distilled DDPG agent is not as smooth as that of the MPC controller, the proposed agent has the advantage of not requiring knowledge of the target's 3D position at each time step.

Moreover, additional reward terms could be added in the future to its training process in order to smoothen the produced UAV trajectories. This could be combined with tasking the proposed agent to also control the camera gimbal based only on visual information. Finally, the method is going to be evaluated with moving filmed targets as well.

This is still on-going research, expected to finish within 2022. A relevant technical report is under preparation, with the intention to have it submitted as a journal article.

3.5.2. Autonomous execution of Camera Motion Types with differential control outputs

In the context of T5.2, AUTH also experimented with a different DRL-based formulation of the autonomous CMT execution task. In this formulation, the actions of the DRL agent are not absolute velocity vectors but differential commands to the autopilot hardware, forcing the UAV to slightly adjust its current velocity vector in a relative manner (turn to the left/right, turn up/down, accelerate/decelerate). Thus, ideally, not only the filmed target's 3D position will not be required to be known during test, but neither the UAV's.

In order to differentiate this method as much as possible from the one presented in Subsection 3.5.1, several research parameters were changed in this work:

- A different virtual training environment in AirSim was employed, with significantly more realistic appearance, much larger scale and a moving filmed target (a car).
- MPC distillation was not exploited at all.
- The improved TD3 DRL algorithm [110] was employed, instead of the less efficient DDPG.
- Instead of simple CNNs, the agent' DNNs were composed of CNN+LSTM architectures, equipping them with long-term memory in lieu of the reactive agents of subsection 3.5.1. This is expected to make them much more robust in conditions of partial environment observability.
- Collision avoidance and visual occlusion avoidance rewards were not included here, in order to purely focus on the autonomous CMT execution task.

Twin Delayed Deep Deterministic Policy Gradient (TD3) [110] is a recent development, building upon DDPG. Instead of using a single critic network to get the predicted Q-value in state s_{n+1} for bootstrapping the Q-value of the current state s_n and action a_n, it uses two separate networks, $Q_{\boldsymbol{\alpha}_1}$ and $Q_{\boldsymbol{\alpha_2}}$ and chooses the minimum output value. In this way, the overestimation problem in Actor-Critic methods [111] is addressed. Additionally, to decouple value and policy, the "target" policy network that is used for bootstrapping is updated periodically, after m regular networks' updates.

States. Since it is desired that the UAV is self controlled using information about its environment that it can obtain without cost-ineffective and heavy sensors or data sent to it by a third party sensor (would induce additional latency), RGB images $\mathbf{I}_n \in \mathbb{R}^{N_1 \times N_2}, N_1, N_2 \in \mathbb{N}$ captured by a standard RGB camera mounted on the vehicle are used as the main state description. In order to aid the gimbal control, additional information about the gimbal orientation is included, more specifically the relative angle ω between the 3D UAV velocity vector $\mathbf{u}_n = [u_{n1}, u_{n2}, u_{n3}]^T$ (WCS) and the 3D camera axis $\mathbf{c}_n$:

$$\cos(\omega) = \mathbf{u}_n^T \mathbf{c}_n / |\mathbf{u}_n| \,. \tag{17}$$

A visual representation of ω can be seen in Figure 12.In conclusion, at time instance n, the observation o_n about the real state s_n will be the set $o_n = \{\mathbf{I}_n, \omega\}$.

Actions. Smooth, visually pleasing cinematographic camera motion can only be achieved via precise drone and camera gimbal motion control. Although most related works [112, 113, 104, 114] use a small set

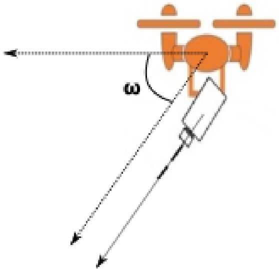


Figure 12. Visual representation of ω, the relative angle between the 3D UAV velocity vector and the 3D camera axis.

of discrete high level action commands to the UAV flight controller in order to translate them to low level motor commands, it can be easily deduced that having such limited options for movement would not be ideal for cinematographic motion. Thus, a continuous action space is selected, which is described by four continuous variables:

- Instant UAV pitch and yaw rotation angle changes $\Delta\theta_{D,n}$ and $\Delta\phi_{D,n}$,
- instant camera pitch and yaw rotation angle changes $\Delta\theta_{C,n}$ and $\Delta\phi_{C,n}$, and
- linear UAV acceleration/deceleration along UAV velocity vector direction Δu_n.

Using certain values for these variables, the UAV can be successfully moved to any position $\mathbf{X}'_{n+1}$ and, similarly, the camera axis can be rotated to point to any direction, becoming $\mathbf{c}'_{n+1}$.

Given a certain action $\mathbf{a}_n = (\Delta\theta_{D,n}, \Delta\phi_{D,n}, \Delta\theta_{C,n}, \Delta\phi_{C,n}, \Delta u_n)$ the UAV velocity changes from $\mathbf{u}'_n$ to $\mathbf{u}'_{n+1}$:

$$\mathbf{u}'_{n+1} = \mathbf{R}_y(\Delta\theta_{D,n})\mathbf{R}_z(\Delta\phi_{D,n})\frac{\mathbf{u}'_n}{u_n}(u_n + \Delta u_n), \tag{18}$$

where:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad \mathbf{R}_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

With the change in the velocity vector $\mathbf{u}'_{n+1}$, the UAV will move from point $\mathbf{X}'_n$ to $\mathbf{X}'_{n+1}$:

$$\mathbf{X}'_{n+1} = \mathbf{X}'_n + \mathbf{u}'_n\Delta t, \tag{20}$$

where Δt is the time step duration. Action $\mathbf{a}_n$ will have an effect on $\mathbf{c}'_n$ as well:

$$\mathbf{c}'_{n+1} = \mathbf{R}_y(\Delta\theta_{C,n})\mathbf{R}_z(\Delta\phi_{C,n})\mathbf{c}'_n, \tag{21}$$

where, the eventual pose of the gimbal with respect to the x axis in the BFS could also be represented by:

$$\theta_{C,n+1} = \theta_{C,n} + \Delta\theta_{C,n}, \tag{22}$$

and:

$$\phi_{C,n+1} = \phi_{C,n} + \Delta\phi_{C,n}, \tag{23}$$

since the gimbal motion is restricted to only rotations around the pitch and yaw axes.

Ultimately, $\mathbf{u}'_{n+1}$ and $\mathbf{c}'_{n+1}$ are given to the flight controller to translate it to actual propeller and gimbal motor torques $\boldsymbol{\tau}_{p,n+1}, \boldsymbol{\tau}_{g,n+1} = f(\mathbf{u}'_{n+1}, \mathbf{c}'_{n+1})$.

Rewards. A reward is essentially a metric that defines how ideal the actions that an actor took at a certain time instance were, regarding to the state in that time instance. In our case, a good action would be one that would:

- drive the UAV from $\mathbf{X}'_n$ to a new 3D location $\mathbf{X}'_{n+1}$ so that it coincides with 3D point $\mathbf{X}_{n+1}$ which is computed using the predefined mathematical definition of the desirable CMT [115].
- keep the target centered on the image, by tweaking the UAV camera look-at vector $\mathbf{c}'_n$ so that it coincides with $\mathbf{c}_{n+1}$, where:

$$\mathbf{c}_{n+1} = \frac{\mathbf{P}_{n+1} - \mathbf{X}'_{n+1}}{\left|\mathbf{P}_{n+1} - \mathbf{X}'_{n+1}\right|}. \tag{24}$$

Taking inspiration from the existing literature [112, 116], rewards are shaped to be a function of the euclidean distance between the resulting vectors $\mathbf{X}'_{n+1}, \mathbf{c}'_{n+1}$ and $\mathbf{X}_{n+1}, \mathbf{c}_{n+1}$, forming the following rewards:

- Positional reward: favor small euclidean distance between $\mathbf{X}_{n+1}$ and $\mathbf{X}'_{n+1}$:

$$R_{n1} = \begin{cases} 1 - \frac{\|\mathbf{X}_{n+1} - \mathbf{X}'_{n+1}\|_2}{r}, & \|\mathbf{X}_{n+1} - \mathbf{X}'_{n+1}\|_2 \le r \\ 0, & \|\mathbf{X}_{n+1} - \mathbf{X}'_{n+1}\|_2 > r \end{cases}$$

  where r is a fixed distance radius, out of which the returned reward equals to zero.

  Alternatively, instead of rewarding the close proximity of the reached UAV position to the ideal UAV position at time $n+1$, the similarity between the selected velocity $\mathbf{u}'_{n+1}$ and the ideal one $\mathbf{u}_{n+1}$ can be considered:

- Velocity reward: favor minimized difference between $\mathbf{u}'_{n+1}$ and $\mathbf{u}_{n+1}$:

$$R_{n2} = \kappa \frac{\mathbf{u}'_{n+1} \cdot \mathbf{u}'_{n+1}}{\|\mathbf{u}'_{n+1}\| \|\mathbf{u}_{n+1}\|} - \lambda \sqrt{\|\mathbf{u}'_{n+1}\|^2 - \|\mathbf{u}_{n+1}\|^2}. \tag{25}$$

  Essentially, the velocity direction similarity is decoupled from the velocity norm error and combines them with different weights κ, λ to give different degrees of importance.

- Lock-target reward: penalize difference between $\mathbf{c}_{n+1}$ and $\mathbf{X}'_{n+1}$:

$$R_{n3} = -\|\mathbf{c}_{n+1} - \mathbf{c}'_{n+1}\|_2. \tag{26}$$

To combine the three reward functions into a single one, a simple linear combination is proposed:

$$R_n = \alpha R_{n1} + \beta R_{n2} + \gamma R_{n3}, \;\; \alpha, \beta \in \mathbb{R}. \tag{27}$$

Training Environment. Since training a reinforcement learning agent in a real world environment would be an extremely expensive, time consuming and dangerous process, the optimal cinematographic policy could be learned in a simulated environment.

For this purpose, a large realistic 3D Environment was created on the Unreal Engine 4 simulator along with Airsim plugin for drone and gimbal manipulation, via the Airsim API. It features a single car that runs in a predefined path, and a single drone with a gimbal holding an RGB camera mounted on the bottom center part. The surrounding environment is rich in terrain textures, includes multiple plant species, has two lakes and several buildings. The developed simulation environment can be observed in Figure 13.

Proposed DRL method. Having opted for continuous control of both the UAV and the camera gimbal, the evident solution is to propose a deep policy gradient reinforcement learning framework. To keep up with recent developments in the area, state-of-the art TD3 [110] RL agent is used.

Due to the fact that an RGB image taken from the UAV camera gives only partial information about the true state of the environment, the problem needs to be formulated as a partially observable Markov decision

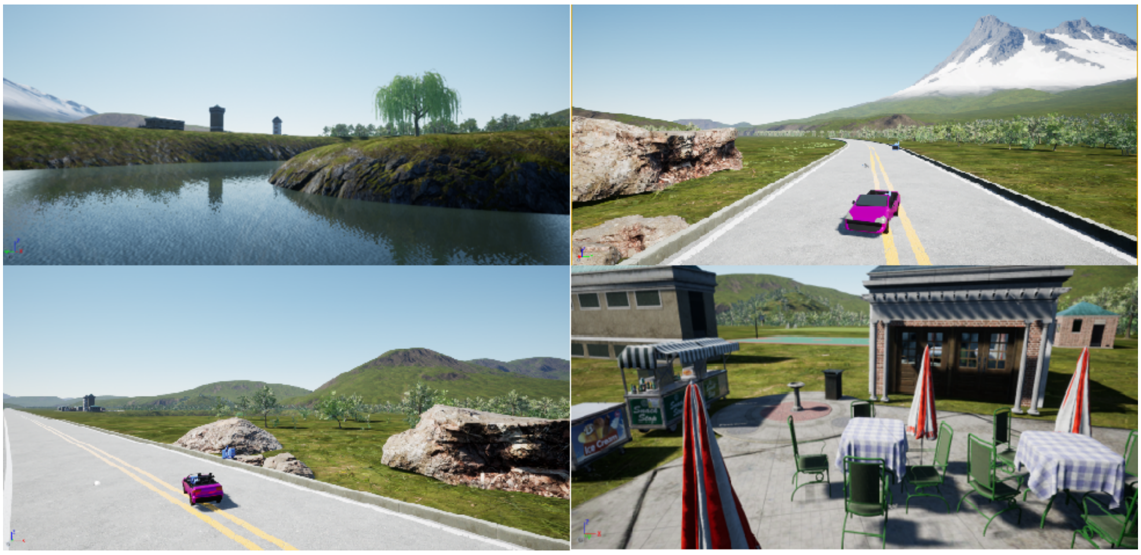


Figure 13. Screenshots of the simulated 3D environment.

process (POMDP). POMDPs extend the standard MDP quadruplets by adding an observation space $\mathcal{O}$, which contains the agent's observations of the true states $\mathcal{S}$, and an observation model Ω. Therefore, they can be defined by the 6-tuple $(\mathcal{S}, \mathcal{A}, P, R, \mathcal{O}, \Omega)$. As the agent moves through space, using past observations along with the current one can give a more complete perception about the true current state. In practice, previous works have used LSTM networks to deal with partially observability [117, 118, 119] as they can encode past information in their hidden state.

In that sense, a TD3 agent is proposed, with both the actor and the critic networks being CNNs following a lightweight backbone architecture, followed by LSTM layers to capture as much of the true environment state as possible, through a history of observations:

$$\mathcal{H}_n = \{(o_i, \mathbf{a}_i, R_i, o_{i+1})\}_{i=0}^{n}. \tag{28}$$

The actor network follows the architecture depicted in Figure 14. The critic networks follow the same architecture, with the exceptions being: i) the current action selected by the actor network is also given as input to the convolutional module, ii) the last linear layer has only 1 neuron, the activation of which represents the estimated Q-value (linear activation function is used).

Evaluation. The "Ascent" UAV CMT is used to evaluate the proposed RL method. According to [100]: *"Ascent" is a parametric CMT, where the camera gimbal is slowly rotating (mainly along the pitch axis), so as to always keep the still or linearly moving target properly framed. The UAV linearly backs away from the target from behind or from the front, at a steadily increasing altitude (relative to the target), with constant velocity. The UAV keeps flying away from the target, with the camera still focusing on the latter.*

The proposed TD3-LSTM agent was trained on the simulated 3D environment that was described in the previous section for 2,000 episodes of 20 time steps each. The velocity reward from Eq. (25) was only used for this experiment with $\kappa = 8$ and $\lambda = 0.02$. In the preliminary experiments performed up to now, the camera was automatically rotated to always point to the target. Consequently, the actor network has only 3 output neurons, to extract commands for $\Delta\theta_{D,n}$, $\Delta\phi_{D,n}$, and Δu_n.

Figure 15 shows the rewards that were accumulated after each episode. As it can be observed, the rewards are maximized approximately at episode 250, meaning that the agent has already almost successfully learned the CMT that it was assigned to perform.

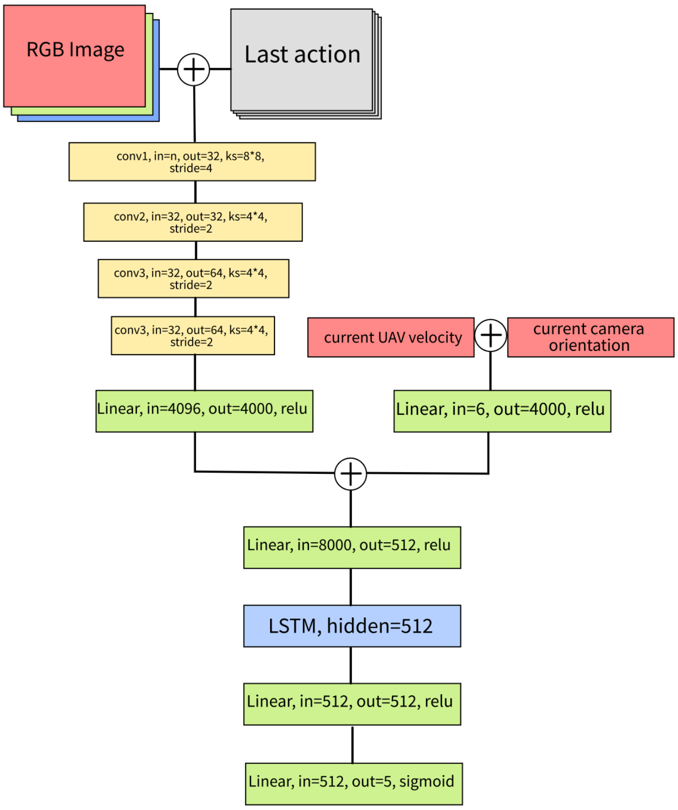

RGB Image
Last action
conv1, in=n, out=32, ks=8*8, stride=4
conv2, in=32, out=32, ks=4*4, stride=2
conv3, in=32, out=64, ks=4*4, stride=2
conv3, in=32, out=64, ks=4*4, stride=2
current UAV velocity
current camera orientation
Linear, in=4096, out=4000, relu
Linear, in=6, out=4000, relu
Linear, in=8000, out=512, relu
LSTM, hidden=512
Linear, in=512, out=512, relu
Linear, in=512, out=5, sigmoid

Figure 14. Actor network architecture. The encircled plus sign represents the concatenation operation. The information flows in a top-down fashion. To concatenate the input RGB image with the last action, each component of the latter is converted to a single channeled image of identical spatial dimensions to the RGB one. The concatenation is then conducted along the channel dimension.

The experimental results showed that the proposed agent was indeed able to successfully plan correct UAV trajectories following the constraints imposed by the "Ascent" CMT.

This is still on-going research, expected to finish within 2022. The proposed method will be extended so that the DRL agent also directly controls the camera gimbal. Additionally, besides "Ascent", the method will also be evaluated in all the 10 target-tracking UAV CMTs described in [100] and [115]. A relevant technical report is under preparation, with the intent to have it submitted as a journal paper.

3.5.3. Leader and breakaway detection in racing videos

Two related and important problems in TV coverage of racing sports events (e.g., cycling, boat or car races) are automated leader and breakaway detection. The leader's proper framing is a pivotal issue from a cinematography perspective, when the racing event has a linear spatial deployment. Similarly, a breakaway is the occasion where, starting from a spatially compact racer group, one athlete accelerates and distances fast from the rest of the athletes of the group, making this event very significant for deciding frame composition. Automating these tasks in the context of autonomous UAV cinematography applied to racing coverage would indeed be highly useful: it is important that the UAV-mounted camera automatically centers on a potential leader, while detecting a breakaway could trigger automatic CMT re-selection based on a rule system. Additionally, automatic leader and/breakaway detection could also be potentially applicable off-line, in archival videos, leading to the extraction of useful metadata.

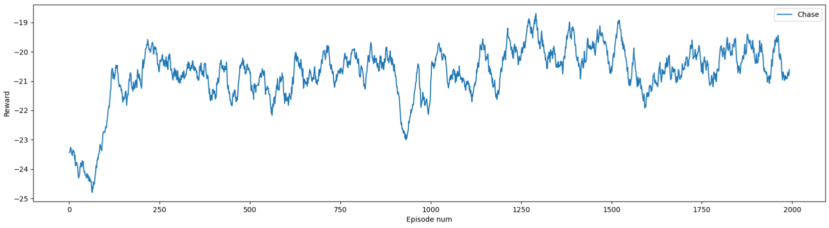


Figure 15. Rewards returned after each episode, while training for the "Ascent" CMT. For the sake of intelligibility, the actual rewards have been filtered using a moving average filter with a window size of 30.

In the context of T5.2 of AI4Media, AUTH explored these issues by extending preliminary work initiated originally during the final stage of the H2020 RIA project MULTIDRONE[1]. In AI4Media, this work was improved, evaluated and presented as a conference paper.

Proposed Method. Although there have been numerous works that study athletic performance at an individual athlete level, primarily by modeling biomechanics [120, 121, 122], there has been no research on the kinematics and dynamics of racing sports involving a group of athletes using only visual information.

The novel leader detection method proposed by AUTH uses global optical flow, in order to estimate camera motion direction. The underlying assumption is that the drone already follows the athlete group, either from above or from a lateral position, according to the chosen drone cinematography mode [100]. A visual target (object) detector and tracker is employed for finding regions of interest (ROI) of the targets (athletes) on the image plane. In the next step, the target ROI centers are projected on the median optical flow unit direction vector and, lastly, the leading target (athlete) is detected. Moreover, the athlete winning order (1st, 2nd, 3rd, etc), as well as their spatial distribution over time can be determined as well, thus providing very useful information for computational racing sports coaching.

In the context of this work, the breakaway detection problem is solved as well, by introducing additional target breakaway detection metrics, the derivatives of which can indicate the happening of a breakaway.

Leader detection performance was evaluated in a video dataset of cycling races provided by Radiotelevisione Italiana (RAI) [2]. The cyclist detector [123] was employed for target detection.

Using a dataset of 1,571 bicycle racing video frame pairs, the leader detection algorithm achieved a high leader detection accuracy of 97.2%, while in detecting the second best it achieved 95.6% accuracy. It can run in real time, as only around 24 ms are needed to process a single video frame pair on a cheap consumer graphics card.

As cyclist breakaway videos are rather scarce, the event was simulated in AirSim. If the derivatives of the proposed breakaway metrics surpass a certain threshold, it can successfully indicate the time instances when the breakaways take place, as shown in Figure 16, illustrating the first derivative of a specific proposed metric. The choice of the optimal thresholds are investigated as well. First the PDFs of the metrics in presence and in absence of breakaway are estimated using Kernel Density Estimation [124] with Gaussian Kernel, and then the optimal thresholds that minimize the detection error are opted to be the intersection point of the two PDFs (Figure 17).

Conclusions. Although the evaluation results showed that the leader and breakaway detection algorithms can yield satisfactory results, there are certain limitations that must be taken into consideration. First and foremost, the leader detection algorithm is applicable only in scenarios where the targets move on a relatively straight line. Turns produce complex optical flow behavior, such that the motion direction cannot be estimated by a single averaging.

[1] https://multidrone.eu/

[2] http://www.aiia.csd.auth.gr/LAB_PROJECTS/MULTIDRONE/AUTH_MULTIDRONE_Dataset.html

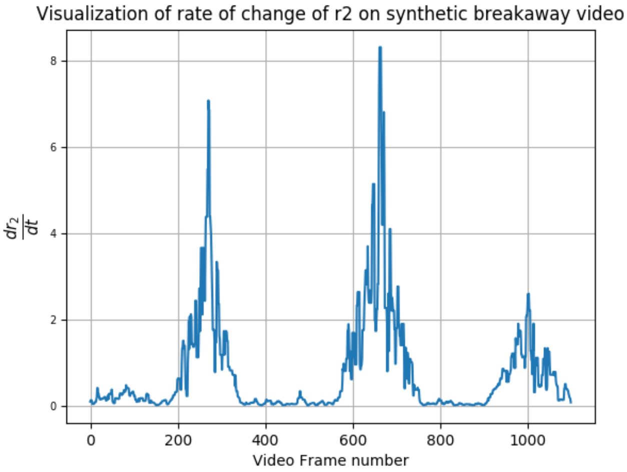


Figure 16. Plot of the first derivative of a proposed breakaway detection metric (r_2).

3.5.4. Relevant publications

- A. Sochopoulos, I. Mademlis and I.Pitas, ”Deep Reinforcement Learning with semi-expert distillation for autonomous UAV cinematography”, technical report, under preparation.
- S. Papadopoulos, I. Mademlis and I.Pitas, ”Autonomous UAV Cinematography using Deep Reinforcement Learning”, technical report, under preparation.
- S. Papadopoulos, C. Symeonidis and I. Pitas, ”Leader and breakaway detection in racing sports videos”, Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP), 2021 [125].
  Zenodo record: https://zenodo.org/record/6141654#.Yg__tpaxVEY.

3.5.5. Relevant WP8 Use Cases

3B1-3 (Video synthesis from 3D environment). Autonomous execution of UAV CMTs is essential for automated media production using intelligent shooting approaches, relying on high-level directorial guidelines. Leader and breakaway detection algorithms could aid an autonomous UAV take better decisions about cinematic coverage of a racing event.
3A3 (Archive Exploitation). Leader and breakaway detection algorithms could be applied off-line in archival videos, so that useful metadata can be extracted.

3.6. Predicting user attention in 360° videos

Contributing partners: 3IA-UCA

Context Immersive media are on the rise, with 360° videos being a major modality of Virtual Reality (VR), with developing applications in storytelling, journalism or remote education. These contents are meant to be watched in a head-mounted display where the user can explore in at least 3 Degrees of Freedom (DoF), possibly in 6DoF.

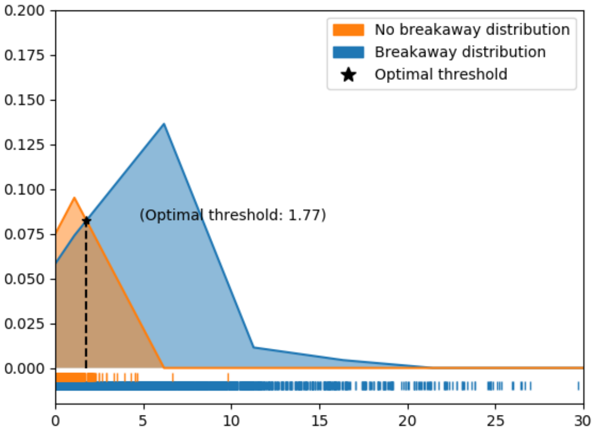


Figure 17. r_3 detection threshold.

To ensure sound content design and high Quality of Experience when accessing these contents online, it is crucial to understand and predict the user's attentional process, specifically the head movements. Indeed, the development of online 360° videos is persistently hindered by the difficulty to access immersive content through Internet streaming: owing to the closer proximity of the screen to the eye in VR and to the width of the content (2π steradians in azimuth and π in elevation angles), the data rate is two orders of magnitude that of a regular video [126]. To decrease the amount of data to stream, a solution is to send in high resolution only the portion of the sphere the user has access to at each point in time, named the Field of View (FoV). This requires to know the user's head position in advance, that is at the time of sending the content from the server. Failing to predict correctly the future user's positions can lead to a lower quality displayed in the FoV, which can impair the user's experience.

In this work, we consider the problem of predicting the user's head motion in 360° videos over a future horizon, based both and only on the past trajectory and on the video content. Various methods tackling this problem with deep neural networks have therefore been proposed (e.g., [127, 128, 129]). We show that the relevant existing methods have hidden flaws, that we thoroughly analyze to overcome with a new proposal establishing state-of-the-art performance. We next present our two contributions, focused on multimodal fusion (past trajectory coordinates and visual content) with deep architectures.

Hidden flaws of existing methods and root-cause analysis After a review and taxonomy of the relevant methods (e.g., [127], [128], [129]), we compare them to common baselines. First, comparing against the *trivial-static baseline*, we obtain the intriguing result that they all perform worse, on their exact original settings, metrics and datasets. Second, we show it is indeed possible to outperform the *trivial-static baseline* (and hence the existing methods) by designing a stronger baseline, named the *deep-position-only baseline*: it is an LSTM-based architecture considering only the positional information, while the existing methods are meant to benefit both from the history of past positions and knowledge of the video content.

From there, we carry out a thorough root-cause analysis to understand why the existing methods perform worse than baselines that do not consider the content information.

Looking into the metrics and the data, we show that: (i) evaluating only on some specific pieces of trajectories or specific videos, where the content is proved useful, does not change the comparison results, and that (ii) the content can indeed inform the head position prediction, but for prediction horizons longer than 2 to 3 sec.. All these existing methods consider shorter horizons.

Looking into the neural network architectures, we identify that: (iii) when the provided content features are the ground-truth saliency, the only architecture not degrading away from the baseline is the one with a Recurrent Neural Network (RNN) layer dedicated to the positional input, but (iv) when fed with saliency estimated from the content, the performance of this architecture degrades away from the *deep-position-only baseline* again.

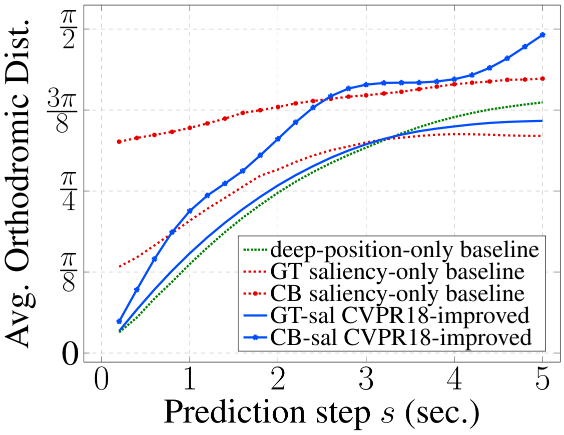


Figure 18. Prediction error of CB-sal CVPR18-improved (with Content-Based saliency) against GT-sal CVPR18-improved (with Ground-Truth saliency) and baselines.

Fig. 18 shows the expected degradation using the content-based saliency (obtained from PanoSalNet) compared with the ground-truth saliency: the CB *saliency-only baseline* (dashed red line) is much less accurate than the GT *saliency-only baseline* (solid red line). We also observe that, despite performing well with ground-truth saliency, method CVPR18-improved fed with content-based saliency degrades again away from the *deep-position-only baseline*.

Building on the above analysis and results, we build a new deep architecture able to benefit from (approximate) content-based saliency, as described next.

A new deep neural architecture achieving state-of-the-art prediction performance The fundamental characteristic of the problem at hand is: over the prediction horizon, the relative importance of both modalities (past positions and content) varies. Indeed, we expect the motion inertia to be more prominent first, and only then the content to possibly attract attention and change the course of the motion. It is therefore crucial to have a way of combining both modality features in a time-dependent manner to produce the final prediction. However, in the best-performing architecture so far, CVPR18-improved, we notice that the single RNN component enables this time-dependent modulation only for the positional features, while the importance of the content cannot be modulated over time. Replacing the ground-truth saliency with content-based saliency, the saliency map becomes much less correlated with the positions to predict. It is therefore important to be able to attenuate its effect in the first prediction steps, and give it more importance in the later prediction step.

From the latter analysis, a key architectural element to add is a RNN processing the visual features (such as CB-sal), before combining it with the positional features. Furthermore, this analysis connects with the seminal work of Jain et al., introducing Structural-RNN in [130]. It consists in casting a spatio-temporal graph describing a problem's structure into a rich RNN mixture following well-defined steps. Though the connection with head motion prediction is not direct, we can formulate our problem structure in the same terms. First, two contributing factor components are involved: the user's FoV and the video content. We can therefore express the spatio-temporal graph of a human watching a 360° video in a headset. Second, these two components are semantically different, and are therefore associated with: (i) an edgeRNN and a nodeRNN for the FoV, (ii) an edgeRNN for the video (only one input to the node), resulting in the architectural block shown in purple in Fig. 19. Embedded into a sequence-to-sequence framework, we name this architecture TRACK.

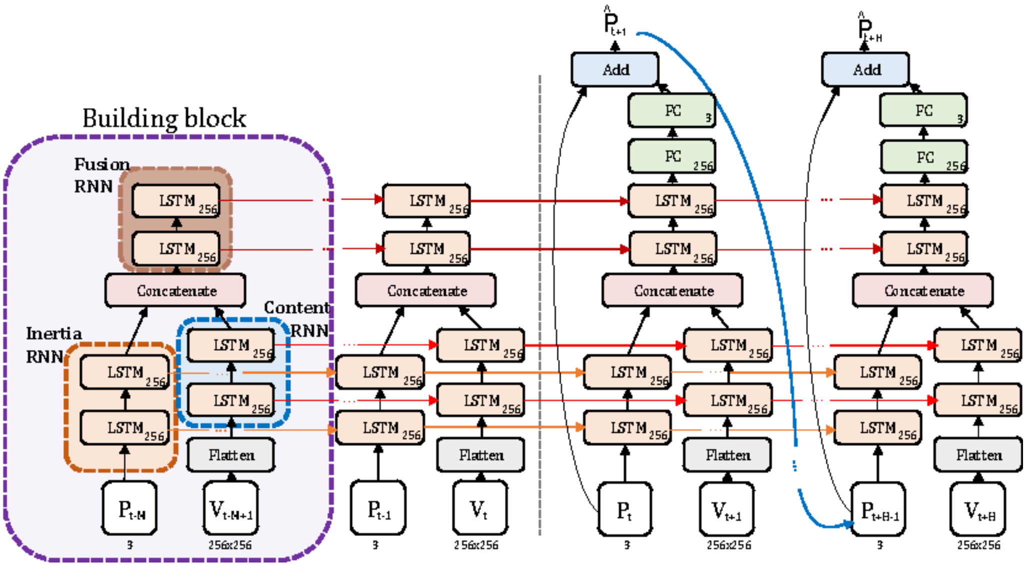


Figure 19. The proposed TRACK architecture. The building block (in purple) is made of a an Inertia RNN processing the previous position (light brown), a Content RNN processing the content-based saliency (blue) and a Fusion RNN merging both modalities (dark brown).

Results **Gains on video categories**: The results of the root cause analysis analyzing the data have shown that the gains that can be expected from a multimodal architecture over the *deep-position-only baseline* are different depending on the video category: whether it is a *focus-type* or an *exploratory* video. The results of TRACK averaged over all the videos of a test set show improvements, but are therefore not entirely representative. We analyze the gains of TRACK over different video categories, by splitting manually the 5 videos of a small test set of the MMSys18 dataset, while for CVPR18 and PAMI18, we consider the entropy of the GT saliency map of each video to assign the video to one category or the other. We sort the videos of the test set in increasing entropy, and we represent in Fig. 20 the results averaged over the bottom 10% (focus-type videos) and top 10% (exploratory videos).

- On the low-entropy/focus-type videos and for $s \geq 3$ sec., TRACK significantly outperforms the second-best method: by 16% for PAMI18 to 20% for both CVPR18 and MMSys18 at $s = H = 5$ sec. TRACK performs similarly or better for $s < 3$ sec.
- On the high-entropy/exploratory videos, the gains of TRACK are much less significant: TRACK often performs similarly or slightly worse than the *deep-position-only baseline*, yet never degrading significantly away from this baseline, as the other methods do. Such results are expected from the root cause analysis showing that the *saliency-only baseline* does not outperform the *deep-position-only baseline* on exploratory videos.

Qualitative examples: In Fig. 21, we exemplify the results on two low-entropy videos, also showing a representative frame with a user's future trajectory and the prediction of TRACK. On focus-type videos, TRACK outperforms significantly the second-best method: by up to 25% in the examples.

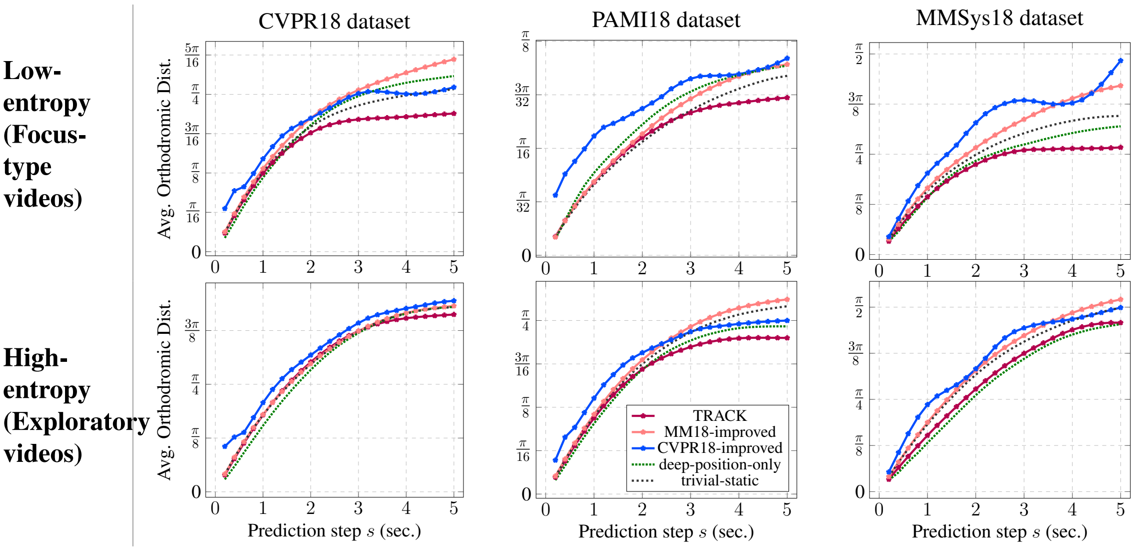


Figure 20. Top row (resp. bottom row): results averaged over the 10% test videos having lowest entropy (resp. highest entropy) of the GT saliency map. For the MMSys dataset, the sorting has been made using the Exploration/Focus categories. Legend and axis labels are the same in all figures.

Figure 21. Example of performance on two individual test videos of type Focus. On the frame, the green line represents the ground truth trajectory, and the corresponding prediction by TRACK is shown in red.

Conclusion This work has brought two main contributions. First, we carried out a critical and principled re-examination of the existing deep learning-based methods to predict head motion in 360° videos, with the knowledge of the past user's position and the video content. We have shown that all the considered existing methods are outperformed, on their datasets and with their test metrics, by baselines exploiting only the positional modality. To understand why, we have analyzed the datasets to identify how and when should the prediction benefit from the knowledge of the content. We have analyzed the neural architectures and shown there is only one whose performance does not degrade compared with the baselines, provided that ground-truth saliency information is provided, and none of the existing architectures can be trained to compete with the baselines over the 0-5 sec. horizon when the saliency features are extracted from the content.

Second, decomposing the structure of the problem and supporting our analysis with the concept of Structural-RNN, we have designed a new deep neural architecture, named TRACK. TRACK establishes state-of-the-art performance on all the prediction horizons $H \in [0 \text{ sec}, 5 \text{ sec}]$ and all the datasets of the existing competitors. In the 2-5 sec horizon, TRACK outperforms the second-best method by up to 20% on focus-type videos, i.e., videos with low-entropy saliency maps.

In future works, we will investigate deep attention mechanisms to refine the time- and space-varying fusion of modalities, as well as consider variational approaches (with VRNN) to also obtain confidence on the prediction, which is crucial for decision-making.

3.6.1. Relevant publications

- Romero-Rondon, M. F., Sassatelli, L., Aparicio-Pardo, R., & Precioso, F. (2021). TRACK: A new method from a re-examination of deep architectures for head motion prediction in 360-degree videos. IEEE Transactions on Pattern Analysis and Machine Intelligence. [131] https://zenodo.org/record/4673531

- M. F. Romero-Rondon, D. Zanca, S. Melacci, M. Gori, and L. Sassatelli, "Hemog: A white-box modelto unveil the connection between saliency information and human head motion in virtual reality," in IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), pp. 10–18, 2021. [132] https://zenodo.org/record/5563115

3.6.2. Relevant software and/or external resources

- M. Romero, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso. A Unified Evaluation Framework for Head Motion Prediction Methods in 360° Videos. ACM International Conference on Multimedia Systems (MMSys), Open Dataset and Software track, Istambul, Turkey, June 2020. Code available at https://gitlab.com/miguelfromeror/head-motion-prediction/tree/master.

3.6.3. Relevant WP8 Use Cases

3C2-9 (Management of contribution under bandwidth constraints). User gaze and head motion prediction is crucial to optimize bandwidth when streaming VR content such as 360 videos.

3.7. Interactive Content Improvement

Contributing partners: MODL

This section focuses on the application of Machine Learning (ML) techniques to model the player experience of videogames based on gameplay telemetry. Such models can help us identify player populations, understand and enhance different aspects of player experience and lead to the design of entirely new and engaging gameplay via game content generation [133]. Beyond building better games for entertainment advancements on the field, it can also help improve other domains. Serious games are used in healthcare and rehabilitation as therapeutic [134, 135] and research tools [136, 137], and in education [138, 139] where adaptive models can enhance and personalise educational content. Games can provide robust test beds for a wide spectrum of domains through a unique combination of interactivity, freedom, and constrained environments.

In this section, we are focusing on motivation prediction based on game telemetry. We focus on user behaviour as it is readily available for capture for most game developers. While affective computing applications in lab environments often rely on physiological signals, in the wild, we face more limited data. To capture dimensions of player motivation, we use the *Ubisoft Perceived Experience Questionnaire* (UPEQ) [140], which is based on the *Theory of Self-Determination* [141]. We use Preference Learning (PL) to model the captured data, which has been proved to be a robust way to process and predict human-generated data [142]. Finally, we present a web interface prototype, which we are developing in React to visualize the data and model predictions. Our preliminary analysis highlights four populations in the observed player-base with a unique behaviour and motivational profile.

Player Modelling Player Modelling (PM) is a field of Games Research that generally aims to predict either the *behaviour* (i.e. what players do) or *emotions* (i.e. how players feel) of users. The presented work threads the line between these two approaches and focuses on player motivation, using methodology from the PM field. Though common computational approaches to user analytics use k-means clustering, self-organising maps [143], matrix factorisation [144], archetypal analysis [145, 146], and sequence mining [147, 148, 149], here we are focusing specifically on the predictive modelling side of PM through supervised Machine Learning (ML) techniques. Notable applications of player modelling include predicting player behaviour [150] such as churn [151, 152, 153], playtime [154], or player experience [155, 142]. While several of these studies focus on multimodal player data that fuse gameplay with physiological data [156, 157, 158] or data from video streams [159], we rely solely on gameplay data, which is more readily available in the wild [160, 161].

Motivation is a crucial element of player experience and research, as it is closely related to engagement and—more directly—to game enjoyment [162]. A meta-review of 87 studies on the subject [162] reveals that a high level of motivation is a core component of the positive emotional valence associated with playing a given game. While the connection between motivation and enjoyment is clear, here, we focus solely on the former. In contrast to enjoyment, which often encompasses flow [163], presence and immersion [164], and fun [165], motivation is a more action-oriented concept. Because of this focus on what drives players to engage with and act in the game, the concept of motivation can be more easily adapted during development into data-driven frameworks such as adaptive and generative systems [166].

Self Determination Theory The Self Determination Theory (SDT)—developed by Deci and Ryan [167, 141]—is a general theory of motivation that focuses on the facilitation of *intrinsic motivation*. Intrinsic motivation describe a drive that is not affected by outside pressures and rewards, and it is generally associated with positive outcomes regarding one's mental well-being [168]. In contrast, *extrinsic motivation* has measurable diminishing returns and can lead to a feeling of burn-out. SDT revolves around three basic *psychological needs*—competence, autonomy, and relatedness. *Competence* describes a sense of accomplishment and mastery; *Autonomy* describes a sense of control and self-efficacy; and *Relatedness* describes a sense of belonging and connection. According to SDT, experiences—such as videogames [164]—that facilitate these psychological needs well, will foster a greater sense of motivation and fulfilment.

The Ubisoft Perceived Experience Questionnaire (UPEQ) [140] was designed to capture players' sense of competence, autonomy, and relatedness—along with spatial presence—specifically in the games domain. UPEQ extends the original three factors with presence, as it is a major facilitator of the game experience [164]. *Presence* is the illusion of an unmediated experience that in a mediated environment, such as a videogame [169, 170]. UPEQ uses 5-point Likert-scales to measure these dimensions through a 24-item survey (with 7 items for each of *competence, autonomy*, and *relatedness*, and 3 items for *presence*). The survey was developed based on the *Basic Need Satisfaction Scale(s)* [171], addressing limitations of contemporary surveys for measuring SDT in games [172, 173, 174]. UPEQ has been used in the past to successfully as input to predict solo playtime, group playtime, and money spent [140] and as output to model motivation based on gameplay [160].

Motivation Modelling In this case study, we instrumented a simple racing game example (see Figure 22) with an integrated survey, which allowed us to collect UPEQ answers and plug them directly into a streamlined pipeline that collects both game telemetry and user response seamlessly. Figure 23 shows how the survey was integrated into the game. We collect 34 telemetry measures from the game, including data on the player's velocity, collision, input controls, and distance from obstacles and bots in the scene. The collected telemetry is aggregated on the session level and describes the player's performance throughout the session.

To model the data, we chose a Preference Learning (PL) approach. PL is a ML paradigm which focuses on the relative relationships within the data [175]. We apply PL through a pairwise transformation to model

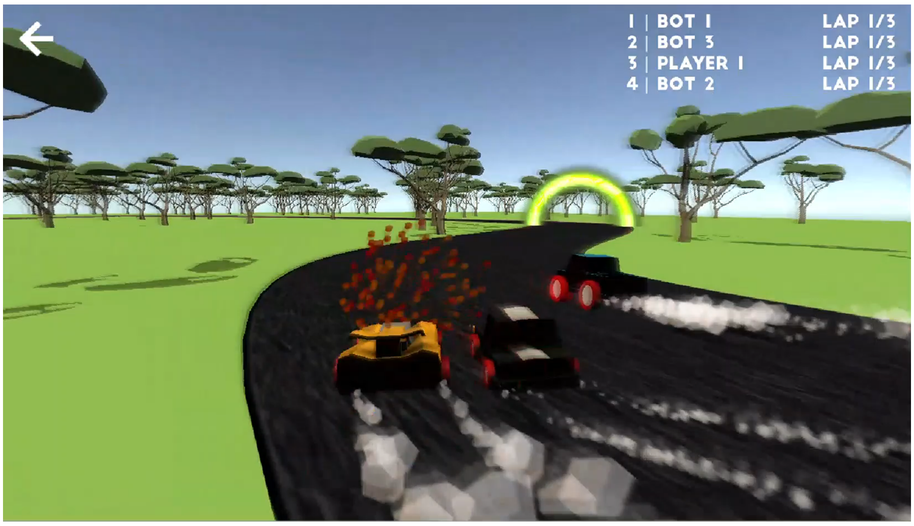
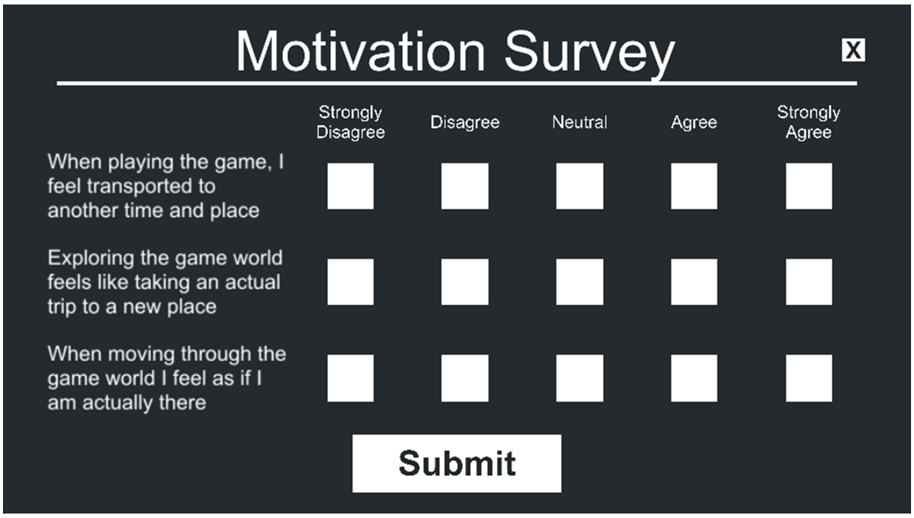


Figure 22. Simple racing game demo to collect telemetry data



Motivation Survey

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
When playing the game, I feel transported to another time and place	□	□	□	□	□
Exploring the game world feels like taking an actual trip to a new place	□	□	□	□	□
When moving through the game world I feel as if I am actually there	□	□	□	□	□

Figure 23. UPEQ survey integrated into the game.

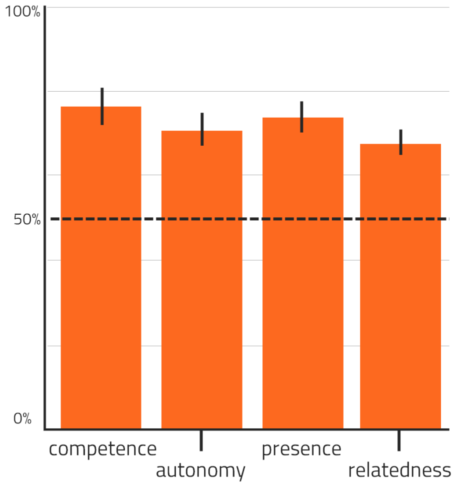
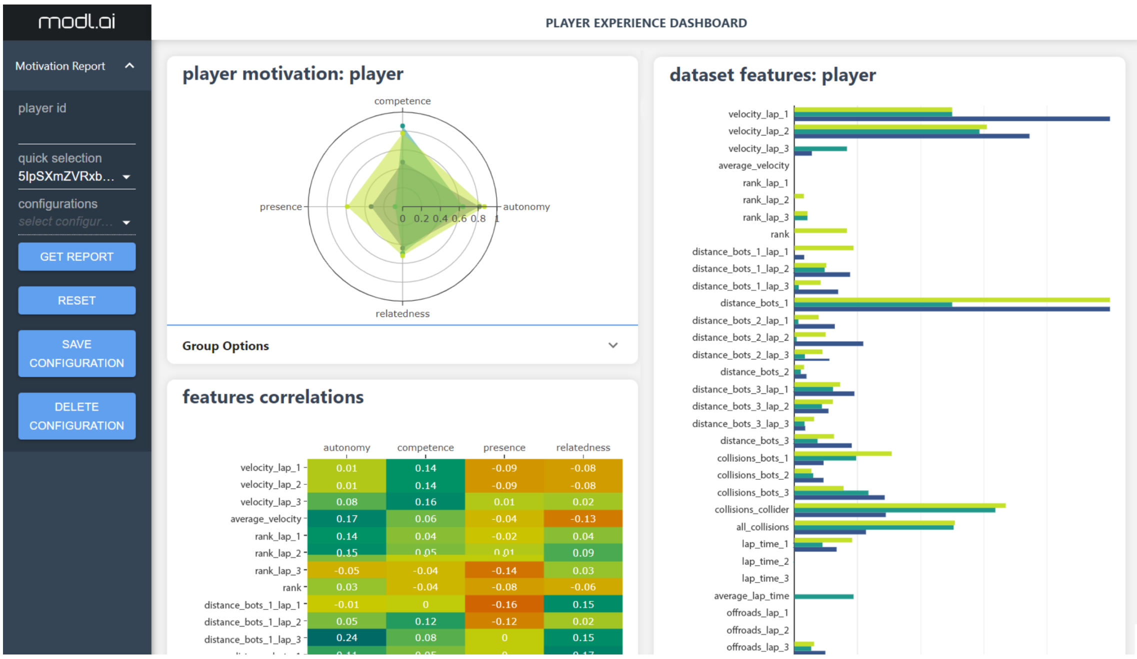


Figure 24. Preliminary results of the motivational models for each measured factor. The error bars show the 95% confidence interval.

the preference relation of data points. The *pairwise transformation* increases the number of observations in a dataset exponentially, providing more information on individual data points, while still preserving the relative relationships in the data. This provides a reliable basis for experience and motivation modelling [142]. Through this method of PL we are leveraging binary classification after the pairwise transformation of the dataset [176]. During this transformation, pairs of data inputs $(x_i, x_j) \in X$ and their corresponding outputs $(y_i, y_j) \in Y$ are observed. The observed points are discarded and two new data points are created. These points express the difference between the original data points. In case of $y_i > y_j$, x_i is preferred over x_j (formally: $x_i \succ x_j$) and we label $x_i - x_j$ as 1 and $x_j - x_i$ as -1 to signify the direction of their preference relation. Learning to predict this relation becomes a problem of binary classification. As an added benefit, because we make two observations every time, the method also creates a 50% baseline, without further data manipulation such as under- and over-sampling.

To solve the binary classification task, we used a simple Random Forest (RF) algorithm. RFs are widely used in PM contexts due to their fast training and robustness [177, 178]. When it comes to predict human data, RF algorithms can often perform comparably to deep learning methods [179]. RFs are ensemble learning methods mainly used for classification and regression. As the name suggests, RFs operate by constructing a number of randomly initialised independent decision trees during training and use their predictions as a meta output. Decision trees themselves operate by constructing an acyclical network of nodes, which split the features of the given dataset into simpler decisions [180]. In this case study, we use an implantation of RFs based on an optimised Classification And Regression Tree (CART) algorithm—first proposed by Breiman [181]. The CART method uses a generalisation of the binomial variance to evaluate the impurity (and thus splitting criterion) of nodes [182] and relies on a process of "overgrowing" and pruning trees [180] to minimise training errors without overfitting.

Figure 24 shows the preliminary results of the modelling effort. Results show that PL works reasonably well on limited datasets, with room for future improvements. The constructed models reach a good level of accuracy at Competence 76% (up to 88%), Autonomy 71% (up to 88%), Relatedness 74% (up to 87%), and Presence 68% (up to 79%). To verify the results, we also look at the Kendall's tau correlation between the predicted motivation factors and the input features. This provides a good sanity check about the survey results but also helps us understand the player population better. Based on the correlations we can identify players' motivational profiles, which more or less correspond to the motivational factors measured by UPEQ. High competence players have an early high velocity and higher standing at the end of the race. High autonomy players have more collisions and off-road travel while still maintaining high standing. High



features correlations

	autonomy	competence	presence	relatedness
velocity_lap_1	0.01	0.14	-0.09	-0.08
velocity_lap_2	0.01	0.14	-0.09	-0.08
velocity_lap_3	0.08	0.16	0.01	0.02
average_velocity	0.17	0.06	-0.04	-0.13
rank_lap_1	0.14	0.04	-0.02	0.04
rank_lap_2	0.15	0.05	0.01	0.09
rank_lap_3	-0.05	-0.04	-0.14	0.03
rank	0.03	-0.04	-0.08	-0.06
distance_bots_1_lap_1	-0.01	0	-0.16	0.15
distance_bots_1_lap_2	0.05	0.12	-0.12	0.02
distance_bots_1_lap_3	0.24	0.08	0	0.15

Figure 25. Web interface prototype to view motivation modelling results and game telemetry.

relatedness players have a larger amount of collisions and an overall smaller distance to bots. Finally, a high presence score generally correlates with novice players (low initial velocity and lower average standing).

Telemetry Visualization To provide a better overview of our results, we are developing a web interface prototype that lets us visualise the collected data and the predictions of our models. Figure 25 shows the intended layout of the web interface. The interface displays the predicted motivation profile of players, and the features associated with their performance. It also visualises the correlation between the predicted motivation rank and their telemetry, giving a fuller picture about the game's population. Although the PL task learns to predict which point is preferred between two examples, we can convert the predictions of PL algorithms into a global ranking based on repeated pairwise observations.

3.7.1. Relevant publications

- Melhart, David, Daniele Gravina, and Georgios N. Yannakakis. "Moment-to-moment Engagement Prediction through the Eyes of the Observer: PUBG Streaming on Twitch." International Conference on the Foundations of Digital Games. 2020. [183] https://dl.acm.org/doi/abs/10.1145/3402942.3402958

3.7.2. Relevant WP8 Use Cases

5A and specifically 5A3-2 (Telemetry logging). The motivation prediction models can be used to make the automated testing of AI agents (Use Case 5A, Automated Testing for Games) more human-like by modeling different user behaviours.

3.8. Synthetic Audio Generation

Contributing partners: IRCAM

The present research aims to develop innovative algorithms to generate synthetic yet realistic musical sound mixes starting from musical scores present in a digital format. First, this approach can be employed to generate music content in media or video games, and it can also have artistic applications for music composers, by rendering previews of their compositions before hiring musicians. Second, on top of the aforementioned artistic purposes, this approach is interesting for producing large datasets of realistic musical mixes from symbolic annotations [184]. Such automatically generated datasets of realistic mixes will be used to further train models for various Music Information Retrieval (MIR) tasks, such as automatic transcription, instrument identification, tempo and down-beat estimation, or key and mode recognition. To this end, we propose a neural synthesis model for realistic instrument sounds from a symbolic musical representation, that is the MIDI format.

3.8.1. Audio generation context

Many methods already exist for the synthesis of musical instrument sounds. A straightforward technique consists in concatenative samplers which playback high-fidelity recordings of isolated notes, but these approaches only provide a very limited amount of control and they cannot reproduce interactions between notes. On the contrary, physical-based modeling can achieve realistic, interpretable and controllable sound synthesis, but requires extensive modeling and precise measurements of physical components [185], which limits the application of this approach to one class of instrument at a time. Traditional signal-based methods [186, 187] are light, flexible and controllable, but often lack realism in the synthesis. Finally, black-box neural-based approaches successfully adapt text-to-speech techniques to produce realistic instrument sounds [188, 189], but they require a significant amount of annotated recordings and they are difficult to control as they do not explicitly model instrument properties.

Recently, the Differentiable Digital Signal Processing (DDSP) library [190] introduced traditional signal-based synthesis tools as differentiable output layers in a neural-based audio synthesis model. The modularity of a DDSP-based architecture enables the injection of acoustic modeling knowledge into a deep learning framework, thus imposing strong apriori on the sound structure. This would alleviate the need for large quantity of training data and reduce the number of parameters, while exploiting the expressiveness of neural networks for realistic sound synthesis. Several improvements have been made to make the approach compatible with MIDI inputs [191, 192, 193] but only in the monophonic scenario.

Our first proposed approach tackles the task of piano sound synthesis from a symbolic representation, by enhancing and adapting the DDSP framework to handle polyphonic MIDI input and to reproduce particular properties of the piano sound, such as partials inharmonicity, partials beating, and noise of the hammers, the keys and the pedals.

3.8.2. Proposed approach for piano sound synthesis

The full synthesis architecture is illustrated in Figure 26. It takes as input all the parameters that a pianist has over its instrument, being the played notes (pitches and velocities), the pedal actions, and the piano and room context. The synthesis is controlled by a neural network, and the audio signal is computed by summing the outputs of multiple monophonic additive and subtractive differentiable synthesizers. Finally, the room reverberation is produced by a learned impulse response. The following paragraphs detail the sub-modules presented in Figure 26:

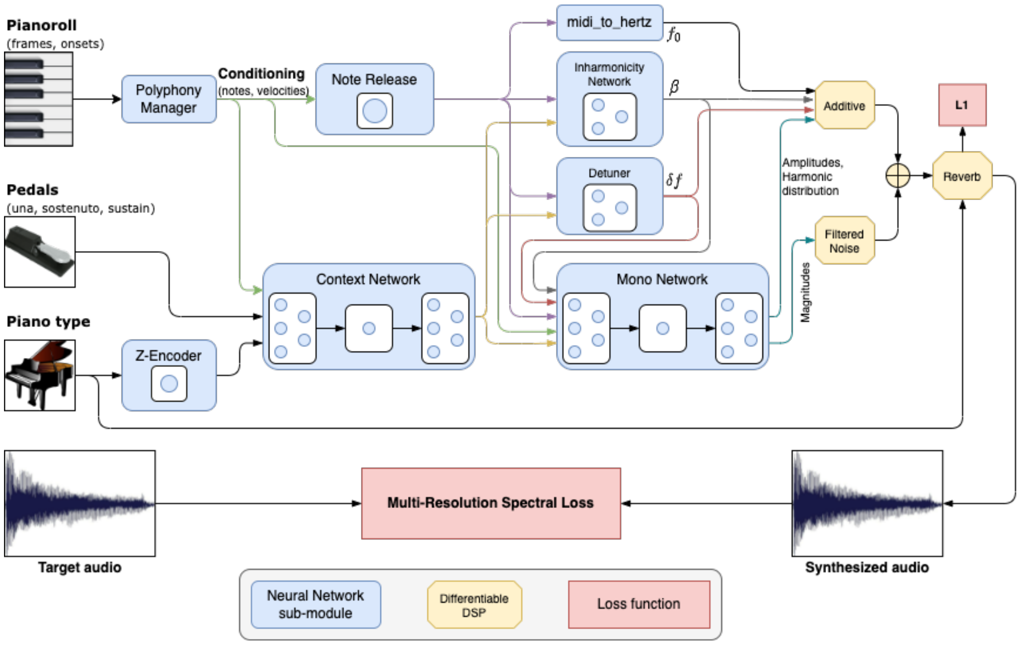


*Figure 26. Synthesizer Architecture. The blue boxes represent the trained modules for the control of the synthesis. The synthesis modules from DDSP are represented by yellow boxes (*Additive*,* Filtered Noise*, and* Reverberation*). Finally, the* Multi-Resolution Spectral Loss *compares the input* target signal *(bottom left) and the output* synthesized sound *(bottom right).*

Polyphony manager: The played notes are encoded as onsets and active pianorolls, as in [194], which informs when the notes are being played, for how long and at which velocity. For memory efficiency, the pianorolls are reduced from the whole piano tessitura (88) to a 16-channel conditioning vector by the *Polyphony Manager* sub-layer. This layer ensures that monophonic note information (pitch and velocity) are contained within a single channel in the conditioning vector. The number of simultaneous notes our model can handle is conditioned by this layer.

Note release, Inharmonicity network, & Detuner: The activity conditioning is extended by a fixed duration to let the model capture the remaining string vibration after the notes are released, since the energy absorption is not instantaneous [185]. Also, the string stiffness leads to the partials of a piano note not being pure harmonics of the fundamental frequency: such characteristic is implemented with an explicit inharmonicity model over the whole tessitura, taken from [195]. Furthermore, piano strings are often doubled or tripled for each note: they are slightly detuned from one to another, which creates partials beating [196]. A detuning factor of the fundamental frequency is computed by the *Detuner* sub-layer.

Context network: The piano model, the effects of the pedals and the interaction between simultaneous notes change the timbre of an individual note [185]. A piano model embedding is forwarded with the pedal signals and the conditioning vector to a *context network*. This recurrent network (based on GRU cells) computes a context vector, which is applied to all monophonic channels to influence the computation of the synthesizer controls.

Monophonic network: The *monophonic network* computes the remaining synthesizers' controls from the extended conditioning vector, the context vector and the notes fundamental frequencies, inharmonicity and detuning coefficients. Since it is applied channel-wise, this recurrent network learns a monophonic string model to predict the notes amplitude, harmonic energy distribution and noise magnitudes.

Differentiable synthesis: The differentiable synthesizer layers convert the controls into audio signals, in the spectral modeling paradigm [186]. The *additive synthesizer* generates the (quasi-)harmonic components of a piano note by summing multiple sinusoids at frequencies computed from the fundamental frequency, the detuning factor and the inharmonicity coefficient, and at magnitude given by the global amplitude and harmonic distribution. The *subtractive synthesizer* generates the noisy elements (hammer, key and pedal noises) by filtering a white noise with filters computed from the noise magnitudes as in [190].

Reverberation: The room acoustics of the piano recordings is modeled by a differentiable convolutional reverberation. An impulse response is learned for each room context and it is applied to the sum of audio signals from the bank of additive and subtractive synthesizers. In some experimentations, it has been observed that this module tried to reproduce the note releases, which provided an abnormal reverberation. So, a L1 regularization loss is applied on the impulse response parameters to prevent the reverberation layer from also learning the note decay and sustain behaviors, which are supposed to be learned by the *monophonic network*.

Multi-resolution spectral loss: The final audio output is compared to the ground-truth audio with a multi-resolution spectral loss, as in [188, 190], and the whole model is trained with the Adam optimizer. Piano recordings and their corresponding MIDI performances of the MAESTRO dataset [189] were used for training, and to be comparable with a pure DNN baseline [188], but realistic results have been also obtained on the smaller MAPS dataset [197].

3.8.3. Results

An example of audio synthesis from the proposed model is shown in Figure 27 against the ground-truth audio. The MIDI input of the model corresponds to the recorded performance of the target audio, which was not seen during training. One can notice that the frequency distribution and amplitude decays of the notes partials (horizontal lines) are well reproduced by the model. The residual noise is also globally matched, with the exception of noisy events that do not come from the piano itself (such as the creak coming from the hall happening at around 1.5 seconds).

Listening tests are currently being conducted to compare the synthesis quality of our model against samplers, a physical-based model and a black-box DNN model. The proposed approach and results are to be then submitted at a conference in the near future.

3.8.4. Relevant WP8 Use Cases

5B (Improved music analysis and synthesis for Games). The developed approach is directly related to 5B1 (Sound synthesis of background audio track) of the use-case UC5 (AI for Games). 6A1 (Automatic Composition), 6C1 (Real-time automatic music generation. The approach can in general help in the creation of original music.

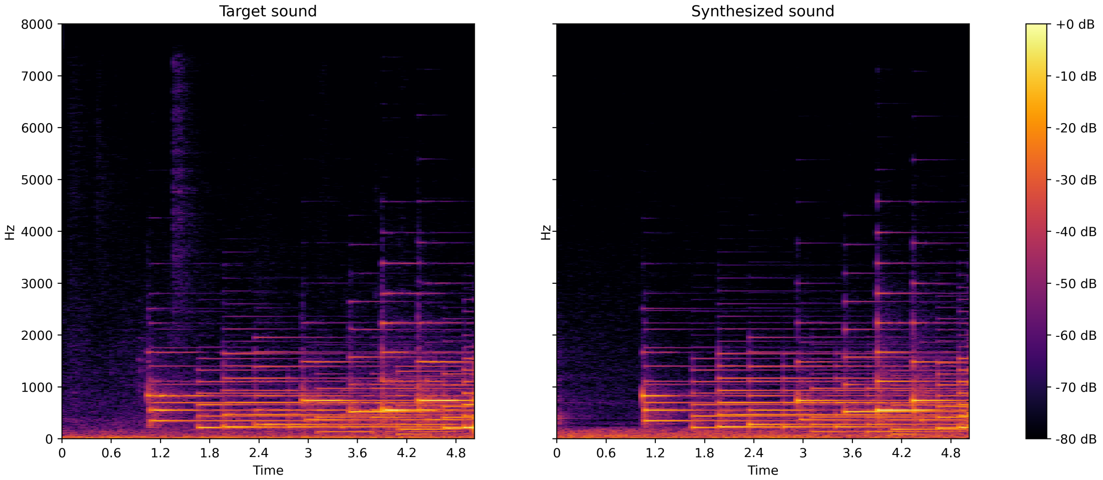


Figure 27. Spectrograms of a target piano recording (left) and the synthesized audio by the proposed model from the corresponding MIDI performance (right). The creak happening at around 1.5 seconds in the target audio is not coming from the piano, which our model cannot reproduce since it has no correlation with the performance input.

4. Contributions to the AI4Media WP8 Use Cases

Each contribution of this deliverable is mapped to a use case in WP8. WP8 operates 7 use cases, which defined their user needs for AI functions in the format of user stories. The user stories (described in detail in deliverable D8.1) were grouped into ”Features” (sub use cases) and ”Epics” (groups of related user stories). Each Epic has its own ID (e.g. Epic 3A3-11) (see Table 11).

Table 11. WP8 Use case Epics in which research from partners in T5.2 will contribute to.

WP8 Use Case Epics			
Use Cases	**Features**	**Epic ID**	**Requirement Title**
AI in Vision	Archive exploitation	3A3-11	Visual indexing and search
	Environment Investigation	3B1-3	Video synthesis from 3D environment
	Archive valorisation	3B2-1	Video super resolution
	Just-in-time content creation & adaptation	3C2-8	Synthetic Video Generation from Single Semantic Label Map
	Management of unexpected event occurrence	3C2-9	Management of contribution under bandwidth constraints
AI for Games	Automated testing for Games	5A3-2	Telemetry logging
	Improved music analysis and synthesis for Games	5B1	Sound synthesis of background audio track
AI for Human Co-Creation	Automatic Music Generation	6A1	Automatic composition
	Real-time automatic content generation	6C1	Real-time music generation

5. Ongoing Work and Conclusions

5.1. Conclusion and Future Work

This document presented the outcomes of AI4Media research activities in Task T5.2 for the period M1-M18 of the project. The majority of the work presented deals with approaches to create or enhance data, thus producing new media, relying heavily on deep learning approaches. Notably, generative models, such as GANs, are often exploited for a variety of modalities, from videos to audio. Instead of relying on machine learning to directly produce or alter videos, we also worked on strategies to automatize the shot of sequences exploiting drones - in this case Deep Reinforcement Learning has been the tool of choice. All in all, this task produced a high amount of relevant publications, totalling 2 top-tier journal publications and 7 conference publications. Moreover, we also shared 4 open source software implementations, which will also be provided for exploitation in use cases. Regarding the AI4Media use cases, all of the proposed approaches are mapped to one or more use cases and requirements, as reported in every section. Below, we briefly summarize the ongoing work associated with each partner in this task.

UNIFI UNIFI has been extending the research on trajectory prediction in order to exploit social cues. This is a relevant direction for automated cinematography and in general improves over our previous work for multiple trajectory prediction, which we developed in conjunction with 3IA-UCA. The current approach is not trained end-to-end and the memory can not be used for reasoning. To this day, no end-to-end memory augmented network for multiple trajectory prediction has been developed.

UNIFI is also keeping active its line of work on image and video enhancement for reuse of media archives. In particular, we are extending our previous works [198, 199], in order to exploit the streaming nature of videos, by leveraging the presence of sporadic high quality frames that can be transmitted. Moreover, we are working towards restoration networks that do not require prior knowledge of the distortion.

3IA-UCA 3IA-UCA has been working on the problem of head motion prediction in 360° videos and has contributed a deep prediction model based on a sequence-to-sequence framework that was able to efficiently fuse both positional and visual input modalities, as presented in section 3.6. However, legacy regression approaches are only partly able to consider the intrinsic uncertainty of the human movement. To date, no prediction method foreseeing multiple possible future trajectories exists for predicting head or gaze movements of users of 360° videos. The immediate future work is hence to design uncertainty-aware prediction models, capable of generating diverse plausible future trajectories, and the respective likelihood of each. To do so, a collaboration has been initiated between UNIFI and 3IA-UCA, as UNIFI has recent prominent contributions on the use of Memory Augmented Networks for multiple trajectory prediction for autonomous vehicles. We are going to investigate such types of approaches, and compare with diverse solutions relying on the broad family of dynamic variational auto-encoders.

UNITN In the near future, UNITN will concentrate on a new Attention-Guided Generative Adversarial Network (AttentionGAN) for the unpaired image-to-image translation task. AttentionGAN should be able to identify the most discriminative foreground objects and minimize the change of the background. At the same time, attention-guided generators in AttentionGAN should be able to produce attention masks, and then fuse the generation output with the attention masks to obtain high-quality target images. Accordingly, we will also develop a novel attention-guided discriminator which only considers attended regions. We are currently preparing a manuscript to be submitted to IEEE Transactions on Neural Networks and Learning Systems.

We have also started to work on an implicit style function (ISF) to straightforwardly achieve multi-modal and multi-domain image-to-image translation from pre-trained unconditional generators. The ISF manipulates the semantics of a latent code to ensure that the image generated from the manipulated code lies in the desired visual domain. Our preliminary results on human faces and animal image manipulations show significantly improved results over the baselines. The proposed model enables cost-effective multimodal unsupervised image-to-image translations at high resolution using pre-trained unconditional GANs. We are currently preparing a manuscript to be submitted to IEEE Transactions on Multimedia.

MODL We are expanding on our preliminary work, both in terms of modelling and visualization. We are working on developing our telemetry visualization interface into an interactive platform that allows for easy access to player data. The users will have access to a statistical summary alongside predicted motivation scores and predicted user profile. We are hoping to provide value and be able to integrate into industrial quality assurance workflows with the final interface. On the modelling side, we are investigating methods to process and model the data, and post-process predictions to provide a clearer picture about the users' motivational drives. We are looking into cleaning and transformation techniques to prepare data for preference learning and investigating different modelling methods, which might yield more robust models. Finally, we are investigating post-processing methods to format the results in a way that is easier to interpret. A common issue with pairwise preference learning is the disconnect between prediction and usage. This type of machine learning models learn a pairwise function that requires two inputs to compare, however, in the wild we often want to see point-wise predictions. We are planning to address this limitation with post-processing of our results.

AUTH We plan to finish currently on-going work about autonomous UAV cinematography, by extending the developed Deep Reinforcement Learning (DRL) methods to all target-tracking Camera Motion Types (CMTs) and improving upon them (e.g., by inserting reward terms for smooth drone trajectories). After preparing and submitting the planned journal papers, we intend to explore DRL for *high-level UAV cinematography planning*. That is, to design, implement and evaluate a DRL agent that takes artistic decisions about the optimal sequence of CMTs in the footage that is about to be filmed, instead of simply planning on-the-fly UAV paths that facilitate the vehicle to execute a desired, prespecified CMT.

IRCAM Using our work on differentiable piano sound synthesis from symbolic musical representation, we will develop a performance generation model that introduces into *simplified* piano music scores small variations of velocity and note placement in time, for example, in way that it sounds more *human*. The model will take inspiration from CycleGANs [37] to generate realistic symbolic performances without aligned music scores, by comparing real performance recordings with synthetic performances rendered with our differentiable piano model. Adding this performance generation models for other instruments [193, 200], we will be able to synthetically generate realistic musical mixes from symbolic music where MIR annotations are easier to extract.

References

[1] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *NeurIPS*, 2017.

[2] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable gans for pose-based human image generation," in *CVPR*, 2018.

[3] M. Wang, G.-Y. Yang, R. Li, R.-Z. Liang, S.-H. Zhang, P. M. Hall, and S.-M. Hu, "Example-guided style-consistent image synthesis from semantic labeling," in *CVPR*, 2019.

[4] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *CVPR*, 2020.

[5] F. Vaccaro, M. Bertini, T. Uricchio, and A. Del Bimbo, "Fast video visual quality and resolution improvement using sr-unet," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1221–1229, 2021.

[6] L. Galteri, L. Seidenari, P. Bongini, B. Marco, and A. Del Bimbo, "Language based image quality assessment," in *Multimedia Asia*, ACM, 2021.

[7] X. foundation, "Derf's collection." https://media.xiph.org/video/derf/. [Online; last time accessed 12-March-2021].

[8] Q. Huang, H. Wang, S. C. Lim, H. Y. Kim, S. Y. Jeong, and C. . J. Kuo, "Measure and prediction of hevc perceptually lossy/lossless boundary qp values," in *2017 Data Compression Conference (DCC)*, pp. 42–51, 2017.

[9] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] L. Galteri, L. Seidenari, M. Bertini, T. Uricchio, and A. Del Bimbo, "Fast video quality enhancement using gans," in *Proc. of ACM Multimedia*, MM '19, pp. 1065–1067, 2019.

[11] V. Dewil, J. Anger, A. Davy, T. Ehret, G. Facciolo, and P. Arias, "Self-supervised training for blind multi-frame video denoising," in *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2724–2734, January 2021.

[12] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. of IEEE Computer Vision and Pattern Recognition*, 2016.

[13] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of IEEE Computer Vision and Pattern Recognition*, vol. abs/1609.04802, 2017.

[14] H. Ko, D. Y. Lee, S. Cho, and A. C. Bovik, "Quality prediction on deep generative images," *IEEE Transactions on Image Processing*, vol. 29, pp. 5964–5979, 2020.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of NIPS*, 2014.

[16] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. of CVPR*, 2019.

[17] T. Park, M. Liu, T. Wang, and J. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. of CVPR*, 2019.

[18] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *CoRR*, vol. abs/1606.03498, 2016.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. of NIPS*, 2017.

[21] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?," in *Proc. of ECCV*, September 2018.

[22] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41 – 65, 2019.

[23] P. Bongini, R. Del Chiaro, A. D. Bagdanov, and A. Del Bimbo, "Gada: Generative adversarial data augmentation for image quality assessment," in *International Conference on Image Analysis and Processing*, pp. 214–224, Springer, 2019.

[24] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. of CVPR*, 2018.

[25] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10578–10587, 2020.

[26] L. Galteri, L. Seidenari, P. Bongini, M. Bertini, and A. Del Bimbo, "Language based image quality assessment," in *ACM Multimedia Asia*, MMAsia '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[27] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *ICLR*, 2018.

[28] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019.

[29] B. Dolhansky and C. Canton Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *CVPR*, 2018.

[30] J. Zhang, M. Sun, J. Chen, H. Tang, Y. Yan, X. Qin, and N. Sebe, "Gazecorrection: Self-guided eye manipulation in the wild using self-supervised generative adversarial networks," *arXiv preprint:1906.00805*, 2019.

[31] J. Song, Y. Guo, L. Gao, X. Li, A. Hanjalic, and H. T. Shen, "From deterministic to generative: Multimodal stochastic rnns for video captioning," *IEEE TNNLS*, vol. 30, no. 10, pp. 3047–3058, 2018.

[32] X. Xu, H. Lu, J. Song, Y. Yang, H. T. Shen, and X. Li, "Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval," *IEEE TCYB*, vol. 50, no. 6, pp. 2400–2413, 2019.

[33] X. Xu, T. Wang, Y. Yang, L. Zuo, F. Shen, and H. T. Shen, "Cross-modal attention with semantic consistence for image-text matching," *IEEE TNNLS*, 2020.

[34] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[35] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018.

[36] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019.

[37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, October 2017.

[38] Z. Yi, H. Zhang, P. T. Gong, *et al.*, "Dualgan: Unsupervised dual learning for image-to-image translation," in *ICCV*, 2017.

[39] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018.

[40] H. Tang, W. Wang, D. Xu, Y. Yan, and N. Sebe, "Gesturegan for hand gesture-to-gesture translation in the wild," in *ACM MM*, 2018.

[41] K. Regmi and A. Borji, "Cross-view image synthesis using conditional gans," in *CVPR*, 2018.

[42] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *NeurIPS*, 2016.

[43] F. Qiao, N. Yao, Z. Jiao, Z. Li, H. Chen, and H. Wang, "Geometry-contrastive generative adversarial network for facial expression synthesis," *arXiv preprint:1802.01822*, 2018.

[44] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan, "Geometry guided adversarial facial expression synthesis," in *ACM MM*, 2018.

[45] K. Regmi and A. Borji, "Cross-view image synthesis using geometry-guided conditional gans," *Elsevier CVIU*, vol. 187, p. 102788, 2019.

[46] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *CVPR*, 2019.

[47] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation," in *CVPR*, 2020.

[48] H. Tang, X. Qi, D. Xu, P. H. Torr, and N. Sebe, "Edge guided gans with semantic preserving for semantic image synthesis," *arXiv preprint arXiv:2003.13898*, 2020.

[49] H. Tang and N. Sebe, "Total generate: Cycle in cycle generative adversarial networks for generating human faces, hands, bodies, and natural scenes," *IEEE Transactions on Multimedia*, 2021.

[50] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[51] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. of NIPS*, pp. 2234–2242, 2016.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[53] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *CVPR*, 2018.

[54] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, "Stochastic latent residual video prediction," *ArXiv*, vol. abs/2002.09219, 2020.

[55] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *ArXiv*, vol. abs/1804.01523, 2018.

[56] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR*, 2016.

[57] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research (JMLR)*, vol. 15, no. 1, pp. 1929–1958, 2014.

[58] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[59] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating the future by watching unlabeled video," *arXiv preprint arXiv:1504.08023*, vol. 2, 2015.

[60] S. W. Kim, Y. Zhou, J. Philion, A. Torralba, and S. Fidler, "Learning to Simulate Dynamic Environments with GameGAN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[61] M. S. Nunes, A. Dehban, P. Moreno, and J. Santos-Victor, "Action-conditioned benchmarking of robotic video prediction models: a comparative study," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8316–8322, IEEE, 2020.

[62] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in neural information processing systems (NIPS)*, pp. 2863–2871, 2015.

[63] Y. Wand, P. Bilinski, F. Bremond, and A. Dantcheva, "Imaginator: Conditional spatio-temporal gan for video generation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[64] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[65] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *CoRL*, 2017.

[66] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *AAAI*, pp. 3215–3222, 2018.

[67] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[68] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 6626–6637, 2017.

[69] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, “Towards accurate generative models of video: A new metric & challenges,” *arXiv preprint arXiv:1812.01717*, 2018.

[70] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. of NIPS*, 2015.

[71] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “First order motion model for image animation,” in *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019.

[72] A. Clark, J. Donahue, and K. Simonyan, “Efficient video generation on complex datasets,” *CoRR*, vol. abs/1907.06571, 2019.

[73] P. Luc, A. Clark, S. Dieleman, D. Casas, Y. Doron, A. Cassirer, and K. Simonyan, “Transformation-based adversarial video prediction on large-scale data,” *ArXiv*, vol. abs/2003.04035, 2020.

[74] D. Weissenborn, O. Täckström, and J. Uszkoreit, “Scaling autoregressive video models,” in *International Conference on Learning Representations (ICLR)*, 2020.

[75] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 29, pp. 2172–2180, 2016.

[76] Y. Kim, S. Nam, I. Cho, and S. J. Kim, “Unsupervised keypoint learning for guiding class-conditional video prediction,” in *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[77] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, “Deep cocktail network: Multi-source unsupervised domain adaptation with category shift,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3964–3973, 2018.

[78] J. L. Fleiss, “Measuring nominal scale agreement among many raters.,” *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[79] J. L. Fleiss, B. Levin, M. C. Paik, *et al.*, “The measurement of interrater agreement,” *Statistical methods for rates and proportions*, vol. 2, no. 212-236, pp. 22–23, 1981.

[80] W. Menapace, S. Lathuilière, S. Tulyakov, A. Siarohin, and E. Ricci, “Playable video generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10061–10070, 2021.

[81] S. Worley, “A cellular texture basis function,” in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1996.

[82] S. Gustavson, “Simplex noise demystified,” *Linköping University, Linköping, Sweden, Research Report*, 2005.

[83] T. Archer, “Procedurally generating terrain,” in *Proceedings of the Annual Midwest Instruction and Computing Symposium*, 2011.

[84] G. Miller, “The definition and rendering of terrain maps,” in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1986.

[85] J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 111–119, 2010.

[86] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011.

[87] X. Mei, P. Decaudin, and B.-G. Hu, "Fast hydraulic erosion simulation and visualization on GPU," in *Proceedings of the Pacific Conference on Computer Graphics and Applications*, IEEE, 2007.

[88] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014.

[89] J. Klein, S. Hartmann, M. Weinmann, and D. L. Michels, "Multi-scale terrain texturing using Generative Adversarial Networks," in *Proceedings of the International Conference on Image and Vision Computing New Zealand (IVCNZ)*, IEEE, 2017.

[90] R. J. Spick, P. Cowling, and J. A. Walker, "Procedural generation using spatial GANs for region-specific learning of elevation data," in *Proceedings of the IEEE Conference on Games (CoG)*, 2019.

[91] C. Beckham and C. Pal, "A step towards procedural terrain generation with GANs," *arXiv preprint arXiv:1707.03383*, 2017.

[92] R. Spick and J. Walker, "Realistic and textured terrain generation using GANs," in *Proceedings of the European Conference on Visual Media Production (CVMP)*, 2019.

[93] E. Panagiotou and E. Charou, "Procedural 3D terrain generation using Generative Adversarial Networks," *arXiv preprint arXiv:2010.06411*, 2020.

[94] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[95] www.geonames.org, "https://www.geonames.org," 2018.

[96] E. Cheng, N. Xie, H. Ling, P. R. Bakic, A. Maidment, and V. Megalooikonomou, "Mammographic image classification using histogram intersection," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010.

[97] S. Sural, A. Vadivel, and A. K. Majumdar, "Histogram generation from the HSV color space," in *Encyclopedia of Information Science and Technology, First Edition*, pp. 1333–1337, IGI Global, 2005.

[98] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of GIST descriptors for web-scale image search," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2009.

[99] G. Voulgaris, I. Mademlis, and I. Pitas, "Procedural terrain generation using generative adversarial networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, IEEE, 2021.

[100] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, "Autonomous UAV cinematography: A tutorial and a formalized shot-type taxonomy," *ACM Computing Surveys*, vol. 52, pp. 1–33, 09 2019.

[101] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," 2015.

[102] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–1, 01 2019.

[103] N. Passalis and A. Tefas, "Deep reinforcement learning for frontal view person shooting using drones," in *Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2018.

[104] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.

[105] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," 2016.

[106] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, 09 2015.

[107] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013.

[108] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," 2016.

[109] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.

[110] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, pp. 1587–1596, PMLR, 2018.

[111] S. Thrun and A. Schwartz, "Issues in using function approximation for reinforcement learning," in *Proceedings of the Fourth Connectionist Models Summer School*, pp. 255–263, Hillsdale, NJ, 1993.

[112] Y. Dang, C. Huang, P. Chen, R. Liang, X. Yang, and K.-T. Cheng, "Imitation learning-based algorithm for drone cinematography system," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[113] A. Tzimas, N. Passalis, and A. Tefas, "Leveraging deep reinforcement learning for active shooting under open-world setting," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2020.

[114] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "Uav coverage path planning under varying power constraints using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1444–1449, IEEE, 2020.

[115] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "Shot type constraints in uav cinematography for autonomous target tracking," *Information Sciences*, vol. 506, pp. 273–294, 2020.

[116] K. C. Goh, R. B. Ng, Y.-K. Wong, N. J. Ho, and M. C. Chua, "Aerial filming with synchronized drones using reinforcement learning," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18125–18150, 2021.

[117] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, “Memory-based control with recurrent neural networks,” *arXiv preprint arXiv:1512.04455*, 2015.

[118] D. R. Song, C. Yang, C. McGreavy, and Z. Li, “Recurrent deterministic policy gradient method for bipedal locomotion on rough terrain challenge,” in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 311–318, IEEE, 2018.

[119] L. Meng, R. Gorbet, and D. Kulić, “Memory-based deep reinforcement learning for pomdp,” *arXiv preprint arXiv:2102.12344*, 2021.

[120] A. Drory, “Computer vision and machine learning for biomechanics applications: Human detection, pose and shape estimation and tracking in unconstrained environment from uncalibrated images, videos and depth.,” 2017.

[121] K. A. Lemmink, S. Morgan, J. Sampaio, and D. Saupe, “Computer science in high performance sport-applications and implications for professional coaching (dagstuhl seminar 13272),” in *Dagstuhl Reports*, vol. 3, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[122] S. L. Colyer, M. Evans, D. P. Cosker, and A. I. Salo, “A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system,” *Sports medicine-open*, vol. 4, p. 24, 2018.

[123] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, “Embedded UAV real-time visual object detection and tracking,” in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.

[124] M. Rosenblatt, “Remarks on some nonparametric estimates of a density function,” *The Annals of Mathematical Statistics*, pp. 832–837, 1956.

[125] S. Papadopoulos, C. Symeonidis, and I. Pitas, “Leader and breakaway detection in racing sports videos,” in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2021.

[126] J. Park, P. A. Chou, and J.-N. Hwang, “Rate-utility optimized streaming of volumetric media for augmented reality,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 149–162, 2019.

[127] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, “Predicting head movement in panoramic video: A deep reinforcement learning approach,” *IEEE Trans. on PAMI*, 2018.

[128] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360° immersive videos,” in *IEEE CVPR*, pp. 5333–5342, 2018.

[129] A. Nguyen, Z. Yan, and K. Nahrstedt, “Your attention is unique: Detecting 360° video saliency in head-mounted display for head movement prediction,” in *ACM Int. Conf. on Multimedia*, pp. 1190–1198, 2018.

[130] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-RNN: Deep learning on spatio-temporal graphs,” in *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

[131] M. F. Romero-Rondon, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso, “Track: A new method from a re-examination of deep architectures for head motion prediction in 360-degree videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[132] M. F. Romero-Rondon, D. Zanca, S. Melacci, M. Gori, and L. Sassatelli, “Hemog: A white-box model to unveil the connection between saliency information and human head motion in virtual reality,” in *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 10–18, 2021.

[133] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018.

[134] C. Holmgård, G. N. Yannakakis, K.-I. Karstoft, and H. S. Andersen, “Stress detection for ptsd via the startlemart game,” in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 523–528, IEEE, 2013.

[135] G. N. Yannakakis, “Enhancing health care via affective computing,” *Malta Journal of Health Sciences*, 2018.

[136] R. El Kaliouby, R. Picard, and S. Baron-Cohen, “Affective computing and autism,” *Annals of the New York Academy of Sciences*, vol. 1093, no. 1, pp. 228–248, 2006.

[137] J. Han, X. Li, L. Xie, J. Liu, F. Wang, and Z. Wang, “Affective computing of childern with authism based on feature transfer,” in *Proceedings of the IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 845–849, IEEE, 2018.

[138] C.-H. Wu, Y.-M. Huang, and J.-P. Hwang, “Review of affective computing in education/learning: Trends and challenges,” *British Journal of Educational Technology*, vol. 47, no. 6, pp. 1304–1323, 2016.

[139] E. Yadegaridehkordi, N. F. B. M. Noor, M. N. B. Ayub, H. B. Affal, and N. B. Hussin, “Affective computing in education: A systematic review and future research,” *Computers & Education*, vol. 142, p. 103649, 2019.

[140] A. Azadvar and A. Canossa, “Upeq: ubisoft perceived experience questionnaire: a self-determination evaluation tool for video games,” in *Proceedings of the International Conference on the Foundations of Digital Games (FDG)*, ACM, 2018.

[141] R. M. Ryan and E. L. Deci, “Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.,” *American Psychologist*, vol. 55, no. 1, p. 68, 2000.

[142] G. N. Yannakakis, R. Cowie, and C. Busso, “The ordinal nature of emotions: An emerging approach,” *IEEE Transactions on Affective Computing*, 2018.

[143] A. Drachen, A. Canossa, and G. N. Yannakakis, “Player modeling using self-organization in Tomb Raider: Underworld,” in *Proceedings of the Symposium on Computational Intelligence and Games (CIG)*, pp. 1–8, IEEE, 2009.

[144] C.-U. Lim, A. Liapis, and D. F. Harrell, “Discovering social and aesthetic categories of avatars: A bottom-up artificial intelligence approach using image clustering,” in *Proceedings of the International Joint Conference of DiGRA and FDG*, 2016.

[145] C. Bauckhage, A. Drachen, and R. Sifa, “Clustering game behavior data,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 3, pp. 266–278, 2015.

[146] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, “Guns, swords and data: Clustering of player behavior in computer games in the wild,” in *Proceedings of the Conference on Computational Intelligence and Games (CIG)*, pp. 163–170, 2012.

[147] H. P. Martínez and G. N. Yannakakis, "Mining multimodal sequential patterns: a case study on affect detection," in *Proceedings of the International Conference on Multimodal Interfaces*, pp. 3–10, ACM, 2011.

[148] G. Wallner, "Sequential analysis of player behavior," in *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp. 349–358, ACM, 2015.

[149] S. Makarovych, A. Canossa, J. Togelius, and A. Drachen, "Like a dna string: Sequence-based player profiling in Tom Clancy's The Division," in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, York, 2018.

[150] S. C. Bakkes, P. H. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.

[151] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn prediction for high-value players in casual social games," in *Proceedings of the Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, 2014.

[152] Á. Periáñez, A. Saas, A. Guitart, and C. Magne, "Churn prediction in mobile social games: towards a complete assessment using survival ensembles," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 564–573, 2016.

[153] M. Viljanen, A. Airola, J. Heikkonen, and T. Pahikkala, "Playtime measurement with survival analysis," *IEEE Transactions on Games*, vol. 10, no. 2, pp. 128–138, 2018.

[154] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, "Predicting player behavior in Tomb Raider: Underworld," in *Proceedings of the Symposium on Computational Intelligence and Games (CIG)*, pp. 178–185, IEEE, 2010.

[155] K. Makantasis, A. Liapis, and G. N. Yannakakis, "From pixels to affect: a study on games and player experience," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 1–7, IEEE, 2019.

[156] H. P. Martínez, M. Garbarino, and G. N. Yannakakis, "Generic physiological features as predictors of player experience," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 267–276, Springer, 2011.

[157] V. Georges, F. Courtemanche, M. Fredette, P.-M. Léger, and S. Sénécal, "Developing personas based on physiological measures," in *Proceedings of the International Conference on Physiological Computing Sysytems*, 2018.

[158] E. Camilleri, G. N. Yannakakis, and A. Liapis, "Towards general models of player affect," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 333–339, 2017.

[159] N. Shaker, S. Asteriadis, G. N. Yannakakis, and K. Karpouzis, "Fusing visual and behavioral cues for modeling user experience in games," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1519–1531, 2013.

[160] D. Melhart, A. Azadvar, A. Canossa, A. Liapis, and G. N. Yannakakis, "Your gameplay says it all: Modelling motivation in Tom Clancy's The Division," in *Proceedings of the IEEE Conference on Games (CoG)*, 2019.

[161] R. P. Prager, L. Troost, S. Brüggenjürgen, D. Melhart, G. Yannakakis, and M. Preuss, “An experiment on game facet combination,” in *Proceedings of the IEEE Conference on Games (CoG)*, 2019.

[162] E. D. Mekler, J. A. Bopp, A. N. Tuch, and K. Opwis, “A systematic review of quantitative studies on the enjoyment of digital entertainment games,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’14, (New York, NY, USA), p. 927–936, Association for Computing Machinery, 2014.

[163] M. Csikszentmihalyi and I. S. Csikszentmihalyi, *Optimal experience: Psychological studies of flow in consciousness*. Cambridge university press, 1992.

[164] S. Rigby and R. M. Ryan, *Glued to games: How video games draw us in and hold us spellbound*. Praeger, 2011.

[165] R. Koster, *Theory of fun for game design*. ” O’Reilly Media, Inc.”, 2013.

[166] J. Togelius and G. N. Yannakakis, “General general game AI,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, IEEE, 2016.

[167] E. L. Deci, R. J. Vallerand, L. G. Pelletier, and R. M. Ryan, “Motivation and education: The self-determination perspective,” *Educational Psychologist*, vol. 26, no. 3-4, pp. 325–346, 1991.

[168] E. Deci and R. M. Ryan, *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media, 1985.

[169] M. Lombard and T. Ditton, “At the heart of it all: The concept of presence,” *Journal of Computer-Mediated Communication*, vol. 3, no. 2, 1997.

[170] D. Melhart, “Towards a comprehensive model of mediating frustration in videogames,” *Game Studies*, vol. 18, no. 1, 2018.

[171] B. Chen, M. Vansteenkiste, W. Beyers, L. Boone, E. L. Deci, J. Van der Kaap-Deeder, B. Duriez, W. Lens, L. Matos, A. Mouratidis, *et al.*, “Basic psychological need satisfaction, need frustration, and need strength across four cultures,” *Motivation and Emotion*, vol. 39, no. 2, pp. 216–236, 2015.

[172] R. M. Ryan, C. S. Rigby, and A. Przybylski, “The motivational pull of video games: A self-determination theory approach,” *Motivation and Emotion*, vol. 30, no. 4, pp. 344–360, 2006.

[173] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart, and J. N. Pidruzny, “The development of the game engagement questionnaire: A measure of engagement in video game-playing,” *Journal of Experimental Social Psychology*, vol. 45, no. 4, pp. 624–634, 2009.

[174] L. E. Nacke, C. Bateman, and R. L. Mandryk, “Brainhex: A neurobiological gamer typology survey,” *Entertainment Computing*, vol. 5, no. 1, pp. 55–62, 2014.

[175] J. Fürnkranz and E. Hüllermeier, “Preference learning,” in *Encyclopedia of Machine Learning*, pp. 789–795, Springer, 2011.

[176] J. Fürnkranz and E. Hüllermeier, “Pairwise preference learning and ranking,” in *Proceedings of the European Conference on Machine Learning*, pp. 145–156, Springer, 2003.

[177] D. Chatzakou, A. Vakali, and K. Kafetsios, “Detecting variation of emotions in online activities,” *Expert Systems with Applications*, vol. 89, pp. 318–332, 2017.

[178] D. Hazer-Rau, L. Zhang, and H. C. Traue, “A workflow for affective computing and stress recognition from biosignals,” in *Proceedings of the Electronic Conference on Sensors and Applications*, vol. 15, p. 30, 2020.

[179] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, “Deep learning for affective computing: Text-based emotion recognition in decision support,” *Decision Support Systems*, vol. 115, pp. 24–35, 2018.

[180] R. J. Lewis, “An introduction to classification and regression tree (cart) analysis,” in *Proceedings of the society for Academic Emergency Medicine (SAEM) annual meeting*, 2000.

[181] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[182] W.-Y. Loh, “Classification and regression trees,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[183] D. Melhart, D. Gravina, and G. N. Yannakakis, “Moment-to-moment engagement prediction through the eyes of the observer: Pubg streaming on twitch,” in *International Conference on the Foundations of Digital Games*, pp. 1–10, 2020.

[184] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Creating dali, a large dataset of synchronized audio, lyrics, and notes,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[185] J. Chabassier, *Modélisation et simulation numérique d’un piano par modèles physiques*. Theses, Ecole Polytechnique X, Mar. 2012. Thèse sous la codirection de Antoine Chaigne, Unité de Mécanique, ENSTA ParisTech.

[186] X. Serra and J. Smith, “Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition,” *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[187] j. m. chowning, “the synthesis of complex audio spectra by means of frequency modulation,” *journal of the audio engineering society*, vol. 21, pp. 526–534, september 1973.

[188] E. Cooper, X. Wang, and J. Yamagishi, “Text-to-Speech Synthesis Techniques for MIDI-to-Audio Synthesis,” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 130–135, 2021.

[189] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” *arXiv preprint arXiv:1810.12247*, 2018.

[190] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, “Ddsp: Differentiable digital signal processing,” in *International Conference on Learning Representations*, July 2020.

[191] R. Castellon, C. Donahue, and P. Liang, “Towards realistic midi instrument synthesizers,” *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2020.

[192] N. Jonason, B. Sturm, and C. Thomé, “The control-synthesis approach for making expressive and controllable neural music synthesizers,” in *2020 AI Music Creativity Conference*, 2020.

[193] Y. Wu, E. Manilow, Y. Deng, R. J. Swavely, K. Kastner, T. Cooijmans, A. Courville, A. Huang, and J. Engel, “Midi-ddsp: Hierarchical modeling of music for detailed control,” 2022.

[194] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 2018*, 2018.

[195] F. Rigaud, B. David, and L. Daudet, "A parametric model of piano tuning," in *Proc. of the 14th Int. Conf. on Digital Audio Effects (DAFx-11)*, pp. 393–399, 2011.

[196] G. Weinreich, "Coupled piano strings," *The Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1474–1484, 1977.

[197] V. Emiya, N. Bertin, B. David, and R. Badeau, "MAPS - A piano database for multipitch estimation and automatic transcription of music," research report, July 2010.

[198] F. Vaccaro, M. Bertini, T. Uricchio, and A. D. Bimbo, "Effective triplet mining improves training of multi-scale pooled cnn for image retrieval," *Machine Vision and Applications*, vol. 33, no. 1, pp. 1–13, 2022.

[199] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2131–2145, 2019.

[200] M. Kawamura, T. Nakamura, D. Kitamura, H. Saruwatari, Y. Takahashi, and K. Kazunobu, "Differentiable digital signal processing mixture model for synthesis parameter extraction from mixture of harmonic sounds," in *ICASSP*, 2022.