AI-based telepresence for broadcast applications

Henrique Piñeiro Monteagudo, Rayan Daod Nathoo, Laurent Deillon, Changsheng Gao, and Touradj Ebrahimi

Multimedia Signal Processing Group (MMSPG), École Polytechnique Fédérale de Lausannenne (EPFL), Lausanne, Switzerland

ABSTRACT

In this paper, we introduce a new solution and underlying architecture that allows remote participants to interact with hosts in a broadcast scenario. To achieve this, background extraction is first applied to video received from remote participants to extract their faces and bodies. Considering that the video from remote participants are usually of lower resolutions when compared to content produced by professional cameras in production, we propose to scale the extracted video with a super-resolution module. Finally, the processed video from remote participants are merged with studio video and streamed to audiences. Given the real-time and high-quality requirements, both background extraction and super-resolution modules are learning-based solutions and run on GPUs. The proposed solution has been deployed in the Advance Mixed Reality (AdMiRe) project. The objective and subjective assessment results show that the proposed solution works well in real world applications.

Keywords: AdMiRe, background extraction, super-resolution, telepresence

1. INTRODUCTION

The goal of this paper is to introduce a new solution and underlying architecture that allows remote participants to be teleported into a studio environment by using their smartphone or webcam, and to process their content to produce high-quality composited video.

In the context of interactive broadcasting between a host and a remote participant, one of the fundamental problems to address is how to show the audience both the host and the remote participant at the same time. Indeed, we cannot stream two separate video to the audience, as it would require too much resources and would not reach the desired sensation of interactivity. Therefore, we propose to put both the host and the participant in a single video and broadcast the result. To achieve this, there is a need to extract the remote participant (face and body) from their video and seamlessly merge it with the host video. Another problem that needs to be considered is that remote participants video are usually of a lower resolution when compared to those from the host. Generally, this results in undesirable mixture of low-quality with higher-quality contents, unacceptable for professional broadcast scenarios. To solve this problem, we propose to apply a super-resolution algorithm to low-resolution video, in order to mix a higher-resolution version with the host video before broadcasting.

In this paper, we focus on two key components in the proposed architecture, namely, the background extraction and the super-resolution modules. We show the interaction between these modules and assess the performance of each alone and when combined. Several methods are proposed in the background extraction section, and one method is proposed for the super-resolution section. We compare them to other existing methods from the state of the art in terms of performance and complexity and conclude which are best suited for the application at hand. Considering the performance requirements, we propose to use learning-based solutions taking into account inevitable complexity-performance trade-offs.

The proposed solution has been deployed in the Advanced Mixed Reality (AdMiRe), an EC funded project composed of a European consortium that aims at development, validation and demonstration of innovative solutions based on Mixed Reality (MR) technologies. The consortium members are BRAINSTORM MULTIMEDIA from Spain as coordinator, DISGUISE SYSTEMS from UK, NORGES TEKNISK-NATURVITENSKAPELIGE

Further author information: (Send correspondence to Touradj Ebrahimi)
Touradj Ebrahimi: E-mail: touradj.ebrahimi@epfl.ch, Telephone: +41 21 693 2606

UNIVERSITET (NTNU) from Norway, Ecole Polytechnique Fédérale de Lausanne (EPFL) from Switzerland, UNIVERSIDAD POMPEU FABRA (UPF) from Spain, NORSK RIKSKRINGKASTING (NRK) from Norway, PREMIERE MEDIA from Ireland, SOCIETATEA ROMANA DE TELEVIZIUNE (TVR) from Romania and SPANISH NATIONAL RESEARCH COUNCIL (CSIC). Three partners in the consortium are end users, while others are technology providers from academia or industry. The primary goal of AdMiRe project is to allow TV audiences a step change in interactivity and to bring content creators a radical improvement in talent immersion and interaction with computer-generated elements.

The rest of this paper is organized as follows. We first present the end-to-end architecture of AdMiRe in section 2. Then, section 3 and section 4 introduce the selected background extraction and super-resolution modules, respectively. Experimental results are presented in section 5. The paper is then concluded in section 6.

2. ARCHITECTURE FOR ADVANCED MIXED REALITY

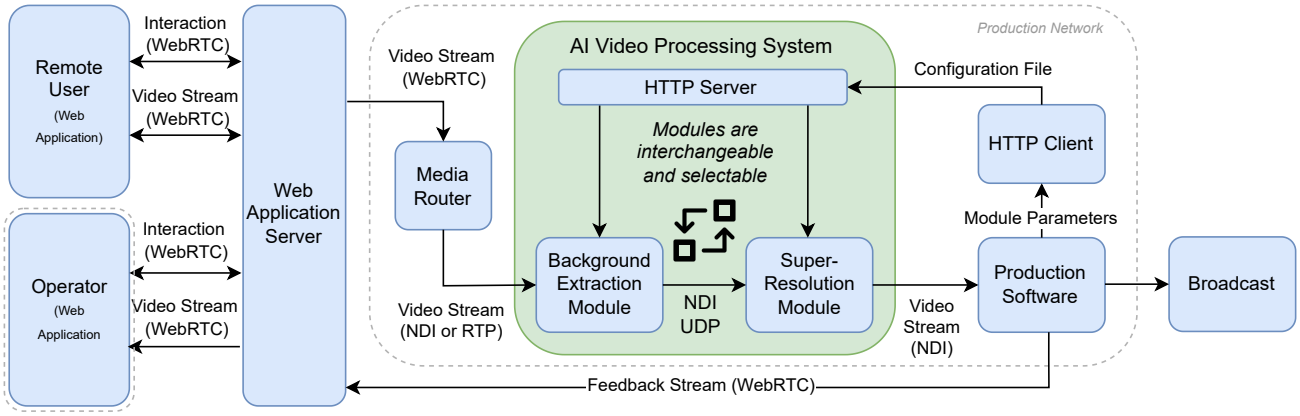

Figure 1. AdMiRe flow graph

The goal of AdMiRe project is to perform a live background extraction and super-resolution in order to best integrate a remote participant in a virtual TV set. The remote participant only uses a smartphone, tablet, or a computer with a webcam as capture device. Participant first logs into a web application and enters a call room administrated by an operator. The operator is then responsible for streaming the participant's live video-feed to background extraction and super-resolution modules before passing the result to the production software prior to broadcast of the final content.

The communication outside of the Production Network is guaranteed with WebRTC (Web Real-Time Communication), a popular programming interface, in particular for its ability to automatically adapt to the available bandwidth by changing the aspect ratio and the resolution of input content. This makes the solution more robust to network issues. The communication inside the Production Network is mainly performed using NDI (Network Device Interface)*, a video-over-IP transport and codec that is low latency and visually lossless. Below, we briefly describe some of the key modules in the end-to-end architecture:

Web Application allows to create and manage call rooms and users. The operator of each room can manage the audio and video streams from the users in the room. Note that the users can also get a live feedback stream of the final virtual TV set through their room.

Media Router is a desktop application that can receive a WebRTC stream from the Web Application Server, and route it as an NDI stream to the selected endpoint such as the local network. The Media Router runs on the same network as the Production Software and the AI Video Processing System.

Production Software is a software to create video content, add effects and usually supports a wide array of inputs, outputs and streaming protocols. It allows the integration of the AI Video Processing System results

*https://www.ndi.tv/

with the virtual environment desired by the production, such as a virtual TV news set or a virtual weather report set.

AI Video Processing System is a system performing the background extraction and super-resolution, which are main focuses of this paper. It is controlled by the production software through HTTP (Hypertext Transfer Protocol), from which it receives a configuration file before starting to process the input. The AI Video Processing System handles changes in resolution and aspect ratio by passing the incoming stream though a virtual camera provided by NDI Tools[†]. If needed, the AI Video Processing System can run on another machine than that hosting the Production Software, but it needs to stay in the same network.

The communication between both modules is ensured using either NDI via TCP (Transmission Control Protocol), or UDP (User Datagram Protocol). For the latter, since there is a limit in packet size, it necessitates to chunk and to encode the input content using JPEG or H.264/AVC. This leads to a low latency output but also to a slight loss in visual quality. The NDI link can tackle this challenge by transmitting raw frames at the expense of a small delay. Both performance/quality options were implemented using GStreamer[‡], a library for constructing graphs of media-handling components and operational on all major platforms. Furthermore a GStreamer plugin[§] was needed to support NDI.

The AI Video Processing System requires an NVIDIA Graphics Card with Tensor Cores and a decent Central Processing Unit (CPU) to achieve the targeted performance of real-time processing at at least 25 frames per second (fps). During the development a core i9 10900K with 20 Threads at 5 GHz was used paired with an NVIDIA RTX 3090.

3. BACKGROUND EXTRACTION

Background extraction can be broadly described as the computer vision task in which some object or objects of interest in an image or video are separated from the rest of the scene known as background, generating a foreground mask in the process. The goal of this operation is to insert a remote participant in a TV studio scene for live broadcasting. The participants will stream video of themselves in real-time through the Internet. This specific application narrows the objects of interest to be extracted from the background to only human subjects and introduces a hard real-time constraint – the used algorithm should provide a throughput of at least 25 frames per second to be usable on live television.

This will enable people to participate remotely in TV programs with widely available hardware (e.g. smartphone cameras or laptop webcams) and minimal effort on the user side. Specialized tools such as a physical green screen can simplify the background extraction process but are not commonly found outside of studio settings. Depth information coming from time-of-flight or Lidar sensors can potentially enhance the quality and simplify the process of background extraction, but the presence of depth sensors on common hardware is rather rare. Thus, the focus is placed on solutions targeting conventional RGB video as captured by a wide range of cameras, with arbitrary backgrounds as input.

An image frame I can be described as the combination in each pixel P of a foreground F and a background B modulated with an α matte factor by the compositing equation:[1]

$$I_P = \alpha_P F_P + (1 - \alpha_P) B_P, \tag{1}$$

If the value of the alpha matte is 0 or 1, the equation will simplify to a typical image segmentation problem. However, replacing the background of an image via image segmentation will result in artifacts visible in the final composition. Some objects, like human hair, can be narrower than a pixel in a digital image. Some other objects, like glasses, are translucent, and their color is a combination of the foreground and background. Constraining the value of α in Equation (1) to a floating point value in the range $(0, 1)$ results in what is known as image matting, which is a more adequate technique for background replacement.[2]

[†] https://www.ndi.tv/tools/

[‡] https://gstreamer.freedesktop.org/

[§] https://github.com/teltek/gst-plugin-ndi

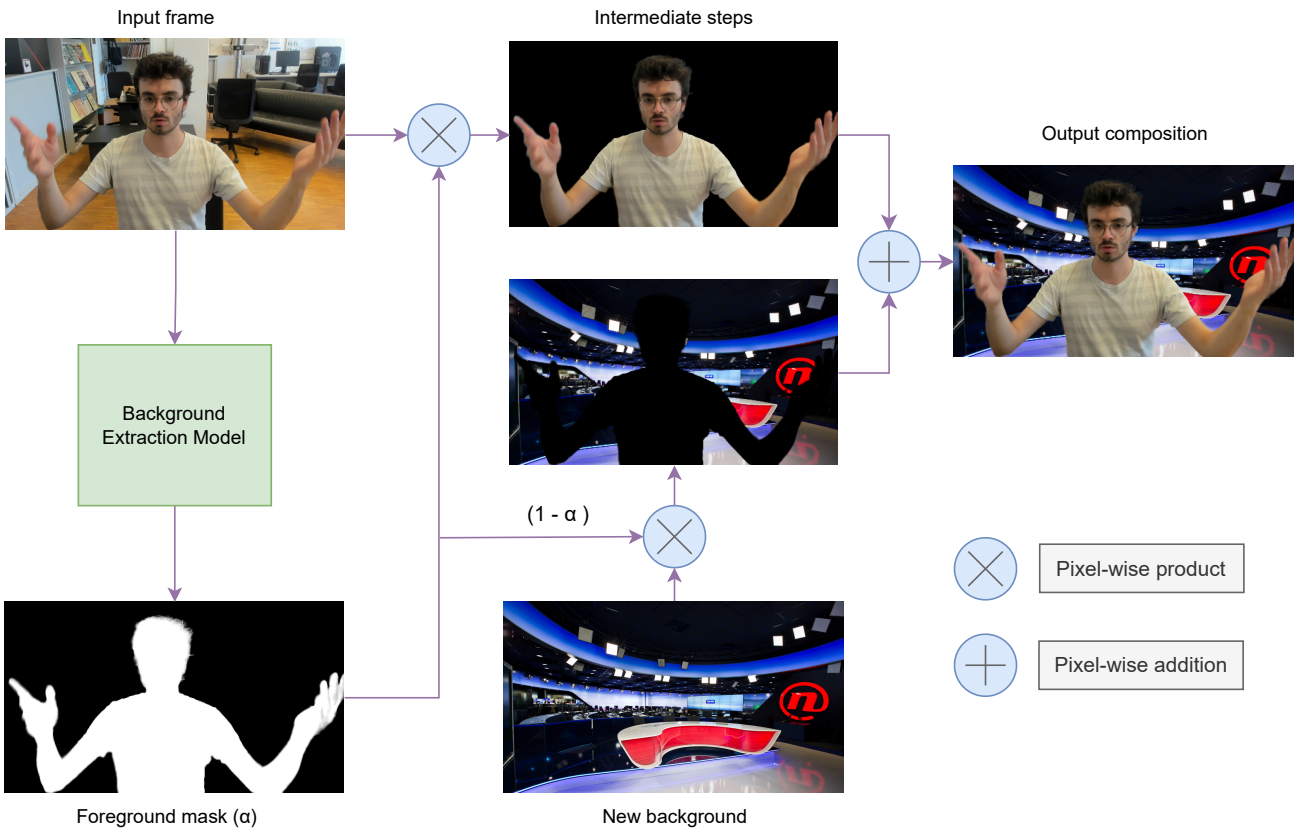

Figure 2. Diagram of the background extraction and substitution process

3.1 State of the Art in Background Extraction

Background extraction is a widely studied topic in the field of computer vision and image and video processing. It is out of the scope of this article to provide an in depth study of the state of the art in the field, so we limit ourselves to works more closely related with our application.

Auxiliary-based matting. Traditional matting algorithms[3,4] require the user to input a coarse segmentation in three regions (background, unknown and foreground) called trimap. These methods tend to be object agnostic, not targeting any specific object category. Exploiting the progress in deep learning in the last decade, some methods[5] use convolutional neural networks with RGB and trimap inputs to solve the matting task. In our particular application, avoiding friction for the user throughout the process is a key objective, thus we focus on approaches that do not need user interaction. Another family of methods[6,7] rely on using an additional image of the background as auxiliary input. Although these algorithms do not require user input, getting the extra background shot still complicates the process for the user and restricts the usability to static backgrounds.

Fully-automatic matting. Recent approaches propose methods for portrait or person matting without any auxiliary input. In order to solve this challenging task such algorithms restrict the foreground to be of only one possible class (e.g. human portrait or full body), and include a semantic estimation component or step in their networks. Fast deep matting[8] is an early successful work in this direction, using a feathering block on top of a light semantic segmentation, achieving real-time performance. However, it places its focus on head-only matting and works in low resolution (128x128). MODNet[9] achieves a high frame-rate on a higher resolution (512x512) by using only a single network optimized on various objectives. Robust Video Matting[10] uses recurrent layers to exploit temporal information and deep guided filter module[11] to generate high resolution outputs (up to 4K) while working in real-time. As part of their Maxine Video Effects software development kit, NVIDIA provides a highly-optimized solution for background matting. While details about the underlying architecture or training

data are not disclosed, the solution supplies an extremely high throughput in terms of frames per second while offering a decent quality.

3.2 Selected Solutions for Background Extraction

Taking into account the main constraints of our application – real-time performance and higher quality of experience for the end user – as well as the overall quality of the results, we select as background extraction solutions to use within our architecture three fully automatic matting methods: MODNet, Robust Video Matting and Maxine.

MODNet.[9] MODNet's architecture is divided in three branches. A low-resolution branch for semantic estimation, for which MobileNetV2[12] is adopted to facilitate real-time performance. This branch is optimized through an L2 loss between the branch output and a downsampled and smoothed version of the ground-truth foreground mask. A high-resolution branch for detail prediction, trained by an L1 loss between the output of the branch and the ground truth mask focused on a detail region. Finally, a fusion branch to generate the final alpha matte, which uses as optimization target the L1 loss between the ground truth and the output with an additional term called compositional loss, which measures the absolute difference between the composited image obtained through the output alpha mask and the composition of the ground truth foreground and background. MODNet is trained end-to-end through the weighted sum of the three losses. The authors report results which outperform previous methods on the PPM-100 and AMD datasets while claiming a throughput of 67 frames per seconds on a GTX 1080Ti GPU with an input size of 512 x 512 pixels.

Robust Video Matting.[10] Robust Video Matting's architecture can be divided in three main components. The first one is a feature-extraction encoder, following the design of state of the art works in semantic segmentation MobileNetV3-Large[13] is adopted as the backbone, although the possibility to use a more complex model (ResNet50) is also suggested by the authors if more computational capabilities are at hand. The second component is a recurrent decoder which adopts convolutional gated recurrent units (ConvGRU[14]) layers to aggregate temporal information, a main novelty in this paper which explicitly targets video. Finally, the third and final component is a deep guided filter module. When high resolution video is taken as input, they are downsampled before passing through the encoder-decoder structure. This module takes as input the low-resolution output mask, the network's hidden features and the high resolution input frame and outputs a high resolution alpha mask and foreground color. The whole network is trained end-to-end. The authors claim to outperform competing methods, while having a lower complexity, reporting a throughput of 104 frames per second on an NVIDIA GTX 1080 Ti GPU at HD resolution.

Maxine.[15] The NVIDIA Maxine Video Effects SDK, provides an effect called AI Green Screen which performs background extraction. The output of this algorithm is the alpha matte mask of the foreground. NVIDIA does not disclose information on the used network architecture or training procedure. However, the algorithm provides decent quality results while operating at up to 200 frames per second with a 1280 x 720 pixel input, which makes it of great interest for an application with severe real-time constraints such as ours. Maxine can take inputs of up to 4K resolution. A strong limitation of Maxine's background extraction for our application is that the model provided by NVIDIA targets head and shoulders shots only, failing to provide adequate results on scenes with the full body of a subject.

4. SUPER-RESOLUTION

Image Super-Resolution (SR) is the process of recovering a higher-resolution output image from one or many degraded low-resolution input images. It is an important research field in image processing and computer vision as it is used in many real-world applications such as medical imaging, security, or telepresence in our case. It can be further divided into Single-Image Super-Resolution (SISR) taking a low-resolution image as input and yielding a higher-resolution image as output, and Multi-Image Super-Resolution (MISR) taking multiple low-resolution images as input and yielding a higher-resolution image as output. Other works focus on getting multiple images as output but we will not cover them here as they are out of the scope of this paper.

In the context of SISR, we can formally refer to the low-resolution image x as a higher-resolution image y that went through a degradation process followed by a downsampling step, to which an independent Additive

White Gaussian Noise (AWGN) was added. The degradation process represents real-world disturbances such as capture sensor noise, motion blur, or compression artifacts. It can mathematically be stated as follows:

$$x = (D(y|\lambda) \downarrow_s) + N \tag{2}$$

with D the degradation function, λ the degradation parameters such as noise or encoding artifacts, s the downsampling factor, and N an independant AWGN noise with standard deviation σ. SISR is therefore the reverse process, i.e finding y given x. It is a very ill-posed problem due to the irreversible loss of information during the degradation process. Additionally, there is usually not a unique mapping between the low-resolution input x and its high-resolution output y, since multiple high-resolution images can correspond to a single low-resolution image. On the other hand, MISR is known to be a lesser ill-posed problem since it can take advantage of the information contained in the multiple input images.

This work is intended to enable people to be teleported in live TV shows with the sole support of their smartphone front-facing camera, or their webcam. Generally, target users do not all have access to high-end devices offering a good enough resolution for TV broadcast, e.g greater than or equal to FULL HD (1920x1080) in our case. It can also happen that the available bandwidth does not allow for streaming high-resolution content, in which case the resolution of the input video will be automatically decreased in order to keep the live feed going. Finally, it is highly desirable that the host and the remote participant's respective video contents have close to the same resolution before their subsequent composition. To this end, an SR module was added to the pipeline, providing a complete solution dealing with a large number of situations. In this section, we will first present the Single-Image Super-Resolution (SISR) and the Multi-Image Super-Resolution (MISR) tasks, followed by some of the state-of-the-art methods, and finally present the selected solution with respect to the requirements and constraints imposed by our use case.

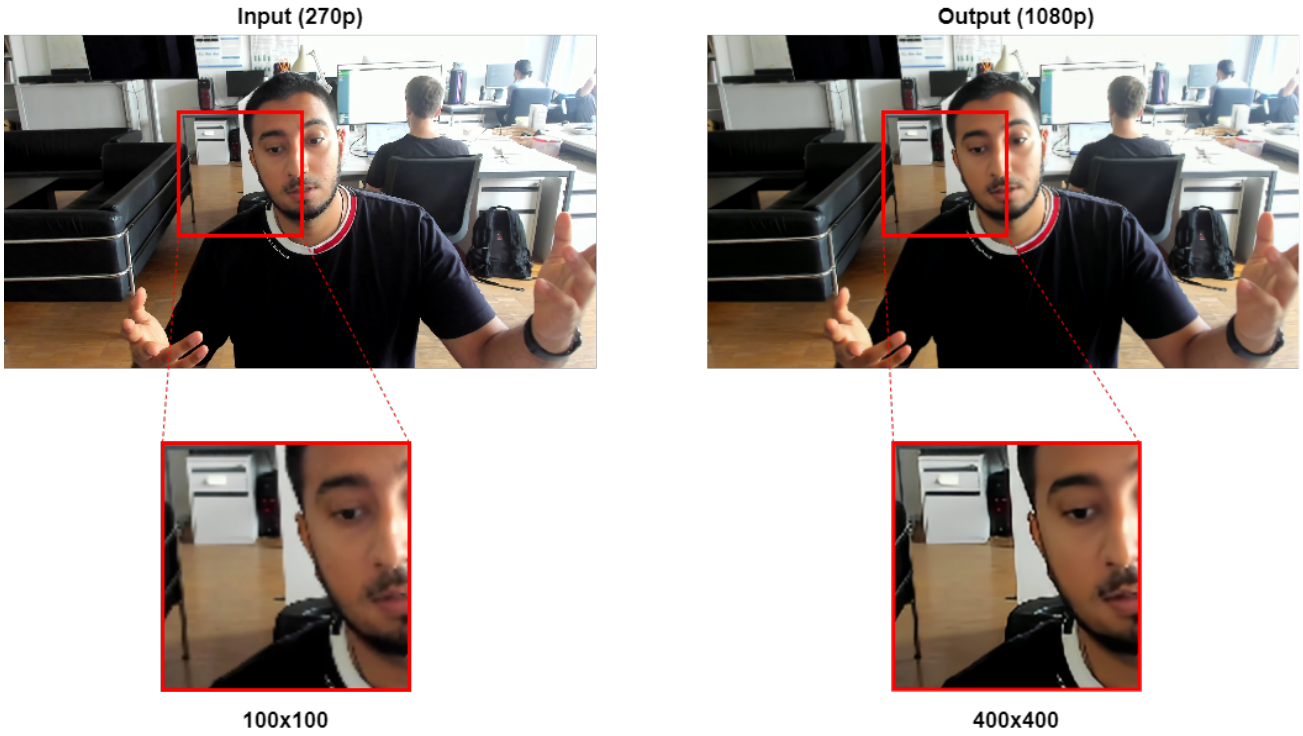

Figure 3. Illustration of super-resolution operation

4.1 State of the Art in Super-Resolution

Super-resolution has been an active field of study for quite some time now. While traditional techniques such as nearest-neighbor, bilinear, bicubic, or Lanczos interpolation are computationally efficient, they suffer from

accuracy limitations, motivating the research community to keep contributing to the field. As stated before, it can be divided into SISR and MISR.

Single-Image Super-Resolution (SISR). In 2014, Dong et al. proposed the first successful attempt towards using only convolutional layers for super resolution, Super-Resolution Convolutional Neural Network (SRCNN),[16] an end-to-end low-resolution to high-resolution mapping with very little pre/post-processing. This work is the pioneer of deep neural network approaches to solve the SR problem. Later, Kim et al.[17] explored deeper architectures with Very Deep Super Resolution (VDSR), by using a network commonly known as VGG-Net. They contributed in three different manners by 1) confirming the efficiency of deeper networks for the SR task, also thanks to Simonyan et al.,[18] 2) introducing residual learning to generate the difference between the high-resolution and low-resolution image, and 3) using gradient-clipping which allowed for very high learning rates and enabling faster convergence. One drawback both SRCNN and VDSR suffered from was their usage of an early-stage upsampling operation, e.g bicubic interpolation, in order to match the output resolution. This step leads to computationally expensive operations in later stages of the network, since the same network progressively grows in proportion to deal with larger sized inputs. To circumvent this issue, Fast Super-Resolution Convolutional Neural Network (FSRCNN)[19] was proposed by Dong et al. in 2016, in which they shifted the upsampling operation close to the output of the network. As a result, the inference latency improved from 1.3 fps with SRCNN, to real-time (24 fps) with FSRCNN. Many follow-up works have then been proposed in the following years[20] such as residual-learning-based networks,[21] recurrent networks,[22] attention-based networks[23] and Generative Adversarial Networks (GAN) with SRGAN[24] and later ESRGAN.[25] SISR algorithms are usually evaluated on Set5,[26] Set14,[27] Urban100,[28] BSDS100,[29] and recently DIV2K.[30] Figure 3 shows an example of typical results obtained with an SISR process.

Multi-Image Super-Resolution (MISR). Kappeler et al. proposed in 2016 VSRnet[31] a network for video super-resolution (VSR) based on the previously mentioned SRCNN.[16] Recent studies[32,33] have then confirmed that the utilization of inter-frame information greatly influences performance. Since then, many video super-resolution contributions have been proposed. According to a recent survey,[34] the general pipeline of deep learning methods for VSR tasks is composed of three modules, namely an alignment module, a feature extraction and fusion module, and a reconstruction step. More generally, VSR can be divided into frame-alignment techniques, and no-frame-alignment techniques. The former category contains methods estimating and compensating motion, as well as deformable convolution methods. The latter uses 2D convolutions, 3D convolutions, Recurrent Convolutional Neural Networks (RCNN), non-local methods, and others. Common MISR literature evaluate their work on Vimeo-90K-T,[35] a dataset of +90'000 RGB video with dimensions 448x256, REDS,[36] a dataset of 270 RGB video with dimensions 1280x720, or Vid4 containing 4 RGB video with diverse content. In the context of a real-time application, a unidirectional lightweight network is preferred in order to achieve the required output framerate. To this end, several studies such as TDAN[33] or ESPCN[37] were proposed achieving processing speeds suitable for real-world applications assuming that the hardware configuration matches the requirements.

Below, two state-of-the-art techniques whose implementations were largely used, namely EDSR, a CNN-based model, and ESRGAN, a GAN-based model, are presented in more details.

Enhanced Deep Super-Resolution (EDSR). EDSR[21] is an improved version of SRResNet,[24] a network using the Residual Network (ResNet)[38] architecture originally designed for the Image Recognition task. In SRResNet, the authors simply employed the ResNet architecture without much modification whereas the EDSR network obtained better performance by simplifying the network architecture and removing unnecessary modules, namely the Batch Normalisation layers and the final ReLU activation function in each residual block. Additionally, they proposed a Multi-scale Deep SR (MDSR) architecture working with multiple scales at the same time. To do so, they took advantage of the inter-scale correlation, as VDSR[17] did, and used shared parameters across different scales to further reduce the overall complexity. Both EDSR and MDSR achieve better performance in terms of quantitative measures (e.g Peak Signal to Noise Ratio (PSNR)) compared to SRCNN[16] and VDSR[17] when trained on the DIV2K dataset[30] and tested on its validation set. They respectively won the first and second places of the NTIRE2017 SR Challenge.[39]

ESRGAN: Enhanced Super-Resolution Generative Adversarial Network. ESRGAN[25] is an improved version of SRGAN[24] in which Wang et al. proposed a new network architecture and used an adversarial as well as an improved perceptual loss to surpass their predecessors' performance. In particular, they removed all

Batch Normalisation layers and replaced the original basic blocks with their proposed Residual-in-Residual Dense Blocks (RDDB), which combines multi-level residual network and dense connections. Among other developments, they also improved the discriminator module of the GAN architecture, using a Relativistic average GAN (RaGAN),[40] which learns to judge "whether one image is more realistic than the other", rather than "whether one image is real or fake". They won the first place with the best perceptual index in the PIRM2018-SR Challenge,[41] the first SR competition that evaluated the performance in a perceptual-quality aware manner.

4.2 Selected Solution for Super-Resolution

Since AdMiRe's use case is related to live TV broadcasts and interaction with remote participants, the main constraint was real-time processing. Moreover, TV stations have to broadcast their content at a certain frame rate to keep the shows visually appealing. Finally, it is important to note that no pre-defined hardware constraint was imposed in the context of the project as one could have in a typical real-time application scenario. To summarise, our full pipeline including the background extraction and super-resolution modules had to run in real-time at a minimum frame rate of 25 fps, and had to satisfy these conditions on a typical high-end graphics card, e.g NVIDIA GeForce RTX 20XX or 30XX models.

With these requirements in mind, we needed a highly efficient solution and therefore decided to rely on Maxine by NVIDIA for the super-resolution module, a suite of GPU-accelerated Software Development Kits (SDK)[¶]. As mentioned before, Maxine is optimised for NVIDIA graphics cards with Tensor Cores and claims to use state-of-the-art algorithms, resulting in fast processing and good results (c.f section 5.4). Note however that since they do not disclose the architecture nor the training process of their models, it is unclear which kind of data it was trained with.

Maxine's super-resolution module upscales the input while also reducing the blocky and noisy artifacts. It can enhance the details and sharpen the output while simultaneously preserving the content[‖]. It supports frame resolutions from 144p to 1080p as input and up to 4320p as output with the latest GPUs, with available scaling factors being 4/3x (∼1.33x), 1.5x, 2x, 3x and 4x. NVIDIA also provides a light-weight upscaling module and an artifact reduction module to deal with encoding artifacts.

5. EXPERIMENTS

5.1 Performance Evaluation Criteria

Performance evaluation criteria for both the background extraction and the super-resolution modules are presented based on objective evaluation using commonly used metrics, as well as ad hoc perceptual quality assessment by illustrating typical examples.

Background extraction. In order to evaluate the quality of an alpha matte, the mean squared error (MSE) or mean absolute deviation (MAD) with respect to the ground truth can be used.

Super-resolution. In order to evaluate the quality of the super-resolution methods, Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Naturalness Image Quality Evaluator (NIQE) were used. NIQE is a no-reference image quality score that evaluates the naturalness of a given image. According to Ma et al.,[42] NIQE and other no-reference metrics are highly correlated with human mean-opinion-scores in the context of super-resolution, making them a fast and simple method to predict the perceptual quality of results obtained from an SR algorithm. The NIQE implementation from Scikit-Video[**] was used for all experiments.

[¶] https://docs.nvidia.com/deeplearning/maxine/vfx-sdk-programming-guide/

[‖] https://github.com/NVIDIA/MAXINE-VFX-SDK

[**] http://www.scikit-video.org/stable/

5.2 Datasets

In order to assess the performance of algorithms selected for inclusion in the proposed system, both popular and publicly accessible datasets as well as a dataset with contents specifically produced for the use case of the project are used.

Background extraction. To provide a fair comparison of the performance in terms quantitative performance metrics of the selected models for background extraction, we use for evaluation the test split of the Video Matte dataset.[7] This synthetic dataset contains 50 video composed from ground truth mattes and a combination of static and dynamic backgrounds. To evaluate background matting on images we use the PPM-100 dataset, composed of 100 images with high-quality ground truth mattes of the subject. The images are human portraits containing a diverse set of subjects, perspectives and poses.

Super-resolution.Since there is no information available on the internal architecture of Maxine's network, it is unclear if it indeed takes advantage of multiple input frames or not. It was therefore decided to evaluate Maxine on one dataset of high-resolution images and one dataset of high-resolution video, in order to assess its performance on both input types. Also, most of the commonly used datasets contain images or video of landscapes, animals, cities, and dynamic backgrounds, which are not in line with the use case under study. Finally, as super-resolution module adopted here only accepts inputs whose minimum dimension is 144 pixels large, only datasets containing images or video whose minimum dimensions are above 288 pixels large are used for an evaluation of the x2 super-resolution results, and more than 576 pixels large for the x4 results.

To satisfy these constraints and to evaluate the super-resolution module in a way that is close to our final use case, it was decided to use the validation split of DIV2K,[30] a recently introduced dataset of various high-resolution (2K) images, and in addition create an evaluation dataset which we will refer to as MMSPG Video Super-Resolution (MMSPG VSR). The latter is composed of eight video of approximately ten seconds each, recorded at a resolution of 1920x1080p, at 25 frames per second, using a Logitech BRIO webcam and the OBS software[††]. In each video, a different subject speaks and/or moves in a way compliant to the use case under consideration, e.g when remotely talking to a TV presenter.

5.3 Results of the Background Extraction Module

To perform experiments, a workstation equipped with an Intel i9-10900KF 10-core processor running at 3.7 GHz base frequency, an NVIDIA RTX 3090 GPU (24 GB RAM) and 32 GB of system RAM memory was used. The platform ran Windows 10 as operating system.

Quantitative results. In Table 1, the results of the experiments ran on the video matting datasets are presented. To generate these results, the pretrained models provided by the authors were used. In the case of Robust Video Matting, MobileNetV3 is selected as backbone model. For the PPM-100 dataset, the RVM model is run without its recurrent stages, as those layers target video inputs and this dataset is composed by still images. In Table 2 the average measured time to process a single frame of a video for each method is reported. These measurements were computed by averaging the mean forward time of each video in a set of eight. The reported time corresponds to a forward pass of each model and does not cover additional processing steps.

Table 1. Background extraction results in terms of mean absolute difference (MAD) and mean square error (MSE) between ground truth and mattes predicted by the selected models. The lower the value, the better the performance.

Dataset	Method	MAD ↓	MSE ↓
VideoMatte 512 × 288	RVM	**0.0061**	**0.0015**
	MODNet	0.0093	0.0042
	Maxine	0.0210	0.0145
PPM-100 Various resolutions	RVM	0.0130	0.0074
	MODNet	**0.0103**	**0.0047**
	Maxine	0.0451	0.0396

[††] https://obsproject.com/

Table 2. Average forward time of each model to process one frame. Averaged over mean processing time per frame in 8 video. The lower the value the lower (better) the complexity.

	Input Resolution	
Method	1280x720	1920x1080
RVM	6.26 ms (159.61 fps)	7.58 ms (132 fps)
Maxine	**2.98 ms (335.36 fps)**	**5.37 ms (186.14 fps)**
MODNet [1]	13.68 ms (73.12 fps)	13.68 ms (73.12 fps)

[1] MODNet's maximum input size is 512 for the smaller side of the image, so input resolution was 896x512 in both cases.

Robust Video Matting performs best on the VideoMatte dataset in both selected metrics while MODNet is the best performing among the three selected models on the PPM-100 dataset. Robust Video Matting explicitly targets video and exploits temporal information, so it is logical that its performance degrades in PPM-100 with respect to VideoMatte. Regarding Maxine, its worse results can be explained by two main factors:

1. Although details about Maxine's architecture or training data are not disclosed, the model targets a video conference setting, where only the head and shoulders of the subjects are visible. Many of the shots contained in these datasets show different poses and perspectives, some of them including the full body of the subjects. These kinds of samples were probably not present in the training data of Maxine but are present on the training data of both other models.

2. Maxine focuses on real-time operations and thus is performance-oriented, while the other methods are willing to sacrifice some speed for a better performance. Comparing the results in Table 2, it can be seen that Maxine is faster than the other selected methods for a given resolution.

Qualitative results. In Fig. 4, the results of applying the three selected background extraction methods on two example images from the VideoMatte dataset are displayed. Representative behavior and failure cases of the models can be observed in these examples. As mentioned before, Maxine only targets upper body shots in a head-and-shoulder configuration. In the second row of the figure, one can observe failure of background extraction in presence of full body shots. MODNet sometimes fails when a complex background presents similar color to the foreground subject or his/her clothes. Robust Video Matting results at times exhibit artifacts in the subjects boundary areas, specially when they moves at high speed, arguably because its underlying design exploits temporal information and may not be able to adapt fast enough to rapid changes in the foreground.

5.4 Results of the Super-Resolution Module

All our experiments regarding the super-resolution module were run using the same hardware platform as in section 5.3.

Quantitative results. For each of the selected datasets, first downsampling operation is performed on their images or video using a bicubic interpolation over 4×4 pixel neighborhood with the corresponding downsampling factor using OpenCV,[43] before applying the different super-resolution algorithms with that same factor. Note that even though the downsampling kernel is similar to the one in Matlab,[44] as commonly used in the field, the averaging weights slightly differ in OpenCV, which could lead to different results when compared to referred publications. Regarding the upsampling step, we consider OpenCV bicubic interpolation as the baseline, and include results computed using EDSR[21*] as well as ESRGAN[25†] in order to quantitatively compare to those by Maxine. Motivated by the fact that Maxine slightly changes the final color distribution of the results and as described in Testolina et al,[45] the luminance component Y for each image/video is then ontained following ITU-R Rec. BT.709-6.[46] Finally, the quantitative metrics for each pair of super-resolution and original (high-resolution) image/video are computed, and averaged them before reporting their corresponding final results. Note that ESRGAN was only available for the x4 upscaling factor.

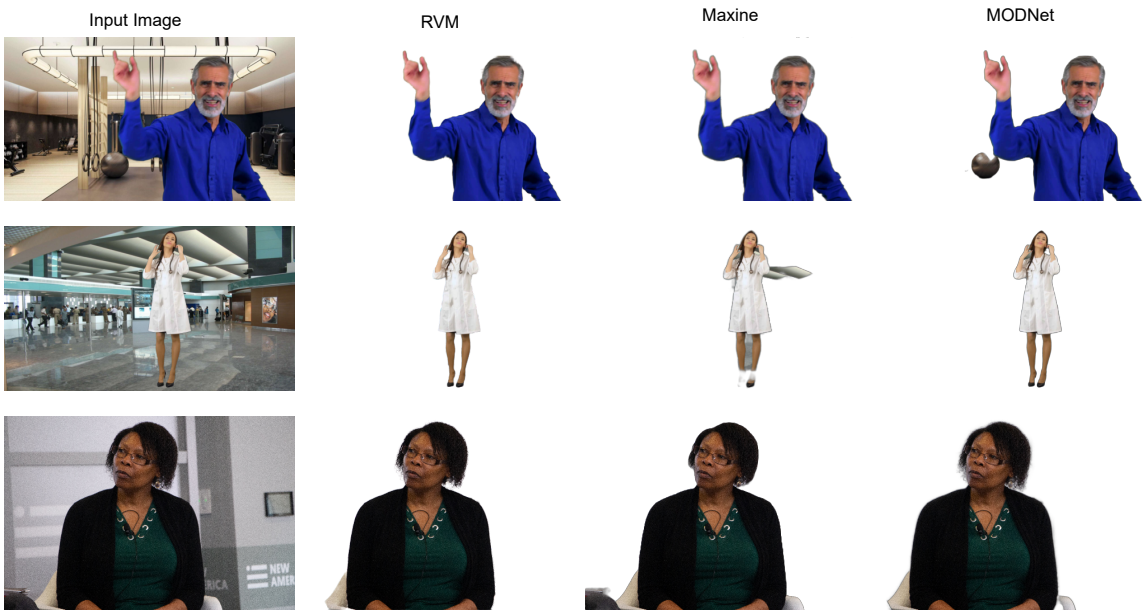

Figure 4. Example results from the VideoMatte (first two rows) and PPM-100 (third row) datasets

In Table 3, the quantitative results using several methods are presented. They were computed on two different datasets, DIV2K[30] and MMSPG VSR, both with x2 and x4 scaling factors. The first line for each dataset shows the quantitative results of the untouched dataset against itself in the case of full-referenced metrics (PSNR and SSIM), and the untouched dataset alone in the case of non-reference metric (NIQE). This gives the theoretical maximum/minimum value one could reach if the corresponding metric was considered to be "perfect".

One can observe that the bicubic interpolation is the closest to the ground truth, pixel-wise, as it overall obtains the largest scores in terms of PSNR and SSIM, the former metric basically being computed based on the squared distance between pixel values. Furthermore, one can notice that the results are subjectively better-looking with Maxine due to their smoothed aspects, as shown in figure 5. One reason that behind this behaviour could be the perception-distortion tradeoff, as explained by Blau et al.[47] In their contribution, they reported that algorithms that are superior in terms of perceptual quality are often inferior in terms of e.g. PSNR and SSIM, especially when they are based on GAN. Blau et al. also introduced a perception-distortion plane, in which they introduce a region they consider to be impossible to reach, and in the proximity of which it is only possible to improve either perceptual quality or distortion, one at the expense of the other. They further state that evaluation metrics that compute the distance between deep-net features have a weaker trade-off with perceptual quality. Finally, they advise that image restoration algorithms should always be evaluated with a pair of full-reference and non-reference metrics, then accounting for both perceptual quality and distortion. Which is why NIQE is included in Table 3.

One can see in the table 3 that the results of the NIQE metric are indeed in line with the perceptual quality difference in the example frames below (figures 5 and 6), except for the ESRGAN which are surprisingly of very low values. Motivated by the questionable naturalness of visual results of ESRGAN such as in figure 5, these results are considered as specific and rate cases and overall Maxine can be considered to perform better when taking into account the three metrics and the visual results altogether. One hypothesis behind this conclusion is that since ESRGAN was trained using a perceptual loss, it could have led to good quantitative values but bad qualitative results with metrics such as NIQE. On a similar note, it is important to state that the perceptual-metric-optimised weights are used rather than the PSNR-optimised counterparts for generating the ESRGAN results. Also, it should be noted that since ESRGAN was trained on images first downscaled with the bicubic

*Implementation and weights from OpenCV Contrib:[43] https://github.com/opencv/opencv_contrib

†Implementation and weights from https://github.com/xinntao/ESRGAN

interpolation function from the Matlab library, it might introduce artifacts when applied after the downsampling step from OpenCV library, leading to different quantitative results when compared the stated conclusion in the original paper. Finally, ESRGAN was only trained on pictures of e.g landscapes, animals, etc, and could further explain different conclusions when applied on images/video containing human faces.

Table 4 shows the average measured forward time to process a single frame of a video for each method. These measurements were computed similar to section 5.3 by averaging the forward time of a set of eight video of different lengths. One can observe that Maxine offers a very fast processing pipeline, allowing to reach the required framerate while still matching main requirements of the use case under consideration.

Qualitative results. Figure 6 contains additional examples of results generated using the MMSPG VSR dataset. This dataset is composed of eight 1920x1080 resolution video recorded for validation purposes particularly focused on the use case under consideration. The first column shows the results of a traditional bicubic interpolation, the second column shows the results computed using Maxine, and the last column presents the ground truths. As explained in the subsection 5.1, the images were first downsampled with factor 4 using the bicubic interpolation function from OpenCV, before being passed through the different upscaling methods. Only bicubic interpolation and Maxine results are shown alongside the ground truth here, for visibility purposes. One can observe the improvement Maxine brings to a traditional bicubic interpolation.

Table 3. Super-resolution results in terms of Peak-Signal-to-Noise-Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Naturalness Image Quality Evaluator (NIQE) between the ground truth and the super-resolution output predicted by the different methods. We show the results for both x2 and x4 upscaling factors and evaluate the methods on the validation set of the DIV2K[30] dataset and the MMSPG VSR dataset. Note that the pretrained weights for x2 upscaling using ESRGAN were not available, we therefore do not report them.

Dataset	Upscaling factor	Method	PSNR/SSIM ↑	NIQE ↓
DIV2K (2K resolution images)	x1	None	∞/1.0	9.25
	x2	Bicubic	**34.36/0.78**	13.29
		EDSR	33.58/0.75	13.13
		Maxine	30.94/0.65	**12.46**
	x4	Bicubic	28.29/0.51	16.15
		EDSR	**28.55/0.50**	16.43
		ESRGAN	18.71/0.15	**6.85**
		Maxine	26.57/0.43	**13.98**
MMSPG VSR (1920x1080p sequences)	x1	None	∞/1.0	11.75
	x2	Bicubic	**39.64/0.91**	12.53
		Maxine	35.57/0.69	**11.56**
	x4	Bicubic	**31.73/0.75**	13.47
		ESRGAN	24.24/0.40	**6.35**
		Maxine	30.69/0.58	**11.75**

Table 4. Average forward time of the super-resolution module to process one frame. Averaged over mean processing time per frame in 8 video. A dash corresponds to a result that could not be computed due to model/GPU constraints.

	Input Resolution		
Method	640x360	1280x720	1920x1080
Maxine (x2)	1.59 ms (628.39 fps)	7.75 ms (129.03 fps)	15.06 ms (66.39 fps)
Maxine (x4)	7.20 ms (138.76 fps)	-	-

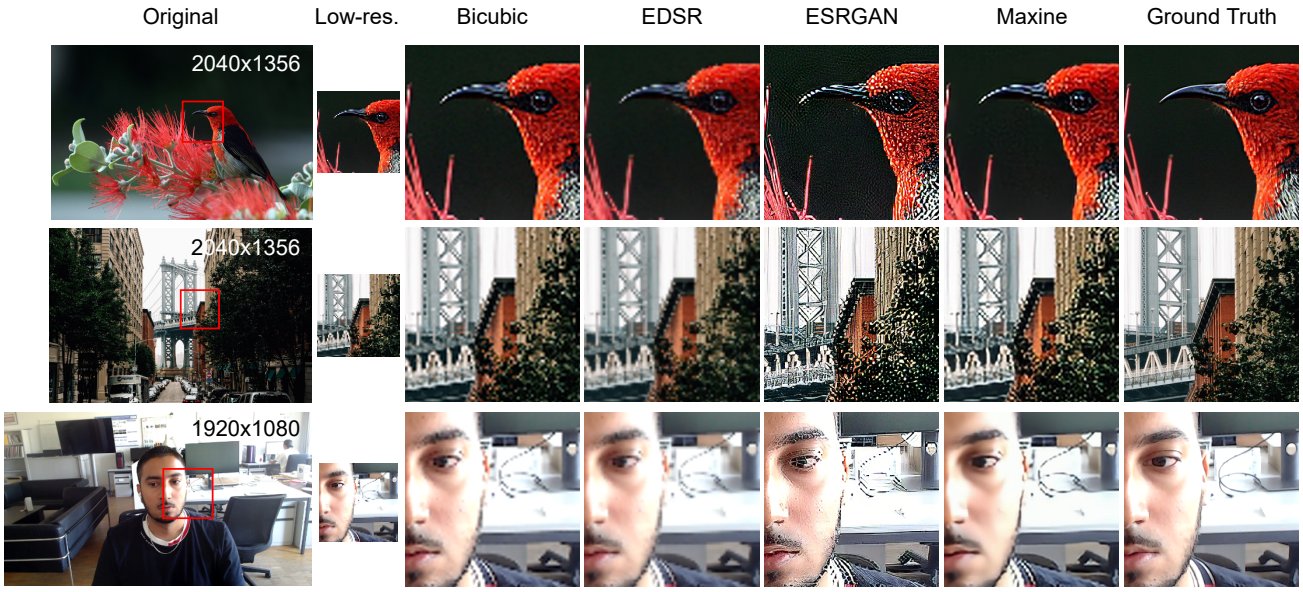

Figure 5. Example results on the DIV2K dataset (first two rows) and the MMSPG VSR dataset (last row). On the left, the original (high-resolution) images are shown, then the crop of interest after bicubic downscaling (x4), then the result from each of the discussed methods.

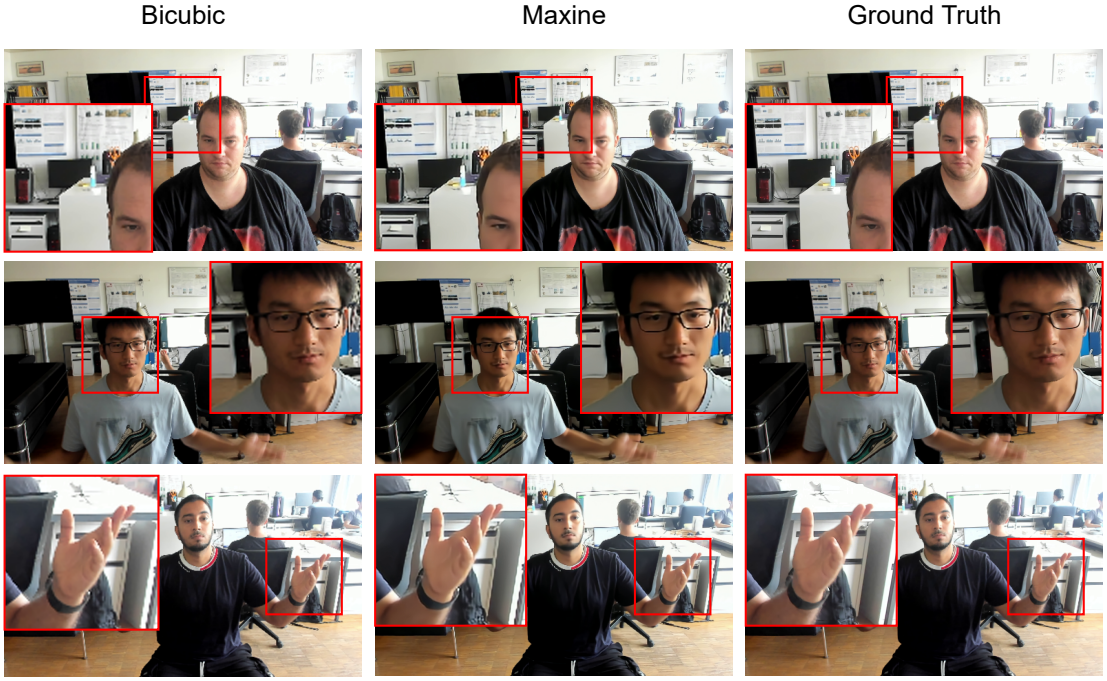

Figure 6. Additional example results from the MMSPG VSR dataset. 1920x1080p frames coming from the set of video recorded to construct MMSPG VSR, matching the use case under consideration.

5.5 Results of the Combined Modules

An most current smartphones and webcams can record video at 1280 x 720 or 640 x 360 resolutions, it is reasonable to assume that most participants will streaming their video in those resolutions through a standard Internet connection. A common configuration would be to first run background extraction on this input video stream, and then apply super resolution on top of it, to bring the video dimensions up to 1920 x 1080, which is a

format commonly used in TV broadcasting today. Running the modules in this order will require less resources when compared to applying super resolution first, as the background extraction module can work at a lower resolution, making continued real-time operation more reliable.

In any event, the proposed system is capable of running both configurations as well as additional processing steps of the pipeline (like decoding the input stream) at a stable 25 FPS with hardware reported in subsection 5.3, allowing for its usage in a real TV studio setting such as in the example shown in Fig. 7 that comes from a trial carried out at NRK studio.

Figure 7. Result of the entire AdMiRe pipeline in a real TV studio setting (NRK, Norwegian public TV). The remote participant in the composited frame is being filmed with a regular smartphone without a green screen.

6. CONCLUSION

This paper introduced a new solution and underlying architecture that allows remote participants to be teleported to a studio environment by using their own smartphones or webcams, and to process the captured video in order to produce a high-quality composited video ready for broadcast. The end-to-end architecture of the proposed system was described, with particular emphasis on two key components, namely, the background extraction and the super-resolution modules. The latter have been combined and validated by end user partners in the AdMiRe project. The experiments show that the proposed solution performs well in real-life situations resulting in good quality composited video in the entire pipeline. The source code of the background extraction and super resolution modules, as well as the necessary interfaces for their integration in a broadcast environment have been made publicly available in open source and can be accessed from [‡] Future work includes further improvement and optimization through training of learning based module for background extraction that can achieve higher quality results for resolutions beyond 4K that are expected to become increasingly popular in broadcast applications. Further optimization through training of learning based super-resolution module that takes into account distortions due to streaming can also further improve the results. Last but not least, in case of successful deployment of upcoming learning based compression standards such as JPEG AI, one can consider performing compression, processing and computer vision operations in the compressed domain (latent space) and achieve at the same time better performance in complexity and quality at the same time.

[‡] https://github.com/mmspg/AdMiRe_EPFL

7. ACKNOWLEDGEMENT

The authors will like to acknowledge contributions from the H2020 Innovation Action "Advanced Mixed Realities (AdMiRe)" under grant agreement 952027. Valuable interactions and feedback received from project partners during integration, testing and validation of the modules reported in this paper are also acknowledged.

REFERENCES

[1] Porter, T. and Duff, T., "Compositing digital images," in [*Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*], *SIGGRAPH '84*, 253–259, Association for Computing Machinery, New York, NY, USA (1984).

[2] Zhu, Q., Heng, P. A., Shao, L., and Li, X., "What's the Role of Image Matting in Image Segmentation?," in [*2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*], 1695–1698, IEEE, Shenzhen, China (Dec. 2013).

[3] Chuang, Y.-Y., Curless, B., Salesin, D., and Szeliski, R., "A bayesian approach to digital matting," in [*Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*], **2**, II–II (2001).

[4] Gastal, E. S. L. and Oliveira, M. M., "Shared sampling for real-time alpha matting," *Computer Graphics Forum* **29**(2), 575–584 (2010).

[5] Xu, N., Price, B., Cohen, S., and Huang, T., "Deep image matting," in [*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 311–320 (2017).

[6] Sengupta, S., Jayaram, V., Curless, B., Seitz, S. M., and Kemelmacher-Shlizerman, I., "Background matting: The world is your green screen," in [*2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 2288–2297 (2020).

[7] Lin, S., Ryabtsev, A., Sengupta, S., Curless, B., Seitz, S., and Kemelmacher-Shlizerman, I., "Real-time high-resolution background matting," in [*2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 8758–8767 (2021).

[8] Zhu, B., Chen, Y., Wang, J., Liu, S., Zhang, B., and Tang, M., "Fast deep matting for portrait animation on mobile phone," in [*Proceedings of the 25th ACM International Conference on Multimedia*], *MM '17*, 297–305, Association for Computing Machinery, New York, NY, USA (2017).

[9] Ke, Z., Sun, J., Li, K., Yan, Q., and Lau, R. W., "Modnet: Real-time trimap-free portrait matting via objective decomposition," in [*AAAI*], (2022).

[10] Lin, S., Yang, L., Saleemi, I., and Sengupta, S., "Robust high-resolution video matting with temporal guidance," in [*2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*], 3132–3141 (2022).

[11] Wu, H., Zheng, S., Zhang, J., and Huang, K., "Fast end-to-end trainable guided filter," in [*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 1838–1847 (2018).

[12] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., "Mobilenetv2: Inverted residuals and linear bottlenecks," in [*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 4510–4520 (2018).

[13] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., and Le, Q., "Searching for mobilenetv3," in [*2019 IEEE/CVF International Conference on Computer Vision (ICCV)*], 1314–1324 (2019).

[14] Ballas, N., Yao, L., Pal, C., and Courville, A. C., "Delving deeper into convolutional networks for learning video representations," in [*4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*], Bengio, Y. and LeCun, Y., eds. (2016).

[15] "Nvidia maxine." https://developer.nvidia.com/maxine.

[16] Dong, C., Loy, C. C., He, K., and Tang, X., "Image super-resolution using deep convolutional networks," (2015).

[17] Kim, J., Lee, J. K., and Lee, K. M., "Accurate image super-resolution using very deep convolutional networks," (2015).

[18] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," (2014).
[19] Dong, C., Loy, C. C., and Tang, X., "Accelerating the super-resolution convolutional neural network," (2016).
[20] Anwar, S., Khan, S., and Barnes, N., "A deep journey into super-resolution: A survey," (2019).
[21] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M., "Enhanced deep residual networks for single image super-resolution," (2017).
[22] Kim, J., Lee, J., and Lee, K. M., "Deeply-recursive convolutional network for image super-resolution," (11 2015).
[23] Choi, J.-S. and Kim, M., "A deep convolutional neural network with selection units for super-resolution," in [*2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*], 1150–1156 (2017).
[24] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W., "Photo-realistic single image super-resolution using a generative adversarial network," (2016).
[25] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C. C., Qiao, Y., and Tang, X., "Esrgan: Enhanced super-resolution generative adversarial networks," (2018).
[26] Bevilacqua, M., Roumy, A., Guillemot, C., and line Alberi Morel, M., "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in [*Proceedings of the British Machine Vision Conference*], 135.1–135.10, BMVA Press (2012).
[27] Zeyde, R., Elad, M., and Protter, M., "On single image scale-up using sparse-representations," in [*Curves and Surfaces*], Boissonnat, J.-D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M.-L., and Schumaker, L., eds., 711–730, Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
[28] Huang, J.-B., Singh, A., and Ahuja, N., "Single image super-resolution from transformed self-exemplars," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (June 2015).
[29] Martin, D., Fowlkes, C., Tal, D., and Malik, J., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in [*Proc. 8th Int'l Conf. Computer Vision*], **2**, 416–423 (July 2001).
[30] Agustsson, E. and Timofte, R., "Ntire 2017 challenge on single image super-resolution: Dataset and study," in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*], (July 2017).
[31] Kappeler, A., Yoo, S., Dai, Q., and Katsaggelos, A., "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging* **2**, 1–1 (06 2016).
[32] Wang, X., Chan, K. C. K., Yu, K., Dong, C., and Loy, C. C., "Edvr: Video restoration with enhanced deformable convolutional networks," (2019).
[33] Tian, Y., Zhang, Y., Fu, Y., and Xu, C., "Tdan: Temporally deformable alignment network for video super-resolution," (2018).
[34] Liu, H., Ruan, Z., Zhao, P., Dong, C., Shang, F., Liu, Y., Yang, L., and Timofte, R., "Video super resolution based on deep learning: A comprehensive survey," (2020).
[35] Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T., "Video enhancement with task-oriented flow," *International Journal of Computer Vision (IJCV)* **127**(8), 1106–1125 (2019).
[36] Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., and Lee, K. M., "Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study," in [*CVPR Workshops*], (June 2019).
[37] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," (2016).
[38] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," (2015).
[39] Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., Zhang, L., et al., "Ntire 2017 challenge on single image super-resolution: Methods and results," in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*], (July 2017).
[40] Jolicoeur-Martineau, A., "The relativistic discriminator: a key element missing from standard gan," (2018).

[41] Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., and Zelnik-Manor, L., "The 2018 pirm challenge on perceptual image super-resolution," (2018).
[42] Ma, C., Yang, C.-Y., Yang, X., and Yang, M.-H., "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding* **158**, 1–16 (2017).
[43] Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools* (2000).
[44] MATLAB, [*version 7.10.0 (R2010a)*], The MathWorks Inc., Natick, Massachusetts (2010).
[45] Testolina, M., Upenik, E., Ascenso, J., Pereira, F., and Ebrahimi, T., "Performance evaluation of objective image quality metrics on conventional and learning-based compression artifacts," in [*2021 13th International Conference on Quality of Multimedia Experience (QoMEX)*], 109–114 (2021).
[46] ITU-R BT.709-6, "Parameter values for the HDTV standards for production and international programme exchange." International Telecommunication Unionn (Jun. 2015).
[47] Blau, Y. and Michaeli, T., "The perception-distortion tradeoff," 6228–6237 (06 2018).