Universal Access in the Information Society
https://doi.org/10.1007/s10209-025-01255-x

LONG PAPER

Human pose estimation solutions: a low cost tool for increasing natural interaction in virtual television sets

Rubén Arenas[2] · Roi Méndez[1] · Luis Pedraza[3] · Enrique Castelló[1] · Julián Flores[2]

Accepted: 4 August 2025

Abstract

Virtual television sets (VTS) have experienced a growth spurt in recent years, paralleling the development of virtual reality tools and the metaverse. Nowadays, this technology is used in multiple broadcasts from major television companies to smaller tv stations ranging from sports programs to election nights, news, etc. Despite significant advances in recent years, such as improvements in real and virtual content composition or real-time realistic rendering, interaction between presenters and virtual content remains a challenge. This is primarily due to the high cost and complexity of the equipment required, as well as the limitations imposed by live production technology. In this context, we propose testing Human Pose Estimation (HPE) tools over the studio camera stream as a potential solution that does not require additional hardware integration into the system. We evaluated 14 HPE solutions using a three-step process. First, we assessed the robustness and viability of reliable real-time execution for each solution, with five solutions passing this initial phase. Secondly, we analyzed frames per second (FPS), RAM consumption, and CPU usage for each alternative in both local and global scenarios, considering both the 'printmetrics' and 'no view' options. BlazePose OpenVINO demonstrated the best performance in these tests and was selected for further testing in real-world scenarios. These tests have confirmed that HPE is a viable alternative for enhancing human-computer interaction in VTS. However, certain limitations remain, such as the lack of reliable depth data and the need for further analysis in detecting complex dynamic gestures. The proposed software-based VTSs promotes universal accessibility by eliminating the need for external control devices, reducing economic barriers and allowing users to customise natural, adaptive interactions that fit their individual capabilities and contextual needs.

Keywords Human pose estimation · Virtual television sets · Human computer interaction · Benchmarking

1 Introduction

Commercial Virtual Television Sets (VTS) made their commercial appearance in 1994 at the IBC in Amsterdam. A boom followed this event and lasted until the early 2000s, when virtual reality and related technologies fell out of the spotlight. With the return of these technologies to the mainstream, starting with the presentation of the Oculus Rift glasses in 2012, and the huge hardware and software development that followed this event until today, virtual TV sets have regained unprecedented importance. Nowadays, almost all television networks make use of this technology in entertainment programs, election nights, sports programs, news programs, etc. The basis for this success is the quality of the final image achieved in real time, as well as the unlimited creative possibilities offered using computer-generated content. Today's hardware and software advancements mean that image quality is no longer an insurmountable

Rubén Arenas, Roi Méndez, Luis Pedraza, Enrique Castelló and Julián Flores have contributed equally to this work.

✉ Rubén Arenas
ruben.arenas.hernan@gmail.com

✉ Roi Méndez
roi.mendez@usc.es

Luis Pedraza
luis.pedraza@unir.net

Enrique Castelló
enrique.castello@usc.es

Julián Flores
julian.flores@usc.es

[1] Communication Sciences Department, Universidade de Santiago de Compostela, Santiago de Compostela, Spain

[2] Electronics and Computer Science Department, Universidade de Santiago de Compostela, Santiago de Compostela, Spain

[3] International University of La Rioja, Logroño, Spain

Published online: 19 August 2025

barrier and attention can be focused on other elements such as the natural interaction of the presenter with the environment or the creation of realistic and striking effects.

In this sense, two fundamental elements must be taken into account to improve the perceived presence of a real actor in virtual environments: the quality of image integration between the actor and the synthetic scene, and the natural interaction that the actor can develop with virtual elements. As we pointed before, nowadays, the quality and integration of virtual and real images are exceptionally high; however, the interactive capabilities of presenters in VTS productions remain limited. This limitation is primarily due to the high cost and complexity of equipment required for their implementation in live broadcasts, as the precision and performance of the systems is key to their viability. Consequently, most productions restrict interaction to choreographed movements or events triggered by remote controls, attempting to simulate natural interaction, which is commonly inexistent.

Over the past decade, human body pose estimation (HPE) solutions have made significant advancements, largely due to progress in real-time artificial intelligence and new devices designed for capturing human movements. The application of these solutions in VTS could potentially replace expensive human tracking hardware with low-cost standard devices and software solutions, thereby facilitating access to natural interaction possibilities in smaller productions.

This paper presents a benchmarking analysis of human body pose estimation solutions applied to VTS. We explore markerless human body tracking solutions using free or low-cost software techniques and hardware already available in a virtual studio, such as the standard video cameras used in the recordings. Through various software combinations, we investigate ways to simplify and enhance natural user interaction within VTS environments. The primary contribution of this paper is to determine whether the latest HPE technologies are sufficiently mature for their inclusion in live broadcasts across professional, medium, and small production sets. Moreover, we compare different available solutions and determine which one is the best for further testing in a real studio focusing on their speed, stability, and hardware requirements. Finally, we conducted a prospective evaluation of the possibility of using the chosen technique for its implementation in a VTS focusing on its ability to track the presenter in three dimensions and its ability to detect gestures using the estimated pose captured data. Therefore, by eliminating reliance on specialised external controllers, this approach promotes more financially sustainable systems, while allowing users to configure and adapt interaction modalities to their own physical, cognitive or contextual conditions, ultimately improving both usability and inclusion in VTS productions.

Section 2 presents a brief state of the art of the virtual television set technology, mainly focusing on the evolution of actor interaction with virtual environments. Section 3 presents the main solutions for HPE applicable to VTS using low-cost or already available hardware. The following section outlines the methodology, materials, and methods used in this study. Finally, Sects. 5, 6, and 7 are dedicated to results, Sect. 8 focuses on discussion and Sect. 9 summarizes our conclusions.

2 Virtual televison sets

The origin of virtual television sets (VTS) dates back to the 1930s when Goldsmith described a method of video compositing in their patent application for a "Television System". This method involved "insetting" information from one camera into the image captured by another camera [1]. The term "inset" refers to the process of segmenting a region of an image from one camera and superimposing it onto a specific area of an image from a second camera. This technique enabled the simulation of an actor's presence [2] in a location where they were not physically present. The first modern system can be considered Synthevision [3], presented in 1980. This pioneering system used analog sensors to synchronize camera movements between real and virtual worlds, enabling broadcast-quality virtual compositions with camera movement. Since then, many solutions have been proposed for each of the challenges faced by the technology [4].

Nowadays, a Virtual Television Studio (VTS) combines real-time computer-generated graphics with live footage generally using chroma-keying techniques. These techniques remove the monochrome background and elements of a stage, typically colored in blue or green, while preserving other elements such as actors or atrezzo which are of a different color [5] (Fig. 1). Then, the removed parts of the captured image are replaced by computer-generated content. Recently, chroma-keying has been partially replaced by LED screens. However, this technology presents some drawbacks that will make both coexist in the next years. For a VTS to work, it is necessary to have a camera to record the real world, a PC to render the virtual elements, sensors to keep both worlds coherent (camera movements, human-computer interaction, etc.) and a video mixer to select the video to be broadcasted and include effects.

However, despite this basic equipment, the level of presence of the real elements in the virtual environment that the viewer perceives depends on several factors related to more complex and advanced technology. Some of these factors

Fig. 1 Virtual Television Set stage, before and after image composition

are visual coherence, visual realism and natural interaction between virtual and real contents [6]. Visual coherence refers to the proper mix between virtual and real worlds, involving camera alignment (even with live camera movements), equal illumination (including shadow casting) and proper chroma-keying (including color spill correction). On the other hand, modern graphics hardware and render engines can produce highly realistic 3D real time graphics and special effects for virtual scenery and props. Global illumination algorithms can now be executed in real time using specific graphics cards such as the Nvidia RTX family [7]. Moreover, render engines such as Unreal Engine can render real time environments composed of millions of polygons with realistic illumination using techniques such as Nanite [8] and Lumen [9]. Besides, modern chroma-keying techniques allow the projection of real shadows over virtual elements, achieving the seamless composition between the real and virtual worlds. However, to maintain believability and improve the communicative capabilities of the presenters, they should be able to interact with the virtual environment similarly to how they would in a real studio. This includes triggering events, grabbing virtual objects, colliding with virtual elements and performing other basic interactive tasks. These interactions will help, along with a visually credible image composition, to create a seamless blend between the real and virtual contents, improving the viewer's perception of presence and enhancing the overall viewing experience.

This technology allows the creation of complex, dynamic virtual environments that can be easily modified and adapted for different productions, enhancing viewer engagement and program retention. The integration of AR and gaming technologies has made virtual studios more accessible and versatile, enabling broadcasters to create sophisticated visual presentations without the need for expensive physical sets. This not only improves the on-air look but also offers cost-effective solutions for broadcasters to enhance their programming and potentially improve their bottom line.

Recent developments in virtual studio technology include new metaphors such as the Motion Scene Camera proposed by Hach et al. [10] for natural real-time interaction in professional productions using custom sensors and hardware. Augmented Reality Integrations in these kinds of systems are also presented by Cho et al. [11] using low-cost equipment for real-time content generation. In the same line of work, several authors propose the use of devices and techniques inherited from the video game industry like in Goussencourt and Bertolino [12], who utilized the Microsoft Kinect v2 sensor to capture depth information, improving alignment between virtual and real elements, Mendez et al. [13] who developed a system supporting various kind of sensors through an open middleware and the VRPN protocol testing devices such as Optitrack IR cameras, Microsoft Kinect (V1 and V2), and Leap Motion, or the ARstudio developed by Aguilar et al. [14], presenting a solution that enables interaction with virtual elements through tangible interfaces using AR techniques. Despite all these advancements, which have the potential to significantly improve the realism and interactivity of virtual studios, the reality is that the industry is reluctant to incorporate then to its workflow due to their cost or lack of reliability. The goal of this work is to determine if a low-cost solution such as HPE algorithms offers a robust solution in terms of human tracking and gesture detection to improve interactivity in VTS.

It is significant for this work to point out the software used for our experiences, called InfinitySet, developed by the company Brainstorm Multimedia. This professional solution enables the creation of real-time virtual TV studio productions, offering advanced capabilities such as real-time alignment of virtual and real cameras, instant interaction with actors, and a 3D representation of presenters. It also allows free virtual camera movements and the implementation of innovative effects like shadow casting and reflections within the virtual environment due to its compatibility with Unreal Engine. Additionally, InfinitySet includes hand tracking technology that enables users to trigger predefined animations and interact with graphics through a limited range of actor movements. However, this interaction is not precise, and it is not possible to implement new gestures besides the ones already included. Moreover, there are other limitations to this software that could be overcome by the use of HPE algorithms. One significant issue is the restricted movement of presenters as their position relative to virtual elements does not update automatically. The presenter can be placed in front of a virtual object or behind it and there is not an automatic approach included in the software to update the

relative positions while the presenter moves in the stage. This can lead to visual inconsistencies, as the presenters could be represented in front of an object when he should be placed behind it and vice versa. Moreover, when using virtual camera movements, the presenter is represented as a 3D object. In these scenarios, not knowing the location of the presenter can lead to errors such as the appearance of presenters floating above the floor or sinking into it. Furthermore, incorporating new sensors into these systems tends to be both expensive and complex. The proprietary and monolithic nature of professional VTS software like InfinitySet complicates modifications and the integration of new sensors for end-users, often requiring specialized development by third parties. To address these challenges and make VTS technology more accessible to smaller TV and online content producers, this work proposes the use of devices which are already available in traditional VTS setups, such as recording cameras, to enhance their interactive capabilities. By integrating Human Pose Estimation algorithms, we propose to develop low-cost or no-cost solutions that improve both affordability and usability. This approach is essential to ensure that this technology is available to a broader range of content creators and offers extended interactive possibilities to professional setups.

3 Human pose estimation

Human Pose Estimation (HPE) is a computer vision technique that detects and tracks the position and orientation of human body parts in images or video streams. This process involves analyzing sequences of images to identify patterns indicating the presence of specific body parts, resulting in a set of hierarchically ordered points that determine the pose of one or more individuals. HPE can provide 2D or 3D information depending on the sensor used for video capture. The output is typically represented as simplified human skeletons, though other representations are possible. Two strategies are typically used in HPE [15, 16]:

- Top-Down Strategy: This approach moves from global to particular analysis. It first uses a person image detector to identify regions potentially containing bodies, then applies HPE methods to detect the pose within each region;
- Bottom-Up Strategy: This method progresses from particular to global detection. It initially analyses the image to find potential body part candidates, then links these points according to hierarchical body representation model constraints.

Recent research [16–22] has demonstrated that applying deep learning techniques to HPE significantly enhances the tracking performance of process. For Virtual Television Studio (VTS) applications, AI solutions using standard RGB video sources are particularly relevant, as existing cameras could serve as tracking sensors.

Deep learning-based HPE methods typically use convolutional neural networks (CNNs) to extract features from input images. These networks can learn complex patterns and representations, making them more effective than classical feature approaches. Popular deep learning models for HPE include OpenPose, High-Resolution Net (HRNet), and DeepCut, among others.

These advanced techniques have improved the accuracy and robustness of pose estimation, enabling better handling of challenges such as occlusions, varying lighting conditions, and complex poses; expanding the potential applications of HPE in fields like augmented reality, sports analysis, healthcare, and, of course, actors tracking in a VTS.

The following is a brief description of the most relevant libraries and toolkits that implement this type of algorithms:

- OpenPose [17, 23] is a real-time multi-person human pose detection library that detects key points of the body, feet, hands, and face from a real-time video source using a Bottom-Up strategy. During the tracking process, it detects and groups the key points of human body parts (called Part Affinity Fields). These elements are linked to a set of 2D vector fields that determine the position and orientation of the different limbs of the people captured in the image;
- Wrnch Body Slam [15] is a library that tries to simulate the skills of human vision, allowing the development of applications that attempt to understand the movement, shape, and intention of tracked people. Wrnch allows real-time tracking of people's movement, gesture recognition and detection of human activity;
- DensePose [24] task is a part of COCO and Mapillary Joint Recognition Challenge Workshop at ICCV 2019. It is an HPE solution developed by Facebook's artificial intelligence research division (FAIR) that focuses on body estimation by mapping all human pixels of an RGB image to the 3D surface of the human body;
- PoseNet [22, 25, 26] is an HPE tool co-developed by TensorFlow and Google that is based on the use of a truncated and modified GoogLeNet architecture in which the softmax classification is replaced by a sequence of fully connected layers to generate the absolute pose contained in an image. PoseNet was the first machine learning-based architecture that introduced the idea of performing absolute pose regression with deep learning;

- MediaPipe [27] is a set of solutions that includes pre-trained and optimized models that accelerate tracking processing even on common hardware. MediaPipe Pose can detect 33 key points of the whole body in RGB images using the BlazePose model;
- EfficientPose [28] is a convolutional neural network architecture that uses the EfficientNets model to perform one-man tracking. It has been one of the most widely applied HPE methods in real world applications on next-generation devices, as it limits the memory footprint and computational cost.

These are some examples of toolkits that could be used in a VTS. However, the evolution of this technology, very related to the development of new IA strategies, is continuous. For this reason, in this paper we look for the best solutions to perform preliminary tests that provide a theoretical and practical foundation for future implementations. We try to determine the potential of the technology in the field of VTS using the best available solution nowadays.

4 Methodology and materials

We propose a two-step methodology. The first step focuses on determining the best HPE solution for its implementation in a VTS and the second one on analyzing and evaluating the use of the selected solution as it is provided in a real production environment.

These two steps have been carried out in the professional Virtual Television Studio (VTS) located in the Faculty of Communication Sciences of the University of Santiago de Compostela which first was designed and built in 2010 [29] and updated in 2024, currently managed by the VIRTUS, the Virtual Research Platform of the University of Santiago de Compostela. The primary tests were performed in the original studio whereas the final performance and gesture detection tests were executed in the new facilities (Fig. 2).

The original studio comprises two cameras equipped with mechanical tracking systems on their pedestals, each paired with a rendering PC that utilizes their data to generate a coherent perspective in the virtual environment. The chroma-keying process is handled by dedicated hardware devices (chroma-keyers) for each camera-PC pair. A video mixer allows for the selection of the broadcast signal and adds other production effects. The software in charge of the render is eStudio, developed by Brainstorm Multimedia.

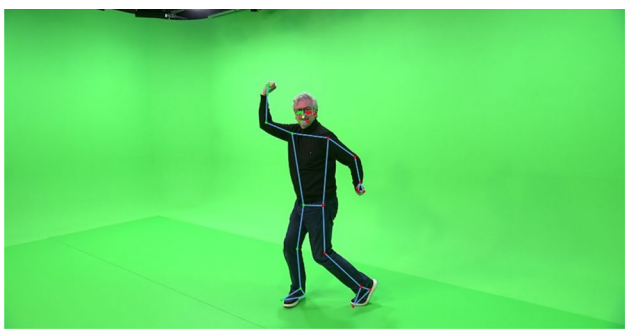

Fig. 2 Example of image obtained in real time

The new studio is equipped with two remotely operated PTZ (pan-tilt-zoom) cameras and a studio camcorder with an inside-out Stype tracking system and teleprompter. Each camera is paired with a render PC which is also responsible for the chroma-keying, which is done by software. The studio is also equipped with a control PC, a PC for the titling machine, a PC for content preproduction, a video player and recorder PC, an audio mixer and a video mixer. This equipment allows the production of professional content with a traditional three-camera production. The software used is InfinitySet, developed by Brainstorm Multimedia, which offers full compatibility with Unreal Engine.

For the first step of this study, focused in comparing different HPE solutions and their performance in a live production, we use the original studio camcorders as video sources, processing their streams to track the presenter's body, effectively turning the cameras into tracking sensors without the need for any additional hardware. The camera model used in the original VTS is the Sony XDCAM EX HD, which offers recording capabilities in 1080i (1920 x 1080 pixels) and 720p (1280 x 720 pixels) with frame rates of 59.94, 50 (interlaced), and 23.98 (progressive). For the second step in this research, focused on analyzing the actual viability of the selected solution in a real production environment, the studio camcorder of the new VTS has been used. This camera is a Panasonic AK-HC3900 with the Fujinon lens ZA12 x 45BRD-S6, capable of recording video at 1080p (1920 x 1080 pixels) with framerates of 25 frames per second.

Both setups allow for high-quality video capture and processing, essential for accurate pose estimation and seamless integration of virtual elements. The use of existing studio cameras as tracking sensors represents an innovative approach to enhance VTS capabilities without a significant additional investment in hardware, as the software solution can be run in the already available render PCs.

The HPE tests were conducted on a PC running Windows 10, with an 8th generation Intel Kaby Lake Core i7-8750 H CPU (2.2GHz), 16GB of RAM, and an NVIDIA GTX 1060 graphics card with 6GB-DDR5 memory.

4.1 Test and acceptance criteria

The research methodology for evaluating Human Pose Estimation (HPE) algorithms in the context of Virtual Television

Studio (VTS) applications involves two types of tests: the Local Test and the Global Test.

The Local Test serves as an initial screening process to assess the performance of HPE libraries or toolkits in isolation from the VTS environment. This preliminary evaluation is crucial because it allows researchers to identify and eliminate poorly performing solutions before integrating them into the real VTS setup. Additionally, it optimizes the testing process since evaluating algorithms directly on the virtual set can be time-consuming and the studio may not always be available. By conducting these local pre-tests, researchers can narrow down a large number of HPE algorithms to the most promising candidates for their specific scenario.

During the Local Test, the HPE system is evaluated using a video streaming dataset with a simulated 5ms delay, which mimics the maximum lag expected when transmitting information over the local network in a real VTS environment.

For this purpose, we use the open access dataset MPI-INF-3DHP [30]. It is 3D human body pose estimation dataset consisting of constrained indoor and complex outdoor scenes. It records 8 actors performing 8 activities from 14 cameras in several different categories. 16 different sequences selected from the Image Sequence Category were chosen, which include only indoor scenes. From the 16 sequences, the videos from camera number 8 were finally selected because of their similarity in framing to that commonly used in a VTS. The original sequences have 1:1 square shape unlike most professional cameras, which have 16:9 or 4:3 formats. In our case 16:9, 1080i (1920 × 1080 pixels) or 720p (1280 × 720 pixels). Therefore, pre-processing of the data has been necessary, so that the performance results match those that could be found in a real time TV broadcast. A resampling has been carried out to obtain a stream as close as possible to those that could be found in our VTS but keeping the aspect ratio to avoid image deformations that would affect the HPE process. Thus, the stream was resized to a square resolution of 720 by 720 pixels using WinX HD Video Converter Deluxe software (Digiarty WinX DVD, 2021). After this process, our final dataset is composed of 16 videos corresponding to the 2 sequences recorded for the 8 actors (1:1 720p). The length of the video ranges from 4 min and 2 s–4 min 27 s, for a total of 67 min of video.

Performance thresholds are also established to determine which algorithms are suitable for real-time applications. Only those HPE solutions that demonstrate sufficient performance in the Local Test are considered for further evaluation in the Global Test. This two-step approach ensures efficient use of resources and focuses on enhancing VTS capabilities with the most effective HPE solutions.

The Global Test is a comprehensive evaluation conducted in a live broadcast scenario within the Virtual Television Studio (VTS). This test assesses the HPE system's integration and impact on the overall VTS architecture during a live TV production. Two test options were introduced to address the potential performance impact of visual monitoring:

- "Noview": Tracks actors without any visualization, sending results directly to the system to minimize additional processing;
- "Printmetrics": Displays the in-process frame, detected skeleton, and performance data on-screen for users.

Once the tests have been proposed, we need to establish the HPE acceptance criteria. To set these criteria, we defined a list of basic requirements that an HPE system must meet in order to be included in a VTS. These criteria are the result of the experiences carried out by the VTS operators over the years:

1. The VTS must produce a final video broadcast or streaming with virtual and real elements consistent with at least 1080p resolution at 30 frames per second;
2. All VTS elements must be robust and guaranteed to work in real-time broadcast conditions;
3. HPE elements must have minimal latency when integrated into the VTS rendering pipeline to ensure coherence between triggered actions and actor movements;
4. VTS elements should be isolated to prevent workflow interference, a requirement addressed by the architecture presented in previous research [31].

These criteria ensure that the HPE system can effectively enhance the VTS capabilities without compromising the quality or reliability of live broadcasts.

The research methodology for evaluating Human Pose Estimation (HPE) systems in Virtual Television Studio (VTS) applications involves rigorous testing to ensure real-time performance and reliability. The key performance indicator is frame rate, which must be consistently over 25 fps to meet the demands of live broadcasting (thus the limit is placed in 30 to avoid drops below 25 frames per second). However, in requirement 3 we introduce additional empirical parameters that must be considered to verify the overall system performance. For example, if the final user is not able to perceive delays or other issues during the broadcast, the system is considered to be working properly although it may not meet completely the theorical thresholds. The latency must not be perceived by the viewer. We place this interactive threshold, in terms of visual interactivity, in 10 fps. Finally, in requirement 2 we refer to the fact that there must be a margin for undesired situations. To address this point, it has been decided to monitor the amount of memory

Table 1 Description of the 3 desirable gestures defined by the experts as candidates for their use in VTS

Name	Description
Pose 1	Point the left hand to the left. The right arm should be relaxed
Pose 2	Point the right hand to the right. The left arm should be relaxed
Pose 3	Point with the right hand towards the sky. The left arm should be relaxed

Table 2 HPE solutions initially selected for testing

Name	Core solution	Correct install	Run	Stable
OpenPose oficial	OpenPose	No	–	–
hpeOpenCv	OpenPose	Yes	Yes	No
EfficientPose	EfficientPose	Yes	Yes	No
MediaPipeTest	MediaPipe	Yes	Yes	Yes
AlphaPose	AlphaPose	No	–	–
Detectron2	DensePose	No	–	–
Simple Pose	AlphaPose+GluonCv	Yes	Yes	No
Wrnch	Wrnch AI Body Slam	No	–	–
posenet-tf	PoseNet+Tensorflow	No	–	–
hpe3D	XnectVnect+Pytorch	No	–	–
PoseNetPython	Posenet Tensorflow.js	Yes	Yes	No
AlphaPose	AlphaPose + Pytorch	No	–	–
Drone-vision	Posenet	Yes	Yes	Yes
BlazePose-OpenVino	OpenVino+MediaPipe	Yes	Yes	Yes

used and the CPU workload. In this sense, solutions with lower memory and CPU are the best candidates.

Several meetings have been held with TV presenters to determine which basic poses could be included in the test to study the pose recognition performance of the HPE. Three poses have been described as candidates to be used in a broadcast production which are tested during this study (Table 1).

Finally, we perform an analysis of the capabilities of the selected solution in the VTS. Considering the results of the local and global tests and testing the selected solution in a real production scenario, we discuss the advantages and disadvantages of implementing this technology in a professional VTS.

5 Results

Firstly, an exhaustive search for possible open source HPE candidates was carried out, and 14 solutions were identified as potential candidates (Table 2). Even though, a priori, the implementations could exceed the requirements presented earlier, nine of them could not be used for different reasons. The main problems encountered were:

1. No Windows 10 version. As mentioned above, for compatibility reasons with the rest of the VTS components, we needed the HPE algorithm to have a Windows 10 version, which was not the case in some of the solutions analyzed;
2. Local execution. Some solutions need to access on disk or download information from the network at runtime, which makes them impossible to use in our real-time environment.

Table 2 presents the results of the first step of the process, which consisted of getting the solution consistently running in local execution. We consider that a toolkit is correctly installed when it is possible to install it on a windows 10 operative system. The solution is runnable if the application runs, and stable if it does so without errors, delays, stops or any kind of artifact. Some solutions presented glitches and errors such as sudden application crashes, which could not be fixed. Among the 14 shortlisted solutions, only 5 have been selected as candidates for benchmarking: hpeOpenCV, MediaPipeTest , PoseNetPython , drone-vision and BlazePose-OpenVino. These five candidates will undergo the local test printMetic, local test No-view and global test, where, as mentioned, the frames per second, memory use and required CPU will be assessed.

5.1 Local test-printMetrics

In these initial tests, several key results were obtained. Firstly, based on expert experience, it was determined that if the frame rate of the Human Pose Estimation (HPE) did not drop below 10 fps, the gestural body tracking interaction between the actors and the 3D environment during live broadcast did not present noticeable visual artifacts from the spectator's point of view. This aligns with the fact reported by different research such as [32, 33] which prove that, when the frame rate is higher than 10 frames per second, the sensation of presence is not disturbed in VR applications. However, the Optimal results correspond to a high FPS, along with low memory (MEM) and CPU usage.

Table 3 shows the local test results, providing information about the process. Analyzing the table reveals that hpeOpenCV (FPS_min = 1.2, FPS_avg = 2.7, FPS_max = 4.4) and PoseNetPython (FPS_min = 2.6, FPS_avg = 7.6, FPS_max = 10.5) have values below real-time performance. The other three solutions-MediaPipe (FPS_min = 12.5, FPS_avg = 53.4, FPS_max = 68.0), Drone-Vision (FPS_min = 11.5, FPS_avg = 34.5, FPS_max = 68.6), and BlazePose-OpenVino (FPS_min = 11.8, FPS_avg = 85.6, FPS_max = 135.0) maintain frame rates above 10, with BlazePose-OpenVino being the best-performing solution with an average frame rate of 85.6 f/s. It is important to note

Table 3 Values resulting from the local test with printmetrics option enabled and noview option disabled

Name	hpeOpenCv	MediaPipeTest	PoseNetPy	Drone-vision	BlazePose-OpenVino
FPS_min	1.2	12.5	2.6	11.5	12.8
FPS_avg	2.7	53.4	7.6	34.5	85.6
FPS_max	4.4	68.0	10.5	68.6	135.0
MEM_min	253.1	163.4	271.5	2614.8	170.9
MEM_avg	264.4	175.5	280.2	2623.5	201.6
MEM_max	269.8	178.6	281.7	2627.2	202.9
CPU_min	15.6	2.8	7.6	2.4	2.9
CPU_avg	45.8	10.6	42.7	13.4	28.9
CPU_max	80.8	78.3	88.0	90.2	103.0

Table 4 Values resulting from the local test with printMetrics option disabled and noView option enabled

Name	hpeOpenCv	MediaPipeTest	PoseNetPy	Drone-vision	BlazePose-OpenVino
FPS_min	0.7	10.0	2.3	9.9	15.3
FPS_avg	4.6	31.6	6.6	23.6	54.7
FPS_max	6.0	60.9	9.1	57.8	77.8
MEM_min	226.9	159.7	252.8	2607.2	171.6
MEM_avg	249.1	169.6	260.9	2618.0	198.4
MEM_max	266.1	171.5	262.3	2622.1	199.8
CPU_min	8.1	2.0	7.1	2.0	2.5
CPU_avg	75.9	6.6	39.8	7.9	23.7
CPU_max	100.2	55.3	81.9	53.5	55.8

that these measurements were taken in a stable situation, without considering system start-up; however, the range of variability in the number of frames per second is very wide.

Examining the memory requirements for operation shows that the values are generally good, indicating that the hardware can support more computational load than required. Only the Drone-Vision solution has a significantly higher memory usage of 2600MB, which may limit its usage.

In terms of CPU usage information, only BlazePose-OpenVino reaches 100% usage at peak moments; however, generally, the maximum values are around 80% with an average value close to 40

Regarding point 5 of the requirements, it is possible to establish threshold values for memory and CPU load. We consider values lower than 500MB for memory and 35% for CPU usage as good indicators to accommodate potential new computational needs. These values are derived empirically from the daily work experience of the system users.

5.2 Local test-no view

The results of the local noView benchmark test, as shown in Table 4, indicate that, as expected, the values obtained are generally superior to those of the previous tests. However, the variation in results is heterogeneous. For the average FPS difference between local test printmetrics and local test no view, we have the following values: HpeOpenCV−1.9, MediaPipeTest 21.8, PoseNetPython 1.0, Drone-Vision 10.9, and BlazePose-OpenVino 30.9. As observed, in some cases the variation is very small, while in others it reaches up to 30 fps. The case of HpeOpenCV is particularly unusual because its values are lower than in the printmetrics tests.

To understand these results, various modifications to the implementation parameters were tested, but similar outcomes were consistently observed, and no explanation for this behavior was found. However, the results allow us to draw several conclusions. The most significant one is that the BlazePose-OpenVino implementation demonstrates the best overall performance, as the FPS obtained in all tests is much higher than that of the other analyzed solutions. Therefore, this solution emerged as the best candidate following the local tests. Nonetheless, all solutions were considered to verify their performance in the real VTS.

5.3 Global test

As discussed, for the global testing, the HPE system was included in the VTS and the video stream from the dataset was replaced with real-time video captured by the studio cameras. In this case, an actor moves in front of the camera to test the tracking performance of each implementation. Table 5 presents the results obtained. Before analysing the data, it should be considered that the size of the image to be analysed is bigger than the images used in the previous tests. In the local tests, 1:1 720p images were used whereas in the virtual study they are 16:9 1080i (1920x1080 pixels). This may result in a different frame rate performance, as the images to be analysed have different dimensions. However, parameters such as network traffic, and other VTS processes, as previously mentioned, reduce the perceived

Table 5 Values resulting from the global test

Name	hpeOpenCv	MediaPipeTest	PoseNetPy	Drone-vision	BlazePose-OpenVino
FPS_min	0.9	10.4	3.0	7.5	9.6
FPS_avg	2.3	29.5	7.5	26.5	29.8
PS_max	3.9	53.5	10.8	48.2	66.7
MEM_min	279.0	178.9	270.8	2633.1	184.1
MEM_avg	287.3	189.7	277.3	2643.8	219.6
MEM_max	288.2	190.8	280.4	2644.5	220.3
CPU_min	14.1	3.0	7.8	2.2	2.1
CPU_avg	42.0	11.7	42.0	15.2	15.9
CPU_max	75.0	56.7	90.8	97.7	52.1

Table 6 General results

Name	hpeOpenCv	MediaPipeTest	PoseNetPy	Drone-vision	BlazePose-OpenVino
Fps_avg test1	2.7	53.4	7.6	34.5	85.6
Fps_avg test2	4.6	31.6	6.6	23.6	54.7
Fps_avg test3	2.3	29.5	7.5	26.5	29.8
Mem_avg test1	264.4	175.5	280.2	2623.5	201.6
Mem_avg test2	249.1	169.6	260.9	2618.0	198.4
Mem_avg test3	287.3	189.7	277.3	2643.8	219.6
Cpu_avg test1	45.8	10.6	42.7	13.4	28.9
Cpu_avg test2	75.9	6.6	39.8	7.9	23.7
Cpu_avg test3	42.0	11.7	42.0	15.2	15.9

impact of this decrease in performance. Memory and CPU usage increases in value but not by a significant amount. It is important to note that the CPU usage of BlazePose-OpenVino is lower than the local test, which may be due to the size of the image to be analysed.

In the variable frames per second hpeOpenCV (FPS_min = 0,9, FPS_avg = 2,3, FPS_max = 3,9) and PoseNetPython (FPS_min =3, FPS_avg = 7,5, FPS_Max = 10,8) have values lower, not reaching real time performance as in previous tests. The other three solutions: MediaPipe (FPS_min =10,4, FPS_avg = 29,5, FPS_max = 53,5), Drone-Vision (FPS_min =7,5, FPS_avg = 26,5, FPS_max = 48,2) and BlazePose-OpenVino (FPS_min =9,6, FPS_avg = 29,8, FPS_max = 66.7) keep real time frame rates. Only Drone-Vision has a minimum value substantially under the threshold, but this was not a problem from a visual perception point of view.

6 General results

Table 6 and Fig. 3 summarize the average results obtained in the previous tests. First, it is important to note that two of the analyzed solutions do not meet the minimum frame rate required for real-time broadcasting applications: hpeOpenCV with 2.3 fps and PoseNetPython with 7.5 fps. In contrast, MediaPipeTest (FPS_avg test3 = 29.5 fps), Drone-Vision (FPS_avg test3 26.5 fps), and BlazePose-OpenVino (FPS_avg test3 29.8 fps) achieved excellent results, well above the minimum threshold of 10 fps and close to 30 fps, making them optimally compatible with broadcast environments. In these cases, producers did not detect any problems, artifacts, or visual delays in the final composite video stream.

When analyzing memory usage, MediaPipeTest (277.3 MB) and BlazePose-OpenVino (219.6 MB) are well-optimized and utilize resources moderately, with BlazePose-OpenVino being the best-performing solution in this regard. However, Drone-Vision (2643.8 MB) performed poorly on low-memory computers.

The average CPU usage indicates that BlazePose-OpenVino (15.9%) consumes fewer computational resources than MediaPipeTest (42%), making it the preferred choice for implementation in real VTS production environments.

7 Gesture detection

Following the tests, which determined that BlazePose-OpenVino obtained the best body tracking performance, the implementation of a body gesture detector, which integrates the gestures described in Table 1 was considered.

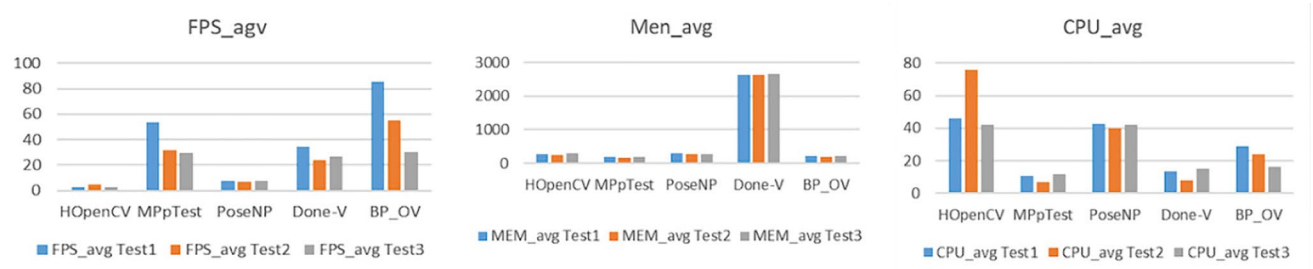

Fig. 3 General results, average values of frames per second, RAM memory usage and CPU usage resulting from tests

Fig. 4 Users performing gestures B, F and D from the semaphore alphabet

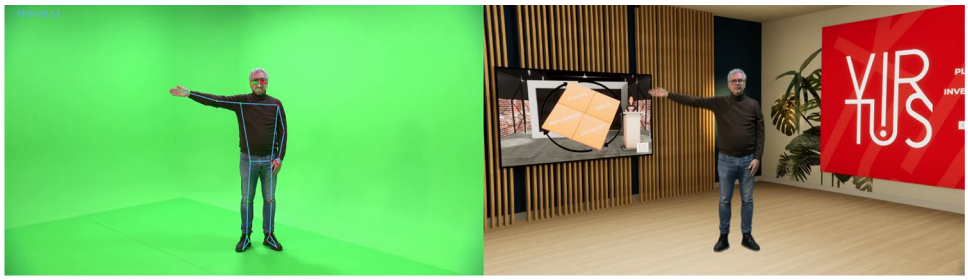

Fig. 5 User performing a gesture that fires the action play video on the virtual TV screen

BlazePose-OpenVino is provided with a base pose recognition system that measures the angles of the arm segments to determine which pose is performed by the user (the presenter in our case). The purpose of this research is to test the solutions as they are presented to the user, so we decided to use this system as it is for gesture detection tests.

The system establishes a correlation between the calculated angles and the octant into which a circle is divided give us a letter form the semaphore alphabet. It is a static detector which searches for poses of the users in a given frame. To implement the poses proposed by the experts (Table 1), this system has been used in a combined form. Thus, pose 1, responsible for recognizing the presenter's pose when he points to the right with his right hand, has been defined by the composition of the letter's 'A', 'B' and 'C' of the semaphore alphabet. Pose 2, in charge of recognizing the presenter's pose when he points to the left with his left hand, has been defined by the composition of the letter's 'E', 'F' and 'G' of the semaphore alphabet. And pose 3, responsible for recognizing the presenter's pose when he points with his right hand towards the ceiling, has been defined using only the letter 'D' of the semaphore alphabet (Fig. 4).

The tests have been performed in real time by 5 users. The images captured by the VTS studio were streamed live to the HPE computer and the events were triggered in InfinitySet by the Middleware developed in [13]. The three events proposed were: play a video on a screen (Fig. 5), trigger an animation and go to a new video in a playlist. The presenters were asked to present a program reading in a teleprompter and performing each of the three gestures at a given moment, so that the events were triggered consistently. Whereas the simplicity of the detection system led to a 100% success rate in gesture detection, some issues, such as false positives, appeared. Once a presenter is performing the static pose the system is always able to determine the correct gesture. However, in some cases, such as in gesture D (right hand raised over the head), getting to the final pose involves passing through some other poses such as A, B, or C, thus triggering the corresponding actions and being false positives. This can be avoided by training these movements so that the presenters raise their hands in front of their bodies instead of laterally. Therefore, this gesture recognition system is usable but not ideal as it only focuses on instant angles of the body parts of the presenters and not on their movements along the time. Probably, these false positives in the transitions from one pose to another could be avoided with a more complex system but it may include new drawbacks such as in the solution presented in [31].

These tests prove that HPE algorithms are a viable solution for gesture or pose detection in VTS, freeing the presenter from rehearsed choreographies and remote controls

and making possible a more natural interaction between real and virtual elements.

8 Discussion

In this research we have explored 14 HPE solutions as candidates for testing the viability of their implementation in VTS environments. In a first step, 9 of these solutions where discarded due to different reasons such as lack of stability or incompatibility with the test system (not running on windows 10). The remaining five alternatives have been tested with local and global (on-set) tests in both their no view (without any visual reference of the results) and printmetrics (offering an image of how the HPE is working) versions. From these tests the conclusion was that three of the selected alternatives were able to work consistently in real time: MediaPipeTest, Drone-Vision and BlazePose-OpenVino. BlazePose-OpenVino was selected as it was the alternative which performed better in terms of use of resources such as RAM and processor load. Once this approach was selected, a test of real-world usage was performed in terms of triggering events in a VTS.

After all the tests performed, we can confirm that HPE solutions can be a powerful tool to introduce in a VTS. They present several advantages that make them a very interesting alternative to today's expensive tracking systems. First, HPE solutions take advantage of hardware that is already present in every studio. The use of the recording cameras as sensors decreases the cost of these tools as they do not require the purchase of specific hardware. Moreover, the configuration of a VTS with a chroma cyclorama eases the work for this image-based approaches as the human figures tend to have a very different color from the background, making the tracking process more accurate than in more complex conditions with lower light or more confusing scenarios. The second fact that makes HPE a viable solution for VTS is the possibility to run the programs in real time. We have proved that there are free available solutions which are able to maintain consistent and robust real time frame rates during their execution, averaging around 25 frames per second. This is compulsory in the scenario of VTS as all their technology is focused on live broadcasting or recording. However, if the HPE solution is only focused on triggering effects, lower frame rates would be also viable as long as the viewer is not able to perceive the delay between the action and the triggered effect. Thirdly, HPE solutions can be easily integrated in a VTS by using tools such as the middleware presented in [13] by just implementing a VRPN server that sends the pose of each body part and a click action when a pose or gesture is detected. Having this implementation, all the data generated by the HPE solution can be used in the virtual environment to implement a more natural interaction. Finally, the precision of this system is sufficient to offer a reliable pose detector which can simulate gesture interaction between the presenters and the virtual environment. In the case of the selected software (BlazePose-OpenVino) this pose detector is already implemented for the semaphore alphabet and has been successfully tested. However, more complex solutions could offer a bigger variety of poses and gestures as well as reduce the problems of false positives when going from one gesture to another.

Fig. 6 Real time BlazePose-OpenVino result in short shot framing. In these cases, pose detection is not viable

In terms of limitations in the usage of these techniques in VTS, we have found three main issues. The first one is intrinsic to the chosen solution. The use of the recording cameras is very advantageous in economic terms, as no new hardware is needed for the system to work. However, in a real production scenario, these cameras will perform movements and zoom in and out, directly affecting the pose estimation and even the viability of the system (Fig. 6). Thus, this kind of system should be processing the stream of a camera which is more focused on general shots than those which have a more variable use. A different solution would be to have a specific camera for the HPE system. This solution will increase the cost of the final implementation but will ensure better framing during the whole production. Moreover, the quality, and therefore the price, of the camera needed to feed the HPE system does not have to be the same as the broadcast equipment as their images will not be transmitted or recorded and would be feasible to implement it with a webcam. The second issue that has been found was the inability of the system to provide reliable depth information. VTS environments are 3D computer-generated models and, to guarantee a real interaction between the actors and the CGI elements, a three axes information is needed. This parameter is necessary, for example, to compute if the presenter is behind or in front of an object, or to locate the presenter 3D object when using virtual camera movements. Moreover, when a direct interaction of the presenters with the virtual objects, such as attaching an object to

the presenter's hand, is needed these types of solutions may not be ideal. This lack of depth information may be overcome using screen coordinates. As the image analyzed by the HPE software and the broadcasted image are the same, the objects could be located in screen space, simulating the visual attachment to the presenter's hand. However, this solution could present problems in terms of size and perspective of the virtual objects as they are just being located in a 2D space. Finally, the tested solution included a static pose detector which made it possible to trigger visual effects when the talents performed one of the selected poses. This solution proved to be very robust, as it always detected the desired poses. However, it presented problems with the dynamic movements of the presenters, as to reach some of the poses some different events were triggered in the way. This is not an intrinsic problem of HPE as different dynamic gesture detectors could be trained and applied to the system but represents a problem when a simpler and more robust alternative, like static pose detection, is used. This problem can be solved by training the presenters in the transitions between poses not to perform determined movements or by selecting poses which are not related in their composition.

9 Conclusions and future work

In this work, we have determined a viable single-person HPE method, which suits the requirements for both accuracy and efficiency, for use in a virtual television set. First, a set of requirements for the seamless inclusion of the HPE module in a VTS have been determined. These requirements are real-time performance, minimal latency, isolation of the HPE module to avoid interferences in the workflow, minimal inference in the final video composition, and the ability to not affect the integrity of the live broadcast beyond the disappearance of the presenter's anatomy tracking capability. These points are discussed, and quantitative thresholds, hardware and software architecture are proposed as empirical data for the test.

Performance criteria for inclusion of an HPE in a VTS have also been defined. The fundamental element is to maintain a framerate over 10 FPS, although it is recommended to be as close to 30 FPS as possible. In our tests, a framerate above 10 FPS has not shown any artefacts or lag in the output. On the other hand, it is possible to empirically establish best practice criteria for memory load and CPU occupancy. In our experience, a memory load of less than 500 MB and a CPU load of less than 35% are sufficient to guarantee correct operation in these types of systems. Again, we would like to emphasize that these data are empirically obtained from the data gathered with our VTS equipment.

A total of 14 possible HPE solutions have been analysed for incorporation into a professional VTS. Of them, 9 were not taken into account due to performance issues or the need to receive data over the network, which would slow down performance in a real-time application. Of the remaining 5 solutions, only 3 meet the frame rate requirements; these are MediaPipeTest, Drone-Vision, and BlazePose-OpenVino. Our experimental results have shown that the proposed approach BlazePose-OpenVino is the best performing implementation due to its high value in the FPS_AVG metric, its moderate value in the CPU_AVG metric and its low value in the MEM_AVG metric making it the best alternative for use. On the other hand, by default, it includes a basic pose recognition toolkit which has been proven to be solid enough for a possible incorporation into the VTS.

Moreover, some additional advantages and drawbacks have been analyzed during the test of the selected solution in a real production scenario. As advantages, we can point out the low-cost of the proposed solution, as it takes advantage of existing studio cameras, together with the ideal scenario that a chroma-keying cyclorama represents for HPE algorithms. In addition, the real-time performance of the system, its seamless integration with existing broadcast tools such as InfinitySet, and the ability to detect pre-defined poses to trigger events increase its feasibility. Beyond these technical advantages, the approach also contributes to universal accessibility by eliminating the need for specific tracking hardware, reducing economic barriers and enabling the design of natural, user-defined interaction patterns, adaptable to the diverse capabilities and contexts of each user. Nevertheless, a common production workflow may vary the framing of the cameras, affecting the performance of the HPE system, 3D data is not available or not consistent making it difficult to include interactions that involve working in three axes and pose detection for even triggering, although very accurate, presents problems if the presenters go through a defined pose to reach another one (both of them would be triggered).

As future work we aim to further study the tests trying to solve the issues that have been detected. Complex interactions will be tested to prove if working in screen space instead of 3D space is a viable solution to solve human-computer interactions. On the other hand, a dynamic gesture detector will be implemented and tested to study its reliability and check if it would be a more robust solution that the one tested in this paper, as this kind of solutions tend to have more false positives and negatives than the simple approach of BlazePose-OpenVino. However, the solution as is has proven to be an interesting alternative for low budget projects implementing an accesible free of cost alternative.

Author contributions R. A. carried out the main tests, wrote the main part of the manuscript, created the Tables and Fig. 3 and contributed

to the discussion and conclusions. R. M.: carried out the interactivity test, created the Figs. 1, 2, 4, 5 and 6, wrote part of the manuscript and contributed to the discussion and conclusions. L. P.: designed the methodology and contributed to the discussion and conclusions. E. C.: designed the methodology and contributed to the discussion and conclusions. J. F.: designed the methodology and contributed to the discussion and conclusions. All authors reviewed the manuscript

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. No funding was received for conducting this study.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Goldsmlith, A.N.: Color micro-facsimile system. US Patent No **2**, 073–370 (1937)
2. Schuemie, M.J., Van der Straaten, P., Krijn, M., Van der Mast, C.A.P.G.: Research on presence in virtual reality: a survey. Cyberpsychol. Behav. **4**(2), 183–201 (2004). https://doi.org/10.1089/109493101300117884
3. Shimoda, S., Hayashi, M., Kanatsugu, Y., Kanatsugu, Y.: New chroma-key imaging technique with hi-vision background. IEEE Trans. Broadcast. **35**(4), 357–361 (1989). https://doi.org/10.1109/11.40835
4. Wojdala, A.: Challenges of virtual set technology. IEEE Multimedia **5**(1), 50–57 (1998). https://doi.org/10.1109/93.664742
5. Petrovic, M., Jaksic, B., Spalevic, P., Petrovic, I., Dakovic, V.: The analysis background on the effect of chroma-key in virtual tv studio. INFOTECH **12**, 937–941 (2012)
6. Gibbs, S., Arapis, C., Breiteneder, C., Lalioti, V., Mostafawy, S., Speier, J.: Virtual studios: an overview. IEEE Multimedia **5**(1), 18–35 (1998). https://doi.org/10.1109/93.664740
7. Sanzharov, V.V., Frolov, V.A., Galaktionov, V.A.: Survey of Nvidia RTX technology. Program. Comput. Softw. **46**(4), 297–304 (2020). https://doi.org/10.1134/S0361768820030068
8. Díaz-Alemán, M.D., Amador-García, E.M., Diaz-González, E., Torre-Cantero, J.: Nanite as a disruptive technology for the interactive visualisation of cultural heritage 3d models: a case study. Heritage **6**(8), 5607–5618 (2023). https://doi.org/10.3390/heritage6080295
9. Tan, T.W.: Mastering lumen global illumination in unreal engine **5**, 223–275 (2024)
10. Hach, T., Arias, P., Bosch, C., Montesa, J., Gasco, P.: Seamless 3D interaction of virtual and real objects in professional virtual studios. SMPTE Mot. Imaging J. **126**(1), 43–56 (2017). https://doi.org/10.5594/JMI.2016.2632398
11. Cho, H., Jung, S.U., Jee, H.K.: Real-time interactive AR system for broadcasting. In: Proceedings–IEEE Virtual Reality, pp. 353–354 (2017). https://doi.org/10.1109/VR.2017.7892322
12. De Goussencourt, T., Bertolino, P.: Using the unity® game engine as a platform for advanced real time cinema image processing. In: Proceedings–International Conference on Image Processing, ICIP 2015-December, pp.. 4146–4149 (2015). https://doi.org/10.1109/ICIP.2015.7351586
13. Méndez, R., Flores, J., Castelló, E., Viqueira, J.R.R.: New distributed virtual TV set architecture for a synergistic operation of sensors and improved interaction between real and virtual worlds. Multimedia Tools Appl. **77**(15), 18999–19025 (2018). https://doi.org/10.1007/S11042-017-5353-Y/FIGURES/11
14. Aguilar, I.A., Sementille, A.C., Sanches, S.R.R.: ARStudio: a low-cost virtual studio based on augmented reality for video production. Multimed. Tools Appl. **78**(23), 33899–33920 (2019). https://doi.org/10.1007/S11042-019-08064-4/METRICS
15. Kruszewski, P., Mahamad, T.J.: The AI powered magic mirror: building immersive AR/VR experiences with only webcams and deep learning. In: ACM SIGGRAPH 2018 Virtual, Augmented, and Mixed Reality. Association for Computing Machinery (ACM) (2018). https://doi.org/10.1145/3226552.3226569
16. Dang, Q., Yin, J., Wang, B., Zheng, W.: Deep learning based 2D human pose estimation: a survey. Tsinghua Sci. Technol. **24**(6), 663–676 (2019). https://doi.org/10.26599/TST.2018.9010100
17. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2021). https://doi.org/10.1109/TPAMI.2019.2929257
18. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: VNect: real-time 3D human pose estimation with a single RGB camera. ACM Trans. Gr. (2017). https://doi.org/10.1145/3072959.3073596/SUPPL_FILE/PAPERS-0079.MP4
19. Nakano, N., Sakura, T., Ueda, K., Omura, L., Kimura, A., Iino, Y., Fukashiro, S., Yoshioka, S.: Evaluation of 3D markerless motion capture accuracy using openpose with multiple video cameras. Front. Sports Act. Living **2**, 538330 (2020). https://doi.org/10.3389/FSPOR.2020.00050
20. Noori, F.M., Wallace, B., Uddin, M.Z., Torresen, J.: A robust human activity recognition approach using openpose, motion features, and deep recurrent neural network. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11482**(LNCS), 299–310 (2019). https://doi.org/10.1007/978-3-030-20205-7_25/FIGURES/9
21. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: efficient online pose tracking. In: British Machine Vision Conference 2018, BMVC 2018 (2018). arXiv:1802.00977
22. Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11218 LNCS, 282–299 (2018). https://doi.org/10.1007/978-3-030-01264-9_17/FIGURES/5arXiv:1803.08225
23. Martinez, G.H.: OpenPose: whole-body pose estimation (2019)
24. Zhu, T., Karlsson, P., Bregler, C.: SimPose: effectively learning densepose and surface normals of people from simulated data. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in

Bioinformatics) 12374 LNCS, 225–242 (2020). https://doi.org/10.1007/978-3-030-58526-6_14/FIGURES/8arXiv:2007.15506
25. Chen, Y., Shen, C., Wei, X.S., Liu, L., Yang, J.: Adversarial PoseNet: a structure-aware convolutional network for human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision 2017-October, pp. 1221–1230 (2017). https://doi.org/10.1109/ICCV.2017.137arXiv:1705.00389
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 07-12-June-2015, pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594arXiv:1409.4842
27. Kim, J.-W., Choi, J.-Y., Ha, E.-J., Choi, J.-H.: Human pose estimation using mediapipe pose and optimization method based on a humanoid model. Appl. Sci. (2023). https://doi.org/10.3390/app13042700
28. Groos, D., Ramampiaro, H., Ihlen, E.A.: EfficientPose: scalable single-person pose estimation. Appl. Intell. **51**(4), 2518–2533 (2021). https://doi.org/10.1007/S10489-020-01918-7/FIGURES/6
29. Castelló Mayo, E., López Gómez, A., Méndez Fernández, R.: La transferencia de conocimiento desde la universidad innovadora. un modelo de gestión de la información en: el contexto digital el caso de estudio piedd. Rev. Lat. Comun. Soc. **74**, 537–553 (2019). https://doi.org/10.4185/RLCS-2019-1344
30. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3D human pose estimation in the wild using improved CNN supervision. In: Proceedings–2017 International Conference on 3D Vision, 3DV 2017, pp. 506–516 (2018). https://doi.org/10.1109/3DV.2017.00064arXiv:1611.09813
31. Méndez, R., Flores, J., Castelló, E., Viqueira, J.R.R.: Natural interaction in virtual TV sets through the synergistic operation of low-cost sensors. Univ. Access Inf. Soc. **18**(1), 17–29 (2019). https://doi.org/10.1007/S10209-017-0586-0/FIGURES/12
32. Meehan, M., Razzaque, S., Insko, B., Whitton, M., Brooks, F.P.: Review of four studies on the use of physiological reaction as a measure of presence in stressful virtual environments. Appl. Psychophysiol. Biofeedback **30**(3), 239–258 (2005). https://doi.org/10.1007/S10484-005-6381-3/METRICS
33. Meehan, M., Razzaque, S., Whitton, M.C., Brooks, F.P.: Effect of latency on presence in stressful virtual environments. In: Proceedings–IEEE Virtual Reality 2003-January, pp 141–148 (2003). https://doi.org/10.1109/VR.2003.1191132