Journal of Software Engineering and Applications, 2019, 12, 127-148
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Hyper 360—Towards a Unified Tool Set Supporting Next Generation VR Film and TV Productions

Barnabas Takacs[1], Zsuzsanna Vincze[1], Hannes Fassold[2], Antonis Karakottas[3], Nikolaos Zioulis[3], Dimitrios Zarpalas[3], Petros Daras[3]

[1]Drukka Kft/PanoCAST, Budapest, Hungary
[2]Joanneum Research Digital—Institute for Information and Communication Technologies, Graz, Austria
[3]Centre for Research & Technology Hellas (CERTH)—Information Technologies Institute (ITI), Thessaloniki, Greece
Email: btakacs@panocast.com, zsuk@digitalcustom.com, hannes.fassold@joanneum.at, ankarako@iti.gr, nzioulis@iti.gr, zarpalas@iti.gr, daras@iti.gr

How to cite this paper: Takacs, B., Vincze, Z., Fassold, H., Karakottas, A., Zioulis, N., Zarpalas, D. and Daras, P. (2019) Hyper 360—Towards a Unified Tool Set Supporting Next Generation VR Film and TV Productions. *Journal of Software Engineering and Applications*, **12**, 127-148.
https://doi.org/10.4236/jsea.2019.125009

Received: April 3, 2019
Accepted: May 24, 2019
Published: May 27, 2019

Abstract

We describe four fundamental challenges that complex real-life Virtual Reality (VR) productions are facing today (such as multi-camera management, quality control, automatic annotation with cinematography and 360˚ depth estimation) and describe an integrated solution, called Hyper 360, to address them. We demonstrate our solution and its evaluation in the context of practical productions and present related results.

Keywords

Deep Learning, Tensor Flow, YoloV3, 360˚ Video, Virtual Reality, Free Viewpoint Video, Quality Control, Automatic Cinematography, Multi-Camera Systems

1. Introduction

Virtual Reality (360˚) video content recently got very popular in the media industry as it allows viewers to experience the content in an immersive and interactive way. Professional 360˚ video cameras capture the entire viewing sphere and create stitched video output typically in 4 K, 8 K monoscopic and 6 K stereoscopic *equirectangular* format. A critical factor for the long term success of 360˚ video content is the availability of convenient tools for producing and editing 360˚ video content for a multitude of platforms (e.g. mobile device, VR Headsets or conventional TV sets via HbbTV). Existing tool sets primarily focusing on a) high quality *video stitching*, *stabilization* and *optimal encoding* with minimal loss of raw camera data and b) interactive annotation tools to place *hot spots, embedded images and videos* combined with visual links for interaction and storytelling purposes.

To date, most research consists of quantitative experiments that concentrate on audience perception of content [1], viewing habits [2] or technologically focused studies, e.g., on watching a video on different devices [3], while the power of actual storytelling for this new, innovative format is still being invented [4].

In this paper we go beyond these elements and focus on four fundamental problems that real-life VR productions are facing today in particular as they relate to immersive story telling as a necessary element for *narrative and technical immersion* [5]. Our set of problems ranges from principles of 1) optimal camera placement & management [6] to 2) on-set quality control extended from methods used in video quality assessment [7], 3) automatic annotation & cinematography to convert panoramic videos into a normal field-of-view for optimal viewing experience [8], and 4) 360˚ depth estimation to be able to estimate structure and semantics in 360˚ scenes [9] and insert additional 3D elements as a post-production step.

To address these challenges, we describe an integrated solution (Hyper 360) and available tools that we have developed and demonstrate their practical use with examples from Music, Fashion and Film industry applications. More explicitly, for camera placement and management we developed a multi-camera control architecture, called OMNICAP [10], capable of optimizing the placement and remote management of up to 64 mixed camera units. For on-set quality control we introduce a novel 360˚ video quality assessment pipeline (Vidicert), capable of on-the-fly assessment of camera setups and imaging conditions. Similarly, for automatic annotation and cinematography, we have developed a deep learning-based object detector and a dense optical flow module operating at nearly 99% accuracy, and finally for 360˚ depth estimation we introduced a novel deep CNN solution to infer depth directly on the equirectangular domain. In the following sections we describe in detail each of these advances and their significant contributions to the production workflow.

The paper is organized as follows. In Section 2 we briefly overview and assess related work and currently available tools in the market. Section 3 outlines several key challenges facing VR productions today with our solutions to these challenges, followed by a presentation of our test results in Section 4. Finally, in Section 5, our Conclusion and Future work are presented.

2. Related Work

There is a vast number of currently available tools for VR content stitching, post-production and annotation. As a first step they primarily focus on high quality ***video stitching*** from an array of spherically arranged miniature cameras with wide angle lenses and overlapping imagery. As the nodal point of these camera lenses is the offset from their ideal central position, the stitching process itself suffers from the inherent problems associated with *parallax.* Specifically, since the position or direction of objects appears to differ when viewed from two neighboring camera positions, there are artifacts appearing in the stitching line. Advanced professional stitching tools, such as *Adobe CC Premier* [11], *Insta 360 Pro* [12] and *SGO Mistika* [13], apply complex optic flow distortions to make these image regions error free and seamless to the Viewer. The second group of solutions aims at ***stabilizing*** the spherical image by tracking a dense set of natural image features and deriving 3D camera tracks and rotating the spherical view frame-by-frame to correct unwanted motion and generate the desired output in the form of a stable *equirectangular* image (Syntheyes [14], Facebook [15]).

Following the initial creation of 360˚ Virtual Reality camera footage, the next step is to turn it into immersive and interactive video via ***authoring tools*** that provide the standard functionalities like adding hot spots or overlays which refer to other resources like text, 2D images or another video. So far, this annotation process *i.e.* adding these interactive elements is a completely manual process, thus the user has to choose the object in the frame where the element is to be added (e.g. attached to a person) and has to change the hot spot location manually in each video frame if the object moves throughout the 360˚ space. As an example, in [16], a web-based annotation tool for 360˚ video is presented, based on the *WebVR*, *Three. Js* and *Node. Js* frameworks. It provides a variety of annotation types (marker, subtitles, arrow, vignette, etc.), each serving a different purpose. User tests were performed in order to measure the effect of the annotations on the immersive viewing experience. A comparative review of commercial solutions for similar purpose (such as *WondaVR*, *Liquid Cinema, Fader, ThingLink, Viar* 360*, Sprawly, Viond, Omnivirt*, and 3*DVista Pro* [17]-[25]) quickly reveals that so far none of these tools supports the automatic extraction of the objects (like persons, animals or cars) occurring in the scene, which is one of the key elements of the tool set our work is focusing on. Our integrated capturing and authoring system, called **OMNICAP & OMNICONNECT** respectively [10], goes beyond the capabilities of these individual authoring solutions by delivering an integrated multi-camera control pipeline with fully automated object detection and tracking module and an added on-set quality control to process the recordings.

Automatic Cinematography from a 360˚ video is another field of active research interest. The goal here is to generate a conventional 2D video from the 360˚ video automatically, by determining a smooth camera path which captures the salient (most interesting) regions of the viewing sphere. These regions often correspond to persons performing specific actions, e.g. for a concert scene the salient regions will correspond to the members of the music band performing the act. The approach proposed in [26] first learns a discriminative model from conventional 2D videos collected from the web. It then uses this model to identify candidate viewpoints and events of interest to capture in the 360˚ video and determines an optimal camera trajectory using dynamic programming. A disadvantage of this approach is that the learned model is specific to a certain category of 360˚ video (e.g. soccer videos or hiking videos), as the conventional 2D videos have been collected on the web via domain-specific query keywords. The work [27] extends the algorithm proposed in [26] so that zoom shots (with a small field of view) and wide-angle shots are also supported. Furthermore, the run time of the algorithm is reduced via a coarse-to-fine strategy for camera trajectory search. In [28], a method is presented for piloting through 360˚ sports videos. Their method learns an online policy to focus on foreground objects (like a skateboarder) and simultaneously minimizes both view angle loss from human annotated ground truth and smoothness loss between consecutive frames. A deep learning based object detector (Faster R-CNN) is employed for generating candidate hypotheses for objects of interest. As the online policy is learned from sport video only, this method is not able to handle other (non-sports) categories of 360˚ video. In [29], design guidelines for extracting conventional 2D video shots from a 360˚ video are given (e.g. subject should be centered, people should not be cropped), derived from user studies. Based on these guidelines, a method is presented for extracting a conventional 2D video shot, relying on face and pose detection. Our method advances the state-of-the-art by extending the camera path generation algorithms to use intelligent automated object detection in combination with low-level image features and saliency and combine those with high-level artistic and compositional rules thus arriving at a stable shot framing structure and sequence.

While ***Stereoscopic 3D*** makes the viewer more immersed in a 360˚ video, by simulating how humans perceive the world through their binocular vision and resulting in the illusion of depth, it only allows for 3 degrees of freedom (3DOF), *i.e.* only rotations (pitch, yaw, roll). To make 360˚ videos more immersive, there is an increasing interest in research towards 3DOF + 360˚ videos, which apart from the viewpoint rotation, also allows for limited translation. To enable viewpoint translation the system must be able to render new views from the new translated viewpoints, thus the 3D geometry of the scene is essential. One of the most important type of information regarding scene geometry is the scene's depth, which gives the ability to generate (*i.e.* render) novel views from arbitrary viewpoints.

Despite the recent popularity of 360˚ media, limited work addresses the problem of monocular 360˚ depth estimation. However, the existing approaches vary from standard methods used in 2D traditional imagery to novel ones tailor-made for 360˚. In [30] the authors follow a Structure from Motion (SfM) approach to create an initial sparse reconstruction of the scene from the new viewpoint by two separate non sequential key frames, in order to later refine it by taking into account the frames in-between. Their method requires to first project each frame from equirectangular to cubemap projection, which in general is computationally costly.

A machine learning approach is presented in [31] where a self-supervised deep Convolutional Neural Network (CNN) is utilized, in order to learn depth and camera pose from 360˚ videos. This method also projects each frame to cubemap projection using cube padding to pad intermediate features from adjacent faces, and uses view synthesis between consequent frames in order to estimate the camera position and the depth of the depicted scene. Finally, in [32], a new 3D video representation is proposed, the Depth Augmented Stereo Panorama (DASP), in which every 360˚ stereoscopic video frame is accompanied by its corresponding depth, thus allowing for novel viewpoint rendering and providing motion parallax in VR videos.

Our solution goes beyond existing techniques by overcoming the costly processing of cubemap projection, and introducing a learning based method for 360˚ depth estimation [33]. This method was trained on and operates directly on the equirectangular domain. To facilitate this training process we also introduced a novel data set of 360˚ indoors scenes with their corresponding ground truth depth annotations.

Having reviewed key research related to our work, we now turn our attention to the practical challenges professionals face, with a special emphasis on 360˚ Film and Television productions as well as the particular requirements our solution addresses.

3. Four Key Challenges of 360˚ VR Film & TV Productions

3.1. Camera Placement, Motion and Multi-Camera Management

360˚ cameras are designed to deliver VR experiences aimed at placing the Viewers right at the center of the action. However, due to the nature of the imaging process that employs multiple tiny camera units with ***fisheye*** lenses, the perceived image in the VR headset appears to be further as in real life. Therefore, for a first person experience, the camera must be as close as possible to its principal subject. On the other hand, the closer the camera is to an object, the stronger the ***parallax*** effect becomes, thus causing undesirable stitching errors. The optimal point for these two constraints is placing the camera within 1.5 - 2 meters away from the principal subject, but that may cause real-life production challenges as demonstrated in **Figure 1**, where a total of 19 cameras work together creating multiple *line of sight problems* for musicians and camera crews alike. Moving 360˚ cameras is often not a preferred option as they need to be synchronized with special effects and other camera motion platforms (cable cams, Jimmy Jib, robotics), but more importantly the resulting imagery may cause dizziness in the Viewers. As a practical compromise a production may use multiple stationary 360˚ units—typically 3 to 9 on stage to cover all aspects of interest—that would have to be remotely managed.

To address this need, we have developed a complex camera control and acquisition tool, called ***OMNICAP*** [10] that integrates a number of modern 360˚ camera rigs (either multi camera arrays or tiny fisheye lenses devices) with regular HD cameras into a generic architecture that can support the remote management of *up to* 64 *mixed camera unit*s and configurations as demonstrated in **Figure 2**. It was designed with both multi-camera 360˚ as well as 3D captures to support *Free Viewpoint Video* suited for the broadcast sector. More specifically,

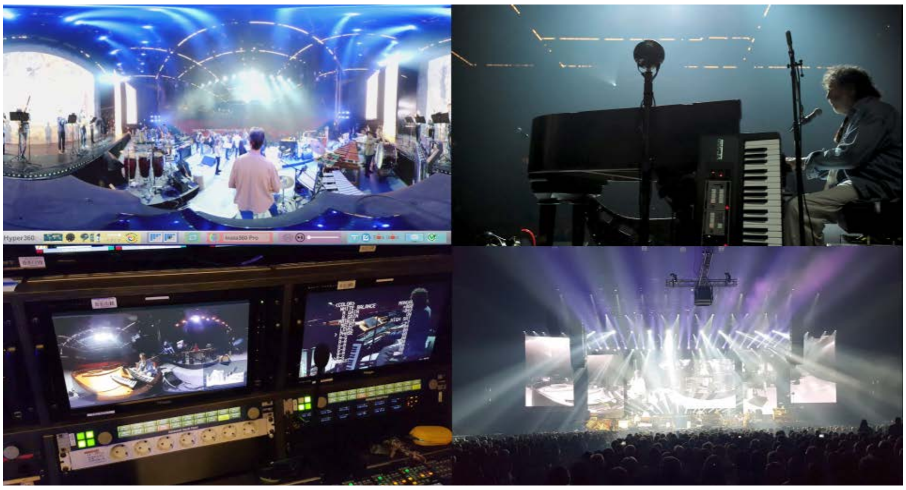

Figure 1. Example of a Live Music Production using a mixed set of 19 cameras including three 360˚ units on stage (see also https://youtu.be/wk-pbMF6RY0).

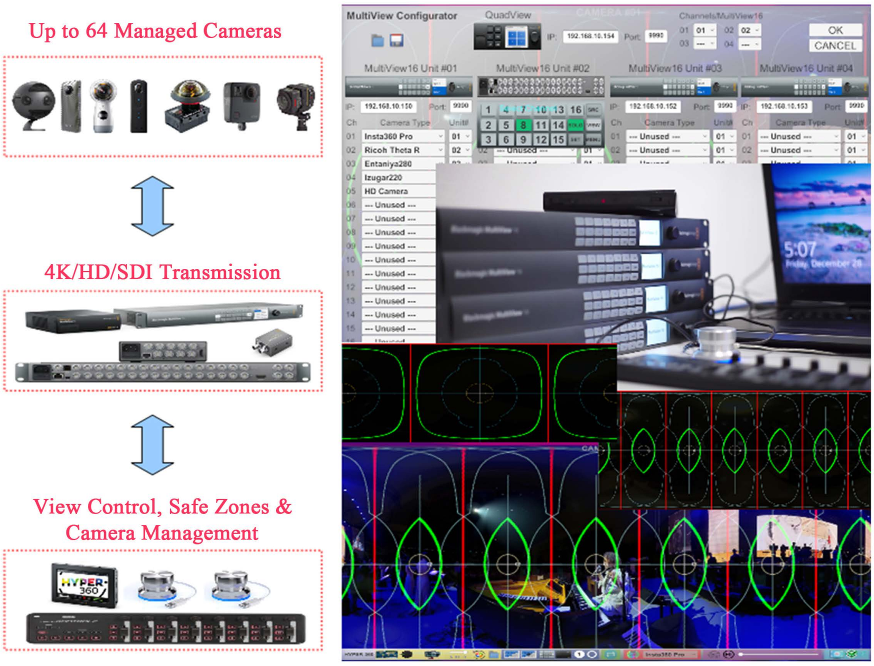

Figure 2. OMNICAP Multi-camera management solution for mixed-type 360 and regular cameras (see also https://youtu.be/ixlYZGN5Jl8).

our solution is based on a generalized inverse fisheye lens transformation implemented in real-time, whereas each individual camera view is first distorted in real-time using *pixel*- and vertex *shader* technology and subsequently combined into a single *equirectangular* view that can be viewed in *Flat mode* (seeing the entire scene in a distorted manner) or mapped onto a *virtual sphere* to be able to frame and look at elements within the scene. Each camera's input is mapped onto a graphic element called "*patch*" and multiple patches are blended and distorted to create seamless transitions in the overlapping areas. Since the purpose of our tool is on-set pre-visualization, control and quality check to find the best camera setup, it also includes additional overlay elements such as the visualization of safe-zones. The same software can also be used to manage *Photogrammetric 3D Capture* and *Free Viewpoint Video* recordings as demonstrated in Figure 3. (For real-time capture and integration of 3D human characters or "*Mentors*" we have developed a dedicated solution described in Section 3.4 below.)

3.2. On-Set Quality Control

The more objects and people are in a scene, the more stitching and other types of imaging errors become a problem. Furthermore, as most 360˚ cameras have small CCD image sensors and a *limited* 8 *bit dynamic range*, *highlights* and *low-light performance* all become important additional issues that yield reduced quality. Encoding and compression artifacts just further add to this mix and each subsequent post-production step (stitching, stabilization, editing) involves multiple rendering passes that lower quality even further.

Therefore, it is vital for any 360˚ TV or Film production to capture information with minimal loss of raw camera data and to ensure, while setting up a production or on-set, that the resulting imagery will meet the accepted high quality broadcast standards. This can be achieved by on-set tools that provide quick assessment and immediate feedback to camera crews in regards to what they need to adapt, change or do differently, by adapting standard video quality assurance processes meeting the special requirements of equirectangular video.

The ***Quality Check*** (QC) component is able to detect a variety of defects commonly occurring in 360˚ video like blurriness, signal clipping, noise, flicker, macroblocking and several others. More specifically, *signal clipping* occurs when the automatic gain control of the 360˚ camera sets the camera gain too high, resulting in blown-out highlight areas. *Flicker* can occur when the camera is out

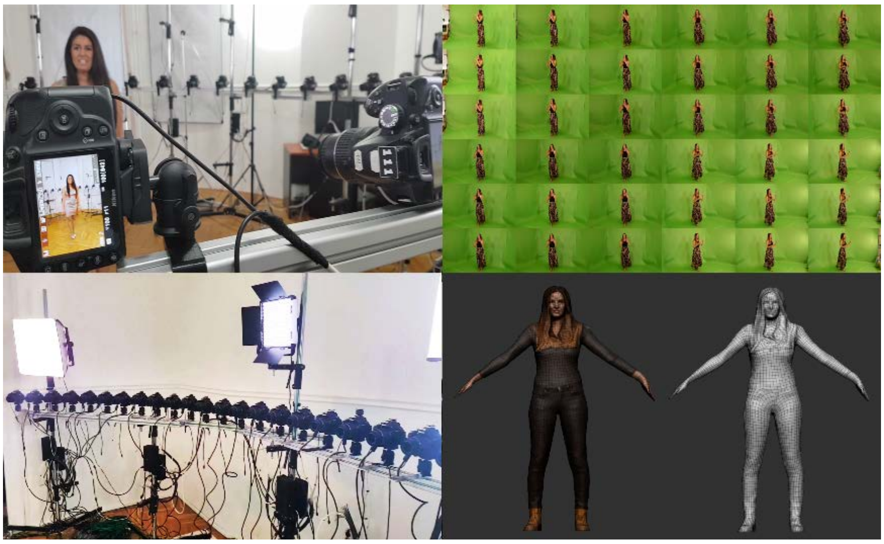

Figure 3. OMNICAP Multi-camera setup used for 3D capture and Free Viewpoint Video recordings (see also https://youtu.be/8qu4RDoYWyI).

of sync with artificial lights in the scene (fluorescent lights, etc.). The quality check component is able to detect these defects (and others) and report them to the user, so that he can take corrective actions on-site, e.g. by setting the camera parameters differently or modifying the setup of the scene. This QC module reports of the detected defects to the user in a separate application named "*VidiCert Summary*" (see Figure 4), which opens automatically after the processing of the video has finished. It provides a comfortable tool for quickly inspecting the detected defects in the video, where each detected defect is shown in a separate timeline either as bar or as graph.

3.3. Automatic Annotation and Cinematography

In order to create ***dense annotations*** with multiple tracked and tagged elements in the scene for extended virtual reality movie sequences, the currently available manual authoring tools (see Section 2) offer only a rather limited set of solutions. This severe drawback comes from to the inherently manual nature of the work process they employ. Hot spot editing applies simple animation principles by inserting a *sequence of key frames* at given times, which the playback system can automatically interpolate for in-between moments. *Visible hot spots* appear in the scene and act as sources of information or as links to videos or other

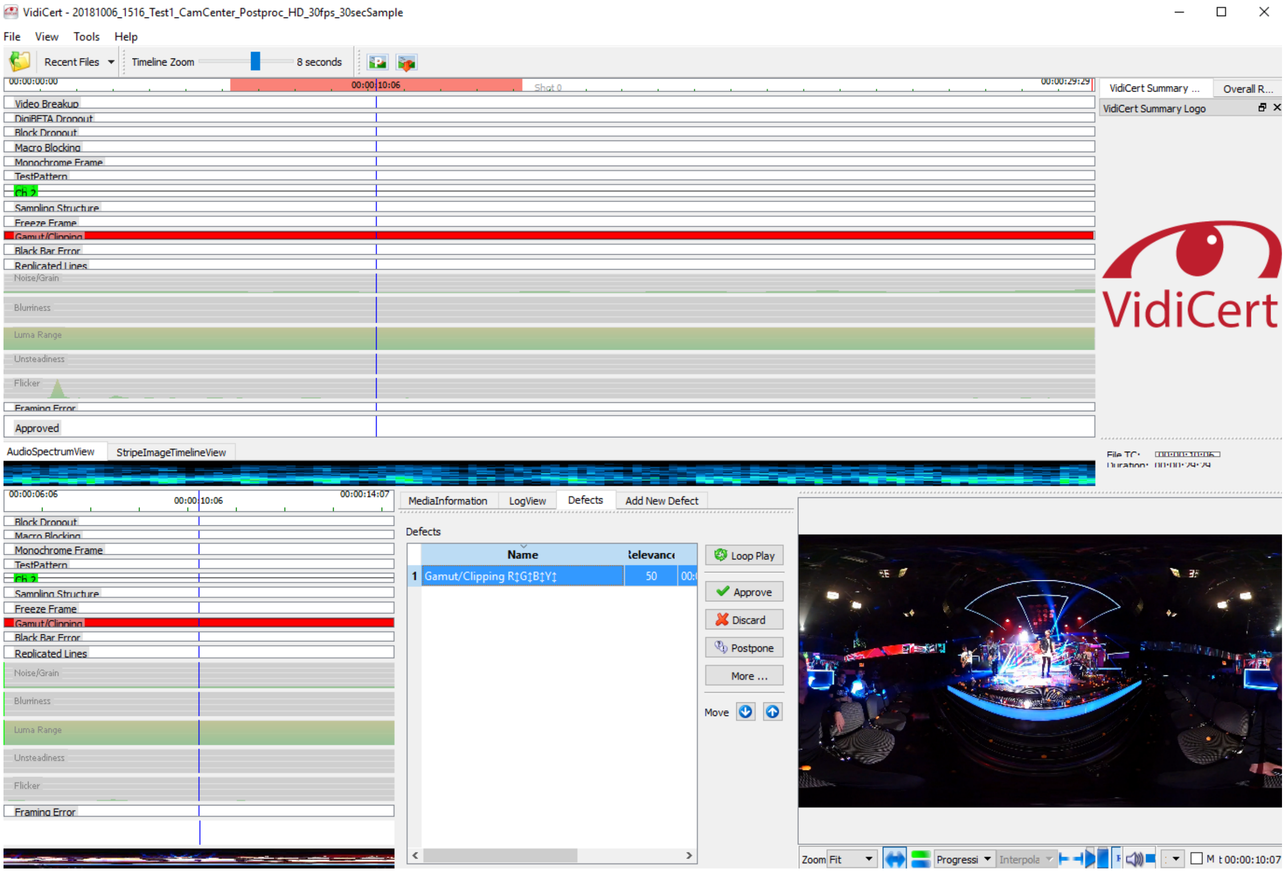

Figure 4. VIDICERT Summary application showing the detected defects in a multi-camera 360˚ video from the 2017 Eurovision Contest (see also https://youtu.be/w_Slr7Hy-WM).

scenes. *Invisible hot spots*, on the other hand, allow the playback system to detect what annotated elements Viewers are looking at in any moment. Therefore this mechanism offers not-only advanced means to gain insight and statistical analysis more refined than simple VR heat maps [34], but also it creates the foundation for ***on-the-fly fuzzy semantic inference*** [35] that can drive recommendation systems.

To support meaningful annotation of hundreds of tagged elements the tasks of ***scene object extraction*** deals with the automatic detection and tracking of all important objects (e.g. persons, animals, cars, furniture, etc.) that occur in the scene. Like many approaches nowadays, it relies heavily on recent advances in neural networks and deep learning. The key components being a deep learning-based object detector and a dense optical flow algorithm for tracking the detected objects from one frame to the next one. For object detection, we employ the *YoloV3* algorithm [36], which is able to detect 80 classes of objects commonly occurring in videos, like persons, various animals (dogs, cats) and vehicles (motorbike, car, truck). Subsequently a high-quality variational *optical flow* algorithm (TV-L1 [37]) is used for calculating the dense motion field between two consecutive frames with both components running entirely on the GPU to achieve high speeds.

For each frame, the workflow of the algorithm is as follows: Firstly, the motion field between the current and next frame is calculated via the optical flow algorithm. Additionally, the object detector is invoked for the next frame, which yields a list of detected objects. For each scene object, the motion field is employed to predict the position of the object in the next frame. A global matching between the detected objects and the predicted scene objects is done. All scene objects which could not matched are considered as lost (e.g. because they got occluded) and excluded from further processing. In contrast, all detected objects, which could not be matched are added to the scene object list.

A typical result of the algorithm can be seen in **Figure 5**. One can see that the algorithm is able to detect the important objects in the scene (like the person or

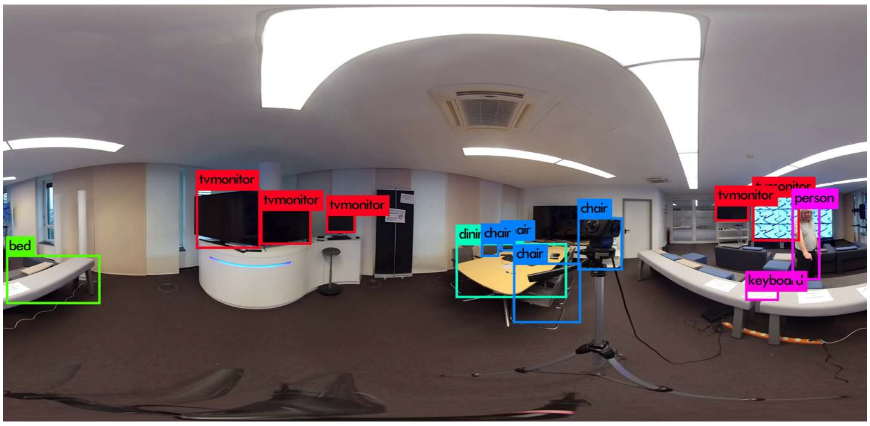

Figure 5. Result of the scene object extraction algorithm for one frame of a 360˚ video showing an office scene (above), and from an ongoing production of Kafka's Metamorphosis (below).

the monitors), although a few false detections are also reported (like the bed in the office scene). However, these false detections can be easily filtered out with the help of domain-specific knowledge. In terms of performance, the equirectangular projection of the image does not affect the algorithm negatively with the average run time being roughly 105 milliseconds for one Full HD resolution frame on a PC with a Geforce GTX 1070 GPU.

This semantic information gathered from scene object extraction is also crucial for the automatic cinematography algorithm. Its goal is to calculate a visually interesting camera path from a 360˚ video, in order to provide a traditional TV-like consumption experience. In contrast to interactive viewing, hard cuts between individual shots of the result video are allowed and even desired as they are a stylistic device commonly occurring in conventional 2D video content. In order to generate a pleasing and visually interesting camera path fully automatically, a *high-level semantic spatio-temporal description* of the video is desired. For our current implementation for this we mainly rely on the output of the scene object extraction algorithm introduced above.

The algorithm works in an iterative fashion, shot after shot (separated by a hard cut). The shot length is fluctuating randomly around a base value, which has to be set by the user. For generating a fast-paced video, the base value should be set to a small value (e.g. three seconds), whereas for slow-paced video it can be set to something like ten seconds. After setting the shot length, the scene object extraction algorithm is invoked, which gives us a list of the objects occurring in their scene and how they move. In the next step, a global visited map is calculated, which steers the camera path for the current shot towards areas of the 360˚ video which have not been viewed in the previous shots. A *saliency score* is calculated now for each scene object. The score gives an estimation of how interesting the specific object is for a human viewer. E.g. persons in a scene are usually more interesting than e.g. animals or cars. Furthermore, persons which are moving around (e.g. because they are performing a specific action) are usually more interesting than persons which are static (like persons sitting in the audience). The saliency score of the object is therefore calculated from several cues like the object class, size, average motion magnitude and its isolatedness.

Having calculated the saliency score for each scene object, we determine the focus object as the object with the highest saliency score. The camera path for this shot is now generated simply by tracking the focus object throughout this shot. The result of the automatic cinematography for a 360˚ video showing a live music concert is shown in **Figure 6**. The base shot length has been chosen as three seconds, in order to account for the dynamic in the content.

3.4. Towards 3DOF+ VR Video via 360˚ Omnidirectional Depth Estimation

In order to be able to provide 3DOF+ VR videos through our system and allow for limited viewpoint translation, we developed a deep CNN for 360˚ depth estimation that infers depth directly on the equirectangular domain (*i.e.* without

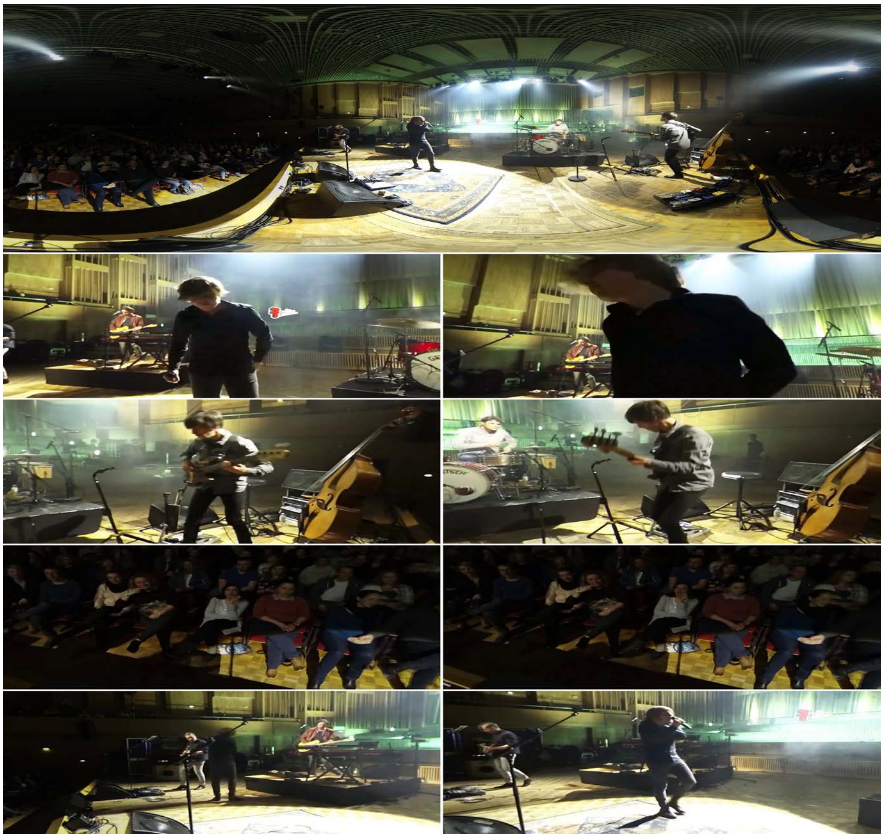

Figure 6. Visualization of the result of the automatic cinematography for a music video (first row). Each row shows two frames from the generated shot (four consecutive shots in total).

projecting to cube map faces), which is presented in [38].

Machine learning models in general, require a lot of ground truth annotated data for training. Although, this is partly addressed for typical pinhole camera data sets by employing depth sensors or laser scanners, it would be very difficult to utilize such methods when using 360˚ cameras due to the larger variation in resolutions between 360˚ color cameras and laser scanners, and because the depth sensor would be in the field of view of the 360˚ camera. To circumvent these difficulties we leveraged existing efforts in creating publicly available large-scale 3D data sets [39] [40] [41] [42], by generating a dataset of 360˚ color images of indoors scenes along with their corresponding ground truth depth annotations via rendering, presented in [43].

For each 3D building of the data sets used, we rendered every room from annotated camera poses if any, or from random camera poses. Additional to the RGB image and the corresponding depth map we also generate a mask to account for missing information of the 3D model. An overview of the data set generation process is presented in **Figure 7**, while **Figure 8** presents a subset of our data set. **Table 1** presents results of our learned model on our data set's test-split, using the metrics described in [42] which are established metrics for depth estimation.

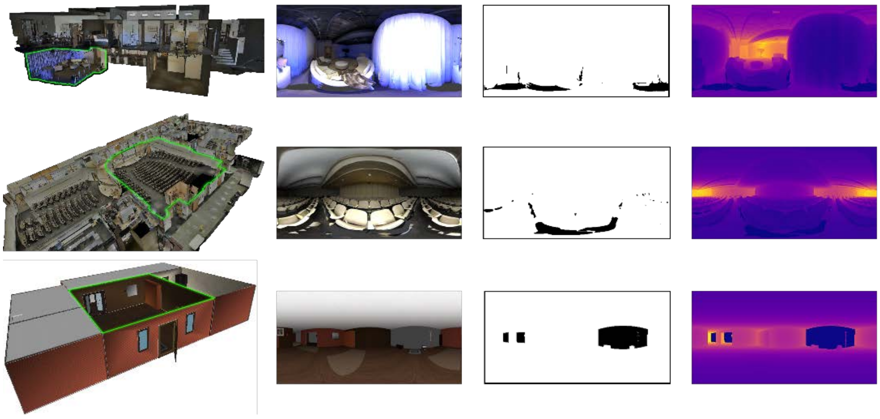

Figure 7. 360˚ data set generation procedure overview. From left to right: 3D model of a building sample with the selected room for rendering outlined in light green, the rendered 360˚ RGB image, a generated depth mask that accounts for missing information (holes) in the depth map, the rendered depth map (the lighter the color, the farther away that pixel is from the camera).

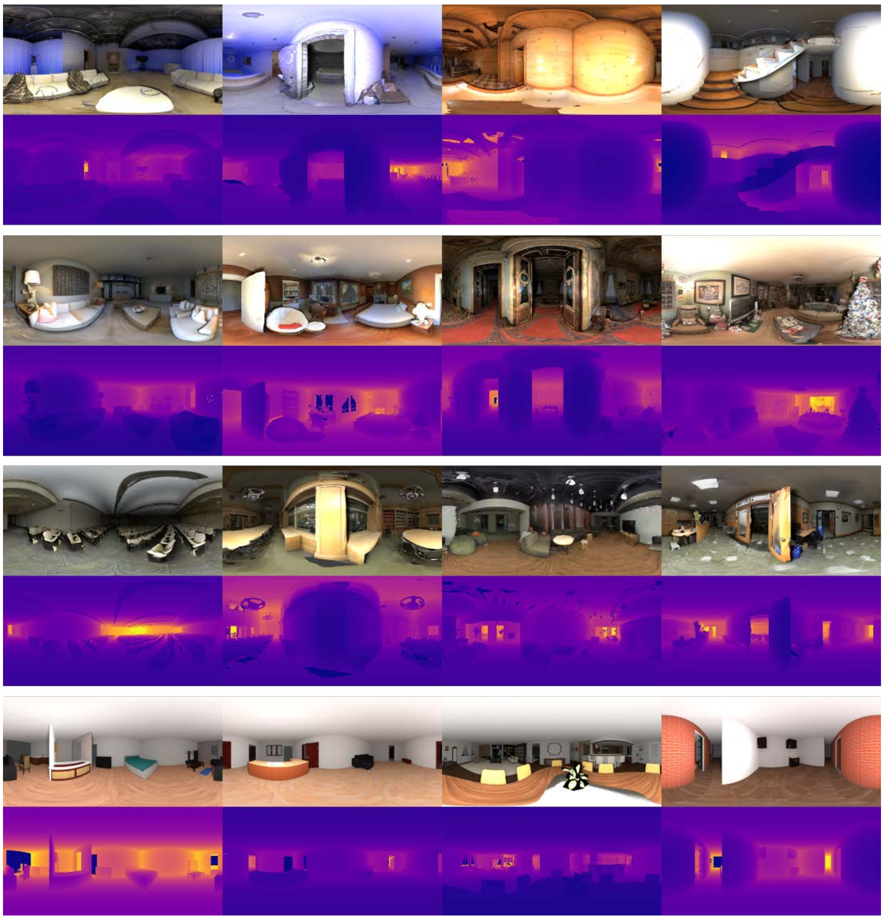

Figure 8. Samples of our generated dataset. For each double row the rendered RGB (above) and its corresponding depth map (bellow).

The average inferring time of our model 0.14 ms, therefore it can run in real-time.

Figure 9 presents qualitative results of our trained model on our data set's test-split. Because our data set is composed of indoor scenes, our network's applicability to outdoor scenes is limited. For this reason we can also refine the quality of the network's prediction by also computing depth maps directly from the stereoscopic imagery captured by our 360˚ cameras. This is demonstrated in Figure 10.

Table 1. Omnidirectional depth estimation results on our data set's test-split. (Downward arrow means lower is better, while upward arrow means that higher is better).

NN model	Abs. Rel↓	Sq. Rel↓	RMSE↓	RMSE (log)↓	$\delta < 1.25$↑	$\delta < 1.252$↑	$\delta < 1.253$↑
UResNet [33]	0.0946	0.0401	0.3084	0.1315	0.9133	0.9861	0.9962

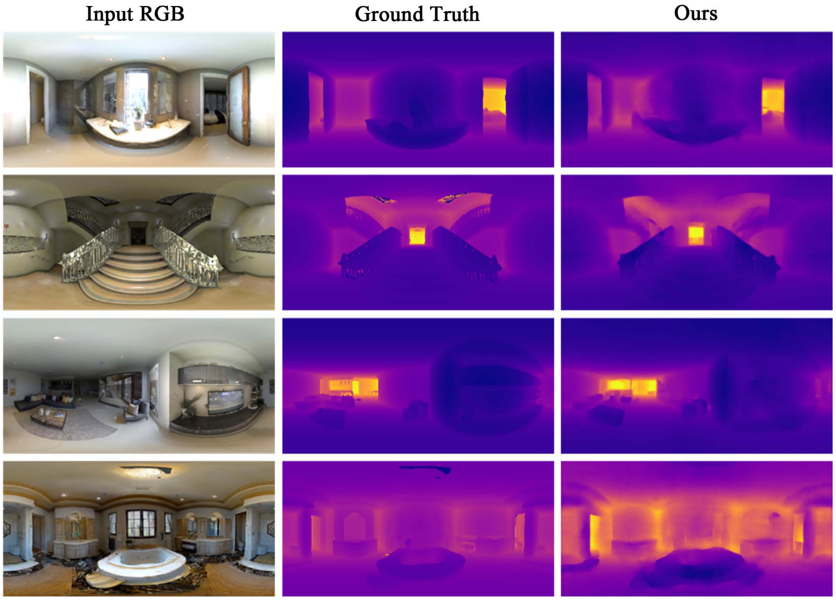

Figure 9. Qualitative results of our depth estimation method in our data set's test-split.

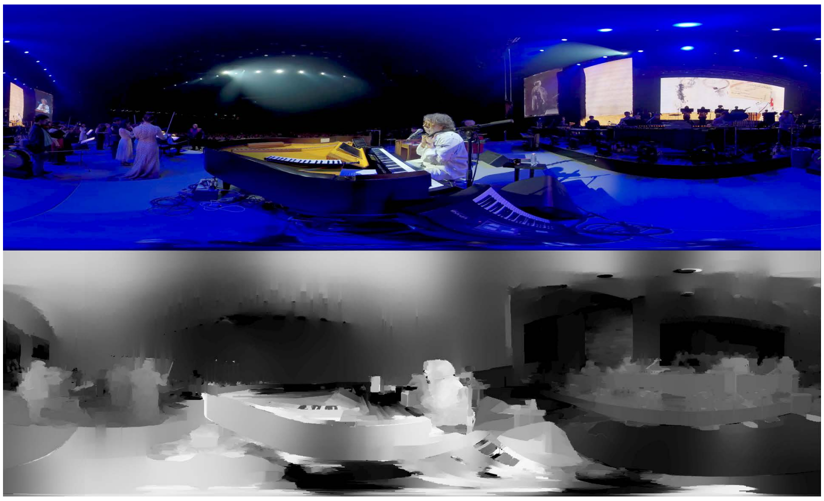

Figure 10. Omnidirectional depth estimation from stereoscopic 360˚ camera using algorithms from [44].

Having estimated the scene's depth, we can synthesize new views and allow for limited translation w.r.t. the motion of the XR device (such as VR headsets, mobile and tablet devices), which offers a more immersive experience to the viewer.

Having presented our results on Depth Estimation, we now turn our attention to the performance characteristics of our core automatic annotation procedure that employs ML-based object detection and tracking.

4. Performance Results

In order to evaluate the performance characteristics of our automated ***scene object extraction*** solutions that detect and track all important elements we used 360˚ VR fashion videos in a special "*5Catwalks*" format as demonstrated in Figure 11. We collected original frontal camera footage from **5 fashion shows** totaling **55 show minutes** and folded them into five 360˚ VR pieces with an average length of **2:11 minutes each.** This transformation turned them into a short and compelling format catering to New Generation Media consumption habits. The resulting test videos represented increasing difficulties and challenges such as noise/fog (for artistic reasons), extreme low view angles resulting in distortions, and crowded scenes with a moving principal camera, etc.

Next, we created **ground truth data** by manually tracking all persons in a total number of **14,339 video frames** with the help of our production-level annotation tool, called *OMNICONNECT* [10]. This step took approximately 6 weeks to complete. Figure 12 demonstrates the output of this manual process for Video #2 (in Figure 11 above) which had total 15 persons tracked and "labeled". Specifically, Figure 12(Left) shows a plot of these Labels vs. their Area. As it is demonstrated, people appear at the end of the catwalk and become larger as they approach the camera before disappearing. In addition, Figure 12(Right) plots Labels vs Yaw, showing the actors appearing on stage in a sequence from each of the 5 distinct directions.

Next, we ran our Automatic Object Detection and Tracking algorithms on the entire data set, totaling a mere **34 minutes** to process the same 14,339 frames and **detected 10,200 objects**. Table 2 shows each of those objects with their *Class ID Strings* and the number of detections (*Count*), respectively, for all 5 videos.

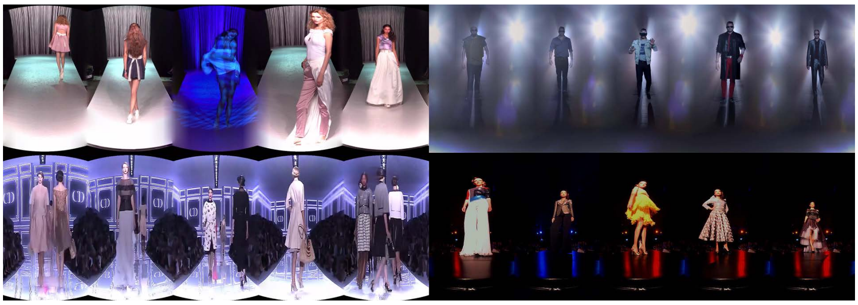

Figure 11. 360VR fashion videos in "5CatWalks" format used to evaluate automatic annotation accuracy. From Top Left Video#1 Standard, Video#2 Noisy, Video#3 Complex, Video#4 Extreme conditions, respectively.

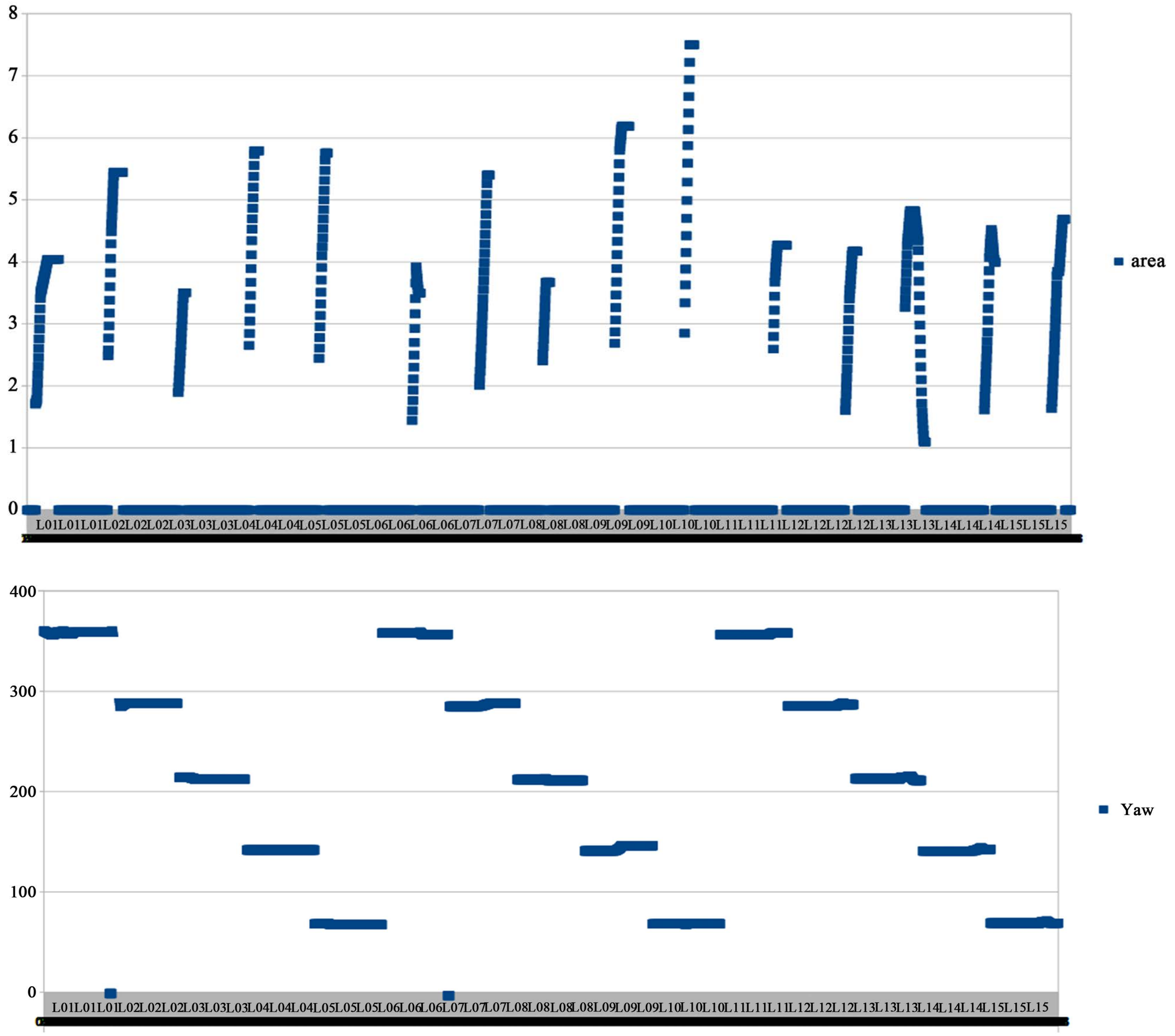

Figure 12. Example of manual ground truth and tracking output (refer to text for detail).

The raw detection performance of our solution (demonstrated in **Figure 13**) produced excellent results. While it did generate a minimal number of false object detections, those were subsequently filtered out using domain-specific knowledge. More concretely, the detection results for the five videos yielded an **average 98.66% accuracy** over the more than 10K objects successfully detected (avg. 1.34% error rate).

More specifically, to create a cleaned up output and eliminate redundancies we grouped the detected labels (or Class Ids) based on their real-life meaning and mapped them onto 3 additional output categories, such as "Face" (Class ID Str = "face", "Human face", "Human head", "Human hair", etc.), "Person" (Class ID Str = "Person", "Man", "Woman", "Girl", "Human body", "Human arm", "Human leg", etc.), "Clothing" (Class ID Str = "Clothing", "Dress", "Footwear', "Trousers", "Skirt", "Jacket", etc.), and finally "False Detections"

Table 2. Automatic detection results on 5 fashion videos (see text).

FashionVideo 1		FashionVideo 2		FashionVideo 3		FashionVideo 1		FashionVideo 1	
Class_id_str	Count	Class_id_str	Count	Class_id_str	Count	Class_id_str	Count	Class_id_str	Count
face	1379	Building	5	Person	81	Person	259	Clothing	777
Clothing	2088	Animal	5	Man	19	face	365	Dress	89
Person	910	Vehicle	44	Clothing	85	Clothing	211	face	397
Building	78	Person	652	face	109	Woman	19	Person	245
Human face	225	face	593	Human face	12	Man	8	Human face	95
Dress	83	Clothing	865	Human arm	1	Human face	3	Swimming pool	1
Man	5	Land vehicle	8	Footwear	1	Dress	12	Woman	53
Woman	73	Train	1	Jacket	1	Human body	5	Human body	25
Footwear	4	Human face	52	Vehicle	1	Sculpture	1	Footwear	15
Trousers	2	Billboard	1					Girl	36
Girl	3	Television	2					Fountain	5
Human body	2	Woman	44					Human leg	1
Sculpture	1	Man	25					Human arm	3
Furniture	1	Dress	40					Human hair	6
		Girl	11					Skirt	4
		Fountain	4					Building	8
		Human body	14					Man	1
		Food	10					Human head	1
		Footwear	4					Furniture	1
		Plant	10						

(Class ID Str = labels that can not appear in a fashion show or out of non-context, like buildings, outside the region of interest, like faces in the audience, etc.).

Figure 14 & Figure 15 show the error rates of this cleaned up detection performance in a graphical form. In Figure 14(Left), the total number of detected objects (good and false results) are shown for each video, while on the right, the overall false detection error rates are plotted. Figure 15, on the other hand, summarizes the global performance characteristics for all videos when mapped onto the selected four main label categories. The concentric doughnut charts refer to Videos #1 through #5 starting with the first one in the center.

As a final step, the manual ground truth data (which contained only the bounding boxes of the models as they walk in front of the camera) was compared to the automatic output of the "Persons" detected by the automatic methods frame-by-frame. This confirmed 100% practical agreement for the envisioned usage of this annotation data, specifically to be able to detect where and what the Viewer is looking at in any moment during an interactive VR section, as well as to guide the underlying automatic camera path generation mechanisms.

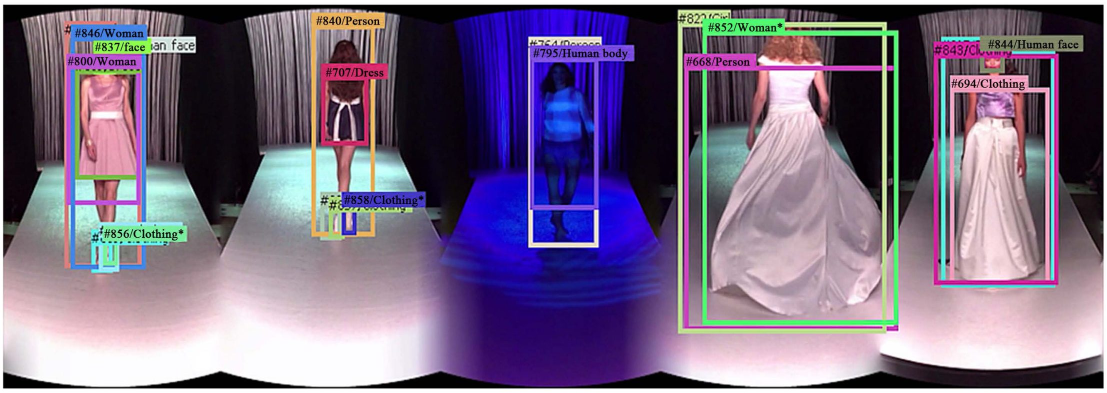

Figure 13. Example of detection output (all 5 videos https://youtu.be/8-0JkLvmDAE).

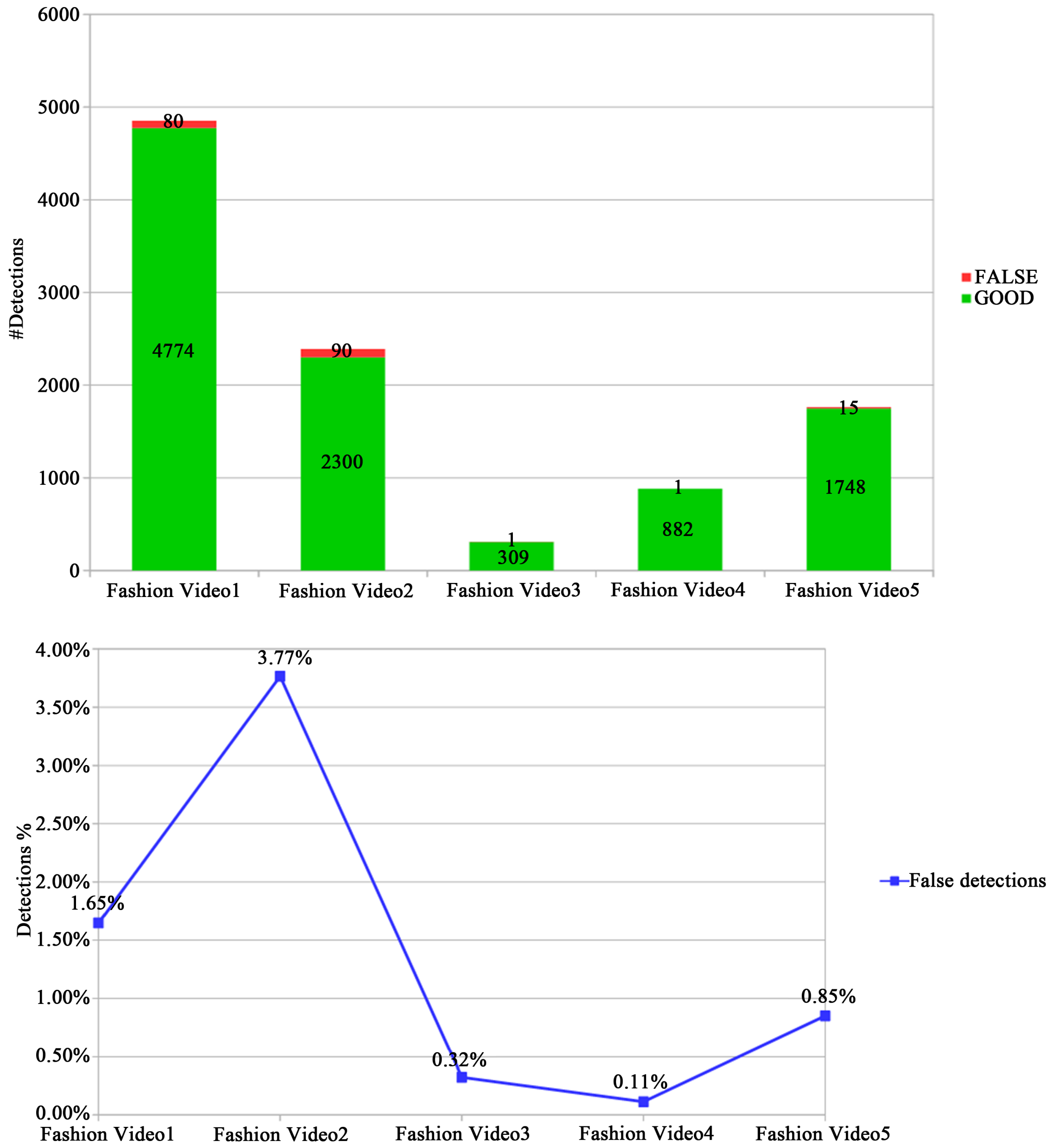

Figure 14. Performance metrics reaching 98.66% average accuracy over 10 K plus detected objects.

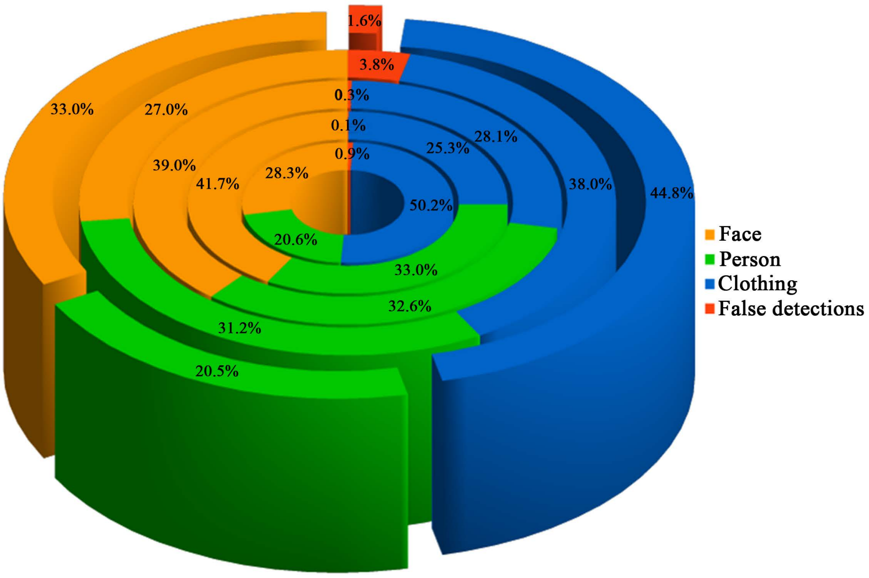

Figure 15. Summary of overall detection performance and results.

5. Conclusions and Future Work

In this paper we have presented four novel challenges that complex VR productions are facing today and offered an integrated solution to address each of these elements via innovative technical solutions. We demonstrated important advances over the state-of-the-art in multiple areas while testing our solution on live productions. Preliminary evaluation of our key annotation, post-processing and 360 scene integration modules using these collected data sets offers a direction for future research.

Specifically, although the initial prototype of the algorithm for automated cinematography works well, there is definitely room for improvements in several directions. Firstly, more semantic information should be extracted from the scene with deep learning methods, in order to understand the video better, which in turn should allow the algorithm to generate a more interesting camera path. More cinematographic techniques (e.g. close-ups) and visual grammar rules for editing/framing should be added, in order to make the generated video more diverse from an artistic point and consequently more interesting to watch. Furthermore, the current prototype of the algorithm relies heavily on the presence of salient objects (especially persons) in the 360˚ video. If no such objects exist in the video (e.g. for a 360˚ video showing a nature scene like an empty beach), an alternative mode should be developed which relies on complementary information.

In addition, we developed a method for omnidirectional depth estimation in order to provide limited viewpoint translating (3DOF+) to the viewer. We overcame data set unavailability by rendering existing 3D datasets, both synthetic and realistic (from large scale 3D scans). Because of the fact that our data set is composed of indoors scenes only, and in the case of 3D scanned data, contains baked light information, our model's performance is limited when testing on real outdoors scenes as well as real captured indoors ones. For this reason we'd like to explore self-supervised methods that use view-synthesis as the supervisory signal, which gives the ability to train neural networks without ground truth data and thus we can circumvent the lack of 360˚ annotated datasets by training with 360˚ video sequences captured using real 360˚ cameras.

We have successfully evaluated the performance of our automatic annotation solution on a complex set of fashion videos demonstrating 98.66% detection and labeling accuracy, proving the practical applicability of our approach while yielding significant time-savings in post-production.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme, grant n˚ 761934, Hyper 360 ("Enriching 360 media with 3D storytelling and personalisation elements").

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Sheikh, A., Brown, A., Evans, M. and Watson, Z. (2016) Directing Attention in 360-Degree Video. *Proceedings of IBC* 2016 *Conference*, Amsterdam, 8-12 September 2016, 1-9. https://doi.org/10.1049/ibc.2016.0029

[2] Tang, A. and Fakourfar, O. (2017) Watching 360˚ Videos Together. In: *Proceedings of the* 2017 *CHI Conference on Human Factors in Computing Systems*, ACM, New York, 4501-4506. https://doi.org/10.1145/3025453.3025519

[3] Fonseca, D. and Kraus, M. (2016) A Comparison of Head-Mounted and Hand-Held Displays for 360° Videos with Focus on Attitude and Behaviour Change. In: *Proceedings of the* 20*th International Academic Mindtrek Conference*, ACM, New York, 287-296. https://doi.org/10.1145/2994310.2994334

[4] Warren, M. (2017) Making Your First 360 Video? Here Are 10 Important Things to Keep in Mind.
https://www.filmindependent.org/blog/making-first-360-video-10-important-things-keep-mind/

[5] Elmezeny, A., Edenhofer, N. and Wimmer, J. (2018) Immersive Storytelling in 360-Degree Videos: An Analysis of Interplay between Narrative and Technical Immersion. *Journal of Virtual Worlds*, **11**, No. 1.
https://doi.org/10.4101/jvwr.v11i1.7298

[6] Malik, R. and Bajcsy, P. (2008) Automated Placement of Multiple Stereo Cameras. *The* 8*th Workshop on Omnidirectional Vision*, *Camera Networks and Non-Classical Cameras—OMNIVIS*, October 2008, Marseille, France.

[7] Chikkerur, S., Sundaram, V., Reisslein, M. and Karam, L.J. (2011) Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting*, **57**, 165-182.
https://doi.org/10.1109/TBC.2011.2104671

[8] Lai, W.S., Huang, Y., Joshi, N., Buehler, C., Yang, M.H. and Kang, S.B. (2018) Se-

mantic-Driven Generation of Hyperlapse from 360 Degree Video. *IEEE Transactions on Visualization and Computer Graphics*, **24**, 2610-2621. https://doi.org/10.1109/TVCG.2017.2750671

[9] Song, S., Zeng, A., Chang, A.X., Savva, M., Savarese, S. and Funkhouser, T. (2018) Im2Pano3D: Extrapolating 360˚ Structure and Semantics beyond the Field of View. 2018 *the IEEE Conference on Computer Vision and Pat tern Recognition*, Salt Lake City, UT, 18-23 June 2018, 3847-3856. https://doi.org/10.1109/CVPR.2018.00405

[10] Hyper 360 Project (2019) http://www.Hyper360.eu/

[11] Adobe (2019) Creative Cloud Premier Pro & After Effects. https://www.adobe.com/creativecloud/video/virtual-reality.html

[12] Insta 360 (2019) Stitching Software. https://www.insta360.com/download/insta360-pro

[13] SGO (2019) Mistika VR Optic Flow Stitcher. https://www.sgo.es/mistika-vr/

[14] Andersson Technologies (2019) SynthEyes 3D Camera Tracking and Stabilization Software. https://www.ssontech.com/synovu.html

[15] Kopf, J. (2016) 360˚ Video Stabilization. *ACM Transactions on Graphics*, **35**, Article No. 195. https://dl.acm.org/citation.cfm?id=2982405 https://doi.org/10.1145/2980179.2982405

[16] Matos, T., Nóbrega, R., Rodrigues, R. and Pinheiro, M. (2018) Dynamic Annotations on an Interactive Web-Based 360 & Deg; Video Player. In: *Proceedings of the* 23*rd International ACM Conference on* 3*D Web Technology*, ACM, New York, Article 22. https://doi.org/10.1145/3208806.3208818

[17] WondaVR (2019) https://www.wondavr.com

[18] Liquid Cinema (2019) https://liquidcinemavr.com

[19] Fader (2019) https://getfader.com

[20] ThingLink (2019) https://www.thinglink.com

[21] Viar 360 (2019) https://www.viar360.com

[22] Sprawly (2015) http://twittertechnews.com/virtualreality/sprawly-the-worlds-first-virtualreality-searchengine-httpsprawly-co/

[23] Viond Re’flekt (2019) https://www.viond.io/

[24] Omnivirt (2019) https://www.omnivirt.com/

[25] 3DVista Pro (2019) https://www.3dvista.com

[26] Su, Y.-C., Jayaraman, D. and Grauman, K. (2017) Pano2Vid: Automatic Cinematography for Watching 360˚ Videos. In: Bares, W., Gandhi, V., Galvane, Q. and Ronfard, R., Eds., *Eurographics Workshop on Intelligent Cinematography and Editing*, The Eurographics Association, Lyon, France.

[27] Su, Y.-C. and Grauman, K. (2017) Making 360˚ Video Watchable in 2D: Learning Videography for Click Free Viewing. 2017 *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 21-26 July 2017, 1368-1376.

[28] Hu, H.-N., Lin, Y.-C., Liu, M.-Y., Cheng, H.-T., Chang, Y.-J. and Sun, M. (2017) Deep 360 Pilot: Learning a Deep Agent for Piloting through 360˚ Sports Videos. In: *Proceedings of* 2017 *IEEE Conference on Computer Vision and Pattern Recognition*, ACM, New York, 1396-1405.

[29] Truong, A., Chen, S., Yumer, E., Li, W. and Salesin, D. (2018) Extracting Regular FOV Shots from 360 Event Footage. In: *Proceedings of the* 2018 *CHI Conference on Human Factors in Computing Systems*, ACM, New York, 316.

https://doi.org/10.1145/3173574.3173890

[30] Huang, J., Chen, Z., Ceylan, D. and Jin, H. (2017) 6-DOF VR Videos with a Single 360-Camera. 2017 *IEEE Virtual Reality*, Los Angeles, CA, 18-22 March 2017, 37-44. https://doi.org/10.1109/VR.2017.7892229

[31] Wang, F.-E., Hu, H.-N., Cheng, H.-T., Lin, J.-T., Yang, S.-T., Shih, M.-L., Chu, H.-K. and Sun, M. (2018) Self-Supervised Learning of Depth and Camera Motion from 360˚ Videos. *CoRR*, abs/1811.05304.

[32] Thatte, J., Boin, J.B., Lakshman, H. and Girod, B. (2016) Depth Augmented Stereo Panorama for Cinematic Virtual Reality with Head-Motion Parallax. 2016 *IEEE International Conference on Multimedia and Expo*, Seattle, WA, 11-15 July 2016, 1-6. https://doi.org/10.1109/ICME.2016.7552858

[33] Zioulis, N., Karakottas, A., Zarpalas, D. and Daras, P. (2018) OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018. *ECCV* 2018. *Lecture Notes in Computer Science*, Springer, Cham, 448-465. https://doi.org/10.1007/978-3-030-01231-1_28

[34] Youtube (2017) Hot and Cold: Heatmaps in VR. https://youtube-creators.googleblog.com/2017/06/hot-and-cold-heatmaps-in-vr.html

[35] Tsatsou, D., Dasiopoulou, S., Kompatsiaris, I. and Mezaris, V. (2014) LiFR: A Lightweight Fuzzy DL Reasoner. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I. and Tordai, A., Eds., *The Semantic Web*: *ESWC* 2014 S*atellite Events*. *ESWC* 2014. *Lecture Notes in Computer Science*, Springer, Cham, 263-267. https://doi.org/10.1007/978-3-319-11955-7_32

[36] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. *Computer Science*, arXiv: 1804.02767. http://arxiv.org/abs/1804.02767

[37] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D. and Bischof, H. (2009) Anisotropic Huber-L1 Optical Flow. In: Cavallaro, A., Prince, S. and Alexander, D., Eds., *Proceedings of the British Machine Vision Conference*, BMVA Press, London, 108.1-108.11. https://doi.org/10.5244/C.23.108

[38] Karakottas, A., Zioulis, N., Zarpalas, D. and Daras, P. (2018) 360D: A Dataset and Baseline for Dense Depth Estimation from 360 Images. 1*st Workshop on* 360*o Perception and Interaction*, *European Conference on Computer Vision* (*ECCV*), Munich, Germany, 8-14 September 2018, 1-4.

[39] Handa, A., Pătrăucean, V., Stent, S. and Cipolla, R. (2016) Scenenet: An Annotated Model Generator for Indoor Scene Understanding. *2016 IEEE International Conference on Robotics and Automation*, Stockholm, 16-21 May 2016, 5737-5743. https://doi.org/10.1109/ICRA.2016.7487797

[40] Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M. and Savarese, S. (2016) 3D Semantic Parsing of Large-Scale Indoor Spaces. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 27-30 June 2016, 1534-1543. https://doi.org/10.1109/CVPR.2016.170

[41] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A. and Zhang, Y. (2017) Matterport3D: Learning from RGB-D Data in Indoor Environments. 2017 *International Conference on* 3*D Vision*, Qingdao, 10-12 October 2017, 667-676. https://doi.org/10.1109/3DV.2017.00081

[42] Eigen, D., Puhrsch, C. and Fergus, R. (2014) Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D. and Weinberger, K.Q., Eds., *Proceedings of the* 27*th Interna-*

tional Conference on Neural Information Processing Systems, MIT Press, Cambridge, MA, 2366-2374.

[43] Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M. and Funkhouser, T. (2017) Semantic Scene Completion from a Single Depth Image. 2017 *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 21-26 July 2017, 1746-1754. https://doi.org/10.1109/CVPR.2017.28

[44] PseudoScience (2019) Volumetric 360 6DOF Video/Stereo2Depth Conversion Algorithm. http://pseudoscience.pictures/index.html