D6.4
Virtual Production prototype toolkit

Grant Agreement nr	780470
Project acronym	SAUCE
Project start date (duration)	January 1st 2018 (36 months)
Document due:	29/02/2020
Actual delivery date	25/02/2020
Leader	Filmakademie [FA]
Reply to	jonas.trottnow@filmakademie.de
Document status	Submission Version

Project funded by H2020 from the European Commission

Project ref. no.	780470
Project acronym	Smart Asset re-Use in Creative Environments
Project full title	SAUCE
Document name	D6.4 Virtual Production prototype toolkit
Security (distribution level)	PU
Contractual date of delivery	29/02/2020
Actual date of delivery	25/02/2020
Deliverable name	D6.4 Virtual Production prototype toolkit
Type	DEM
Status & version	Submission Version
Number of pages	21
WP / Task responsible	FA
Other contributors	UPF, DRZ
Author(s)	Jonas Trottnow, Simon Spielmann, Volker Helzle
EC Project Officer	Ms Adelina Cornelia Dinu - adelina-cornelia.dinu@ec.europa.eu
Abstract	With this deliverable a demonstrator for the usage of smart assets in a virtual production toolkit is introduced. An exemplary semantic descriptor developed by DRZ provides labels for a 3D scene. Based on the labels, the scene can be set up for a usage in the Virtual Production Editing Tools (VPET) developed by FA. On top, VPET has been extended to support character animations that can be context and scene aware. An API and protocol has been developed to transfer an animatable character to the VPET clients. A user can then direct the character by defining a new position and walking path for them through the tablet frontend. The new position is sent to an arbitrary animation engine and solved there to a bone animation. An implementation of the protocol for the MEDUSA animation engine is developed by UPF. FA has also implemented a demo animation solver using the Unity based scene host as an animation engine. Further this deliverable introduces a possibility to import smart assets and scenes directly into VPET without first importing assets to applications like Unity or Katana. This direct importer reads USD (universal scene description) files and provides them to the clients directly.
Keywords	Virtual Production, Character Animation, Semantic Description
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	No
Read by partners	No

Mgt. Board approval	No

Document History

Version and date	Reason for Change
1.0 21-10-2019	document created by Jonas Trottnow
1.1 20-02-2020	Version for internal review
1.2 25-02-2020	Revisions in response to internal review: final versions submitted to Commission

Table of Contents

1 EXECUTIVE SUMMARY

With this deliverable a demonstrator for the usage of smart assets in a virtual production toolkit is introduced. An exemplary semantic descriptor developed by DRZ provides labels for a 3D scene. These labels transform the 3D scene into a smart asset providing not only the pure geometry data, but also additional metadata, valuable in a virtual production scenario. This also makes the 3D scene easily reusable. Based on the labels, the scene can be set up for a usage in the Virtual Production Editing Tools (VPET) developed by FA. VPET has been identified as the most appropriate virtual production platform for this purpose in D6.2. Through the labels generated by the semantic descriptor, movable objects like cars, benches etc. can be separated from static parts like houses. Movable objects are then marked as editable in the VPET system, enabling a user to select and modify it in realtime on set with the VPET tablet clients.

On top, VPET has been extended to support character animations that can be context and scene aware. This is made possible by separating the solving of the animation from the actual visualization of the animation on the VPET clients. An API and protocol has been developed to transfer an animatable character to the VPET clients. A user can then direct the character by defining a new position and walking path for them through the tablet frontend. The new position is sent to an arbitrary animation engine and solved there to a bone animation. This solving can include semantic behaviour (e.g. grab a door handle to open and walk past a door) as well as scene awareness (e.g. collision avoiding path planning). These animations are sent back to all VPET clients in real-time, visualizing the animation there. An implementation of the protocol for the MEDUSA animation engine is developed by UPF. FA has also implemented a demo animation solver using the Unity based scene host as an animation engine. This concept makes existing characters and animations reusable and smart. In order to evaluate the usage of smart assets in virtual production, a use case for the Search and Transformation framework developed in WP4 in a virtual production environment is discussed in this deliverable as well.

Further this deliverable introduces a possibility to import smart assets and scenes directly into VPET without first importing assets to applications like Unity or Katana (both already supported by VPET). This direct importer reads USD (universal scene description) files and provides them to the clients directly.

2 BACKGROUND

Virtual productions get increasingly common in modern movie productions. The possibilities to visualize, edit and explore virtual 3D content directly on a movie set make it invaluable for VFX rich productions. Many of the virtual production scenarios also involve animated characters and motion capturing. But the complexity of animation systems prohibits its usage on an interactive film set. Within SAUCE an extensive research on available virtual production tools and frameworks has been carried out (D6.2). While most of them are not publicly available or open source, none of them had the possibility to interactively and intuitively animate characters on set.

The R&D team of Filmakademie Baden-Württemberg developed the open source 'Virtual Production Editing Tools' (VPET) over the last years. With as little hardware overhead as possible, VPET offers the possibility to stream an arbitrary 3D scene to tablet clients. On the tablets, the 3D scene can be aligned with the real world in augmented reality (AR) making the tablet a window to the provided set extension. Users can explore and edit 3D elements, lighting as well as rigid body animations. All clients and the scene host communicate changes among themselves through a synchronisation server, keeping the scene consistent.

Along with the intent of WP6, this deliverable presents a real-time control system for authoring animated content using smart assets. It also addresses the challenges of automatically synthesizing new scenes from existing ones in the scope of virtual productions and how to integrate smart assets into virtual production in general. Several aspects addressing animation in WP6 could be connected and integrated towards a virtual production use case. Relations to work packages like WP4 (Search and Transformation framework) and WP5 (Smart Asset transformations) are laid out.

3 INTRODUCTION

Although virtual productions get increasingly popular, setup time is still quite high. Often tools are implemented only for a single use case utilizing manually prepared special assets. This represents a large overhead for a production and adds additional pipeline requirements on it. VPET already tried to go into another direction. A versatile tool was created to view, edit and explore arbitrary 3D content used e.g. for set extensions directly on set. Special attention was given to the user interface. Directors, artists, Director of Photography and other set staff should be enabled to interact with the virtual set by simply grabbing a tablet being a window to the scene, augmented with 3D content.

Nevertheless there is still an overhead by preparing a scene for the virtual production system. To make it attractive also for Previsualizations and other usage scenarios this needed to become more convenient and streamlined with large production pipelines. Often existing assets are used in previsualizations. Smart assets are a solution here. By (automatically) labeling 3D assets, they can be fastly found and reused. The labels can then be used further to support scene preparation and provide semantic information to animation engines. A demonstrative implementation of making use of all these aspects is presented with this deliverable.

3.1 Main objectives and goals

This deliverable describes concepts and a demonstrator for the utilization of smart assets in virtual production scenarios. The main objective is to explain how and why a virtual production can benefit from smart assets. Along with this, a functional demonstrator is developed, implementing these aspects.

3.2 Methodology

Based on the requirement to fastly and easily set up a virtual production asset, concepts and associated implementations are developed to demonstrate the usefulness of smart assets. By providing a full implementation, a pipeline is established that can easily be adopted by studios and other interested people when approaching virtual productions. SAUCE thereby provides a generic foundation platform for smart virtual productions.

3.3 Self Assessment

The Objective of WP6 is to develop and demonstrate real-time control systems for authoring animated content using smart assets, automatically synthesizing new scenes from existing ones and integrating smart assets into virtual production scenarios with editable cameras and lights. With this report on the developed demonstrator, this has been implemented within VPET. To achieve this, two main aspects have been successfully addressed that will be reported on in this document.

Especially with IKinema leaving, time schedules for their efforts in SAUCE (now taken over by UPF) were delayed and explain the 2 month delay of this deliverable. Nevertheless interfaces have been implemented and tested to easily integrate smart asset usage into virtual productions through VPET. Experimental productions including the developments reported on in this document, have been planned and communicated in D8.2. A new release of VPET is being published on the iOS App Store and is thereby accessible for any interested people, creative user groups and content creators.

VPET has been identified to be the most appropriate solution for this work as has been reported on in D6.2. FA and UPF presented a joined short paper and poster on "Approaching real-time Character Animation in Virtual Productions" at the Conference on Visual Media Production (CVMP) in December 2019 in London [4].
By extending VPET with the animation capabilities described in this deliverable and addressing multiple other aspects also being reported on here, VPET was able to be lifted from a technology validated in lab (TRL level 4) to a tool ready to be demonstrated in experimental productions (TRL level 6).

4 Character animation in VPET

To approach character animation in the virtual production toolset, FA developed an open character streaming protocol for VPET. The entire character (including weights, skeleton etc.) can be transferred to the tablets at run time. The newly developed API then allows arbitrary external animation solving engines to animate the character through streamed bone animations. These animations are represented as a root bone translation and a new rotation for each bone. The updated pose is automatically synchronized and held consistent between participating tablet clients.

This interface extends the open architecture of VPET [3]. Although the client is based on Unity, any host (providing the scene to the VPET clients) could be connected. We have demonstrated this with Foundry Katana in the past. The latest addition allows to connect an arbitrary animation engine.

VPET tablet user interface for character animation, crosshairs is the target waking position for the character, character automatically navigates between the cars

4.1 Concept

It does not make sense to author a complete animation from scratch with an on-set virtual production tool set, as it requires too much time and expertise to get a convincing result. Nevertheless providing the possibility to easily direct virtual characters on a film set is often desired.

Tablets offer an intuitive way of interacting with elements during a virtual production e.g. in augmented reality (AR) being useful for directing characters. Fast visualisation and ease of use are the main targets for our work. Towards this goal, FA and UPF are working on a joint effort within the SAUCE project. On set, only high level commands can be used to drive a character. Commands like 'Go there' should be used on e.g. the VPET tablet tools. This requires that procedural animations are generated and that the character is animated in a scene aware manner. Obstacles should be avoided, uneven grounds need to be compensated etc. These technologies are also known from game creation, but only slowly they are introduced into the film industry. Such complex, highly adaptable animation solving required for virtual productions, cannot be executed on a tablet conveniently and with the required customizability. Therefore we propose an interface for animation data streaming, leaving all complex and resource intensive animation tasks to the host application running the animation engine.

4.2 VPET API

VPET [3] supports the usage of multiple client tablets in the same scene at the same time. All updates applied to scene objects by a client are communicated to all others through the SyncServer. The SyncServer is a standalone software running on the same computer serving the scene. The communication and redistribution of updates is realized by using zeroMQ, an open source software library implementing a vast amount of networking patterns as well as solutions for e.g. automatic queuing of messages etc. To support external animation engines, or more general external tools, the SyncServer was enabled to also listen and redistribute updates coming from a standard TCP/IP port. Thereby the external tools do not need to implement zeroMQ but are able to utilize TCP messages.

A new class of update message was also implemented to support character animation updates. This message contains the updated position of the hip bone, a number to identify the character and an quaternion for each bone in the rig, representing the new rotation of it. This update is sent by the animation engine and applied to the character on each VPET client. A user of the tablet tool can define a target position for a character by clicking on it within the tablet user interface. This target position is sent to the animation engine, which in turn is able to solve the animation for the character.

Even before the SAUCE project, a scene transfer API was implemented. The VPET clients are available as general purpose Virtual Production Editing Tool, not limited to a specific production or scene. Instead the scene transfer API allows arbitrary software to serve the scene to the clients. The API is currently implemented for Foundry's Katana and the Unity engine. Additionally, as it is an open protocol, it can be adapted for any software capable of providing scene data. The protocol is able to handle mesh information, textures, materials, scene graphs as well as instancing information.

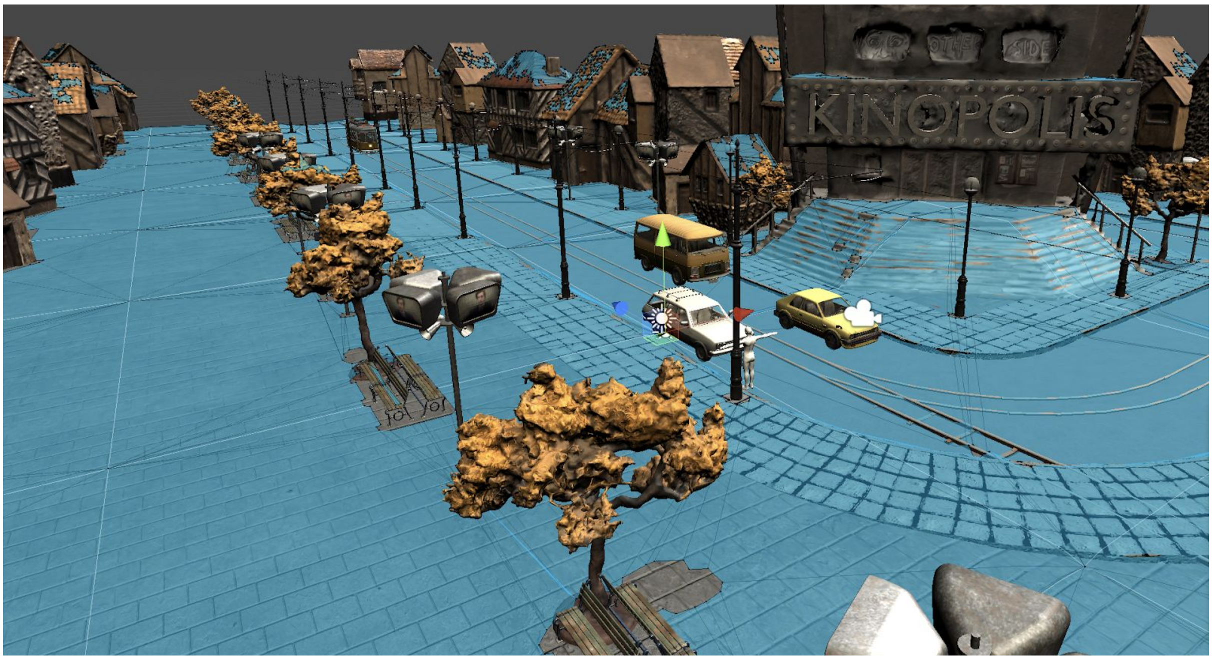
Within SAUCE, the scene transfer protocol has been extended to support animated characters. A skeleton and bone mapping is transferred for each character. Each vertex of the character mesh can be influenced by up to 4 bones. This mapping is transferred in form of vertex weights. The initial bind pose is also included into the extended scene transfer protocol. This bind pose defines the initial pose a character mesh is in. Generally this is the so-called T-pose. By transferring this bind pose, also other poses can be supported.

With the latest updates of the Unity Engine it became necessary to also change to AR backend. Instead of using ARKit directly, Unity now requires to use ARFoundation, a generalized AR API. By changing this API, VPET is now also prepared to run on Android mobile devices, formerly the client was only able to run on iOS devices with AR capabilities. Also the tracking accuracy has been greatly improved by upgrading the API.

A new release of VPET is being published on the iOS App Store and is thereby accessible for any interested people, creative user groups and content creators.

4.3 Unity based animation engine demonstrator

To demonstrate the functioning of the implemented protocol and API, the VPET Unity Host has been extended to also serve as an animation solver engine. Before, the Unity Host was used to provide a Unity scene to all VPET clients. Unity, being a fully featured game engine, provides a rich API to add autonomous and scene aware characters to a game. This Unity API has been used to implement an animation engine, streaming solved animations to the VPET clients.



Navigation Mesh (blue) generated for the 50 megatons scene in Unity

As a preparation step, user adjustable so-called navigation meshes (NavMesh) are generated by the Unity Engine. These meshes represent walkable areas in a 3D scene for a character. In the image above, the NavMesh of the utilised scene is represented in blue. In addition all moveable objects in a scene are labeled as obstacles for a character. Movable objects are objects that a user can select and edit in the VPET clients. This means they can be arbitrarily moved in the scene or even animated as 'rigid body'.

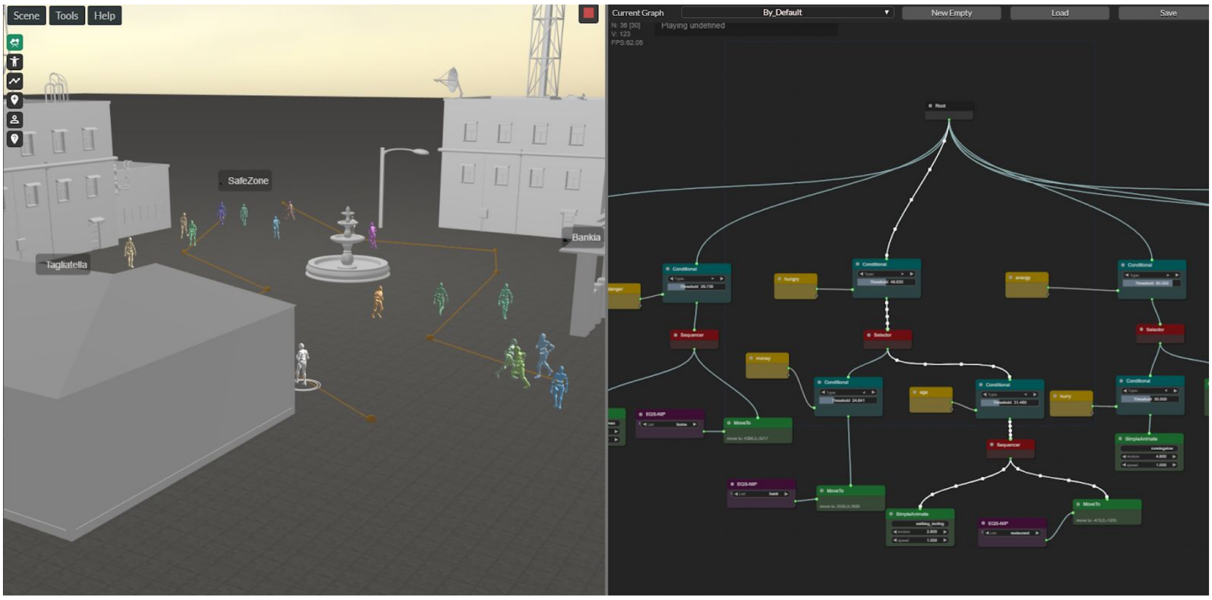
For a character a so-called locomotion tree is setup in Unity. For each direction a character can walk towards (left, right, forward, backward) and for different speeds, an animation cycle can be defined. These animation cycles will then be blended together by Unity to form the final animation. This final animation is computed by the Unity animation engine purely based on the target position a character should go to, which has been defined by a user of the VPET clients. The calculated path automatically avoids obstacles and areas not defined as walkable.

This demonstrator makes full use of the introduced API and implementation in VPET. It uses Unity functionality to solve the animation.

4.4 MEDUSA

UPF is working on a machine learning (ML) based human character animation engine. The ML network learns how humans move in different styles (e.g. running, walking, sad, happy...). Through the core technology, the learned animation can be applied to an arbitrary human character. This approach is thought for virtual characters which receive the orders directly from the artist. Furthermore, UPF is working on another approach, focused on automatically controlling background characters, reducing the time spent if this task was done manually. UPF has developed a web tool, called MEDUSA, in order to determine the characters' identity, including the behaviour and the style. This background character will be controlled by a graph programmed Hybrid Behaviour Tree, an evolution of common Behaviour Trees.

With this approach virtual characters are always aware of their surroundings to allow the Hybrid Behaviour Tree system to compute the final action taking into account the characters properties and their environment. Additionally a character can interact with scene elements and extract relevant information from them. For example, a virtual character could go to a store and acquire some item needed to perform a concrete task, or react in real-time to changes on the scenario or its properties, adapting automatically the animations to those events. All this can be authored in a low-coding tool in a web browser as the engine is based on WebGL [1].



Interface of the web based Hybrid Behaviour Tree editor MEDUSA

The Hybrid Behaviour Tree system is controlled by a library called HBTree.js, which is agnostic to the rest of the components of the tool. This fact allows that library to be integrated into different web based render engines, as an API is provided to overwrite the needed methods (used by specific nodes) to access the scenario information and the virtual character properties. The generated HBT only deals with data and returns a behaviour, which is interpretable by each approach. Thus, Medusa could be used as a testing or simulation platform for other developments, like game AIs or virtual productions.

Through the introduced VPET animation streaming API, both approaches are easily integrated into the tablet tools and can even coexist within the same scene. FA provided the training data for the ML algorithm as high quality optical motion capturings [2]. This database is also released publicly.

4.5 Summary

The combination of a new character animation streaming API, that can be used flexible with any animation engine, and a couple of animation solving engines for different purposes form a solid foundation for adding intuitive, high-level character animation to an on-set virtual production.

In future, even more interaction possibilities can be added to the tablet clients as well as the animation solver. The tablet client could e.g. be extended to select different walking styles for a character and look-at positions could be defined as well.

FA and UPF presented a joined short paper and poster on "Approaching real-time Character Animation in Virtual Productions", representing the work reported on in this chapter, at the Conference on Visual Media Production (CVMP) 2019 in London [4].

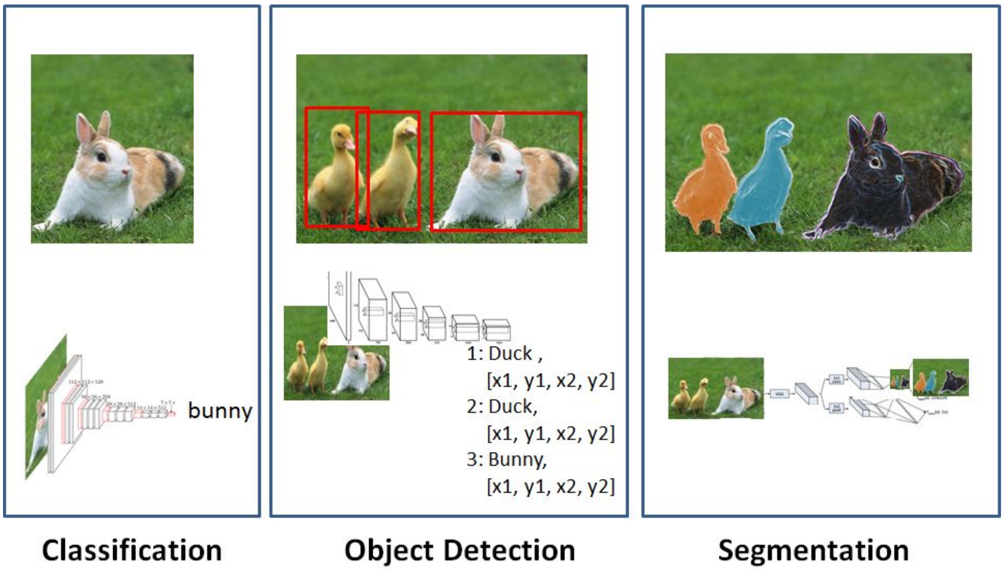
5 Semantic Description: Preparation & Application

FA prepared and distributed sets of 3D data coming from former productions at Filmakademie. These have been used by DRZ to extract labels (semantic descriptors) for the assets included in the dataset. Thereby the assets have been converted to smart assets, that could also be stored in the Search & Transformation framework developed by DNEG. In addition to the pure searching, finding and reusing aspects of this revaluing approach, the labels generated for the assets can on top be used for the semantic animation aspects developed in SAUCE. Based on the labels, an autonomous character can be enabled to understand the surrounding 3D scene and act appropriately.

5.1 Background

Interesting for VFX work are two main data types: 2D images and videos of all types and 3D scenes in all its diversity. There are three known approaches for automatic 2D image recognition:

- Image classification - creating a list of words associated with an image
- Image detection - creating bounding boxes around object classes
- Image segmentation - creating pixel masks around each object



Source: http://ataspinar.com/wp-content/uploads/2017/11/deeplearing_types.png

Each of these methods has to be trained with supervised known training datasets. Typically the public datasets have 1000 classification words, +100 for detection, 100 for segmentation (indoor & outdoor combined). This can be done for objects, captioning, individual instances, faces, emotion, dominant colors, extracted text, etc.
All big tech companies offer cloud based image recognition Amazon[1], IBM[2] and Google[3].

Adding 3D scene classification/object detection to already existing 2D classificators, is a rather new field but might therefore be especially interesting. A diverse set of data is already available from the 3D representation itself (contrasting 2D image analysis) and is thereby already providing metadata to be used in a scene description and possible input for machine learning: lights, geometry, animated rigs, shading parameters, textures etc. all arranged in a scene hierarchy tree. Especially the available tree hierarchy provides perfect segmentation opportunities without much effort (only the correct cut

[1] https://aws.amazon.com/rekognition/
[2] https://www.ibm.com/watson/services/visual-recognition/
[3] https://cloud.google.com/vision/

in the tree hierarchy needs to be found). Finding meaningful ‚cuts' in the tree hierarchy separating e.g. a car from a street scene, a plant from a forest, an arch from a plant, a leaf from an arch etc. becomes a new challenge. Object localization (a key aspect of 2D machine learning) in terms of model X stands on 3D position Y in Scene Z is rather irrelevant for the envisioned use cases as reusability of existing assets is key. Main task would be tagging these extracted assets with searchable/findable words based on the above described data (e.g. for a car: SUV, truck, van, sports car, old-timer, luxurious interior, brand, model). Also lighting could be tagged and extracted from the scene (tags could be: night, daylight, twilight), same applies to animation rigs (tags: running, falling, jumping…). Also additional metadata like ‚highly detailed', ‚only usable in background' could be interesting to extract.
As a note, the semantic metadata needs sufficient training examples to learn the classification correctly.

5.2 Usage Scenario

The envisioned usage scenario would be that a 3D scene like 'The Ranch' or the '50 megatons' scene serves as input. The scene can then either be manually described or is fed into a machine learning framework. This framework would generate an output like the XML provided below. Technical, semantic as well as contextual descriptions of the scene and the individual assets need to be filled to make them reusable. Otherwise they will not be detectable through a search framework.

In a virtual production context the descriptions (metadata) greatly simplify the setup and authoring of an initial scene. Not only can assets be found conveniently, that are explicitly suiting virtual production needs (e.g. real-time capable rigs, meshes with less polygons etc.) but the metadata also serves as guide for the transformation part of frameworks like the search and transformation framework developed within SAUCE. If an asset does not fulfill all requirements of a search, the framework can look up if transformation plugins are capable of automatically transforming the asset to fulfill all search criterias. Based on the labels, the scene could be set up for a usage in the Virtual Production Editing Tools (VPET) developed by FA. Through the labels, movable objects like cars, benches etc. can be separated from static parts like houses. Movable objects are then marked as editable in the VPET system, enabling a user to select and modify it in realtime on set with the VPET tablet clients. In addition the labels can be used to provide scene understanding to e.g. the animation solving engine in order to generate scene aware animations.

5.3 Exemplary scenes and demonstrators

This chapter presents multiple 3D scenes that have been and are used in current Filmakademie productions. They serve as demonstrators for the Semantic Descriptors and other aspects of SAUCE. Additionally they are partly used in experimental productions.

5.3.1 The Ranch

This scene is based on the Filmakademie Production "Tears in Heaven". As these digital assets have been prepared for a typical VFX/CG workflow, they serve as a reference dataset for various prototype environments in Filmakademies' research projects.

Renderings of 'The Ranch' scene by FA

5.3.1.1 Semantic Description

The semantic description will be presented in human and machine readable XML style. The scene as a whole and each individual object are described for demonstration purposes. A semantic description as the one presented here, could be added to the universal scene description (USD) as an addon. USD will be addressed in the next paragraph. Separated assets would be stored in detached USD files to be easily included into new projects. The scene itself would then be a nested tree of USD assets enriched with metadata contextually describing the asset.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<scene>
	<name>ranch scene</name>
	<global_scene_descriptor>
		<daytime>night</daytime>
		<mood>cold night</mood>
		<lighting>
			<type>mixed lighting</type>
			<detected_sources>
				<lightsource>moon</lightsource>
				<lightsource>lantern</lightsource>
			<detected_sources>
		</lighting>
		<tags>
			<tag>outdoor</tag>
			<tag>nature</tag>
			<tag>wooden buildings</tag>
			<tag>courtyard</tag>
			<tag>multiple leafed trees</tag>
			<tag>rocket launching pad</tag>
			<tag>ranch in a moody night</tag>
			<tag>moonlight + tungsten lanterns illumination</tag>
		</tags>
	</global_scene_descriptor>
	<assets>
		<asset>
			<name>house</name>
			<technical>
				<type>static mesh</type>
				<3Dscan>false</3Dscan>
				<polycount>56147</polycount>
				<animated>false</animated>
			</technical>
			<classification>
				<high_level>house</high_level>
				<lower_level>wooden cabin</lower_level>
				<appearance>photoreal</appearance>
				<tags>
					<tag>wood</tag>
					<tag>veranda</tag>
				</tags>
			</classification>
		</asset>
	</assets>
</scene>
```

Explanation:
The **<assets>** group should not contain instances of an object, but only one **<asset>** description per unique asset. The provided **<tags>** are only meant to be exemplary, further, automatically derived tags shall be added.

5.3.2 Love and 50 Megaton

"Love and 50 Megaton" is another Filmakademie Production that was realized in 2019. It was distinguished with a 2020 VES nomination in the category best Visual Effects in a Student Project. Already during the production, the VPET tools were tested. Now the assets are released publicly[4] and being reused in experimental production in SAUCE. The entire scene has been automatically analysed by the machine learning based semantic descriptors.

5.3.2.1 Semantic Description

bench park

line road

[4] https://animationsinstitut.de/en/research-development/projects/sauce/semantic-animation

tram

rail straight

Above 4 assets can be seen that have been extracted from the publicly released 50 megatons assets. The entire scene has been automatically analysed by the machine learning based semantic descriptors. Besides other labels, the descriptive labels shown above were generated by the classifier.

As a note, the semantic metadata needs sufficient training examples to learn the classification and generalize to upcoming additional scenes.

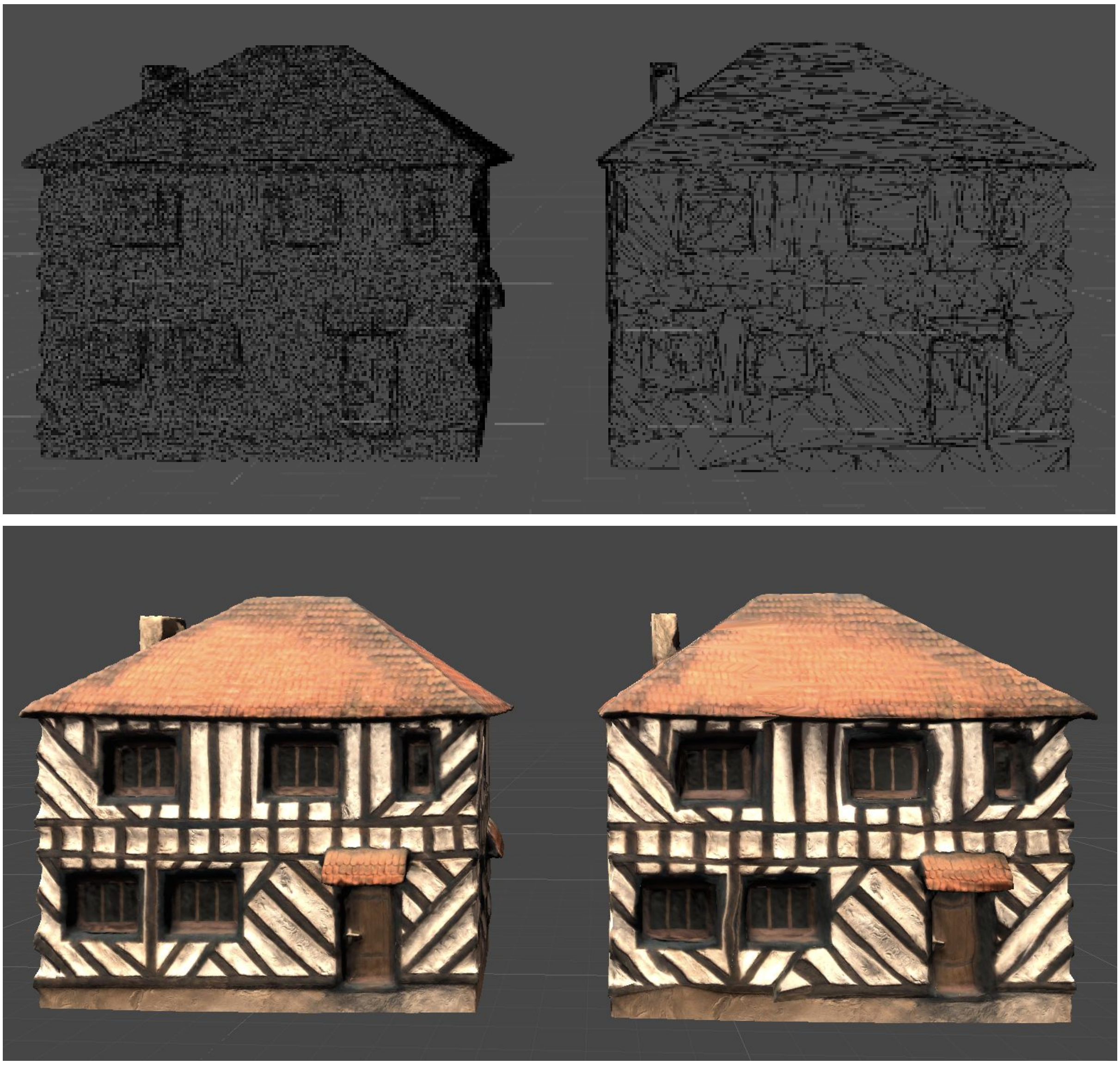
These labels transform the 3D scene into a smart asset providing not only the pure geometry data, but also additional metadata, valuable in a virtual production scenario. This also makes the 3D scene easily reusable. Based on the labels, the scene could be set up for a usage in the Virtual Production Editing Tools (VPET) developed by FA. Through the labels, movable objects like cars, benches etc. can be separated from static parts like houses. Movable objects are then marked as editable in the VPET system, enabling a user to select and modify it in realtime on set with the VPET tablet clients. In addition the labels can be used to provide scene understanding to e.g. the animation solving engine in order to generate scene aware animations.

6 Asset transformation for virtual productions

In this chapter an exemplary use case of asset transformations within virtual productions is discussed. An often underestimated work overhead for virtual productions is the asset preparation. Assets generally need to be available in high detail, to be able to judge visual quality, and in a reduced detail version used for realtime purposes e.g. on the VPET tablet tools.

The Search and Transformation framework developed in WP4 could integrate an automatic quality reduction transformer.

6.1 Exemplary reduction



Automatic asset complexity reduction from high detail (left) to lower detail (right)

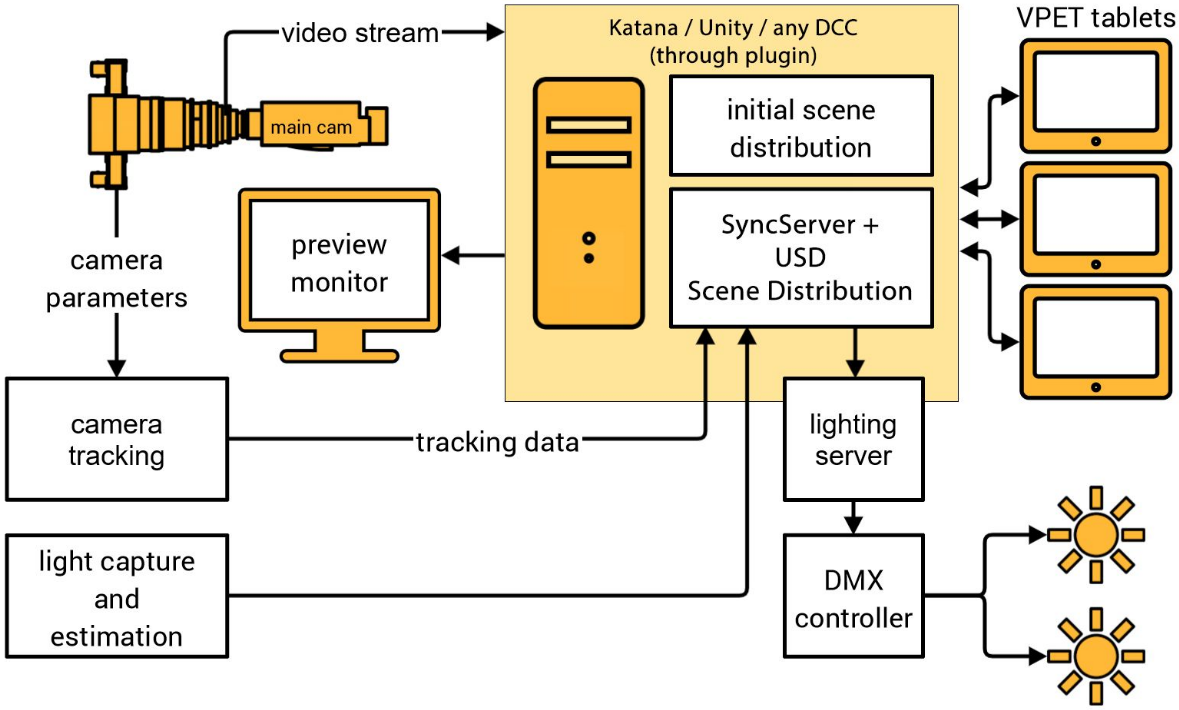
The above reduction reduces the mesh by a factor of 5, resulting in roughly 10000 triangles (right side) instead of 50000 (left side). Only minor visual compromises have to be accepted.

Since the VPET tablet client can be considered as a remote control for the 3D scene, visual quality on these mobile devices is not the key target. VPET allows the usage of different versions of the same asset on the client and the host. This means that a user can interact with the lower quality asset on the tablet, but all updates are applied to the high quality version on the scene host, which thereby is capable of providing a high quality preview of the final look.

This approach requires two versions of the same asset. Creating them manually is time consuming. Numerous algorithms are able to reduce the complexity of an asset automatically while maintaining shape and textures. Also the above sample has been generated with such a tool. The Search and Transformation framework developed in WP4 would provide the possibility to integrate such a transformer as plugin. This would simplify the preparation of a Virtual Production, making it more attractive for smaller studios or individuals.

7 Integrating USD in Virtual Productions

Up to now, a 3D scene had to be provided by a digital content creation (DCC) application to the VPET clients [3]. To realize this, plugins have been written e.g. for the Unity Engine as well as Foundry's Katana to communicate with VPET. Loading and preparing the scene in the DCC application adds an additional step to the preparation phase of a virtual production that can be avoided in some cases.



Schematic overview of VPET

In order to offer the possibility to provide 3D assets into the VPET ecosystem without the need to operate a dedicated DCC application, an independent scene server has been developed. Similar to the already existing solutions as plugins within an DCC-Application, the server will provide the required scene to the VPET clients on demand. To maintain already established data formats and at the same time keep assets in their scene specific relation the Universal Scene Description (USD) has been chosen as a suitable exchange and backend format. This makes it convenient to utilize 3D content from a production pipeline, directly into VPET.

7.1 Background

USD is an open-source 3D scene description and file format developed by Pixar. It aims to address all aspects related to 3D scene handling. Its broad and fast adoption by all major 3D software providers, makes it a good, modern choice as scene exchange format. Adding USD to the capabilities of the transformation framework therefore appears more than reasonable.

USD follows a different concept compared to older formats like alembic, FBX, OBJ etc. All these formats strictly define how a 3D asset has to be stored and which features can be supported. Furthermore, these formats were never developed for storing entire scenes, which is why the necessary structures and functionality for this are either not available or only sparsely available. In contrast, USD provides a generalized frame for exchanging 3D sceneries. All USD assets are arranged in a tree structure. Thereby multiple USD files can reference each other. On top of a base object called 'Prim' further properties and metadata can be added to the object that describe geometry, materials rigs etc. A set of standard geometry and material descriptions only serves as the fundamental. Studios, application developers and others can extend USD with the properties and data structure they use in their pipeline. The concept of a non destructive edits and the possibility of layering these modifications fits nicely into a virtual production workflow, where one of the main goals is to be able to make spontaneous decisions on set and record them.

USD also fits well into the concepts of smart assets. Since USD can be extended with additional metadata, also data describing the scene and objects semantically could be added in future. This data can then be read by e.g. the animation engine to automatically generate semantic and scene aware animations. Nodes in the USD scene tree are also called layers in USD syntax. These layers can hold references to different versions of the same asset. In the context of virtual productions, higher and lower level-of-detail (LOD) versions of the same object could be stored in the same scene tree, making USD a holistic container for all necessary and relevant aspects for a virtual production. The search and transformation framework could potentially also automatically generate USD files being usable in the developed stand alone USD scene server. These could include different LODs and semantic descriptions obtained from reused, existing assets enriched by the search and transformation framework.

7.2 Implementation and usage

The USD scene server for VPET is a standalone console application based on the C++ API of USD. The communication of the scene server with the VPET clients is realized through the network library zeroMQ, which is also used for communicating scene edits between the VPET clients. Further information on the architecture of VPET can be found in deliverable D6.2.

Based on a given parameter, containing the path and the filename to the USD resource, the program will load all dependent assets hierarchically, parse them and convert them into several VPET specific packages. These packages include the scene graph, storing all scene hierarchy relevant data, the object geometries, the materials and textures and animated character specific data. Afterwards a zeroMQ based server is started, waiting for clients requesting the scene data packages.

8 Conclusion

This deliverable reported on various aspects being developed in SAUCE WP6 by Filmakademie in regards to an extended Virtual Production Toolkit. Semantic descriptors can be used to enrich existing assets. The thereby generated smart assets are capable of providing semantic information in the form of metadata containing labels generated by the semantic descriptor. This semantic description can then provide important scene information for the animation engine and the setup of the VPET system in general. The Search and Transformation framework enriched with an asset transformation plugin further simplifies this setup.

Adding USD digest to the capabilities of the Sync Server Component of the VPET System extends its use dramatically. As a widely adopted format it further extends the capacities of potential USD extensions to the Search & Transformation Framework.

Combining all of these aspects, a consistent story for utilizing SAUCE technology in virtual productions is presented. This greatly showcases the SAUCE efforts in this field targeting e.g. the creative user group. The developed aspects will be evaluated further in upcoming experimental productions as reported in D8.2.

9 References

[1] Javi Agenjo, Alun Evans, and Josep Blat:
Webglstudio: a pipeline for webgl scene creation. In Proceedings of the 18th International Conference on 3D Web Technology, pages 79–82. ACM, 2013.

[2] SAUCE: Motion capturing data, 2019.URL https://animationsinstitut.de/en/phs-ml

[3] S. Spielmann, V. Helzle, A. Schuster, J. Trottnow, K. Götz, and P. Rohr:
Vpet: Virtual production editing tools. In ACM SIGGRAPH 2018 Emerging Technologies, SIGGRAPH '18, New York,NY, USA, 2018.

[4] S. Spielmann, J. Trottnow, D. Moreno, J. Agenjo:
Approaching real-time Character Animation in Virtual Productions. In CVMP 2019 Short Papers, London, UK 2019

10 Trademarks and Copyrights

Unity - Game Engine developed by Unity Technologies
Katana - Software developed by The Foundry

11 Acronyms and abbreviations

VPET - Virtual Production Editing Tools (Open Source by Filmakademie)
USD - Universal Scene Description (Open Source by Pixar)
DCC - Digital Content Creation
AR - Augmented Reality