D6.2 - Fast Renderer Demonstration

Grant Agreement nr	856879
Project acronym	PRESENT
Project start date (duration)	September 1st 2019 (36 months)
Document due:	August 31st 2021
Actual delivery date	August 31st 2021
Leader	Framestore
Reply to	Richard.Ollosson@framestore.com
Document status	Submission Version

Project funded by H2020 from the European Commission

Project ref. no.	856879
Project acronym	PRESENT
Project full title	**P**hotoreal **RE**altime **S**entient **ENT**ity
Document name	Fast Renderer Demonstration
Security (distribution level)	Public
Contractual date of delivery	31/08/2021
Actual date of delivery	31/08/2021
Deliverable name	D6.2 Fast Renderer Demonstration
Type	Demo
Status & version	Submission Version
Number of pages	19
WP / Task responsible	Framestore
Other contributors	all partners
Author(s)	Theo Jones, Richard Ollosson
EC Project Officer	Ms. Diana MJASCHKOVA-PASCUAL Diana.MJASCHKOVA-PASCUAL@ec.europa.eu
Abstract	Demonstration of the developments undertaken in the areas of realtime rigging and the translation of high fidelity character shading from offline rendering to realtime.
Keywords	digital human, real-time, Unreal Engine
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 14-06-2021	Document created by Theo Jones
1.1 09-08-2021	Version for internal peer review
1.2 27/08/2021	Submission version

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

The work presented in this deliverable demonstrates the exciting developments undertaken in the areas of realtime rigging and the translation of high fidelity character shading from offline rendering to realtime.

As part of WP6 Framestore has undertaken some groundbreaking research aimed at utilising machine learning (ML) to improve the computational efficiency of high-end character rigging. These developments have allowed us to take a full resolution Film VFX rig across into Engine and run it at interactive framerates without any discernible loss in quality. Although further improvements are possible, and are currently being worked on, the results so far have already shown the potential for this technology to transform the rigging and animation pipeline for realtime applications and allow for interactions with high-end characters in Engine that were not before feasible. This technique alone has moved realtime rig performance many years into the future.

When it comes to the realtime shading models utilized for high fidelity interactive characters major companies such as NVidia and Epic Games are focussing huge amounts of resources in this area and change is very rapid. Therefore rather than focus on developing new shading models in this rapidly evolving area we have instead focussed on the challenge of translating the high end offline rendered character shading work, for which Framestore is so highly regarded, faithfully across to the current state of the art realtime renderer available in Unreal Engine.

When the machine learning rigging innovations are combined with the novel ways of translating high end film character shading into Engine we feel we have entirely removed the requirement for 'super-computer' rendering power from the Present production use cases. The removal of this reliance on a custom built 'super computer' is a huge advantage for the project and for the consortium partners as it brings this high-end visual quality for characters within the reach of practical production environments and vastly improves the business case.

This document presents the methodology behind the machine learning rigging approach for both body and facial rigs as well as demonstrating the results of the improvements made in in-Engine shader translation from offline render models. Finally the current performance benchmarks are presented along with next steps for further improvements. In the results section the numerical case is made for these technological advancements negating the need for a custom 'super-computer' build.

2. INTRODUCTION

The aim with PRESENT is to create the best possible visual representation of a digital human rendered in realtime.

The challenge of recreating the human body and face with a degree of visual accuracy so as to make it indistinguishable from the real thing is the greatest challenge in visual effects. However this is a challenge that Framestore is frequently asked to rise to for major Hollywood movies. So the challenge on PRESENT became, how to take all the expertise and technology that allows this challenge to be overcome in the offline rendered world of movies and bring it into the computationally constrained world of realtime rendering.

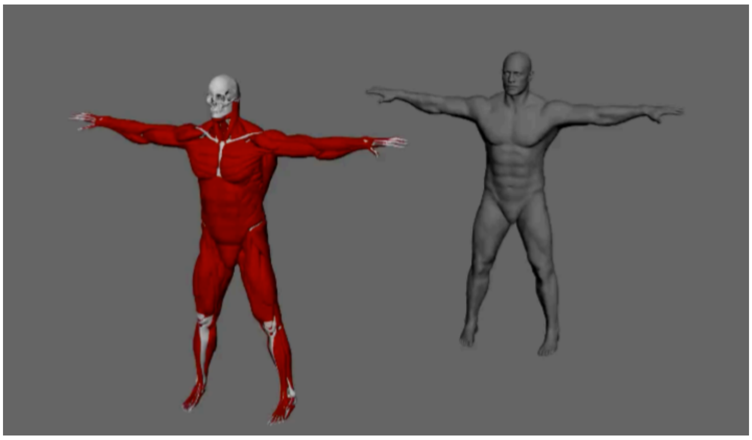
When the PRESENT project was first proposed the solution to this problem was seen as increased compute power, ie daisy-chain together multiple GPU's and throw as much computing power as possible at the problem. Although this is still an option it carries some distinct disadvantages i) it adds significant cost to the setup, ii) it adds significant technological complexity iii) it raises the barrier of entry for others trying to use the system to the point that it takes it out of reach of most common production scenarios.

For these reasons we have focussed all our resources towards improving the rigging and rendering efficiency of the high-res agent to the point that it can be run on the highest current spec consumer workstation without compromising quality.

This deliverable therefore focuses on the technological innovations that have allowed us to get to this point and make us confident of deprecating the need for a custom built 'super-computer'.

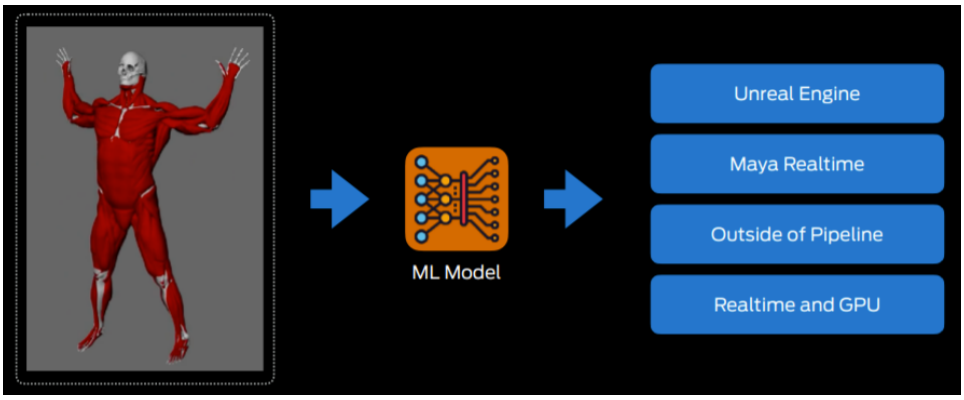
3. MACHINE LEARNING RIGGING OPTIMISATION

For offline rendered content significant compute power is spent to generate the complex deformation required to faithfully recreate the subtle behaviour of human skin. At Framestore this work is done using 'render rigs', which utilise years of artist experience and training with a stack of custom software built on top of Maya to model skeleton, muscle, fat and skin. These rigs often consist of hundreds of thousands of nodes comprising both basic Maya functionality as well as custom Framestore simulation tools. It is not uncommon for these rigs to take up to 20 seconds per frame to evaluate.

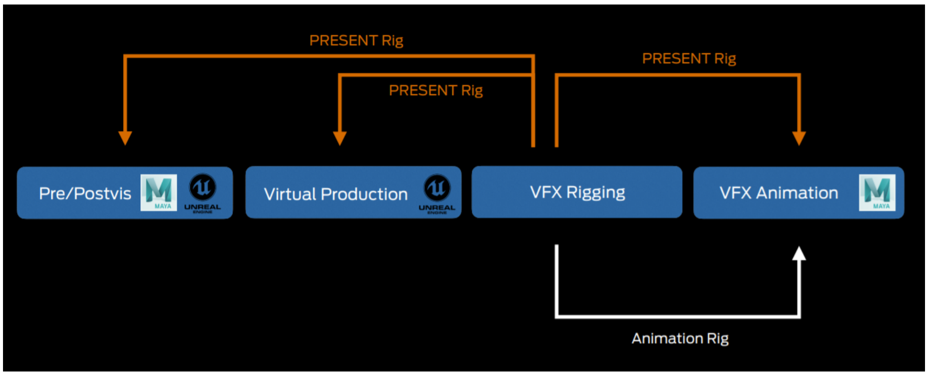


It should be clear from this description that porting these rigs into a realtime piece of software such as Unreal Engine is prohibitive both from a software complexity point of view as well as a compute budget standpoint.

However machine learning offers a potential solution to both these obstacles. By encoding these complex deformations into a machine learning model there is the potential to generate a highly portable, black-box solution that would allow all the hundreds of thousands of custom nodes required for the original rig to be replaced by a single neural network model.

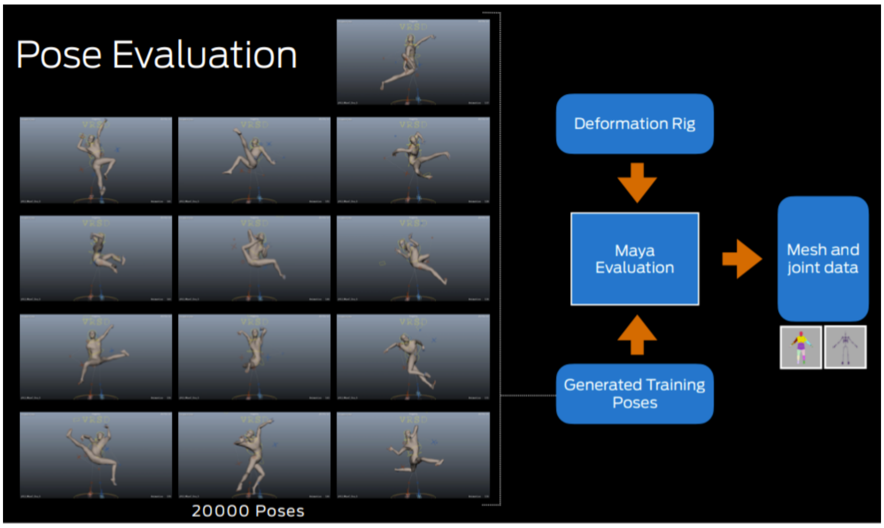


This allows for the rig complexity to be essentially decoupled from the final evaluation, providing for vast improvements in efficiency and portability.

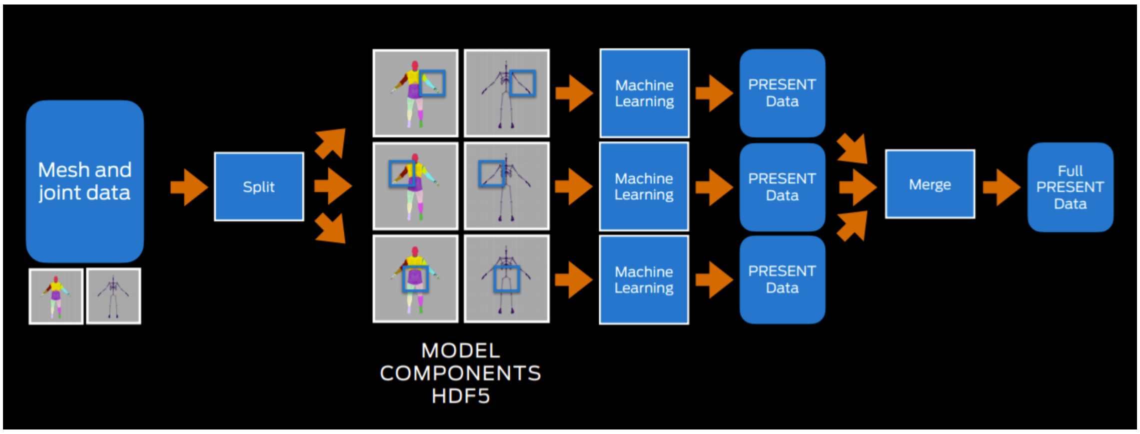


3.1 Model Training

Machine learning models are only as good as the training data they are provided. Therefore the first challenge in any use of machine learning is sourcing good training data. In the case of our rigging challenge that involves generating 20,000 random, yet anatomically plausible rig poses. These are fed into the offline render-rig and the joint rotations and final mesh generated by this rig is written out for every pose.



These meshes are then divided into smaller sections roughly corresponding to independently articulated body parts. Each of these sections, along with the joint rotations that influence them, are then used to train individual neural network models. Finally these are merged into a single entity, which is then optimized further and saved out for runtime evaluation.

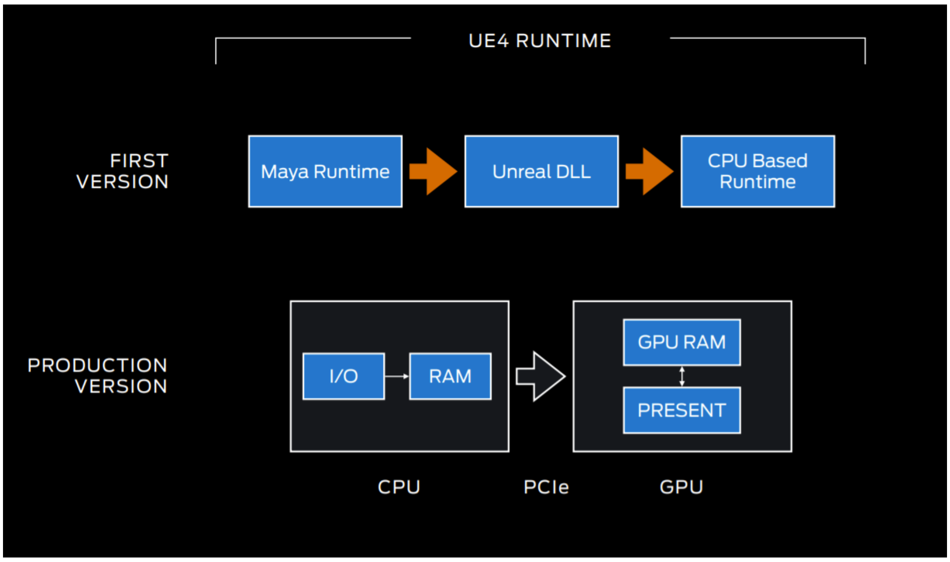


3.2 Run Time Evaluation

An initial proof-of-concept, multi-threaded, CPU implementation of the model evaluation was done as a Maya plugin to gauge results. This was also implemented so that the animation teams could benefit from the faster evaluation times and visual fidelity that this technique allows.

For an initial Unreal Engine implementation this code could then very quickly be ported over using a DLL to retain the CPU multi-threading. This allowed us to very quickly see the same results in Unreal but this time driven by native Unreal skeletal mesh joints.

The work since then has been in moving this CPU implementation onto the GPU.

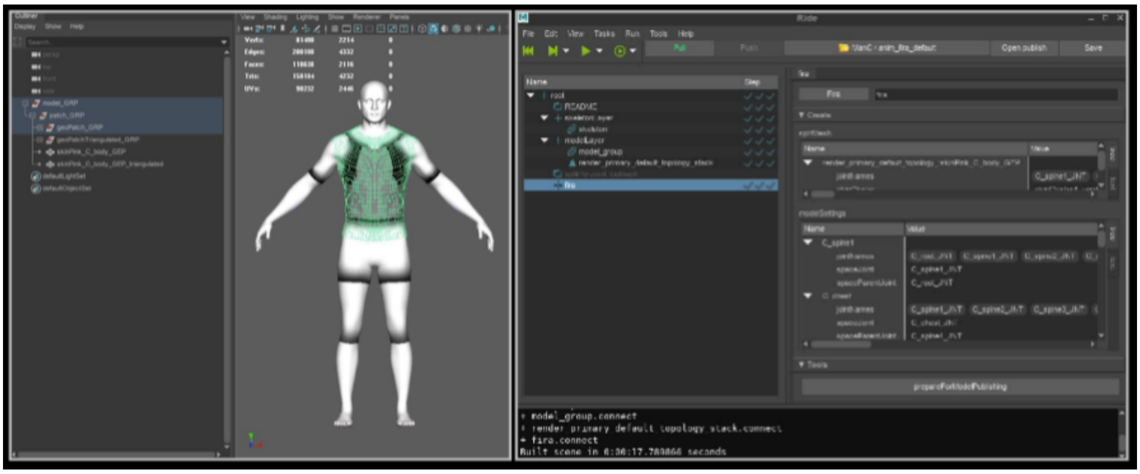


The motivation for this has been to take advantage of the much greater memory bandwidth available on the GPU. Since most of the computation for the machine learning evaluation is memory bound this has allowed for much greater framerates and also opens up the possibility of having multiple hi-res characters running simultaneously in the scene before resource limits are hit.

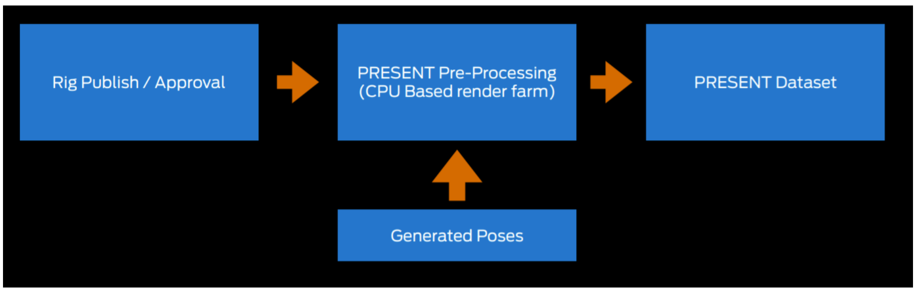
3.3 Pipeline

Since the original POC of the machine learning rig acceleration was developed a significant amount of work has gone into transforming this into a production ready tool and incorporating it into the wider rigging and animation pipeline. This has resulted in a robust and production ready set of tools that can be deployed at scale with the intensive computation required for training distributed across many machines.

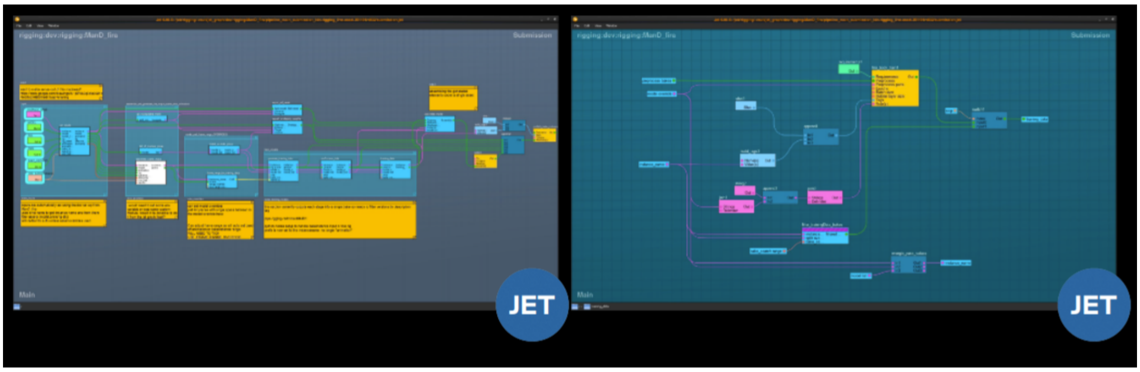
Rigging tools have been developed to allow riggers to control the division of the mesh into the sections required of the ML models. This creative control opens up the possibility for this technique to be expanded to characters of all types.



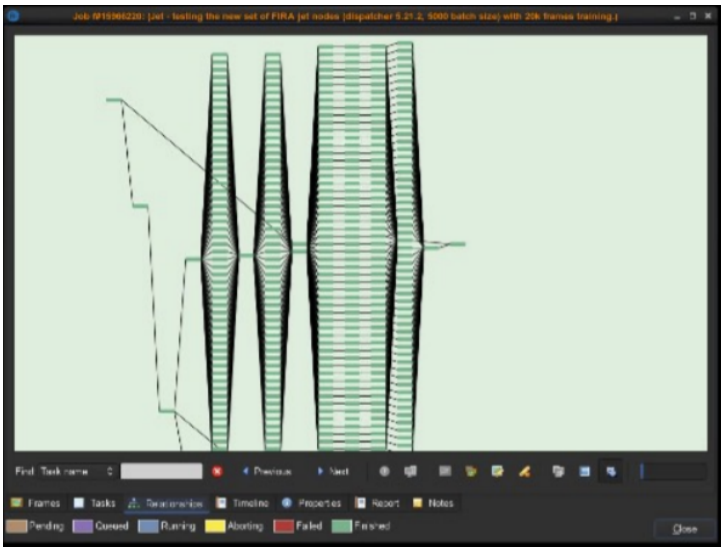
An automated process has also been put in place to generate the poses and process the learning model automatically each time a new rig version is published into the pipeline. This automation is critical to utilising these tools within a large scale production environment.



Custom computation graphs can be setup per rig using our in-house node based automation tool Jet. This makes it easy to customise the setup for individual rigs.



This tool also allows expensive computational tasks, such as the training pose generation, to be distributed on our render farm, massively parallelising the computation and therefore reducing the real world time for a new ML rig to be published to less than an hour.

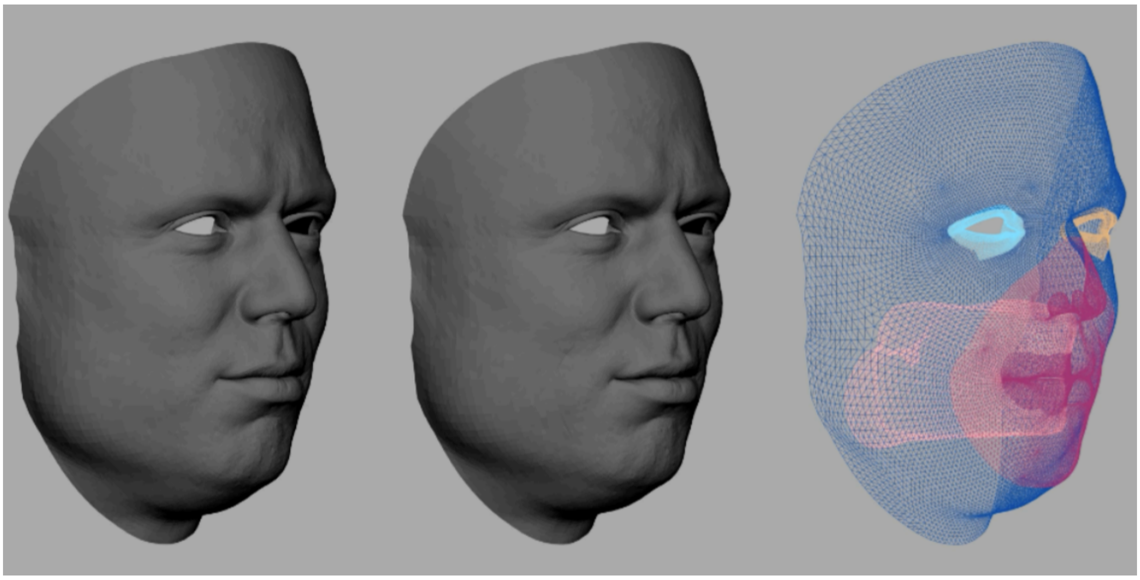


3.4 ML Facial Rigging Optimisation

Facial rigging presents a slightly different, and in certain aspects more complex, challenge than that faced in solving the body rig. At Framestore the facial rigs are predominantly blendshape based with subtle eye, mouth and tweaker deformers layered on top. The relationship that must therefore be learnt by the neural network model is not that of joint rotations to final mesh, but rather that of arbitrary face controls to final mesh.

The sensitivity to error is also much greater for the face. An error of a millimeter or two on the body can be tolerated but on the face this can completely change the characters expression.

For these reasons the facial mesh is divided into 5 distinct areas: i) lips, ii) inner mouth, iii) right eye, iv) left eye and v) everything else. Different neural network approaches are then used for the different areas depending upon the complexity and fidelity required of the deformation. This approach allows for maximum optimisation without loss in fidelity in critical areas.



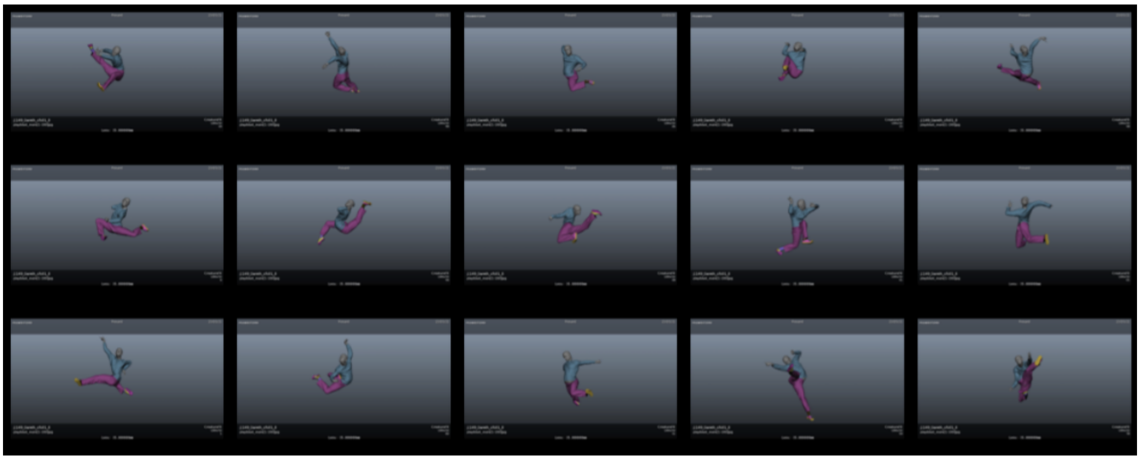
The facial machine learning work is in its final stages but has not yet gone through the translation from CPU prototype to final GPU implementation. Given the success of this process for the body rigging solution and the similarity in the challenge we do not anticipate any issues and expect this work to be completed in good time for the D6.3 deliverable at M30.

3.5 ML Cloth Simulation Optimisation

The challenge of representing cloth deformation as a neural network model is very similar to that faced for the body rig with a few key caveats. Cloth simulation has a temporal component that is not present in the body rig and which the current neural network architecture is incapable of learning.

However for tight fitting clothing much of the fine detail wrinkling behaviour is not dependent on time and can purely be derived from the characters pose. We have therefore selected relatively tight fitting clothing for the agent thereby limiting the challenge to one that very closely resembles that of the body rig, ie joint rotations as input and final cloth mesh as output. For this reason, although not as far advanced as the body rigging solution, we are very confident of finding a swift solution to this challenge based on the same approach as that taken for the body rig.

Early tests have reinforced this confidence. Clothing simulation rigs have been setup and random poses generated as with the body rig.



These poses are currently being used to train a network architecture similar in topology to that used for the body rig and development is advancing steadily.

4. SHADER TRANSLATION

The shading and rendering architecture that Framestore employs for high end visual effects work on feature films has been developed over many years and it is regarded as a global leader in this field. Framestore's proprietary renderer, Freak, has been used for all film visual effects output for the past 3-5 years and is at the cutting edge of offline render development. For this reason, and to take advantage of the extraordinary talent and experience Framestore has at its disposal in the area, the first approach with the shading and rendering of the high-res agent was to create a best-in-class digital double using this offline rendered pipeline.

The arena of realtime shading and rendering is a very rapidly evolving one that is receiving a great deal of attention and resources from companies such as NVidia and Epic Games, and these advancements are constantly being made available via updates to realtime rendering software such as Unreal Engine. For this reason the decision was made early on not to attempt to run parallel and overlapping research with these large companies who are focussed exclusively on realtime rendering, but rather to focus the Framestore Present rendering research and development on finding novel ways of translating the best-n-class shading and look development work that Framestore produces using its offline rendered technology faithfully across to the more simplified shading models available in the latest cutting edge realtime rendering software. It was felt that this approach would give the Present project the best chance of taking advantage of the latest developments in both the offline and realtime rendering domains.

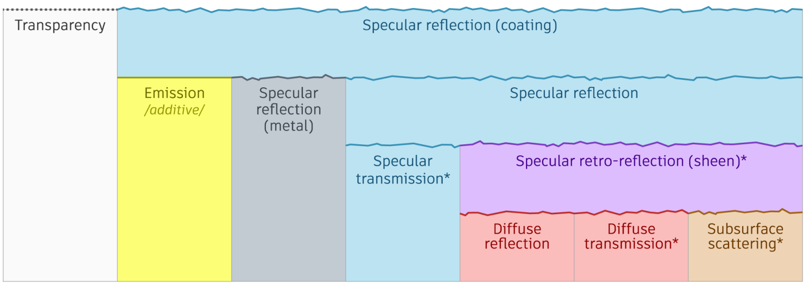
In the area of offline rendering for high end film and television visual effects render times of 100+ hours per frame are not uncommon. For Present the target is an interactive 30 frames per second, which translates to a render budget of 33 milliseconds per frame.

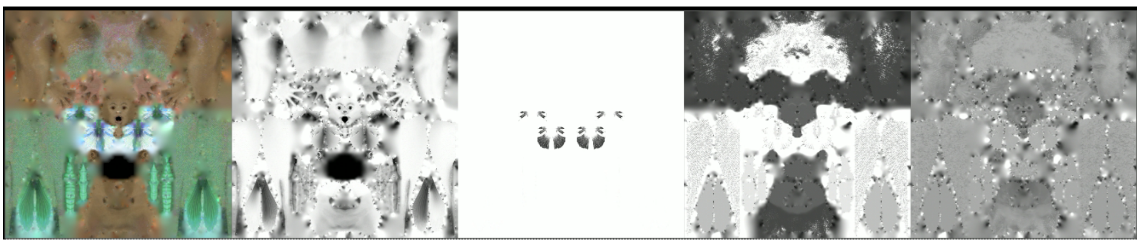
Due to these vastly different compute targets the complexity of shading models available in offline rendering software are more varied and exceedingly more complex than in realtime rendering. Texture inputs to offline rendered shaders can number in the tens or often hundreds with each texture consisting of tens of individual 4k or 8k frames and a vast array of control parameters. In realtime rendering the shading models are generally limited to some derivation of the Disney PBR shading model. In the case of Unreal Engine, our target realtime software, the default shading parameters are listed below:

- Base Color
- Metallic
- Specular
- Roughness
- Emissive Color
- Opacity
- Normal
- World Position Offset
- Subsurface Color
- Ambient Occlusion
- Pixel Depth Offset

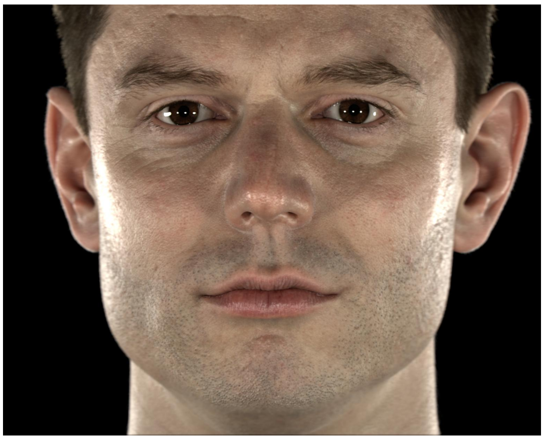
The development challenge being faced on Present therefore became how to translate or 'bake-down' the extreme texture detail and shader graph complexity of the look development work available in the offline shader models into more compressed input textures that could be used to drive the simplified shading models available in the latest realtime rendering software.

To overcome this challenge a novel projection based rendering pipeline was developed along with a mathematical model for parsing shader graphs of arbitrary complexity created in Freak and compressing these down into a set of weighted maps that correspond to the input parameters of the classic realtime PBR shading model.

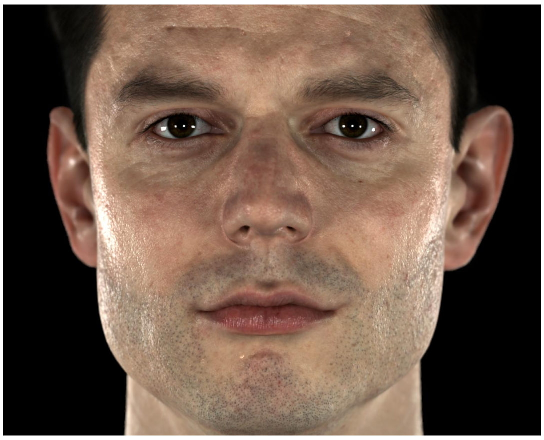




This approach has led to a great advance in our ability to faithfully recreate the meticulous texturing and look development work carried out in the offline rendered realm in Unreal Engine.

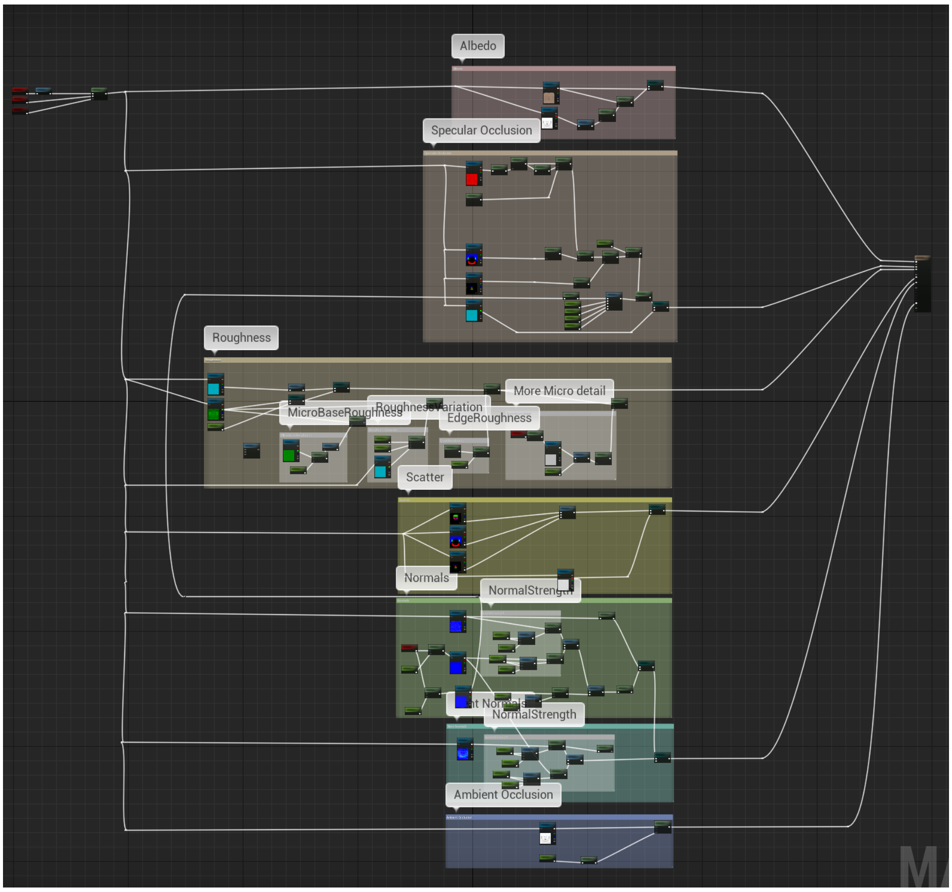


Offline Rendered Asset



Game Engine Rendered Asset

Further work has been carried out within Unreal Engine to subtly modify these textures in order to fully maximize the limited realtime render budget but this process is greatly aided by the strong foundational inputs coming from the novel offline translation process. Below is an example of one of the shading networks within Unreal Engine that takes advantage of these 'baked-down' offline rendered inputs.

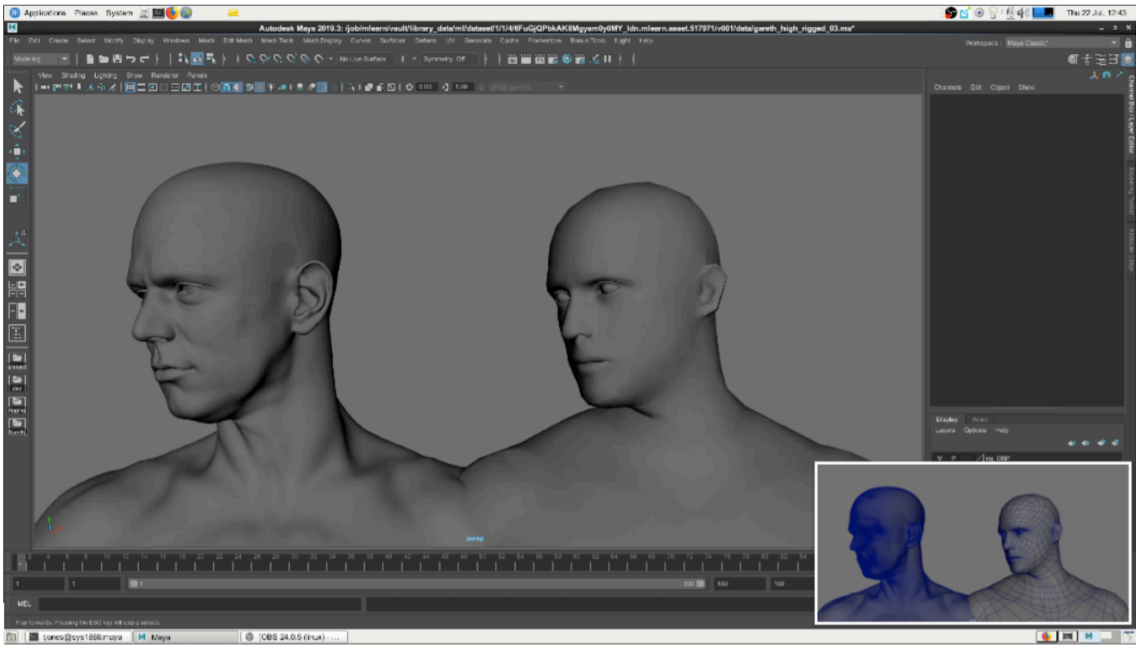


The final result of this process is an artistically controllable shader setup within the realtime renderer that as closely as possible reproduces the look of the original offline rendered asset.

5. RESULTS

The results for the machine learning approach to rig deformation have exceeded expectations and have the potential to completely transform animation workflows and advance realtime rigging capabilities forward many years into the future. For our final render rigs we are routinely seeing upwards of a 5000 fold reduction in evaluation times. This is enabling rig resolutions and deformation complexities that were previously reserved exclusively for offline baked simulation processes to be made available to animators within interactive viewports and for the first time made viable for interactive game engines such as Unreal.

The image below shows the difference in quality in terms of resolution and deformation that is now available to animators in the Maya viewport thanks to the machine learning developments carried out as part of Present WP6.



5.1 Evaluation Times

When it comes to the hi-res agent in the Present use cases a framerate of 30fps is targeted to allow for seamless user interaction. This translates to a render budget of 33 milliseconds per frame.

For the ML accelerated body rig plugin that has been developed for Unreal Engine running in the hi-res agent body the current evaluation time is 2ms on a RTX 3090 graphics card. Even with the additional overhead required for the skin, eye and hair shading this is still well within budget for 30fps.

Maya (CPU Based)

PRESENT per frame compute	2-6 ms
Maya VP2 frame rate with full control rig attached	~30 FPS

Intel i9-7940X
200k triangles mesh

UE4 (GPU Based)

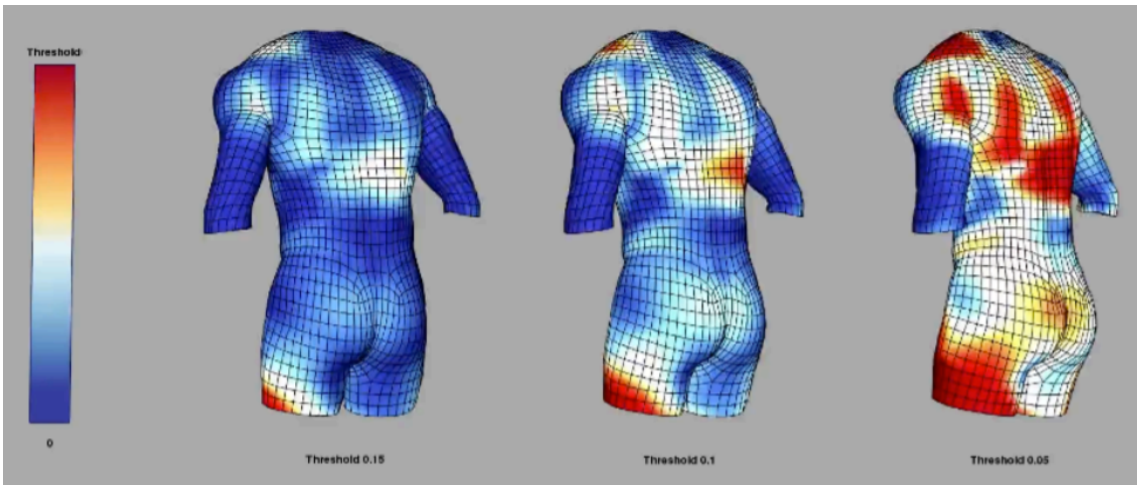
PRESENT per frame compute	2 ms
Frame rate with groom, shaders and animation	60-90 FPS

RTX 3090
200k triangles mesh

The ML accelerated facial rig plugin is still running exclusively on the CPU and is therefore currently requiring 20-30 ms per frame to evaluate. However we fully expect the port of this code over to the GPU, which is underway currently, to yield similar results to that for the body code and therefore the final evaluation time for the face to also sit at around 2ms per frame.

5.2 Error Metrics

It should always be remembered that the neural network model that is utilised in this approach to rig deformation provides only an approximation of the original rig, albeit a very good one. For this reason the error between the ML rig and the original rig must be verified with every new rig release.



This is done by running the original rig and ML rig through an extensive range of motion animation. If any poses within the animation result in an error greater than an acceptable user defined threshold then additional training data can be generated around that pose and used to improve the next iteration of the model. This process can be repeated until the error is reduced to an acceptable level. This ability to keep refining and improving the model is another key advantage of this approach.

6. DEMONSTRATION

6.1 Agent Shading

The first two demonstration videos show the current state of the static hi-res agent running in realtime on a mid-range graphics card. This clearly demonstrates the quality of the look development and shading facilitated by the novel offline translation process.

staticHiResAgentShading_v001.mp4
staticHiResAgentShading_v002.mp4

6.2 Body ML Rig

The next video shows a mocap clip loaded onto the ML body rig running in Unreal Engine. This demonstrates that a full res VFX body rig can be run in Unreal via the machine learning plugin.

mlBodyRigUnreal.mp4

6.3 Body and Face ML Rig

The final video shows a WIP demo of the ML body and facial rigging plugins running together in Unreal Engine. There is still outstanding work to be done to attach the groom to the geo, which is why eyebrows, eyelashes and peach fuzz hair are missing from this demo. There is also further refinement needed in order to seamlessly integrate the head and neck with the body. And a final bug is currently present which prevents the normal maps from rendering correctly which results in an overly smooth complexion.

However the work to fix these issues is trivial in comparison to the work required to get to this point and should be comfortably achieved in time for the final M30 deliverable of the hi-res ML accelerated agent as part of D6.3.

mlBodyAndFaceUnreal.mov

7. CONCLUSIONS

A number of strands of development have been presented in this document that when combined represent a seismic shift in the potential quality of rigging and shading of realtime characters.

The machine learning solution for body rig optimization is fully implemented and deployed internally at scale and into production.

The machine learning solution for facial rigging has been developed to the point of having a solid CPU implementation that demonstrates high fidelity facial performance. This is a key milestone of the project, and given the work already undertaken to optimise the ML body rig on the GPU few obstacles are anticipated in porting the facial rig across using the same approach.

The cloth work, although at a slightly earlier stage, is almost identical to the body work already carried out and therefore swift results are anticipated using the existing body rig software.

Techniques and workflows have been developed and deployed that enable the translation of high end visual effects offline character shading into realtime renders. This approach has been validated on the high res agent and has yielded Unreal Engine renders of a quality that is world leading.

The quality of the high res agent renders produced in Unreal Engine and the performance benchmark of the fully optimised ML body rig in Unreal, of 2 ms on a high end consumer workstation, gives great confidence that the 'super-computer' requirement can feasibly be removed from the deliverable. This has many advantages for the Present consortium and vastly increases the business opportunities for this technology.

Overall this deliverable shows that key project milestones have been achieved and that the potential is being realised for a best in class high resolution digital realtime character.