CGI Report
Stephen Brock - uq20042

Final Cut: `https://youtu.be/ZZthasJOR5w`
Alternate Cut: `https://youtu.be/dsW39nPJ1mw`

Introduction

The prompt for this animated film was 'Olympoids', a world in which the Olympics are now played by robots representing a nation.
The 'Olympiods' in this animation were cancelled during construction due to an unknown bio-hazard, but the autonomous systems continue to play regardless.

The sport chosen was basketball, featuring two robots going head to head. While two characters were not required and it increased the time required to animate drastically, it allowed for some more interesting interaction.

Planning

The first step taken in the planning process was picking a sport.
Diving was first considered, however, a water simulation could end up being too time-consuming to render. Basketball was eventually chosen as it allows for a lot of dynamic animation where two characters can interact with each other and the environment.

Blockout

The limit on this project was 10 seconds of animation, which does not leave time for a lot of action. To visualise this, a very rough storyboard was made (Appendix Figure 20). From this storyboard, it was clear the length of the first plan would not fit within a 10-second window, so some shots were removed.

Confident that the short story would fit into the time frame, the next stage was blocking out the scene. A 'blockout' [1] is a method of using simple primitives and shapes to get a sense of the space and composition of the scene. Starting with a simple arena with a roof that opens, with the characters represented with cuboids (Figure 1). A very simple animation was created only translating these characters. This was important to see if the movement could realistically take place within the time-frame, and some further shots being removed to make more space.

Cameras and simple directional lighting were added to the scene to get a sense of the composition. With an idea of the camera angles, lighting and positions, more refined animation could later use these features for better composition. Setting up basic lighting in the block-out stage influenced the modelling and scale of the arena greatly as well. The roof opening needed to be large enough to allow enough light in to illuminate the scene but not wash it out. The arena needed to be big enough so that the court is visible in light but the exterior of the arena is in darkness. A lot of the composition centres around this spotlight from the roof, so it was important that it is defined early on in the process.

Tech Stack

The decision was made to use some other software in the process other than Maya to take advantage of their individual strengths. Blender [2] is another 3D package similar to Maya, containing tools to model, texture, render and composite 3D models. The modelling process in Maya and Blender are similar, however, modelling can be much more efficient in Blender and exporting geometry between software is not an issue. Blender's modifier system is used extensively in this project for non-destructive modelling processes. After trying a few methods early on, Human-IK animations and constraints could be exported from Maya to Blender via Alembic with UVs enabled. The only downside is that no material data is sent so materials have to be set again, however, this is only a process that has to be done once per export.
Most textures were sourced from publicly available PBR texture libraries [3] [4], however for more complex texturing Substance Painter [5] was used on an education license. Substance Painter allows for texture painting using various baked maps (e.g. normals, ambient occlusion, curvature etc.), and can perform much more complex texturing than Maya or Blender.

Figure 1: Animation block-out.

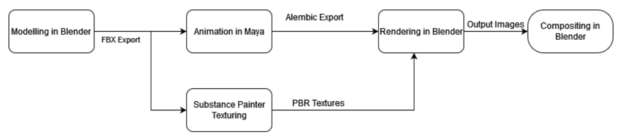

Figure 2: Tech pipeline for robot animation.

Rendering was done in Blender Cycles. Side-by-side comparisons of Cycles with Arnold is equivalent [6], although I find Blender's node system for materials and compositing preferable over Maya's.
The final pipeline (Figure 2) consists of modelling in Blender, exporting to Maya for animation then exporting via Alembic for rendering in Blender. The upside of this pipeline is that I can use Blender for rapid modelling and creating complex materials while taking advantage of Maya's Human-IK animation.
The downside is that once exported, it is difficult to make changes and re-export without possibly breaking any

changes made further down the pipeline. This meant that robot was almost entirely modelled before being animated to avoid further problems. As exporting the animation and reassigning materials was time-consuming, the animation was only exported twice: one early draft to start testing rendering techniques and lighting, and then the final animation.

Reference

Once there was a plan for the scene and its implementation, the final stage of planning was started to gather images to match the aesthetic of the animation. All reference images were used only for inspiration, not in the actual animation. These reference images were arranged in PureRef [7] software as it is created for this purpose.

Before finding any reference for the arena, the blockout designed the arena to have a circular roof opening. A Brutalist style of architecture was chosen as the concrete makes the arena feel like an unfinished bunker, further expanding on the post-apocalyptic theme. One environment I referenced was an abandoned basketball court in Pripyat, near Chernobyl [8], which although didn't match the sci-fi aesthetic was a great inspiration for the atmosphere of the scene. Other references were generic sci-fi or abandoned/overgrown arenas either from real-life or video games (see reference images in Appendix Figure 22).
For the robot itself, reference images for very humanoid robots were chosen. Given the prompt, the Human-IK rig and the amount of human animation reference available, a humanoid robot would be the most viable option. As the robots would be built for basketball, reference robots which seem athletic, agile and not overly built were chosen more frequently. Also established in the blockout was the character's ability to fly, the rocket shoes were inspired by real basketball shoes (see reference images in Appendix Figure 21 or comparison in Modelling Figure 6).

Design

Robot

As a humanoid robot would be more natural to animate, a simple blockout of a human body was created and modelled the robot around those proportions. To make the robot more humanoid, the basic silhouette also had to match those of a human. Therefore, the skeletal system was mirrored via metal rods (Figure 3), muscles and skin by a thin outer plating and joints with motors. Certain standout muscles were given particular attention to make the characters look humanoid. The Sternocleidomastoid neck muscle [9] for example is very prominent and was represented with a hydraulic rod, and the calf muscle silhouette was mimicked with a gas canister for the rocket boots.
Assuming the robot was built specifically for basketball introduced constraints to the design, like requiring the robot to be lightweight. At a point in the modelling process, the character looked too heavy to move believably quickly, so holes were cut in the back of the torso and thigh chassis. This shows the thickness to visually give the impression that the character is made of lightweight material.

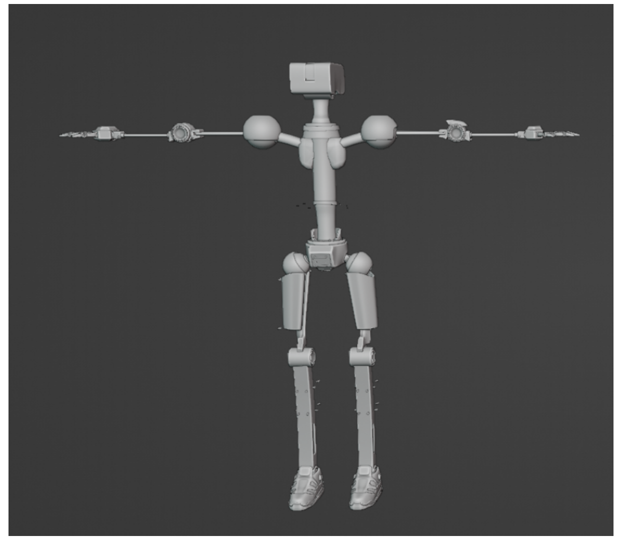

Figure 3: Underlying 'skeleton' of the robot design.

It was also important the character looked realistically built. The chassis is modelled to look like multiple panels welded or bolted together. The motors on the joints have ball bearing details and are connected to give the impression that the robot would be mechanically sound.

Arena

After blocking out, it was clear the high contrast lighting used would hide a lot of the background in darkness, therefore lending to a simpler design as a lot of detail would be lost. Within the confines of the story, it makes sense that an unfinished arena would be lacking in details, either not present or eroded away. Therefore, the most important areas would be the ones in light: the roof, the basketball hoops and the court.

Modelling

Robot

Most of the robot was modelled using a subdivision box-modelling style where sharp edges were made using edge creases (Figure 4). This ended up giving a smooth geometry, lending to the 'built' aesthetic, yet also increased the poly count significantly. As geometry needed to be finalised and all modifiers applied before exporting, this meant animating with such a high-poly model was challenging. Low frame rates made visualising the animation particularly difficult. Edge creases also couldn't be exported, so applying a subdivision after animating was not very applicable.
The modelling process like most box modelling consisted mostly of extruding, insetting and occasional proportional editing. Bevels were usually done in a separate modifier to be non-destructive. Solidify modifiers were also used on the chassis and other objects to give thickness. Spin functions were occasionally used on objects like the joints to duplicate ball bearings in a uniform cylindrical pattern (Figure 5). The spine vertebrae were created mostly with a modifier stack,

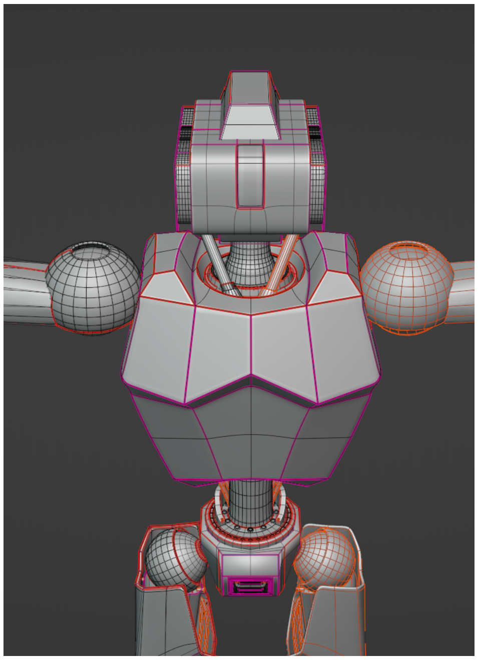

Figure 4: Subdivision creases are shown in purple, and UV seams are in orange.

with an array modifier to stack each vertebra and a curve modifier to transform the geometry into the shape of a spine Bézier curve.
Details like screws were also modelled to make the build of the robot more believable.
The shoes were trace modelled largely from an image, made up of 3 subdivided objects modelled similarly with edge creases, as shown in Figure 6.

Arena

The arena itself is modelled pretty simply, starting with a cylinder and a mirror modifier. More detail was given to the roof, with a cutout for the opening and scaffolding surrounding it. Seating is instanced using Blender's hair particle system with an instance per vertex, to render in an optimised way instead of duplicating a large amount of geometry. This mesh can be seen in the background as a grid overlapping the edges of the arena in Appendix Figure 23. Small details like a corridor, scaffolding and the Olympoid logo (logo shown on the Title Page) surround the arena to break up the uniformity. Scaffolding was done mostly using modifiers for a non-destructive workflow that can be applied for any mesh, shown in Figure 8.
Another complex modelling problem with the stadium was the air vents circling in the roof. To get repeated geometry, one section of the air vent was made at a particular size to match a single segment in a cylinder (Figure 7). Then the spin function was used to create duplicates wrapping around the roof completing the cylinder, and overlapping vertices were merged to create seamless geometry. Further variation was added by randomly extruding and insetting certain faces.

Upon further lighting experimentation, screens and spotlights were added at each end of the arena. The screen structure was made using the same modifier stack to create the scaffolding and Bézier paths with depth for the rope holding them up. The spotlight had a simple cuboid with a square

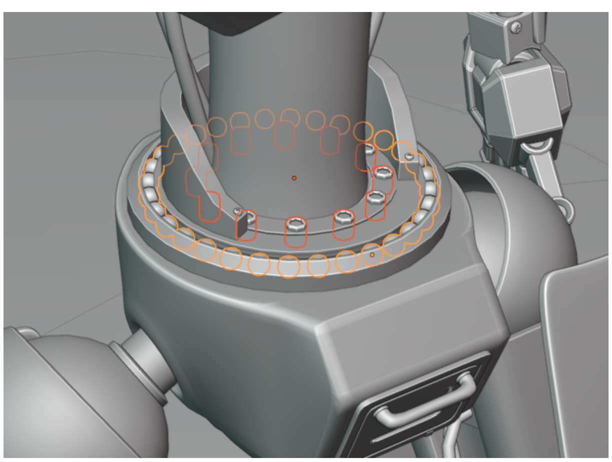

Figure 5: Geometry selected was created with spin function.

Figure 6: Subdivided model of the shoes next to the reference image.

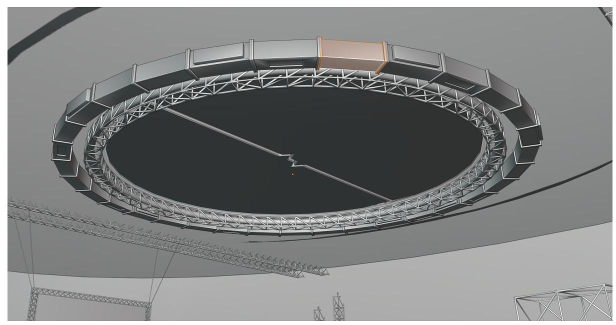

Figure 7: Air vent modelled with spin function, with a single segment selected.

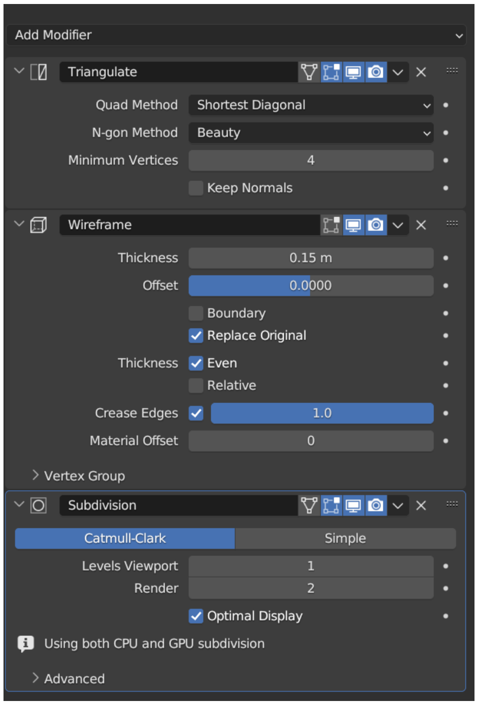

Figure 8: Scaffolding modifiers.

inset to visualise the light source emitter.

Basketball

The basketball was complex to model as it consists of one part but its geometry is very intricate. This could be simplified as just a sphere with a texture applied, however, this lack of detail could be obvious in any close-up shots.
Instead, a low-poly UV-sphere was used with a subdivision surface modifier. To match the flow of the seams, extra edges were created with the knife tool at the polars of the UV-sphere, and the edges corresponding to the seams were moved into the correct positions and selected. These selected edges were then bevelled for thickness and then extruded inwards against their normals for depth. The subdivision surface modifier allowed for a high-poly smooth surface without having to manipulate a large number of edges and faces (Figure 9).

Texturing

Unwrapping

Most of the robot mesh was mirrored via modifier so that symmetric updates were non-destructive. This means that there are no asymmetric differences in geometry. The same UVs were used for each arm or each leg as well to save on texture space, as VRAM could cause future issues depending on how many textures were used. However, for a robot, the lack of symmetry is not as much of an issue. A robot would be built to be symmetrical and would wear largely symmetrically after long use, especially as a robot is unlikely to bias any particular side unequally in the way that a person might have a dominant hand. The torso and

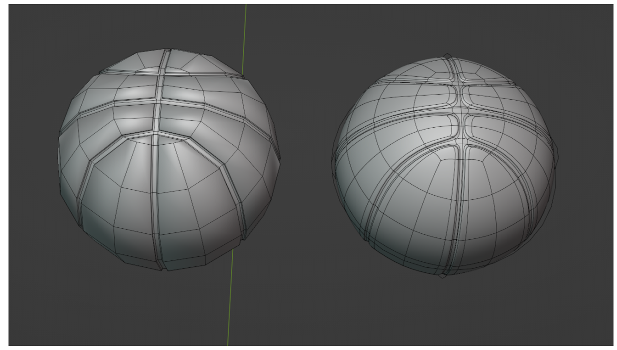

Figure 9: Basketball before and after subdivision.

head however are unwrapped after the mirror modifier, so have asymmetric textures.

Unwrapping was done manually for most objects in the robot, except very small insignificant objects which were unwrapped by the edge angles with Blender's smart UV unwrap [10].
Most of the robot was texture painted in Substance Painter, with materials grouped by limb (Figure 10), excluding only smaller objects like screws and hinges which shared a generic metallic material. That means every object on the arm shared a UV map and every object in the leg shared a UV map. This is preferable to the whole object sharing a singular UV map for a few reasons. It avoids having to work in one huge UV map and splits very large textures into smaller ones, especially as painter applications can start to slow down with larger texture resolutions. It also means that changes to one area of the UV map or texture do not require the whole texture to be recreated or exported again. For objects that were going to be painted, distortion was less of an issue as textures could be tri-planar mapped or painted on directly. For these objects, it was more important that the size of the UV face is proportional to the visibility or 'importance' of a particular section. For example, the torso chassis would take up a larger proportion of the UV space than the underside of the robot.
For objects which were going to be textured procedurally or with seamless textures, they were unwrapped individually prioritising non-distortion and invisible seams.

Robot Texturing

Texturing for the robot was done almost entirely in Substance Painter. As a base for all materials, a 'smart material' was used. This is a preset set of layers for setting up a material quickly, with the layers using a variety of generators to mask textures depending on baked maps. Combinations of ambient occlusion, normal and curvature maps allow create rust and wear realistically. These preset materials are used as they are already set up to be physically correct and have a variety of parameters within the layers to adjust the level of wear or rust. Sometimes multiple smart materials were used and masked, for example, the head and torso had a steel material and a rubber tire material masked for the neck

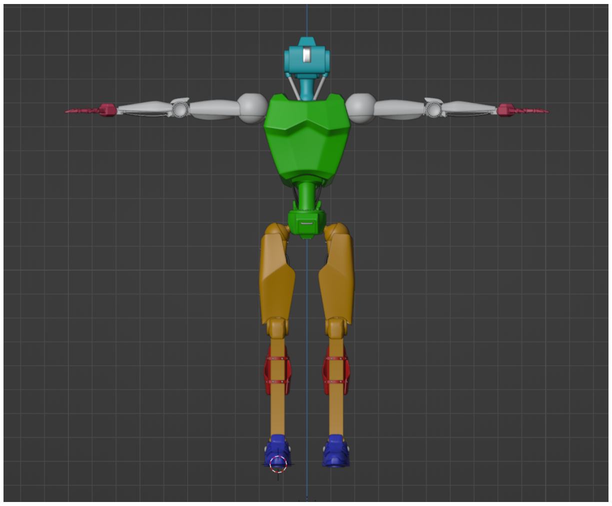

Figure 10: Material groups visualised by colour.

Figure 11: Albedo torso texture made in Substance Painter.

Figure 12: Closeup decals in Substance Painter.

and spine.

Extra layers were added to change the base colour of the metal and further for decals. The decals shown close up in Figure 12 were added manually with a brush with height adjusted to give the impression of an imprint. Normal maps of screws were added at the corners to help the impression that the torso was created from multiple sections of metal screwed or bolted together.
To create variation between the two characters, two separate texture sets were created. The only variation between the two is the base colours and the random seeds used for the procedural textures. As most of the textures are from smart materials this is an easy way to create variation.

Arena Texturing

Most of the texturing was pretty simple, as the arena design is very Brutalist design with a lot of detail hidden by the lighting, therefore, a simple PBR concrete texture sufficed for most of the arena. The chairs and the Olympic rings are simple dielectric plastic. The most complex texturing was the screen and the court. The court was a variety of textures stacked: the concrete textures, the court marking created in Photoshop and some mossy textures to break the uniformity. The screens used a simple score texture created in Photoshop, with a more complex material to pixelate it (See Rendering/Complex Materials).

Animation

There is some animation which has not been mentioned like the poster being squashed in the second shot, the basketball hoop movement reaction and the drone movement. These animations were simple transformation keyframing done in Blender, and although an effort was put into detail like conserving momentum, the methods used vary little to the mentioned animation in the Robot section so no detail will be given.

Robot

As the animation was done in Maya, the robot character was rigged with the Human-IK system with the bones moved to match the proportions of the robot. The parent relations between different parts of the robot had already been set up in Blender, so only the associated object for each bone needed to be set using the parent constraint. The spine and the neck required deformation (Figure 13 and Figure 14), therefore had to be removed from the parent and attached to the rig using skinning. Further more complex rigging were objects like the pistons near the neck, which were using aim constraints to face each other regardless of rotation (Figure 14).

Once rigged, the robots were imported twice into an arena scene, once for each robot. The animation was done via an iterative process. Firstly, only the vague torso and foot positions and rotations were animated, similar to the block-out animation, to make sure the bodies react and move at believable speeds within the 10-second time frame. Next, more complex animation was added with the movement of the feet and hands. Specific animation reference videos [11] were used as well as a variety of generic basketball videos found online by copying important poses and specific times. The final iterations were changes made to make the animation more believable. Smaller changes were made to add startup and follow-through to each animation, either by tweaking the curves or adding extra frames before or after a change, giving the characters more momentum and weight. As geometry was very high-poly, Maya was unable to playback in real-time on the computer used, so after each significant change the animation was rendered and tweaked in areas that looked less believable.

Basketball

The basketball itself required more complexity due to the transitions between being held by either hand and being free to move. This was done via two parent constraints for each hand with the weight blended depending on what hand it's attached to. Once the ball needed to be attached to a hand, the blend weight is increased to 1 on that frame and the offset translation and rotation are updated and keyed to its current position.
As the basketball is also affected by gravity, projectile calculations were used for the initial launch of the ball. This allowed the calculation of the initial velocity for the ball to be at the right height at the right time. The bounces were hand animated as it is too fast to benefit from these calculations.

Arena

The roof had two main animations, the doors and the cogs within. To make animating easier and more versatile, the animation had to be driven by a single value. An empty transform was used for this, with the doors' locations copied via constraint. The cogs animation was driven via a driver expression, with the value of the Z-axis rotation set by the position of the empty multiplied and offset by two parameters (Figure 15). This meant by simply moving one empty transform, the doors would follow and the cogs would rotate offset correctly to fit with each other.

Destruction

Destruction was also animated within Blender, due to the integrated 'Cell Fracture' add-on [12] that makes creating destruction a lot easier. The robot and backboard destruction were animated in similar ways: a duplicate version was made, with the mesh merged into one. Cell fracture was used on the duplicate, with parameters being tweaked depending on how many parts the mesh needs to be split into. Active rigidbodies are added to each cell, with the collision being set to the mesh and the 'animated' toggle on. On the frame before the destruction, the real mesh is switched out for the cells by toggling the visibility and the animated toggle is turned off on the cells. The physics are baked, and the final result shows a mesh getting split into multiple physics-based objects on the correct frame.

Occasionally the physics would make these objects initially move in the incorrect direction, so a force field is activated for a couple of frames to guide the cell fracture in the correct direction of the force applied.
For the robot, a particle system simulates sparks being given off from the collision, with the start frame being the frame of the collision. The sparks are simple low-poly icospheres with an emission shader that falls off with particle lifetime.

Camera

Most of the camera animation was simple keyframing on the translation, rotation, depth-of-field distance and exposure. One more complex animation is from the shot coming through the roof. This was achieved with a track constraint, and the position being animated so that it zooms toward the ball.

Rendering

Complex Materials

Most of the materials were standard PBR, excluding the court and the screen. The court material is shown in Appendix Figure 29, containing procedural damage to the court lining depending on the height of the concrete displacement and subsurface scattering for the moss. The screen material in Appendix Figure 28 contains pixelation, chromatic aberration and displacement all done procedurally. This means the texture could be swapped out for anything independent of the look of the screen.

Optimisations

As volumetrics and interior renders are expensive and require a lot of light bounces, a few methods were used to reduce the cost. Firstly, a 'portal' is placed near the roof. This is an indicator to the renderer that this is where light passes into the scene and where to direct environment rays, allowing the scene lighting to converge faster and reduce noise.

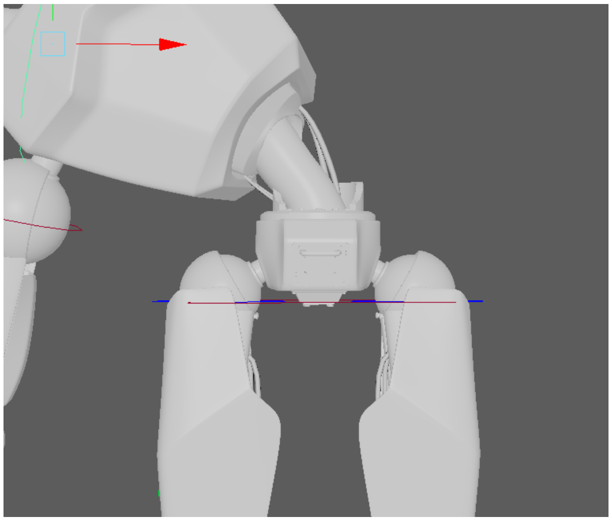

Figure 13: Spine deformation using skin binding.

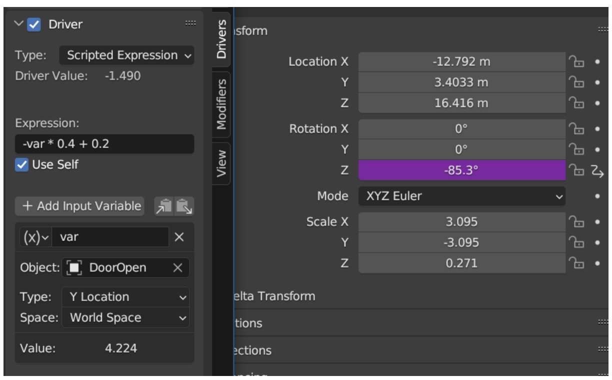

Figure 15: Driver for one of the cog's rotation.

Volumetrics is only applied to a cube surrounding the arena rather than the whole world, as volumetric light bounces outside of the arena do not need to be considered. The 'homogeneous' toggle is enabled on this volumetric object as this uses the assumption of constant density to speed up volumetric rendering and reduce noise.

Small emissive objects also can create a large amount of noise, so some emissive materials like rocket boots are set up so that the material's emission does not contribute to illumination. Emission is done instead by a larger area lamp, replacing two emissive objects with one larger lamp. The material to disable transmission is shown in Appendix Figure 27.

The render was also denoised in the compositing stage, this allows for a smooth image with fewer samples with the cost of some visual artefacts (see Composition/Compositing for steps taken to reduce these visual artefacts). Although denoising has an overhead, for this project around 6 seconds

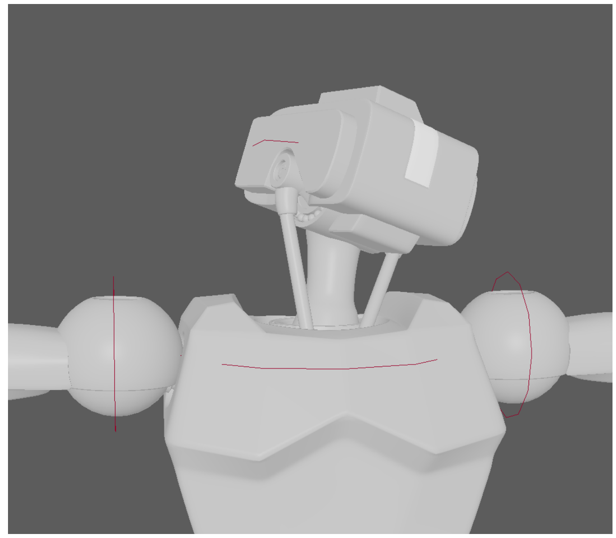

Figure 14: Pistons on the neck using aim constraints to face the correct direction.

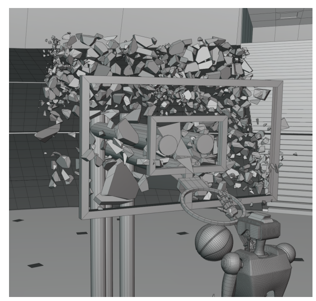

Figure 16: Cell fracture of the glass on the backboard.

Figure 17: Cell fracture of robot's limbs, influenced by a force field.

per frame, it is insignificant compared to the large number of samples needed for a perfectly smooth render.

Composition

Lighting

Lighting was essential to this animation. As a lot of the scene is in darkness, lighting is fundamental to the composition of the scene. Lighting further adds to the dynamism of the scene, contributing to the overall pacing. The first half of the animation is led entirely by the slow entrance of the environment lighting from the roof, with contributions from the large screens adding contrast of colour. Further into the animation, the lighting is dominated by the screens instead, supported by the spotlight and environment lighting. The environment lighting is a simple 'sky texture' node, which is Blender's procedural sky. The intensity was increased to cut through the volumetrics and provide enough light to illuminate the rest of the arena.

The addition of the drone makes the lighting even more dynamic, allowing the area of focus to be lit at all times despite the inconsistent luminosity across the scene. This extra light source also allows for the scene to be naturally lit with a three-point lighting system [13] filling in for either the key or fill light depending on the situation.

Compositing

As the scene is overexposed by default, with bright sunlight and spotlights, exposure is adjusted in compositing. Appendix figure 32, shows the denoising setup and compositing, with some colour correction, bloom, and lens distortion. This helps simulate a real camera's flaws and gives the image a less 'rendered' look.

For denoising, I was using the OpenImageDenoiser, available through a node in the post-processing compositing. This node can take albedo and normal inputs to greater inform the denoiser of the correct geometry and colours, however with volumetrics these inputs confuse the denoiser as the albedo and normals show that there should be geometry there even if it's entirely obscured by volumetric light (see Appendix Figure 33). One solution is to not use albedo or normal inputs and denoise purely based on image data. This works correctly, however pure image denoising is not as effective and results in a lot of visual artefacts, which can become very apparent in an animation. The solution used was to render a separate pass for direct and indirect volumetric lighting data. By subtracting this from the final image, an image without volumetrics is obtained. This image is then denoised correctly with albedo and normals, the volumetrics are denoised without parameters and these results are added together for the final denoised image. See Figure 18 for the breakdown of the denoising procedure. Although the extra pass and two denoises have a performance overhead, the result keeps the benefits of albedo and normal denoising without the volumetrics obscuring the result.

Editing

While the animation itself is only 10 seconds, with 240 frames at 24fps, the renders are composed together so that their frames overlap totalling 19 seconds. As quite a few camera angles are used, without this overlap the changes would be too rapid to grasp. To mask this, camera angles start and end on obscure frames with little identifiable animation. This makes the frame overlap less obvious and keeps the illusion of a constant time frame. To show more of the animation, an unedited clip was rendered focused on one of the characters and uploaded on YouTube [14].

Further Improvements

Given more time, further improvements could be made to this animation. Rather than exporting mesh animation via Alembic, the model would be rigged in Blender and only the bones animated or constrained in Maya. This means that the model data could be independent of Maya, allowing lower poly versions of the character to be used and ideally allowing Maya to run the animation in real-time. This means a lot more iterations could have been done on the animation as it would not require rendering often. With more time, bespoke animation references could be filmed and traced for more accurate movement and either Maya or Blender's physics engine could be tweaked to animate the basketball.

Finally, with more time sound effects and music would have been added. Audio would aid making the story more cohesive, as relying only on the visual for such a fast-pace animation is difficult.
Outside of the scope of this project, this film would benefit greatly from a longer run time. I believe some shots are too rapid to fit within the 10 seconds and therefore could benefit from each shot being active for longer and the animation being padded out for better pacing.

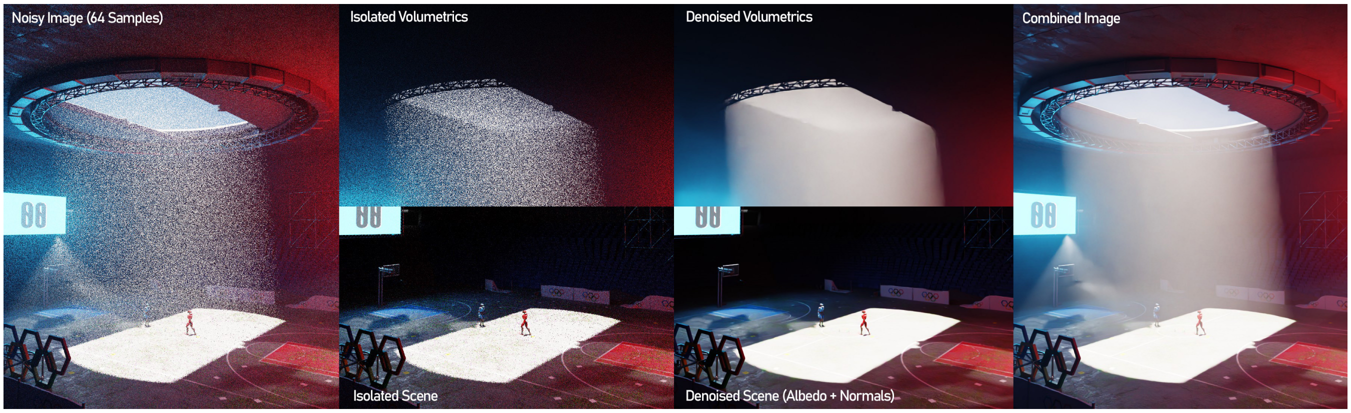

Figure 18: Composition process for denoising.

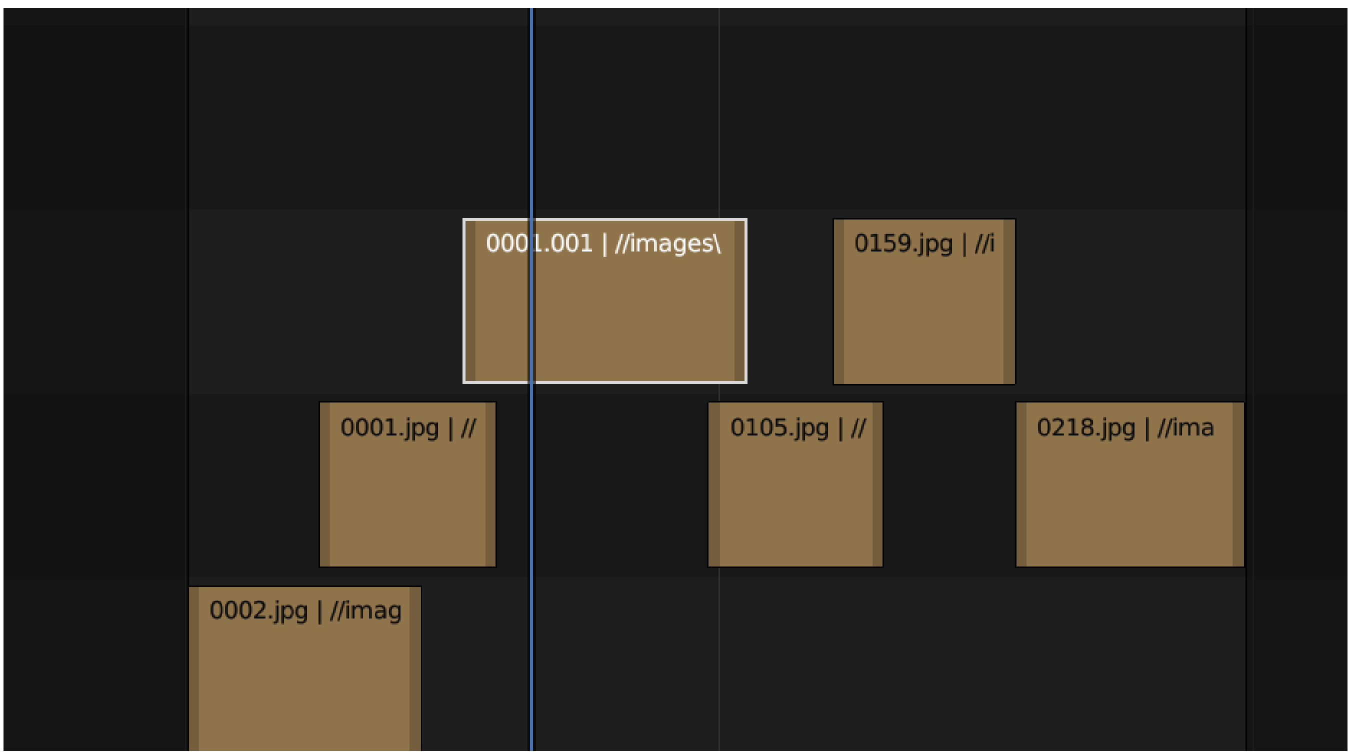

Figure 19: Clips edited together.

References

[1] *Blockout*. [Online]. Available: https://book.leveldesignbook.com/process/blockout.

[2] Blender Institute, *Blender*. [Online]. Available: https://www.blender.org/.

[3] L. Demes, *Ambient cg - public domain resources for physically based rendering*. [Online]. Available: https://ambientcg.com/.

[4] P. Haven, *Poly haven - textures*. [Online]. Available: https://polyhaven.com/textures.

[5] Adobe, *Substance painter*. [Online]. Available: https://www.adobe.com/uk/products/substance3d-painter.html.

[6] el presidente - ArtStation, *Arnold vs cycles (research)*. [Online]. Available: https://www.artstation.com/artwork/rrLBG.

[7] Idyllic Pixel, *Pureref*. [Online]. Available: https://www.pureref.com/.

[8] *Chernobyl tour: Pripyat's abandoned basketball court and swimming pool*. [Online]. Available: https://landofsize.com/chernobyl-tour-pripyats-abandoned-basketball-court-and-swimming-pool/.

[9] Physiopedia, *Sternocleidomastoid*. [Online]. Available: https://www.physio-pedia.com/Sternocleidomastoid.

[10] *Mapping types*. [Online]. Available: https://docs.blender.org/manual/en/2.79/editors/uv_image/uv/editing/unwrapping/mapping_types.html.

[11] *Animation reference videos*. [Online]. Available: https://www.youtube.com/@endlessreference/videos.

[12] *Cell fracture*. [Online]. Available: https://docs.blender.org/manual/en/latest/addons/object/cell_fracture.html.

[13] Wikipedia, *Three point lighting*. [Online]. Available: https://en.wikipedia.org/wiki/Three-point_lighting.

[14] S. Brock, *Single shot for cgi*. [Online]. Available: https://www.youtube.com/watch?v=dsW39nPJ1mw.

Appendix

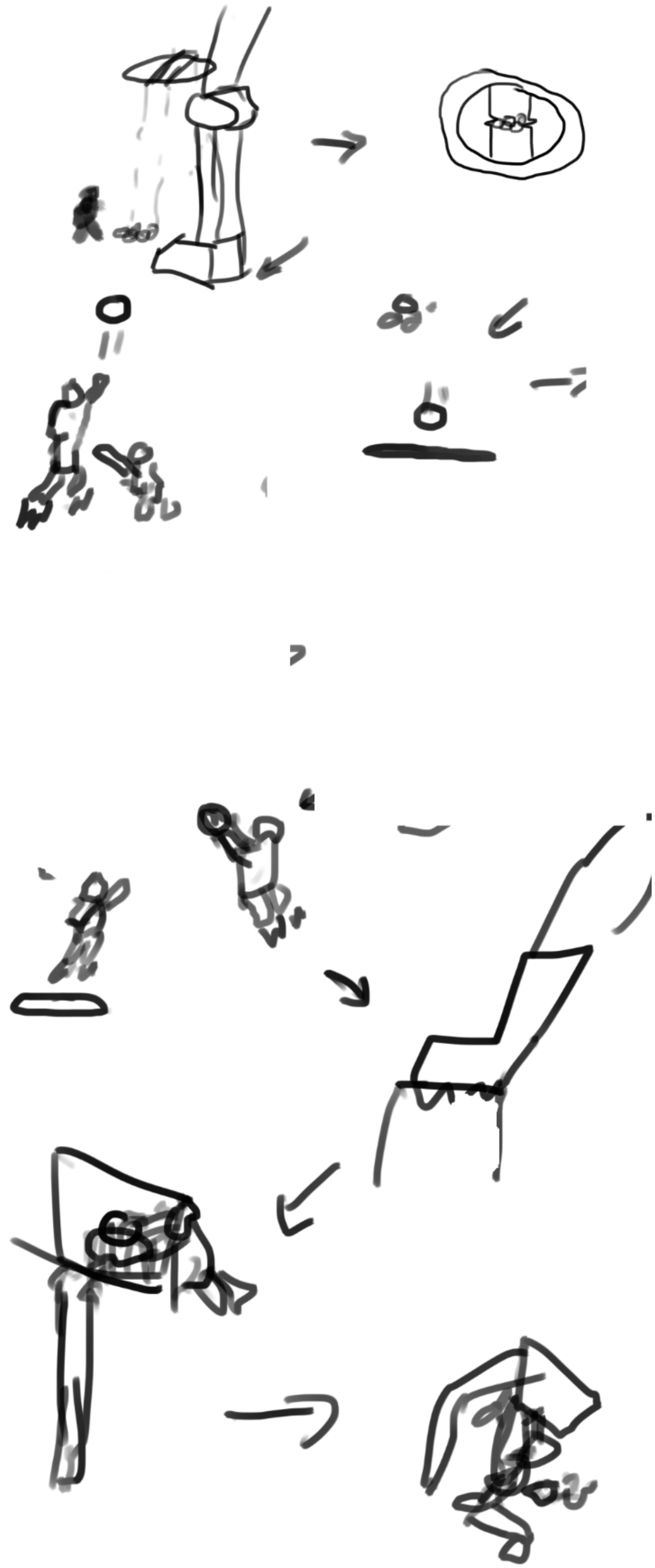

Figure 20: Rough storyboard for scene.

Figure 21: Robot reference.

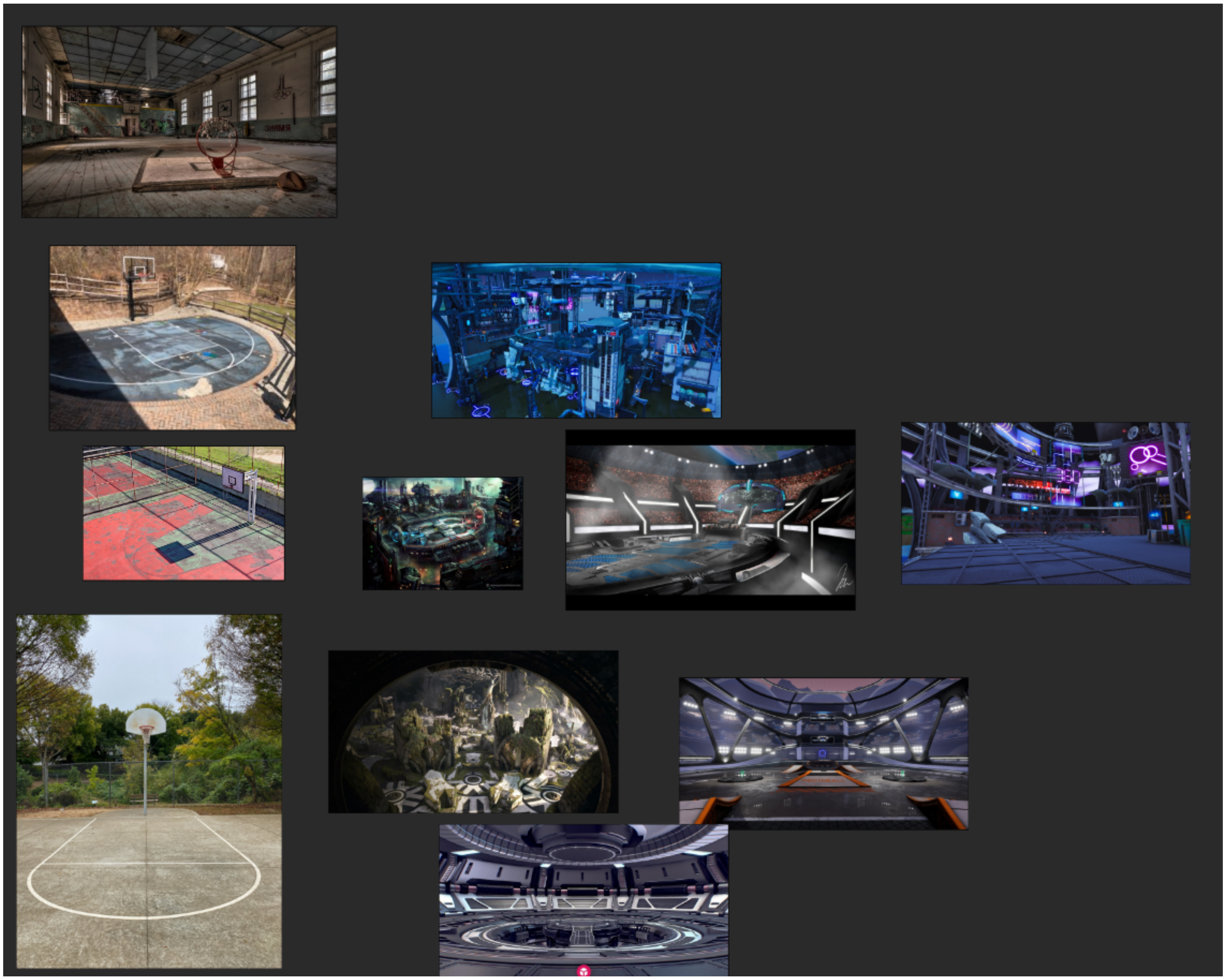

Figure 22: Arena reference.

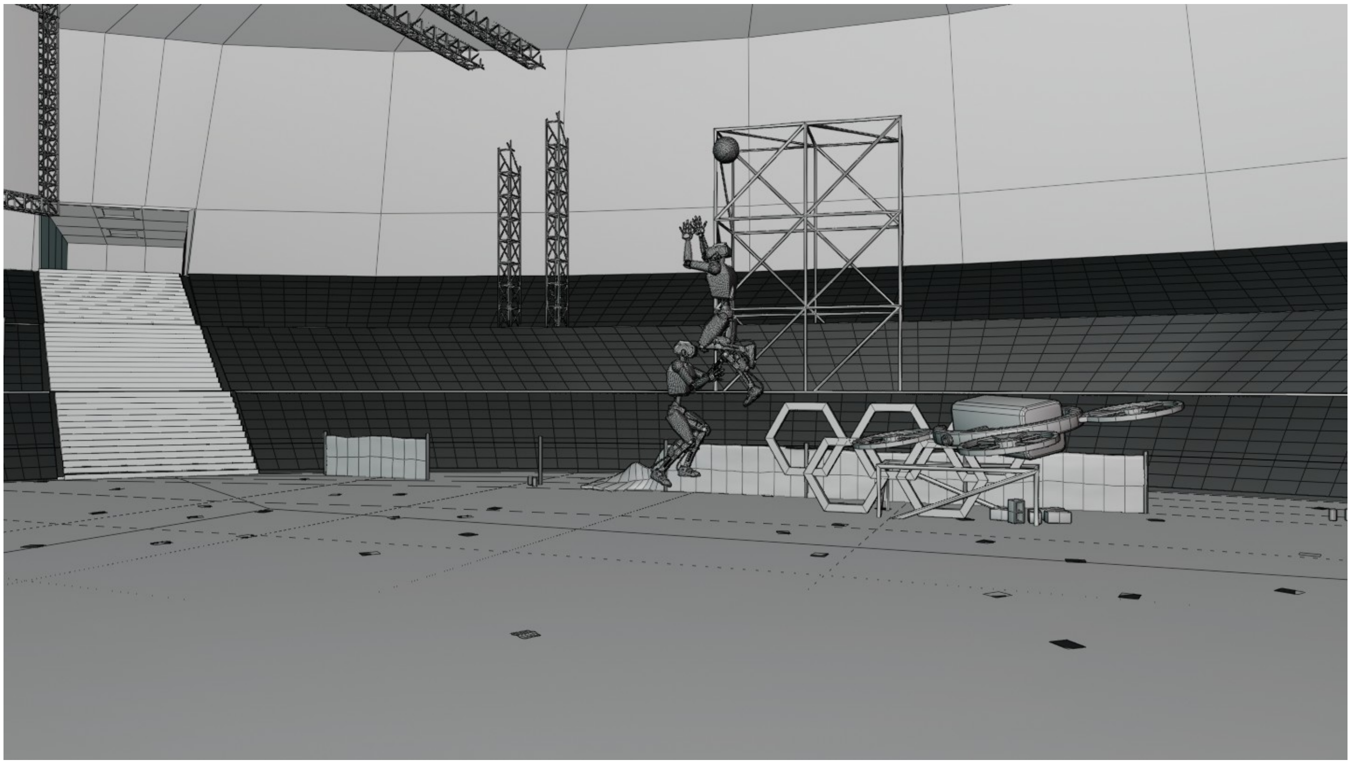

Figure 23: Wireframe of the whole scene.

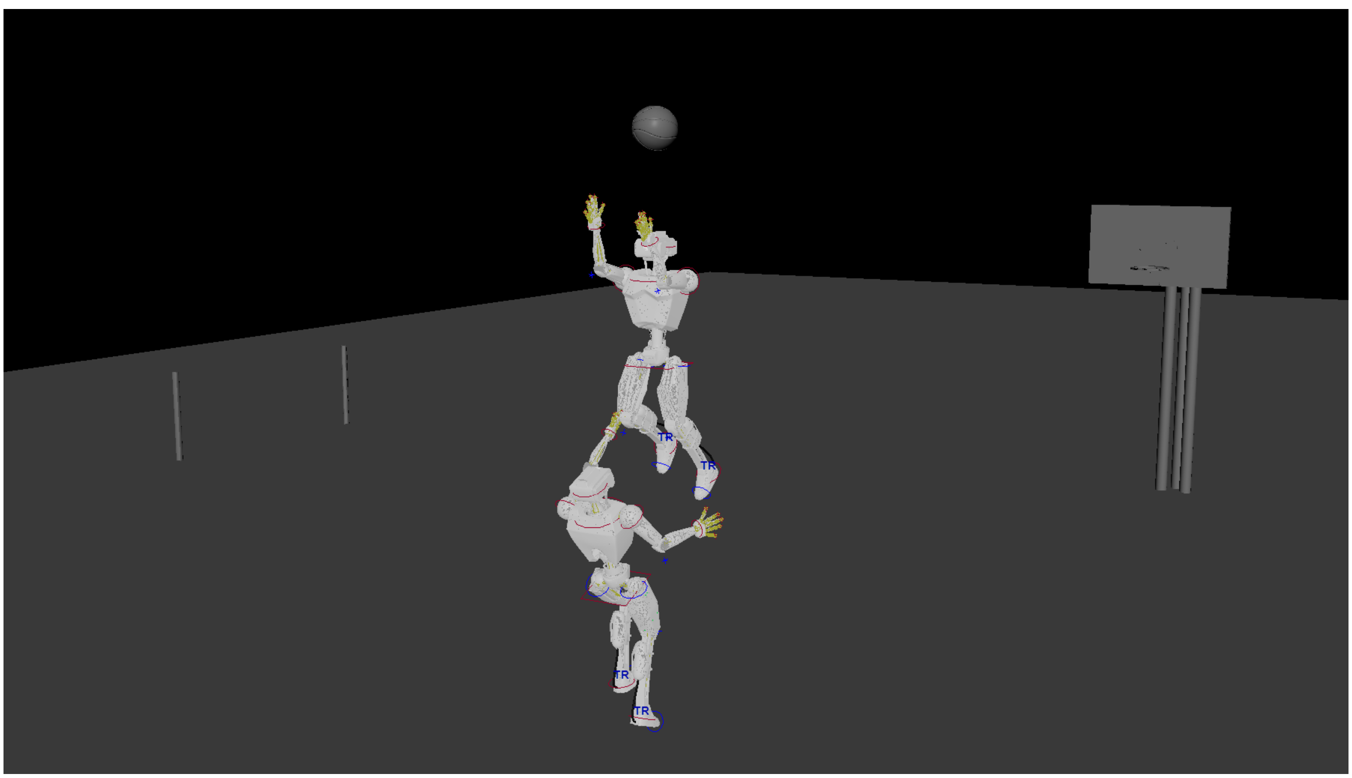

Figure 24: Stripped down Maya scene just for animation.

Figure 25: Close up of the spotlight drone.

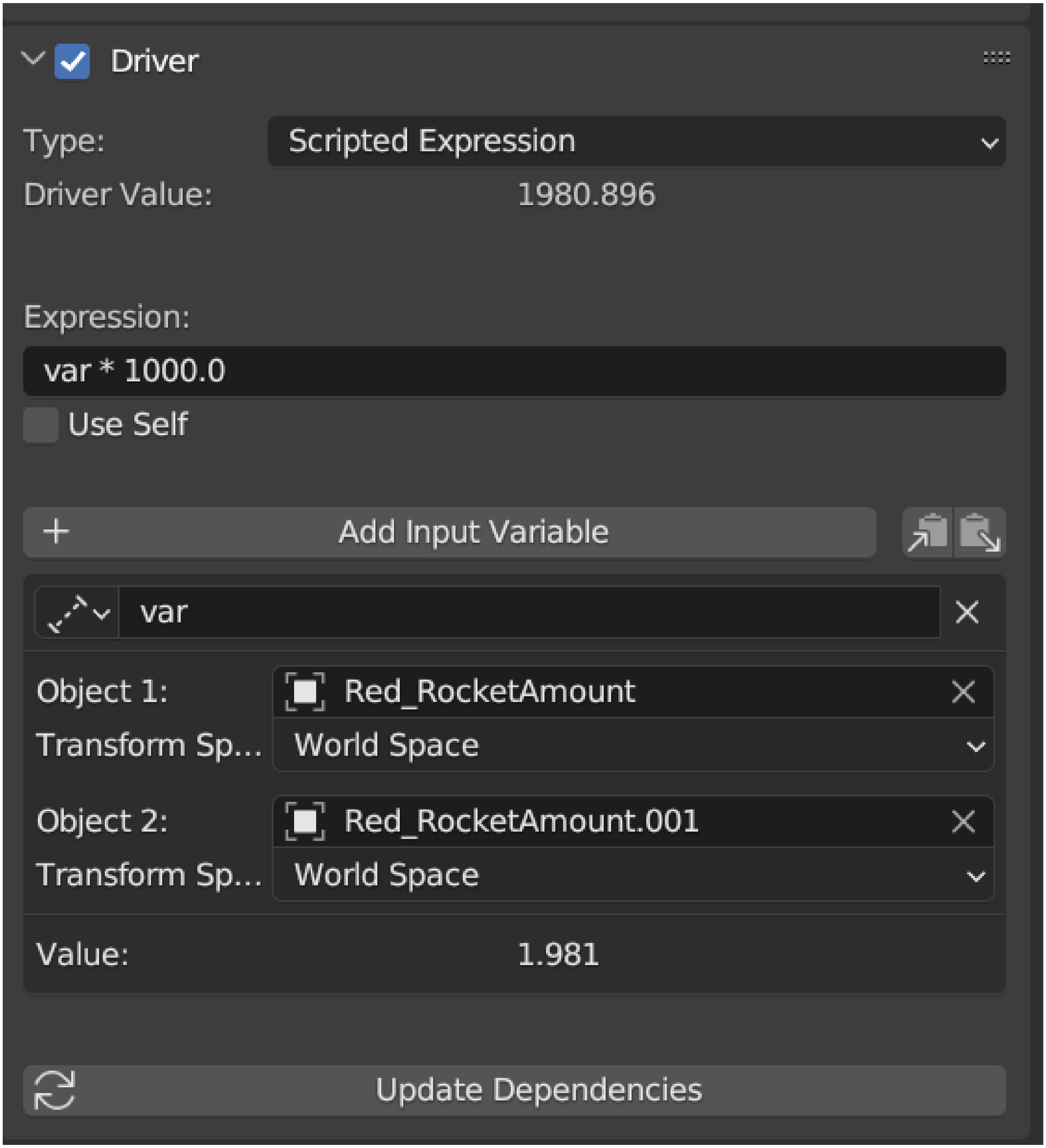

Figure 26: Driver controlling emission from 'rocket boots'.

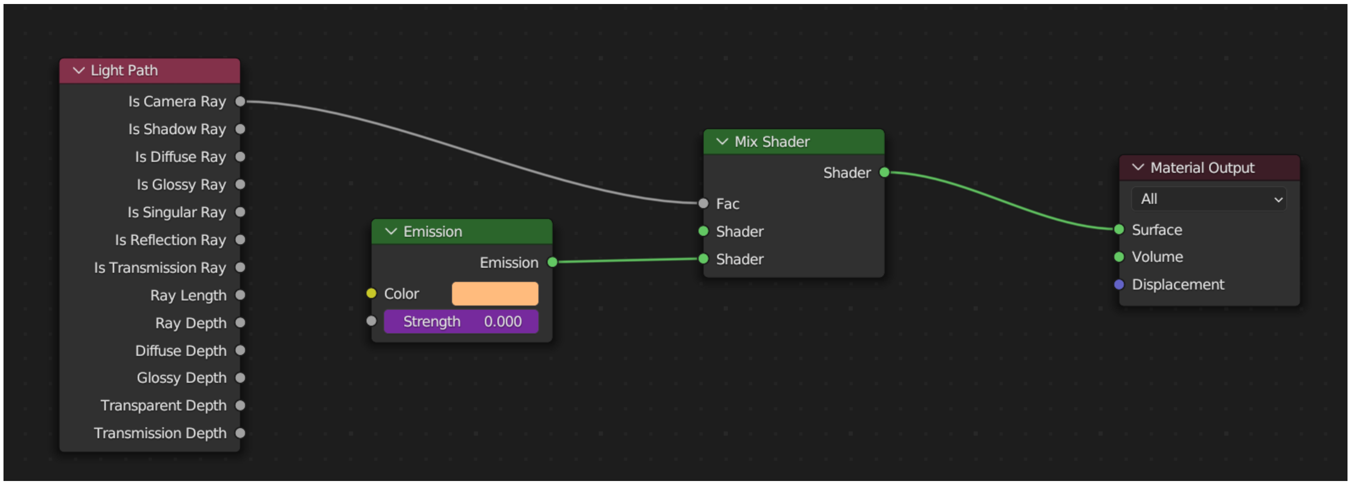

Figure 27: Material to remove transmission from emission.

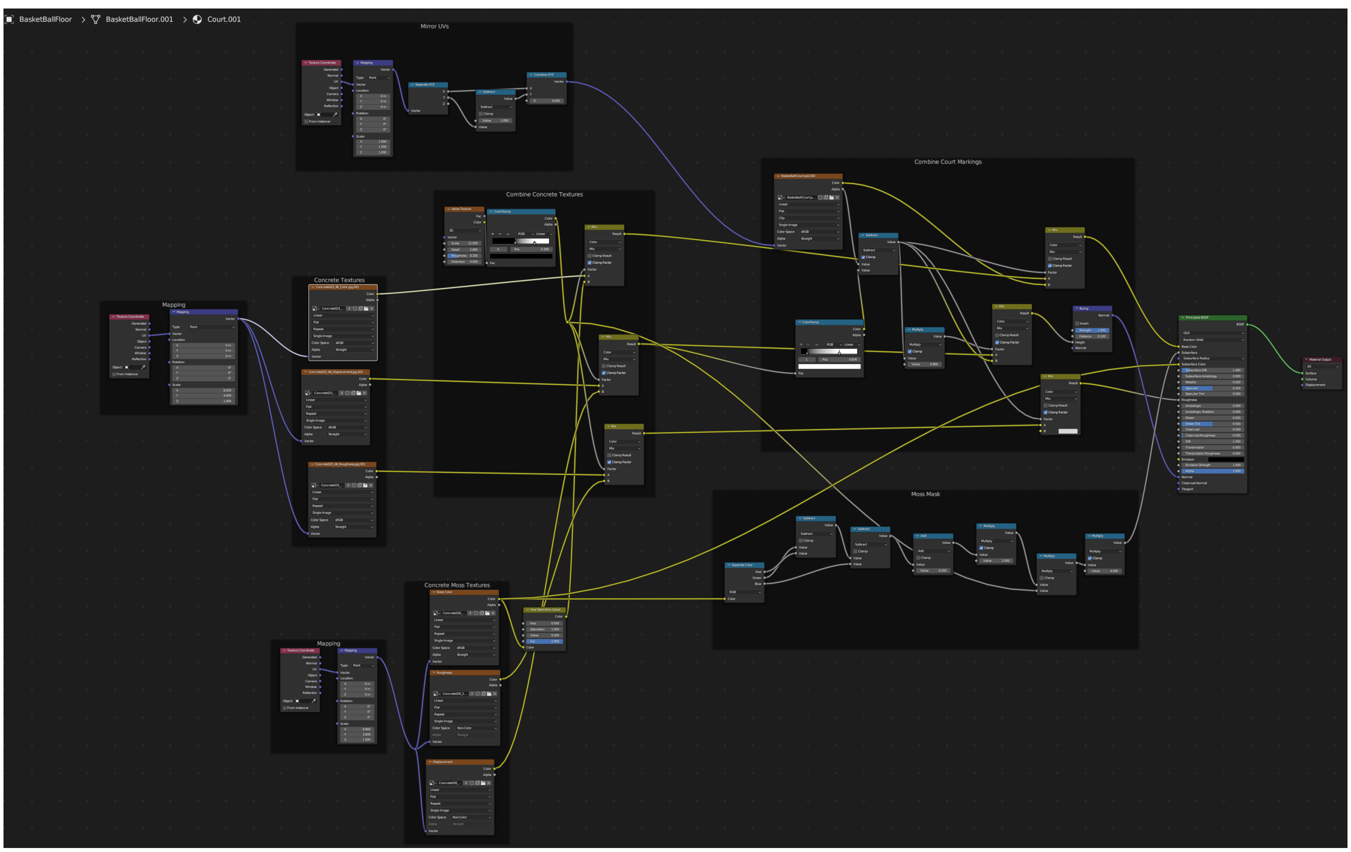

Figure 28: Material nodes for court material.

Figure 29: Close up of court material.

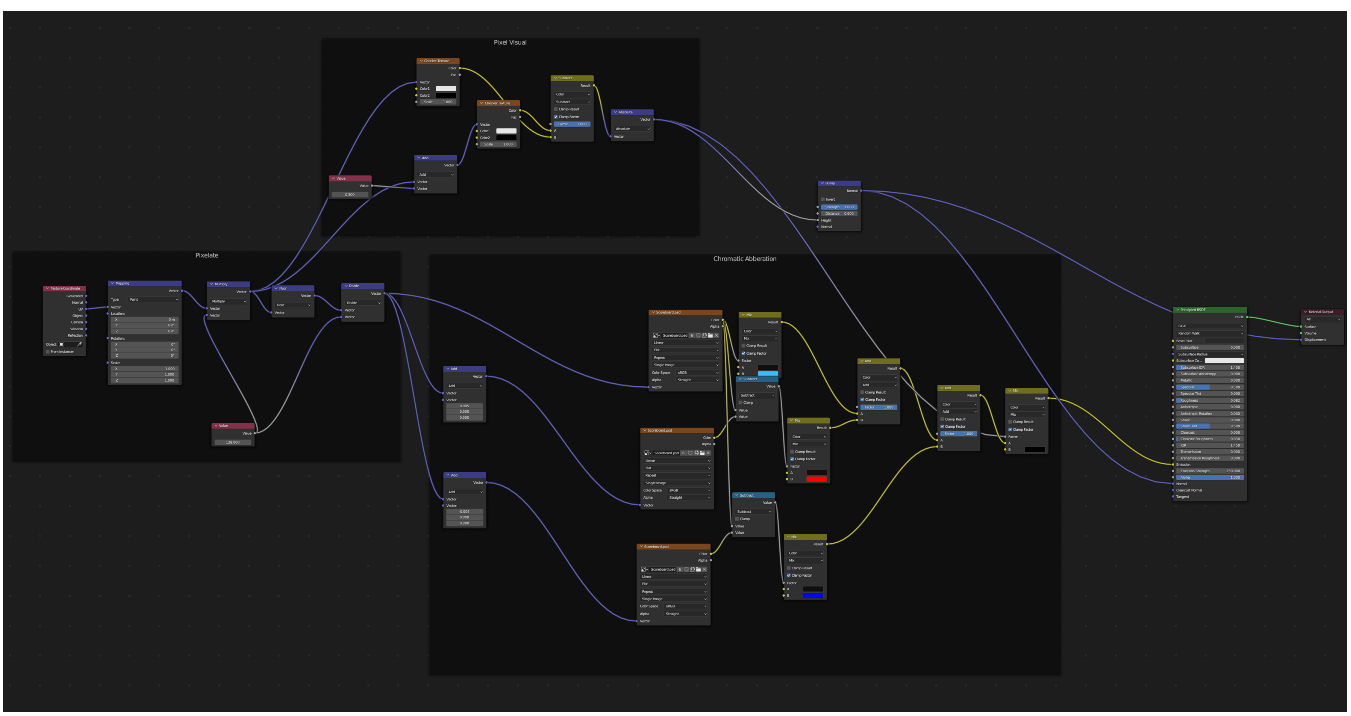

Figure 30: Material nodes for screen material.

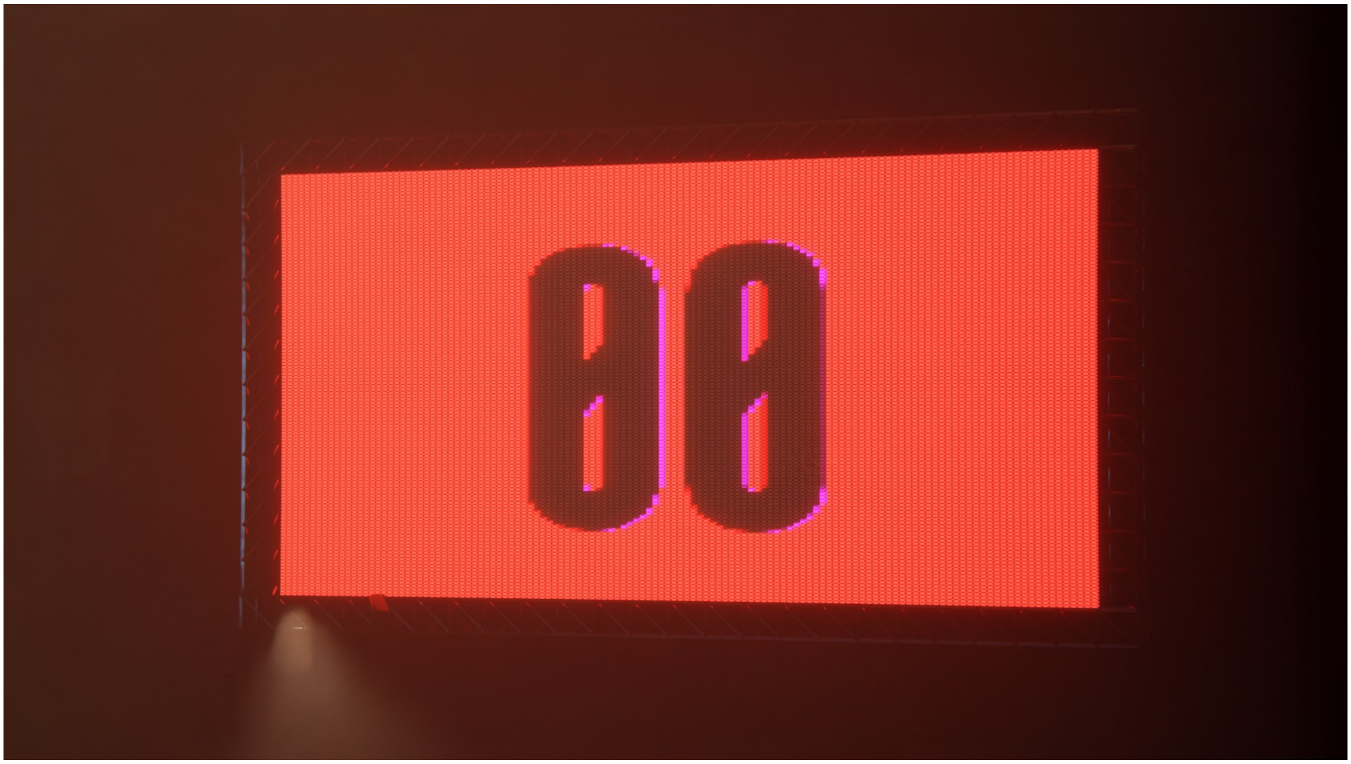

Figure 31: Close up of screen material.

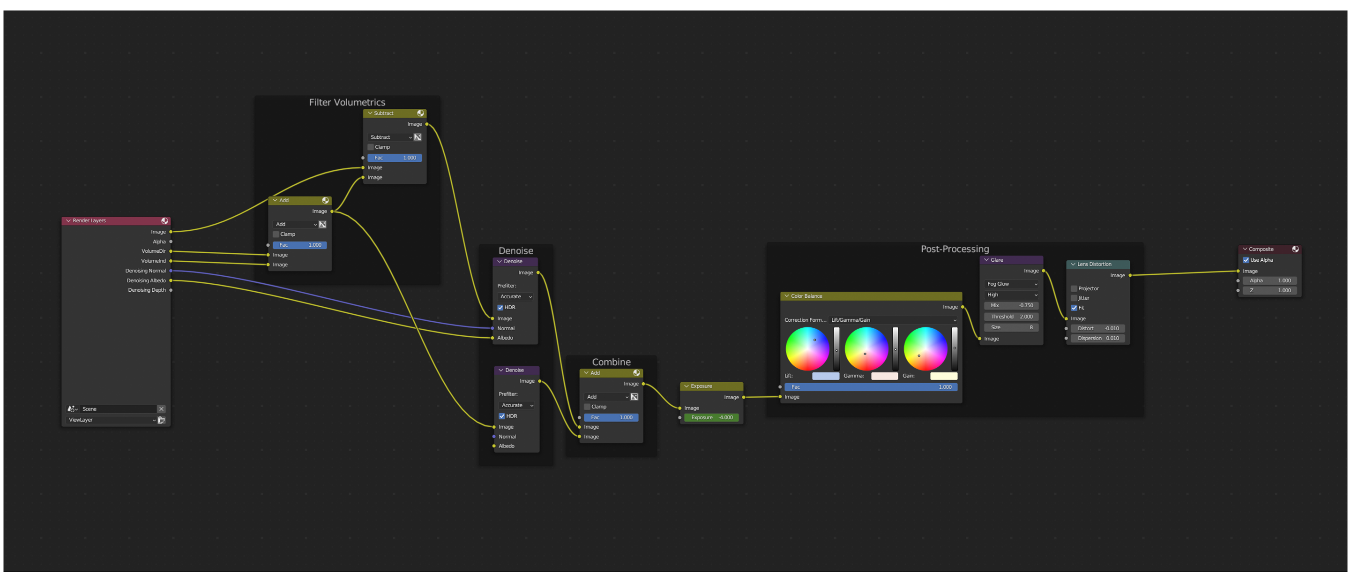

Figure 32: Compositing node system with animated exposure.

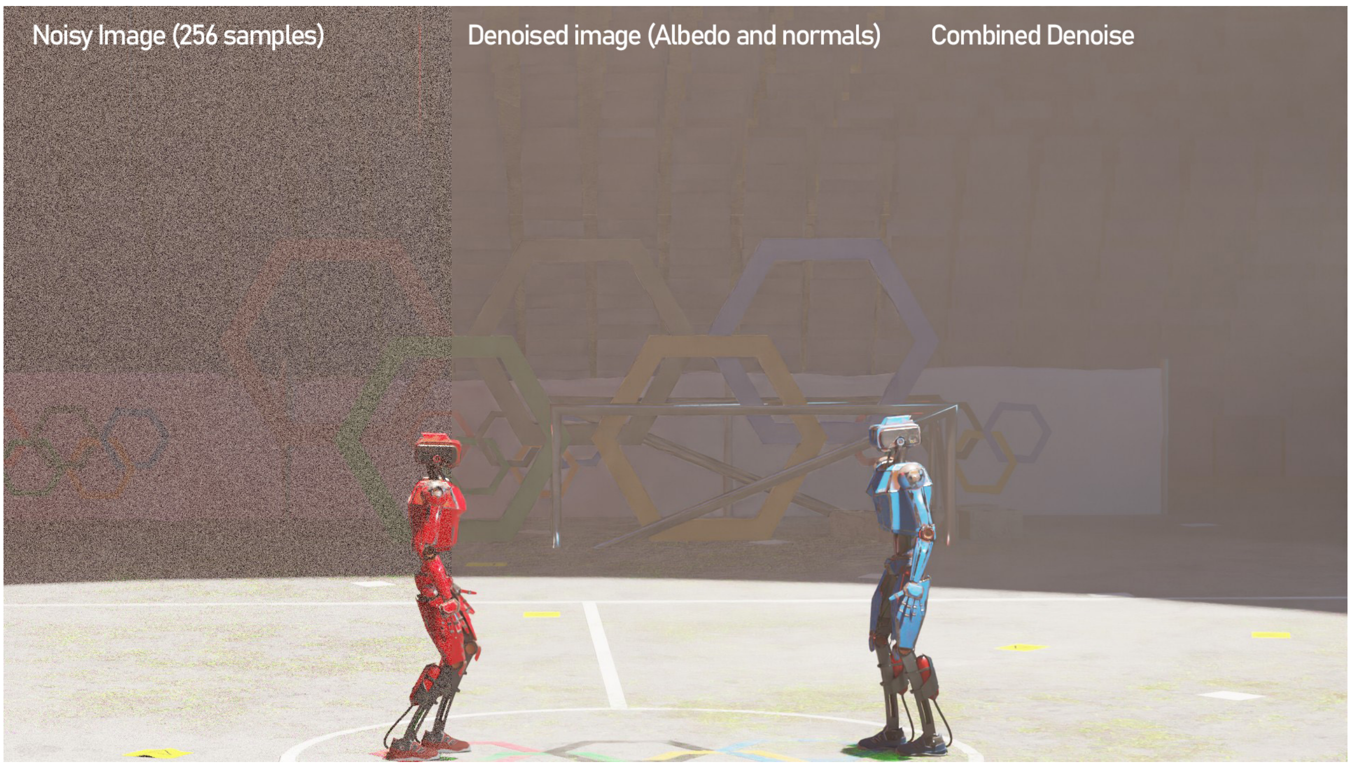

Figure 33: Comparison of the denoise method.