

Bibliography Summaries (CG2)

Kevin Le

September 2020 - December 2020

1 Week 1 Bibliography Summaries

An engine was created for users and programmers to create their own 3D web based application with little to no struggle. It is composed of two main parts, the “Engine CORE” and the “Engine API” which is essential for the engine to function properly [2]. The Engine CORE includes a method to handle and load glTF files, a physics and graphics controller which contains beneficial functions, and important 3D engines and graphic libraries to easily build the 3D environment. The Engine API on the other hand allows users to access all the functions in the Engine CORE. Together with the 3D open source software Blender, users can easily load in 3d objects with included physics and collisions on the web easily. Because of the growth of VR and AR this generation, the group behind the 3D application engine plans on adding these features in the future for a more engaging experience between the user and the web.

The web is a place where users can engage with each other. One way is through a web based framework called VRIA where users can view data, plots, and graphs through VR with other people and actively engage with it. The experience is known as Immersive Analytics in which the goal is for users to feel more “immersed” through a web interactive and collaborative system. VRIA produced this experience by using WebVR for real time communications, DOM (Document Object Model) for web tools that help with functions of objects and JSON for data structure [1]. Similar to the Engine CORE, the VRIA framework was made to be user friendly so that even those novice to programming can create an Immersive Analytics experience on the web. This is thanks to VRIA’s workflow which allows new programmers to just use the built in VRIA Builder that is already provided. Experienced programmers can avoid this route and use the apps code to develop their own experience too. From these articles and papers, I learned that computer graphics nowadays not only switched over to 3D, but it is doing its best to transition itself to a more user friendly and immersive experience through VR and AR.

2 Week 2 Bibliography Summaries

Mobile phones today can do much more than what they could a few years ago. Most if not all phones now contain sensors and cameras that can help make everyone's lives easier. A good example of this is a sample app produced for field analysts called FieldView [19]. Field analysts such as construction workers, firefighters and forestry's collect data to help people around a specific environment learn about its current condition. There are potential problems however such as datas requiring to be physically sent to a location for analyzing before getting feedback during an ongoing mission or more new data not being collected and updated after submitting data. These problems can be fixed with an app on the mobile phone which shows users a visualization system of the field they are operating on through AR. With a phone camera, sensor and cloud datastore with MySQL, field analysts can use Immersive Analysts (IA) to scan, update, and send data with little to no problem in real time. Users with the app can create a visual field with colored models to indicate what each color means, create their own interface when collecting data and update data collectively as a team. The fact that mobile phones can do AR with their cameras and sensors and be used as mini VR machines shows how much computer graphics have progressed.

Other researchers also took advantage of the mobile phones camera and sensor to create a monocular 3D reconstruction system. The app known as Mobile3DRecon was designed to fuse the virtual 3D world and real scene together to capture collision and occlusion in real time [23]. This requires an online mesh generator with 6Dof tracking to constantly update the pixels of the virtual world when scanning with the real scenes. This idea seems like it requires strong hardware and would work better with a PC desktop. However the group behind Mobile3DRecon manage to make their app work by making the depth estimate and incremental mesh back end while making real time dense mesh and 6DoF front end. The team was able to produce a 3D reconstruction system with 1 CPU core on a mobile phone and was able to showcase how better it was compared to other applications such as MVDepthNet and DPSNet. The app was able to do the TSDF (Truncated Signed Distance Field) voxel method and incremental mesh update on a middle end phone. These papers showed how far mobile has progressed throughout the years and how it has grown more useful and stronger in the computer graphics field.

3 Week 3 Bibliography Summaries

Shaders play an important role in computer graphics since it works alongside the GPU and helps generate 3D graphics accordingly. There are multiple types of shaders and all of them play a big part in the graphics pipeline. One of these shaders is called a tessellation shader and it handles terrains in open world games or CGI projects. Rendering wide terrain in real time while maintaining efficiency is key for a large open world game to run and play well. Tessellation

shaders handle this type of situation by doing mesh refinements and culling in the GPU. However, some researchers found the usual method behind tessellation shaders to be a problem since the shaders max tessellation capacity is too low. Because of this, a new algorithm was proposed to create a new mesh shader pipeline to increase the max tessellations capacity [16]. Increasing the capacity is known to cause low performances which the team behind this algorithm knew, so it was further modified to keep the culling and CPU efficient. The proposed function handled tessellation factors with task shaders, performed a culling pass method and relied heavily on mesh and task shaders. The results for the suggested algorithm proved to be promising and did indeed improve wide terrain tessellation. This paper demonstrated to the readers that there are ways to modify shader types to improve performances in computer graphics.

Shaders are programmed to make a 3D virtual environment look more detailed and at times realistic with its lighting, shadows, and other various graphical improvements. A small computer graphics team wanted to make virtual worlds more complex through the creation of realistic weather behavior [8]. The paper focused on virtual winter environments where snows can be blown by winds, potentially accumulate, or be interacted by human movements. A proposed method was to utilize compute shaders so that a dynamic manipulation of meshes was created to make transformable surfaces behave accordingly. The process of this method was to have the render buffer implemented then the surface shader sent to the compute shader. This renders the mesh into separate frames in which each frame is then calculated in the GPU to have the vertices of the surfaces get updated and the mesh be manipulated in the compute shaders. When it came time to use the following method in a snowy environment with 100 reinforced learning AIs, the computer was able to pick up realistic weather behavior and run at a smooth 30 frames per second. From this article, I learned that shaders play a big role in making 3d virtual environments look realistic and at times run smooth.

4 Week 4 Bibliography Summaries

Out of all the computer hardware components, the one that is essential for computer graphics nowadays is the Graphics Processing Unit (GPU). GPU's are important since they deal with frames and speeds in 3D graphic applications such as video games. The one key performance people look for in GPU's is speed improvements since it enhances experiences in 3D applications. Computers users are always creating methods to further improve the GPU, one of them includes implementing a Turing method NVIDIA GPUs [15]. Turing is used in this context to boost performance enhancements and increase speedups in modern gaming applications. The design was used to counter load-to-use stall which increased speedups by around ten percent. By countering load-to-use stalls, the Turing was able to lower hit latency in global loads, enable configuration of data RAM through cache merge, and enable higher warp concurrency improving memory latency. I find speed performance in GPU's to be very essential this

generation since games today are all made up of large 3D environments. The faster the speed the better the experience and if Turing can indeed improve NVIDIA GPUs, I hope it gets implemented in the future for all NVIDIA GPUs.

GPUs provide high performance and handle data movement. Researchers notice however that the cost of swapping and moving data affects application performances negatively due to memory oversubscription. This is due to two main reasons, memory thrashing which occurs due to high latency of tasks and memory contention which causes leading tasks to suffer delay. To counter these problems in GPUs, researchers proposed two main ideas. One was to allow the GPU to access data directly in the storage system to avoid swap in swap out and memory contention [22]. The memory contention aware priority assignment was created to reduce the length of latency tasks and prevent high swapping cost time which causes delay and performance issues. The other idea was to propose a virtual memory management to avoid the problem of memory thrashing. The key plan was to have the lowest priority task lead with the highest latency to avoid the highest priority task from suffering. When two of these ideas were used, latency for each task was significantly reduced and the highest priority task did not suffer from performances. The amount of improvements that can be made to the GPU is fascinating and makes me wonder what researchers will discover next in the future to further improve its performance.

5 Week 5 Bibliography Summaries

Video games have continued to evolve graphically every few years since its creation. The oldest home game consoles used to contain games like Pong which consisted of mere pixels as graphics. Fast forward to today's graphics in some video games, everything looks realistic from the characters to the grasses on the floor. Gameplay used to be the main core part of video games, but due to the vast power of hardware in consoles these days, graphics has also become a must for most gamers. If programmed wrong, it can throw game users off and affect their gaming experiences. It has come to the point where even the smallest details like character expressions matter and can affect gameplay experiences. A team researched a way to create real time muscle based facial animation using shell elements and force decomposition for instance [cite]. Shell elements are used to better model the muscle geometry while modal warping and force decomposition is used to achieve real time performance [10]. For shell elements, a geometrically driven energy function called "Thin Shell Dynamics" is created to allow for stretching and bending of the face. Since force plays an important role for muscles and skins to undergo stretch deformation, force decomposition was created to enable region of interest (ROI) for each muscle to move differently and accordingly. The results for both these techniques ended well with the 3D model faces reacting correctly. Although this paper focussed on using these techniques for medical teams, it can be noted that game companies also use similar programming mechanics to have video game characters react and look realistic as possible.

Facial expressions and graphics in video games is one thing, but sound is another. According to Kenwright, sounds in video games create engagements with players, convey information, and impact mood and performances. In fact, a game with visuals but no sounds is not considered a game since it does not make players feel engaged and immersed which is the main goal of video games. Throughout the paper, Kenwright discussed all the different types of sounds that impact gameplay. From non linear sounds like how a weapon shoots to passive sounds like the background wind [9]. There were also discussions on how sound recording and audio playback for game audio is more of a past tactic. Due to technological advances, most game companies now produce procedural sounds instead which relies on data extraction and training models (machine learning). Although Kenwright's paper is considered a subject within the computer graphics field, it primarily focused on the importance of sounds in video games. Nothing about 3D models or graphics was really discussed. The only factor that was mentioned that slightly indicates computer graphics is the player's sights in video games. A statement that Kenwright wrote about sights that I wholeheartedly agree is that sounds and sights is what defines a video game.

6 Week 6 Bibliography Summaries

Applying advanced lighting and shading to models can make them look realistic in any environment if done correctly. Online artists and game designers apply these techniques to make their work stand out since it creates immersion and projects realism. A team of 3D artists for instance proposed an approximate shading model method for image object modeling and insertion. The goal is to build an object relighting system that allows artists to select an object from an image and insert it into a 3D scene to make it look as if the image is part of the scene already [12]. This technique takes an image fragment and simply puts it into another scene using a pure image based approach for shading consistency. The method does not involve using any 3D models surprisingly since the team behind it found them time consuming and hard to build. By doing object insertion, the group was able to cut time and resources and get the same if not better results than creating and inserting 3D models. The technique is decomposed into three main components. The first component being the smooth component involves using coarse shape. Coarse shape assumes the object to be inserted as an image and estimates its appearance under new illumination. An equation was formed under the team to find the coarse shape which handles the lighting. The other two components which handle shadings are the following, the gross shading and surface detail. With these three components, the group behind the approximate shading model method was able to produce promising results. Images and objects that are non 3D models and are inserted in 3D scenes look real and as if they were part of the environment the whole time. It is amazing to see how much advanced lightings and shadings can do to graphics nowadays.

Advanced lightings and shadings have a powerful effect on 3D models and

art since they help set the mood. A small team investigated commonly used lighting techniques on an animated virtual character to determine which one is more intense and the effects they have on people's mood [20]. The two lighting conditions used are low contrast lighting which uses a large area light to produce a subtle transition from light to dark on an illuminated area, and high contrast lighting which uses a small intense light source to produce hard shadows. These two lighting techniques were projected in different directions as well to demonstrate different intensities. Forty five degrees to the right, above and below the eye level, and no visible light direction at all. Two computer graphic shades, toon shading and smooth shading, were also used in order for the lighting to work properly for accurate results. All of the following that were mentioned are applied to a virtual character's face which will be expressing five different emotions. Happiness, anger, sadness, fear, and disgust. Each person that played a part in the experiment had to deal with a total of 100 trials since they had to deal with five emotions, five lighting conditions, two shading styles, and two repetitions. The goal of the experiment was to see if people were able to identify what emotions the character was portraying, how expressive the emotions were portrayed, and how appealing the character was overall. The following results were found, smooth CG shading was rated more appealing than toon shading for all emotions and low contrast conditions were rated more appealing than high contrast conditions. The most interesting result found was that the intensity of emotion perception is consistent across lighting conditions for all emotions. For example, it was expected that the angry emotion with high contrast would be considered the most intense. This however was not the case and all emotions ended up having the same intensity! This paper showed more unique results lighting and shading has on computer graphics.

7 Week 7 Bibliography Summaries

Skeletons in computer graphics are composed of points which are centered inside shapes to provide a shape representation. 2D shapes contain a set of medial curves while 3D models contain a set of medial surfaces. For years, computer graphic researchers have found ways to extract curved skeletons. The common method used was called the grassfire analogy in which the shape boundary is set on "fire" to form a skeleton where the fire fronts meet and extinguish. The grassfire analogy is used to find the centeredness measurement which is needed to find the 3D curve skeleton. A team of researchers discovered that many people use the grassfire technique to form their own methods of extracting 3D curve skeletons. However most of these methods required tedious work and were costly. To avoid these conflicts and make the process easier, the group produced their own method which provided a simple and stable centeredness measurement that can run fast and produce high quality results [11]. The method also employs minimum set covers to optimize ridges which helps generate clean and compact curve skeletons without the tedious adjustments. The process goes as follows, first the team generates the medial surface of the 3D shape. This is done by

using a divergence based technique to detect ridges within the 3D model then using a connection strategy to connect these detected ridges with the gradient vectors. The team then performs centeredness measurement by creating their own algorithm which computes the burning time of a point on the medial surface. This algorithm creates centered points and contains two parts, a main loop that simulates fire propagation from a source point on the medial surface and a termination condition that checks if the manifold surface source point is broken or not. The centered points from the centeredness measurements then become optimized using the “minimum set covers” technique which creates a centeredness field. The optimized points are then connected through neighboring which produces the end result which is a 3D curve skeleton. The team used their new 3D curve skeleton method and compared them with three other known ones which were the ET, contradiction based, and advection based method. The result ended up being really positive with Li, Chu, and Wang’s 3D skeleton curve technique being the cleanest, accurate, and fastest one out of the three.

There are other factors that can affect curved surfaces in computer graphics as well like energy. An energy known as smoothness energy is used for optimization in geometry processing and can smooth data on surfaces, denoise data, and do various other things. The Laplacian energy for instance is a smoothness energy that uses minimizers to solve the biharmonic equation to create smooth results on computer graphic surfaces. However, the Laplacian energy uses the zero Neumann boundary condition which results in boundary distortions. A team wanted to apply smoothness energy on curved surfaces but wanted to avoid distortions as well. To achieve this, the group used a pre-made Hessian energy equation that was produced by a researcher named Stein and made further improvements and tweaks to it [18]. The team formed their own Hessian energy equation that took in curved surfaces since the previous one did not accommodate them and only took in surfaces in R^2 space. The new Hessian energy basically corresponds to the Laplacian energy but with a “linear as possible” natural boundary to decrease distortion. Discretization which included the vector of Dirichlet’s energy and the Crouzeix Raviart formula was also used in the new equation for various smoothings and to solve interpolation problems. The team’s Hessian energy formula was able to apply smoothness energy towards curved surfaces while also avoiding distortions and other problems. I never knew how important curved surfaces were in computer graphics until I read these two papers which showed me how far researchers went to improve them.

8 Week 8 Bibliography Summaries

Shadows in computer graphics can be used to discover more information about lighting in 3D scenes. A team of computer graphic researchers presented a photometric registration approach to estimate reflectance and illumination results in indoor scenes. This approach takes shadows into account and requires multiple steps. The team’s process goes as follows, first the specular reflections and

cast shadows are detected on arbitrary textured surfaces to estimate the 3D position and direction of light sources [7]. The light source direction is achieved by using Phong's equation to find the illumination map value which is then used to detect "specularities". The light source position is done differently and incorporates information brought by specular effects from an analysis approach of cast shadows. Using the illumination map value and the subset of light points from the light direction and position, the team was able to estimate the color of the light sources which is used to recover illumination. Once the illumination was found, the team worked on finding the reflectance next. Finding it required the group to solve and obtain an estimated specular reflectance value through Phong's model once more. With an algorithm formed to find estimated values of illumination and reflectance, the group moved on to testing it on synthetic and real scenes to see how accurate and effective their formula was. The results ended up being quite satisfying with the error only being around 10 percent or less than average. This shows that the estimated values were close to the actual values which is quite amazing. This paper showed me that shadowing methods such as cast shadowing can benefit and help solve computer graphic problems like estimating values for reflectance.

When lighting such as low-key lighting or dark shadows is casted on realistic 3D models or shapes, it is known to add a sense of gloom. With this notion in mind, it can be assumed that if dark shadows were applied to animated characters with facial expressions, it would increase its intensity. However this was proven wrong by a team that did a set of novel experiments investigating the effects of brightness and its intensity on cartoon and realistic characters. The experiments used a static three point lighting setup to manipulate light brightness and shadow intensity on CG characters. The group behind this research was interested in the effects of lighting on recognition of emotion and overall appeal [21]. Because shadows had a more impactful result during the experiment with realistic characters, I will primarily focus on that part of the research. Throughout the procedure, the team discovered that dark shadows have a significant effect on recognition rate. Participants that were given the question of if they recognized the character expression or not overall rated the emotions as less recognizable and less appealing. However when users were asked to rate how intense the realistic characters looked with dark shadows, the group discovered that the angry and fear emotion had no effect. As mentioned before, dark shadows generally add a sense of gloom to art pieces and because of this notion it was predicted that emotions such as anger or sadness would receive an increase in intensity or have some sort of intensity change, but that was disproven for realistic models under three point lighting. Overall the team discovered that the more realistic the characters look, the less effect shadow has on appeal. Before this article, I used to think that shadows in general always leave some sort of impact on 3D models and shapes. But I was wrong and these results taught me that under certain lighting, shadows can be ineffective.

9 Week 9 Bibliography Summaries

Ray Tracing is a rendering technique that handles lighting to make 3D applications look realistic. When people hear the phrase ray tracing, they think of beautiful graphical games like Metro Exodus or Battlefield V. What about old 3D games that don't look realistic or beautiful back in the mid 2000's. Can ray tracing be applied to such video games? The answer to that is surprisingly a yes. A small research group decided to take a look at an old project and paper developed back in 2004 that dealt with ray tracing in an old game. The 2004 project was done by a college student who applied a real time ray tracing library which was new at the time to a 3D first person shooter game called Quake 3. The team basically took a look back at the program and equipment used and analyzed the process and procedures that were done to ray trace Quake 3 [14]. Back then a PC with a single core was used and due to its weak performance, the rendering was done in 64x64 or 128x128 pixels. An OpenRT library which used a C++ interface was applied to create rays for tests on shadows and reflections. Here are the following highlights and techniques used to create ray tracing in the old game. For lighting, the team added a "flashlight" on the player which created a spot light. To my surprise, this causes an effect of dynamic lighting since it introduces real time shadow cast which was not in the original Quake 3. For shading, the team used a couple of shadow rays around a center of light to give a better performance and pleasing look. New special effects were also added into the old game to further achieve ray tracing such as reflections, glass shaders which were used to make water look realistic and a shader based trilinear filter which were implemented to improve the image quality. It is amazing to see implementations of ray tracing in old 3D video games. I thought it was not possible, but this paper taught me that with proper lighting, shading and shadow effects, ray tracing can be made possible.

Ray tracing is used to compute high quality images and some methods that successfully do that can face high incoherent ray sets. Incoherent rays are costly and can cause problems when ray tracing, however there are ways to mitigate this issue and increase performance. One of these techniques is called ray reordering in which incoherent rays are reordered to form coherent ray sets to accelerate ray scene interactions (better known as trace kernels). Multiple research teams have already done ray reordering before, however one group wanted to analyse this algorithm even further to see if it can be more improved. The method Meister and his group went with was Two Point Sorting Key which achieves consecutive ray subsets by forming compact sets of ray origins and termination points [13]. Despite this past ray reordering technique being the best one, it still had a problem and required the team to find the termination point first before computing the sorting key. This is where the team comes in and adds their own method, a termination point estimation technique that further maximizes the ray coherence. The termination point estimation can be done in two separate ways. The first procedure uses what's called a "bounding box" which sets a constant estimated ray length value. The other method uses a spatial hash table to store the sum of ray lengths and counts to find the termi-

nation points. Once the termination points are estimated and found, they are used in the Two Point Real method for ray reordering and testing. The results were overall decent with the reordering technique achieving speed ups. However it did not pay off in general since there was some disproportion between the fast trace kernel and slow ray sorting and data reordering. From this paper I learned that it is not easy to find the best solution for ray tracing.

10 Week 10 Bibliography Summaries

Radiosity is a computer graphic application that handles global illumination which is a key component for image realism. An algorithm called “Instant Radiosity” approximates the illumination of an image through the use of virtual point lights. This method represents the last bounce of illumination that is emitted by a set of virtual point light sources (VPLs). The Instant Radiosity algorithm however has a main issue. In large occluded scenes, the method has a hard time distributing VPLs in a correct manner which prevents it from giving high quality indirect illumination. This problem sparked a research group to make further improvements to the algorithm. The amount of new changes and additions that were added made the new and improved Instant Radiosity algorithm one of the first interactive global illumination algorithms [5]. The method works in highly occluded scenes, supports moving cameras and light sources, and places VPLs in locations that matter the most. The group achieved this by minimizing temporal flocking (which kept the VPL distribution stable), distributing VPLs according to the amount of lights on the image and by implementing multi bounce indirect illumination. The whole process is quite long to explain so I will only go over a few interesting techniques that were implemented. Instead of assuming that the VPL is synonymous with a single light path, the team marginalized over all light paths that ended up at the same VPL location. That way when it comes to computing the total radiosity, the average over several paths is taken instead of a single path. If the VPLs are distributed according to their radiosities, then all VPLs will have the same intensity. The revised Instant Radiosity method outperformed the original and other similar methods in terms of temporal stability and illumination quality. Radiosity is a graphic application that is new to me, so the research paper taught me more about it and the algorithms surrounding it.

Translucent models that are rendered well can greatly enhance realism. The interactions of light within or in between these objects and the environment can produce pleasing visual effects like color bleeding, soft shading and light glows. There are key techniques that are used to help render these types of scenes quickly and realistically. Monte Carlo’s ray tracing method used both inter reflection and sub surface scattering to achieve such rendering, however it was too slow to be practical. Many researchers attempted to improve these techniques, but they still failed to achieve interactive rendering speed and only focused on a single translucent object. It wasn’t until 2013 when one team managed to achieve interactive rendering speed on multi-layer translucent objects.

This was done using a simple analytic model that combines the diffuse inter-reflection and subsurface scattering technique. The algorithm basically extends the classic radiosity work in 1984 by including a subsurface scattering matrix which is constructed using an analytical model [17]. The model created by the team uses an infinite series of light interactions (bounces) to compute the final radiosity of an object. The entire light transport or radiosity of a scene is basically computed using the sum of all bounces of light with an equation formed by the group. This is one of the few important parts of the analytic model which involved radiosity. It is interesting to see many research teams adding and extending the original radiosity work into their algorithms. I never knew radiosity can be used to further improve rendering techniques until I read these research articles.

11 Week 11 Bibliography Summaries

Visual Realism is the idea of making images appear more like a photo rather than computer generated. There are a couple of ways to assess visual realism such as automated computational predictions or subjective human judgement. Automated prediction uses image equality metrics to calculate distortions of global illumination to predict CG quality while human judgement relies on psychophysics experiments to measure how realistic the rendered images are. One research group had a goal of using computational models to understand how people perceive visual realism. Instead of detecting image artifacts, the model will handle realism assessment. The team achieved this goal by establishing a new benchmark data (Visual Realism Dataset) that compose of visual realism score and human annotated attributes and by developing computational models that predict visual realism [3]. To start, the team created a dataset that collected human ground truth and analyzed data to gain insights into human perception of visual realism. Two psychophysics studies were tested and the following outcomes were produced. Participants who were gamers or photographers have higher sensitivity than regulars and males have greater sensitivity compared to females. It's very interesting to learn that gender has an influence on visual realism perception. The visual realism dataset is then applied to three computational models the team implemented for realism assessment. Two are empirically based models that introduce the design of image features that encode human perception. The features that capture the factors of realism are naturalness, attraction, oddness, and face. These traits were tested and trained under 2 classifiers, Support Vector Machine (SVM) and Multi Layer Perceptron (MLP). The third computational model uses convolutional neural networks (CNN) to get performances on all images from the Visual Realism dataset. These models provided realism prediction and image classification and helped the team identify image attributes that were most related to visual realism. It is fascinating to see the different types of experiments and models researchers used to help them gain a better analysis of visual realism.

There are multiple visual realism experiments that were conducted to under-

stand the possible effects it has on humans. One group performed an experiment that used virtual reality to test the effects visual realism has on people's spatial memory and exploration patterns [6]. Another general goal for this experiment was to see if there was a significant relationship between visual realism and object location memory. The experiment was done using two different levels of visual realism, high and low visual realism, and two different virtual environments, the kitchen and living room. High visual realism is just your standard visual realism while low visual realism is more unique with its different lighting and lower polygon count. The group predicted that high visual realism will improve spatial memory because it was believed that people will remember things better if it was more realistic. However the experiment results showed that visual realism had no significant factor in influencing object location memory. In the experiment, participants were quickly timed and were given the task of walking around the two previously mentioned virtual environments remembering the placement of objects. Once the timer was done, participants waited in an empty room for 30 seconds and were teleported back to the virtual environments with the last task of reconstructing and placing all the objects back in their original locations since the environments were changed and manipulated. After the experiment, participants had to answer a set of questions that tested their object location memory skills such as if they noticed any differences between the two environments throughout the experiment. Paired t-test was conducted after getting the results to compare the error between levels of visual realism. Shockingly, the outcome showed no significant difference in error for high visual realism and low visual realism. The team also conducted an analysis that showed no differences between high and low fidelity and the participants' performance in high and low visual realism in general. Some interesting discoveries the team managed to find is that most people gaze longer in low visual realism environments and that females were better than males in identifying object locations in the virtual kitchen environment. Otherwise, the overall outcome demonstrated that there is no significant relationship between visual realism and object location memory.

12 Week 12 Bibliography Summaries

Technological advances have made virtual reality become more common in today's society. Plenty of users nowadays have some sort of virtual reality headset and overall there has been an increase of users in the VR space, especially in 2020 during the year of a global pandemic. More people using the virtual reality space simply means more VR creators and researchers looking for additional ways to further improve the VR experience. One simple way to make a virtual reality experience better is to increase how realistic things look. This is especially true for virtual avatar characters and animations. If a person wants to interact with other users in a virtual reality space but has an unrealistic looking virtual avatar with no animations, it can make the VR experience weird and uncomfortable. One research group understood the importance of virtual experience that it

pushed them to do research on facial animations. Franco and his team found avatar self identification to be very important since avatars represent the user's "true self" in a virtual environment [4]. To increase one's virtual experience, one must feel embodied within an avatar they can identify with and have some sort of ownership over. Most related work surrounding avatar self identification focused on the body, however this team's self identity is strongly associated with the avatar's own face. In this paper, the team explores well designed facial animations that can affect self identification. Three facial animation conditions were tested in the experiment with the first condition being a still face in which the avatar's face remains static. The second condition is lip sync in which the avatar's lips moved in real time to the participants mouth movement. The last condition adds animations alongside lip sync such as eye blinking and subtle facial animations. During these conditions, participants were asked to read a speech in front of a virtual mirror. The following results were demonstrated, when the group used Conover tests, they were able to find significant differences in gaze percentages of chest and face. People looked more to the face during the animated plus lip sync or lip sync face conditions. However during the still face condition, people looked more in the chest area. Participants were also most embodied in the animated plus lip sync and lip sync face conditions than still face which makes sense. Enfacement illusion, the idea that participants experience tactile stimulation on the face as they touch the avatars, was also researched in this paper. It was discovered that enfacement and embodiment were closely linked and that animation plus lip sync and lip sync face conditions produced higher enfacement levels. In summary, facial animations increase self identification on virtual characters which can also increase accountability and good behaviors. Although the outcome for this experiment was somewhat predictable, it is amazing to see the amount of impact animations have on virtual reality experiences.

Augmented reality is another interactive experience that has become quite common presently. One research team recently created an AR application called ARAnimator which handled animations of virtual characters with real life environments [24]. The very idea of having virtual contents interact with real environments is known to be very challenging since it requires virtual objects to align accurately with the real environment. Some AR work even requires complicated setup such as motion capture hardwares to function properly. The group behind ARAnimator however managed to develop an intuitive tool that achieved their task without the need for precise hardware and scenes. What was the tool? Well the tool simply uses mobile phone motion gestures to create virtual character animations. For example, if a user takes their mobile phone and creates a jump like motion gesture, ARAnimator will create a virtual doll character doing a jump animation. Users will then be able to preview the animation on their phone based on the gesture they made. If the animation looks bad or is inaccurate, the tool also has an edit option that allows users to make changes to the animation in its x,y, and z coordinate. Past related work that handled virtual character animations in real environments only used 2DoF or 3DoF, which limited the amount of animations that can be done. Since Ye and

his team are using mobile phones for motion gestures however, it allows their ARAnimator system to have a 6DoF design which allows for free form character controls and more expressive creations. The group also employed an SVM approach in their system to accurately predict motion gestures. The experiment results for ARAnimator ended up being very positive with all participants agreeing to the system being easy to use and intuitive. It was astonishing to see a good amount of research being done on animations for interactive experiences such as augmented and virtual reality.

References

- [1] P. W. S. Butcher, N. W. John, and P. D. Ritsos. Vria: A web-based framework for creating immersive analytics experiences. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–12, 2020.
- [2] N. Eamnapha, S. Nuratch, and W. Lenwari. The graphics and physics engines for rapid development of 3d web-based applications. In *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 89–92, 2019.
- [3] S. Fan, T. T. Ng, B. L. Koenig, J. S. Herberg, M. Jiang, Z. Shen, and Q. Zhao. Image visual realism: From human perception to machine computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(9):2180–2193, 2018.
- [4] M. Gonzalez-Franco, A. Steed, S. Hoogendyk, and E. Ofek. Using facial animation to increase the enfacement illusion and avatar self-identification. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):2023–2029, 2020.
- [5] P. Hedman, T. Karras, and J. Lehtinen. Sequential monte carlo instant radiosity. *IEEE Transactions on Visualization and Computer Graphics*, 23(5):1442–1453, 2017.
- [6] Jiawei Huang and Alexander Klippel. The effects of visual realism on spatial memory and exploration patterns in virtual reality. In *26th ACM Symposium on Virtual Reality Software and Technology, VRST '20*, New York, NY, USA, 2020. Association for Computing Machinery.
- [7] S. Jiddi, P. Robert, and E. Marchand. Detecting specular reflections and cast shadows to estimate reflectance and illumination of dynamic indoor scenes. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–12, 2020.
- [8] Andreas Junker and George Palamas. Real-time interactive snow simulation using compute shaders in digital environments. In *International Con-*

- ference on the Foundations of Digital Games, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery.
- [9] B. Kenwright. There’s more to sound than meets the ear: Sound in interactive environments. *IEEE Computer Graphics and Applications*, 40(4):62–70, 2020.
 - [10] Jungmin Kim, Min Gyu Choi, and Young J. Kim. Real-time muscle-based facial animation using shell elements and force decomposition. In *Symposium on Interactive 3D Graphics and Games, I3D '20*, New York, NY, USA, 2020. Association for Computing Machinery.
 - [11] L. Li, W. Wang, and Y. Chu. A simple and stable centeredness measure for 3d curve skeleton extraction. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–15, 2020.
 - [12] Z. Liao, K. Karsch, and D. Forsyth. An approximate shading model for object relighting. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5307–5314, 2015.
 - [13] Daniel Meister, Jakub Boksansky, Michael Guthe, and Jiri Bittner. On ray reordering techniques for faster gpu ray tracing. In *Symposium on Interactive 3D Graphics and Games, I3D '20*, New York, NY, USA, 2020. Association for Computing Machinery.
 - [14] D. Pohl, S. Panneer, D. S. Vembar, and C. S. Marshall. 15 years later: A historic look back at “quake 3: Ray traced”. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 401–412, 2020.
 - [15] R. Rangan, N. Turakhia, and A. Joly. Countering load-to-use stalls in the nvidia turing gpu. *IEEE Micro*, pages 1–7, 2020.
 - [16] B. Santerre, M. Abe, and T. Watanabe. Improving gpu real-time wide terrain tessellation using the new mesh shader pipeline. In *2020 Nicograph International (NicoInt)*, pages 86–89, 2020.
 - [17] Yu Sheng, Yulong Shi, Lili Wang, and Srinivasa G. Narasimhan. A practical analytic model for the radiosity of translucent scenes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '13*, page 63–70, New York, NY, USA, 2013. Association for Computing Machinery.
 - [18] Oded Stein, Alec Jacobson, Max Wardetzky, and Eitan Grinspun. A smoothness energy without boundary distortion for curved surfaces. *ACM Trans. Graph.*, 39(3), March 2020.
 - [19] M. Whitlock, K. Wu, and D. A. Szafrir. Designing for mobile and immersive visual analytics in the field. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):503–513, 2020.

- [20] Pisut Wisessing, John Dingliana, and Rachel McDonnell. Perception of lighting and shading for animated virtual characters. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, page 25–29, New York, NY, USA, 2016. Association for Computing Machinery.
- [21] Pisut Wisessing, Katja Zibrek, Douglas W. Cunningham, John Dingliana, and Rachel McDonnell. Enlighten me: Importance of brightness and shadow for character emotion and appeal. *ACM Trans. Graph.*, 39(3), April 2020.
- [22] Su-Wei Yang, Zhao-Wei Qiu, and Ya-Shu Chen. Gpu swap-aware scheduler: Virtual memory management for gpu applications. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 1222–1227, New York, NY, USA, 2020. Association for Computing Machinery.
- [23] X. Yang, L. Zhou, H. Jiang, Z. Tang, Y. Wang, H. Bao, and G. Zhang. Mobile3drecon: Real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2020.
- [24] Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. Aranimator: In-situ character animation in mobile ar with user-defined motion gestures. *ACM Trans. Graph.*, 39(4), July 2020.