Towards Enhanced Visual Clarity of Sign Language Avatars Through Recreation of Fine Facial Detail

Ronan Johnson

Abstract Facial nonmanual signals and expressions convey critical linguistic and affective information in signed languages. However, the complexity of human facial anatomy has made the implementation of these movements a particular challenge in avatar research. Recent advances have improved the possible range of motion and expression. Because of this, we propose that an important next step is incorporating fine detail such as wrinkles to increase the visual clarity of these facial movements for the purposes of enhancing the legibility of avatar animation, particularly on small screens. This paper reviews research efforts to portray nonmanual signals via avatar technology and surveys extant illumination models for their suitability for this application. Based on this information, The American Sign Language Avatar Project at DePaul University has developed a new technique based on commercial visual effects paradigms for implementing realistic fine detail on the Paula avatar that functions within the complexity constraints of real-time sign language avatars.

Keywords sign language avatar · facial expression · face gesture · fine detail · wrinkles · nonmanuals

1 Introduction

Some of the most important aspects of signed languages are communicated through the face, yet it is one of the least studied areas by avatar developers. Much effort has been devoted to recreating lexical movements, particularly animating glosses and the transitions between them to generate increasingly complex utterances. However, the relative anatomical complexity of the face has proven a formidable hurdle in effectively developing the emotive and linguistic capabilities of sign language avatars. Recent progress has been promising, but we propose that artistic refinements are necessary to further advance the quality of facial animation portrayal. While not considered a linguistic feature, fine facial details provide redundant information to a signed message and contributes to the comprehension of signed utterances (Elliott et al. 2010). Of particular importance are the brows, lips, and eyes, which convey meaningful information that may be lost when displayed on a small screen. For example, the position of the eyebrows and eyelids is a necessary component distinguishing a yes/no question from a WH question (Bridges and Metzger, 1996).

Creating fine details in the face will emphasize subtle changes of expression, increasing visual clarity and legibility. Additionally, these fine details will increase the overall realism of avatar portrayal in accordance with our chosen realistic style.

The primary challenges of implementing such features are motivated by the constraints of the complexity and real-time nature of sign language avatars. Any extensions to existing frameworks should be as light and unobtrusive as possible, not only to avoid hampering the original system, but also to save room for other, perhaps unforeseen, additions in the future.

2 Previous Work

In reviewing the current literature, it is difficult to find mention of fine detail with relation to sign language avatar portrayal, let alone an implementation. Efforts that include descriptions and incorporation of facial expression include ViSiCAST (Elliott et al. 2004), ESign (Elliott et al. 2008), and Dicta-Sign (Efthimiou et al. 2010). These projects made major contributions to representing sign language by developing the notational system SiGML, a mark-up language designed to represent signing information in textual form for the purposes of driving an avatar's movements. One major consideration was the clear readability of generated language, including high quality rendering of facial expressions. In a Deaf user study, participants stressed the importance of facial expressions to comprehensibility (Elliott et al. 2000, 2004).

The JASigning project attempted increased realism by fitting their new avatar, Anna, to the exact biometric proportions of a human (Figure 1). While this is a solid step toward implementing realism, Anna's facial features often remain difficult to discern except in an extreme closeup (Elliott et al. 2010).

Fig. 1 The avatar Anna from the JASigning project (Ebling and Glauert, 2016).

VCom3D was an early effort from the commercial sector to develop an avatar and associated software for its animation. While they included some number of facial expressions, there have been no published advances since 2002 (Sims and Silverglate 2002).

In 2011, The DePaul University's avatar Paula was extended to include a facial implementation of non-manual signals. Their user study results found the addition of facial animations improved the clarity and comprehensibility of signed sentences, especially in questions for which facial movements are mandatory (Schnepp 2011).

One of the more promising efforts in the portrayal of facial gestures is the Huenerfauth and Kacorri modifications to the EMBR project (2015). Their work is built on a general-use avatar they adapted for sign language. Their most relevant advancement is an implementation of the forehead wrinkles visible during eyebrow movement. A more detailed discussion of their work is presented later in this paper.

Each of these projects faced difficulties communicating precision and clarity of facial expressions. This leads to user dissatisfaction with the facial capabilities of the current avatars (Kipp et al. 2011). Furthermore, these factors affect readability of nonmanual signals which can in turn interfere with an avatar's usability. One approach to improve precision and clarity is the additional rendering of fine detail. In particular, wrinkles are a promising starting point due to their visual prominence and role in communicating information on facial movements.

2.1 Clarity through facial wrinkles

It is desirable to display wrinkles because they add clarity to facial poses and emphasize the information conveyed by those poses. Facial wrinkles are folds and creases produced by the forced reshaping of skin and fat as adjacent muscles push on an area during a deformation (Osipa 2010). They are seen on the human face primarily through the shadows they cast when occluding a light source. These fine details provide visual information to the viewer to distinguish which muscles are compressing, how hard they are compressing, and in which direction. For example, the presence of creases in the corners of the eyes helps differentiate between gently and forcefully closed eye-lids. Such nuance can affect a viewer's ability to differentiate emotions or facial nonmanual signals, especially when viewed from a distance or on a small display. In general, more prominent wrinkles imply more force or intensity.

2.1.1 Avatar stylistic choices

Using a computer to create an avatar provides a wide variety of visualization options. Sign language can theoretically be recreated on any number of characters, provided they have sufficiently detailed anatomy with which to do so. Figure 2 shows an extreme example of this with the cartoon mouse character Quill from the virtual reality game "Moss." Quill uses some basic sign language to communicate with the player in lieu of voice acting (Lico 2018). The fact that she has hands with four fingers, opposable thumbs, and a range of arm motion approximating that of a human gives her the basic physical range of motion necessary to accomplish this.

Fig. 2 Quill from "Moss" asking the player for help using ASL (Lico 2018).

As a whole, cartoon-styled avatars have some distinct advantages. They do not need the kind of fine geometric detail necessary to realistically portray a human. Instead of indicating intensity and range with the presence of wrinkles, the animators push the facial poses to extremes in order to convey the same information. A character's eyebrows might raise much higher than is physically possible by a human, or a face might scrunch together so tightly, the eyes disappear. The more stylized an avatar looks, the less accurate their movements need to be in order to maintain an audience's suspension of disbelief. Exaggeration or "pushing the pose" is one of the fundamental principles of animation created by the artists at Walt Disney Animation Studios in the 1930s (Johnston and Thomas 1981). This is how animators are able to clearly and effectively communicate movement in highly stylized characters.

However, Deaf adults prefer realistic avatars to cartoon-styled avatars, even going so far as to discourage such exaggeration (Kipp et al. 2011). Because avatars are seen in the Deaf community primarily as a representative of an interpreter, it is most appropriate to use a realistic human model. However, an avatar approaching photorealism runs the risk of falling into the uncanny valley (Mori 1970). This happens when a human or human-like creature appears to be realistically plausible, but is subtly flawed in ways that humans find unnerving. Often, the uncanny valley becomes especially noticeable when the character is animated, especially when speaking and emoting. As social creatures, humans are particularly attuned to subtle discrepancies in the face. The film industry frequently encounters this problem when trying to create a computer-generated character that is indistinguishable from a real human. Even with the best technology, this remains a challenge. However, when a character looks unrealistic enough, the brain does not try to pick up on these details and instead, suspends its disbelief. When creating a realistic avatar, it is important not to push the realism too far in order to avoid creating an unintentionally disturbing character.

2.2 Increasing realism

There are two conventional methods for increasing the realism of a computer-generated image: enhancing the detail of a model with geometry and simulating the complex behavior of natural light. Both of these methods attempt to recreate the real world as closely as possible.

2.2.1 Modeling: Geometry

Modeling in computer graphics refers to the sculpting of a shape in the three-dimensional space by specifying the position of vertices and the edges that define their relationships with each other. These relationships are then visualized by connecting the edges into a group of polygons that define the basic shape of the final model. In general, we refer to this collection of polygons as geometry or a mesh (Hughes et al. 2014).

Because polygons are always flat, portraying uneven or detailed surfaces, including wrinkles, requires a fairly high-fidelity mesh. Complex detail requires a large number of polygons. Good modeling techniques for production keep this geometry localized to minimize overall the polygon count, meaning artists strive to increase the number of polygons in only the areas of the mesh where it is needed (Vaughn 2011). Nevertheless, defining even a rudimentary curve requires several polygons. Figure 3 shows the necessary increase in geometry to define smooth, naturalistic curves.

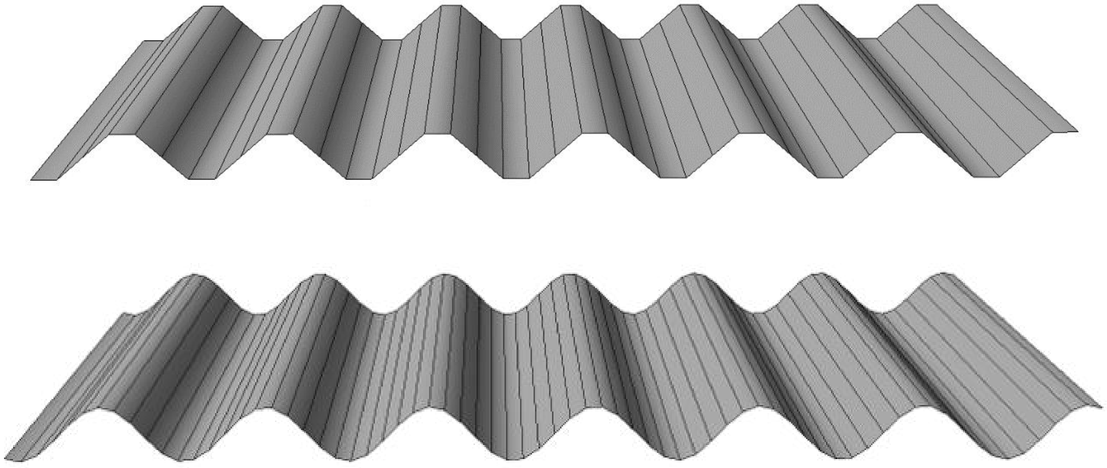


Fig. 3 Two planes warped into a sine wave. Minimal geometry results in harsh edges. Doubling the polygon count yields a more desirable shape at the cost of greater complexity.

2.2.2 Modeling: Morphs

Animating a complex mesh requires the transformation of many vertices. This is commonly accomplished using morphs, also known as blend shapes (Hughes et al. 2014). The process of creating a morph starts by copying the vertices of the original model. The vertices in the morph can then be moved to create a target shape. The computer interpolates the vertex positions from the original model to those of the target morph. Figure 4 demonstrates how the interpolation allows the morph to affect the original mesh to varying degrees. In this way, an artist can shape a morph target to represent the most extreme appearance of a set of wrinkles, and the intensity of the effect can be driven by some corresponding movement of the face. For example, as the lips pull towards the center of the mouth when making a pucker, a morph can automatically display fine detail of the skin folding into ridges as the space is compressed.

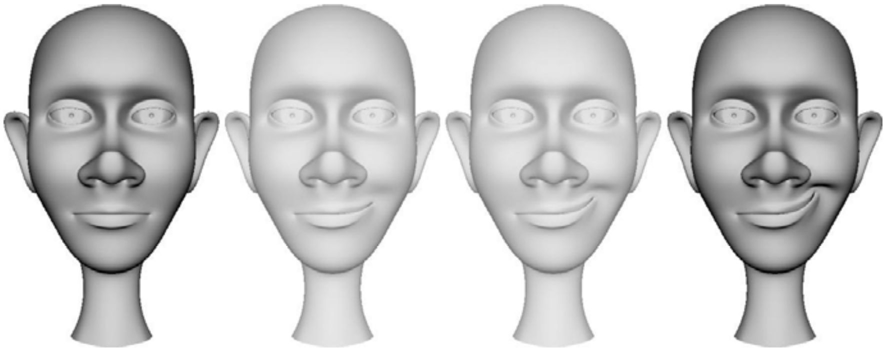


Fig. 4 A computer-generated head (left) with an accompanying morph (right). The middle images portray two potential states of interpolation between the two extremes.

Morphs are common in animation because they provide great artistic control over the exact shape of the final target. In sign language avatar research, they have been the primary mode for conveying non-manual gestures on the face (Huenerfauth and Kacorri 2015). They can also be combined to create complicated interactions between different parts of the face. Furthermore, because the interpolation considers only positional data, morphs are quick and easy to compute.

Figure 5 shows the progress Huenerfauth and Kacorri (2015) made in implementing wrinkle display as part of their work on the EMBR project. They modeled forehead wrinkles into the mesh using additional geometry, then controlled the deformation with a morph. The position of the avatar's eyebrows determines the intensity of the deformation.

Fig. 5 the EMBR avatar with visible forehead wrinkles (Huenerfauth and Kacorri 2015).

While they were successful in displaying forehead wrinkles, there are some notable drawbacks to their methods. For one, any amount of added geometry will slow down rendering. In order to animate a model, a computer must calculate the transformations of every vertex on the model for every frame. Because of this, geometry should be kept to a minimum for real-time applications. Even commercial animators working on high-end workstations will animate on a low fidelity mesh to optimize their workflow in real-time on a single machine. Rendering the high-fidelity version of this mesh often requires several hours and the use of a large-scale render farm to calculate the final image (Vaughan 2011).

While the simplicity and functionality of morphs make them a common choice when implementing facial gestures, they do come with some drawbacks. One is file size. Every time a new morph is added to a scene, a copy of the vertex information must be created and stored. This is plausible to implement for a subset of actions that would be difficult to portray otherwise, such as puffing out the cheeks, but employing this technique too frequently will make avatars unwieldy for applications on smaller devices. Another is the amount of time necessary to create a new desired shape. Each morph must be sculpted by a human artist to achieve proper results. Depending on the complexity of the underlying mesh and the number of existing morph targets to coordinate, adding a new target can be costly in both time and money.

Huenerfauth and Kacorri were also limited to using harsh overhead lights for the raised geometry to cast dark enough shadows that the wrinkles are visible in the final render. The clarity of the detail relies heavily on the orientation of the head relative to the light source. Reasonably common movements such as tilting the head upwards will eliminate the visual appearance of the fine detail. This limitation impedes generalization. In order to increase realism, choice of illumination model must now be considered.

2.3 Illumination models

In real life, a photograph is taken when photons from a light source bounce off an object, then enter a camera's lens and strike a light-sensitive medium. This leaves an impression of the scene on the medium. Similarly, in computer graphics, rendering refers to the color calculation of each pixel of a frame based on the interaction between the light sources; the location and orientation of the geometry; the location of the camera; and the shading and texturing information of that geometry. The final result is a two-dimensional representation of the three-dimensional scene. Illumination models define how these calculations are performed (Hall 1986). A model that uses physics to simulate the interaction of real light will produce more realistic renders. At face value, the most accurate model would seem to be the most appropriate choice, but technical limitations concerning their implementation must also be considered. Illumination models can be broken into three distinct categories:

- Local illumination
- Global illumination
- Semi-global illumination

2.3.1 Local illumination

The hallmark feature of local illumination models is their simplicity. One of the most widely-used of these is the Phong illumination model. First proposed in 1975 by Bui Tuong Phong, this model calculates the color of a pixel solely from the interaction between direct light sources and the surface properties of a point on any visible geometry (Phong 1975). This model accounts for only four such properties. Figure 6 displays the ambient, diffuse, and specular reflections. The fourth property is the surface normal, the vector perpendicular to the slope of any given polygon.

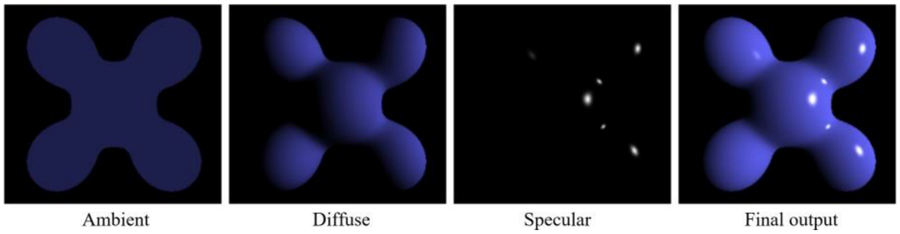


Fig. 6 The three visible components of the Phong illumination model and the three combined components in a final render. (Smith 2006)

2.3.2 Global illumination

Global illumination models are able to increase the realism of a render by simulating the real-life interactions between light and a surface. Ray tracing is an example of such a model. This method is implemented by tracing rays of light across a scene and calculating the material properties of any geometry they intersect. In Figure 7, we see that for every pixel on the camera plane, a vector (ray) is directed from the camera, through the pixel coordinates. If the ray intersects an object, an incident ray will be generated from the point of intersection at an angle perpendicular to the surface normal of the intersected geometry. The algorithm will continue in this way until the ray intersects the light source, or some predetermined number intersections is reached. The final rendered pixel color is an aggregate of the surface properties of the objects such as ambient color and reflectivity, combined with the characteristics of the light source such intensity and color (Appel 1968). By simulating the behavior of real light more closely, ray tracing is capable of rendering more obvious shadows than local illumination. The tradeoff is added time complexity that, until very recently, made it untenable for real-time applications, even on the most advanced hardware. Although there are ongoing efforts to make hardware more cost effective, the current expense makes this technique prohibitive for general real-time uses (Liu et al. 2019).

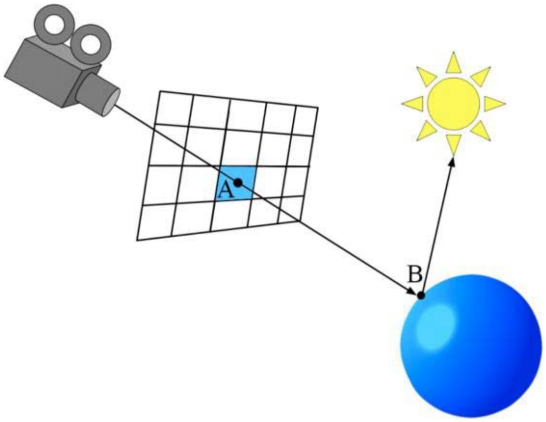


Fig. 7 In this graphic, a raytracing algorithm calculates the color of pixel A on the camera plane based on the surface properties of the ball at the intersected point B and the properties of the scene's light source.

Another global illumination model, radiosity, is capable of rendering photorealistic images by even more accurately simulating the behavior of real light. In this model, rays are cast from the light source, and allowed to bounce across the entire scene by generating incident rays at any points of intersection. The difference between this and raytracing is that the ambient color at the point of intersection is changed based on the properties of that ray's previous sequence of bounces. Because of this, the ambient component of a light in radiosity is not constant, but is the result of the interactions of light rays bouncing off every point of every surface having a diffuse reflection component. Unlike the constant number of light sources in Phong illumination and in ray tracing, with radiosity, any visible geometric point effectively becomes a light source. This yields two defining features in the final render: color bleeding and ambient occlusion. Color bleeding occurs when objects pick up color detail from other nearby, diffusely-reflecting objects (Cohen and Greenberg 1985). Ambient occlusion creates soft, subtle shadows resulting from the occlusion of some ambient light sources by adjacent geometry (Scherson and Caspary 1987). These shadows can highlight fine detail that might otherwise be difficult to discern. While radiosity provides the most realistic renders of these three models, as shown in Figure 10, the computational requirements are unattainable in most real-time applications.

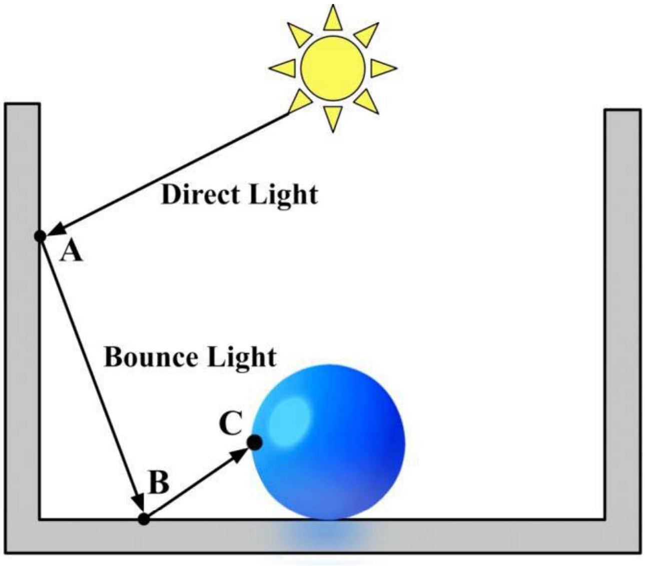


Fig. 8 An example series of rays used for the calculation of radiosity. The color of the ball at point C is influenced by the properties of the surrounding walls at points A and B, creating the property of non-constant ambient light. This results in the colors of the walls "bleeding" into the color of the sphere, and visa versa as seen on the floor below the sphere.

2.3.3 Semi-global illumination

While radiosity is computationally costly, an alternative method adapts semi-global illumination by using approximation techniques to achieve ambient occlusion without completing full radiosity calculations (Ritschel et al. 2009). As seen in Figure 9, an isolated, approximated ambient occlusion pass can be composited onto an image to enhance realism. Unfortunately, this too comes at a significant time penalty when compared to local illumination models. Real-time optimization remains under research (Jiménez et al. 2016).

Fig. 9 A sequence of renders. The first is without ambient occlusion. The second is the isolated approximate ambient occlusion shadows. The final render is the result of compositing the two together into a single image (Thewusa 2006).

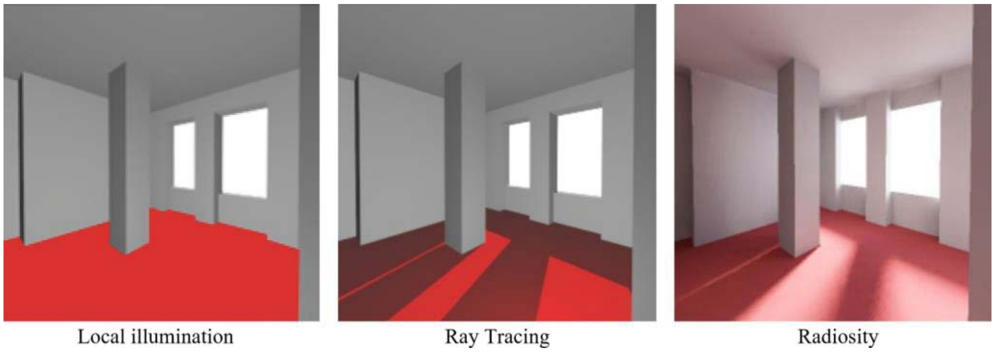


Fig. 10 A sample scene rendered with the three illumination models (Hugo 2006).

Considering the inefficiencies of high-fidelity illumination models, the implementation of finely detailed wrinkles using geometry is not fit for generalized real-time applications. Ideally, realism should be considerably enhanced without substantially reducing runtime. Fortunately, there is a better alternative for effective avatar technology.

3 A novel approach

A texture map is a digital image that can be applied to a model to give it the appearance of color, texture, and other visual properties. For example, DePaul University's avatar Paula uses a number of texture maps to define the look of her hair, eyes, and skin. Without such maps, she would appear simply as a gray mannequin. This technique provides a way to represent fine detail without physically modeling it in the geometry, such as mimicking the look of hair without modelling every individual strand.

The use of textures in facial wrinkle portrayal draws on knowledge from theatrical tradition. By exaggerating facial features through stage makeup, the expressions and emotions of the actors on stage are more legible to audience members, even ones in the back row of the furthest balcony (Corson et al. 2015). Following this example, we can improve facial readability by painting on the appearance of wrinkles and other fine details using a texture map.

Of course, this approach does not come without challenges. For one, the underlying software implementation of frequently swapping one map for another becomes cumbersome. Furthermore, simply replacing one map with another will cause an unnatural pop-in as there is no native interpolation between two maps in the same way there is with morphs. This would create very noticeable discontinuities when transitioning between different expressions. Moreover, replacing the entire texture map restricts the appearance of concurrent fine detail such as wrinkles in both the forehead and mouth if the avatar needs to perform some combination of movements such a puckering her lips while furrowing her eyebrows.

An initial solution proposed by (McDonald et al. 2016) attempted to fix this pop-in issue by layering facial textures on top of each other to form a stack, one with the appearance of wrinkling layered on top of a neutral texture as in Figure 11. Then, an algorithm would control the transparency of the top layer depending on the degree to which the eyebrows were raised, starting at 100% transparency (invisible) at a neutral brow position and going to 100% opacity when the brows were at their most extreme raised position. This technique allowed for smooth transitions between the appearance of the textures, taking into account the property that the wrinkles will appear more intense as the brows are raised further. However, layering textures of the entire face inhibits the display of co-occurring fine details or occludes them altogether. Any details present in the middle of the stack will appear washed out when the opacity of the layers above it is increased, eventually being obliterated entirely. Furthermore, most of the facial texture is unchanged between each version, creating redundancies, and therefore unnecessarily increasing storage requirements.

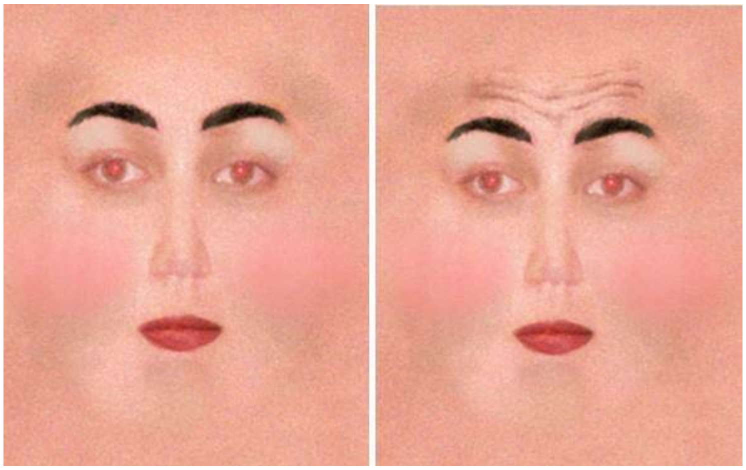
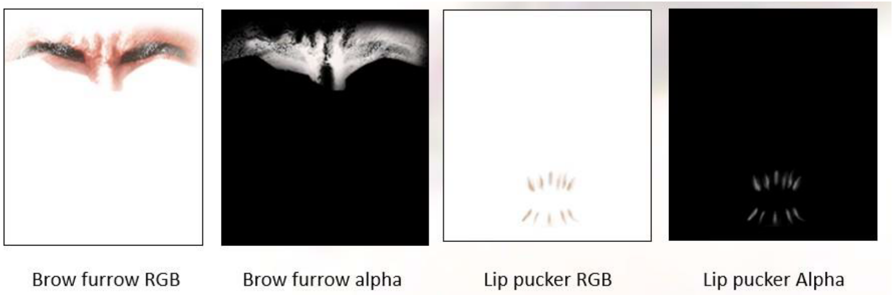


Fig. 11 A neutral face texture and a wrinkled forehead face texture.

In response to these limitations, we extended this technique by isolating areas of fine detail by incorporating an alpha channel into the texture, giving it an element of transparency. Instead of each layer containing color information for the entire face, the texture portraying the fine detail can be mostly transparent, combining with color information present only in the goal position of the desired detail, demonstrated in Figure 12. The same method of increasing and decreasing the opacity of a given layer based on some facial movement such as brow position is preserved. The end result is a series of layered textures that can be concurrently displayed without interference to allow for the appearance of co-occurring fine detail. Figure 13 demonstrates an example case of the wrinkles that form when the eyebrows are furrowed. An additional benefit is that the appearance of each detail can be controlled independently. This is of critical importance when animating complex facial gestures with different timing between different parts of the face. The relatively simple calculation of opacity needs to occur only once per frame, greatly reducing computational overhead as contrasted with the mathematical intensity of geometric deformations.



Brow furrow RGB

Brow furrow alpha

Lip pucker RGB

Lip pucker Alpha

Fig. 12 Color and alpha channels for two face texture maps displaying only localized detail.

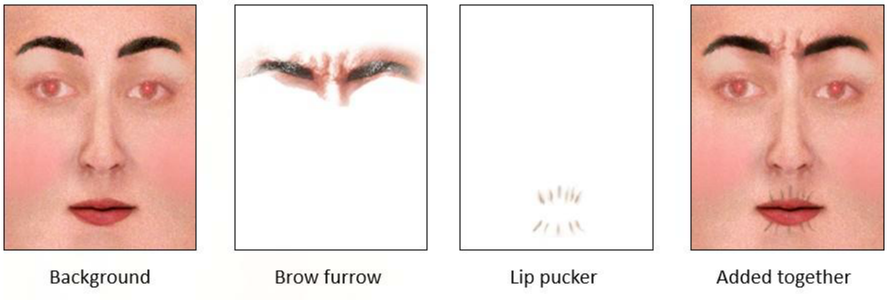


Fig. 13 The result of layering localized textures onto the base facial texture map.

4 Results

Taking into account all of these considerations, a combination of techniques can provide a satisfactory trade-off between clarity, speed, and realism. Ideally, an avatar should run smoothly on many different devices with potentially limited hardware and storage capacity. Considering the potential intent to incorporate avatars into web applications (Kennaway et al. 2007), it is necessary to prioritize speed with minimal compromise in realistic appearance. In light of this, a combination of layered texture maps with masking alpha channels and the Phong local illumination model is the most efficient combination for portraying fine detail.

The addition of a new texture layer does not impact final render times. The texture maps merely provide color information which is easily sampled by the computer and factored into the final calculation of the rendered pixel. Because the soft shadows typically achieved by radiosity can be represented in this color information instead of being calculated in real-time, render complexity is reduced. This also avoids the need to add any extra geometry to generate those shadows, allowing for high quality animations to be generated in real time. Additionally, the model's head orientation with respect to the light source becomes less critical. Combining this texture information with local illumination saves time and reduces complexity without sacrificing comparable visual results to the more taxing illumination models.

The video accompanying this paper, available at http://sltat.cs.depaul.edu/2019/johnson.mp4, demonstrates the merit of this approach. In the animation, the avatar Paula produces a surprised expression. The forehead wrinkles seen on the right are generated through a texture map such as the ones in Figure 13. Their inclusion emphasizes the upward movement of the eyebrows, improving the readability of the expression compared to the head in the middle which uses only the simple base facial texture. The left-most head shows the underlying geometry in motion. Of particular note is the relatively few polygons that define the forehead. Creating the appearance of wrinkles through a texture map gives the avatar the desired visual fidelity while maintaining the low polygon count required for real-time applications.

Additional benefits include extendibility and storage requirements. Incorporating another section of fine detail requires only the creation of a new masked layer for an isolated part of the face. Only one underlying face texture is required for the baseline appearance. This reduces the amount of time an artist must spend to extend and improve the avatar's visual fidelity, compared to the process of adding geometry, sculpting the corresponding morph, and then incorporating it into the existing face. The necessary increase in file size with each addition is minimal compared to the dozen or so morphs needed for comparable results with extra geometry.

5 Conclusion and future work

Rendering clear facial expression is a complex undertaking. Given the limitations imposed by real time rendering, we have been able to adapt new solutions to maintain optimal performance without sacrificing our ability to significantly improve the realism and comprehensibility of our avatar. We are confident moving forward that our methods will scale with the increasing scope of our research. We expect further improvements will use these concepts as a foundation.

Future work includes the incorporation of further fine details, especially around the mouth. We expect our improvements will increase not only the results of user studies, but also the quality of feedback we receive on our avatar's appearance. With further realism, disconcerting facial animation will no longer cause a distraction during user studies and will increase the accuracy of results. Feedback from the Deaf community will continue to guide this project as it continues to improve. User studies to measure the effectiveness of various facial details are forthcoming.

References

Appel A (1968) Some techniques for shading machine renderings of solids. AFIPS 1968, Proceedings of the Spring Joint Computer Conference. 32:37-45. https://doi.org/10.1145/1468075.1468082

Bridges B, Metzger M (1996) Deaf tend your non-manual signals in ASL. Silver Spring, MD: Calliope Press

Cohen M, Greenberg D (1985) The hemi-cube: A radiosity solution for complex environments. ACM Siggraph Computer Graphics. 19(3):31-40. https://doi.org/10.1145/325165.325171

Corson R, Glavan J, Norcross B.G. (2015) Stage Makeup, 10th edn. Pearson Canada

Ebling S, Glauert J (2016) Building a Swiss German Sign Language avatar with JASigning and evaluating it among the Deaf community. Universal Access in the Information Society. 15(4):577-587.

Efthimiou E, Fontinea S.E., Hanke T, Glauert J, Bowden R, Braffort A, Goudenove F (2010) Dicta-sign sign language recognition, generation, and modelling: a research effort with application in Deaf communication. Proceeding of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies. pp 80-83

Elliott R, Bueno J, Kennaway J, Glauert J (2010) Towards the integration of synthetic sign language animation with avatars into corpus annotation tools. 4th workshop on the representation and Processing of Sign Languages: Corpora and Sign Language Technologies. Valletta, Malta. pp 29

Elliott R, Glauert J, Jennings V, Kennaway, JR (2004) An Overview of the SiGML Notation and SiGML Signing Software System. Sign language processing satellite workshop of the fourth international conference on language resources and evaluation, LREC. 98-104

Elliott R, Glauert J, Kennaway J, Marshall I, Safar E (2008) Linguistic modelling and language-processing technologies for Avatar-based sign language. In Universal Access in the Information Society 6(4):375-391. https://dx.doi.org/10.1007/s10209-007-0102-z

Elliott R, Glauert J, Kennaway J, Marshall I (2000) The Development of Language Processing Support for the ViSiCAST Project. In: Fourth Annual ACM Conference on Assistive Technologies, 2000, pp 101-108. https://doi.org/10.1145/354324.354349

Hall R (1986). A characterization of illumination models and shading techniques. The Visual Computer 2:268–277. https://doi.org/10.1007/BF02020427

Huenerfauth M, Kacorri H (2015) Augmenting EMBR virtual human animation system with MPEG-4 controls for producing ASL facial expressions. International Symposium on Sign Language Translation and Avatar Technology, volume 3.

Hughes JF, Van Dam A, Foley JD, Mcguire M, Feiner SK, Sklar, DF (2014) Computer Graphics: Principles and Practice. Pearson Education.

Hugo E (2006). Difference between standard direct illumination without shadow umbra, and radiosity with shadow umbra. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Radiosity_Comparison.jpg

Jiménez, J., Xianchun, W., Pesce, A., and Jarabo, A. (2016) Practical real-time strategies for accurate indirect occlusion. SIGGRAPH 2016 Courses: Physically Based Shading in Theory and Practice.

Johnston O, Thomas F (1981). The Illusion of Life: Disney Animation. Disney Editions New York.

Kennaway JR, Glauert JRW, Zwitserlood I (2007) Providing signed content on the Internet by synthesized animation. ACM Transactions on Computer-Human Interaction, 14(3):15. https://doi.org/10.1145/1279700.1279705

Kipp M, Nguyen Q, Heloir A, Matthes S (2011) Assessing the deaf user perspective on sign language avatars. The proceeding of the 13th international ACM SIGACCESS conference on Computers and accessibility. pp 107-114. https://doi.org/10.1145/2049536.2049557

Lico R (2018) Topics in real-time animation. ACM Siggraph 2018 courses. 17:1. https://doi.org/10.1145/3214834.3214882

Liu E, Llamas I, Cañada J, Kelly P (2019). Cinematic rendering in UE4 with real-time ray tracing and denoising. Ray Tracing Gems, pp 289-319

McDonald J, Wolfe R, Schnepp J, Hochgesang J, Jamrozik DG, Stumbo M, Berke L, Bialek M, Thomas F (2016). An automated technique for real-time production of lifelike animations of American Sign Language. Universal Access in the Information Society 15(4):551-566

Mori M (1970) The Uncanny Valley. IEEE Robotics & Automation Magazine 19(2):98-100 https://doi.org/10.1109/MRA.2012.2192811

Osipa, Jason (2010) Stop Staring: Facial Modeling and Animation Done Right, 3rd edn. Sybex.

Phong BT (1975) Illumination for computer generated pictures. Communications of the ACM. 18(6):311-317. https://doi.org/10.1145/360825.360839

Ritschel T, Grosch T, Seidel H (2009) Approximating dynamic global illumination in image space. Proceedings of the 2009 symposium on Interactive 3D graphics and games. pp 75-82. https://doi.org/10.1145/1507149.1507161

Scherson ID, Casparay E (1987) Data structures and the time complexity of ray tracing. The Visual Computer 3(4):201-213. https://doi.org/10.1007/BF01952827

Schnepp J (2011) A representation of selected nonmanual signals in American sign language. Dissertation, DePaul University

Sims E, Silverglate D (2002) Interactive 3D characters for web-based learning and accessibility. ACM SIGGRAPH 2002 conference abstracts and applications, pp 314. https://doi.org/10.1145/1242073.1242322

Smith B (2006) Phong components version 4. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Phong_components_version_4.png

Thewusa (2006) AmbientOcclusion German. Wikimedia Commons. https://en.m.wikipedia.org/wiki/File:AmbientOcclusion_German.jpg

Vaughan W (2011) Digital Modeling. New Riders.