

Augmented Reality Interfaces for Additive Manufacturing

Eiríksson, Eyþór Rúnar; Pedersen, David Bue; Frisvad, Jeppe Revall; Skovmand, Linda; Heun, Valentin; Maes, Pattie; Aanæs, Henrik

Published in:
Scandinavian Conference on Image Analysis

Link to article, DOI:
[10.1007/978-3-319-59126-1_43](https://doi.org/10.1007/978-3-319-59126-1_43)

Publication date:
2017

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Eiríksson, E. R., Pedersen, D. B., Frisvad, J. R., Skovmand, L., Heun, V., Maes, P., & Aanæs, H. (2017). Augmented Reality Interfaces for Additive Manufacturing. In Scandinavian Conference on Image Analysis (pp. 515-525). Springer. (Lecture Notes in Computer Science, Vol. 10269). DOI: 10.1007/978-3-319-59126-1_43

DTU Library

Technical Information Center of Denmark


General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Augmented Reality Interfaces for Additive Manufacturing

Eythor R. Eiriksson¹, , David B. Pedersen¹, Jeppe R. Frisvad¹, Linda Skovmand¹, Valentin Heun², Pattie Maes², and Henrik Aanæs¹

¹ Technical University of Denmark, Kgs. Lyngby, Denmark,

² MIT Media Lab, Cambridge MA, USA.

eruei@dtu.dk

Abstract. This paper explores potential use cases for using augmented reality (AR) as a tool to operate industrial machines. As a baseline we use an additive manufacturing system, more commonly known as a 3D printer. We implement novel augmented interfaces and controls using readily available open source frameworks and low cost hardware. Our results show that the technology enables richer and more intuitive printer control and performance monitoring than currently available on the market. Therefore, there is a great deal of potential for these types of technologies in future digital factories.

Keywords: 3D Printing, Additive Manufacturing, Augmented Reality

1 Introduction

With recent efforts in industrial digitalization, commonly referred to as 'industry 4.0', the core aim is to realize the factory of the future. These factories are envisioned to be agile and flexible using complex autonomous manufacturing technologies combined with the human skills of reasoning. As the manufacturing technology becomes more and more autonomous the need for intuitive and fluid human-machine interaction becomes necessary. This is in part due to the massive increase in data available to the users. Additionally, it is due to increasing abstraction where complex inter-connectivity between factory elements has become invisible to the operators.

A promising, highly digitized and automated manufacturing technology that frequently comes up in relation to the factory of the future, is additive manufacturing (AM), commonly known as 3D printing. For large scale manufacturing using not only several AM systems but also other highly automated machine tools, the need for a larger control framework is required. Managing multiple machinery as well as monitoring their performance often cannot easily be achieved through their conventional physical user interfaces. However, these interfaces can be dynamically scaled in an augmented reality control interface. With such an interface, it can be evolved and iteratively tailored towards the specific use cases at any given time. As an example, multiple machines can be controlled using a single AR user interface.

This is the authors' version of the work.

The final publication is available at link.springer.com

(c) Springer International Publishing AG 2017

P. Sharma and F.M. Bianchi (Eds.): SCIA 2017, Part I, LNCS 10269, pp. 515–525, 2017.

http://doi.org/10.1007/978-3-319-59126-1_43

In this paper, we focus on the integration of an augmented reality based human-machine interface that substitutes a conventional user interface on an additive manufacturing (AM) machine tool. This is achieved using readily available open source frameworks, web technologies and low cost mobile hardware.

2 Related Work

Mobile embedded systems such as smartphones, tablets and head mounted displays (HMD) [1, 2] now have enough computing power to enable augmented reality applications at a low cost. In an industrial scenario, applications such as remote technical support [3], worker training [4] and design [5], can be achieved.

In the context of AR industrial machine control and visualization, Zhang et al. [6] performed augmented cutting simulations on a industrial milling machine, as well as provided an interface control panel using a camera and high end processing computer. Olwal et al. demonstrated an industrial AR projection system that augments machine information onto a transparent display [7]. Both examples above, classify as tethered solutions that do not offer the flexibility achievable with a mobile device. Kollatsch et al. [8] presented a mobile implementation using a Windows tablet PC that displayed machine status parameters when presented with an AR marker, however no machine control was possible.

Our method takes advantage of recent advancement in high performance mobile technology that enables use of both consumer tablets and smartphones. Using open source systems we have implemented bi-directional AR interfaces that both monitor and control a 3D printer, in such a way that any changes made on the printer from external sources are reflected in the AR interface and vice versa.

3 Method

As consumer mobile tablet devices have already made their way into the industrial environment, we use a traditional iPad Mini 3 for this study. It is responsible for both displaying augmented interfaces as well as performing the computer vision necessary to identify and pose estimate unique optical markers that are placed on relevant locations on the machine. For this study, we employ the Ultimaker Original [9] desktop 3D printer to represent an industrial AM machine tool on a factory floor. It is connected to a laptop via a USB-serial interface, where the control and readout takes place. For this implementation, any device capable of running the `node.js` JavaScript runtime environment [10] could be used to handle the printer communication. This includes popular devices such as the Raspberry Pi, Arduino Yún as well as many industry standard embedded systems built on ARM technology. Thus it is entirely feasible for several machines to be equipped with AR functionality at a very low cost. Both the tablet, and laptop, are connected to a common wireless network (WIFI). The laptop runs a `node.js` web server that communicates to both tablet and printer. See Figure 1.

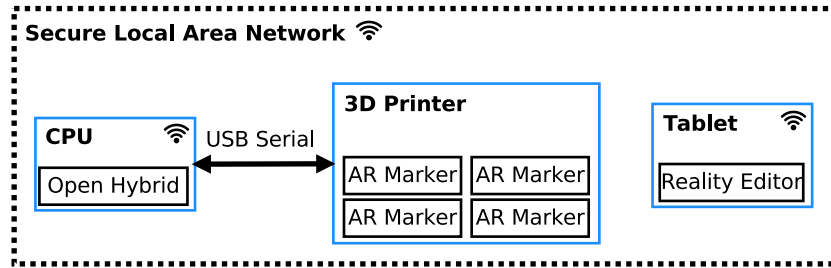


Fig. 1. Diagram illustrating the hardware and software components used.

3.1 Additive Manufacturing Machine Tool

The Ultimaker Original is entirely open source and belongs to the class of extrusion based printers. Its construction is cartesian with a motorized horizontal X-Y carriage system. It is used to trace 2D cross-sectional slices of 3D objects, whilst simultaneously extruding melted thermoplastic material through a heated extruder nozzle. The vertical Z-axis, that translates the build plate, is then moved downward. Typically in sub-mm increments for each new layer to be produced. The printer belongs to a large family of numerically controlled (NC) devices following the G-code standard [11]. It is possible to control the machine tool through its USB hardware interfaces which presents itself as a standard serial port when connected to a computer. The printer can then be operated using G-code commands which are conventionally used in computer-aided manufacturing to control automated technologies (CAx). It is thus possible to set and query all features of the printer ranging from motion control to tooling settings.

3.2 Open Hybrid

Open hybrid [12] is a highly customizable open source platform from the MIT Media Lab, that enables flexible interactions between physical objects through augmented user interfaces. Objects or systems powered by Open Hybrid enable the possibility of inter-connectivity between each one. Such that the underlying input- and output connectivity routing between objects can be dynamically altered though simple drag and drop gestures. The augmentation system is enabled by web standards, and as such each object is set up solely in standardized protocols through a simple Application Programming Interface (API).

3.3 Reality Editor

The Reality Editor [13] is a publicly available mobile iOS app that functions as a digital window and multi-tool for Open Hybrid objects. Allowing the user to connect and manipulate the functional behavior of any physical object that is enabled by the Open Hybrid platform. As a mobile device (tablet or phone) is pointed toward an object AR marker, the Reality Editor recognizes it and

performs camera pose estimation in real time. It then proceeds to augment and display the corresponding user interface on the mobile device with the correct perspective transformation. Physical buttons, indicator lights and sensors may be shown as nodes between which relationships can be defined through gestures. Once configured, the object will retain its state until the mobile tool is used again to alter the previously defined relations.

3.4 Object Markers

Throughout this study we use a novel type of quick response (QR) codes called Human Readable Quick Response Codes [14] (HRQR). The marker consist of high contrast visual features that allow for robust marker detection and pose estimation. Furthermore, as the marker is readable by humans, its text will give an indication to what type of interface is to be expected.

4 Use Cases

The following sections describe use cases implemented in this study. We identified these cases based on common interactions for this type of printer and how they might be improved using AR.

4.1 Print Visualization

In a manufacturing setting, the ability to see what object is being produced per machine can be beneficial. Giving the user a overview on the current status on the production floor. Such functionality can be provided by AR interface. Currently, there are two ways of identifying the object under print. One is by visual inspection of the actual part and secondly by its filename, which is usually displayed on the printers graphical display. For the latter, special care must be taken in naming the files accordingly, especially when dealing with different versions of the same part. Throughout a print process it can be unclear what model is being printed. This also holds true when parts are being printed that vary only slightly in form. That being said, there could be cases where it is non-trivial what part is being printed. To address the above, we implemented a use case where the entire print bed consists of an AR marker. When the marker is observed using the Reality Editor, the 3D model being printed is augmented on the print platform itself. As the AR interface supports standard web technologies, we use the `three.js` Javascript 3D library for loading and rendering the mesh geometry. Camera pose estimation is obtained each frame by observing the AR marker, and the mesh is perspectively transformed accordingly. See Figure 2.

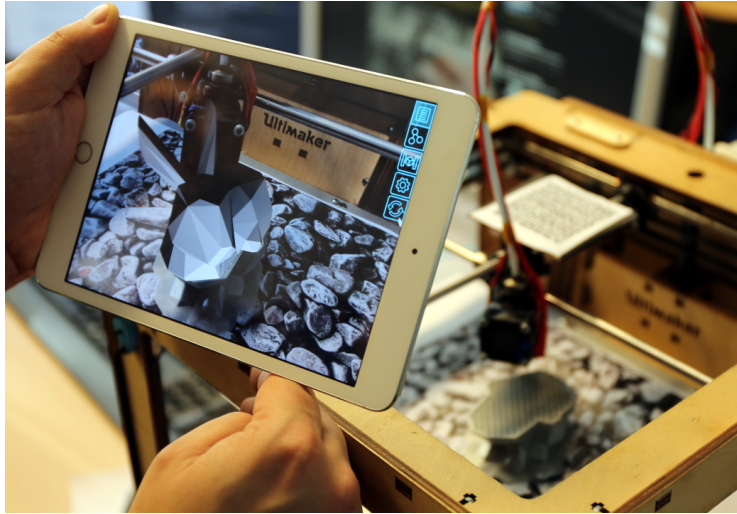


Fig. 2. Print visualization example. A 3D model (Bunny) is augmented whilst printing.

In line with our previous work on pre-visualization of printed parts prior to printing [15, 16], we here add the staircase effect often observed in layered manufacturing [17] to visualize for the user the surface structure that the printer will produce. We render the triangle mesh using a pixel shader with standard Phong lighting [18] and shadow mapping [19]. We customize our shader by adding procedural normal mapping. Having the Z -axis as the printing direction, our procedure for normal mapping is to find the layer index i of the surface fragment observed in the pixel, calculate the distance t to the intersection of the eye ray with the next layer, and use this to modify the Z -coordinate of the normal. In mathematical terms,

$$i = \left\lfloor \frac{p_z}{\ell} \right\rfloor + H(\omega_z) \quad , \quad t = \frac{i \ell - p_z}{\omega_z} \quad , \quad \mathbf{n}^* = (n_x, n_y, H(\ell - t) \text{sign}(n_z))$$

where \mathbf{n} is the unit length surface normal, \mathbf{p} is the world space fragment position, $\boldsymbol{\omega}$ is the unit length direction vector from the camera position toward \mathbf{p} , ℓ is the layer thickness, $H(x)$ is the Heaviside step function (which is 1 for $x \geq 0$ and 0 otherwise), and \mathbf{n}^* is the modified normal that produces the staircase effect. See Figure 3 for an example.

The sharp edges in our staircase normal mapping produce aliasing artifacts in excess. We anti-alias our renderings by sampling $N = 50$ positions \mathbf{p} in an area of 2 by 2 pixels around the original fragment position. We use a linear congruential generator with the same seed in all pixels to generate a pseudo-random distribution of window space positions without expensive texture look-ups. Still in the pixel shader, we invert the graphics pipeline transformations to get the world space coordinates of the sampled positions. For every sampled position, we then compute the normal using the given procedure and evaluate the lighting.

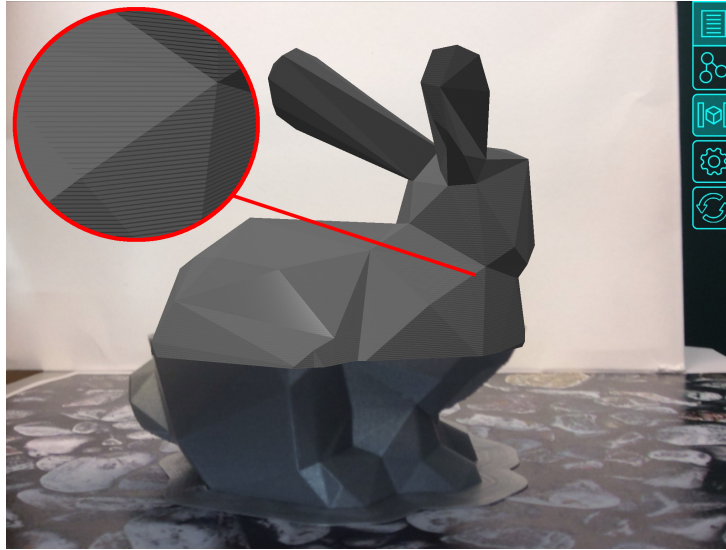


Fig. 3. Reality Editor screenshot showing model augmentation using the staircase effect shader. Due to high performance AR marker tracking, obscured viewing angles such as this can be achieved. Note: Only the part of the model that is missing is augmented.

The final pixel value is the mean of these N lighting evaluations. By reducing the number of samples ($N = 25$), we can trade render quality for performance. Very thin layers (small ℓ) introduce Moiré patterns in the renderings.

4.2 Nozzle Thermal Control

Thermal control of the nozzle that extrudes liquid plastic, is one of the most frequently operated settings on the Ultimaker. Currently, users have to perform five menu operations to reach this setting. We propose to tie the relevant parameters to a simple AR interface that is accessed through an AR marker on the print tool-head. There, the user can instantaneously monitor the current temperature, as well as easily change the set temperature through an intuitive slider. Figure 4 shows the interface under use.

4.3 3D Model Selection

The work flow for setting up a print job is conventionally by pre-processing a 3D geometry in a standalone software that generates a G-code print job. However, in a manufacturing setting, more often than not, the same set of models will be printed repeatedly. For most printers, these job files are stored on an on-board memory but cannot be pre-visualized or identified, other than by its filename. To improve upon this, we propose a print job selector, that allows the user to preview available models and select which should be printed. Allowing for very rapid job selections. The interface may be seen in Figure 5.



Fig. 4. A minimal augmented interface showing the target and current temperatures of the heated print nozzle. Additionally a slider is included for easy manual operation.

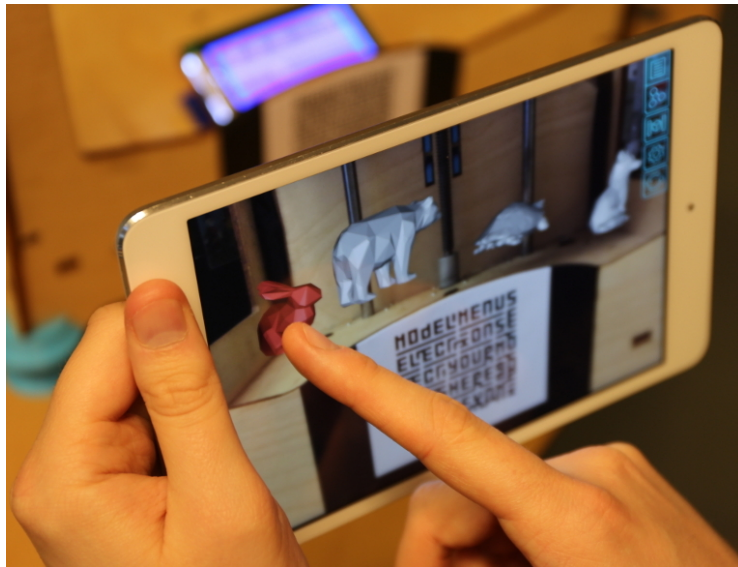


Fig. 5. Our implemented selection interface for easy selection of parts to be printed. To select a job, the user simply presses on the responding augmented model.

4.4 Carriage Motion Control

Frequently, the X-Y carriage of the printer is moved for maintenance. To do this, only one axis can be jogged at a time through a cumbersome menu system on the printers current interface. For some systems, the implementation is non-direct. Then by pressing a button, the axis starts by moving slowly. However, the speed of the axis increases the longer the button is pressed. This non-direct implementation results in less controlled motion by the user. We have implemented direct augmented control of the carriage that allows it to be moved using the AR interface. Enabling a more precise motion control tool that follows the speed of the user. The implementation can be seen in Figure 6. When a user touches the augmented joystick, we record the markers (u, v) screen coordinates on the display device. Then the user can proceed to pan the device, which updates the markers coordinates on the screen. This causes a disparity which is used as a control signal for the X and Y motors. The greater the disparity, the faster the carriage will move. In practice, the absolute scale of the AR marker could be used as control for the Z-axis. This Z-axis control was however not implemented.

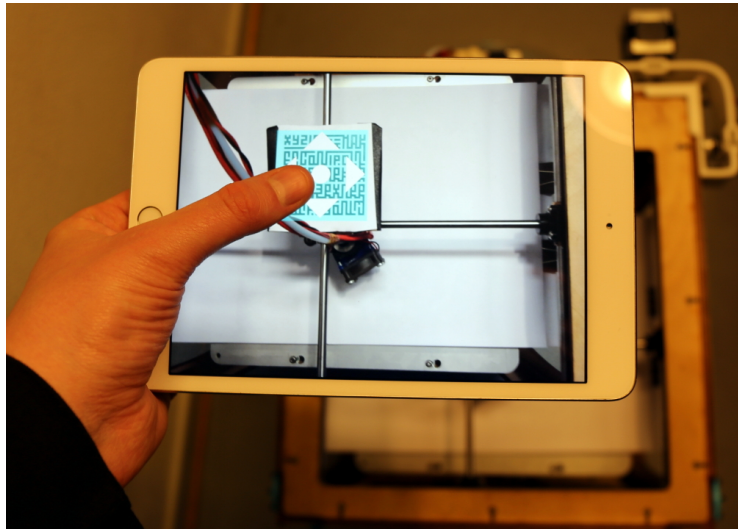


Fig. 6. Example interaction of intuitive carriage motion control.

5 Future Work

For future interfaces we envision the following:

Extrusion control that enables users to change plastic feed-rates, as well as the initiation of extrusion.

Model placement on the build plate to be used to position one or more 3D objects in an intuitive fashion. Object collision or out of bounds events may be visualized by changing the objects color. Additionally, dimensional scaling of the geometries could be performed through common zoom pinching gestures.

Printer-to-printer interactions are possible as more printers are enabled by the Open Hybrid platform. This could allow for 'drag and drop' of 3D models between printers such that high demand geometry easily can be placed in a print queue. This allows for agile production ramp up of parts. An implementation could allow various types of printer technologies to interact, as the de-facto standard fileformat is based upon wireframe meshes.

Drag and dropping of models directly from community driven services such as thingiverse.com could be made possible. The back-end AR server would then be responsible for downloading the STL geometry, slicing the model and generating the machine code.

Layer by layer visualization could be implemented, where tool planning and support structures are shown. In addition, model areas lacking support structures could be highlighted.

Drawing shapes on the build plate, as well as simple sculpting operations can be supported. Drawing modes could for example include manual drawing of support structures.

6 Discussion and Conclusion

In this paper, we have explored and touched upon possible use cases for augmented reality systems in a manufacturing environment. This was done as an effort to explore how augmented reality can be used as a seamless interface for the factory of the future. In this study, we have successfully integrated state of the art and open source platforms that have been developed at the MIT Media Lab. In this spirit, the implementation has been realized on the next-gen manufacturing method of 3D printing. We demonstrate novel use cases in which augmented reality may outperform conventional control mechanism in such systems. However, it is important to note the presented augmented reality interfaces are not limited to this specific family of manufacturing technologies. In fact, similar augmented interfaces can be implemented on any kind of manufacturing tools and

machinery such as CNC machine tools, water jet- and laser cutters, injection moulding machines, forging presses etc. Any modern conventional manufacturing process chain can readily be set up as an augmented object and monitored and/or controlled through means presented in this paper.

With the rapidly increasing amounts of data produced by modern machine tools, AR can serve as a medium that provides operators with context related data on demand. Thus bridging a gap that may be created as content complexity increases. The scalability enabled by AR interfaces has the potential to be an enabling technology for modular production platforms, rapid product development and hyperflexible automation. By using constantly evolving bi-directional interfaces and the ability to instantaneously switch between them, it is possible to rapidly adapt to changes in production, bring products to market faster and small series production platforms can be set up quickly.

With few limits and near endless possibilities we have demonstrated the importance of embracing augmented reality in manufacturing engineering such that advanced manufacturing processes and process chains can be interfaced in a simple and intuitive manner.

References

1. "Microsoft HoloLens." <https://www.microsoft.com/microsoft-hololens/en-us>. Accessed: 2017-01-05.
2. "DAQRI Smart Helmet." <https://daqri.com/products/smart-helmet/>. Accessed: 2017-01-05.
3. M. Billinghurst and H. Kato, "Collaborative augmented reality," *Communications of the ACM*, vol. 45, no. 7, pp. 64–70, 2002.
4. S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, "An augmented reality training platform for assembly and maintenance skills," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 398–403, 2013.
5. A. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis, "Augmented reality applications in design and manufacturing," *CIRP Annals-Manufacturing Technology*, vol. 61, no. 2, pp. 657–679, 2012.
6. J. Zhang, S. Ong, and A. Nee, "A volumetric model-based CNC simulation and monitoring system in augmented environments," in *Cyberworlds, 2006. CW'06. International Conference on*, pp. 33–42, IEEE, 2006.
7. A. Olwal, J. Gustafsson, and C. Lindfors, "Spatial augmented reality on industrial CNC-machines," in *Electronic Imaging 2008*, International Society for Optics and Photonics, 2008.
8. C. Kollatsch, M. Schumann, P. Klimant, V. Wittstock, and M. Putz, "Mobile augmented reality based monitoring of assembly lines," *Procedia CIRP*, vol. 23, pp. 246–251, 2014.
9. "Ultimaker Original, BOM and parts." github.com/Ultimaker/UltimakerOriginal. Accessed: 2017-01-05.
10. "Node.js." <https://nodejs.org/en/>. Accessed: 2017-01-09.
11. "ISO 6983-1:2009. Automation systems and integration – Numerical control of machines – Program format and definitions of address words – Part 1: Data format for positioning, line motion and contouring control systems.," standard, International Organization for Standardization, Geneva, CH, Dec. 2009.

12. “Open Hybrid: Platform for interaction with everyday objects.” <http://openhybrid.org/>. Accessed: 2017-01-05.
13. V. Heun, J. Hobin, and P. Maes, “Reality editor: Programming smarter objects,” in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 307–310, ACM, 2013.
14. “HRQR: Human Readable Quick Response Code.” <http://hrqr.org/>. Accessed: 2017-01-05.
15. E. R. Eiriksson, D. B. Pedersen, and H. Aanaes, “Predicting color output of additive manufactured parts,” in *ASPE 2015 Spring Topical Meeting*, pp. 95–99, 2015.
16. E. R. Eiriksson, A. Luongo, J. R. Frisvad, D. B. Pedersen, and H. Aanaes, “Designing for color in additive manufacturing,” in *Proceedings of the Aspe/euspen 2016 Summer Topical Meeting on Dimensional Accuracy and Surface Finish in Additive Manufacturing*, ASPE–The American Society for Precision Engineering, 2016.
17. Y. He, G. huai Xue, and J. zhong Fu, “Fabrication of low cost soft tissue prostheses with the desktop 3D printer,” *Scientific reports*, vol. 4, no. 6973, 2014.
18. B. T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, pp. 311–317, June 1975.
19. L. Williams, “Casting curved shadows on curved surfaces,” *Computer Graphics (Proceedings of SIGGRAPH 78)*, vol. 12, pp. 270–274, August 1978.