PhysLights: a Tangible User Interface for CG Lighting

Kyung yul Kevin Lim
Department of CSE
UC San Diego
kyungyullim@ucsd.edu

Bryan Binotti
Department of ECE
UC San Diego
bbinotti@gmail.com

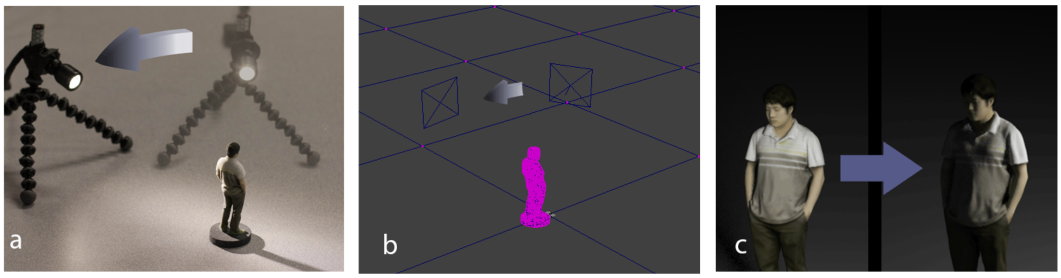

Figure 1: Physical spot lights and a 3D printed character are used to represent elements in a virtual environment. Moving the physical spot lights (a) result in the virtual lights moving (b), and the lighting render changing (c).

ABSTRACT

We present a novel approach to CG lighting named *PhysLights*. *PhysLights* enables lighting artists to execute their vision and artistic choices in the physical space through a Tangible User Interface. This lighting is then represented in a conventional 3D package and rendered. Through *PhysLights* we explore the advantages and disadvantages of a Tangible User Interface in CG Lighting workflows, and the extent to which it can augment or replace traditional pipelines. *PhysLights* is part of our vision to make both new and existing CG Animation production more approachable, and more collaborative.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Tangible Interace, Animation, Lighting

INTRODUCTION

With the amount of releases of 3D animated films each year, interest in 3D animation production seems to be at an all-time high. Further, we observe improvements in sensing technology and more affordable 3D printing [7]. These trends lead us to envision a future where many audiences of CG animation aspire to find easier ways to become producers themselves, and part of the future in CG animation pipeline will involve physical representations and a Tangible User Interface (TUI) to manipulate them. In this paper, we explore a possibility for this vision using mainly off-the-shelf solutions already affordable today. Specifically, we present *PhysLights*, a TUI for CG lighting as a slice into what a tangible production pipeline might look like, and discuss the benefits and limitations of such a system.

Traditional CG lighting is done with 3D software that allows a user to position lights in a virtual 3D world and render a frame to see the result. A typical drawback of such systems is that render feedback is not instantaneous and the lighter is blind to the final result until the render comes back in several minutes or even hours. Physical lights used in movie sets or photography studios, on the other hand, provide instantaneous feedback. Recent developments in affordable sensors, computer vision systems, and 3D printing enable a close proxy of the virtual lighting scene to be physically created, opening up possibilities for lighting at lightspeed.

PhysLights enables a user to light a CG scene as a cinematographer would light a movie set. Furthermore, the system provides the general benefits of a TUI: two-handed interactions, multi-person collaborative use, leverage of existing physical object manipulation and spatial reasoning, and intuitive metaphors and avoidance of jargon [2, 4, 6].

Using technology available today, we aim to demonstrate that such a future is within grasp. Our contribution is an example of how such a lighting system might look like in a tangible production pipeline and test data with CG animation professionals at DreamWorks animation PDI.

In the remainder of the paper we introduce the *PhysLights* system and outline how it was created. We then describe the method through which we tested the system with 6 DreamWorks artists, and the results that we obtained. We conclude with discussion of the limitations found and future research directions.

CSE216, December 15, 2014, San Diego, California, US.

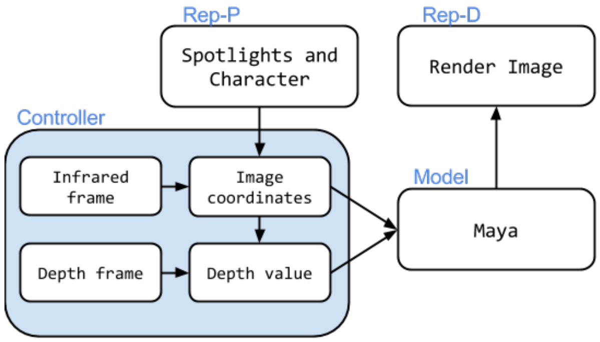

Figure 2: PhysLights MCRpd design

(a) Physical LED spotlight (b) 3D printed character

Figure 3: The physical components

Related Works

The framework for our work is the MCR(Model, Control, Representation) pattern[4, 6]. The strength of these interfaces is in the ability to bridge the gap between "Representation" and "Controller" for the user. The framework Ullmer and Ishii lay down helps us measure the cognitive engagement of the artist with the task when using a TUI.

There exists studies that find improvements in controlling camera motion, character animation, and spatial navigation[3, 5, 8]. Our study adds results specific to the CG animation lighting workflows, and provides test data with professional artists.

PHYSLIGHTS

The *PhysLights* system uses the MCRpd(Model-Controller-Representation Physical and Digital) model for Tangible User Interfaces [6]. The user controls the *Physical Representation* of two LED spot lights around a 3D printed character. A Microsoft Kinect camera that tracks the movements of the spotlights, a MATLAB program runs the detection algorithm, and a C++ program relays that movement information to virtual world *Model*. The *Model* is represented through a virtual scene in Autodesk Maya, a commonly used 3D package, that contains the light, camera, and character position. Finally the Solid Angle Arnold renderer renders an image that is displayed on a monitor, showing the *Digital Representation* back to the user.

Tangible Lights

The Physical Representation (the Rep-P) consists of two LED spot lights mounted on a flexible tripod (Figure 3a). These lights are placed around a 3D printed model of the character (Figure 3b) identical to the virtual model. The user can move the lights with their hands to illuminate the character in any way that is physically possible (the Controller). The LED lights cost $30 each to construct using off-the-shelf parts, and the 3D print cost $20 to from an online 3D print shop.

Computer Vision Detection

In order to obtain accurate virtual light positions from the physical representation, an automated Computer Vision system was developed to detect retroreflective tags placed on each LED light. Taking advantage of the fact that retroreflective tape has a high response to infrared rays, the frame by frame tracking is performed with the Kinect infrared data, simplifying object and background segmentation.

Once the tags are found for each light, the system can then obtain the depth values at each of the light coordinates through the Kinect depth data. However, due to the low resolution, refraction, and the small size of the LED lights, the depth data at the detected coordinates is often undefined. Therefore, the closest approximation of the light depth is taken from the point on the table directly under the light.

Virtual Scene

A C++ client program, receives the light positions and transforms them to the virtual world coordinates used in Autodesk Maya. The program then connects and sends transformation commands for each light to a Maya command server. This controls the virtual scene inside Maya (the Model). Autodesk Maya was chosen for it's widespread use in the 3D industry.

Rendered Image

Finally, Maya's Interactive Rendering feature signals Solid Angle Arnold to re-render the scene. This is shown in a Render Window to the user through a monitor (the Rep-D). Solid Angle's Arnold was used due to it's fast interactive render performance.

Figure 4: Target Image that testers were asked to match

METHOD

We took our *PhysLights* system to DreamWorks' PDI office to test with 6 artists (two lead lighters, two lighting technical assistants, one lighting technical director, and one effects artist). We had them perform two different tests: matching a rendered image (Figure 4) directly with *PhysLights*, and matching the same image with Maya's 3D Viewer interface. Half of the participants used the TUI first, and the other half used the GUI first, determined by random assignment, based on years of lighting experience to avoid confounding variables. We then measured their performance by time it took until the image matched. The accuracy was determined on the spot by a researcher with 3 years of professional lighting experience.

This test modeled a common lighting pipeline in which the Director or Art Director paints a color key, a small painted sketch, and the lighter has to match the CG render to that painting, by manipulating lights.

This experiment was followed by a 10 minute interview. We asked 4 questions, as well as opening up the dialogue for open-ended suggestions:

- What did you enjoy in lighting with the *PhysLights* system compared to Maya?
- What did you not enjoy in lighting with the *PhysLights* system compared to Maya?
- What would be on the top of your feature wish list if you were to use *PhysLights* system in production?
- If you were teaching a person new to CG lighting with *PhysLights*, how would you construct lighting exercises or explain lighting concepts to them?

The first three questions were intended to gauge what the strengths and weakness of the system are, while the last question was aimed at answering the open-ended question to what extent a TUI would help in educating new CG lighters.

RESULT

We present our measurements and interview reponses from the experiement described in the Method section.

Time

The time results measured are presented in the Table 1. The *PhysLights* time were measured to when the user matched the lighting on the physical representation only, and not the digital representation, due to technical difficulties. Even including the technical difficulties, only one user's time went over the time spent matching in Maya.

On average, users were 3.1 times faster when creating the virtual scene with *PhysLights* than with Maya. A Student's t-test was used to confirm that the measured difference in average matching time is statistically large enough to reject the null hypothesis (alpha = 5%, t statistic = 3.9782, p-value = 0.0026, df = 10). Additionally, users who used *PhysLights* in their first trial were 1.8 times faster in matching the virtual scene when using Maya during their second trial than those who used Maya as the first trial (alpha = 10%, t statistic = 2.7350, p-value = 0.0522, df = 4). However, the speed improvement of using *PhysLights* was independent of the order of trials.

Also both times showed no correlation with years of lighting experience in the user.

Advantages

A Lighting Technical Director pointed to the fact that there are significantly less steps involved compared to working in a conventional 3D pakcage. A lead lighter and lighting technical assistant mentioned that the system made lighting fun.

Three users utilized many more expressive controls than would be possible in a traditional mouse-keyboard lighting systems. For example, often times both hands were used to move two lights simultaneously, an impossible task in Maya as it will only allows control of one light at a time. Two artists described the system in words that evoke physical concepts such as "gravity" and "space", which suggests they were accessing their natural spatial reasoning skills [4].

All this points to the fact that the system successfully reduced the cognitive distance between the artist and the rendered image. These features make use of a TUI "Artist-Friendly", as

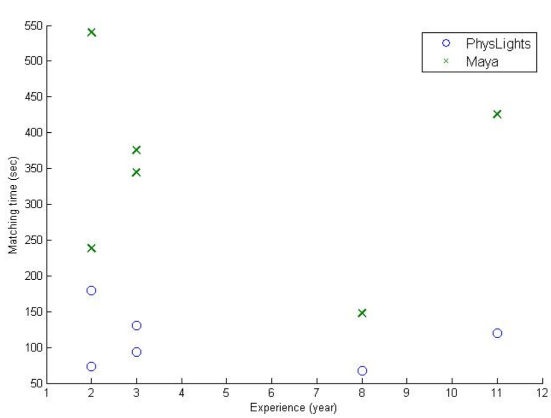

Figure 5: Matching time was faster independent of user's lighting experience

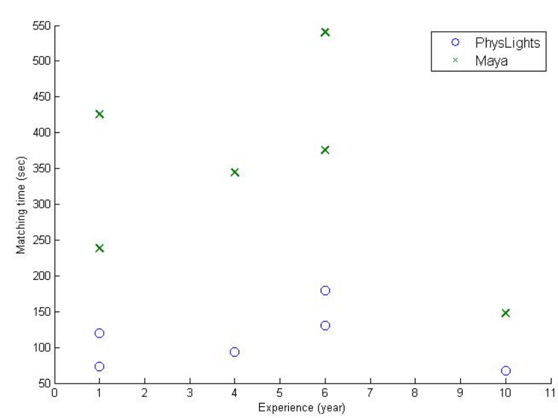

Figure 6: Matching time was faster independent of user's Maya experience

Tester	Position	Lighting Experience	First experiment	*PhysLights* Time	Maya Time
1	Lead Lighter	11 yrs.	Maya	2:00	7:06
2	Lead Lighter	8 yrs.	PhysLights	1:07	2:28
3	Lighting Technical Assistant	3 yrs.	Maya	2:10	6:16
4	Lighting Technical Director	3 yrs.	PhysLights	1:34	5:44
5	Effects Artist	2 yrs.	PhysLights	1:13	3:58
6	Lighting Technical Assistant	2 yrs.	Maya	2:59	9:00

Table 1: Time data showing the generally faster *PhysLight* time.

was described by a tester. Note, however, CG lighting inevitably involves many technical tasks.

Disadvantages

Another unanimous opinion was that fine tuning the lighting condition to the extent of a 3D package is challenging, or even impossible in some instances. For example, a user's motor skills are not as precise as typing in a numerical coordinate. Also, 3D packages allow artists to use "light-linking"—a feature that makes lights only illuminate specified objects, and nothing else—a physically impossible task. These tasks are examples that require deep technical knowledge of the artist and features in the system to support such actions.

Also the current system's lack of physical representation of the camera sometimes caused dissonance between the user's mental model and the CG render. A similar issue happened with the character, as moving the physical character did not move it in the virtual scene.

Two users raised concerns that in virtual lighting they could simply add more lights, but in *PhysLights* they would need to have these lights on hand. The effects artist brought up a similar concern about scalability, as a lighting shot in production could contain up to several hundred lights, and placing that many lights in physical space would become prohibitively complex and expensive. The same artist also said for the specific use case of illuminating ephemeral effects such as smoke or fire would be very challenging.

These limitations are discussed further in the Future Features and Discussion sections.

Future Features

The first feature that all testers requested was the ability to control the qualities of light, such as exposure, color, cone angle by the *PhysLights*. The two lead lighters made a related feature request to be able to add more tools from a set lighter's arsenal such as bounce cards, blocker cards, or soft boxes.

Three artists wished they could also move or position the camera and characters so that they can directly manipulate elements of the physical layout other than the lights and see the change in the virtual world.

Education

Our open-ended question of how we might teach a complete novice using this system was met with generally encouraging comments. A lighting technical assistant relayed their experience in school of learning physical lighting before attempting to recreate that lighting with a 3D software, and *PhysLights* would have alleviated a need for such an exercise. Three testers expressed that because it is so "WYSIWIG", the beginner can be taught the basics of lighting without the jargons required to use a typical 3D package. One lead lighter suggested since the system is so fun, they would simply let the novice experiment with the system.

One tester expressed their skepticism and that although it's a good intro to CG lighting, the student should be taught in a 3D software after maybe playing with the system after the first hour.

DISCUSSION

PhysLights proved to be fast, intuitive, and easy to learn. However, there are serious limitations we need to consider.

Quality of Light

As all the testers pointed out, the current implementation of *PhysLights* does not track quality of light such as color, softness, and exposure. Also tracking of lights from different sources such as bounce cards or sunlight is not possible.

In the next implementation of *PhysLights* the use of a DSLR camera and spherical mirror to capture a bracketed HDRI map [1]. Instead of tracking light position, this will directly capture the lighting information and capture any changes in lighting that occurs. The choice of LED spotlights were because they could programmably dimmed for these future extensions. However, lights aren't the only factor that's used in lighting.

Quality of Surface

A much more challenging problem is the ability to accurately represent surface information. With current 3D printer technology, the materials and colors that can be used to print a 3D character is limited, and representing non-diffuse, highly reflective, or refractive materials is prohibitively expensive.

Also, as mentioned in the Disadvantages section, non-solid surfaces such as smoke or fire is impossible to represent even with near-future advances of 3D printers.

Even within character lighting, accurately lighting eyes, a refractive and reflective surface, is considered one of the most important tasks. This is a severe limitation of the current system, even in the context in which it was tested.

One-way Sync

PhysLights currently takes in changes in the physical representation and translates it to a digital representation. It does not sync in the other direction of translating changes in digital representation into the physical space. This can be problematic in the scenario that the user wants to come back to work in the physical lights scenario after having worked inside a 3D package.

A possible solution is to relay the light positions to the user either with a display on the table or a projector. Even then, this wouldn't be a possible solution with the HDRI capture method described above.

Cost

The effects artist raised concerns about scalability not only in the number of lights a user can control, but also in the number of lights at which it becomes cost prohibitive. Perhaps this is where it becomes most clear that there are advantages and disadvantages to TUI systems, and it is more likely to that a lighting artist might take a hybrid physical and digital approach.

A lighting artist might more effectively start exploring the scene with a small number of lights in *PhysLights*; during this phase, they can communicate and collaborate with the director or cinematographer rapidly to adjust the vision. With a much clearer consensus between the artistic leadership and artist, they can then take the work they've done to seed into their traditional workflow and add complex and fine-tuned lighting systems. This avoids the increase in cost as more components and lighting equipment are added to the scene, while preserving the cost benefit of requiring less training, allowing faster iteration, and reducing back-and-forth with higher-cost individuals such as the director, ultimately bringing production cost down.

Despite the limitations, we hope *PhysLights* demonstrated the ease-of-use and the human-cost benefits of using a TUI in CG Animation, and influenced further development of these interfaces in CG Animation production.

CONCLUSION

We outlined the *PhysLights* TUI system and its testing in with CG lighting professionals. *PhysLights* invites many users new to CG animation production with an easy to learn interface, and empowers experts with a faster, and more intuitive workflow. With the improvement of sensing technology, reduction of CG production cost, and ever-increasing interest in 3D animation production, we believe now is an ideal time to explore the capabilities of TUIs in CG animation production. We foresee future TUI systems for CG animation productions to become even more prevalent and even though that time is not upon us yet, *PhysLights* demonstrates what's possible today.

ACKNOWLEDGMENTS

We would like to thank Scott Klemmer and Vineet Pandey for the amazing CSE 216/COGS 230 class. Without the six DreamWorks artists who volunteered their time this project wouldn't have been possible. We would also like to express gratitude to Jonathan Gleit from DreamWorks Videography who helped film and edit the introduction video.

REFERENCES

1. Kusuma Agusanto, Li Li, Zhu Chuangui, and Ng Wan Sing. Photorealistic rendering for augmented reality using environment illumination. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 208–216. IEEE, 2003.

2. George W Fitzmaurice, Hiroshi Ishii, and William AS Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449. ACM Press/Addison-Wesley Publishing Co., 1995.

3. T Haley-Hermiz, PE Connelly, A Gasper, V Scalone, N Sceusa, and C Staehler. Augmented cinematography: A look at the use of augmented reality in film production. In *66 th Mid-year Conference of ASEE Engineering Design Graphics Division*, pages 214–223, 2012.

4. Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM, 1997.

5. Sageev Oore, Demetri Terzopoulos, and Geoffrey Hinton. A desktop input device and interface for interactive 3d character animation. 2002.

6. Brygg Ullmer and Hiroshi Ishii. Emerging frameworks for tangible user interfaces. *IBM systems journal*, 39(3.4):915–931, 2000.

7. Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. Printed optics: 3d printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 589–598. ACM, 2012.

8. Andy Wu, Derek Reilly, Anthony Tang, and Ali Mazalek. Tangible navigation and object manipulation in virtual environments. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 37–44. ACM, 2011.