The Smart 'AnaLamp'

Marisa Lu
HCI Design, Carnegie Mellon University
Pittsburgh, PA, United States
findme@marisa.lu

Max Maguire
ECE, Carnegie Mellon University
Pittsburgh, PA, United State
wmaguire@andrew.cmu.edu

Maayan Albert
HCI Design, Carnegie Mellon University
Pittsburgh, PA, United States
malbert@andrew.cmu.edu

Abstract— AnaLamp, inspired by the Pixar short Luxo Jr, is a smart lamp that serves as a desk assistant and companion. It utilizes IoT technology, computer vision, and physical computing to improve the experience of working at and away from a desk. The lamp provides manual Web application control and automated features for face tracking, remote occupancy detection and hand gesture control.

Keywords—IoT; Ubiquitous Computing; Computer Vision; User Sensor Systems

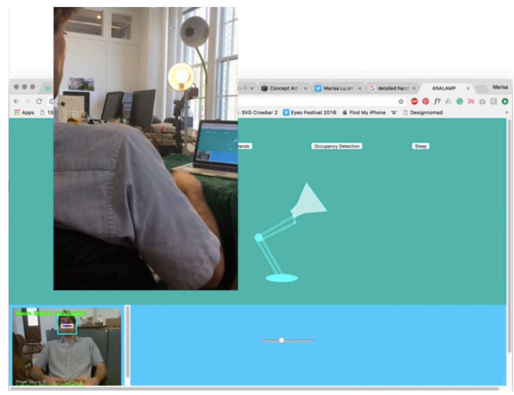
1. INTRODUCTION

Currently, much attention has been placed on enhancing office workflows within the realm of virtual environments. Software such as word, pages, and Google Drive help us create, while applications such as Slack and Messenger allows us to communicate between one another. However, there are few technologies that are aimed at improving how users interact and experience their physical environmental. This is the goal of AnaLamp, a smart lamp engineered to improve the desk environment. AnaLamp accomplishes this by assisting users by performing tasks that would otherwise be difficult to carry out. These tasks may include locating objects of interest remotely, detecting and/or tracking occupants, providing remote/interactive controls and serving as a source of animated entertainment in an otherwise dull environment. In this paper we will discuss about system architecture of the Analamp along with its mechanical design, summarize the built-in features and discuss the social implications of such a smart lamp.

CONTEXT

As the capabilities of embedded computing constantly expand, IoT enthusiasts have become empowered to digitize everyday objects to explore new use cases. However, many of these products become obfuscated with informatics that can only be accessed on a virtual platform while adding little utility to the natural function of the original object. With this in mind, we focus on design approaches that could be used to leverage the natural utility of a common appliance to merge digital and physical spaces. We were compelled to build around a common desk lamp whose primary function is to illuminate regions of interest. By remodeling a common desk lamp into a user sensor system, we hope to demonstrate the range of assistive capabilities that can be seamlessly incorporated to enhance a users workspace. In addition to technological feasibility of creating such an appliance, we explored what type of functionalities might be useful for such a product and the possible social affordances it offers.

Figure 1: AnaLamp Web App and Lamp Setup



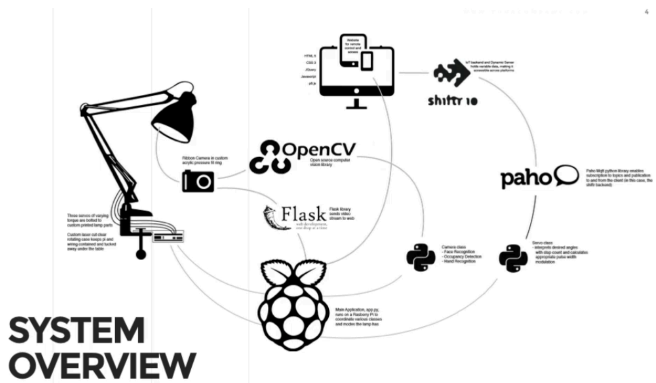
2. DESIGN APPROACH

Integrating interactive features for the Analamp to engage with its environment required an abstracted system level design and mechanical design. The system level abstraction would allow for a bi-direction data flow for remote and automated purposes. The mechanical design would provide Analamp with a range of functional movements and usability features. In the following sections we will provide a higher-level overview of these designs.

SYSTEM LEVEL DESIGN

The processing flow of Analamp is built off of the main application, which provides bi-directional communication between the Camera feed, the servomotors and the Web application. At run-time, the main application initializes a Camera object, three servo objects and a threaded client object to communicate with the Web Application backend shiftr.io.

Figure 2: Flow Diagram of System Architecture



The main application is based off of the Flask framework so as to stream video frames from the raspberry pi camera module as MJPEGs on a local server. After initialization, the main application waits for a client to contact the server at which point it generates the Camera stream to render onto the Web application. The main application also uses the paho library to construct its own threaded client object at run-time for receiving messages from the Web Application via the IoT backend 'shiftr.io'.

The Camera object in the main application is a class, which contains methods for creating a continuous video stream in a separate thread. Addition image-processing functions can be applied to each frame of the video stream. The states of which image-processing tasks that are to be applied are updated within the main application. The Pertinent data that is computed while capturing and rendering a frame (state of image-processing tasks and results of image-processing tasks) is stored in Camera class instance variables. All Image processing tasks are performed using the image-processing library OpenCV.

The Servo classes use the Adafruit servo hat library to generate PWM signals to the servo channel for which it is assigned. The Servo class contains a method to rotate at a user-defined speed to a specified angle as well as a method to retrieve its current position. The servomotors are actuated within the main application after processing messages that are received from both the Web application and the Camera class.

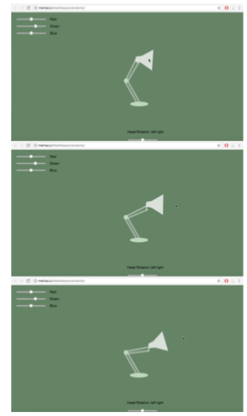


Figure 4: Analamp Web App Manual Control

The web application provides graphical controls to enhance the UI/UX experience. Messages are sent from the web application to the main application in order to control the feature state and movement of the AnaLamp. For example, when the user selects the manual control mode, the main application toggles off all the camera image-processing states and uses angle positions based on the user-web application interface to actuate the specified servomotors. The interface's animated graphics are constructed using a Javascript library called p5.js. The interface allows users to pinpoint a single target location for the lamp to illuminate by moving their cursor.

MECHANICAL DESIGN

The AnaLamp contains three degrees of rotation at which it can re-orient itself. Servomotors control the three degrees of rotation: one at the middle linkage of the lamp and two more linking the head to the body. The motor at the middle linkage controls the vertical rotation of the lamp's upper body like a mechanical tendon, while the motors connected to the head control its vertical and horizontal rotations.

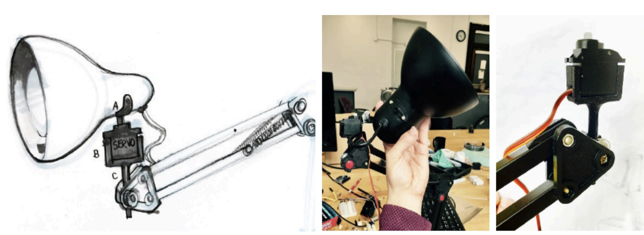
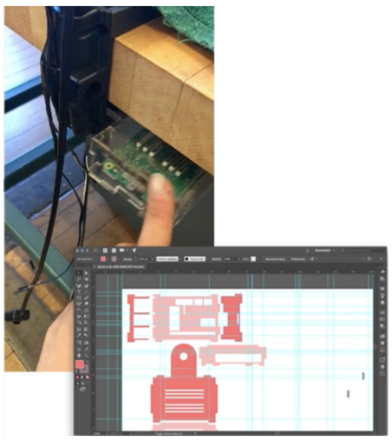


Figure 3: Mechanical design of servo mount for lamp head hinge

Custom 3D printed hinges were designed to mount the servomotors at each degree of rotation: body and head (pitch and yaw). These servomotors are controlled through a servo hat that is attached to a raspberry pi 3 located in a case at the bottom of the lamp. The laser printed case containing the raspberry pi was designed to freely rotate for user accessibility and to keep the wires safely/ aesthetically out of view.

Figure 5: Mechanical design for raspberry pi case



3. AUTOMATED FEATURES

The smart lamp system provides a Manual Web app control and Automated facial tracking, occupancy detection, and hand gesture features.

OCCUPANCY DETECTION

The AnaLamp uses a simple motion detection algorithm to determine if a potential occupant is in the room. The AnaLamp accomplishes this by scanning the room by rotating its 'head' by a fixed step size, capturing an initial grayscale image at each step and using it as a reference to track differences between it and all consecutive frames at the position. This measurement yields a magnitude at each pixel value, indicating regions of significant change (representing a potential occupant). For regions that are large enough to exceed a predefined threshold value, the room's state is then classified as occupied. At this point, Analamp actuates the appropriate servomotors in order to reorient itself so that the area of significant movement is centered in the middle of the frame. If enough time has passed without an occupant being detected, Analamp proceeds to scan.

Figure 6: Occupancy Detection demo

FACE TRACKING

AnaLamp uses the Haar Cascade classification method in order to detect a user's face. This object detection method uses a large pool of labeled images to compose a feature set that can then be referenced to identify similar objects in an image. For our purposes, we used a preprocessed xml file designed by Intel to identify frontal faces that matched within a scaled range. This was used to implement a face-tracking mode for AnaLamp, adjusting the orientation of the lamp to follow a user's face. Alternatively, this can be extended to detect other objects in the image (watch, phone, computer). A new cascade xml files can be generated by uploading an image containing an object of interest to the web application.

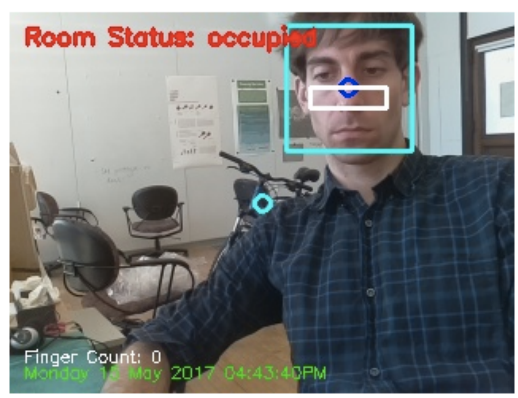
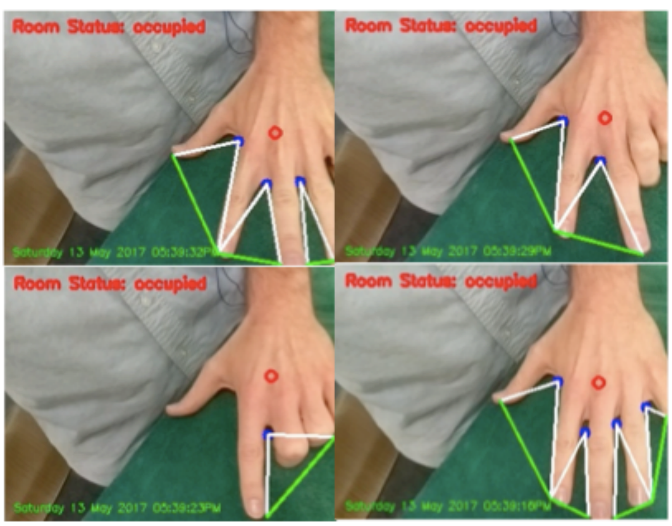


Figure 7: Face Tracking demo

HAND GESTURE CONTROL

In order to implement an adaptive skin filter for hand detection, we used the face detection method described in the previous section. Once we identified a face in the image, we selected a sub-region of the face with the most skin (below the eyes) to calculate the distribution of hue across the user's skin. This gave us an expected mapping for the user's skin tone that we then back projected into a threshold for determining which part of the image to consider skin for a particular individual. Once this information was gather, the lamp repositioned itself and illuminate the desk (where the user would place their hands). This threshold was then applied to the frames of a desk, providing a mask to separate the users hand from the background on which they were working. Once the hands are identified, the contours of the hand were measured to identify key features: the tip of the fingers, the gap between each finger and the center of the palm. These features were then used to compute the length of each finger, the direction each finger is pointing and the angle between fingers.

Figure 8: Hand Gesture demo



4. **SOCIAL IMPLICATION**

While working on this interactive technology, we wanted to address three key elements that would promote user adoption: Security, Engagement, and Utility. The passive surveillance that leads to loss of both control and privacy is often brought up as general critique for ubiquitous computing and IoT. The lamp offers a unique social affordance over traditional methods of passive surveillance by its overtly dynamic design. A user can assure security by scanning the state of his/her workspace by remotely connecting into his/her desk lamp. The animated mechanics of the lamp also provides a continual feedback to inform occupants in the room of its current state. Furthermore, the range of dynamic movement enables the lamp to assist the user in more ways without disrupting his/her workflow. Features like the hand gesture control and object detection allow users to remotely customize the lighting of their environment.

5. CONCLUSION

While this is an initial exploration, we believe that there are many exciting areas of further development for the 'Analamp'. Functionally, the lamp can be modified to provide lighting control that would be useful to the user (i.e. RGB intensity control/image projection). Also, The facial tracking feature can be expanded to track or detect user specified objects of interest from a library of labeled images. In addition to this, a face recognition algorithm could be implemented to identify/track specific users. Lastly, the smart lamp can refine its movement to provide additional cues through more emotive animation, making it more user-friendly and further encouraging user adoption. Smart appliances, such as the Analamp, serve a range of purposes and will become increasingly common within work and professional spaces. With this concept, we strive for technology that can more effectively merge digital and physical spaces and provide features that encourage human interaction.

ACKNOWLEDGMENT

This project has been done as part of the User Sensor System course (08-735) taught by Professor Mayank Goel.

REFERENCES

1. "Face Detection Using Haar Cascades." OpenCV: Face Detection Using Haar Cascades. Doxygen, n.d. Web. 16 May 2017. <http:// docs.opencv.org/trunk/d7/d8b/tutorial_py_ face_detection.html>.
2. "Opencv Python Hand Gesture Recognition." Iftheqhar's Blog. Blogger, n.d. Web. 16 May 2017. <http://creat-tabu.blogspot.ro/2013/08/ opencv-python-hand-gesture-recognition. html>.
3. Weiser, Mark. "The Computer For The 21st Century." Scienti c American (1991): 94-104. Web. 16 May 2017. <https://www.lri.fr/~mbl/ Stanford/CS477/papers/Weiser-SciAm.pdf>.
4. Lu, Marisa, Gautam Bose, Austin Lee, and Peter Scupelli. "Knock Knock to Unlock: A Human Centered Novel Authentication Method for Secure System Fluidity." (2017): n. pag. Web.
5. Pinokio. Dir. Adam Ben-Dror and Shanshan Zou. Vimeo, 2012. Web. 16 May 2017. <http://ben- dror.com/pinokio/>.
6. Luxo Jr. Dir. John Lasseter. N.p., 12 Aug. 1986. Web.