Life of the Party

REAL-TIME MOTION CONTROLLED ENTERTAINMENT LIGHTING

JUSTIN FRAUMENI · JEREMY WARNER · AMELIA KELLER · MATTHEW JOHNSON

UNIVERSITY of ROCHESTER

HAJIM SCHOOL OF ENGINEERING & APPLIED SCIENCES UNIVERSITY of ROCHESTER

INTRODUCTION

Modern light systems for live entertainment applications are traditionally rigid in design. For dance shows and concerts, light schemes are often pre-programmed scene by scene to synchronize with the energy of performances, a process that can be as inflexible as it is time consuming. Lighting for party environments often reacts to music levels, but can be repetitive and predictable. "Life of the Party" applies principles of computer vision to provide a plug-and-play solution to lighting design. The system automatically responds to movement in a predefined space and changes lighting levels to correspond with motion, adding a layer of real-time human interactivity to entertainment events.

TECHNOLOGY

SaikoLED

SaikoLED lights provide full spectrum RGB color and utilize an integrated arduino microcontroller to generate a wireless network, through which RGB values can be manipulated.

Kinect for Xbox

Kinect is a motion sensing device originally designed for use with the Xbox 360 platform. The Kinect features an RGB camera and depth sensor, which it uses to track the full body movement of multiple users in three dimensions. Critical to our application, the depth sensor utilizes an infrared laser and CMOS sensor allowing it to quantize position points in low lighting levels.

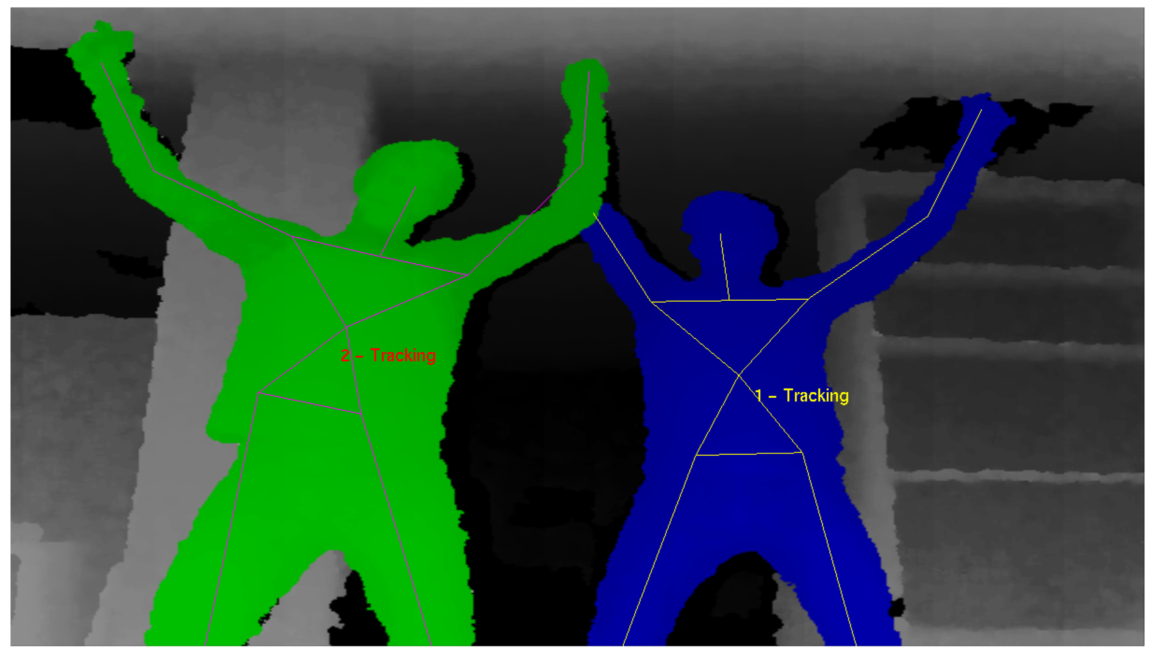

Figure 1. OpenNI user tracking depth field image

OpenNI SDK

OpenNI is an open source C++ image processing library which can used to interface Xbox Kinect with Unix-based systems. OpenNI provides access to the euclidian coordinates of 15 "joint" data points on up to 6 concurrent users within the Kinect's viewfield.

Python Liblo

Liblo is an implementation of the Open Sound Control protocol for POSIX systems. Our system utilized pyliblo, an implementation in python, to communicate between the laptop hub and the SaikoLED's specific IP address.

SYSTEM

High level discription

Upon each camera frame, the Kinect outputs position and depth metrics to a computer hub containing the program files, scripts, and libraries through a high speed usb connection, at an update rate equal to the Kinect's frame rate of 30 fps average. The computer hub then sends update commands to the SakioLEDs through their self-generated, dedicated wireless network.

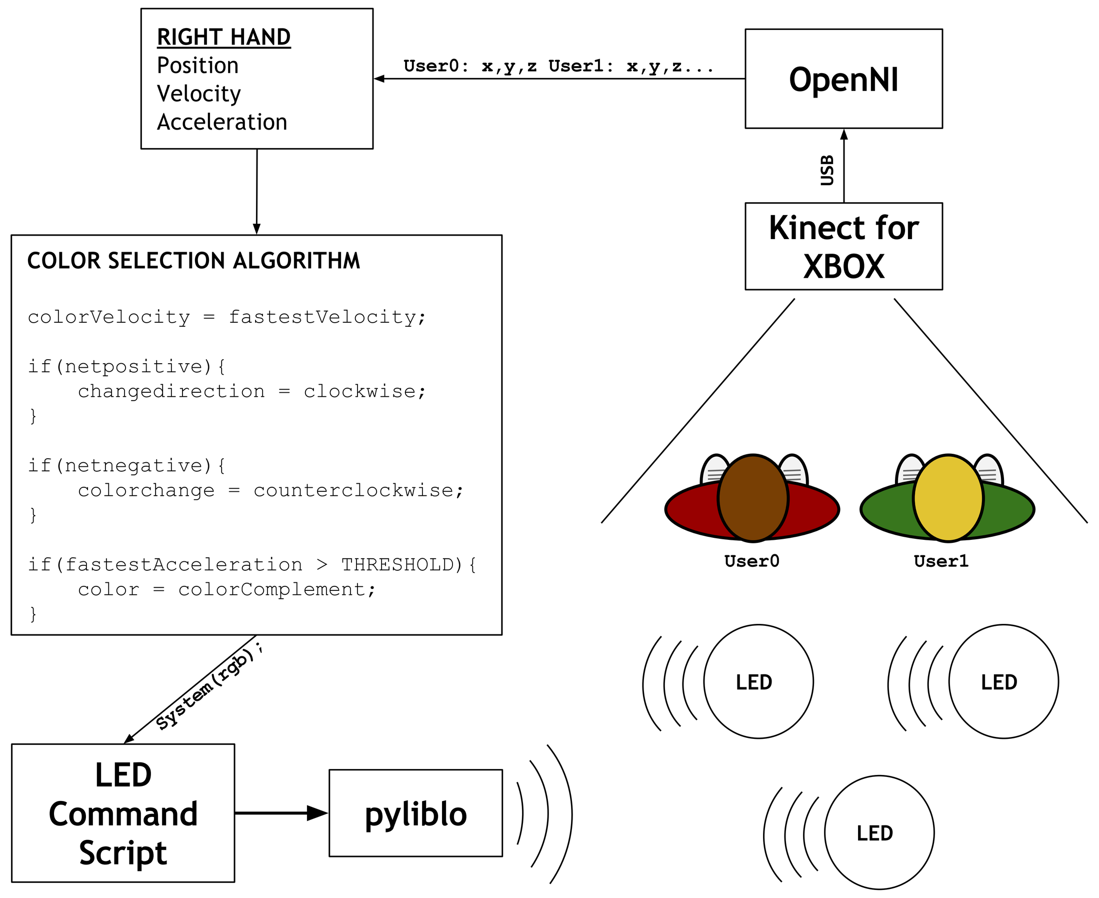

Figure 2. System diagram, highlighting color selection algorithim

User joint position, velocity, and acceleration

OpenNI stores the number of users within its viewfield in an array register automatically as they appear, and assigns them a corresponding ID. Users can then be referenced by this ID and euclidian position of skeleton joints and extremities can then be extracted. Upon every camera frame, euclidian distance is calculated between the current and prior position of a selected joint of a dancer. The first implementations of Life of the Party rely on the right hand as a representation of dance intensity. Average velocity is computed for each user using a recent history of distance values and time change. This in turn is used to calculate an acceleration value for each user. These metrics are now ready to be used in the color selection algorithms to determine the light control command sent.

Net positive and net negative assignment

Velocity is interpreted by the program logic as either "net positive" or "net negative", depending on its direction in euclidian space. From the perspective of the Kinect, movements directed to the right, or up are considered "net positive", and those directed downwards and to the left are considered "net negative".

Multiple user functionality

When the system is extended to multiple users, the highest average velocity and acceleration values for a frame are selected as inputs for the color selection algorithm. Generally, the most active user during each the frame determines the color selection command sent.

RGB color shifts

The color selection algorithm has a base velocity threshold of 1.0 m/s. Average velocities below this are too slow to be used in color selection and considered noise. Velocities greater than this contribute to a direct color change "velocity" around the RGB color wheel, in a direction defined according to the net direction of the movement: clockwise for "net positive" and counterclockwise for "net negative". This specification is made because of the fundamental nature of many styles of dance, which feature alternating directional movements. The ideal end result is alternating color change between two colors adjacent on the color wheel, in phase with dance moves.

Complementary color jumps

Abrupt, intense movement by dancers, characterized by a high acceleration value, results in immediate color change. Acceleration values which exceed a constant threshold result in a color "jump" across the color wheel to a complementary color. *Source code: https://github.com/jeremywrnr/life-of-the-party*

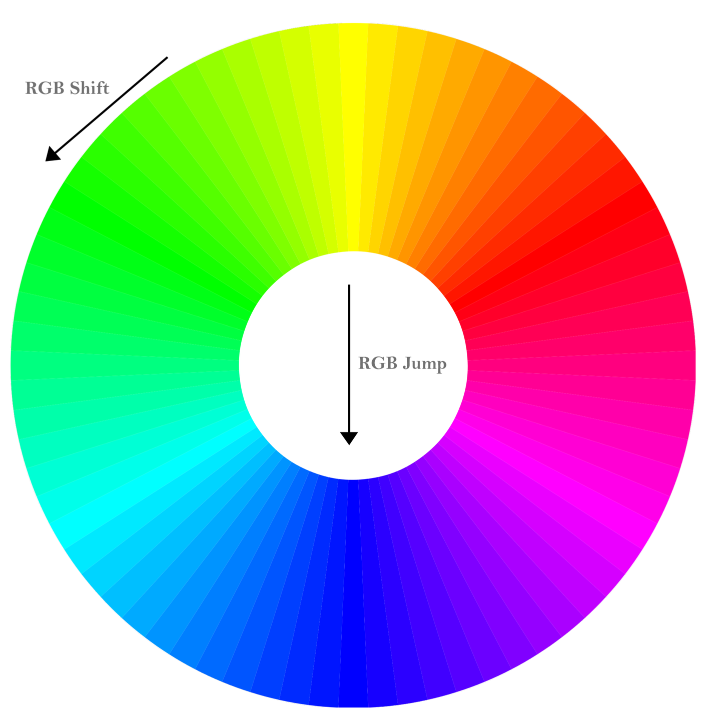

Figure 3. RGB color wheel with labeled jumps and shifts

EVALUATIONS

Two iterations of user demo and evaluation took place, in between prototype versions of the system. The first of these involved 11 individuals. The volunteers were debriefed on the functionality of the system and allowed to interact with it freely. They were then queried on their reaction and suggestions.

How would you rate the responsiveness of the system?
Could you see a high correlation between your movements and the lights?
What problems were encountered while using the system?
Would you use the system at a party or dance event in place of sound controlled lighting?
What features could be added to improve the system?

Feedback received during the first iteration was mixed. All users expressed potential for the system enthusiastically: 100% responded that they saw themselves using the system at a party in place of sound controlled lighting, however many were confused about the correlation between their movements and the color change, in the words of one user "sometimes the color change was too unpredictable." The first round of user demos led to several key changes in the system, including the positive/negative color shifting concept, implemented in order to increase the correspondence and predictability of the color change to motion.

The second iteration of user study recruited members of Louvre, the University of Rochester ballet dance group to perform with the system. The performance, along with the interview that followed, highlighted the improvements, along with several key remaining weaknesses of the system that are revealed when the system is scaled to multiple concurrent users:

Delay -The time delay that OpenNI implements between dropping and adding users to its register can prove problematic to a performance in which dancers are rapidly entering and leaving the viewfield.
Occlusion - While the Kinect has rich 3D image processing abilities, recognition failure still occurs when users are superimposed.
Overreactivity - Depending on the style and intensity of dance, sensitivity must be tuned to avoid erratic light change.

FUTURE PROSPECTS

In accordance with recomendations made by staff of University of Rochester' s Event Support crew, future versions might implement color and sensitivity parameter selection through a graphic user interface, as well as concurrent joint tracking. This would increase the flexibility of the system when applied to different performances and events.

ACKNOWLEDGEMENTS

The LOTP team thanks the following organizations and individuals for their support of the project: CSC 212 Prof. Hoque, University of Rochester Event Support, Louvre Performance Ensemble, ROC HCI, Alex Wilson, Michelle Fung, Computer Science Department, Open Source Software and Firmware (Arduino, SaikoLED, OpenNI.).