Recognizing Gestures with Ambient Light*

Raghav H. Venkatnarayan
North Carolina State University
Raleigh, North Carolina
rhampap@ncsu.edu

Muhammad Shahzad
North Carolina State University
Raleigh, North Carolina
mshahza@ncsu.edu

ABSTRACT

There is growing interest in the research community to develop techniques for humans to communicate with the computing that is embedding into our environments. Researchers are exploring various modalities, such as radio-frequency signals, to develop gesture recognition systems. We explore another modality, namely ambient light, and develop LiGest, an ambient light based gesture recognition system. The idea behind LiGest is that when a user performs different gestures, the user's shadows move in distinct patterns. LiGest captures these patterns using a grid of floor-based light sensors and then builds training models to recognize unknown shadow samples. We design a prototype for LiGest and evaluate it across multiple users, positions, orientations and lighting conditions. Our results show that LiGest achieves an *average* accuracy of 96.36%.

CCS CONCEPTS

• Human-centered computing → Gestural input;

KEYWORDS

Gesture recognition; Ambient light

ACM Reference Format:

Raghav H. Venkatnarayan and Muhammad Shahzad. 2018. Recognizing Gestures with Ambient Light. In *10th Wireless of the Students, by the Students, and for the Students Workshop (S3 '18)*, November 2, 2018, New Delhi, India. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3264877.3264883>

1 INTRODUCTION

Motivation: With the advent of the Internet of Things, we are witnessing rapid infusion of intelligent devices such as smart TVs and thermostats. The increasing number of such

devices requires new interfaces that enable seamless interaction. A natural choice for such interaction is gestures as they are an integral part of how humans interact. Indeed, there is a growing integration of gesture recognition systems into various consumer electronics such as gaming consoles. While state-of-the-art device-free gesture recognition approaches explore radio frequency (RF) based modalities [1, 4], such as WiFi and RFID (e.g. by monitoring changes in the RF signals due to human movements), there are other modalities, such as ambient light, that can also enable gesture recognition (e.g., by monitoring changes in the intensity of light due to human movements) and deserve exploration, but have not yet received enough attention from the research community. Since each modality has a unique set of limitations and cannot enable gesture recognition in every scenario, we envision that a truly practical gesture recognition system can be developed only by integrating multiple gesture recognition systems that complement each other and each of which use a unique modality such as ambient light, WiFi or ultrasonics.

Problem Statement: Towards realizing this vision, our objective is to develop an ambient light based gesture recognition system, LiGest, that has the following three properties:

- 1) **Lighting Condition Agnostic:** The system should not be affected by changes in the intensity, number, and/or position of light sources in the environment. The system should also not assume any control over the lighting infrastructure and should work with all types of ambient light sources.
- 2) **Position and Orientation Agnostic:** The system should recognize gestures irrespective of the user's position and orientation in a given indoor environment.
- 3) **User Agnostic:** The system should recognize gestures of any arbitrary user in a given indoor environment that it has not trained on and not only of a specific set of users.

2 SYSTEM DESIGN

In this section, we provide a description of LiGest's sensing platform, and then outline its gesture-recognition algorithm.

2.1 Sensing Platform

To collect gesture samples from users, we first develop a sensing platform that comprises a portable and sturdy 6ft. × 6ft. mat and $N = 36$ light sensors placed on it in the form of a square grid with an even spacing of 1ft between adjacent sensors as shown in Figure 1. Each light sensor (TSL237)

* A full paper is available in [3]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

S3 '18, November 2, 2018, New Delhi, India

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5932-0/18/11...\$15.00

<https://doi.org/10.1145/3264877.3264883>

Figure 1: Floor-based Sensing Platform

measures intensity of light due to moving shadows when a user performs gestures while standing on the platform, and reports them to a central server at a fixed sampling rate (100hz) through an Arduino Uno board. We term each set of reported sensor values as a stream. Next, the server uses the received streams to either train a set of classification models or consult the set of trained models to perform gesture recognition. To obtain gesture samples, we place the platform on the floor in a rectangular room and create 6 different illumination levels and ask volunteers to stand on it at different positions (numbered 1 to 9 in Figure 1) and in different orientations (4 directions) and perform 5 whole-body gestures (*viz.* Clap, Hug, Jump, Punch, Step).

2.2 Gesture Recognition

In this section, we provide an overview of LiGest's gesture recognition algorithm. To recognize a given set of gestures, LiGest needs to build a set of classification models. For this, LiGest first acquires training samples of the gestures from at least one user in at least one position and orientation under at least one lighting condition. It then builds the classification models for the given set of gestures in the following three steps.

2.2.1 Preprocessing. In this step, LiGest takes N continuous streams of all N sensors and processes them to make them usable for generating accurate classification models. During this step, LiGest performs the following three operations.

1) Denoising: The raw sensor streams can contain two types of noise: stray shadow noise, which occurs due to the shadows cast by the static objects in the environment, and hardware noise, which occurs due to the 60Hz flicker of fluorescent lights and minor hardware imperfections. LiGest first removes the static offset caused by the stray shadow noise from a given sensor stream by taking its first order difference. We name the resulting stream *dS-stream*. Next, it removes any flicker noise from the *dS-stream* by applying a low-pass filter and then filters out hardware noise by performing hard wavelet thresholding[2]. We name the resulting stream as a denoised *dS-stream*.

2) Gesture Detection: After denoising, LiGest automatically detects the start and end times of gestures. When a user is not performing a gesture, theoretically, all values in the

denoised *dS-streams* should be 0, irrespective of the environment. Practically, we observe that the values are equal to few tenths of a decimal, and thus, when the absolute values of a small set (e.g. 50) of the denoised *dS-streams* exceed a threshold slightly greater than 0, LiGest detects the start of a gesture and conversely, when these values drop below the threshold, LiGest detects the end of a gesture.

3) Standardization: Due to differences in the intensities of different light sources or due to differences in the distances of a user from the light sources, the shadow of the user falling on a given sensor for a gesture sample may be darker/lighter compared to another gesture sample for the same sensor. Consequently, one sample can see larger/smaller changes in the intensity of light compared to another sample for the same gesture, which hampers LiGest's accuracy. Fortunately, we observe that despite the differences in magnitude, the shape of the waveforms are similar for each gesture, and thus the magnitude differences can be largely removed by scaling the waveforms by their deviation. Therefore, after detecting a gesture, LiGest performs a standardization operation on the denoised *dS-streams* of each sample during which it applies Z-score transform temporally on the denoised *dS-streams*. We name the resulting streams standardized *dS-streams*.

2.2.2 Feature Extraction. Next, to generate classification models for the gestures, LiGest needs features that satisfy following three requirements: 1) possess high classification potential 2) be agnostic to changes in the position, number of light sources, user's position and orientation. 3) minimal in number to keep the computational complexity low. LiGest extracts features that satisfy these requirements by applying the following three operations on the standardized *dS-streams* of each gesture.

1) Wavelet Transformation: While the values in the standardized *dS-streams* can be directly used as features, the resulting number of features would be very large. For example, a gesture with duration of 512 samples and $N = 36$ results in $512 \times 36 = 18432$ features. To reduce the number of features, LiGest applies stationary wavelet transform (SWT) on the standardized *dS-streams* and calculates detail-coefficients from a SWT level that produces minimum number of coefficients without losing any useful information (e.g. 32 Level 5 coefficients for each stream).

2) Rasterization: On obtaining the coefficients, LiGest applies a transformation operation, which we call rasterization. The objective of rasterization is to make LiGest independent of the number and position of light sources as well as the position and orientation of the user. To achieve this objective, we observe that *any change in the number/position of light sources or position/orientation of the user across multiple samples of a given gesture largely changes the actual sensors that record the pattern, but not the pattern itself*. For example,

if there is a change in the position of a light source or the position/orientation of a user with respect to the light source between 2 samples for the same gesture, then we will see a change in the length and/or direction of the user's shadow. Now, if this change in position only alters the direction of the shadow, then the shadow will fall on a set of sensors different from the previous sample. However, this new set of sensors will still see same pattern for a given gesture as the previous sample. Similarly, if the change in position increases the length of shadow, then the shadow will fall not only on the same set of sensors as the previous sample, but also on a new set of sensors along its increased length. However, the previous as well as the new set of sensors will still see the same shadow pattern as the previous sample. Finally, a change in the number of light sources changes the number of shadows of the user. In this case, although each shadow will fall on a distinct set of sensors depending on its direction, the patterns of change in the intensity of light seen by each set of sensors after standardization are the same as they arise from the same user. This implies that in all the

Figure 2: 3D plot of SWT Coefficients for Clap

above cases, the same pattern appears redundantly across a varying number of sensors. Figure 2 shows an example, where a 3D plot of SWT coefficients of a clap gesture sample has 3 black lines that are very similar in shape to each other as they correspond to 3 sensors that see 3 redundant patterns. Thus, if we combine the time-series of all sensors in a way that captures the redundant pattern irrespective of the set of sensors on which the shadow falls, then the change in the position of the light source or user does not have an impact. To do so, we observe that if we take a projection of this 3D space onto the 2D x, z -plane, the projection of each of these three lines will almost exactly lie on the same points on the x, z -plane as shown in Figure 3 *i.e.*, the projection essentially contains a single contribution from these three lines. We call the x, z -plane a raster. LiGest thus performs rasterization by mapping the values of the SWT coefficients onto a 2D binary raster image, such that the mapping of each coefficient depends on the number and value of the coefficient.

3) Feature Compression: To further reduce the size of the raster, LiGest applies a 2D DWT to obtain a compressed image. Next, it minimizes redundant information in the compressed image by applying principal component analysis

(PCA) on this compressed raster to get an even smaller set of features to train its classification models.

2.2.3 Classifier Training and Recognition. After extracting the features from all training samples of a given set of gestures, in this step, LiGest builds support vector machine based classification models for that set of gestures. LiGest then uses these classification models to recognize gestures of a user at runtime.

3 PERFORMANCE EVALUATION

Next, we evaluate LiGest's performance using data collected from 20 users totaling 15175 samples for 5 gestures across 9 positions, 4 orientations and 6 lighting conditions in the center of a 25ft $\times$ 16ft room. We measure the performance in terms of accuracy, defined as the percentage of samples that LiGest recognizes correctly in a given set that contains unseen samples of the 5 gestures.

We then use 10-fold cross validation on the collected samples to demonstrate LiGest's accuracy. Figure 4 shows the resulting confusion matrix for all gestures, from which we see that LiGest achieves an average

Clap	0.92	0.03	0.00	0.04	0.00
Hug	0.03	0.95	0.00	0.02	0.00
Jump	0.00	0.00	0.99	0.00	0.00
Punch	0.03	0.01	0.00	0.96	0.00
Step	0.00	0.00	0.00	0.00	0.99
	Clap	Hug	Jump	Punch	Step

Figure 4: Confusion matrix

accuracy of 96.36%. The accuracy is fairly stable around all the gestures, despite the variations in position, orientation, lighting and users. Further, the clap gesture shows the least accuracy of 92% due to the shadow of the hands sometimes being completely blocked by the body.

4 CONCLUSION

In this paper, we proposed LiGest, an ambient light based gesture recognition system, which is robust against changing lighting conditions, user positions and orientations, and users. LiGest achieves an average accuracy of 96.36%, which is comparable to the average accuracies of existing RF based gesture recognition systems.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation, under Grant # CNS 1565609.

REFERENCES

- [1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *Proceedings of IEEE INFOCOM*.
- [2] Maarten Jansen. 2012. *Noise reduction by wavelet thresholding*. Vol. 161. Springer Science & Business Media.
- [3] Raghav H Venkatnarayan and Muhammad Shahzad. 2018. Gesture Recognition Using Ambient Light. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 40.
- [4] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 65–76.