TIME REVERSAL BASED ROBUST GESTURE RECOGNITION USING WIFI

Sai Deepika Regani[*†], *Beibei Wang*[*†], *Min Wu*[*†], *K. J. Ray Liu*[*†]
[*]University of Maryland, College Park, MD 20742, USA
[†]Origin Wireless Inc., 7500 Greenway Center Drive, Suite 1070, MD 20770, USA
Email: rdeepika@terpmail.umd.edu, {minwu, kjrliu}@umd.edu, beibei.wang@originwireless.net

ABSTRACT

Gesture recognition using wireless sensing opened a plethora of applications in the field of human-computer interaction. However, most existing works are not robust without requiring wearables or tedious training/calibration. In this work, we propose *WiGRep*, a time reversal based gesture recognition approach using Wi-Fi, which can recognize different gestures by counting the number of repeating gesture segments. Built upon the time reversal phenomenon in RF transmission, the Time Reversal Resonating Strength (TRRS) is used to detect repeating patterns in a gesture. A robust low-complexity algorithm is proposed to accommodate possible variations of gestures and indoor environments. The main advantages of *WiGRep* are that it is calibration-free and location and environment independent. Experiments performed in both line of sight and non-line-of-sight scenarios demonstrate a detection rate of 99.6% and 99.4%, respectively, for a fixed false alarm rate of 5%.

***Index Terms*—** Gesture recognition, Wireless sensing, Channel state information, TRRS, Time reversal.

1. INTRODUCTION

In recent years, gesture recognition has been considered as a new dimension to human-computer interaction and the number of possible applications is inconceivable [1]. Among the many attempts to achieve robust gesture recognition using device-free approaches, sensor-based approaches [2, 3] and camera-based approaches [4] became superior due to their reliable performance while at the cost of user convenience and scalability. Due to the ubiquitous deployment of wireless devices, wireless sensing-based gesture recognition approaches have gained increasing attention.

Early attempts have used features extracted from the received signal strength indication (RSSI) to differentiate between gestures [5]. Since the RSSI is very sensitive to indoor environment changes and lacks fine-grained information about the channel profile, researchers turn to the more fine-grained Channel State Information (CSI) readily available from 802.11 ac devices, and developed many new applications [6] such as breathing detection [7], indoor localization [8], human authentication [9], gait recognition [10], driver monitoring [11] and more.

Most of the current gesture recognition approaches using CSI are learning-based, where patterns are extracted from the CSI time series and matched with those obtained in the training phase, either by using machine learning techniques or by similarity metrics [12, 13]. A similar idea has been exploited in [14, 15] which further adopted the dynamic time warping (DTW) technique to recognize fine-grained finger gestures and keystrokes. *WiCatch* considered the angle of arrival extracted using the smooth-MUSIC algorithm to recognize one and two hand gestures [16]. However, most of the approaches suffer from the "location-dependency" problem. This is because the CSI is sensitive to small changes in the environment, and when the environment or the location/orientation of the gesture or the transceiver changes, the performance will be degraded significantly, which is called the "location dependency" problem. In addition, utilizing simple gestures can potentially cause a lot of false alarms in a practical scenario. A few other works used specialized hardware like USRPs and antenna arrays to recognize gestures [17, 18]. However, specialized devices may not be readily available and could incur additional costs.

Different from most existing works that try to determine the exact nature of a gesture assuming a specific gesture is mapped to a unique command/operation, in this work we propose *WiGRep* (Wireless Gesture Repetition counter) that defines different number of repetitions of a specific gesture (e.g., hand waving) as different gestures. *WiGRep* does not require any training and only needs to estimate the number of times a gesture segment is repeated. It overcomes the "location-dependency" issue and reduces false alarm greatly in a practical set up.

The main contributions of this work can be summarized as follows:

- We propose *WiGRep*, a robust gesture recognition approach that can distinguish different gestures by counting the number of repeating gesture segments. Since the proposed approach relies on the number of repetitions, we do not impose any restrictions on the location, environment or type of gesture. Also, the approach does not require any training and can work in both line of sight (LOS) and non-line-of-sight (NLOS)

scenarios.

- We define an effective metric to characterize the similarity of the CSI collected when a gesture is repeated and propose a low-complexity algorithm to automatically extract the number of gesture repetitions. This allows *WiGRep* to be implemented in real-time.
- We perform extensive experiments to demonstrate the capability of *WiGRep* in various scenarios and potential applications.

The rest of the paper is organized as follows. Section II presents the preliminary knowledge of TRRS. The *WiGRep* algorithm is discussed in Section III. Section IV presents the experimental evaluation and potential applications of *WiGRep*. Finally, Section V concludes the paper.

2. TIME REVERSAL RESONATING STRENGTH

In this work, we define the different numbers of repetitions of a specific gesture segment as different gestures. Thus, the problem of gesture recognition is converted to estimating the number of gesture segment repetitions. To determine if a segment is repeated, a similarity metric on CSI needs to be defined. Inspired by the time reversal focusing effect [7, 9], we adopt time reversal resonating strength (TRRS) as the similarity measure that can be defined as:

$$TRRS(\boldsymbol{H}(t_1), \boldsymbol{H}(t_2)) = \frac{|\sum_{k \epsilon \nu} H_k(t_1) H_k(t_2)^*|}{||\boldsymbol{H}(t_1)||_2 ||\boldsymbol{H}(t_2)||_2}, \quad (1)$$

where $\mathbf{H}(t_1)$ and $\mathbf{H}(t_2)$ are the channel frequency responses at times t_1 and t_2 respectively, $(.)^*$ is the complex conjugate operator, $||.||_2$ is the L2 norm and ν is the set of subcarrier indices. The TRRS values range from 0 to 1, with a value of 1 achieved by a pair of identical CSIs.

Each gesture can be captured as a CSI time series. To investigate the correlation/similarity between CSIs at every two time instances, a TRRS matrix is built, whose $(i, j)^{\text{th}}$ entry corresponds to the TRRS between CSIs at the i^{th} and j^{th} time instances. As an example, let us consider a gesture consisting of 3 turns and 4 repeating gesture segments as shown in Fig. 1, where a "turn" is said to have taken place when the hand movement reverses direction and begins to retrace the previous path. When a gesture is performed, due to the movement of the hand, the multipath environment is altered. An illustration of such a multipath profile change is shown in Fig. 2a, where scatterers S_1 and S_2 are static while S_3 (hand) is dynamic. Now, as the hand moves, the multipaths involving S_3 are changed from time t_1 to t_2. These changes impact the CSI time series that essentially gets reflected in the TRRS matrix according to (1), as shown in Fig. 2b.

Let us observe the TRRS matrix formed from the CSI time series of the gesture, which is symmetric by construction. The

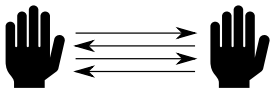

Fig. 1. An example of a gesture.

diagonal entries correspond to the TRRS between CSIs at the same time instance and are thus equal to 1. Among the non-diagonal entries, a relatively high value indicates a match between the CSIs that occurs when the dynamic scatterer(i.e., hand) is in the same position at the corresponding time instances. In a typical gesture, the hand follows the same path just before and after a "turn", which results in a symmetric CSI time series about the point of a "turn". These matching pairs of CSIs result in relatively high values of TRRS, which form symmetric "cross" patterns with the main diagonal of the matrix, as can be visualized from Fig. 2b. The time instances at the intersection points of the cross patterns correspond to the "turn" events. Then, counting the number of repeated gesture segments can be achieved by estimating the number of "turn" events, which will be discussed in the next section.

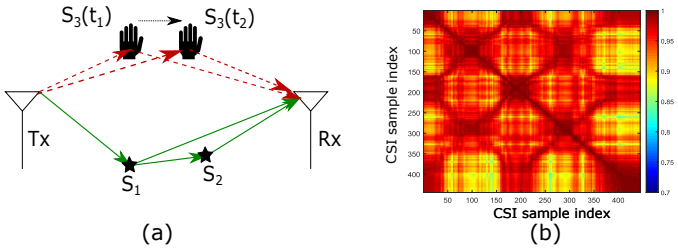

Fig. 2. Demonstration of changes in CSI due to a gesture. (a) Multipath profile change, (b) TRRS matrix.

3. WIGREP ALGORITHM

We have seen that the cross patterns in the TRRS matrix correspond to the turn events. Hence, the problem now reduces to finding the number of cross patterns, which is solved using the following three steps.

In the first step, the points which form the non-diagonal lines in the cross patterns of the TRRS matrix are extracted. Each point in these lines represents a pair of matching CSIs around a turn and are termed as "matching points". These points are extracted by using a moving window as shown in Fig. 3a, that gathers the maximum value in the window. The resulted matching points are shown in Fig. 3b. The window length is controlled by the tunable parameter α, which is the minimum number of time instances separating any two turns. This parameter needs to be carefully chosen, as a very large value of α tends to include matching points corresponding to the adjacent cross pattern, while a very small value of α

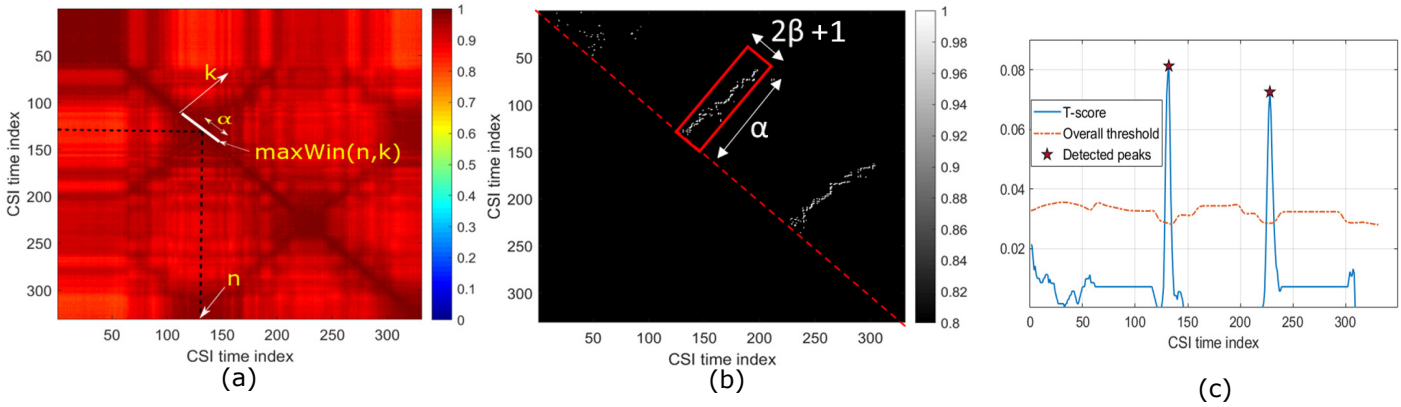

Fig. 3. Visualization of *WiGRep* algorithm on the TRRS matrix. (a) Moving maximum window to select "matching points", (b) Selected matching points and the moving sum window to calculate the T-$score$, (c) T-$score$, adaptive overall threshold and the detected peaks which correspond to the "turns" in the gesture.

might result in false alarms due to detection of false matching points. The moving window on the TRRS matrix (**T**) can be explicitly written as:

$$\textbf{maxWin}(n,k) = [\mathbf{T}(n+w-k, n+w+k)], \qquad (2)$$

where -$\alpha \leq w \leq \alpha$, $1 \leq k \leq \alpha$ and n is the time index. In each window, the maximum value is extracted only if it is greater than the average of the values in the window by a predefined threshold r_{th}. For each time instance n and $1 \leq k \leq \alpha$, let $\mathbf{T}(n\text{-}k,n\text{+}k) = p$. We define a score$(n,k)$ as follows:

$$score(n,k) = \begin{cases} p, & if \quad p \geq max(\boldsymbol{maxWin}(n,k)) \quad \& \\ & \quad p \geq mean(\boldsymbol{maxWin}(n,k)) + r_{\text{th}} \\ 0, & else. \end{cases} \qquad (3)$$

The threshold r_{th} above the mean of the entries in the moving window, reduces the false detections in corner cases such as a static environment, in which all the entries are high.

In the second step, a score which indicates the chance of a time instance being a turn is computed. This is termed as the T-$score$ and is calculated by computing a sum of the scores calculated in the first step, using a moving window method as,

$$T-score(n) = \sum_{p=-\beta}^{\beta} \sum_{k=1}^{\alpha} score(n+p,k). \qquad (4)$$

where a window of width β and height α is used as shown in Fig. 3b. The choice of the parameter β is crucial to robustly capture turns that involve differences in the speed of the gesture segments. A very low value of β can only detect turns in which the gesture speed is uniform. On the other hand, a very high value of β may cause false alarms of turns. This implicit design of the T-$score$ metric captures the total number of matching CSIs around any time instance. From Fig. 3b, we can see that there are a large number of matching points near the turn events. Thus, the higher the T-$score$, the higher the chance of the corresponding time instance being a turn.

All the significant peaks of the T-$score$ profile indicate potential turn events. Thus as the final step, we detect such significant peaks in the T-$score$ profile by adopting the CFAR thresholding [19] approach. In this approach, a CFAR window is designed as follows:

$$C_{\text{win}} = [\underbrace{1,1,..,1}_{\alpha}, \overbrace{\underbrace{0,..,0}_{\alpha}}^{10\beta}, \underbrace{1,1,..,1}_{\alpha}]. \qquad (5)$$

The overall threshold O_{th} is calculated as:

$$O_{\text{th}} = C_{\text{noise}} + c_{\text{th}}, \qquad (6)$$

where $C_{\text{noise}} = C_{\text{win}} * T$-$score$, * is the convolution operator, and c_{th} is the additional threshold parameter that is chosen in accordance with the acceptable false alarm rate. In our experiments, we used a value of $c_{\text{th}} = 0.025$ which resulted in a false alarm of about 5%. With this setting, Fig. 3c shows the T-$score$ profile, the overall CFAR threshold and the detected peaks which correspond to the locations of the cross patterns and also the locations of turn events in a gesture. The number of gesture repetitions is simply the number of turns plus 1.

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of *WiGRep* in both the LOS and NLOS scenarios.

4.1. Experiment results

In our experiments, commercial Wi-Fi devices operating on 5.845 GHz carrier frequency with a bandwidth of 40 MHz and 3x3 MIMO transmission have been used. The devices were placed in the 10^{th} floor of an office building in (A) LOS and (B) NLOS scenarios as shown in Fig. 4.

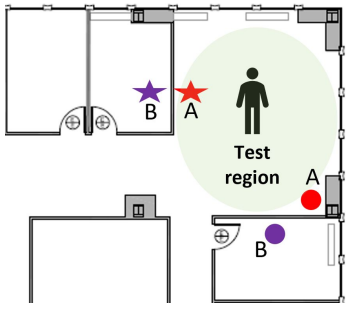

Fig. 4. Experimental setup.

A total of 50 gestures with 10 repetitions were collected for each scenario using a sounding rate of 200 Hz, gathering a total of 500 repetitions per scenario. These gestures were performed in different locations that are randomly chosen in the test region indicated in Fig. 4. The parameters α, β and r_{th} are fixed at $\alpha = 70$, $\beta = 2$ and $r_{\text{th}} = 0.3$ in our experiments. By varying the additional threshold parameter c_{th}, we obtain the ROC curves for the detection of these 500 repeating gesture segments, which are shown in Fig. 5. The detection rates for the LOS and NLOS scenarios are 99.6% and 99.4% respectively for a fixed false alarm rate of 5%. The area under the ROC curve (AUC) for the LOS and NLOS scenarios is 0.99934 and 0.99926 respectively, indicating an effective gesture repetition detector in both the cases.

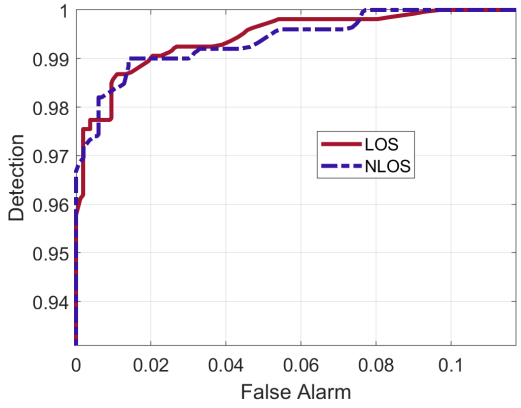

Fig. 5. ROC curves for varying c_{th} in LOS and NLOS scenarios.

4.2. Discussions

WiGRep's ability to recognize gesture repetitions can be utilized for many applications. Here are two examples.

1. Gesture recognition: Fig. 6 shows the confusion matrix for five gestures classified based on the number of repeating segments. The detection rate decreases with the number of repetitions as the probability of missing a repetition/turn increases.

2. *WiGRep* can be used to input characters. Similar to *WiMorse* [20], two gestures can be used to encode "dot" and "dash", and using the International Morse code, characters can be formed. The codes for "dot" and "dash" can be separated by defining "silent" periods between them.

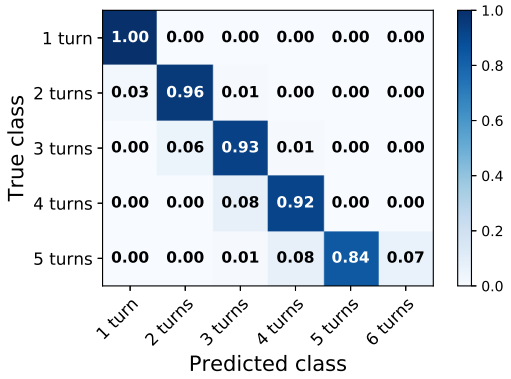

Fig. 6. Confusion matrix for five gestures.

One main limitation of WiGrep is that it can only handle gestures which are "periodic" repetitions and requires careful calibration when there are other repeating motions in the environment. As a future work, we will study the impact of different experimental conditions, background motions, parameter settings, and with different users.

5. CONCLUSION

We propose *WiGRep*, a time reversal based gesture recognition system that classifies gestures based on the number of repeating patterns. Relying on the number of repetitions allowed us to create location and environment independent features, overcoming the major drawback of most of the existing gesture recognition systems. A detection rate of 99.6% and 99.4% has been achieved in the LOS and the NLOS scenarios respectively, for the repeating gesture segments. With *WiGRep*, we envision a robust and user-friendly gesture recognition system that can be integrated into the future smart homes.

6. REFERENCES

[1] Hongyi Liu and Lihui Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[2] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-Han Kim, "Typingring: A wearable ring platform for text input," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 227–239.

[3] Jue Wang, Deepak Vasisht, and Dina Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 235–246, 2015.

[4] Thad Starner and Alex Pentland, "Real-time american sign language recognition from video using hidden Markov models," in *Motion-Based Recognition*, pp. 227–243. Springer, 1997.

[5] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[6] Beibei Wang, Qinyi Xu, Chen Chen, Feng Zhang, and KJ Ray Liu, "The promise of radio analytics: A future paradigm of wireless positioning, tracking, and sensing," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 59–80, 2018.

[7] Chen Chen, Yi Han, Yan Chen, Hung-Quoc Lai, Feng Zhang, Beibei Wang, and K.J. Ray Liu, "TR-BREATH: Time-reversal breathing rate estimation and detection," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 3, pp. 489–501, 2017.

[8] Chen Chen, Yan Chen, Yi Han, Hung-Quoc Lai, and K.J. Ray Liu, "Achieving centimeter-accuracy indoor localization on WiFi platforms: A frequency hopping approach," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 111–121, 2016.

[9] Qinyi Xu, Yan Chen, BeiBei Wang, and K.J. Ray Liu, "Radio biometrics: Human recognition through a wall," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1141–1155, 2017.

[10] Wei Wang, Alex X Liu, and Muhammad Shahzad, "Gait recognition using WiFi signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 363–373.

[11] Shihong Duan, Tianqing Yu, and Jie He, "WiDriver: Driver activity recognition system based on WiFi CSI," *International Journal of Wireless Information Networks*, vol. 25, no. 2, pp. 146–156, 2018.

[12] Wenfeng He, Kaishun Wu, Yongpan Zou, and Zhong Ming, "WiG: WiFi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–7.

[13] Mohammed Al-qaness and Fangmin Li, "WiGeR: WiFi-based gesture recognition system," *ISPRS International Journal of Geo-Information*, vol. 5, no. 6, pp. 92, 2016.

[14] Sheng Tan and Jie Yang, "WiFinger: leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*. ACM, 2016, pp. 201–210.

[15] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad, "Keystroke recognition using WiFi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 90–102.

[16] Zengshan Tian, Jiacheng Wang, Xiaolong Yang, and Mu Zhou, "WiCatch: A WiFi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16911–16923, 2018.

[17] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 142, 2016.

[18] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 27–38.

[19] Mark A. Richards, *Fundamentals of radar signal processing*, Tata McGraw-Hill Education, 2005.

[20] Kai Niu, Fusang Zhang, Yuhang Jiang, Jie Xiong, Qin Lv, Youwei Zeng, and Daqing Zhang, "WiMorse: A contactless Morse code text input system using ambient WiFi signals," *IEEE Internet of Things Journal*, 2019.