Design and Implementation of Message Exchanging System Based on Short-range Wireless Communication and Gesture

Niwat Thepvilojanapong*, Ryo Kanaoka§, Hiroki Saito‡, Takahashi Junji†, and Yoshito Tobe†
* Department of Information Engineering, Mie University
1577 Kurimamachiya-cho, Tsu city, Mie 514–8507 Japan
E-mail: wat@net.info.mie-u.ac.jp
§ Department of Science and Engineering, Aoyama Gakuin University, Japan
5–10–1 Fuchinobe, Chuo-ku, Sagamihara city, Kanagawa 252–5258 Japan
Email: ryoryo@rcl-aoyama.jp
‡ School of Interdisciplinary Mathematical Sciences, Meiji University
4–21–1 Nakano, Nakano-ku, Tokyo 164–8525 Japan
E-mail: hrksaito@meiji.ac.jp
† Department of Integrated Information Technology, Aoyama Gakuin University
5–10–1 Fuchinobe, Chuo-ku, Sagamihara city, Kanagawa 252–5258 Japan
E-mail: takahashi@it.aoyama.ac.jp, yoshito-tobe@rcl-aoyama.jp

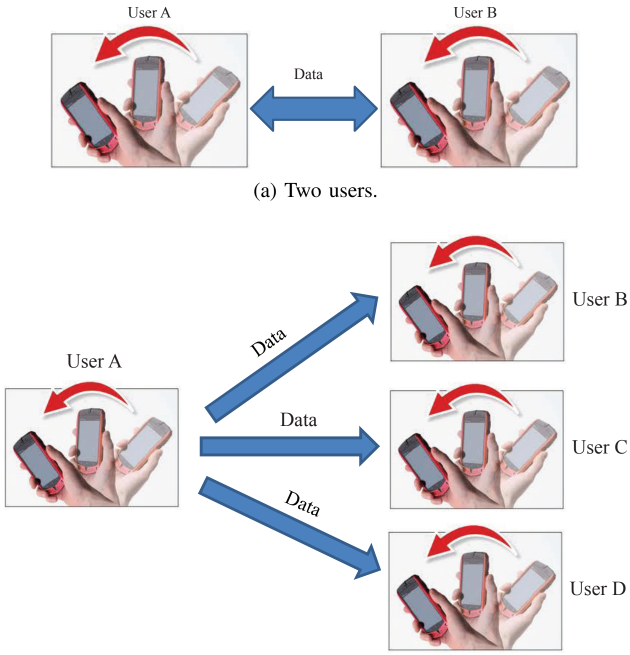
***Abstract*—The number of mobile devices equipped with short-range wireless communication modules such as Bluetooth, infrared and near field communication (NFC) is increasing rapidly. People always carry the devices and use them for various purposes in their daily lives. For example, mobile users may exchange electronic business card or contact address through infrared or Bluetooth. To establish a connection between two devices, the users have to specify the communicating party by using some kinds of identifiers, e.g., device's ID, name or MAC address. Since communicating parties are in a nearby area, it would be convenient if we can exploit such physical adjacency to establish the communication instead of using conventional addressing scheme or complicated setup. Thus this paper proposes *EriCC*, a gesture-based session establishment for multi-user data exchange. EriCC uses gesture to identify and verify a communicating party. In particular, the users do the same gesture in order to establish a connection without knowing the other party's address. EriCC then transforms the acceleration signal of the gestures into the frequency domain by applying Discrete Fourier Transform (DFT). If frequency components of the acceleration signal are below a threshold, data exchange is allowed. We implemented EriCC on Android smartphones and conducted experiments to study its performance and practicability. The experimental results show that EriCC is able to remove undesirable receivers from actual receivers with high recall and reasonably high precision.**

I. Introduction

Recently, we have seen the proliferation of mobile devices including smartphones, tablet PCs and portable game players [1]–[3]. People always carry such devices with themselves all the time. In addition to cellular communication capability (2G/3G/4G) as a standard feature, the mobile devices are equipped with short-range wireless communication modules, that is, Bluetooth, Wi-Fi 802.11 a/b/g/n, infrared, near field communication (NFC), and various kinds of sensors, that is, accelerometer, gyroscope, magnetometer, compass, GPS, barometer, microphone, ambient light, camera etc. Mobile users are likely to exploit the short-range communication to exchange small-sized data such as electronic business cards or contact address with nearby people. The benefit of using electronic data is convenient management and reducing paper consumption. To establish the communication session, the users specify the communicating party by using some kinds of identifiers such as device's ID, name or MAC address. In particular, the users have to acquire such identifiers by some means. In the most cases, the users directly ask the identifier from the communicating party which is likely to be the easiest method. However, such inconvenient processes of querying the identifier and establishing the session obstruct people from using short-range wireless communication for data exchange.

To encourage the usage of short-range wireless communication, this paper introduces a new method to establish a session without using conventional addressing scheme or complicated setup. Since communicating parties are in a nearby area, it would be convenient if we can exploit such physical adjacency to establish the session. We realize that gesture-based interaction is as a natural way for human-computer interaction according to a wide range of ubiquitous applications [4]–[7]. Thus we propose *EriCC*, a gesture-based session establishment for multi-user data exchange. EriCC uses gesture to identify and verify a communicating party instead of conventional address. In particular, the communicating parties do the same gesture in order to establish a session and exchange data without knowing the other party's address (see Figure 1a). EriCC transforms the acceleration signal of the gesture into the frequency domain by applying Discrete Fourier Transform (DFT), and then compares whether the gestures are identical. The users whose the gestures are identical are allowed to exchange data. This mechanism can also be applied to a group communication, i.e., all the users having the same gesture as the sender can receive the sender's data (see Figure 1b). We implemented EriCC on Android smartphones

DOI : 04.IJAI.2015.1.1



(a) Two users.

(b) Group communication for multiple users.

Fig. 1. Session establishment for data exchange.

and conducted experiments to study its usefulness. The results of gesture comparison show that we can achieve high recall and reasonable precision when establishing a session.

The remainder of this paper is organized as follows. Section II discusses related work. Section III details the session establishment employed by EriCC. Section IV describes prototype implementation and experimental methodology. The experimental results are analyzed in Section V. We conclude our paper and give some future works in Section VI.

II. Related Work

Smart-Its' users take two devices they wish to connect and move them together to establish a connection [4]. If the difference of time-series accelerometer data is below a specified threshold then it accepts the other as friend and establishes a dedicated connection. Though the interface is easy to use, Smart-Its requires both communicating parties must be in contact physically because two devices must be hold and moved by one person. In contrast with Smart-Its, physical contact does not need by EriCC such that the movement (or gesture) does not need to be carried out at the same time. In addition, EriCC analyzes acceleration signals in the frequency domain in order to recognize the gesture.

Touch-and-Connect framework [5] provides two kinds of buttons on devices, that is, a plug button and a socket button. A user can request a connection between any pair of devices by simply pushing the button on the devices. In particular, the user first pushes the plug button of the source device, then she/he pushes the socket button of the destination device. The state of the buttons is represented by the color of the button. The interface is user-friendly but a dedicated hardware is necessary to establish a connection. In contrast, EriCC, which is independent of hardware, can determine a communicating party by using a smartphone equipped with accelerometer.

u-Photo [8] uses ARToolKit [9] to realize image-based interaction in ubiquitous environment. First, users take a photo of a target device which is attached with a marker. Then the application recognizes the target devices from the marker. As a result, u-Photo can determine communicating devices, e.g., determining a printer to transfer a file for printing service. Snappy [10] is an interaction method to select a device by swinging a mobile device following the swing reference code of each device. The swing reference code is unique such that users can determine target devices. However, both approaches require users to attach markers to all target devices in advance.

Bump application allows transferring data between smartphones and/or computers [7]. To transfer data between a pair of smartphones, a user holds the phone and gently bumps hands with the other person. Bump consists of two parts: (i) the application running on smartphones and (ii) a smart matching algorithm running on servers in the cloud. The application on smartphones uses embedded sensors to literally "feel" the bump, and it sends that info up to the cloud. The matching algorithm listens to the bumps from phones around the world and pairs up phones that felt the same bump. Then the information will be routed between the two devices in each pair. GPS-based location information is necessary in order to limit the number of other phones the servers have to check for the correct match. Similar to Bump, LINE [11] uses a combination of gestures, acceleretion data, GPS-based location and IP address to determine a communicating party. All information are transmitted to a server for processing. In contrast with Bump and LINE, EriCC does not require location information and a central server. GPS information is not available indoor and the central server may receive too many requests from users for real-time processing.

Many approaches have been proposed to recognize a gesture from acceleration signals. Such approaches adopt several techniques including DTW (dynamic time warping) [12], HMM (hidden Markov model) [13], [14] and SVM (support vector machine) [15]. Instead of recognizing exact gestures as the above research, the purpose of EriCC is to determine similarity of two gestures. A requirement of EriCC is fast processing on smartphones without training phase of recognition such that we get the results immediately.

III. EriCC: Multi-User Data Exchange

EriCC aims to establish a connection for exchange small-sized data, e.g., electronic business card, between multiple users. Users who do the same gesture are allowed to exchange data without knowing the device address of the other party. This section details the design of EriCC.

A. Overview of EriCC

The system consists of a *provider* and *receivers* who aim to send and receive data, respectively A user can choose to be the provider or receiver after starting the EriCC application. In our implementation, EriCC uses Bluetooth as a short-range wireless communication medium. The communication between smartphones is based on the network sockets. Generally, the provider, who provides data, is likely to be a server and create the sockets on start up. However, EriCC employs a reverse approach, i.e., the receiver is a server while the provider is a

DOI : 04.IJAI.2015.1.1

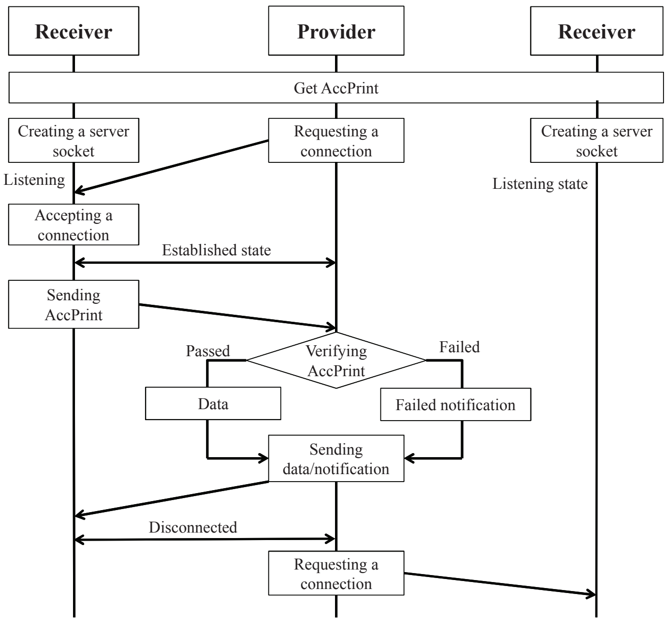


Fig. 2. Procedures of EriCC system.

client. As a result, the provider connects to the server sockets created by receivers.

The procedures to establish a session using socket communication is as follows (see Figure 2).

- Step 1 A provider and receivers do the same gesture with smartphones to capture accelerometer data. Then the smartphones calculate *AccPrint* based on the accelerometer data. The calculation of AccPrint is detailed in Section III-B.
- Step 2 Each receiver creates a server socket that is in listening state.
- Step 3 The provider sends a request to establish a connection with a desired receiver.
- Step 4 The receiver accepts the request and the socket is in established state.
- Step 5 The receiver then sends its AccPrint to the provider.
- Step 6 The provider verifies the receiver's AccPrint and its own AccPrint. The verification process is described in Section III-C. The action of provider depends on the result of the verification as follows.
  - If the verification successes, the provider sends data to the receiver.
  - Otherwise, the provider sends a notification of failed verification.
- Step 7 The provider disconnects from the receiver after finished data transfer. It then repeat from Steps 3 to 6, if the provider wants to send data to other receivers.

B. AccPrint: Gesture's Footprint

To characterize the action performed by a user, EriCC calculates *AccPrint (Acceleration Print)* based on acceleration signals recorded by the smartphone. Thus AccPrint can be considered as a footprint of gesture.

The user explicitly indicates the time of beginning and end by pushing buttons shown on a screen of smartphone, and moves the phone, e.g., waving, shaking, etc. during the duration. Let $a_x(t)$, $a_y(t)$ and $a_z(t)$ be the signal of the three-axis acceleration. First, EriCC calculates the magnitude of the time-series acceleration as expressed in Equation 1.

$$\|\mathbf{a(t)}\| = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}. \tag{1}$$

The time-series magnitude is transformed into a frequency domain $\mathbf{A[k]}$ by applying Discrete Fourier Transform. Though there are many spectrums from DFT, EriCC chooses two values, f_1 and f_2, of which the spectrum intensity is the largest and the second largest of all the spectrums. The pair of f_1 and f_2 or the 2–tuple (f_1, f_2) is called *AccPrint*, which represents the characteristic of user's action. Since AccPrint is expressed in frequency domain, strict time synchronization between smartphones is not required. One may argue that we can use other spectrums f_3, f_4, ... for AccPrint, but we observed that the pair of the two largest values is sufficient to distinguish different types of actions. We show the examples of spectrums in Section V.

C. Gesture Verification

When a provider receives AccPrint from a receiver, it verifies the AccPrint by comparing with its own AccPrint. The provider sends data to the only receiver whose AccPrint passes the verification.

Let (f_1^P, f_2^P) and (f_1^R, f_2^R) be the 2–tuple AccPrint of a provider and a receiver, respectively. The gesture of both provider and receiver is assumed to be identical if the difference of AccPrint is less than a threshold δ as expressed in Equations 2 and 3.

$$|\Delta f_1| = |f_1^P - f_1^R| \quad < \quad \delta. \tag{2}$$

$$|\Delta f_2| = |f_2^P - f_2^R| \quad < \quad \delta. \tag{3}$$

IV. Experiment Setup

In this section, we brief prototype implementation of EriCC, detail experimental methodology and define evaluation metrics.

A. Implementation

We used Android software development kit (SDK) [16] to implement EriCC on Samsung Galaxy S (Android OS 2.1), HTC Nexus One (Android OS 2.3.4) and Samsung Galaxy S II (Android OS 2.3.4). As mentioned earlier, wireless communications between smartphones are done through Bluetooth technology. Bluetooth employs a master-slave structure where a master may communicate with up to seven slaves in a piconet. However, the smartphones used in our experiments do not have a profile to create a piconet such that we use Serial Port Profile (SPP) to establish socket communications. We note here again that the provider is a client or slave while the receiver is a server or master when establishing a connection. A smartphone uses Service Discovery Protocol (SDP) to discover services offered by other smartphones. Each

DOI : 04.IJAI.2015.1.1

service is identified by a Universally Unique Identifier (UUID) which is determined in advance for EriCC application. A connection will be created if and only if the provider's UUID matches the receiver's UUID.

A receiver opens `BluetoothServerSocket` by calling `listenUsingRfcommWithServiceRecord(String, UUID)`, and then waits for a connection request by calling `accept()`. A provider requests a connection by calling `BluetoothSocket.connect()` method. The `BluetoothSocket` will be created, if the UUIDs match. `InputStream` and `OutputStream`, which are handler for socket communication, are acquired by calling `socket.getInputStream()` and `socket.getOutputStream()`. Then data read and write are done on both sides by `read()` and `write()` methods.

B. Experimental Methodology

We recruited three students in our university to serve as a provider, a receiver and an imitator in our experiments. The first experiment assumes the provider (the first student) want to send a file (e.g., a business card) to the receiver (the second student). Both start the EriCC application and shake their hands while holding a smartphone. The second experiment assumes the provider (the first student) want to send a file to the receiver (the third student), while the imitator (the second student) tries to get the file without the provider's permission. First, all start the EriCC application and hold their smartphone all the time. The imitator mimics the gesture of shaking hands while the provider and the receiver shake their hands. By setting the experiment like this, the imitator already had experience of shaking hands with the provider in the first experiment. Each experiment was repeated 100 times such that the experiments were done 200 times in total.

In addition to the above experiments, we conducted another experiment to measure delay of the EriCC system. We used two pairs of smartphones: (i) pair A: Samsung Galaxy S II (provider) and Samsung Galaxy S (receiver) and (ii) pair B: Samsung Galaxy S II (provider) and HTC Nexus One (receiver). The provider sent a small-sized file (117 bytes) to the receivers after the verification has passed. The small-sized file represents a digital business card according to our application scenario, though it may contain any kind of contents. Each pair of smartphones was done for 10 times and the time of each step was recorded for further analysis.

C. Performance Metrics

Since data can be exchanged if the persons carry out the same gesture, the verification of gesture (Equations 2 and 3) is the most important process of EriCC. We employ precision, recall, accuracy and F-measure to study the verification process of EriCC. The details of four metrics are as follows.

The verification between the provider and the receiver can be true positive or false negative, while the verification between the provider and the imitator can be true negative or false positive Let tp, fp, fn and tn be true positive, false position, false negative and true negative, respectively. The four metrics are defined below.

$$\text{Precision} = \frac{tp}{tp+fp} \tag{4}$$

$$\text{Recall} = \frac{tp}{tp+fn} \tag{5}$$

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn} \tag{6}$$

$$\text{F-measure} = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right) \tag{7}$$

Note that F-measure is F_1 measure where recall and precision are evenly weighted.

In addition, we also investigate delay of two processes and total delay. First, we consider the delay of session establishment by measuring the elapsed time between the socket is created by a receiver and the connection is established. This is the time used by the processed in Steps 3 and 4 summarized in Section III-A. This delay is completely affected by the Bluetooth specification and should be considered separately. Second, we study the delay of data transfer by measuring the elapsed time between a receiver sends AccPrint and a file has been received by the receiver completely. This is the time used by the processed in Steps 5 and 6 summarized in Section III-A. Note that, this delay includes the verification of AccPrint which is an additional process of EriCC before being able to send the data. Lastly, we measure total delay which includes all processes of EriCC. In addition to the above delays, the total delay also includes the time used to do gesture which varies widely from one gesture to another.

V. Experimental Results and Analysis

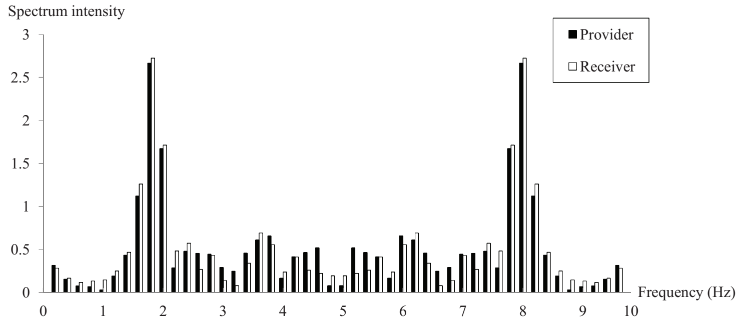
This section investigates the results of the experiments.

A. The Number of AccPrint Tuples

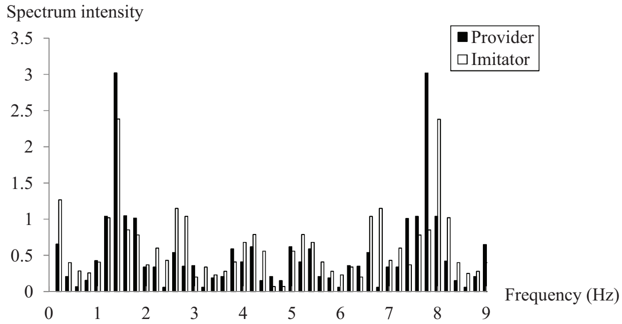
The acceleration signals of the receiver and the imitator compared with the provider's signal are illustrated in Figure 3. The figure shows the result of one experiment from 100 experiments. Note that the G-force (gravitational acceleration) is included in the figure because we do not know which axis is the G-force included and the effect of the G-force will be removed later when converting the acceleration signal from time domain to frequency domain. It is apparent from the figures that the signal of the receiver is similar to that of the provider while the imitator's signal differs explicitly.

By applying the discrete Fourier transform, the acceleration signals in Figure 3 are converted into frequency domain and shown in Figure 4. We removed the result at zero Hz from the figure because the value is quite large due to the G-force. The characteristics of spectrum intensity follow the time-series signal, that is, the spectrum intensities of the receiver including the largest and the second largest ones are similar to those of the provider while the spectrums of the imitator are different. Based on the result of one experiment, the two largest spectrums are likely to be sufficient to extract the receivers from the imitators.

Since the provider carried out the same gesture for 100 experiments, the two largest spectrum intensities are expected to have similar property. Thus we calculate the average AccPrint

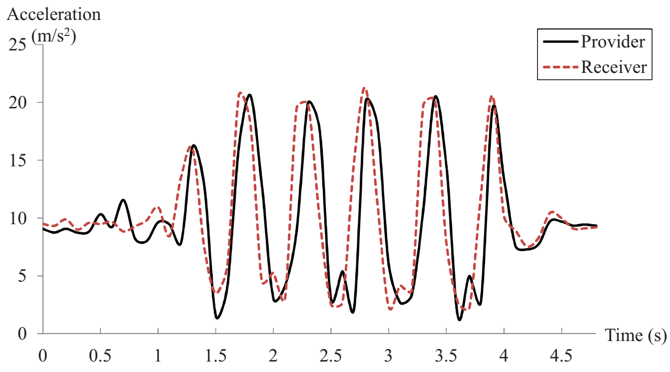


(a) Provider and receiver.

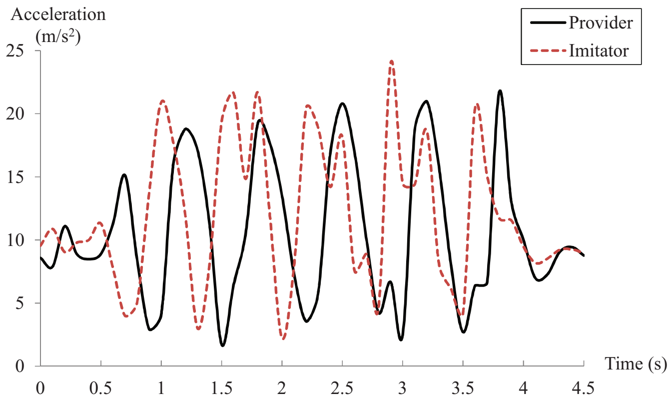


(b) Provider and imitator.

Fig. 4. Comparison of spectrum intensity: (a) similar intensity between the provider and the receiver, (b) different intensity between the provider and the imitator.



(a) Provider and receiver.



(b) Provider and imitator.

Fig. 3. Comparison of acceleration signals: (a) similar signals between the provider and the receiver, (b) different signals between the provider and the imitator.

TABLE I. AVERAGE VALUES OF ACCPRINT (f_1, f_2) FROM 100 EXPERIMENTS.

	Provider (Galaxy S)	Receiver (Galaxy S II)
$(\overline{f}_1, \overline{f}_2)$	(2.3, 1.8)	(2.3, 1.8)

	Provider (Galaxy S)	Imitator (Galaxy S II)
$(\overline{f}_1, \overline{f}_2)$	(2.2, 1.7)	(3.0, 2.0)

(f_1, f_2) of 100 experiments as shown in Table I. Accordingly, the differences of average AccPrint are shown in Table II. The two largest spectrums of the provider and the receiver

TABLE II. DIFFERENCES OF AVERAGE ACCPRINT FROM 100 EXPERIMENTS.

	$\lvert\Delta\overline{f}_1\rvert$	$\lvert\Delta\overline{f}_2\rvert$
Receiver	0.0	0.0
Imitator	0.8	0.3

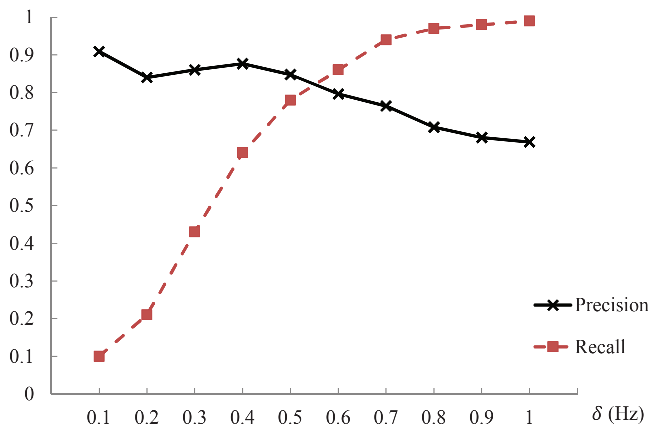


Fig. 5. Precision and recall when varying δ.

appear exactly at the same frequencies, while the two largest spectrums of the imitator does not match those of the provider. The experiments convince us that the two largest spectrums are sufficient to verify the receivers from the imitators.

B. *Verification Threshold*

Next we study the effect of choosing the value of δ for verification. The precision and recall of 200 experiments are shown in Figure 5 when varying δ from 0.1 to 1.0 with 0.1 increasing step. When δ increases from 0.1 to 1.0, the precision degrades slowly from 0.91 to 0.67 while the recall increases abruptly from 0.10 to 0.99. One would expect high precision and high recall in order to avoid the effect of false positive and false negative. The results, however, show that there is a trade-off between achieving high precision and high recall. Both metrics do not achieve the highest at the same choice of δ. To determine the optimal value of δ, we plot accuracy and F-measure in Figure 6. Both metrics achieve the highest, i.e., accuracy and F-measure are 0.83 and 0.84, respectively, when δ is set to 0.7. We conclude that 0.7 is the optimal δ where we achieve quite high recall (that is, 0.94) and reasonably high

DOI : 04.IJAI.2015.1.1
NNGT

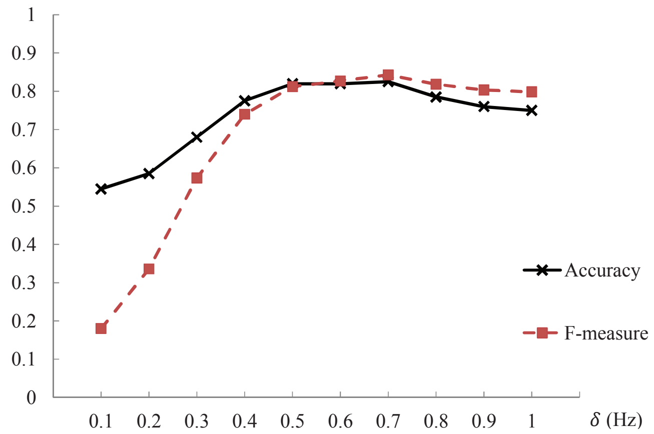


Fig. 6. Accuracy and F-measure when varying δ.

TABLE III. Median, average, standard deviation (SD) and 95% confidence interval (CI) of delays (s). The table shows delays of two pairs of smartphones: (i) Pair A is Galaxy S II and Galaxy S, (ii) Pair B is Galaxy S II and Nexus One.

	Median (s)	Average (s)	SD	95% CI
Session establishment				
Pair A	3.13	2.74	1.09	0.67
Pair B	3.54	3.45	1.05	0.65
Data Transfer				
Pair A	0.238	0.238	0.038	0.023
Pair B	0.251	0.259	0.076	0.047
Total delay				
Pair A	23.06	18.60	3.38	2.96
Pair B	26.35	23.55	2.31	2.02

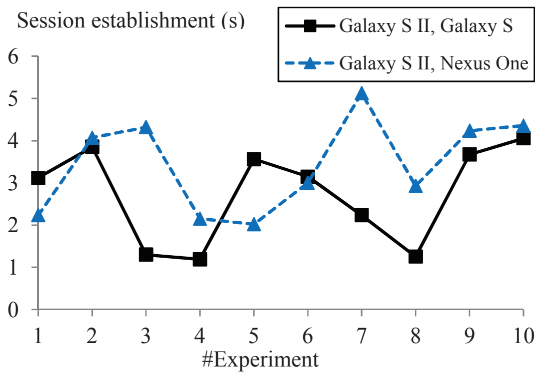


Fig. 7. Delay used for establishing a session.

precision (that is, 0.76). However, an appropriate value of δ may depend on applications and the purpose of usages. Users can freely determine the value by themselves and change it when necessary.

C. Delay

Table III summarizes the median, average, standard deviation (SD) and 95% confidence interval (CI) of three kinds of delays we consider. The results are from two pairs of provider and receiver, where 10 experiments were carried out for each pair. First, let us consider session establish where Figure 7 shows variation of the delays from one experiment to another. The delay highly fluctuates between 1.19 to 5.13 seconds as indicated by standard deviation of 1.09 seconds. It is explicit from the figure that pair A is faster than the other

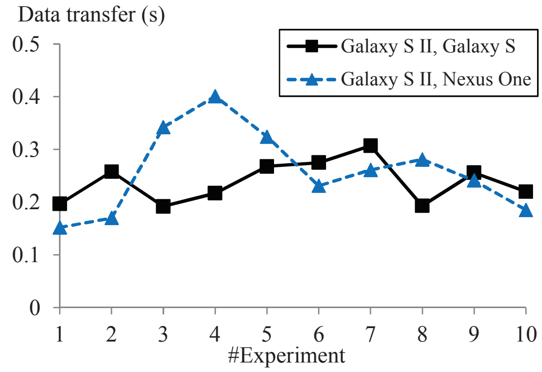


Fig. 8. Delay used for data transfer.

both average (26% faster) and median. The results show that a session is established within five seconds in the most cases. Since we use Bluetooth as a short-range communication, the results highly depend on the Bluetooth specification though it also affected by the devices and Bluetooth versions. We would like to note that the results are not a significant metric for directly evaluating EriCC, but it is necessary to evaluate practicability of EriCC based on Bluetooth technology. Other short-range technologies can be used instead of Bluetooth. We convince that users can tolerate the delay of five seconds where newer version of Bluetooth can reduce the delays both session establishment and data transfer. Note that Galaxy S II and Galaxy S come with Bluetooth 3.0, while Nexus One uses Bluetooth 2.1.

Figure 8 shows variation of the delays used to transfer 117-byte data. Similar to session establishment, pair A is faster than pair B both average (9% faster) and median. This may be pair A uses newer version of Bluetooth. Though it depends on data size, data transfer took less than 10 times of session establishment. It means EriCC is practical to share larger file size with small amount of additional delay because session establishment is fixed cost. In addition, standard deviation shows that the delay of data transfer is more stable than those of session establishment.

Table III also shows total delay which includes the time used to do gestures (shaking hands in our experiment) and calculate AccPrint. It took between 13.6 to 26.4 seconds for overall processes. It less than half a minute which is tolerable. In the case of exchanging name cards, both parties are assumed to have a short conversion which is much longer than the total delay of EriCC. The results shows the similar trends, i.e., pair A is faster than the other.

VI. Conclusions and Future Works

This paper has proposed EriCC, a gesture-based session establishment for multi-user data exchange. A benefit of EriCC is to establish connections between several devices without a requirement of conventional addressing scheme, complicated setup or dedicated hardware. In particular, the receivers merely imitate the action or gesture of the provider in order to establish a connection. EriCC provides a user-friendly interface such that anyone including novices and elderly people can use it easily. Several receivers can do the same gesture in order

DOI : 04.IJAI.2015.1.1

to receive information from the same provider, i.e., one-to-many or multicast communication is possible. The gesture do not need to carry out at the same time as the provider. However, the verification, which takes very short period, needs to be processed one by one. We implemented EriCC on Android smartphones and recruited subjects in order to conduct experiments. We found that AccPrint, which is the footprint of a gesture, can be generated in real-time manner. The recognition and verification of a gesture between two users is fast and does not require a training phase. Based on the experimental results, the verification of EriCC is able to achieve high recall and reasonably high precision. The optimal threshold for verification (δ) is 0.7, but users can also choose their own δ that suits their applications and purposes. The results of measured delay also show that EriCC is a practical application for exchanging data. Though current EriCC uses Bluetooth as a wireless medium, other short-range wireless technologies including NFC [17], [18], ZigBee [19], [20] and iBeacon [21] are available.

One of future works is to improve the verification in order to achieve higher precision while maintaining high recall. We also plan to conduct a comprehensive experiment by recruiting at least 10 subjects, though we believe that the results will show the same trend as presented in this paper. In addition, questionnaires will be used for subjective evaluation.

REFERENCES

[1] R. T. Llamas, "Worldwide smartphone 2015–2019 forecast and analysis," IDC Corporate USA, Tech. Rep., Mar. 2015, http://www.idc.com/getdoc.jsp?containerId=254912 (Accessed: Jun 28, 2015).

[2] M. Meeker and L. Wu, "2012 Internet trends," Kleiner Perkins Caufield & Byers, May 2012, http://www.kpcb.com/blog/2012-internet-trends (Accessed: Jun 28, 2015).

[3] A. Mohamud, "EU5 smartphone penetration reaches 55 percent in October 2012," comScore Press Release, Dec. 2012, http://www.comscore.com/Insights/Press_Releases/2012/12/EU5_Smartphone_Penetration_Reaches_55_Percent_in_October_2012 (Accessed: Jun 28, 2015).

[4] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-Its friends: A technique for users to easily establish connections between smart artefacts," in *Proceedings of the 3rd international conference on Ubiquitous Computing (UbiComp 2001)*. Atlanta, Georgia, USA: Springer-Verlag, 2001, pp. 116–122.

[5] Y. Iwasaki, N. Kawaguchi, and Y. Inagaki, "Touch-and-connect: A connection request framework for ad-hoc networks and the pervasive computing environment," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (Percom 2003)*, Fort Worth, Texas, USA, 2003, pp. 20–.

[6] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: a ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70–77, January-March 2006.

[7] "Bump - easily transfer photos, files and contacts between your phone and computer," https://bu.mp/ (Accessed: November 11, 2013).

[8] G. Suzuki, S. Aoki, T. Iwamoto, D. Maruyama, T. Koda, N. Kohtake, K. Takashio, and H. Tokuda, "u-Photo: Interacting with pervasive services using digital still images," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, vol. 3468, pp. 190–207.

[9] "ARToolKit," http://www.hitl.washington.edu/artoolkit/ (Accessed: Jun 30, 2015).

[10] T. Ito, K. Hashizume, K. Kawada, N. Nakagawa, N. Namatame, M. Ito, J. Nakazawa, K. Takashio, and H. Tokuda, "Snappy: A snap-based human interaction for multiple device collaboration," in *the 7th International Conference on Pervasive Computing (Pervasive 2009), Demonstrations*, Nara, Japan, May 2009.

[11] L. Corporation, "LINE," http://line.me/ (Accessed: Jun 30, 2015).

[12] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, Dec. 2009.

[13] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," in *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI 2008)*, Bonn, Germany, 2008, pp. 11–14.

[14] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," *Personal Ubiquitous Computing*, vol. 10, no. 5, pp. 285–299, Jul. 2006.

[15] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," in *Proceedings of the 6th international conference on Ubiquitous Intelligence and Computing (UIC 2009)*, Brisbane, Australia, Jul. 2009, pp. 25–38.

[16] Google Inc., "Android software development kit (SDK)," https://developer.android.com/sdk/index.html (Accessed Jun 29, 2015).

[17] "Near field communication (NFC)," http://www.nearfieldcommunication.org/ (Accessed: Jun 30, 2015).

[18] "NFC forum technical specifications," http://members.nfc-forum.org/specs/spec_list/ (Accessed: Jun 30, 2015).

[19] "ZigBee alliance," http://www.zigbee.org/.

[20] "Microchip stack for the ZigBee protocol," http://ww1.microchip.com/downloads/en/AppNotes/00965a.pdf (Accessed: Jun 30, 2015).

[21] "iBeacon for developers," https://developer.apple.com/ibeacon/ (Accessed: Jun 30, 2015).

DOI : 04.IJAI.2015.1.1