Directional Localization and Activity Recognition with ESP32 Wi-Fi

1st Ratthamontree Burimas
School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Pathum Thani, Thailand
bank23232525@gmail.com

2nd Teerayut Horanont
School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Pathum Thani, Thailand
teerayut@siit.tu.ac.th

***Abstract*—**

***Index Terms*—trajectory data, trajectory simplification, quantum algorithm, local integral square synchronous Euclidean distance, Quantum Maximum or Minimum Searching Algorithm**

I. Introduction

Wifi is a common medium in many kinds of field nowadays. Normally, It is used as a network to connect to the internet. But there are still many more functions Wifi is good at. Wifi can also be applied in fields beside connecting to the internet according its stability being developed. Decent Wifi connectivity can extract more data other than the data to be transmitted like concentration, speed, obstacle between the transmission. Those can be composed to be many useful data on their own like localization, activity prediction and etc.

Camera is a very good tool for monitoring things and is being used as a very effective data collector for mapping to the ground truth to create many popular machine learning model like pose estimation, text segmentation, object detection and many more. However, camera are unavoidably judged as a serious privacy infringement since the data obtained like a photo or video are too clear and possess too much information that might be used in a bad way.

This paper proposes a tool for exploiting Wifi to be 3D pose estimation machine by using ESP32 as Wifi transmitter and reciever.

II. Background

A. Pose Estimation

There many machine-learning based human body pose detection tools had been proposed and available online. Those can de found both 2D and 3D. In our paper, we chose a light weight 3D pose detection as an annotation due to its simplicity and the hypothesis that 3D should suit the most in our work. The project can be found at [github-lw3d] and is based on [paper-Lightweight OpenPose] and [paper- Single-Shot Multi-Person 3D Pose Estimation From Monocular RGB]. Its job is to simply create 3D human pose annotation from an image. Then, feed to our works training process as an annoation.

B. Wifi

Wifi is a well-known connectivity with no wire needed (wireless). It has been used as a medium for connecting to the internet for over 10 years. However, the Wifi is the name covering IEEE 802.11 n/g/ac protocols. It mostly deliver data through 2.4/5GHz frequency with multiple channels. The bandwidth in each channel is 20/40/80/160MHz. the data are to be transmitted pararelly with multiplexing technique named orthogonal frequency division multiplexing (OFDM). Each carrier may propagate to a reciever with encountering many obstacles. The effect of that situation is the Doppler Effect. So, Channel State Information (CSI) is represented as physical layer indicator that can be used to investigate how each channel propagate to the reciever.

If a sender sends data to a reciever through Wifi, the data will be surely not transmitted without any loss.

C. CSI data

As mentioned in II-B that data propagating to the reciever while touching surrounding environment, the CSI is a variation of the data. The CSI can be found at both sender and reciever since reciever may transmit data back. We consider to mainly use CSI at the reciever. Let the sender use the modulation method of 16-quadrature amplitude modulation (16-QAM) which one carrier can carry 4 bits. When the sender needs to send a '1111', the modulation returns $x = 1 + 1i$ then, transmit to the reciever. At the reciever, let the obtained data is $y = 0.8+0.9i$. So, the CSI can be computed by the variation $h = y/x = 0.2 + 3.4i$. [] and [] have proven that human body can affect the CSI.

1) ESP32: ESP32 is a popular single-board computer (SBC). With its affordable price and many available additional tools, ESP32 is commonly used in Internet of Things field.

Moreover, it can be applied in research field. Quantitative CSI can be obtained from Wifi in ESP according to [Wi-ESP]. The number of available subcarriers in ESP32 is 64.

2) Faraday cage: Faraday cage is invented by Michael Faraday in 1836. It is an enclosure to block electromagnetic fields. It is made of conducting material which can affect any radio frequency (e.g. Wifi) to be unable to pass through.

III. Proposed Method

A. Preparation

As shown Fig[] , ESP32 is used as a reciever and a sender. The data is transmited from one ESP32 to another through every possible propagation way except straight way since it is blocked by strong Faraday cage. So, the CSI obtained at the reciever is considerably propagated by reflecting from outside environment only.

B. CSI Clipping

The typically obtained CSI is a measurement of the traffic in an area where the signal propagate. The area may be considerably too big since the CSI can be affected by any unstable environment that is out of our focus e.g. we are focusing on a human standing still 3 meters away from the device but, the CSI is significantly fluctuated by a dog jumping 10 meters away from the device. Those uncontrollable factors are obvious not included in the annotation. So, it causes bad result in both training and testing processes.

To solve this, we use IFFT to turn the CSI into Power Delay Profile (PDP). The PDP is time-to-signal-power value where we can distinguish how strong signal power arrive at a certain time and we know that the signal normally travel at speed of light. Then, we filter out the PDP in the range that is surely arrive from a time range that is not in our focus e.g. we are focusing in an area of 7.5 meters then, we ignore the signal that arrive at the time after $(7.5/speed - of - light) = 2.25$ nanoseconds. Finally, we use FFT to turn the clipped PDP back into the CSI and use that CSI for further processing.

Fig. 1 top-left shows the originally obtained CSI. Then, it was turn into the PDP by IFFT (Fig. 1 top-right). And, let time domain more than 10 is considered out of focus. So, the PDP after time 10 are forced to be 0 in Fig. 1 bottom-left. The clipped CSI where affected only from objects within the distance we focus is resulted back by FFT from the clipped PDP in Fig. 1 bottom-right.

C. CSI Splicing

As mentioned in the above section, we can simply cliping the CSI to maintain only the part within the range we focus by using the PDP. But, the normal resolution of PDP is usually not that high. In short, the PDP resolution $\Delta\tau$ is depended on the CSI bandwidth. In 802.11n, the bandwidth is 22 MHz for one Wi-Fi channel. And, $\Delta\tau = 1/B$, where B is the bandwidth. So, the PDP resolution $\Delta\tau$ in a normal CSI data is $1/(22 * 10^6) = 45.5$ nanoseconds (in Fig. 2 top) which can separate the distance of $(45.5 * speed - of - light) = 13.6$ meters. The 13-meter-gap is significantly not practical in our work.

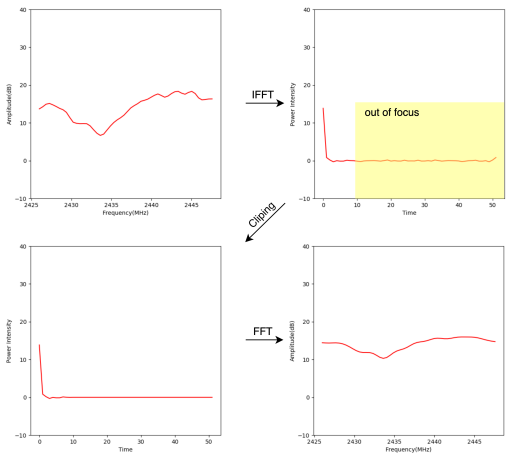

Fig. 1. CSI clipping.

To enhance the PDP resolution, a straightforward idea is to expand the CSI bandwidth. There is a method called CSI Splicing proposed by []. This leads to the collections of CSI from many channels and merge with some error filtering to create a single CSI with the bandwidth of $\leq$ 160MHz.

The fully spliced CSI with max bandwidth can result the PDP with resolution $\Delta\tau = 1/(160 * 10^6) = 6.25$ nanoseconds (in Fig. 2 bottom) which allow us to separate the distance of $(6.25 * speed - of - light) = 1.87$ meters.

The example of merging CSI from Wi-Fi channel 1, 4, 6, 8 and 11 into single CSI with bandwidth 73 MHz is show in Fig. 3.

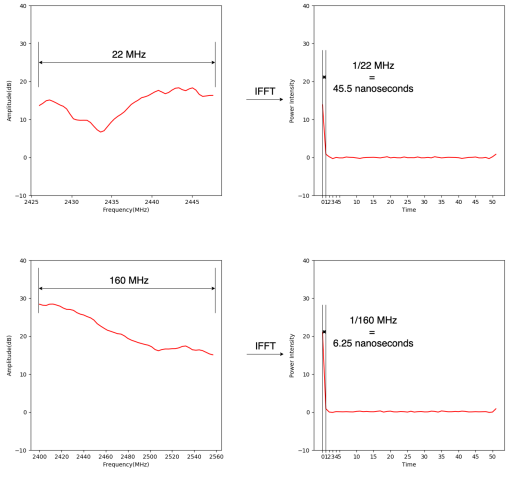

Fig. 2. PDP resolution depending on CSI bandwidth.

D. Pre-processing

As mentioned in II-C1, There are 64 subcarriers available in CSI data from ESP32. So, we can construst tensor of $n \times 64$,

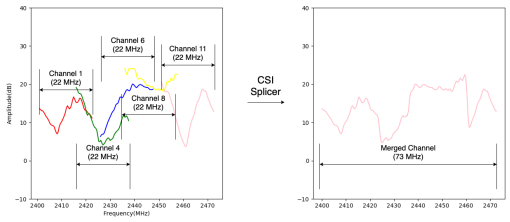

Fig. 3. CSI splicing.

where n is a number of recieved package. We are to map 3D human pose annotation from a camera to CSI from the ESP32. The sampling rate of the camera and ESP32 Wifi are set as 100Hz and 20Hz, respectively. So, we can map 5 CSI samples to a 3D human pose annotation from 1 image with the corresponding timestamp.

The 3D human pose annotation is as shown in []. There are 19 keypoints. So, we can construct a matrix of 19×3 from a single annotation.

Conclusively, we are making a model by mapping a group of matrices of 5×64 from CSI to each matrix of 19×3 from human pose annotation with the corresponding timestamp.

E. Form A Pose Adjacent Matrix

As mentioned in [], there are 19 keypoints for forming a single 3D human pose. So, the matrix of 19 seems enough for using as annotation. But, we know that too much independency of each keypoint may lead to some impossible pose alignment e.g. body keypoint found very far from neck keypoint or right shoulder keypoint attached to left leg knee keypoint. We assume some is quite not possible for normal human pose. To preserve those constraint, we form pose adjacent matrix (PAM) from a direct 19×3 matrix. the PAM is applied for all x, y and z axes. Each are to be form their 19×19 by the following equations.

$$x'_{i,j} = \begin{cases} x_i - x_j & i \neq j \\ x_i & i = j \end{cases} \tag{1}$$

,

$$y'_{i,j} = \begin{cases} y_i - y_j & i \neq j \\ y_i & i = j \end{cases} \tag{2}$$

and

$$z'_{i,j} = \begin{cases} z_i - x_j & i \neq j \\ z_i & i = j \end{cases} \tag{3}$$

.

The PAM is finally a $3 \times 19 \times 19$ matrix of 3 matrices of x',y' and z' combined. And, can be used as an annotation that preserves human pose constraint.

F. Form Network Layer

Let D be a set of pairs of synchronized image and 5 CSI data package. Each pair has corresponding timestamp.

$$D = (I_t, C_t), t \in [1, n] \tag{4}$$

, where n is a number of pairs, I_t is an image frame from a camera, C_t is series of 5 CSI data from ESP32 and t is the timestamp when those 2 were collected.

We use ((Teacher and Student as [] and [])). Let the lightweight 3D human pose [] be a teacher network and our works (ESP32 pose sense) be a student network. The teacher and student network are termed as $T()$ and $S()$, respectively.

Iteratively, $T()$ takes I_t then, returns a single 3D human pose as a matrix of 19×3 . And, we convert the obtained matrix to PAM with Eq. 1, Eq. 2 and Eq. 3. In short, $T()$ is formulized as $T(I_t) \rightarrow PAM_t$. Recall that main purpose of $T(I_t)$ is to teach $S()$.

Simultaneously, $S()$ takes C_t then, returns the prediction of PAM matrix. The $S()$ is to be supervised by $T()$ until $S()$ can predict PAM close to the one at the $T()$. The process of transfroming from C_t to PAM_t is shown in Fiq. **??**. There are 3 modules to achieve the transformation those are the encoder, feature extractor and the decoder.

a) Encoder: The encoder takes CSI series (5×64 matrix) as an input and outputs a tensor ($320 \times 148 \times 148$ matrix). The main purpose of the process is to upsample C_t since ResNet[] is the convolutional network that is used in our work and it requires higher dimension of input. A single C_t is 5×64 matrix. We need to map it to a single RGB image which is $3 \times 244 \times 244$ matrix, where 3 is for 3 channel of colors and 224s are the dimension of the image. [CSI-NET] uses 8 stacked transposed convolutional layers to upsample and [WiSPPN] uses one bilinear interpolation operation. In our work, we decide to use ..... to convert C_t to $320 \times 148 \times 148$ matrix.

b) Feature Extractor: The feature extractor is the main part of training how to extract 3D human pose feature from the upsampled C_t. CSI data does not obviously contain any spatial information. So, we need to use deep network to enhance the feature extractor so that it can extract spatial infomarion from the CSI. Resnets [] is responsible for this job..... The feature extractor consists of 4 stacked blocks of ResNet [] as shown in Fig. **??**. The parameter of each layer is shown in Table **??**. At the end, we obtained $300 \times 19 \times 19$ matrix as a feature output F_t.

1st block : Take a tensor ($320 \times 148 \times 148$ matrix) as an input and outputs a tensor ($320 \times 148 \times 148$ matrix)

2nd block : Take a tensor ($320 \times 148 \times 148$ matrix) as an input and outputs a tensor ($150 \times 74 \times 74$ matrix)

3rd block : Take a tensor ($150 \times 74 \times 74$ matrix) as an input and outputs a tensor ($300 \times 37 \times 37$ matrix)

4th block : Take a tensor ($300 \times 37 \times 37$ matrix) as an input and outputs a tensor ($300 \times 19 \times 19$ matrix)

c) Decoder: the decoder takes a F_t ($300 \times 19 \times 19$ matrix) as an input and outputs the prediction of PAM ($3 \times 19 \times 19$ matrix). Its main job is to return the prediction of 3D human pose in a form of PAM where can be simply turned into 3D human pose. To do so, we stack two convolutional layers as shown in Fig. **??**.((Conv1 is mainly to release channel-wise information (from 300 to 36); and Conv2 is mainly to

further reorganize spatial information of Ft with the 1 × 1 convolutional kernels.))

To be concluded, these 4 modules are the step to turn CSI data series into predicted 3D human pose which is as $S(C_t) \rightarrow pPAM_t$. As this is teacher and student network, we believe that $pPAM_t$ is initially not correct and needed to be supervised by the $T(I_t) \rightarrow PAM_t$. Eventually, we obtain $S()$ that can predict 3D human pose effectively. The loss computation and the training of $S()$ are discussed further.

G. Pose Adjacent Matrix Similarity Loss

The $T()$ is to supervise the $S()$ with the PAM from images and Lightweight 3D human pose esitmation. L2 loss is applied for optimizing the $S()$.

$$\begin{aligned} L = &|| \, pPAM^x \text{ - } PAM^x \, ||_2^2 + \\ &|| \, pPAM^y \text{ - } PAM^y \, ||_2^2 + \\ &|| \, pPAM^z \text{ - } PAM^z \, ||_2^2 \end{aligned} \tag{5}$$

, where $||||_2^2$ is an operator to compute L2 distance. pPAM^x, pPAM^y and pPAM^z, from the $S()$, are the predictions of PAM of 3D human pose on x-axis, y-axis and z-axis respectively. And, PAM^x, PAM^y and PAM^z, from the $T()$, are the annotations of PAM of 3D human pose on x-axis, y-axis and z-axis respectively.

H. Training Details and Pose Association

The environment of the implementation is python 3.7.9, scipy 1.6.0 and pytorch 1.7.1. The network is trained for ... epochs with initial learning rate of ...., batch size of ... and Adam optimizer. The learning rate decays by ... at the epoch of 5th, 10th and 15th.

When the $S()$ is smart enough, we let it take 5 CSI data series as input and output pPAM on its own. Obviously, we can look at only diagonal values of the pPAM and plot 3D human pose on 3D canvas.

$$\begin{aligned} x_k^*, y_k^*, z_k^* = &\text{ for } (k = 1 \rightarrow 19 \text{ }): \\ &(pPAM_{1,k,k}, pPAM_{2,k,k}, pPAM_{3,k,k}) \end{aligned} \tag{6}$$

IV. Evaluation

A. Data Collection

B. Experimental Result

(((Percentage of Correct Keypoint (PCK) is widely used to evaluate the performance of proposed approach [15, 31, 32].

where I(·) is a binary indicator that outputs 1 while true and 0 while false. are the same as Equation. N is the number test frames. i denotes the index of body joint and $i \in 1, 2, ..., 18$. The rh and lh are for the positions of the right shoulder and the left hip, respectively. Thus the $2rh2 + lh2$

can be regarded as the length of the upper limb, which is used to normalize the prediction error, $|| \, pd_i\text{-}gt_i \, ||_2^2$

, where pd is prediction coordinates and gt is the ground-truth. [Table. I] shows the estimation performance of 18 body keypoint in PCK@5, PCK@10, PCK@20, PCK@30, PCK@40, and PCK@50. From the table, we can see WiSPPN do pose estimation well. Figure. 9 illustrates some estimation comparisons between AlphaPose and WiSPPN. The results show that WiSPPN can work single pose estimation with comparable results to cameras.)))

Github[1].

a) Fixed-# Evaluation:

TABLE I
FIXED-# EVALUATION $c = 5$

Trajectory Data	#iteration	#satisfying condition	#the finest solution
Geolife000	100	100	100
Geolife001	100	100	0
Geolife002	100	100	100
Geolife003	100	100	100

TABLE II
FIXED-# EVALUATION $c = 10$

Trajectory Data	#iteration	#satisfying condition	#the finest solution
Geolife000	100	100	100
Geolife001	100	100	100
Geolife002	100	100	100
Geolife003	100	100	100

Conclusion and Future Development

This paper proposes 2 algorithms for 2 simplification types, Fixed-# and Fixed-LSSED. Classically, to achievement the best satisfying result for both types, the time complexity of $O(2^{N-2})$ is needed, where N is a number of trajectory point. By exploiting quantum mechanics from the QSMA, we can do the same job by consuming only the time complexity of $O(\frac{\pi}{2}\sqrt{\frac{2^{N-2}}{M}} \times c)$, where M is a number of the finest solution and c is an adjustable parameter.

Acknowledgment

We thank Sirindhorn International Institute of Technology for providing technical environment and supportive information.

[1]https://github.com/rtmtree/

TABLE III
FIXED-LSSED EVALUATION $c = 5$

Trajectory Data	#iteration	#satisfying condition	#the finest solution
Geolife000	100	100	0
Geolife001	100	100	100
Geolife002	100	100	100
Geolife003	100	100	0

TABLE IV
FIXED-LSSED EVALUATION $c = 10$

Trajectory Data	#iteration	#satisfying condition	#the finest solution
Geolife000	100	100	0
Geolife001	100	100	100
Geolife002	100	100	100
Geolife003	100	100	100

REFERENCES

[1] Y. Chen, S. Wei, X. Gao, C. Wang, J. Wu and H. Guo, "An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits," arXiv:1908.07943 [quant-ph] 21 Aug 2019.

[2] G. L. Long, "Grover algorithm with zero theoretical failure rate," arXiv:quant-ph/0106071, 13 Jun 2001.

[3] A. Ahuja and S. Kapoor, "A Quantum Algorithm for finding the Maximum," arXiv:quant-ph/9911082, 18 Nov 1999.

[4] C. Dürr and P. Høyer, "A quantum algorithm for finding the minimum," arXiv:quant-ph/9911082, 18 Nov 1999.

[5] M. Chen, M. Xu and P. Fränti, "A Fast Multiresolution Polygonal Approximation Algorithm for GPS Trajectory Simplification," IEEE Transactions on image processing, col. 21, No. 5, May 2012.

[6] D. Zhang, M. Ding, D. Yang, Y. Liu, J. Fan, H. T. Shen, "Trajectory Simplification: An Experimental Study and Quality Analysis," Proceedings of the VLDB Endowment, vol. 11, No. 9, Rio de Janeiro, Brazil, August 2018.

[7] N. Meratnia and R. A. de By, "Spatiotemporal compression techniques for moving point objects," In EDBT, pages 765–782, 2004.