This repository contains the datasets of Channel State Information (CSI) and Body-coordinate Velocity Profile (BVP) for human gesture recognition using WiFi.

Cite the Paper

> Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. **"Zero-Effort Cross-Domain Gesture Recognition With Wi-Fi"**. In *Proceedings of MobiSys '19*. Association for Computing Machinery, New York, NY, USA, 313–325.

Readers may also check the homepage of project Widar3.0 for more details about WiFi sensing.

Dataset Description

Data Type	File Name Format	Description
CSI	id-a-b-c-d-Rx.dat	'**id**': user's id; '**a**': gesture type, '**b**': torso location, '**c**': face orientation, '**d**': repetition number, '**Rx**': Wi-Fi receiver id.
BVP	id-a-b-c-d-suffix.mat	'**id**': user's id; '**a**': gesture type, '**b**': torso location, '**c**': face orientation, '**d**': repetition number. Each file is a **20*20*T** matrix, where the first dimension represents the velocity along x axis ranging between [-2,+2] m/s, the second dimension represents the velocity along y axis ranging between [-2,+2] m/s and the third dimension represents the timestamps with 10Hz sampling rate.

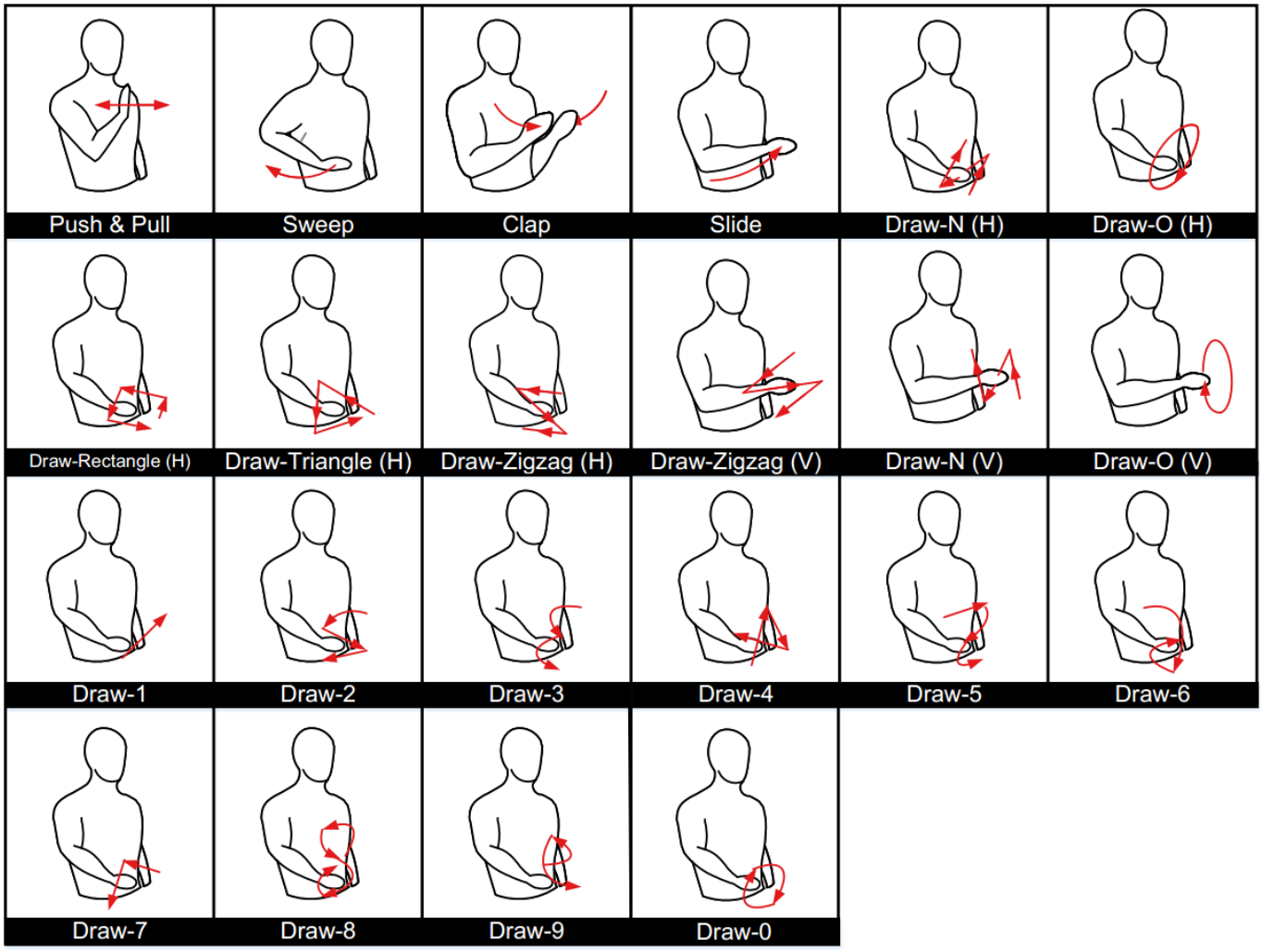
Note: gesture type refers to Gesture Description part. Rx location, torso location and face orientation refer to Device Deployment part.

Data Type	Files	Room#	Gestures	Users
CSI	CSI/20181109.zip	1	1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; 5: Draw-Zigzag(Vertical); 6: Draw-N(Vertical);	User1,2,3
CSI	CSI/20181112.zip	1	1: Draw-1; 2: Draw-2; 3: Draw-3; 4: Draw-4; 5: Draw-5; 6: Draw-6; 7: Draw-7; 8: Draw-8; 9: Draw-9; 0: Draw-0;	User1,2
CSI	CSI/20181115.zip	1	1: Push&Pull; 2: Sweep; 3: Clap; 4: Draw-O(Vertical); 5: Draw-Zigzag(Vertical); 6: Draw-N(Vertical);	User1
CSI	CSI/20181116.zip	1	1: Draw-1; 2: Draw-2; 3: Draw-3; 4: Draw-4; 5: Draw-5; 6: Draw-6; 7: Draw-7; 8: Draw-8; 9: Draw-9; 0: Draw-0;	User1
CSI	CSI/20181117.zip	2	1: Push&Pull; 2: Sweep; 3: Clap; 4: Draw-O(Vertical); 5: Draw-Zigzag(Vertical); 6: Draw-N(Vertical);	User4
CSI	CSI/20181118.zip	2	1: Push&Pull; 2: Sweep; 3: Clap; 4: Draw-O(Vertical); 5: Draw-Zigzag(Vertical); 6: Draw-N(Vertical);	User2,3

Data Type	Files	Room#	Gestures	Users
CSI	CSI/20181121.zip	1	1: Slide; 2: Draw-O(Horizontal); 3: Draw-Zigzag(Horizontal); 4: Draw-N(Horizontal); 5: Draw-Triangle(Horizontal); 6: Draw-Rectangle(Horizontal);	User1,2,3
CSI	CSI/20181127.zip	2	1: Slide; 2: Draw-O(Horizontal); 3: Draw-Zigzag(Horizontal); 4: Draw-N(Horizontal); 5: Draw-Triangle(Horizontal); 6: Draw-Rectangle(Horizontal);	User2,5
CSI	CSI/20181128.zip	2	1: Push&Pull; 2: Sweep; 3: Clap; 4: Draw-O(Horizontal); 5: Draw-Zigzag(Horizontal); 6: Draw-N(Horizontal);	User6
CSI	CSI/20181130.zip	1	1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; 5: Draw-O(Horizontal); 6: Draw-Zigzag(Horizontal); 7: Draw-N(Horizontal); 8: Draw-Triangle(Horizontal); 9: Draw-Rectangle(Horizontal);	User5,10~17
CSI	CSI/20181204.zip	2	1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; 5: Draw-O(Horizontal); 6: Draw-Zigzag(Horizontal); 7: Draw-N(Horizontal); 8: Draw-Triangle(Horizontal); 9: Draw-Rectangle(Horizontal);	User1
CSI	CSI/20181205.zip	2	[**User2**]1: Draw-O(Horizontal); 2: Draw-Zigzag(Horizontal); 3: Draw-N(Horizontal); 4: Draw-Triangle(Horizontal); 5: Draw-Rectangle(Horizontal); [**User3**]1: Slide; 2: Draw-O(Horizontal); 3: Draw-Zigzag(Horizontal); 4: Draw-N(Horizontal); 5: Draw-Triangle(Horizontal); 6: Draw-Rectangle(Horizontal);	User2,3
CSI	CSI/20181208.zip	2	[**User2**]:1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; [**User3**]:1: Push&Pull;2: Sweep; 3: Clap;	User2,3
CSI	CSI/20181209.zip	2	[**User2**]:1: Push&Pull; [**User6**]:1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; 5: Draw-O(Horizontal); 6: Draw-Zigzag(Horizontal);	User2,6
CSI	CSI/20181211.zip	3	1: Push&Pull; 2: Sweep; 3: Clap; 4: Slide; 5: Draw-O(Horizontal); 6: Draw-Zigzag(Horizontal);	User3,7,8,9
BVP	BVP.zip	All	The gestures correspond to those in CSI data.	All

Note: The prefix of zip file represents the data collection date.

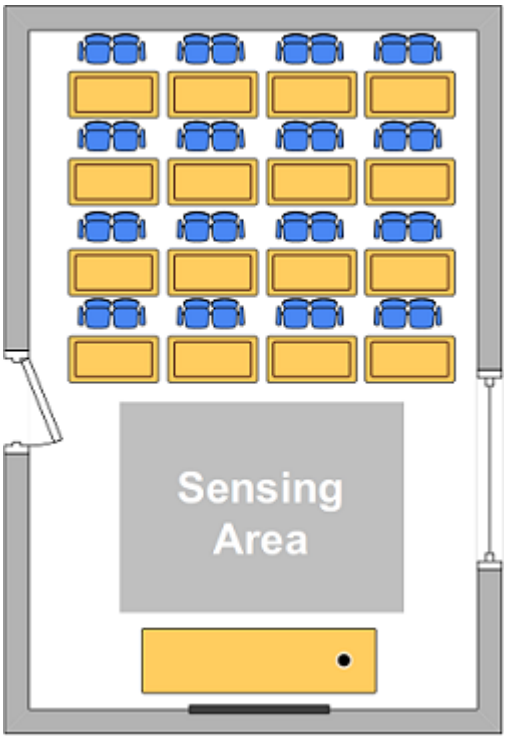
Gesture Description



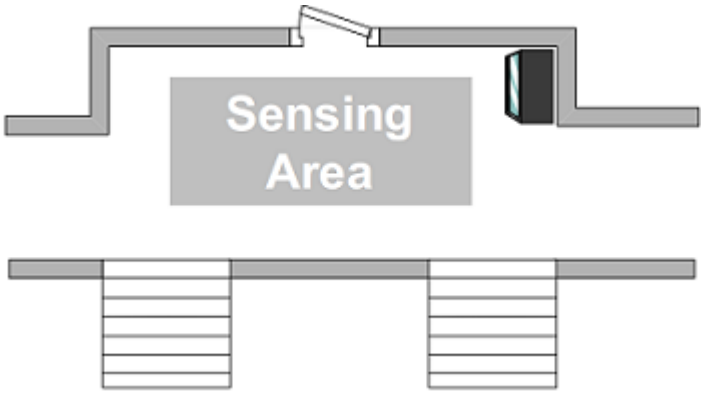
Note: ***(H)*** represents Horizontal and ***(V)*** represents Vertical.

Floor Plan

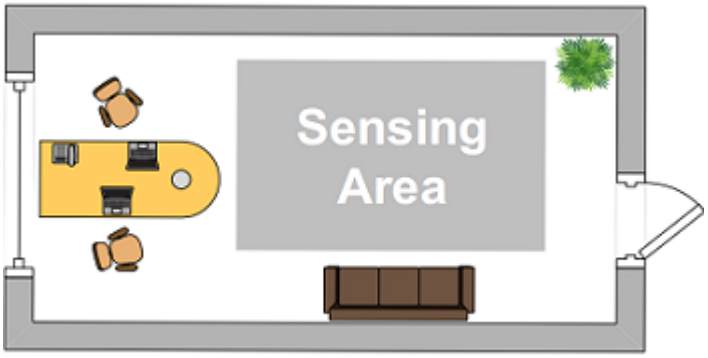
Room#1 - Classroom



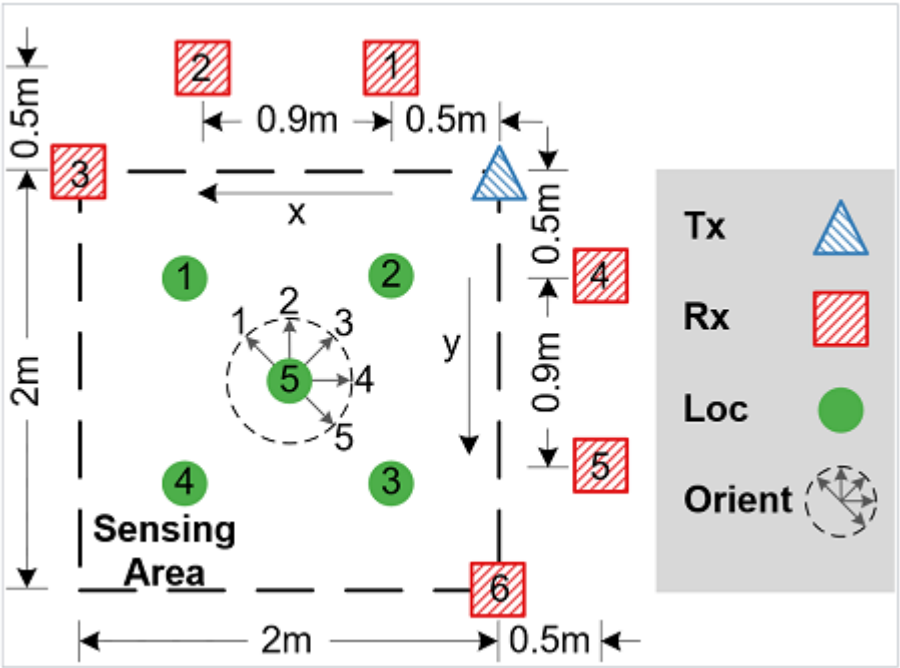
Room#2 - Hall



Room#3 - Office



Device Deployment



Torso Location#	Location (m)	Face Orientation#	Orientation (degree)
1	(1.365, 0.455)	1	-90
2	(0.455, 0.455)	2	-45
3	(0.455, 1.365)	3	0

Torso Location#	Location (m)	Face Orientation#	Orientation (degree)
4	(1.365, 1.365)	4	45
5	(0.91, 0.91)	5	90
6	(2.275, 1.365)		
7	(2.275 2.275)		
8	(1.365 2.275)		

Note: Torso Location #6-8 are not shown in the figure. The reference orientation is denoted as facing the Tx (Orient#3). Tx location is (0, 0).

Code Example

The following shows demo code for BVP extraction and DFS extraction from raw CSI data. Detailed BVP extraction algorithm can be found in our paper.

1. **Extract BVP from CSI**

```
clear;
clc;
close all;

start_index = [1 1 1 1];
total_mo = 1;   % Total motion count
total_pos = 1;  % Total position count
total_ori = 1;  % Total orientation count
total_ges = 1;  % Total gesture repeatation count
start_index_met = 0;
rx_cnt = 6;     % Receiver count(no less than 3)
rx_acnt = 3;    % Antenna count for each receiver
dpth_pwd = './';
dpth_date = 'Data';
dpth_people = 'userA';
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dpth_ges = [dpth_pwd, dpth_date, '/'];
dpth_vs = [dpth_pwd, 'BVP/'];
for mo_sel = 1:total_mo
    for pos_sel = 1:total_pos
        for ori_sel = 1:total_ori
            for ges_sel = 1:total_ges
                spfx_ges = [dpth_people, '-', num2str(mo_sel), '-',
num2str(pos_sel),...
                    '-', num2str(ori_sel), '-', num2str(ges_sel)];
                if mo_sel == start_index(1) && pos_sel == start_index(2) &&...
                        ori_sel == start_index(3) && ges_sel == start_index(4)
                    start_index_met = 1;
                end
                if start_index_met == 1
                    disp(['Running ', spfx_ges])
```

```
                    try
                        DVM_main;
                    catch err
                        disp(['Exception Occured' err.message]);
                        fprintf(exception_fid, '%s\n', spfx_ges);
                        fprintf(exception_fid, '%s\n', err.message);
                        continue;
                    end
                else
                    disp(['Skipping ', spfx_ges])
                end
            end
        end
    end
end
```

The `mo_sel pos_sel ori_sel ges_sel` will traverse all of the CSI file and call `DVM_main` to extract BVP, which is the core algorithm of our paper.

The extracted BVP data will be stored in /BVP directory. Make sure you load all of the code component provided in *BVPExtractionCode.zip* in Matlab working space.

2. **Extract DFS from CSI**

```
clear;
clc;
close all;

start_index = [1 1 1 1];
total_mo = 1;   % Total motion count
total_pos = 1;  % Total position count
total_ori = 1;  % Total orientation count
total_ges = 1;  % Total gesture repeatation count
start_index_met = 0;
rx_cnt = 6;     % Receiver count(no less than 3)
rx_acnt = 3;    % Antenna count for each receiver
dpth_pwd = './';
dpth_date = 'Data';
dpth_people = 'userA';
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dpth_ges = [dpth_pwd, dpth_date, '/'];
dpth_vs = [dpth_pwd, 'BVP/'];

for mo_sel = 1:total_mo
    for pos_sel = 1:total_pos
        for ori_sel = 1:total_ori
            for ges_sel = 1:total_ges
                spfx_ges = [dpth_people, '-', num2str(mo_sel), '-',
num2str(pos_sel),...
                    '-', num2str(ori_sel), '-', num2str(ges_sel)];
                if mo_sel == start_index(1) && pos_sel == start_index(2) &&...
```

```
                        ori_sel == start_index(3) && ges_sel == start_index(4)
                    start_index_met = 1;
                end
                if start_index_met == 1
                    disp(['Running ', spfx_ges])
                    try
                        [doppler_spectrum, freq_bin] = get_doppler_spectrum(...
                            spfx_ges, rx_cnt, rx_acnt, 'stft');
                    catch err
                        disp(['Exception Occured' err.message]);
                        continue;
                    end
                else
                    disp(['Skipping ', spfx_ges])
                end
            end
        end
    end
end
```

The `mo_sel pos_sel ori_sel ges_sel` will traverse all of the CSI file and call `get_doppler_spectrum` to extract DFS. Both *CWT* and *STFT* algorithms are supported. CSI denoising algorithms can be found in our previous works widar2.0 and widance.

The extracted DFS profile is stored in variable `doppler_spectrum` in Matlab working space. You can use it for your custom applications and algorithms. Make sure you load all of the code component provided in *DFSExtractionCode.zip* in Matlab working space.

Bug Notice

List of empty files

- 20181109/user2/user2-6-4-4-2-r1.dat
- 20181109/user3/user3-1-3-1-8-r5.dat
- 20181118/user2/user2-3-5-3-4-r4.dat
- 20181209/user6/user6-3-1-1-5-r5.dat
- 20181211/user8/user8-1-1-1-1-r5.dat
- 20181211/user8/user8-3-3-3-5-r2.dat
- 20181211/user9/user9-1-1-1-1-r1.dat

Please skip these files when loading them into your applications.