

Indoor Floor Map Construction with Video Survey

Tsz-lung Wong, Albert Kai-sun Wong, Ka-sing Lin, and Chin-Tau Lea

Emails: tlwongaa@ust.hk, eeablert@ust.hk, ksclin@ust.hk, eelea@ece.ust.hk

ECE Department, The Hong Kong University of Science and Technology, Hong Kong

Abstract—This paper describes a system which uses simple video surveys to automatically construct an indoor floor map for the purpose of supporting indoor people localization, tracking, and navigation applications. We show that this video-based system, using the concept of spatial segmentation through similarity matching followed by graph construction as developed in the WiFi RSS-based Intelligent Mobility Mapping System (IMMS) presented in an earlier work, is capable of constructing an indoor floor map based on simple video survey sequences alone. We show also that when used together with WiFi RSS, the video-based system can easily identify and label points-of-interest such as rooms along corridors, enriching the information content in the indoor map.

Keywords: intelligent mobility mapping system, simultaneous localization and mapping, indoor localization and tracking, SIFT, optical characters recognition

1. Introduction

Indoor localization based on WiFi RSS (Received Signal Strength) fingerprinting has been a subject of research for over a decade. The most important advantage of using WiFi RSS is convenience. WiFi coverage is found in most indoor environments today and WiFi RSS can be measured without requiring permission. However, the off-line survey process to create a fingerprint, or a radio map, is tedious and time-consuming. Recently, an Intelligent Mobility Mapping System (IMMS), proposed in [1], applies the concepts of crowd-sourcing and Simultaneous Localization and Mapping (SLAM) to simplify the off-line survey step. The concept in IMMS is to make use of crowd-sourced WiFi RSS sequences or traces to identify location segments within the indoor coverage area by identifying highly similar segments across many traces. Afterwards, the identified location segments, which presumably are corridor segments in a grid-like indoor environment, can be used to construct a graphical indoor map through a graph construction algorithm. However, IMMS will create some false location segments and vertices. In addition, the orientation of the individual location segments are unknown, making matching of the graphical map to the physical map difficult.

Other researchers have conducted research into indoor localization using video devices. Compared to WiFi RSS, video is more accurate for identifying the exact number and positions of intersection points of corridor segments.

On the other hand, video is processing and bandwidth intensive and possibly less convenient for the online localization/navigation phase. Our work intends to combine WiFi RSS with video for indoor floor map and radio map construction and localization.

In this paper, we describe methods to extract useful information from video sequences labeled with WiFi RSS. Our work first focuses on indoor physical spaces that are grid-like and can be modeled as an interconnection of corridor segments called Atomic Location Segments (ALSs) [1]. The information of interest is the number and location of the intersection points of corridors, and rooms and their room numbers alongside corridors. Accurate identification of corridor intersection points, which are combined with WiFi RSS data, will enable accurate automatic reconstruction of the indoor map, and labeling of rooms along corridors will enrich the information content of the indoor map.

This paper is organized as follows. Section 2 gives an overview of the system. Section 3 describes the algorithm to find the intersection points. Section 4 describes the unique ALS identification. Section 5 describes the floor map construction. Section 6 describes the algorithm to find and recognize room numbers. Experimental results are shown in section 7.

2. System Overview

Figure 1 shows a simple floor map which illustrates the concept of Atomic Location Segment (ALS), which are corridor segments, and Breaking Point (BP), which are intersections of corridors. A BP is where people can make turn into different corridor segments. We call an intersection a BP because it is where the similarity between two WiFi RSS or video traces may end if the two WiFi RSS or video signal collectors turn into different corridor segments. An ALS is the corridor segment between two BPs.

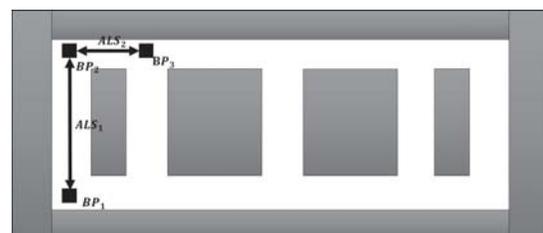


Fig. 1: An indoor floor map model

Our indoor map construction involves three steps:

1. Data collection: As described in the introduction, our overall objective is to combine video and WiFi RSS to perform indoor floor and radio map constructions to support WiFi fingerprint based indoor localization. The primary focus of this paper is on using video data to enhance the map construction. To enhance the detection of corridor intersections and points of interest, we will record two video clips. The first clip is taken with the surveyor walking causally around the coverage area without pausing, making turn (randomly left or right) whenever an intersection is encountered, until most of the corridor segments in the coverage area is traversed more than once. In taking the second clip, the surveyor pauses at all or selected rooms along the corridors to direct the camera at the doorplate of each for several seconds. Both clips are taken while WiFi RSS measurements are collected at an interval of one second. The timestamp for each frame in the video clips and for each measurement in the RSS trace are synchronized.

2. Information extraction: In this stage, for clip 1, each encounter of an intersection and the surveyor's turning direction is identified and the occurrence time is marked. For clip 2, each pausing and the room number captured is identified. The methods of Scale Invariant Features Transform (SIFT) and Optical Characters Recognition (OCR) are applied.

3. Map construction: In this stage, video clip 1 is divided into segments based on the BPs identified. Video segments that are taken at the same corridor segment are identified so that a list of unique corridor segments called ALSs can be created. The list of unique ALSs, their connectivity relationships through BPs, and turning directions are used to construct the floor map. The room numbers identified in clip 2 are used to mark points of interest in the floor map.

3. Intersections Finding Algorithm (IFA)

For our indoor map construction, accurate identification of the set of breaking points is very important. In the WiFi RSS-based approach in [1], some superfluous BPs are often identified, resulting in false location segments. Another problem with the WiFi RSS approach in [1] is that the right-handedness or left-handedness of intersections of location segments is not determinable, and hence the graphical planar floor map constructed is not unique. Thus, unless the graphical floor map is manually aligned with the actual floor map, the value of the graphical floor map is limited in navigation applications as it cannot be used to guide a user to make right turn or left turn. The objective of IFA here is to use video clips to determine the set of BPs accurately, and to record the direction of each turn.

3.1 Scene matching

In the video trace recorded as the surveyor walks through the coverage area, scenes in successive frames are expected to change slowly unless the surveyor is traversing a BP.

This means that we can find the BPs by monitoring changes in successive video frames. We have considered different ways for measuring changes in video frames. We first tried a relatively simple method which is to compute the correlation between frames (e.g., [3], [4]). However, this does not work well since the correlation can be easily affected by small changes in the camera position or orientation. A more robust method is to match two frames based on finding some interest points in the frames and creating a descriptor for these interest points. We can then match two frames by matching their descriptors. There are many research studies and many local descriptors have been proposed, including differential invariants [7], steerable filters [6], SIFT [2], PCA-SIFT [5] and SURF [8]. All these build a descriptor to model an image patch around some interest points. Over time, SIFT, PCA-SIFT and SURF are generally accepted to be superior. [9] gives a comparison of SIFT, PCA-SIFT and SURF. [10] also compares the performance between SIFT and SURF. In terms of speed, SURF gives the fastest runtime while the SIFT has the best performance in invariants, especially in scale, rotation and blurring. In our study, the run time is not a critical concern as BPs finding is performed offline. Therefore, we choose the SIFT descriptor for matching our video frames.

3.2 Review of the SIFT Algorithm

SIFT is a method to extract invariant features from images [2]. There are four major stages in SIFT:

1. Scale-space extrema detection: A Difference-of-Gaussian (DOG) function is used to find all potential interest points by searching over all scales and image locations.

2. Keypoint localization: A model is applied to all potential interest points to select keypoints based on their stability.

3. Orientation assignment: Based on the local image gradient directions, one or more directions are assigned to each keypoint.

4. Keypoint descriptor: A model is applied to each keypoint and its neighbours to transform them into a matrix form to represent the keypoint.

3.3 Turning Point Position and Turning Direction

Since the camera records at 30 frames per second, matching each frame with its next frames will lead to very high computational cost. Furthermore, the changes over too short a time interval may not be large enough to be detectable. Therefore, we choose only one in every $t = 10$ frames and compare it with the $d_1 = 40$ frames later. Figure 2(a) shows the number of matched interest points between the f_i and f_{i+d_1} as a function of time in the video. By applying a threshold T and a smoothing function, we obtain the binary graph shown in Figure 2(b), in which each zero region represents the a BP region during which

the surveyor is making a turn.

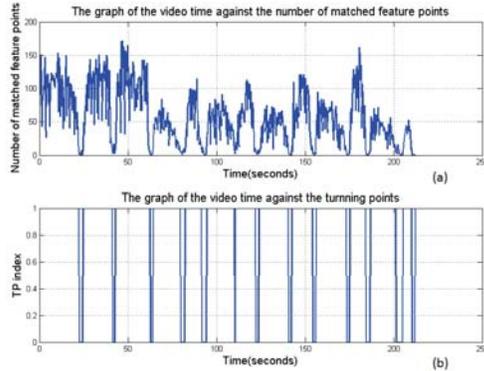


Fig. 2: BPs Finding. (a) Number of matched interest points as function of time. (b) Binarized Graph.

After finding a BP, the next step is to determine the turning direction of the surveyor at this BP. We assume that there can be only two possible turning directions (left L , right R), and compute the average shifting of the matched interest points inside each BP region in Figure 2(b). For a BP region covering N frames, for every $t = 10$ frame we sample a sample frame f_i and compute the average shifting of the frame $f_i + d_2$ where $d_2 = 5$. d_2 is smaller than d_1 so that shifting of shifting in the frames can be tracked. Assuming there are n matched interest points between frame f_i and f_{i+d_2} , the shift of frame f_{i+d_2} is computed as:

$$s_{i+d_2} = \frac{1}{n} \sum_{j=1}^n (g_{i+d_2}(x_j, y_j) - g_i(x_j, y_j)), \quad (1)$$

where s_{i+d_2} is for shift for frame $i + d_2$ and $g_i(x_j, y_j)$ is the coordinate of the j -th interest point in the i -th sample frame.

The shifting in a particular turning point region can be computed as:

$$S = \frac{1}{N/t} \sum_{i=1}^{N/t-1} s_{i+d_2}, \quad (2)$$

A negative S means the surveyor is turning to left while a positive S means the surveyor is turning to right.

4. Unique ALS and BP Identification

As described in the system overview, the segment between two intersection points is called an ALS. During the survey process, the surveyor go around the survey area randomly, and makes turn whenever an intersection is encountered, until most of the ALSs are passed through by the surveyor in the same direction more than one time. Identification of repeated ALSs will enable us to construct the floor map.

Assume K BPs are found by IFA, we divide the video trace into $K + 1$ individual video segments and label them as $(l_i; i = 1, \dots, K + 1)$. Then, we determine if two video segments l_i and l_j are taken at the same ALS by computing the average number of matched interest points (A_m) between these two segments, as described in **Algorithm 1**. In the algorithm, a unified number of N_s frames are selected at a regular interval from each of the two segments for comparison. N_s is chosen as $\min(C, N_{short}/30)$ where $C = 10$ is a constant and N_{short} is the number of frames in the shorter of l_i and l_j . This means at most 10 frames totally or one frame per second are used for the comparison. The value of A_m between all pairs of video segments are computed. We assume that the surveyor is traversing the same corridor segment in the same direction if A_m for the two corresponding video segments is larger than a threshold.

Algorithm 1 Function:TestVideoSeg

```

1: function TESTVIDEOSEG( $l_i, l_j$ )
2:    $C = 10$ 
3:    $N_s = \min(C, N_{short}/30)$ 
4:    $A_m = \frac{\sum_{k=1}^C match(l_{i_k}, l_{j_k})}{C}$   $\triangleright$  where  $l_{i_k}$  and  $l_{j_k}$  are the k-th
      selected frames in trace  $l_i$  and  $l_j$ , and  $match()$  is a function in SIFT
5:   if  $A_m > Threshold$  then
6:      $Tru = 1$   $\triangleright Tru$  is the Boolean type variable.
7:   else
8:      $Tru = 0$ 
9:   end if
10:  return  $Tru$ 
11: end function

```

The video-based approach allows us to recognize BPs without ambiguity and to identify with very high accuracy video segments taken while surveyor is travelling in the same corridor in the same direction. However, if the surveyor traverses the same corridor segment in opposite directions, the images observed by the camera would be quite different and the two video segments will not be recognized as a match. To deal with this reverse path problem, we make use of the sequence of WiFi RSS values recorded and reuse the function TestHCP described in [1]. This function examines the correlation of two WiFi RSS trace segments and returns as output whether the two WiFi RSS trace segments are in the same ALS by detecting if there is a "high correlation pattern" in the correlation matrix that extends in either the +45 or -45 degree direction corresponding to two trace segments traversing the same corridor segment in the same or opposite direction.

With the addition input from WiFi, we can proceed to identify all the unique atomic location segments (ALS) contained in the video. We consider the $K + 1$ video segment one by one, as described in **Algorithm 2**. If a segment l_i has not been identified as a match to any segment already examined, we assign a new ALS ID, and l_i will become the reference segment of this new ALS ID. The directionality of the reference segment is labelled as +1. For a segment l_j , if $TestVideoSeg(l_i, l_j)$ is true in **Algorithm 1**, it will

be given the same ALS ID and assigned a directionality of +1. On the other hand, if $TestHCP(l_i, l_j)$ is true but $TestVideoSeg(l_i, l_j)$ is false in **Algorithm 1**, it will be given the same ALS ID but assigned a directionality of -1. After all video segments are examined, the result is a set of unique ALSs $E = \{E_1, \dots, E_{N^e}\}$, where N^e is the number of unique ALSs identified. Knowing the set of unique ALSs, we can further identify the set of unique BPs which connects to the unique ALS. Let the set of unique BPs be $U = \{U_1, \dots, U_{N^u}\}$. Further, we produce through **Algorithm 3** a vector $D = \{D_1, \dots, D_{N^d}\}$ that describes the turning direction between any two ALSs that are connected as follows: D_i is a vector that includes the two connected ALSs (E_i and E_j), the BP (U_k) that connects them, and a turning direction indicator T_{ij} which is either L or R.

Algorithm 2 Unique ALS Identification

```

1:  $e = 1$ 
2: for ( $i = 1, i \leq N + 1, i++$ ) do
3:   if  $l_i$  is unlabeled then
4:      $l_i = +e$   $\triangleright$  The first direction found is label as positive.
5:     for ( $j = i + 1, j \leq N + 1, j++$ ) do
6:       if  $l_j$  is unlabeled then
7:          $V_s(i, j) = TestVideoSeg(l_i, l_j)$ 
8:          $cpc(i, j) = TestHCP(l_i, l_j)$ 
9:         if  $V_s(i, j) = 1$  and  $cpc(i, j) = 1$  then
10:           $l_j = +e$ 
11:        else if  $V_s(i, j) = 1$  and  $cpc(i, j) = 0$  then
12:           $l_j = -e$ 
13:        end if
14:      end if
15:    end for
16:     $e = e + 1$ 
17:  else if  $l_i$  is labeled then
18:    for ( $j = i + 1, j \leq N + 1, j++$ ) do
19:      if  $l_j$  is unlabeled then
20:         $V_s(i, j) = TestVideoSeg(l_i, l_j)$ 
21:         $cpc(i, j) = TestHCP(l_i, l_j)$ 
22:        if  $V_s(i, j) = 1$  and  $cpc(i, j) = 1$  then
23:           $l_j = l_i$ 
24:        else if  $V_s(i, j) = 1$  and  $cpc(i, j) = 0$  then
25:           $l_j = -l_i$ 
26:        end if
27:      end if
28:    end for
29:  end if
30: end for
31: return  $N^e = e$ 

```

Algorithm 3 Turning Direction Vector Generation Algorithm

```

1:  $j = 1, k = 1$ 
2: for ( $i = 1, i < N^e, i++$ ) do
3:    $D(i, 1) = E_i$ 
4:    $D(i, 2) = E_j$ 
5:    $D(i, 3) = U_k$ 
6:    $D(i, 4) = T_{ij}$   $\triangleright T_{ij}$  is the turning direction either left or right.
7:    $j = j + 1, k = k + 1$ 
8: end for

```

5. Floor Map Construction

With the set of vertices U , the set of unique edges E and the set of turning vectors D , the floor map construction algorithm aims to create a planar graph $G = (E, U, D)$ that is more intuitive for human observers to read. There are three steps in the algorithm: (1) Depth First Block

Search (DFBS), (2) path search, and (3) straight-line and turning direction embedding. Steps (1) and (2) follow [1] and step (3) incorporates information on turning directions to construct a floor map that would more closely resemble the actual physical map.

1. Depth First Block Search

The step of DFBS is based on Tarjan's DFS block search algorithm [16]. We also follow the notation in [1]. DFS starts from a vertex of G with the highest node degree and chooses an edge to follow. Traversing the edge leads to new vertex. If it reaches the end vertex in the path, it goes back to the preceding vertex and goes to another unexplored edge. It stops when all the edges are explored. DFS labels each vertex with a DFS number $DFSN(v)$ and create a spanning tree for the path search in step (2). For any vertices v and w in U , the spanning tree is constructed by a set of arcs $v \rightarrow w$, where $DFSN(v) < DFSN(w)$, and a set of fronds $v \dashrightarrow w$, where $DFSN(v) < DFSN(w)$.

Moreover, DFS assigns each vertex a number called the low point value (LPV) to determine whether a 'block' is a biconnected component of the graph or not. A biconnected component of graph G is a subgraph G_i . such that the remaining graph remains connected if the biconnected component is removed. For any vertex ($v \in U$), its low point value is defined as:

$$LPV(v) = \min(\{DFSN(v)\} \cup \{LPV(w) \mid v \rightarrow w\} \cup \{DFSN(w) \mid v \dashrightarrow w\}), \quad (3)$$

Initially, the low point value in a vertex v is set equal to its $DFSN(v)$. After all of the low point values are calculated, we look for the vertices with $DFSN(v) \leq LPV(v)$, excluding the start vertex of the spanning tree. A path including these vertices is grouped into a sub-graph G_i called a block. Figure 3 shows an example of spanning tree. Each node is a vertex in the graph, and the numbers next to each node are the DFS number and low point value respectively. There are three blocks in the example. The largest block is the original graph with edges $(2, E)$ and $(1, S)$ excluded, and $(2, E)$ and $(1, S)$ each forms its own block, while S and E is the start and end BPs of the video path. The ALS ID of each edge can be identified by its start and end vertices and is labelled by E_1 to E_{18} in Figure 3.

2. Path Search

In path search, we search all blocks one by one. In the beginning, all the vertices and edges in the spanning tree are labeled as unexplored. We start from the vertex with the smallest DFS number in a block and mark it as explored. Then we extend the path to the next unexplored edge that is connected to the vertex until all of the vertices in the block are explored. The output of the search of each block is a path that covers all vertices in the block. The search step is completed when all the blocks are explored.

3. Planar Embedding

In this step, we use the paths identified in step (2) to draw

the floor map. We assume that all edges/corridor segments are straight-lines. While [1] uses a planar layout algorithm based on [17], here we can simply make use of the direction relationships described in the matrix D . The result is a draft floor map that shows the relationship between vertices and edges. An example is shown in Figure 8.

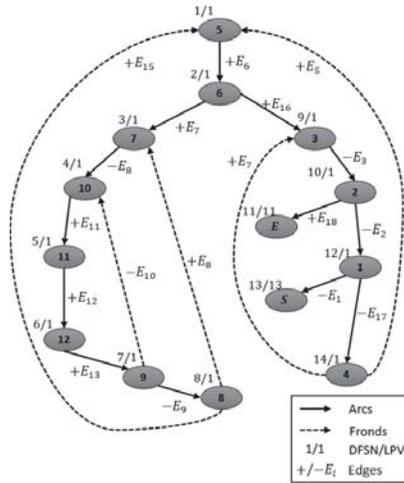


Fig. 3: A sample of spanning tree.

6. Room Number Finding Algorithm

Having a floor map that is labeled with room numbers is useful for navigation and other applications. In our Room Number Finding Algorithm, we make use of the WiFi RSS trace associated with clip 2 to reduce the search process in finding potential doorplate frames in the survey video. We have also applied different methods for preprocessing the image for character recognition.

6.1 Doorplate Frame Extracting

First, we need to find the video frames in which a doorplate may be present. To increase the searching speed, we get help from the WiFi RSS data. We compute a single-trace correlation (STC) which is the autocorrelation of the WiFi RSS trace to identify the times that the surveyor is pausing in front of a doorplate. The curve in Figure 4 shows the autocorrelation of the WiFi RSS sequence. Peaks in the autocorrelation curve reflect the potential times at which the surveyor is pausing in front of a doorplate. From the figure, we see that a video clip with more than 18000 frames can be reduced to about 80 frames for which the surveyor is potentially pausing. It needs to promise that there is a doorplate template in order to find the accurate doorplate frames inside the potential frames. The number of matching interest points between the template and each found frame will be calculated. The frame will be determined to be a doorplate frame if the number of matched interest points is larger than a threshold.

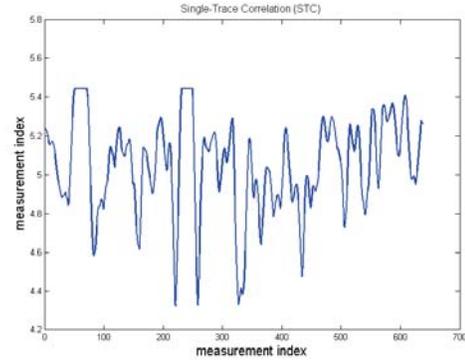


Fig. 4: The autocorrelation of the WiFi RSS measurements of a path.

6.2 Character Recognition

Character recognition is the final step of the Room Number Finding Algorithm. We use optical character recognition (OCR) [11] technique to perform the character recognition. There are standard steps in OCR for off-line characters recognition [14], [11], [13]:

1. Preprocessing: Before applying any feature extraction technique, the character images must be converted into a black-and-white image. The recognition performance is often highly dependent on this step, and we will discuss in greater details this later.

2. Location and segmentation: Segmentation is the segregation of words into individual alphabets/characters. The segmentation step typically works under the assumption that the characters are not connected together.

3. Feature Extraction, or template-matching and correlation method: There are two methods to complete the final step of OCR. The first method is to extract features contained in the character represented as a binary matrix. The second method is to compute the distance between the character with a set of templates to find the best match. We use the second method and the templates are provided in [12].

6.3 Binarization

The preprocessing, especially the binarization, is extremely important to the OCR performance [11]. Binarization methods can be classified as local or global. Global binarization methods include the Fixed Thresholding Method, Otsu Method and Kittler Method, while local binarization methods include the Niblack Method, Adaptive Method, Sauvola Method and Bernsen Method [15]. As [15] points out that the Otsu Method and Sauvola Method produce the best result for global binarization and local binarization respectively, we will compare only these two methods and choose one to perform the binarization step. Figure 5 shows that the Otsu Method gives a better binarization quality in

our case. Hence, the Otsu Method is used as the binarization method.



Fig. 5: The image under different binarization methods. (a)The original gray image. (b)The image under binarization with the Otsu Method. (c)The image under binarization with the Sauvola Method

7. Experiment and Result

The experiment took place in the lab area on the second floor of the academic building of our university. The actual floor map of the area is shown in Figure 6. In the experiment, a student equipped with an LG Nexus 5 walks at a relatively constant speed around the area. Two traces, one for the Intersections Finding Algorithm and the other for the Room Number Finding Algorithm, are recorded. Both videos are in 720p and WiFi RSS values are recorded along with them. The first trace starts from point A, and covers the area in a zigzag pattern until ending at point B. The second trace travel through the area randomly and stops for seconds to capture doorplates.

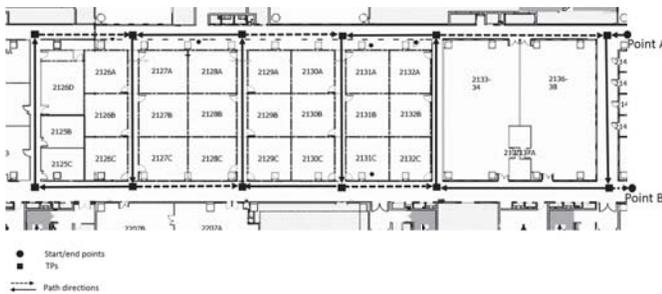


Fig. 6: Physical topology of corridors in survey area

7.1 Experimental Result for TPs Finding Algorithm

The BPs and turning directions in the trace are determined by the Intersections Finding Algorithm. As shown in Figure 7, the trace passes through 12 different BPs a total of 22 times, passing through 10 BPs twice. The result of IFA is shown in Figure 7. A threshold is applied into the curve in Figure 7(a) to obtain the curve in Figure 7(b). Figure 7(b) shows that all the BPs, each indicated by a zero region of the curve, have been found by the algorithm.

The turning time of the BPs, taken as the middle of the zero regions, are shown in Table 1. The turning directions at each BP is also shown in the table, where 1 indicates a left turn and 0 a right turn.

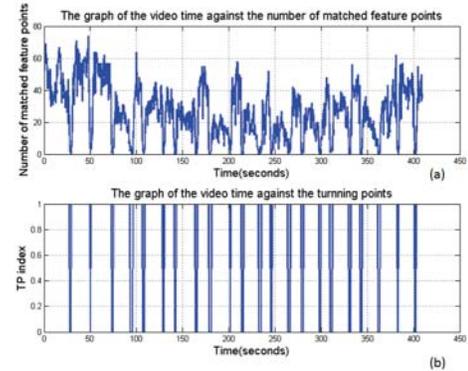


Fig. 7: BPs Finding. (a) Number of matched interest points as function of time. (b) Binarized Graph.

Table 1: The turning time and its turning direction in the trace

The n^{th} BP	Time (seconds)	Turning direction (1/0)
1	28.7	1
2	50.4	0
3	74.2	0
4	94.7	1
5	107.9	1
6	129.4	0
7	142.2	0
8	164.9	1
9	179.9	1
10	201.9	0
11	215.0	0
12	233.9	0
13	246.2	0
14	265.5	1
15	279.7	1
16	298.9	0
17	311.0	0
18	331.2	1
19	343.5	1
20	362.7	0
21	283.2	0
22	402.0	1

7.2 Experimental Result for Unique ALSs Labeling and Map Construction

Table 1 shows all the BPs in video clip 1 and table 2 shows the level of matching of the segments between the BPs. The high matching values are shaded, indicating that the two corresponding segments are the same ALS. Each unique ALS is assigned an ID. From the table, 5 pairs of segments are repeated segments so there are 18 unique ALSs. After identifying the unique ALSs, the data will be passed to the WiFi-RSS based system to construct the radio map. The resulting draft floor map shown in Figure 8 can also easily construct by followed the steps in floor map construction.

7.3 Experimental Result for Room Number Finding Algorithm

There are 34 rooms in the survey area, as shown in Figure 6. The surveyor goes around records all the doorplates in video clip 2. The result of the room number finding algorithm is shown in Table 3. There are 3 doorplates

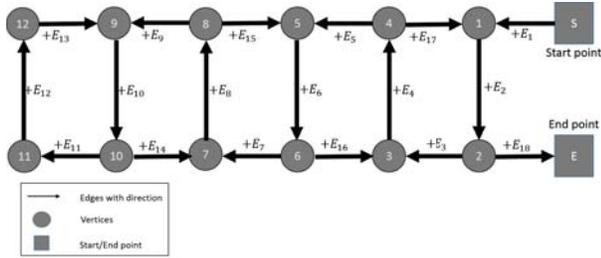


Fig. 8: The resulting graphical floor map

Table 2: The average matched interest point(A_m) between two trace segments

Segments	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1 (ALS ₁)	-	0.6	0.50.4	1.40.8	0.60.5	1.20.4	1.20.9	0.70.1	0.30.5	1.0	0.4	0.40.7	0.30.3	0.5										
2 (ALS ₂)	0.6	-	2.12.2	1.0	1.11.3	0.31.5	1.71.5	1.60.9	0.32.1	0.80.8	0.61.1	0.9	47.60.6											
3 (ALS ₃)	0.52.1	0.5	0.90.1	1.80.9	0.60.8	2.41.1	1.25.0	0.50.7	1.0	0.3	0.81.1	1.82	0.6											
4 (ALS ₄)	0.42.2	0.5	0.91.4	0.33.2	0.50.7	0.61.9	1.11.5	0.42	0.21.4	0.2	20.5	0.71.2	0.5											
5 (ALS ₅)	1.41	0.90.9	0.1	0.50.4	3.80.4	0.40.7	1.40.5	0.70.1	0.70.4	0.80.7	0.40.8	0.4	0.8											
6 (ALS ₆)	0.80.9	0.11.4	0.1	1.73.6	0.64.2	0.91.5	1.26.6	0.56	0.4	58.1	0.21.6	1	2	0.3										
7 (ALS ₇)	0.61.1	1.80.3	0.51.7	0.5	1.06.6	1.31.1	1.60.1	0.70.8	1	0.2	0.60.4	0.81	0.7											
8 (ALS ₈)	0.51.3	0.93.2	0.43.6	0.5	0.52	0.21.8	1	0.4	39.2	0.92.5	0.62.4	0.31.7	0.4											
9 (ALS ₉)	1.20.3	0.60.5	3.80.6	1	0.5	0.30.5	1.21.1	0.10.7	0.1	0.40.1	1.70.2	0.30.7	0.3											
10 (ALS ₁₀)	0.41.5	0.80.7	0.44.2	0.62	0.3	0.81.8	0.73.9	0.32.2	0.43.7	0.71.5	0.61.5	0.2												
11 (ALS ₁₁)	1.21.7	2.40.6	0.40.9	1.30.2	0.50.8	1.31.1	1.10.3	0.30.8	1.70.3	0	1.6	0.31.2	1											
12 (ALS ₁₂)	0.91.5	1.11.9	0.71.5	1.11.8	1.21.8	1.13	1.40.6	0.11.7	0.71	0.51.8	0.72	0.6												
13 (ALS ₁₃)	0.71.6	2.51.1	1.41.2	1.61	1.10.7	1.11.4	0.20.1	0.10.6	2.60.5	0.50.3	0.50.9	0.3												
14 (ALS ₁₄)	0.10.9	0.31.5	0.56.6	0.12	0.139.1	0.60.2	0.32.7	0.55.4	0.20.8	0.51.9	0.3													
15 (ALS ₁₅)	0.30.3	0.50.4	0.70	0.70.4	0.70.3	0.30.1	0.10.3	1	1.10.4	2.60.7	0.71.3	0.7												
16 (ALS ₁₆)	0.52.1	0.72	0.15.6	0.8	39.2	0.12.2	0.81.7	0.62.7	1	0.73.4	0.22	0.21.5	0.2											
17 (ALS ₁₇)	1	0.8	1	0.2	0.70.4	1	0.9	0.40.4	1.70.7	2.60.5	1.10.7	0.3												
18 (ALS ₁₈)	0.40.8	0.31.4	0.4	58.1	0.22.5	0.13.7	0.31	0.55.4	0.43.4	0.6	0.21.4	0.51.8	0.5											
19 (ALS ₁₉)	0.40.6	0.80.2	0.80.2	0.60.6	1.70.7	0.50.5	0.22.6	0.2	0.50.2	0.5	0.60.6	0.4												
20 (ALS ₂₀)	0.71.1	1.1	20.5	0.71.6	0.42.4	0.21.5	1.61.8	0.30.8	0.72	0.31.4	0.5	0.60.6	0.4											
21 (ALS ₂₁)	0.30.9	1.80.7	0.41	0.80.3	0.30.6	0.30.7	0.50.5	0.70.2	0.70.5	0.60.5	1	0.4												
22 (ALS ₂₂)	0.3	47.62	1.2	0.82	1	1.7	0.71.5	1.22	0.91.9	1.31.5	0.41.8	0.62.4	1	0.7										
23 (ALS ₂₃)	0.50.6	0.60.5	0.40.3	0.70.4	0.30.2	1	0.60.3	0.30.7	0.2	0.30.5	0.40.4	0.40.7	0.3											

discarded because of failed matching with the template. Two of these are because of low brightness in the environment, as the performance of SIFT is highly sensitive to differences in brightness. The third is because of a low number of matched interest points with the template.

Table 3: The number of doorplates under different stage of the algorithm

The stage of algorithm	Percentage of doorplate under the stage
Correct output doorplates	90.6%
Incorrect output doorplates	0%
Objective doorplates discarded by matched with template	9.4%
Objective doorplates discarded by STC	0%

8. Conclusion and Future Work

We have proposed to use video data to assist crowd-sourced WiFi RSS data in floor map and radio map construction for indoor WiFi localization applications. This paper shows that intersections and turning directions can be accurately determined from video data, by using SIFT to match interest points in successive video frames. This paper focuses on indoor areas that are grid-like, with corridor segments joining at intersected. The next step of our work

will be to include different types of areas such as open flow and enclosed areas.

References

- [1] X. Luo, A.K.S. Wong, M. Zhou, X.N. Zhang, and C.T. Lea, "Automatic Floor Map Construction for Indoor Localization," *The Tenth Advanced International Conference on Telecommunications*, 2014, p.155–160.
- [2] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [3] Armin Gruen, "Adaptive least squares correlation: a powerful image matching technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, pp. 175–187, 1985.
- [4] F. Zhao, Q.M. Huang, and W. Gao, "Image Matching by Normalized Cross-Correlation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, May 1985.
- [5] Yan Ke, and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 506–513, Jun.2004.
- [6] W.T. Freeman, and E.H. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 891–906, 1991.
- [7] J.J. Koenderink, and A.J. van Doorn, "Representation of local geometry in the visual system," *Biological Cybernetics*, vol. 55, pp. 367–375, 1987.
- [8] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Speeded-Up Robust Features (SURF)" *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.
- [9] L. Juan, and O. Gwun, "A comparison of sift, pca-sift and surf" *International Journal of Image Processing (IJIP)*, vol. 3, pp. 143–152, 2009.
- [10] N.Y. Khan, B. McCane, and G. Wyvill, "SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset" *International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 501–506, Dec.2011.
- [11] Eikvil Line, "Optical character recognition", 1993.
- [12] (2006) OCR (Optical Character Recognition) website. [Online]. Available: <http://www.matpic.com/>
- [13] G. Vamvakas, B. Gatos, and S.J. Perantonis, "Handwritten character recognition through two-stage foreground sub-sampling" *Pattern Recognition*, vol. 43, pp. 2807–2816, 2010.
- [14] N. Arica, and F.T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting" *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31, pp. 216–233, 2001.
- [15] N.K. Garg, "Binarization Techniques used for Grey Scale Images" *International Journal of Computer Applications*, vol. 71, 2013.
- [16] R. Tarjan, "Depth-first search and linear graph algorithms" *SIAM journal on computing*, vol. 1, pp. 146–160, 1972.
- [17] P. Rosenstiehl, and R. Tarjan, "Rectilinear planar layouts and bipolar orientations of planar graphs" *Discrete & Computational Geometry*, vol. 1, pp. 343–353, 1986.