3D Wi-Fi Signal Strength Mapping

Marc Katzef
Department of Computer Engineering
University of Canterbury
Christchurch, New Zealand
mka122@uclive.ac.nz

Richard Green
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract*—This paper proposes a method to generate three-dimensional (3D) visualisations of signal strength of an 802.11n (Wi-Fi) network within a small office/home office environment. The proposed system collects signal strength data using a pair of Wi-Fi-enabled microcontrollers affixed to a rod – the position of which tracked using images from a depth camera, and merged with the signal strength information to form a 3D point cloud. Through the combination of Kalman filtering to predict a search region and thresholding and contour finding for marker detection, the proposed system could track the position of markers on the rod with an accuracy of 4, 3, and 8 mm in the camera's x, y, and z directions respectively, slightly exceeding the expected maximum of 4 mm in each direction. An uncertainty propagation calculation yielded a tracking accuracy of 10, 10, and 20 mm for the microcontrollers themselves, which proved sufficient to show trends in signal strength in 3D.**

Keywords—heatmap, signal strength, ESP8266, 3D tracking, Kalman filter

I. Introduction

Wireless communications have been an important aspect of society for many decades. From radio broadcast to high-speed 4G networks, technological advancement has helped shape how the world accesses information, with wirelessly-communicating portable devices now ubiquitous. While interactions with the myriad of wireless networks surrounding us are often seamless, wireless communications have their limitations. Dependent on radio waves, the signal strength of wireless communications decreases with increased distance between receiver and transmitters, and worsened due to interference from obstructions and even other radio waves [1].

As radio waves fall outside of the visible spectrum, identifying reduced signal quality regions (dead zones) can prove difficult. This report proposes a method to visualize signal strength of the 802.11n class of wireless communication (Wi-Fi) within an indoor space. The proposed method involves the combination of a Wi-Fi-scanning device and a depth camera-based tracking system respectively collecting signal strength and three-dimensional (3D) position information to be merged, forming a point cloud.

II. Background

Most existing research into the signal strength of wireless networks (in indoor environments) considers variation in only two dimensions. By taking signal strength measurements at evenly-spaced points at a constant height, 2D heatmap visualizations like that shown in Figure 1 may be produced.

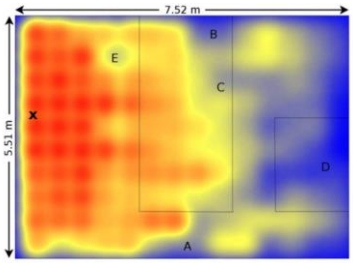

Figure 1. Wi-Fi signal strength heatmap for a small flat [2].

2D Wi-Fi heatmaps can convey the effect of phenomena such as reflection, refraction, diffraction, and absorption as they act on transmitted radio signals. This information can be used to determine suitable locations for wireless devices. With walls, household objects, and people themselves each contributing to interference effects, these visualisations show complex variations. However, 2D Wi-Fi heatmaps do not consider any signal strength variation which occurs with variation in height.

Typical Wi-Fi signal strengths vary from -30 dBm (very strong) to -90 dBm (very weak) [3]. With approximately 100 mm of the human body attenuating a 2.4 GHz radio signal (a frequency used by 802.11n) by 17 dB [4], interference can force sudden changes in signal strength – rendering 2D Wi-Fi heatmap values unreliable at other heights.

Existing implementations of 3D signal strength mapping have targeted two opposing scales - large, course-grained and small, fine-grained. The produced mapping tools have depended on drones [5, 6] or computer numerical control (CNC) milling machines [7] respectively to position a signal-scanning device. Neither of these approaches aimed at generating the 3D equivalent of the heatmap shown in Figure 1.

The found drone signal strength mapping was aimed at two use-cases – identification of network vulnerability, and the convenient surveying of cellular network strength. Both tasks involved scanning signal strength throughout a large outdoor volume. While moving a drone through the tested regions, the signal strength of the tested networks was sampled to merely identify the positions in which the networks were usable. Due to the use of global positioning system (GPS) tracking, the positioning accuracy was low, the best being only 0.63 m [5].

Theoretically, the CNC milling machine approach would be expected to achieve positioning accuracy within 0.02 mm [8]. Instead of relying on GPS location information like the drone approach, the CNC milling machine could accurately position the an ESP8266 (a Wi-Fi-enabled microcontroller) at any

location within the mill, as requested by the connected computer. The signal strength read by the ESP8266 could then be sampled at a high frequency and merged with position data by the same computer. Once collected, this data could be composited into a 3D image as shown in Figure 2.

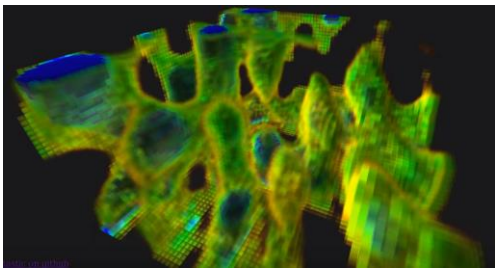

Figure 2. 3D Wi-Fi signal strength map from data collected using a CNC milling machine.

The CNC mill approach's accuracy comes at the cost of practicality. The CNC mill used to generate the image shown in Figure 2 had a functional volume measuring 360 x 360 x 180 mm. This range limitation forces many separate scans to be performed to map a room like in Figure 1. Notably, as Figure 2 shows the relative Wi-Fi signal strength as an array of 64 x 64 x 32 (discrete) voxels, this level of accuracy is orders of magnitude greater than required. Even with the voxels measuring 5.625 mm in each dimension, variation in signal strength is visibly smooth.

The CNC mill approach was extended to avoid the size limitation using long-exposure photography and an RGB light emitting diode (LED) fitted to the ESP8266. As the ESP8266 was moved, the approximate position and signal strength data could be recorded by a camera (where the colour of the LED was chosen based on the current signal strength). Position and signal strength data would then need to be recovered in post processing – a step presenting severe reductions in accuracy for both data streams. No results were published regarding the final accuracy of this approach.

While the image processing technique investigated for Wi-Fi mapping reduced the accuracy of both position and signal strength readings, the solution allowed arbitrary motion (within the field of view of the camera) to be recorded. By substituting the long-exposure photography component with camera-based 3D object tracking, this approach could achieve high levels of positioning accuracy and practicality. A suitable tool for this 3D object tracking is an inexpensive time-of-flight (TOF) depth camera such as that contained in the Microsoft Xbox Kinect Sensor v2 (referred to as the "Kinect Sensor" in this report) shown in Figure 3. By emitting light and observing the phase shift of that which is reflected [9], the Kinect Sensor can measure depth with an accuracy of below 4 mm [10] with the target at or below 3.5 m from the sensor. With increasing distance, or poor target reflectivity [11] the accuracy is measured to rapidly drop, reaching 15 mm at 4.5 m [12].

Figure 3. The Microsoft Xbox Kinect Sensor v2, with camera space coordinate frame attached.

The Kinect Sensor can collect arrays of the mentioned depth values measuring 512 x 424 at 30 frames/s [13]. To translate this data into position information for the signal scanning device, a 3D tracking method was required. For this purpose, three existing approaches were identified – correspondence grouping, particle filtering, and contour finding.

The correspondence grouping approach interprets the collected depth images as point clouds, comparing them with a 3D model of the tracked object (using a 3D Hough voting scheme or geometric consistency [14]). If a collection of points is found to match the 3D model, the output of such a comparison is a 6 degrees of freedom (DOF) pose of the object in the scene, with the same accuracy as the points in the searched point cloud. An implementation (which uses 3D Hough voting) was measured to achieve a 91.6% successful identification rate [15]. However, processing a single frame took an average of 0.507 s (giving a rate of 1.97 frames/s).

The particle filter approach also acts on the point clouds themselves, but uses a set of predicted object poses (hypotheses) and their corresponding likelihoods (weights) when compared to the received point cloud. The approximated pose is found as the sum of weighted hypotheses. An implementation of this method (which tracked any moving object in view) achieved a rate of successful detection over 99% [16]. However, even with the processing power of a dedicated GPU, a maximum rate of only 21 frames/s was achieved.

Contour finding was found to be the least computationally-demanding method, primarily due to the method acting on (2D) binary images. Instead of searching the point clouds from received depth images, contour finding identifies collections of pixels belonging to a continuous feature (a contour) in a binary image using a border following technique [17]. The identified contours may then be compared with a simple 2D representation of the tracked object for positioning information within the image. Using image coordinates, the depth values at those coordinates, and the intrinsic parameters of the depth sensor (specifically, centre, c, and focal length, f, for both x and y), the 3D position (but not orientation) of the tracked object may be calculated through Equations 1 and 2.

$$y_{camera\ space} = \frac{y_{image,depth} - c_{y,depth}}{f_{y,depth}} * z_{camera\ space} \quad (1)$$

$$x_{camera\ space} = \frac{x_{image,depth} - c_{x,depth}}{f_{x,depth}} * z_{camera\ space} \quad (2)$$

As the contour finding method requires only a binary image, the depth or colour image from the Kinect Sensor may be used. To use the (higher resolution) colour images instead, an additional mapping (dependent on the translation and rotation between cameras and the colour camera's intrinsic parameters)

must be performed from colour image coordinates to depth image coordinates. This mapping may be performed internally in the Kinect Sensor, however, the required registration between the cameras is of low accuracy [18]. However, the accuracy of either implementation could be improved by making a common extension to the method – incorporating a Kalman filter. Requiring only a covariance matrix and regular observations as input, a Kalman filter can predict the approximate location of the tracked object [19]. These predictions can be used to smooth noisy observations, indicate likely locations, and even estimate other parameters such as the radius of the tracked object [20]. However, when using a linear model in the Kalman filter system, their output becomes unreliable while tracking accelerating objects.

III. Method

The proposed 3D Wi-Fi signal mapping method consists of two components – a Wi-Fi scanning device and a 3D tracking system. Together, these components could collect received signal strength indication (RSSI) readings at precise locations in 3D. Implementation details for the components, the data processing, and testing procedures are described in the following subsections.

A. Wi-Fi Scanner

The Wi-Fi-scanning device (referred to as the "scanner") needed to satisfy three constraints – it needed to be able to stream RSSI readings to a connected computer, it needed to have clearly-identifiable features for 3D tracking, and it needed to be portable. The resulting design (shown in Figure 4) consisted of five ESP8266 microcontrollers, a pair of tennis balls, and cardboard tubing. The ESP8266 microcontrollers were chosen as they are compact and inexpensive devices capable of communicating using the 802.11 b/g/n standards [21], and being programmed with the simplicity of Arduino devices [22]. These features have made the ESP8266 series the ideal candidate for this and other RSSI-based studies [7, 23]. The tennis balls were chosen to act as visual markers due to their simple shape, high visibility (from many viewing angles), and (like the cardboard tubing) due to their low weight and low cost.

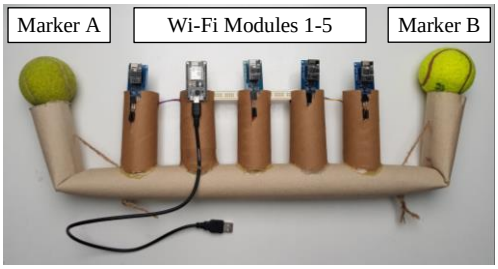

Figure 4. The produced Wi-Fi scanning device.

To allow for either depth- or colour-based tracking methods, the scanner was designed to situate the visual markers away from the scanner's handle. To further aid the 3D tracking effort, the antennae of the Wi-Fi modules were situated along the straight line connecting the visual markers centre-to-centre. This placement required the tracking system only to consider the position of the two markers (and not their rotation). With the 3D position of each marker ($\mathbf{p}_a$ and $\mathbf{p}_b$) identified by the tracking system and the distance from marker A or B to each antenna ($\mathbf{p}_{a1}$, p_{a2}, p_{a3}, p_{b4}, and p_{b5}) known (shown in Table 1), the position of the antennae ($\mathbf{p}_1$ to $\mathbf{p}_5$) could be calculated using Equation 3 or 4 (whichever yields the lowest uncertainty).

Table 1. Scanner dimensions, with measurement uncertainties.

Parameter	Nominal Value (mm)	Uncertainty (mm)
d_{ab}	490	10
d_{a1}	110	5
d_{a2}	185	5
d_{a3}	255	5
d_{b4}	165	5
d_{b5}	100	5

$$\boldsymbol{p}_n = \boldsymbol{p}_a + \hat{\boldsymbol{u}}_{ab} * d_{an} \quad (3)$$

$$\boldsymbol{p}_n = \boldsymbol{p}_b - \hat{\boldsymbol{u}}_{ab} * d_{bn} \quad (4)$$

Where $\hat{\boldsymbol{u}}_{ab}$ is the unit vector in the direction from marker A to marker B, calculated using Equation 5.

$$\hat{\boldsymbol{u}}_{ab} = \frac{\boldsymbol{p}_b - \boldsymbol{p}_a}{\|\boldsymbol{p}_b - \boldsymbol{p}_a\|} \quad (5)$$

The cardboard tubing did not need to house any more electrical connections than shown in Figure 4. Also shown in Figure 4, one of the ESP8266 modules was in a package which supported USB-connectivity. While this could have been used to transfer collected signal strength information, the wired connection would have limited the device's portability, and would have complicated the data acquisition process for the one module acting as host for the others. Instead, this connection was used for power alone; each device was programmed to act as a simple web server (providing web pages as shown in Figure 5) with which a host could communicate individually.

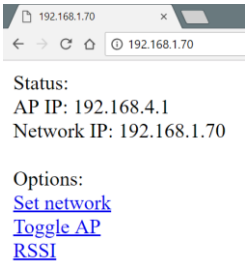

Figure 5. The programmed ESP8266 configuration web page.

On start-up, each ESP8266 would enter "soft access point" mode, forming an open network to which a Wi-Fi-enabled device may connect. Once connected, a device may send HTTP POST or GET requests (destined to the ESP8266's local IP address) to remotely perform any of the following actions:

1. Supply the ESP8266 with the SSID and password for a WEP, WPA, or WPA2 (non-enterprise) protected Wi-Fi network,
2. Toggle the ESP8266's soft access point network,
3. Get a current RSSI reading from the ESP8266.

Typical use of the scanner begins by performing actions 1 and 2 (above) on each ESP8266 to connect it to the scanned

network (recording the IP addresses awarded to each module) and to disable the soft access points (ruling out any signal interference). Before step 3, the device used to configure the ESP8266 modules must connect to the same network as the modules. From that time onwards, the RSSI measured by each ESP8266 could be sampled by sending an HTTP GET request to the individual device IP addresses. The ESP8266 modules would promptly respond with a message containing the current RSSI value (in dBm, rounded to the nearest integer) collected using the Arduino Wi-Fi API [24]. To simplify the operation of the 3D tracking components and improve performance, the communication with ESP8266 modules was performed in a dedicated thread for each module on the host PC.

B. 3D Tracking

The proposed 3D positioning solution was implemented on a personal computer with an Intel® Core™ i5-4300U (dual core) CPU with maximum frequency of 2.5 GHz, and 4 GB of RAM installed. The development machine was running Microsoft Windows 10 and had installed Microsoft Visual Studio Community 2015, Kinect for Windows SDK v2.0, and OpenCV 3.4.1. The positioning solution consisted of two phases – initialisation and tracking. These phases consisted of largely the same steps, described in the following subsections.

1) Initialisation

The initialisation step was responsible for identifying the initial position of the visual markers on the Wi-Fi scanner. This was performed by analysing a predetermined region of the Kinect Sensor's received depth images for each of the two scanner markers. The region for each marker was set on a horizontal line through the depth image's centre, at 1 m from the camera. Marker A's region was centred ⅓ of the image from the left, and Marker B's ⅔. After successfully identifying a marker in either region for five consecutive frames, that marker would enter the tracking phase.

The process of marker identification consists of:

1. Cropping a depth frame to a specified size centred about the expected marker origin,
2. Thresholding the depth image,
3. Using morphological erosion and dilation,
4. Searching for contours using the OpenCV method *findContours*,
5. Fitting circles to the identified contours (using the OpenCV method *minEnclosingCircle*) in order of decreasing contour size (number of contained points) until one is found to have an origin and radius in the expected ranges,
6. Correcting a Kalman filter with the marker's properties (x, y, and radius in pixels).

Step 1 (above) is performed to reduce the size of the searched area (reducing computation cost). Step 2 eliminates all entries in the depth frame other the features in a band at the expected distance from the camera. Step 3 is performed to reduce the effect of noise on the remaining features before further processing. Step 4 is performed to identify clusters within the remaining non-zero entries in the depth image. Step 5 identifies any such cluster which may be the expected marker (by examining the contours in order of most to least reliable). Step 6 is performed to initialise the filter which is used in the tracking phase.

2) Tracking

The tracking phase mirrors the initialisation phase, but with one key difference. Instead of using hard-coded values for the expected region occupied by a marker, the tracking phase uses the Kalman filter's predictions. Once a marker has entered the tracking phase, it is assumed that the Kalman filter has been updated regularly with the marker's position and radius readings. For each received depth frame, the Kalman filter can predict the approximate value for these three parameters. The x, y, and radius values may be used directly in the identification scheme described for initialisation. The remaining required information (the expected depth) is approximated by taking the median of depth image values in a bounding box centred around the predicted x and y coordinates.

Instead of using a median value as the marker's depth, mean was tested. However, any values missing in the sample region of the depth image caused this method to be unstable. A similar reduction in stability was found when substituting radius prediction with a depth prediction in the Kalman filter. The current values for all tolerances and noise parameters were chosen as the minimum values which provided tracking deemed reliable in informal, brief tests. The minimum value was chosen to reduce the chance of falsely identifying regions of noise as the tracked objects. The resulting Kalman filter covariance matrices used process noises of 0.01 pixels for x, y, and radius, 3.0 seconds for time differences, and errors of 1.0 (pixels or seconds) for all variables.

If the marker is not found in the region predicted by the Kalman filter, no position values are recorded for that frame. If this occurs for five consecutive frames, the marker exits the tracking phase – returning to the initialisation phase. If, however, both markers are successfully identified in the same depth frame, the recording process may be performed. The first step of which being to map the position of each marker from the Kinect Sensor's depth frame to camera space. While this could be performed using Equations 1 and 2, the Kinect SDK function *GetDepthFrameToCameraSpaceTable* (which simultaneously corrects for radial distortion) was used instead.

With the 3D position of each marker known, one last check is performed to ensure the data is reasonable – the Euclidean distance between the two markers is calculated and compared with the measured scanner length. If the distance falls within 20% of the measured length, the position of each of the Wi-Fi modules is calculated (using Equation 3). These distances and the most recent RSSI value from each ESP8266 are then stored in a point cloud for later analysis.

C. Data Processing

The output of the proposed tracking process is a point cloud in the PCD file format – a concise format developed by the Point Cloud Library (PCL) [25]. Each entry in the point cloud file consists of position and signal strength information. Position is stored as a set of three floating point numbers specifying the x, y, and z coordinates of the point in the Kinect Sensor's camera space. The absolute value of the measured RSSI values are

stored as the red channel in each point's colour attribute. A PCD file reader can render this point cloud as the path through which the Wi-Fi scanner moved while being tracked, where each point is coloured dark red (due to the typical RSSI range of -30 to -90).

Two modifications were made to make the collected data more intuitive. The RSSI values were mapped to colours ranging from green (for strong signal) to red (for weak signal), and a coloured point cloud of the scanned room was recorded during start-up. The mapping from signal strength, s, to red and green colour channels, r and g respectively, was performed by comparing signal strength to some upper and lower bound as shown in Equations 6 and 7.

$$r(s) = \left\lfloor \frac{s - s_{min}}{s_{max} - s_{min}} * 255 \right\rfloor \quad (6)$$

$$g(s) = 255 - r(s) \quad (7)$$

The upper and lower bounds for RSSI could be set as the full range of usable signal strengths, or as the maximum and minimum recorded RSSI values in the data set. These ranges were respectively used to generate signal strength maps referred to as "absolute" or "relative" in this report.

A coloured point cloud of the scanned room was generated to provide bearings in visualisations of the collected signal strength points. These environment point clouds consist of points (from a depth frame) representing the surfaces visible to the Kinect Sensor, each coloured with the value of the corresponding pixel from the colour frame. Such a point cloud may be generated by mapping each pixel of the depth frame to a pixel in the colour image using the following three steps:

1. Calculate the camera space point for each pixel using Equations 1 and 2,
2. Map each camera space point to the frame attached to the colour camera (using the translation and rotation between cameras),
3. Project the colour camera space points onto the colour image using Equations 8 and 9.

$$y_{image,colour} = \frac{y_{camera\ space,colour}}{z_{camera\ space,colour}} * f_{y,colour} + c_{y,colour} \quad (8)$$

$$x_{image,colour} = \frac{x_{camera\ space,colour}}{z_{camera\ space,colour}} * f_{x,colour} + c_{x,colour} \quad (9)$$

Again, these calculations were performed using the Kinect SDK. The entire depth image could be mapped to colour space using MapDepthFrameToColorSpace (which would apply radial distortion correction).

D. Tracking Accuracy

The only aspect of the Wi-Fi scanning system to be analysed numerically was the positioning accuracy for each of the markers and Wi-Fi modules. This analysis relied on data collected by placing the Wi-Fi scanner at five different locations relative to the Kinect Sensor (with both markers in the tracking phase) and taking repeated position and RSSI measurements. Five separate test locations were chosen each at distances of 1, 2, and 3 m in the sensor's z-direction and offsets of -1, 0, or 1 m in the x-direction as shown in Figure 6.

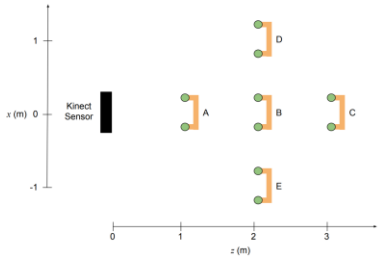

Figure 6. Scanner test positions (relative to the Kinect Sensor).

From the sets of position and RSSI data, means and standard deviations could then be calculated. Means were to be compared with manual measurements. Standard deviations were to be used in uncertainty propagation calculations (based on Equation 3) to identify the accuracy of the estimated Wi-Fi module positions.

IV. Results

The proposed Wi-Fi scanning system was could successfully record position and RSSI information as a point cloud. The contained RSSI data could then be mapped to contrasting colours and superimposed on a point cloud of the scanned room to generate visualisations like Figure 7 (where green indicates strong signal, red weaker, and pink the location of the wireless access point.

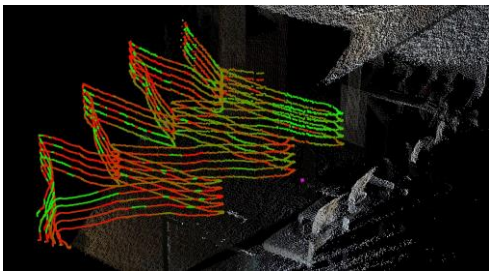

Figure 7. Relative 3D Wi-Fi signal strength map.

With basic functionality established, the accuracy measurement method in Section III.D could be performed. The measurements from this experiment are described in a following subsection along with system performance.

A. Accuracy

As described in Section III.D, repeated position and RSSI measurements were to be recorded and used to calculate statistics, before performing an uncertainty propagation calculation. This was performed where the number of samples in each location was set to 1000. The resulting standard deviations for each marker in each of the five test positions (shown in Figure 6) are shown in Table 2.

Table 2. Statistics from 1,000 position samples at each test position.

Test Position	Standard Deviation (mm)					
	Marker A			Marker B		
	σ_x	σ_y	σ_z	σ_x	σ_y	σ_z
A	1.01	0.095	1.25	1.20	0.829	1.39
B	0.489	1.81	4.08	3.12	2.20	7.92
C	2.69	2.60	5.65	3.05	2.94	5.41
D	2.79	2.51	2.65	3.65	2.01	5.40
E	1.99	2.29	3.71	2.82	2.75	2.51
Max	2.79	2.60	5.65	3.65	2.94	7.92

The expected cause for the differences in standard deviation between the two markers was the variation in texture and reflectance between the markers themselves. The maximum value for each dimension (to 1 significant figure) was used as the absolute uncertainty for each marker – representing recorded positions for the centre of marker A and B as shown in Equations 10 and 11.

$$\boldsymbol{p}_a = (x_a, y_b, z_a) \pm (3,3,6)\ [mm] \tag{10}$$

$$\boldsymbol{p}_b = (x_b, y_b, z_b) \pm (4,3,8)\ [mm] \tag{11}$$

Before calculating the three standard deviations for each marker for each test position, means were calculated and compared with manual position measurements. In each test position, the mean, minimum, and maximum reading (in each dimension) for each marker matched the corresponding measurement (within measurement tolerance). This indicated that any systematic error present was insignificant.

The uncertainty in the position of each Wi-Fi module was calculated by performing the calculation shown in Equation 3 using the measurements with their corresponding uncertainties (listed in Table 1 and the representations of position in Equations 10 and 11). As the absolute uncertainty for each marker varied across (x, y, and z) directions, this calculation was performed with the scanner in each of three orientations – with the unit vector in Equation 5 parallel with the x-, y-, or z-axis. The results for the modules found to have the best and worst uncertainties are listed in Table 3.

Table 3. Best- and worst-case W-Fi module position uncertainty in each direction.

Parallel Axis	Absolute Uncertainty (mm)					
	Module 1			Module 3		
	δ_x	δ_y	δ_z	δ_x	δ_y	δ_z
x	10.7	3.84	8.70	14.5	5.49	12.7
y	4.23	10.1	8.70	6.14	13.4	12.7
z	4.23	3.84	16.7	6.14	5.49	24.8
Max	10.7	10.1	16.7	14.5	13.4	24.8

With the worst-case uncertainty for each Wi-Fi module (in each direction) identified, the overall positioning accuracy was established as the maximum of each. The resulting uncertainty (stated to 1 significant figure) was 10, 10, and 20 mm in the x, y, and z directions respectively. Assuming scanner movement was slow enough to be comparable to the stationary measurements, the difference between every true position and approximated position was expected to be less than these tolerances.

The RSSI samples were processed like position to confirm that the Wi-Fi readings were stable. The standard deviation of each set of 1,000 samples in each test position were calculated (the best and worst module results listed in Table 4). With the largest standard deviation being 1.25 dBm (just 2.1% of the RSSI range -30 to -90 dBm), the readings were deemed suitable for mapping processes.

Table 4. Best and worst standard deviations in RSSI readings for the Wi-Fi modules in each test position.

Test Position	Standard Deviation (dBm)	
	Module 2	Module 4
A	0.560	0.000
B	0.785	1.25
C	0.694	0.782
D	0.841	0.389
E	0.547	1.04
Max	0.841	1.25

B. Performance

Through a combination of formal and informal measurements, the additional performance characteristics of speed, stability, and practicality were recorded. These are addressed individually in the following subsections.

1) Speed

While generating a point cloud, in 60 s, the host computer (described in Section III.B) could process 1,796 frames from the Kinect Sensor and receive 550 samples from each ESP8266 module. The effective frame rate was 29.9 frames/s (close to the 30.0 frames/s supported by the Kinect Sensor) and RSSI sampling rate was 9.17 Hz. This resulted in each RSSI reading being used across an average of 3.27 frames. To reduce correspondence error (between location and RSSI readings), improve tracking stability, and keep the (stationary) uncertainty findings as relevant as possible, the scanner device needed to be moved at low speeds while scanning.

2) Stability

During testing, the scanner was tracked at distances as far as 4.45 m from the sensor. From distances past 3 m, however, the tracking was unstable – dropping out while the scanner underwent only small accelerations. At distances below 3 m, the tracking system could function through moderately-paced direction changes and significant rotations. In each of the test positions, the scanner could be rotated past 45° from the line connecting the sensor to the scanner before dropping out.

The average speed for each marker on the scanner was calculated for the point clouds shown in Figure 8 and Figure 9. The average speed for marker A was calculated using Equation 12 which depends on the position of the marker in frame i, $\boldsymbol{p}_{a,i}$, for the N consecutive frames recorded at a rate of r Hz.

$$\bar{v}_a = \frac{r}{N} * \sum_{i=2}^{N} \left\| \boldsymbol{p}_{a,i} - \boldsymbol{p}_{a,i-1} \right\| \tag{12}$$

The average speed for marker A was calculated as 0.204 and 0.253 m/s for the point clouds in Figure 8 and Figure 9 respectively. The distances travelled in each scan were 19.5 and 26.9 m, taking 95.7 and 103.9 s to record. The highest average speed for an uninterrupted recording was found to be 0.256 m/s (or 0.922 km/h).

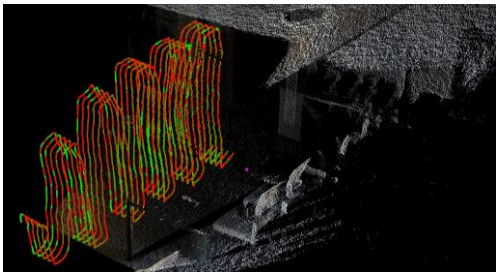

Figure 8. Relative Wi-Fi signal strength map showing vertical variation.

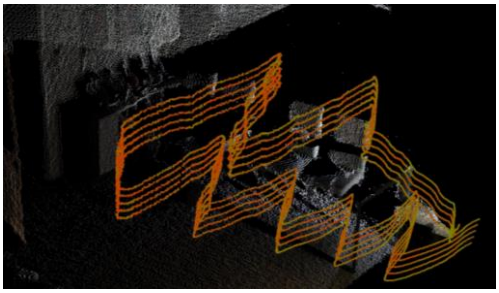

Figure 9. Absolute Wi-Fi signal strength map showing minor vertical and lateral variation (with the wireless access point not in view).

3) Practicality

While data collection was found to be a slow process, the output of the proposed system is an intuitive visualisation of the signal strength in a scanned region, as desired. As shown in Figure 8 and Figure 9, the recorded point clouds clearly indicate variations in signal strength both vertically and laterally. However, also shown in these figures is sudden signal strength variation which may be the result of measurement noise or a source of error inherent in the proposed system – the movement of the user. As the user positions the scanner in different locations, they act as a change in the system – introducing a new obstruction, generating new reflections with an unknown effect on the scanned signal.

V. Conclusion

Through the combination of Kalman filtering to predict a search region and thresholding and contour finding on a depth image for marker detection, the proposed system could reliably track the position of a Wi-Fi scanning device. The position and signal strength information was successfully merged to generate visualisations of Wi-Fi signal strength in 3D.

Each sample in the produced visualisations was taken at a sampling rate of 29.9 Hz and based on tracking accuracy below 4, 3, and 8 mm in the x, y and z directions. A position calculation for each Wi-Fi module yielded a final tracking accuracy of 10, 10, and 20 mm. This performance proved sufficient to capture trends in Wi-Fi signal strength like those observed in previous research [7]. However, these 3D visualisations covered a region larger than that which existing (indoor) solutions could achieve – a region comparable in size to that examined in typical 2D Wi-Fi heatmaps [2].

The practicality of the proposed system was hindered by restrictions on speed, acceleration, and range. Instead of the ideal case of moving the scanner swiftly through the room of interest, the user must slowly (below 1 km/h) and thoroughly trace a path (with a continuous clear line-of-sight of the camera) through the key regions of the room to generate an indication of signal strength. During implementation, possible improvements which address these shortcomings were identified, and are listed in the following subsection.

A. Future Research

Improvements could be made to the proposed system through several changes to its two core components – 3D tracking and RSSI collection. Improvements to the tracking method could reduce the Wi-Fi module tracking uncertainty to (or below) the current values for the markers themselves, and allow for faster scanner movement. Improvements to the RSSI collection scheme could reduce correspondence error, and reduce measurement noise.

The following actions were identified to improve upon the proposed tracking system:

1. Formally investigating and updating the system parameters for the Kalman filter, and the tolerances used in thresholding and marker identification,
2. Using the known distance between markers to refine the approximated marker locations,
3. Adding visual markers to the Wi-Fi modules themselves – avoiding the error introduced through uncertainty propagation.

Similarly, the following actions were identified to improve the RSSI collection technique:

1. Programming each ESP8266 to send RSSI data unprompted, reducing overhead and increasing the sampling rate,
2. Calculating a rolling average RSSI value on the ESP8266 modules to reduce the effect of noise.

References

[1] A. Cook. “5 Phenomena That Impact Wi-Fi Signal.” Internet: http://www.mirazon.com/5-phenomena-that-impact-wi-fi-signal/, [May 9, 2018].

[2] M. Kulawiak and W. Wycinka, "Dynamic Signal Strength Mapping and Analysis by Means of Mobile Geographic Information System", *Metrology and Measurement Systems*, vol. 24, no. 4, pp. 595-606, 2017.

[3] “Understanding WiFi Signal Strength.” Internet: https://www.metageek.com/training/resources/wifi-signal-strength-basics.html, [May 9, 2018].

[4] I. Dove. “Analysis of Radio Propagation Inside the Human Body for in-Body Localization Purposes.” M.Sc, University of Twente, Netherlands, 2014.

[5] S. J. Pack, “Multi-Rotor--Aided Three-Dimensional 802.11 Wireless Heat Mapping.” M.Sc, Birmingham Young University, USA, 2014.

[6] T. Metkarunchit, K. Charoenpojvajana, S. Srivichitranond, “Develop 3D Map Signal Strength Surveying System for Cellular Mobile Phone.” *Journal of Business Administration and Languages*, vol.4, pp. 44-47, Dec. 2016.

[7] C. N. Lohr. “Wifi Power Mapping.” Internet: https://hackaday.io/project/4329-wifi-power-mapping, [May 9, 2018].

[8] I. Kuric, M Kosinar, M. Cisar. “Measurement and Analysis of CNC Machine Tool Accuracy in Different Location on Work Table”, *Proceedings in Manufacturing Systems*, vol.7, pp. 259-264, Dec. 2012.

[9] L. Li, “Time-of-Flight Camera – An Introduction.” *Texas Instruments Technical White Paper*, Jan. 2014.

[10] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, A. El Saddik, “Evaluating and Improving the Depth Accuracy of Kinect for Windows v2.” *IEEE Sensors Journal*, vol.15, pp. 4275-4285, Aug. 2015.

[11] P. Fürsattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, C. Riess, “A Comparative Error Analysis of Current Time-of-Flight Sensors.” *IEEE Transactions on Computational Imaging*, vol.2, pp. 27-41, Mar. 2016.

[12] E. Lachat, H. Macher, T. Landes, P. Grussenmeyer, “Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling.” *Remote Sensing*, vol.7, pp. 13070-13097, Oct. 2015.

[13] T. Breuer, C. Bodensteiner, M. Arens, “Low-cost commodity depth sensor comparison and accuracy analysis.” Fraunhofer IOSB, 2014.

[14] “3D Object Recognition based on Correspondence Grouping.” Internet: http://pointclouds.org/documentation/tutorials/correspondence_grouping.php, [May 9, 2018].

[15] R. Beserra, G. Bruno, M. F. Silvaa, L. K. M. Rochaa, R. V. Aroca, L. C. P. R. Velho, L. M. G. Gonçalvesa, “Efficient 3D object recognition using foveated point clouds.” *Computers and Graphics*, vol.37, pp. 496-508, Aug. 2013.

[16] S. Li, S. Koo, D. Lee, “Real-time and Model-free Object Tracking using Particle Filter with Joint Color-Spatial Descriptor.” *Intelligent Robots and Systems*, Sept. 2015.

[17] S. Suzuki, K. Abe, “Topological structural analysis of digitized binary images by border following.” *Computer Vision, Graphics, and Image Processing*, 1985.

[18] C. Kim, S. Yun, S. Jung, C. S. Won, “Color and Depth Image Correspondence for Kinect v2.” *Lecture Notes in Electrical Engineering*, vol.354, pp. 333-340, Jul. 2015.

[19] H. Tao, “Object Tracking and Kalman Filtering.” Internet: https://pdfs.semanticscholar.org/presentation/6f34/99a784e6543ad0f762012e7ec00b54e58225.pdf, [May 9, 2018].

[20] R. B. Fisher, “Ball Tracking Example.” School of Informatics, University of Edinburgh, 2014.

[21] “ESP8266 Overview.” Internet: https://www.espressif.com/en/products/hardware/esp8266ex/overview, [May 9, 2018].

[22] “ESP8266 Arduino Core.” Internet: https://arduino-esp8266.readthedocs.io/en/latest/, [May 9, 2018].

[23] M. H. Habaebi, N. I. N. B. Azizan, “Harvesting WiFi Received Signal Strength Indicator (RSSI) for Control/Automation System in SOHO Indoor Environment with ESP8266.” *Computer and Communication Engineering*, Jul. 2016.

[24] “WiFiRSSI.” Internet: https://www.arduino.cc/en/Reference/WiFiRSSI, [May 9, 2018].

[25] “The PCD (Point Cloud Data) file format.” Internet: http://pointclouds.org/documentation/tutorials/pcd_file_format.php, [May 9, 2018].