

# 國立台北大學資訊工程學系專題報告

## 以室內定位為例之服務型機器人

專題組員：黃覺修、賴柏宏、張庭爾、曾偉庭

專題編號：PRJ-NTPUCSIE-107-09

執行期間：107 年 9 月至 108 年 5 月

### 1、摘要

互動式機器人過去在台灣的應用上，主攻定點或小場域服務，因為更強調情感互動和資訊諮詢，像 Zenbo 智慧居家機器人，陪伴兒童的說故事互動，提醒大人的代辦事項，到老人遠端視訊照護而 Pepper 更譽為重視情感陪伴的機器人，但若能夠擴及到大型公共場域的應用，給予人類提供的幫助將不僅如此，於是我們將機器人結合室內精準定位，達到移動的精準與便利性，希望機器人能走到我們預想的位置(座標)上，讓有如如虎添翼的機器人可以擁有更多結合室內移動的功能，而回傳的位置將之應用，比如帶路導覽功能：走到警衛室並介紹可以提供的服務，額外增添的功能有該地區空氣中 pm2.5 的值和氣溫，臉部辨識性別和年齡及增加機器人對話語句，總和以上功能做到辦公大樓的移動導覽和提供室外氣溫、空氣品質，也欲結合更多辦公功能，可望機器人走入大型公共場域。

### 2、簡介

#### (1) 研製背景

機器人(採用 Zenbo 作研究)可以做到幼兒園的家長接送時間和孩童吃藥時間提醒，但結合精準定位之後它可以走到門口辨識誰的家長來了然後呼叫該孩童的名字，更貼合人性，也不單單只有語音服務，再者機器人(例如 Pepper)也有運用到商場上，在門口提供商品位置和功能諮詢，但若結合室內定位在得知商品位置的同時，可以帶領你走到該架位取得商品，機器人這樣的功能更能吸引消費者注意，長期下來省去人力成本，達到應用的商機。

#### (2) 目標

在本專題中，希望能將 zenbo 與室內定位技術整合，並且結合雲端儲存的相關數據，達到提供使用者帶位導覽與資訊提供的服務。

#### (3) 預估成果

- 利用 UWB 精準定位技術讓機器人完成室內帶位、導覽之功能。
- 使用性別識別讓機器人與使用者的互動更人性化。
- 運用網頁爬蟲與雲端運算讓機器人能獲取使用者詢問的相關數據。

### 3、專題進行方式

#### (1) 實作平台

##### 1. Zenbo 機器人

Zenbo 機器人是華碩旗下的家用機器人，本身搭載 Android 系統，機器人頭部除了有 10.1 吋的觸摸屏幕，還有 1300 萬畫素拍照相機、紅外線控制器與觸摸感測器；機身則有防掉落感測器與多組聲納感測器。

##### 2. RESTful Server

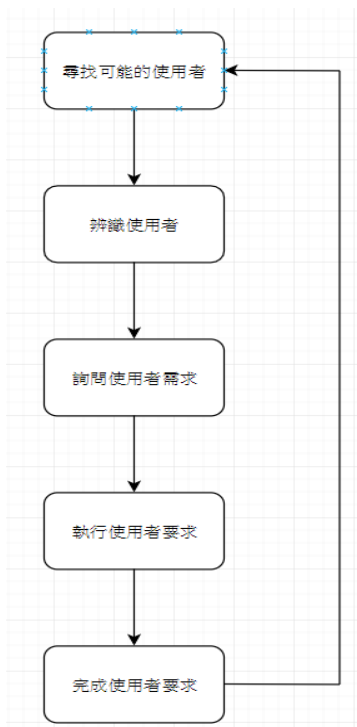
#### (2) 系統分析與設計摘要

在本專題中，將依下列方法及步驟來時作並完成計畫內容。

##### 1. 系統整體操作設計圖

在本次的計畫中機器人要在公共場所與使用者作接觸，對於機器人與使用者之間的互動需要事先設計和模擬。因此我們將機器人實際在運作時的情境以下面的流程圖呈現：

圖 1：本計畫機器人運作流程圖



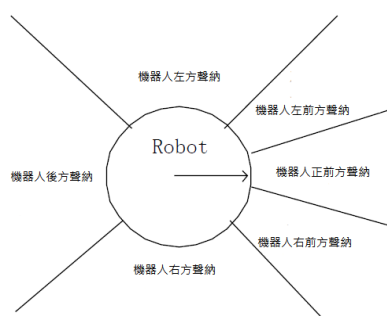
在機器人電量充足的情況下，機器人會在特定點等待使用者靠近，當使用者靠近時，機器人會判斷此人是否有與機器人互動的意願，若有意願，則詢問該使用者的相關需求，並且執行。執行完畢後若是電量充足，就會回到特定點再次等待下個使用者靠近，若是電量不足則回充電站充電。

## 2. 機器人的感測元件

### A. 聲納：

機器人身上一共有六顆獨立聲納，位置圖如下。而這些聲納可以幫助我們得知機器人與外界障礙物的距離，有助於開發移動功能時的「安全距離」的功能。

圖 2：機器人聲納位置圖



### B. 紅外線感應器：

機器人前方有四顆紅外線感應器，用途是避免因為機器人往有高低落差的地

方移動而發生危險，因此當機器人偵測到前方有很大高低落差時便會強制停止前進

### C. 相機鏡頭：

機器人的頭部基本上是一個平板電腦，而上面附有一個 1300 萬畫素的相機鏡頭，可以讓我們擷取外界的景象，這些影像可以幫助我們完成人臉辨識等需要影像的功能。

## 3. 中介層(Middleware)

uwb 基地的信號儘管可以直接傳輸機器人端而不經過一個 Server 減低延遲，但是將運算等繁雜的工作指派給不論是機器人，或是 uwb 基地的樹莓派(Raspberry Pi)上，都會造成太大的負擔。為此我們在遠端的主機架設了一個伺服器當作中介層，不僅可勝任繁雜程序的處理，將來也能更輕易地擴展其他功能。

此處我們選擇使用 Python 的 Flask 網路應用框架 (Web Application Framework)作為一個 REST 風格的伺服器。比起其他程式語言和框架，選擇 Flask 的原因為 Python 語言的強大及眾多套件資源方便擴展各式功能，以及 Flask 的方便簡約和高自由度。我們的伺服器也結合了 MongoDB 資料庫、網頁爬蟲套件，定期將我們需要的數據從氣象局網站擷取下來整理分析後，存入資料庫並提供給機器人端做應用。

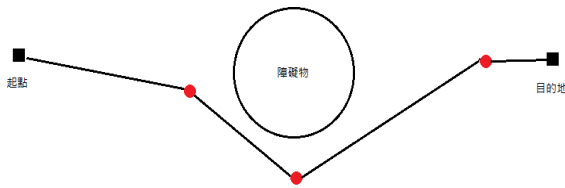
本計畫整體上是各用戶端透過對不同的 URL 進行不同的 HTTP 請求方法存取資料而運作。例如 uwb 基地對指定的 URL 進行 POST 方法傳輸座標值，而機器人端再到指定 URL 進行 GET 方法取得指令來進行移動到目的地的程序。機器人的應用程式也可以拜訪另一 URL，使伺服器端呼叫副程序執行人臉辨識的程式並回傳結果，或是獲得 PM2.5 的預測值或天氣值。

## 4. 機器人的移動控制

機器人身上會安裝我們準備好的 uwb 信標，此信標可以幫助我們得到機器人在架設基地的室內環境中之絕對座標，藉由機器人當下的絕對座標與已知的目的地絕對座標我們可以根據演算法在伺服器端算出機器人移動到目的地的最佳路線，伺服

器再將計算好的移動方向與距離傳給機器人，機器人就依照這些資訊來移動。然而機器人的移動與室內定位都會有誤差，因此需要多次的計算來校正移動路線。在這個問題中，我們將機器人的移動方式分為轉向與直行兩種，伺服器每次在下達移動指令時都是給一組轉向與直行的數據。機器人每做完一組指令便會重新向伺服器請求新的移動指令，藉由此方式來達到校正的功能。

圖 3：機器人向 server 請求移動指令示意圖



上圖為機器人在移動時的示意圖，黑線代表著機器人的移動路徑，紅色點代表著每次移動指令的結束並重新向伺服器請求新指令的時間點。

## 5. 最佳路徑演算法

預先將辦公大樓平面地圖方格座標化，每一方格單位定義為一公尺，並標示出所有障礙方格，及給予地圖大小，起始點終點，再來採用 **A\* algorithm**[5][6]:

$$F=G+H$$

F: 整體代價

G: 沿著目前路徑起點到指定方格代價

H: 指定方格到終點預測代價

此預測代價為指定方格到終點之直線距離 (歐幾里得距離 Euclidean distance)。

機器人的移動有 8 個方向，垂直移動的代價為 1，斜向移動的代價為 1 和 4，此以距離作為代價，有兩個預先設定的開啟列表和關閉 (障礙或已走) 列表，尋找開啟列表中最低的 F 值作為當前格移出開啟列表存入關閉列表，對鄰居做以下動作：

- 不可走則略過。
- 如果它不在開啟列表中，把它添加進去。把當前格作為這一格的父節點。記錄這一格的 F, G 和 H 值。
- 如果它已經在開啟列表中，拿 G 值當作參考，檢查新的路徑是否更好。更低的 G 值意味著更好的路徑。如果是這

樣，就把這一格的父節點改成當前格，並且重新計算這一格的 G 和 F 值。

重複以上不斷找出當前格和紀錄父節點，當前格為目標點或無路可走時停止，從終點回推父節點值找到起點至終點的最短路徑座標。由於最短路徑點座標過多，運用 **Basic Theta\* algorithm**[6]: 斜率不同時擷取該點簡化成僅剩轉彎點座標，機器人行走更順暢便捷。

## 6. 性別與年齡識別

group name	output size	block type = $B(3,3)$
conv1	$32 \times 32$	$[3 \times 3, 16]$
conv2	$32 \times 32$	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$
conv3	$16 \times 16$	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$
conv4	$8 \times 8$	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$
avg-pool	$1 \times 1$	$[8 \times 8]$

圖 5: WRN 架構

[7] 為了提升辨識性能，通過增加 output channel  $k$  的數量來使模型變得更 wider，從而  $N$  可以保持很小的值，網絡可以達到很好的效果，在模型最後添加性別及年齡分類層。

通過將年齡進行分組，比如 0-10, 11-20...，將年齡劃分到某一個區間，然後在這個區間進行分類回歸，[8] 將人的年齡劃分為 0-100 共 101 個類別，通過對年齡進行分類，並獲得每個預測年齡的可能性，再乘以各標籤年齡，最終得到年齡。

## 7. DS 測距法[3]

首先，設備 A 向設備 B 發送一訊號，並記錄下發送的時刻  $T_{as}$ ，設備 B 收到訊號後，記下收到的時刻  $T_{br}$ 。設備 B 等待一段設定好的  $T_{reply}$  時刻，在  $T_{bs}T_{bs}=T_{br}-T_{reply}$  時，向設備 A 發送一訊號，設備 A 收到訊號後記下時刻值  $T_{ar}$ 。此操作兩次通訊後，可以彌補因為時間不同步而造成的誤差，以下代數 1 為第一回通訊，2 為第二回通訊。即可算出電磁波的傳送時間，傳送時間乘以光速即為兩設備間的距離。

$$l = \frac{l - l}{l}$$

$$\begin{aligned} \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \\ \frac{1}{2} &= \frac{1}{2} - \frac{1}{2} \end{aligned}$$

使用 DS 測距法之誤差如下：

$$= \frac{1}{2} \times \left( 1 - \frac{1}{2} \right)$$

假設設備 A 和設備 B 的時間精準度是 20ppm (ppm=百萬分之一，極差)，則 ka 和 kb 分別是 0.99998 或者 1.00002，ka 和 kb 分別是設備 A、B 時間的實際頻率和預期頻率的比值。設備 A、B 相距 100m，電磁波的傳送時間是 333ns。則時鐘引入的誤差為 2033310-9 秒，因此誤差為 2.2mm，可忽略不計。因此雙向測距是最常採用的測距方式。

## 8. TOA(Time of Arrval)三點定位法[2]

到達時間定位法的幾何原理如下圖所示。使用 3 個天線做定位時，主動式標籤和天線  $i(i=1, 2, 3)$  的距離可定義為：

$$r_i = (x_i - x_m)^2 + (y_i - y_m)^2 \quad (1)$$

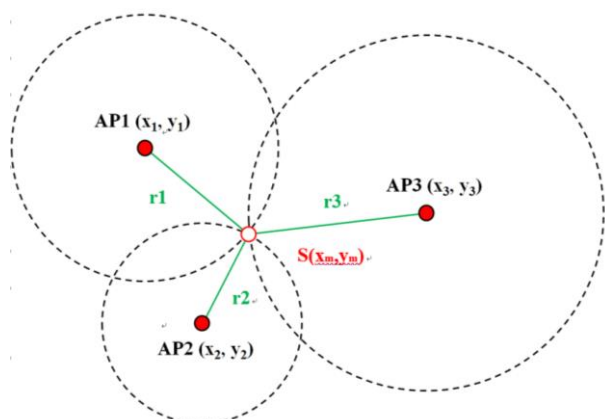


圖 6：3 點定位計算之示意圖

其中  $t_0$  為讀取器開始發射到標籤的時間常數， $t_i$  為天線訊號到達主動式標籤的時間， $c$  為光速。

藉由三個天線的距離半徑  $(r_1, r_2, r_3)$ ，可利用下列式子估算測量出主動式標籤的位置  $(x_m, y_m)$ ：

$$r_1^2 = (x_m - x_1)^2 + (y_m - y_1)^2 \quad (2)$$

$$r_2^2 = (x_m - x_2)^2 + (y_m - y_2)^2 \quad (3)$$

$$r_3^2 = (x_m - x_3)^2 + (y_m - y_3)^2 \quad (4)$$

假設  $x_1 < x_2 < x_3$ ，且  $x_1 = 0, y_1 = 0$  時，將式(3)減式(2)可得

$$r_2^2 - r_1^2 = x_2^2 - y_2^2 + y_1^2 - x_1^2 \quad (5)$$

同理，將式(4)減式(2)可得

$$r_3^2 - r_1^2 = x_3^2 - y_3^2 + y_1^2 - x_1^2 \quad (6)$$

整理式(7)和式(8)，且轉換成矩陣形式，可得

$$\begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_2^2 - y_2^2 + y_1^2 - x_1^2 \\ x_3^2 - y_3^2 + y_1^2 - x_1^2 \end{bmatrix} \quad (7)$$

其中

$$r_i^2 = x_i^2 + y_i^2 \quad (8)$$

將式(7)改寫成

$$\mathbf{H} \mathbf{x} = \mathbf{b} \quad (9)$$

其中  $\mathbf{H}$ 、 $\mathbf{x}$  與  $\mathbf{b}$  可表示為

$$\mathbf{H} = \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_m \\ y_m \end{bmatrix}, \mathbf{b} = \frac{1}{2} \begin{bmatrix} x_2^2 - y_2^2 + y_1^2 - x_1^2 \\ x_3^2 - y_3^2 + y_1^2 - x_1^2 \end{bmatrix} \quad (10)$$

可改寫成

$$\mathbf{H} \mathbf{x} = \mathbf{b} \quad (11)$$

TOA 定位法需要天線與主動式標籤兩者的時間有精確的同步，如此所量測到的才



能近似實際的估計距離。通常天線本身會有幾個微秒的偏移誤差值，這樣的偏移量會造成 TOA 定位法產生估計距離上的誤差。TOA 定位法可適用於各個距離的定位，但因為訊號的傳播速度非常快，所以 TOA 對於時間的敏感度相當高，必須十分精準測得訊號實際的傳播時間，即使是 1 微秒的時間誤差，也會造成 300 公尺的距離誤差，導致計算之結果與主動式標籤實際位置之間的誤差相當可觀。

故本計畫改良 TOA 及 DS 測距，以達到精準定位。雖然因為訊號強弱及干擾的問題，三個圓所相交出來的可能為一範圍，此範圍即為實際位置跟定位結果所產生的誤差，但因為 uwb 的特性，此範圍可以縮減到非常小，幾乎能夠控制在 30 公分以內。

## 9. 道路辨識技術

在實際的應用上，uwb 技術會因為障礙物或是基站的距離而出現很大的誤差，對於這種不適合用室內定位技術的情況，我們選擇用道路辨識加上 ds 測距來彌補。

我們使用消失點作為道路辨識的原理依據。一個理想的道路圖都會有一個消失點，如下圖：



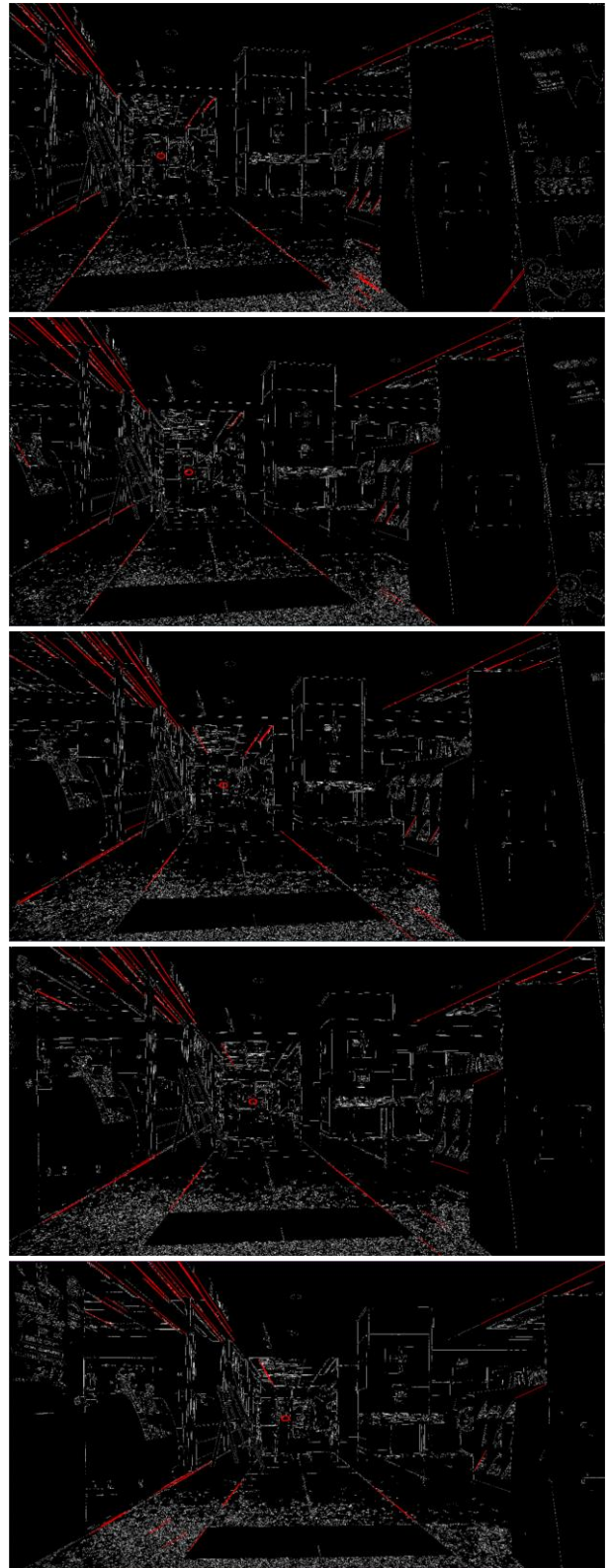
圖七：圖片中的消失點

這個消失點就代表著道路的方向，如果機器人的面向與道路的方向相同，消失點會在畫面的正中間，反之則會偏向畫面的某一邊，因此我們可以用消失點與畫面中線偏離的程度來調整機器人的轉向。

當機器人擷取到它正前方的畫面後，我們利用高斯模糊來去除圖片中的雜訊，再來用 Canny 邊緣偵測與霍夫轉換提取

畫面中可能的道路線條。取得線條後求兩兩交點，這些焦點都是可能的消失點。

為了從這些點中找出最正確的消失點，我們選擇與其他道路線條點線距和最小的點當作我們要的消失點。我們用下列圖片模擬機器人面向歪掉時如何用消失點校正面相：



## 4、主要成果與評估

精準定位上，透過uwb接收當前位置計算轉向和行走距離再將之傳送給機器人，反應時間控制在一秒內，並採用最短路徑演算法作計算，也就是說機器人走到我們要的指定位置，過程順暢且路徑簡潔。

在往後，此功能可與機器人圖書館書籍查找，商場商品架位查找，甚至博物館特定點自動語音導覽等等作結合，作到室內精準定位與資料庫合併，有效率的空間分配和資源統整，或對特定座標位置作到特定功能。

而本計畫目標環境在辦公大樓，走進辦公大樓機器人主動向你打招呼，有意願互動將問你需要什麼服務，天氣、空氣品質或是帶你去你未知的辦公處室，基本的導覽問答，之後也欲新增新功能，在辦公地點新增機器人上班打卡，各處室人員暫離或上線顯示，將上述功能作整合，希望達到機器人脫離居家或小場域環境，帶來新的商機。

## 5、結語與展望

現在GPS衛星定位相對現在的室內定位成熟及應用廣泛，我們欲結合室內定位和影像處理做到室內的移動及跨樓層擴及更多應用，以達到精簡人力及降低成本，並結合人臉辨識可達到客製化功能（如：打卡上下班、辨識信件包裹收件人），以減少日常生活中繁瑣的小事。

## 6、工作分配

黃覺修：Android 撰寫、道路辨識

賴柏宏：RESTful Server、A\*

張庭爾：精準定位

曾偉庭：性別識別、A\*

## 7、銘謝

國立台北大學資訊工程研究所分散及行動計算實驗室

## 8、參考文獻

- [1]<https://wiki.mbalib.com/zh-tw/UWB>(last accessed on March 2019)
- [2][http://designer.mech.yzu.edu.tw/article/esystem/article/compressedfile/\(2009-07-03\)%20室內定位技術簡介.aspx?ArchID=961](http://designer.mech.yzu.edu.tw/article/esystem/article/compressedfile/(2009-07-03)%20室內定位技術簡介.aspx?ArchID=961) (last accessed on March 2019)
- [3]<https://blog.csdn.net/xiongdauw/article/details/76637697> (last accessed on March 2019)
- [4]<https://ieeexplore-ieee-org.sw.library.ntpu.edu.tw:8443/document/4380918> (last accessed on March 2019)
- [5]<https://www.gamedev.net/articles/programming/artificial-intelligence/a-pathfinding-for-beginners-r2003/>(last accessed on March 2019)
- [6] Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; Jurišica, L.: Path planning with modified a star algorithm for a mobile robot. *Procedia Eng.* 96, 59 – 69 (2014)
- [7] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [8] R. Rothe, R. Timofte, and L. V. Gool. DEX: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015.
- [9]<https://zh.wikipedia.org/wiki/%E8%B6%85%E5%AF%AC%E9%A0%BB>(last accessed on March 2019)

[10]<https://zh.wikipedia.org/wiki/IBeacon>  
(last accessed on March 2019)