

基於智慧手機的室內 PDR 定位系統模組化快速開發平台

林恭煥 石佳雯 許雅婷 簡嘉瑩 廖峻鋒

逢甲大學資訊工程系

{m0103411, d9917112, d9971158, d9970555, cfliao}@fcu.edu.tw

摘要

航位推算 (Pedestrian Dead Reckoning, PDR) 是一種不需額外基礎設施的定位技術,利用少數感測器,在適當調校下即可達成不錯的精準度,因此近年來有許多以 PDR 技術為基礎的室內定位系統被提出。然而,在實務上,建構精準的 PDR 定位系統過程相當繁瑣,必須考量應用服務特色、使用環境及對象,不斷嘗試各式演算法並調整參數。有鑒於此,本研究從軟體工程的角度出發,提出一個基於智慧手機的室內 PDR 定位系統開發平台,將 PDR 室內定位系統中的不同演算法與參數模組化,使研究人員可在開發過程中藉由組裝、建構與測試各種演算法,快速建構適用於特定場合的定位機制。本論文同時也藉由一個實際案例,說明如何透過此平台逐步改善定位系統的精準度,經由案例討論,我們發現此一平台藉由彈性模組組合及提供預設模組實作成品的方式,可有效降低研究人員於開發過程的負擔。

關鍵詞 室內定位系統、手機定位、航位推算

1 前言

室內定位系統在智慧生活空間應用服務扮演不可或缺的角色,在空間中,智慧生活空間應用服務需要得知使用者當前位置,才能給予最適當的服務。以居家健康照護為例,獨自生活的老人大多有行動不便的問題,當得知老人走到危險易滑倒區域時,會發出輕柔語音提醒其留意步伐;或是老人半夜如廁時,貼心地延著路徑自動開啟照明設備等,都必須正確得知使用者位置。因此,位置資訊在情境感知計算 (Context-Aware Computing) 領域被列為第一級重要情境 (Primary Contexts) 資訊 [1]。

目前室外定位技術主要以 GPS 及 Wi-Fi 為主,此類系統準確度約 10 公尺左右,在旅遊、尋找地標位置等大範圍服務有不錯的應用成效,不過此一精準度不足以滿足室內定位應用需求,例如在室內透過 GPS 定位誤差往往過大,連使用者位於那個房間都難以正確判斷。一般住家及市區 Wi-Fi 基地台相當普及,因此也有相當多研究是透過分析 Wi-Fi 訊號來定位,然而,這種方法最大的問題是訊號源容易受到環境因素 (如室內裝潢材質、溫濕度、風速風向) 的影響,當導致訊號強度消滅、不穩定、反射或折射時,精準度就大受影響。此外,Wi-Fi

為主的室內定位需在附近有數個 Wi-Fi 基地台做為基礎設施,因此無法在特定場所使用,例如對無線基地台有所限制的醫院或地點偏遠的老人長照中心 [2]。最後,此類定位系統使用前必須進行訊號觀測記錄動作 (site survey),此記錄的品質對定位效果有重大影響。但一般未經訓練的使用者很難做出良好的訊號觀測記錄 [3]。透過類似原理,其它也有以無線感測網路的節點來進行室內定位 (如 Zigbee 等) 的機制 [4],這些設備除了建構額外基礎設施外,無線感測設備還需要定期更換電池,這種維護工作非常辛苦且耗費力。另外,也有利用紅外線陣列的定位方法 [5],使用者於身上配戴紅外線發送器,與事先安裝好的紅外線接收器一同運作。但其傳輸距離範圍極短 (少於 5 公尺),且訊號極易受環境影響。

近年來,智慧手機已深入人們的生活,成為隨身攜帶的配備,再加上電子技術的進展,各式感測器的成本降低,大部份智慧手機均內建三軸加速計 (Accelerometer) 及方向感應器 (Orientation sensor),使智慧手機成為理想的感測平台。因此,有許多研究人員嘗試結合手機,在室內使用航位推算 (Pedestrian Dead Reckoning, PDR) 技術進行定位。首先,利用手機上的三軸加速計及方向感應器計算出使用者的移動距離與步數,再利用方向感應器得知使用者當前移動的方向,再計算出移動的軌跡,並推算相對於出發點的座標。

目前在學術上有關 PDR 定位系統的研究很多,這些系統雖然採用了不同的感測器及演算法,在設計結構上卻都有十分類似之處 (如圖 1),我們將在下一節詳細說明此一結構。相較於其它室內定位機制, PDR 最大的優勢在於不需事先佈署基礎設施,也不需事先進行訊號觀測記錄工作,因此不易受環境影響,定位品質穩定,且成本低廉 [6, 7]。然而,近年來也有學者指出,大部份 PDR 定位機制研究為了學術貢獻,往往將機制設計得過於複雜,使得實作困難,且目前仍缺乏對此類定位機制實作挑戰的探討,造成實務上該類系統難以開發 [8]。在 PDR 定位系統開發期間,針對特定應用服務與特定環境嘗試不同演算法是很常見的,但是開發過程中,若發現效果不好要更換演算法,就必須重新改寫整個系統,因此在實務上開發 PDR 定位系統所付出的成本經常居高不下。造成實務上該類系統難以開發 [8]。在 PDR 定位系統開發期間,針對特定應用服務與特定環境嘗試不同演算法是很常見的,但是開發過程中,若發現效果不好要更換演算法,就必須重新改寫整個系統,因此在實務上開發 PDR 定位系統所付出的成本經常居高不下。

* 本論文接受教育部 102 年度資訊軟體人才培育推廣計畫與國科會研究計畫編號 102-2221-E-035-039 與 102-2627-E-002-001 經費補助。

針對此一觀察,本研究應用物件導向軟體設計技術,發展一套可使得研究人員快速開發低成本 PDR 室內定位機制的開發平台。此開發平台將 PDR 各資料處理階段設計為具有相同輸出介面的獨立模組,並提供預先實作的各式常用演算法模組(參考表 1),這些模組可以在平台支援下,藉由 pipeline 或 workflow 般的設計方式,以 Builder 設計樣式構成的簡單特定領域語法(Domain Specific Language, DSL) [9],自由組合獨立模組來完成低成本的室內定位系統,使開發人員可在定位機制設計階段,快速嘗試各種演算法模組的組合,為其特定應用找到最好的搭配方式。

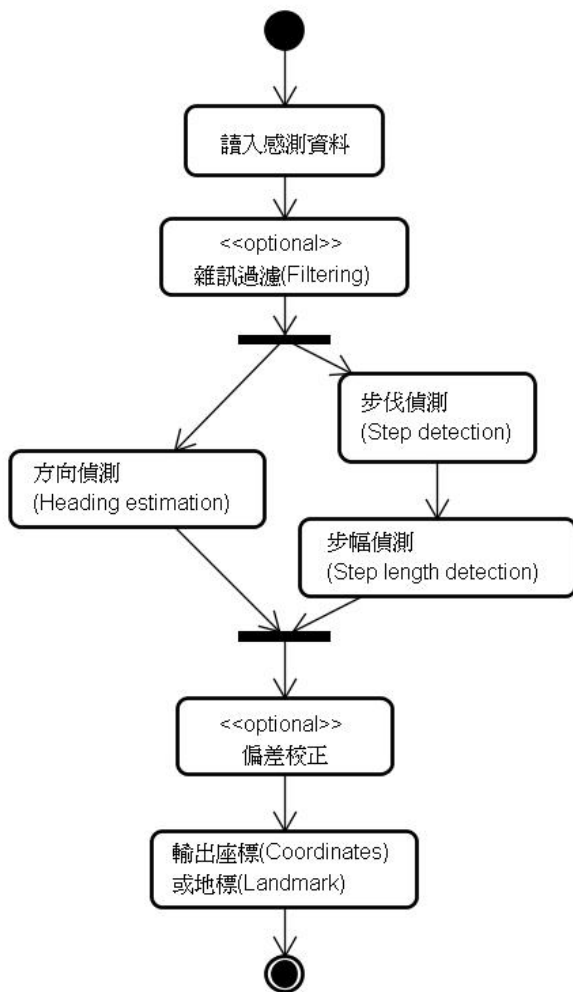


圖 1: PDR 室內定位系統資料處理步驟

2 研究背景

以智慧手機為基礎的 PDR 定位機制雖然採用的感測器、演算法與精準度各有不同,但在設計上都有十分類似之處。經過分析,PDR 定位系統共通的資料處理流程可表示如圖 1 的流程圖。首先,從感測器讀入資料,接下來為增進後續辨識結果的穩定性,通常會引入各式雜訊過濾機制來對資料進行前處理,以求將不正常的感測值去除。常被使用的方法包含 High pass 或 low pass filter [2] 及自動控制領域常用的 Kalman filter [10] 等。但並非所有 PDR 機制都具有此一步驟,例如 Kang 等人提出的機制就沒有進行雜訊過濾 [3]。

接下來必須分別判讀人員的行走距離和行走方向,行走距離通常結合步數偵測 (Step Detection) 及步幅偵測 (Step Length Detection)。步數偵測目前大部份的研究都基於加速計讀值的趨勢來辨識步伐,例如波動高峰偵測法 (Peak Detection) [11] 及波形急速下降偵測法 (Sudden drop detection) [2],週期偵測法 (Zero-crossing detection) [12],類神經網路 [13] 等。由於加速計一般有 x、y、z 三個讀值,因此也有學者認為應先整合為一個值再進行後續處理 (Synthetic acceleration) [3]。步幅偵測則包含了以加速度平均值為主的 Kim 法 [14]、以加速度的振幅為主的 Weinberg 法 [15] 及以加速度值最大值、最小值以及平均值間關聯的 Scarlet 法 [16]。此外,Shin 等人提出了根據步伐頻率以及速度推估出來與步履長度的線性方程式的推算步幅的方法 [7]。行走方向則和智慧手機平台有關,有些透過加速計讀值與磁力計加以計算 [3],有些則直接採用方位及手機姿態資訊。

最後,由於 PDR 定位機制具有誤差累積的問題 [2],因此可加入偏差校正措施,在適當時機將誤差縮小 [2, 10]。但如同雜訊過濾機制,並非所有 PDR 定位機制都具有偏差校正模組。在取得一段時間的行走方向與距離後,在平面上就形成一個有向性的線段,結合多個線段即可形成人員位移的軌跡。圖 1 顯示了 PDR 定位系統的主要資訊處理流程及各個主要模組。根據應用場合及所需解析度的不同,各個不同的 PDR 定位系統在特定模組會採用不同的演算法。事實上許多此類研究事實上只是針對各模組選取或提出適當的演算法,再將之組成一個完整的定位系統。表 1 整理了各重要模組常見的演算法。

表 1: PDR 室內定位系統主要步驟所使用的演算法

步驟	主要演算法
雜訊過濾 (Filtering)	High pass and low pass filter [2], Kalman filter [10]
步伐偵測 (Step detection)	Synthetic acceleration [3], Peak detection [11], Sudden drop detection [2], Zero-crossing detection[12], Neural network [13]
步幅偵測 (Step length detection)	Weiberg method [15], Scarlet method [16], Kim method [14], Linear combination [7]
方向偵測 (Heading estimation)	Direct reading, Kang method [3]
偏差校正 (Calibration)	GPS [10], Sequence alignment [2]

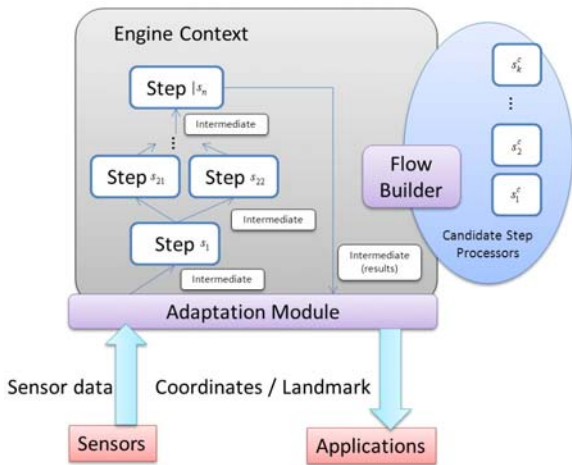


圖 2: 開發平台系統架構圖

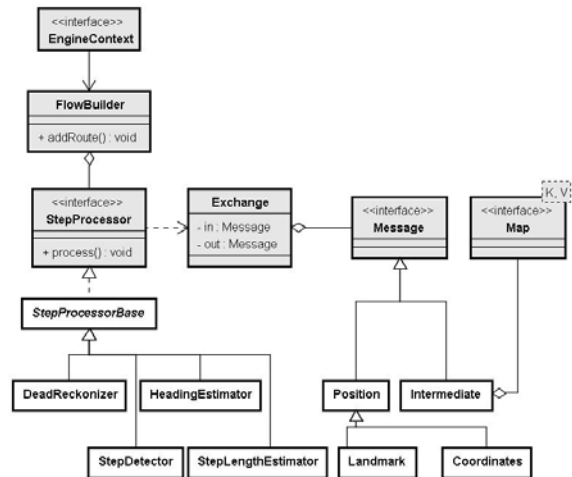


圖 3: 開發平台核心類別圖

3 平台系統架構與開發流程

本節將說明 PDR 室內定位系統快速模組化開發平台的設計理念。如同第 2 節所述,室內 PDR 定位系統雖然採用了不同的感測器及演算法模組,但資料處理流程十分類似。以下我們首先將從軟體設計的角度,分析此一開發平台的系統架構,接下來說明基於此一平台的開發流程。

3.1 平台系統架構

本開發平台的系統架構如圖 2 所示,定位系統從感測器中取得讀值後,透過轉換模組 (Adaptation module) 將讀值轉換為系統內部共通的格式 (稱為「中繼資料 (Intermediate)」)。考量中繼資料儲存結構的泛用性需求,我們將 Intermediate 定義為聚合結構 (Aggregate) [17]。若令 $p = (k, v)$ 為一個鍵-值組,則 $P = \{p_j\}$ 為鍵-值組集合 (或稱 Map),且 Intermediate $I = \{P_i\}$ 。根據平台核心類別圖 (圖 3),Intermediate 資料結構在實體上我們以 $\text{Set}\langle\text{Map}\langle\text{String},\text{Object}\rangle\rangle$ 來實作。

取得感測資訊並轉為 Intermediate 後,會經過多個處理步驟單元 (Step Processors),每個 Step Processor 封裝了特定步驟的演算法,例如雜訊過濾、步伐偵測、步幅偵測、方向偵測及偏差校正等步驟所包含的各個演算法都可視為一個 Step Processor 加以模組化 (參考圖 3)。將特定的 Step Processors 依照開發人員指定的次序結合後,會成為「資料流 (Flow)」。每一個由開發人員定義的 Flow 在本質上是一個擴充形式的有向圖 $F = (S, E, I)$,其中 F 代表資料流,也就是定位系統的資料處理流程本身, $S = (s_1, s_2, \dots, s_n)$ 是處理步驟單元所形成的有序集合,而 $E \subseteq S \times S \times I$ 是處理步驟單元的有向連結,且對所有的 $e \in E$ 都附屬一個特定格式的 Intermediate I_x 做為 s_y 的輸出或輸入介面。就直觀的意義來說,一個 Flow 就代表了一個特定的定位系統。在定位系統執行時,Step Processor 會將輸入的 Intermediate 處理後,轉為另一個 Intermediate 並加以輸出至下一步驟,到達最後步驟之後,再透過轉換模組將 Intermediate 轉換為應用程式 (Applications) 所需要定位資訊 (圖 3 中的 Position 類別),例如座標 (Coordinates) 或地標 (Landmark)。由於所有 Step Processors 都屬於同一個介面 (StepProcessor),且輸出都是 Intermediate,因此開發人員可以基於 Builder 設計樣式構成的簡單 DSL 語法 (如圖 4) 自由組合 Step Processors 定義新的定位系統。

```

EngineContext engine = new DefaultEngineContext();
engine.addFlow( new FlowBuilder(){
    public void configure() {
        from("direct:sensorGateway")
        .to(new SyntheticAccelerationDetector ())
        .parallel ().to(new WeibergStepLengthDetector()),
            new MixMagnetGyroHeadingEstimator())
        .pdr(new CoordinatePositionResovler())
        .out("direct:myApplication");
    }
});
engine.start();
    
```

圖 4: 基於本平台開發的定位系統程式碼片段

此外,為了加快開發人員的開發速度,平台也提供了預先實作好的 PDR 定位系統中各階段各式演算法(請參考表 1) 的 Step Processor 模組,這些預先提供的實作可預先在開發平台中登記成為候選處理步驟單元(Candidate Step Processors),如果 Candidate Step Processors 不敷使用,開發人員也可以透過繼承 StepProcessorBase 類別,自訂新的 Candidate Step Processors。

這種架構最具彈性的地方是各式 Step Processors 可以透過 Flow 的定義加以組裝,甚至可以表達平行處理的語意。原本開發 PDR 最耗時之處在於,在不同的資料處理階段(請參考圖 1),根據應用程式特色,不斷更換與測試各式演算法,而在此一開發平台上,只要透過 Flow Builder 重新定義,開發人員可以快速測試各種不同的組合。

3.2 開發流程

欲使用此一平台的開發人員,首先必須了解所要開發的定位系統要支援的應用服務需要的準確度,例如需要正確的座標、需要軌跡、只需要地標,還是只要知道有走幾步即可。事實上大部份智慧生活空間應用服務不需要太高的精準度,例如自動照明、關燈,只需正確辨識使用者在那個房間即可;而健康照護用的計步器,更只需透過步伐偵測模組得計算步數即可。了解應用服務所需定位精準度,可以降低所使用模組的數量,降低所需硬體成本,加快開發時間(要達成精準的定位必須重覆進行實驗與更換嘗試不同模組),更可以降低軟體除錯的困難度。此外,若應用服務只能依賴特定的硬體或作業系統版本,也必須事先列入考量,例如可能有某些智慧手機只不支援方向感測器、或是一些作業系統上輸出的加速計不夠精準等狀況。

當服務精準度及軟硬體確定後,接下來可使用開發平台依下列步驟進行開發:

表 2: 透過實測校正的方位感測讀值判斷表

讀值	判斷方位
337.5-360 或 0-22.5	北
22.5-67.5	東北
67.5-112.5	東
112.5-157.5	東南
157.5-202.5	南
202.5-247.5	西南
247.5-292.5	西
292.5-337.5	西北

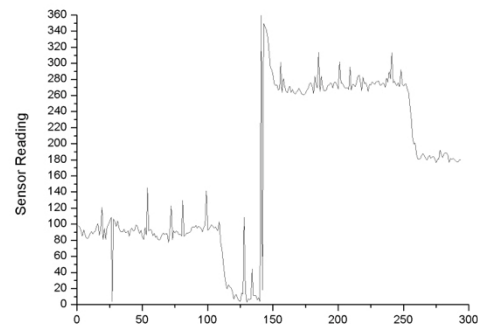


圖 5: 方向感測器實驗

- 1.初步考量與評估 PDR 的必要步驟,例如,若感測器訊號輸入品質良好時,雜訊過濾與偏差校正階段可以考慮先不實作。
- 2.針對每個 PDR 的必要步驟,初步選定「步伐偵測、步幅偵測、方向偵測」的演算法模組。如果所需要的演算法模組未提供,則必須自行透過寫作 StepProcessorBase 的子類別實作。
- 3.使用平台所提供 FlowBuilder 類別與 DSL,將選定的演算法模組依所需要的次序建構為一 Flow。
- 4.編譯程式並佈署至手機,在需要佈署的空間中進行實地步行測試。平台提供工具函式可將各階段產生之資訊寫入手機上的本機資料庫。
- 5.根據資料庫計算定位準確度,若準確度達到理想值則完成,否則回到步驟 2。

4 案例探討與實驗評估

為驗證平台設計,我們基於此平台實作了一個 PDR 定位系統初步雛型。以下將說明我們是如何基於此一平台與開發流程,逐步改善此一 PDR 定位系統雛型。本案例中,要開發的 PDR 定位系統將用於「影音如影隨形」

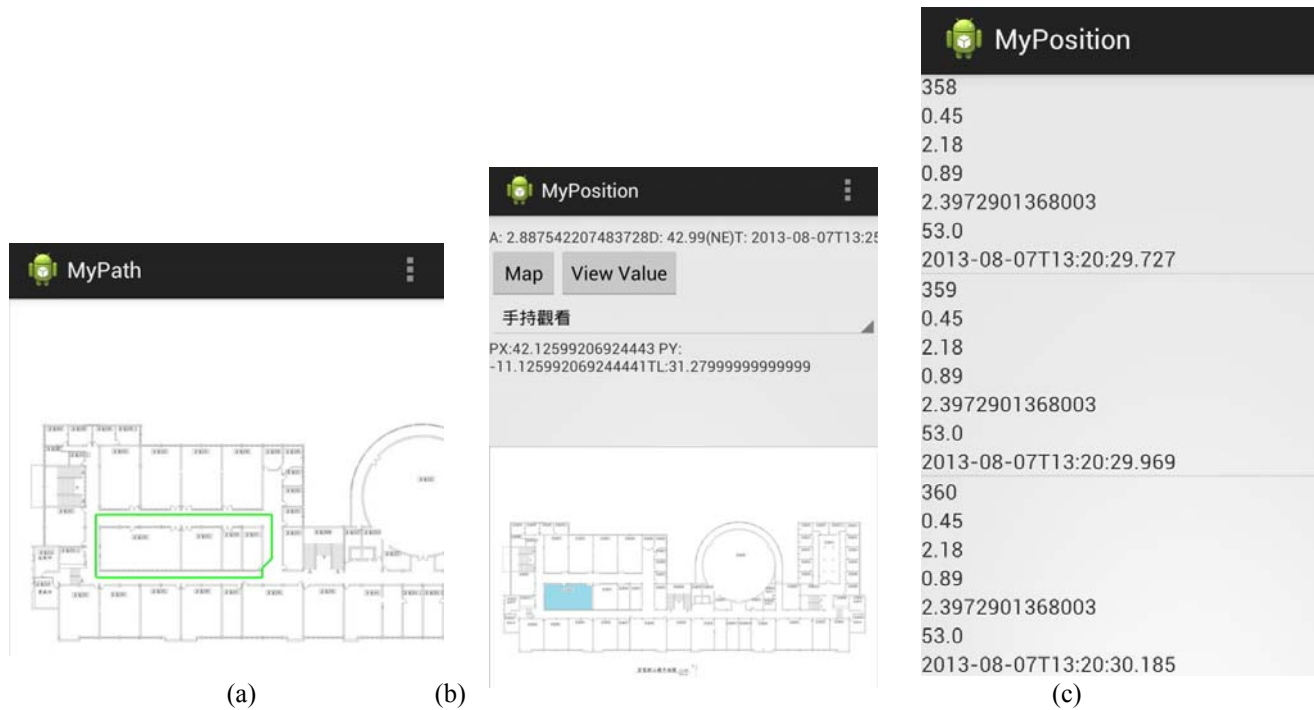


圖 6: 基於本平台開設之 PDR 定位系統雛型: (a) 軌跡模式; (b) 座標/地標並陳模式; (c) 步行資料庫模式

應用服務,使用者在一個房間聽音樂或看影片,當需要而移動到另外一個房間,也想聽到同一首音樂或同一部影片,而且接著之前中斷的部份再繼續播放。此一應用服務只需要地標 (Landmark) 層次的解析度,定位精準需求約在 3 公尺以內。我們所使用的開發硬體為 HTC One X 1.5Hz 四核心智慧型手機,記憶體為 1G,儲存空間 16G,使用 Android 4.0.3。此款手機上配備的感測器有陀螺儀、加速計、方向感測器、環境光線感測器等。

初步評估,由於此應用服務需求的精準度不高,因此我們選擇只實作步伐偵測、步幅偵測及方向偵測模組。步伐偵測的演算法我們選用同時考慮三軸向加速度的 Synthetic acceleration [3]。此方法主要的概念是先從手機內的三軸加速計取得時刻 t 的 x 、 y 、 z 三軸的加速度值,再利用

$$a(t) = \sqrt{a_x^2 t + a_y^2 t + a_z^2 t - g}$$

計算一段時間內加速度的變化,利用 $a(t)$ 的規律波動週期計算步數。

步幅偵測部份則使用 Weiberg method [15]。其主要概念是取得 $a(t)$ 波動週期的最大最小值,配合估算參數 η 來推算步伐長度。第 k 步的步幅可以下列公式計算:

$$l_k = \eta \cdot \sqrt[4]{(a_{max}(t) - a_{min}(t))}$$

其中 $\eta = d_{real}/d_{estimated}$ 是一個需要預先設定好的常數。最後,在方向偵測方面,我們直接藉由實際測試結果,我們直接依據表 2 將特定讀值對應到特定方向,做為方向偵測模組的輸入值。在上述模組調整完成後,我們透過如圖 4 中的程式碼將各模組整合。表 3 顯示了第一個版本的定位系統的精準度實驗結果。實驗場地是在逢甲大

學資訊電機大樓,進行方式則為選定 4 條方向不同的相連路線,其中路線 1、3 為 50 公尺,路線 2、4 為 16 公尺。實驗時持手機步行,於步行時透過手機即時定位,每次判定時間間隔為 0.2 秒 ($t=0.2s$),並將定位結果即時顯示於畫面(圖 6a、b),定位結果及收集到的資料則儲存於手機上的 SQLite 資料庫內

(圖 6c)。實驗完成後,離線進行準確度分析,實驗共重覆進行 5 次,實驗的結果如表 3 所示,開發畫面則如圖 6 所示。

由表 3 可發現,雖然平均誤差在 1 公尺內,但路線 2 的誤差達到 50 公尺,即使是用來做房間定位,也屬於無法接受的精確度。其主要原因是在轉彎時,方向偵測一旦有誤差,之後做航位推算時,此誤差會不斷累積。因此,我們透過平台提供的組裝機制,將其它模組拿掉,單單只對方向感測器做測試,其結果如圖 5 所示。我們發現,硬體的的方向感測器讀值並不穩定,如果轉彎時遇到不正常的讀值,就會產生很大的誤差。

基於此一發現,在第二個版本的定位系統中,我們改用 Kang 等人提出來的的方法 [3] 來改善方向偵測讀取穩定度的問題。Kang method 主要的概念是額外參考加速計讀值來輔助校正方向感測器的讀值。接下來透過平台提供的 DSL,可以很容易地將原來的模組替換為 KangOrientationDetector 類別來完成實作。改用此方法後重新進行實驗的結果如表 4 所示。可發現由於轉彎時方向偵測誤差縮小,因此整體準確率也大幅提高,因此也達成了應用服務所要求的精準度。

本節透過展現一個基於我們所提出的開發平台,實際為一個應用服務開發 PDR 室內定位系統的過程。在依照 3.2 節提出的流程的開發過程中,可發現開發人員需要藉由不斷調整參數及更換測試不同模組來完成一

個夠準確的 PDR 室內定位系統,而我們所提出的平台,藉由彈性的模組組合及提供數個預設模組實作成品的方式,可以大幅降低開發人員於此一持續改善精準度過程的負擔。

表 3: 初步實驗結果

序號	路線1	路線2	路線3	路線4	平均
	累計	累計	累計	累計	量測
	誤差	誤差	誤差	誤差	誤差
	(m)	(m)	(m)	(m)	(m/次)
1	0	8	51.4	9.6	0.522
2	0	0	50	0	0.379
3	0	6	50	0	0.424
4	0	0	50	0	0.379
5	0	6	50	2	0.515

表 4: 改用 Kang method 做方向偵測後的實驗結果

序號	路線1	路線2	路線3	路線4	平均
	累計	累計	累計	累計	量測
	誤差	誤差	誤差	誤差	誤差
	(m)	(m)	(m)	(m)	(m/次)
1	0	0	0	1	0.007
2	0	1.4	2.8	2	0.047
3	7	0	2.36	0	0.070
4	0	4	0	2	0.045
5	0	1.4	0	0	0.010

5 結論

本論文的主要從軟體工程的角度出發,發展一個可快速建構 PDR 室內定位系統的開發平台。此平台將 PDR 各資料處理階段設計為具有相同輸出介面的獨立模組。開發人員可自由組合獨立模組,快速嘗試各種演算法模組的組合,為其特定應用找到最好的搭配方式。我們也基於此一平台實際開發出一個定位系統,評估其精準度,並運用在一個典型的智慧生活空間應用服務上,以驗證此一平台的可行性。PDR 定位技術先天具有誤差累積問題,一旦開始發生偏差,則往後的移動會產生更大的誤差。未來針對這一點,我們正在研究利用 WiFi 訊號或是二維條碼來進行位置偏差的修正。其次是手機在使用者身上的配戴位置不同,會對定位產生影響,例如手機可能會拿在手上、放在口袋或是放在包包裡,即使同樣是行走,也會因為擺放位置不同而會產生不同的加速計波形,方向判斷也會因此而難以判斷,未來我們也將透過手機內建的姿態感應器,自動修正因手機放置位置所產生的誤差,開發不需特定使用姿態的 PDR 定位機制。

參考文獻

[1] A.K.Dey, "Understanding and using context," *Personal and Ubiquitous Computing Journal*, Vol.1, No. 5, 2001.

[2] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "FootPath: Accurate Map-based Indoor Navigation Using Smartphones," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[3] W. Kang, S. Nam, Y. Han, and S. Lee, "Improved Heading Estimation for Smartphone-Based Indoor Positioning Systems," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2012.

[4] M. Popa, J. Ansari, J. Riihijarvi, P. Mahonen, "Combining Cricket System and Inertial Navigation for Indoor Human Tracking," in *Proc. WCNC*, 2008.

[5] C. Lee, Y. Chang, G. Park, J. Ryu, S. G. Jeong, S. Park, J. W. Park, H. C. Lees, K. S. Hong, M. H. Lee, "Indoor positioning system based on incident angles of infrared emitters," in *Proc. Annual Conference of the IEEE Industrial Electronics Society*, 2004.

[6] A. R. Pratama, Widyawan, R. Hidayat, "Smartphone-based Pedestrian Dead Reckoning as an Indoor Positioning System," in *Proc. International Conference on System Engineering and Technology*, 2012.

[7] S.H. Shin, M. S. Lee, C. G. Park, H. S. Hyun, "Pedestrian dead reckoning system with phone location awareness algorithm," in *Proc. IEEE/ION Position Location and Navigation Symposium*, 2010.

[8] C. Fischer, P. T. Sukumar, and M. Hazas, "Tutorial: Implementing a Pedestrian Tracker, Using Inertial Sensors," in *IEEE Pervasive Computing*, Vol. 12, No.2, 2013.

[9] M. Fowler, *Domain Specific Languages*, AW, 2010. [10] R. Jirawimut, P. Ptasinski, V. Garaj, F. Cecelja, W. Balachandran, "A Method for Dead Reckoning Parameter Correction in Pedestrian Navigation System," in *Proc. IEEE Instrumentation and Measurement Technology Conference*, 2001.

[11] Z. Sun, X. Mao, W. Tian, X. Zhang, "Activity Classification and Dead Reckoning for Pedestrian Navigation with Wearable Sensors," in *Measurement Science and Technology*, Vol.20, No.1, 2009.

[12] J. Kappi, J. Syrjarinne, and J. Saarinen. "MEMSIMU based pedestrian navigator for handheld devices," in *Proc. International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2001.

[13] K. Aminian, P. Robert, E. Jequier, Y. Schutz, "Level, Downhill and Uphill Walking Identification using Neural Networks," in *Electronics Letters*, Vol.29, No.17, 1993.

[14] J. W. Kim, H. J. Jang, D-H. Hwang, and C. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System," in *Journal of Global Positioning Systems*, Vol.1, No. 8, 2004.

[15] H. Weinberg, "Using the ADXL202 in Pedometer and Personal Navigation Applications," in *Analog Devices AN-602 Application Note*, 2002.

[16] J. Scarlet, "Enhancing the Performance of Pedometers Using a Single Accelerometer," *Analog Devices AN-900 Application Note*, 2005.

[17] E. Evan, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison Wesley, 2003.