

人本計算專題:核災應變機器人

Human Centric Computing: Nuclear Disaster Strain Robot

李聖誠、張博凱
Sheng-Cheng Lee, Po-Kai Chang

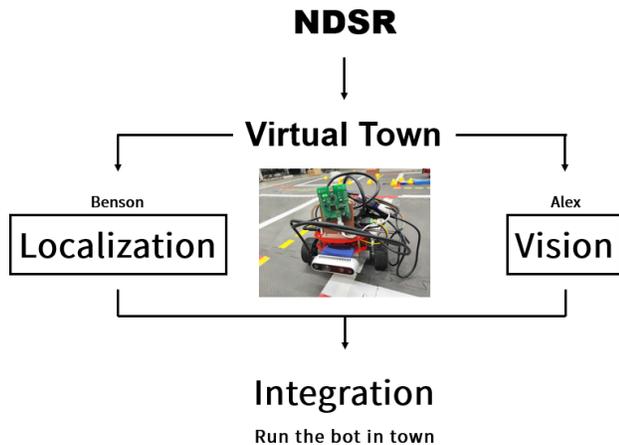


Fig. 1: 專題架構與主要研究貢獻

Abstract—目前，全球機器人市場持續增長，全球機器人產業市場規模逾270億美元，但開發目標卻是讓機器人執行人類原本就能完成的事，如說話、跳舞、拉小提琴，甚至主持婚禮，而人類做不到的事。根據世界核能協會2012年8月的數據，全世界31個國家有435座工作反應爐。在2011福島核泄漏時，美國「國防部高級研究計劃局」(Defense Advanced Research Projects Agency, 簡稱DARPA)的研究人員也參與了救援計劃，他們深感機器人在救災方面的局限。本團隊此次主題將針對，核災現場以及人類無法到達之地區進行搜救、探勘，運用AI影像辨識技術及定位系統，結合自走車相關技術，進入高輻射區域建置現場地圖並標示出相關物體(生還者、燃料棒)之位置。

潛在使用者為世界上任一擁有核能發電廠之國家。

關鍵詞：無人載具、影像辨識、UWB定位、人本專題。

I. 前言

隨著智慧型機器人與人工智慧的蓬勃發展，近年來各國均投入於無人載具之開發與研究，已經在非常多場域漸漸出現。在有結構性的環境(structured environments)下，無人駕駛車在從室外之高速公路、郊外人車較少之公路，到室內工廠物流運輸機器人系統已經有相當成熟的發展，其原因在於其所產生的應用是非常貼切人類的日常生活，並且能創造更大之經濟效應，例如：交通工具、商業物流等。但在意外發時，例如：輻射外洩。等人類無法進入之嚴峻場地，機器人更能發揮其不可取代性，代替人類進入災區進行移動、導航、定位、救援。

國防科技一直以來是各個國家的發展重點。美國國防高等研究計劃署(Defense Advanced Research Projects Agency, DARPA)以及美國太空總署(NASA)針對這樣的挑戰，舉辦相關競賽，如：DARPA Subterranean Challenge(簡稱SubT)，這場競賽旨在開發能夠「探索快

速繪製、導航、搜索和開發複雜地下環境的新方法」，包括人造隧道系統、城市地下和自然洞穴網絡等。大賽要求參賽者研製出能幫助人類在地下導航、繪圖以及搜尋的系統，所有救援系統在地下結構發生塌方或其他災難時，能夠在對人類來說有危險的地方移動和導航，幫助救援。

而本次專題就是根據核災發生現場，或任何人員無法進入之場地、城市，由應變機器人進入災區偵測目標物，並且能夠在無GPS之場域進行定位，準確得知無人車輛目前位置與目標物之相對位置，讓搜救人員迅速進行偵查、救援。

綜合上述，此搜救機器人需具備立體視覺整合系統感知偵測目標物並與UWB定位結合，需考量目標物大小、定位準度、目標物深度資訊等等計算出相對位置，因此，本報告的目標與貢獻如下：

- 1) UWB定位及MobileNet-SSD目標辨識
- 2) 合併且整合系統
- 3) 將結果可視化並顯示於Rviz

II. 核災應變機器人設計

為了模擬發生核災的城市，我們利用MIT Duckietown的巧拼建置縮小版核災城市，以scale down之環境呈現核災發生後人類無法進入之城市。城市大小訂為5*5之巧拼，每片巧拼60cm，城市總大小300cm*300cm。在這樣的城市中，我們的機器人要能夠辨識受害者及周遭環境，並且定位出自己所在的位置，才能夠發揮核災應變的功效。所以本次的專題，我們主要心力就放在視覺(Vision)以及定位(Localization)這兩項。

A. Localization

我們採用Ultra-wideband作為取得定位資訊的裝置，使用pypozyx函式庫，利用UWB Anchor的訊號強度與TWR(Two Way Ranging)定位技術(圖4) [1]同時利用三個以上的Anchor裝置就能定位出2D座標，同時利用四個以上能定位3D座標。同時UWB裝置上也搭載IMU感測器，能夠知道UWB裝置的方向。結合位置與方向這兩樣資訊我們能夠得出固定在車上的TAG三維姿態(Pose)。

B. Vision

在核災中，我們必須辨識出該城市內之物品，找出可能的目標物品以及生還者。由於此次人本專題我們使用了Duckietown城市模擬核災發生並作為驗證場地，所以重新組裝了一台Super Duckiebot。在視覺部份因考量到車體大小，故選擇使用體積較小的Jetson Nano處理相關視覺運算，同時也使用運算量較小的MobileNet-SSD Model，在影像辨識部份利用RGB資訊進行Training，精準度、輕巧度及靈敏度是Artifact search相當重要的指

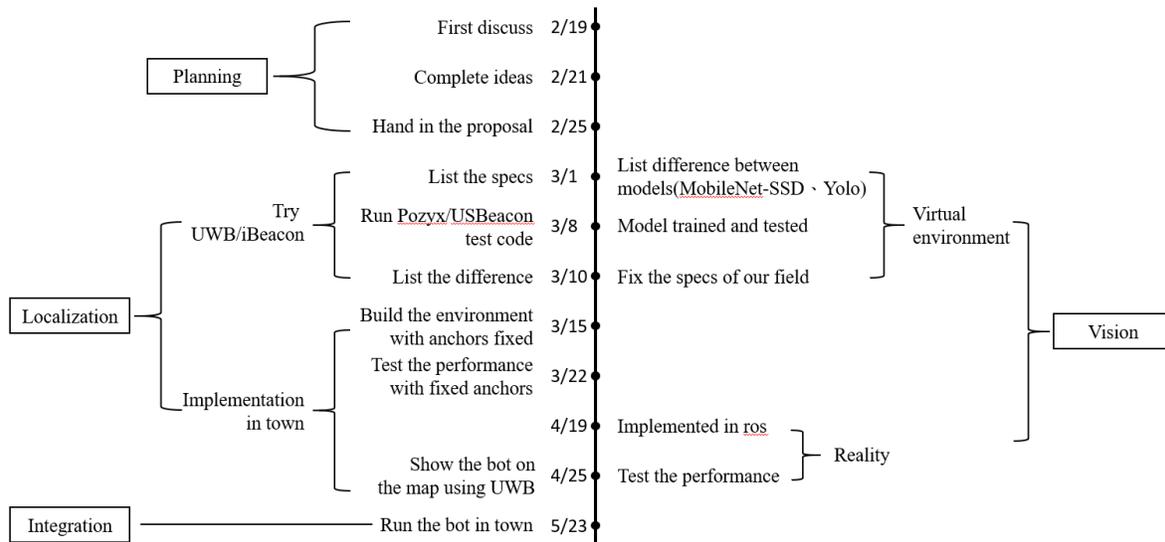


Fig. 2: 專題進度分工圖,博凱-Vision、聖誠-Localization

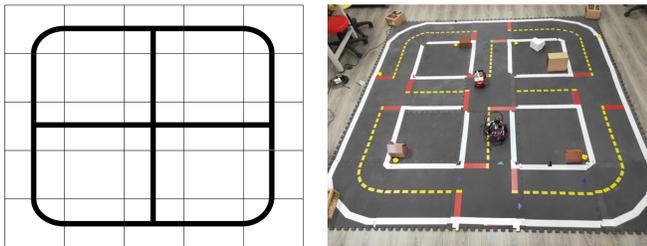


Fig. 3: 模擬城市場地設計

標，控制訓練後Model的大小將其放置於較輕巧的Jetson Nano上運行，並且考量到幀數（FPS）也就是一秒讀取幾張圖片進行辨識，各個環節環環相扣。

當辨識出目標物後，隨即使用Depth資訊利用投影矩陣取得目標物與相機之相對座標，結合UWB定位所取得之姿態訊息，定位出目標物之絕對位置，最後呈現於Rviz。

III. 系統設計

A. 機器人硬體設計

機器人的車體部分，我們沿用MIT Duckietown之車架壓克力板，但是為了更好的載重量我們將馬達改成具有編碼器之馬達，除了增加穩定性與馬力以外，未來也可加入Wheel Odometry之功能提供更完整的SLAM。主控板部分我們使用Nvidia發行之Jetson Nano，具有強大的圖形運算能力，能為深度學習的Model提供更可靠的運算處理。為了能夠定位以及辨識物體，我們在車上加上Pozyx公司推出的UWB裝置，以及Intel RealSense的D435深度相機。前者能夠提供3D Pose定位後者能夠提供RGBD之深度影像有利於視覺辨識與計算物體相對位置。

B. 機器人作業系統

在機器人上運行的系統，我們安裝Ubuntu18.04與ROS Melodic。ROS(Robot Operating System)是專為機器人軟體開發所設計出來的一套電腦作業系統架構。它是一

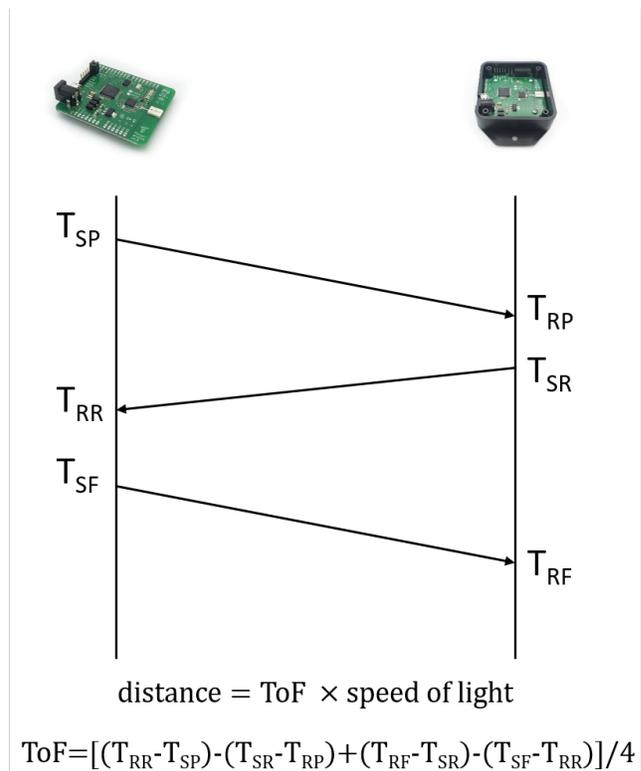


Fig. 4: UWB Localization TWR technique

個開源的元級作業系統，提供類似於作業系統的服務，包括硬體抽象描述、底層驅動程序管理、共用功能的執行、程序間消息傳遞、程序發行包管理，它也提供一些工具和庫用於獲取、建立、編寫和執行多機融合的程序。利用ROS我們能夠快速建立各種Node並利用Publisher與Subscriber互相傳送資訊。

C. 機器人定位與移動

我們用pypozyx的函式庫取得Tag Pose並且Publish至ROS topic，另一個Node將每個Pose記錄下來後，可在地圖上畫出一個行進軌跡，方便搜尋到物



Fig. 5: 車體設計及相關環境

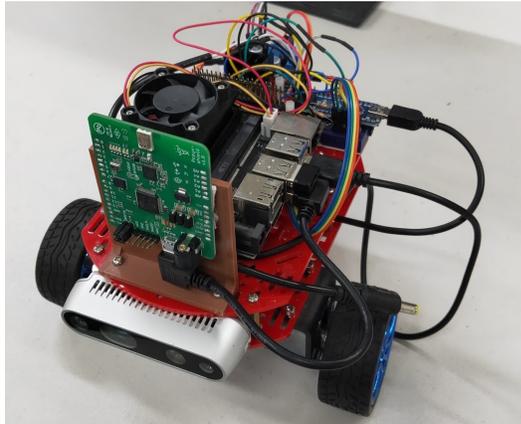


Fig. 6: 核災應變機器人之硬體設計

品之後進行後續的動作。機器人的移動是利用ROS的Joy Package搭配ROS Serial將car command傳送至Arduino驅動馬達，達到利用Joystick控制車子行進之功能。

D. 機器人視覺與物品定位

抽取D435之RGB影像後，首先使用MobileNet-SSD進行物品辨識，我們將搜索到的物品在畫面中標出Bounding Box之後，利用Depth資料以及CameraInfo將影像投影至三維空間，標出物品相對位置以後，利用TF函式乘上Pozyx的Pose資訊，

$$T_{obj} = T_{veh} \cdot T_{cam}$$

Fig. 7: 物品座標轉換

IV. 實驗與實體驗證

A. 機器人定位

此專題以室內高精度定位為目標，配合深度相機之影像辨識與物體深度，找到目標物之位置並標記。

a) 定位準確度測試: 本專題之定位裝置使用Ultra-wideband超寬頻通訊天線，為提供目標物位置之量測精準度，設計此定位精準度測試。鋪設5*5巧拼，量測各點之UWB定位座標，比較Ground Truth，進行多次量測後計算方均根值作為誤差數據，實驗結果如表 I所示。將結果之數據量化後標示於地圖上如圖8所示。

b) 定位測試影片: 本專題利用UWB之定位結果經過移動限制過濾器後繪製出行進軌跡，測試影片連結如下：

- 載具利用遙控行進場地一圈並繪製出軌跡圖
https://youtu.be/uJ_-_KYRU44

Ground Truth X (m)	Ground Truth Y (mm)	定位誤差 (mm)
300	300	68.43
1500	300	117.58
2700	300	64.69
2700	1500	195.94
2700	2700	220.11
1500	2700	200.33
300	2700	56.93
300	1500	27.56
1500	1500	50.39

TABLE I: 定位測試

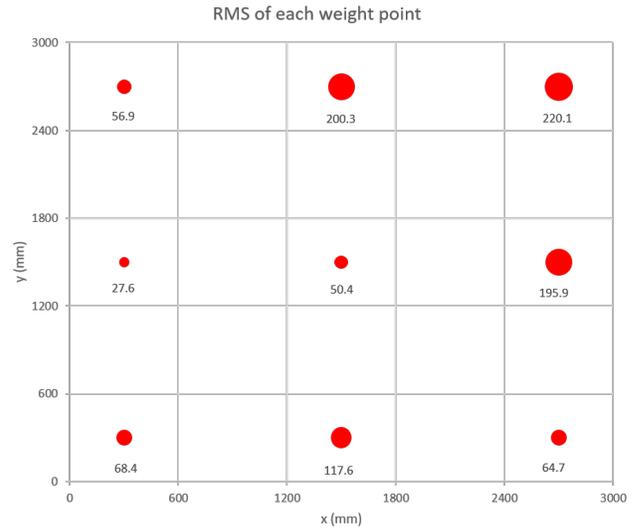


Fig. 8: 以UWB定位之誤差量測實驗結果

B. 影像辨識

本專題比較目標物於各版本SSD運算量及FPS後，如II，選擇FPS較高且運算量較低的MobileNet-V1 SSD，使用Jetson Nano執行Model。並配合深度相機於標記目標物後，將bounding box之中點位置利用矩陣投影的方式找出目標物與相機之距離，如10。

由如9可以發現到Duck的Average Precision會比其他的還要來的低只有0.58，明顯的看出MobileNet-SSD對於小物品的偵測相對於其他model如yolo較不理想。

Frame Rate (FPS)	desktop 1080	Jetson Nano
SSD	36	1.5
MobileNet-V1 SSD	90-100	16.5
MobileNet-V2 SSD	78-80	12

TABLE II: 各版本FPS比較

V. 結論

本次專題，使用Duckietown之虛擬城市環境，設計小型探測車裝置，並開發影像辨識與定位之功能，從而瞭解系統運作原理以及學習使用定位與影像辨識之技術。本學期已達成物件辨識與定位之功能，未來希望能加入Lidar與編碼器馬達之SLAM技術，進行更準確之定位與搜救平台研究開發。

REFERENCES

- [1] M. Kolakowski and V. Djaja-Josko, "Tdoa-twr based positioning algorithm for uwb localization system," in 2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON), May 2016, pp. 1-4.

```

alex@alex-K501UX: ~/pytorch-ssd-mobile
alex@alex-K501UX: ~/pytorch-ssd-mobile 100x30
process image 302
Load Image: 0.003154 seconds.
Inference time: 0.0755615234375
Prediction: 0.102396 seconds.
process image 303
Load Image: 0.004129 seconds.
Inference time: 0.07557326594787598
Prediction: 0.094034 seconds.
process image 304
Load Image: 0.003166 seconds.
Inference time: 0.07658171653747559
Prediction: 0.095763 seconds.
process image 305
Load Image: 0.003159 seconds.
Inference time: 0.07688236236572266
Prediction: 0.088536 seconds.
process image 306
Load Image: 0.004450 seconds.
Inference time: 0.07576532409694824
Prediction: 0.092369 seconds.

Average Precision Per class:
duckie car: 0.9110267642688139
house: 0.8664368693772458
broken: 0.7358197358197358
duck: 0.501639916939417

Average Precision Across All Classes:0.7737306215996032

```

Fig. 9: MobileNet-V1 SSD evaluation

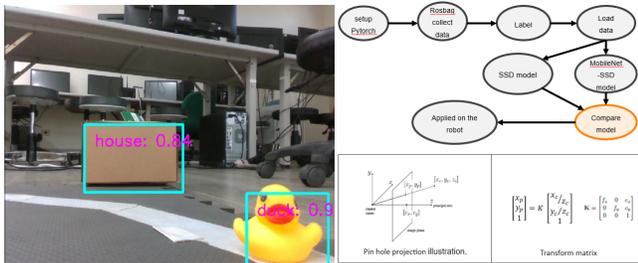


Fig. 10: 預測目標物之實驗步驟及相對距離投影矩陣

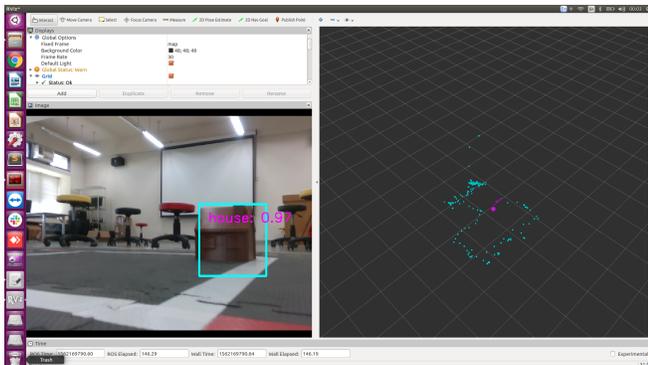


Fig. 11: 專題成果呈現