

運用雲端分散式容錯即時運算技術 Storm 處理
連續性複雜前景之移動性人物檢測與行為
分析之監視系統

執行成果報告

執行系所：電機系

計畫主持人：吳明芳

聯絡電話：0912994102

E-mail：wumifa@mail.ksu.edu.tw

計畫名稱：運用雲端分散式容錯即時運算技術 Storm 處理
連續性複雜前景之移動性人物檢測與行為分
析之監視系統

計畫編號：102-N-270-NSC-S-068

執行期間：102 年 11 月 1 日~103 年 10 月 31 日

中華民國 103 年 10 月 23 日

(一) 計畫之背景及目的

本產學合作計畫首先瞭解到合作廠商柏力揚興業有限公司主要經營及開發整合智慧型監視系統、視訊會議系統、多媒體影音視訊及視聽教學系統等設備。雖然該公司在智慧型監視系統、多媒體及視聽等設備的開發整合方面，其市場經驗與上、下游的整合已有相當豐富的成果，目前亦積極著眼於雲端分散式容錯即時運算技術Storm處理連續性複雜前景之移動性人物檢測與行為分析之監視系統設備的技術開發與整合，但柏力揚興業有限公司在雲端分散式容錯即時運算的技術研發能力較為不足，若能與學界進行產學合作，積極研發及整合智慧雲端監視設備與系統，則將可實現產、學雙方長短產生互補之優勢，並能提高智慧雲端監控產品的競爭力。

網路通訊是目前國家重點科技項目，其各種不同的有線及無線寬頻通訊基礎建設的普及，可為人們帶來生活上及工作上的便利，而人們各式各樣的需求更使得其應用會一直不斷的推陳出新。此外，數位生活也是國家目前的主要政策，透過數位化技術，配合各種數位設備，以有線及無線的網路串連家庭、辦公室或公共空間等，可提供各項內容、功能及服務，使人們在現代數位化中，能安全便利的享受生活、娛樂、休閒、學習、工作等，而安全監控的應用更成為近世代所需要發展的科技。因此，本計畫將結合合作廠商柏力揚興業有限公司來開發運用雲端分散式容錯即時運算技術-Storm處理複雜前景之移動性人物檢測與行為分析之監視系統，由於雙方的合作，使得學校既有的充份理論知識以及核心技術研究能力，可輕易加強廠商實務應用與整合方面之優勢。若再搭配業界廠商的影音監控硬體與系統平台的實務應用經驗，以及學界的跨網域影音傳輸及調變的技術，則產與學兩方面的結合，可互補兩方面的缺點與需求，進而達到雙贏的局面。本產學合作計畫所研發出來的成果，對廠商而言，可省去研發的時間與成本，使得廠商可直接與本計畫所開發的視訊會議進行技術轉移。

(二) 執行優勢

雖然本產學合作計畫之合作廠商柏力揚興業有限公司於目前視聽教學、多媒體影音視訊及智慧型監視等系統的整合方面，在市場經驗與上、下游的整合已有相當豐富的成果，且亦擅長於數位多媒體網路、自動控制以及網路監視、監控等系統的設計及開發，但在數位影像及後端伺服器產品的技術研發部份較為不足，因此，若進一步透過參與開發之學校老師與研究生的助力，將雲端運算、影像分析的技術等結合起來，可達到運用現有之廠商的硬體，產生更多的加值服務，也因此使研發成本、人力支援與合作廠商之原有技術與設備的配合，形成執行本計畫

的最大效益。為實現本計畫運用雲端分散式容錯即時運算技術-Storm處理複雜前景之移動性人物檢測與行為分析之監視系統，軟體系統開發上，如上述雲端運算之系統架構、人臉循序偵測及追蹤分析技術以及事件產生機制等，可由學校來負責，而在硬體平台與系統整合方面，可利用廠商的網路監視、監控等系統設備，以實現基於雲端分散式容錯即時運算技術-Storm處理複雜前景之移動性人物檢測與行為分析之監視系統。在本計畫執行之前，所有的硬體開發環境都已備妥，且人力分配相當清楚明瞭為執行本計畫的最大優勢。

(三) 研究方法

1. 研究方法與原因

1.1 需求及情境

從人類發明第一台電腦以來，到七〇年代Internet出現，再到現在雲端計算，整個過程就像是從量變到質變的飛躍。維基百科上對於雲端計算的定義是：透過網路，把多個成本相對較低的計算實體，整合成一個具有強大計算能力的完美系統，並借助SaaS、PaaS、IaaS、MSP等先進的商業模式，把強大的計算能力分佈到終端用戶手中。

雲端計算的一個核心理念就是，透過不斷提高「雲」的處理能力，進而減少用戶終端的處理負擔，使用戶終端簡化成一個單純的輸入輸出設備，並能按需求，享受「雲」的強大計算處理能力。

現在Internet上已經有很多應用具備雲端計算的特徵，如google的地圖搜尋。雲端計算的特性，還可以再用一個淺白的例子來比喻：假如你每年買一本最新的百科全書，仍然有可能查不到最新流行的名詞、事件，但如果上網進入維基百科，幾個小時前剛發生的熱門事件，在上面就可以查到詳盡的資料了。從這個角度看，雲的服務是融合，而且無所不包的。

至於安全監控產業，雲端計算會把整個行業真正提升到智慧安防的時代。然而現今大多路口都有加裝監視設備，若能加以利用這些設備，不光只是以往只有錄影的功能，可以在發生事故時，即時的發出警報、通知救援，就可以使這樣設備發揮出更大的功效，並且主要應用在行人方面，因為人與汽、機車在路上來說相對的比較危險，尤其是當有沒公德心的人將車子停在行人可走的範圍上，迫使行人往往需要走到機車道、甚至跨越到快車道，此時行人的生命安全將受到很大的威脅，然而當發生事故時，白天或許有目擊證人，當到了夜晚若是發生意外當下又沒目擊證人時，便可以藉由此系統對於此狀況加以判斷並且通報相關單位來處理。

然而，在雲端計算的模式下，前端設備（包含DVR、IP Camera、Video Server等）只是負責影像訊號蒐集、壓縮與傳輸，至於建立樣本資料庫，以及比對搜尋，都是在雲端完成。

當大量設備接入雲端，樣本資料庫的蒐集與建立自然不是問題，過去辨識演算法很耗運算

量，而雲端只需要增加運算節點，一樣可以解決問題。因此，就好像google將以前需要去圖書館申請才能做的資訊檢索，變得免費且實用。

然而，在以目前最熱門的雲端運算技術Map/Reduce(Hadoop)[24]來說，Hadoop是比較適合使用在分散式批次處理計算，強調批次處理，常用於資料採擷、分析，對於即時運算的能力相當不足，因此運用在即時的影像運算處理上，沒有辦法達到預期的效果，因此，需要另行採用不同的技術來達到即時運算即時回應的目的。

1.2 研究方法

雲端計算改變了人類獲取知識的方式，也可能把即時辨識技術真正變得實用。因此，本產學計畫擬結合崑山科技大學研發之複雜前景之移動性人物檢測與行為分析運用在雲端運算Storm技術上，並結合事件警示機制，以配合運算結果自動通報相關人員或者機構。

1.2.1 雲端系統架構

雲端運算是資料中心因應網路上資訊暴增而提出的服務及管理思維，資訊服務提供者投入資源進行雲端運算的服務及架構開發，Twitter可說是大量使用雲端運算的公司之一，為了使用戶能夠更方便實時分析大量數據，Twitter宣佈開放了分散式、容錯的雲端運算軟體 - Storm[22]。這套軟體是Twitter收購BackType後得到的技術。BackType主要替用戶即時分析組織在Twitter上的Feed、總結不同Feed及留言的數量、瀏覽量等等的數據。這套軟體與Hadoop相似，不過要是說針對進行即時分析，這套軟體卻比Hadoop強得多。首先它能夠即時的將分析後的數據儲存在資料庫又或者是外部的儲存裝置之中，而且此軟體的分析功能更不只限於Twitter，針對任何unstructured的數據亦可以進行即時分析工作。以下簡單的比較了Storm與Hadoop的不同之處：

- Storm：分散式即時計算，強調即時性，常用於即時性要求較高的地方
- Hadoop：分散式批次處理計算，強調批次處理，常用於資料採擷、分析

Storm 基本概念

Storm 是一個開放原始碼的即時運算系統，它提供了一系列的基本元素用於進行計算：Topology、Stream、Spout、Bolt 等等。

在 Storm 中，一個即時應用的計算任務被打包作為 Topology 發佈，這同 Hadoop 的 MapReduce 任務相似。但是有一點不同的是：在 Hadoop 中，MapReduce 任務最終會執行完成後結束；而在 Storm 中，Topology 任務一旦提交後永遠不會結束，除非你顯示去停止任務。

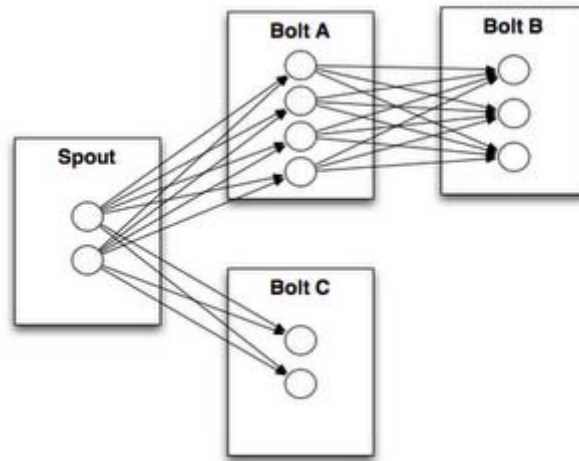


圖 1、Topology 的結構示意圖

計算任務 Topology 是由不同的 Spouts 和 Bolts，通過資料流程 (Stream) 連接起來的圖。如圖 1 所示是一個 Topology 的結構示意圖。

圖 1 中包含有：

Spout：Storm 中的訊息源，用於為 Topology 產生訊息 (資料)，一般是從外部資料來源 (如 Message Queue、RDBMS、NoSQL、Realtime Log) 不間斷地讀取資料並發送給 Topology 消息 (tuple 元組)。

Bolt：Storm 中的消息處理者，用於為 Topology 進行消息的處理，Bolt 可以執行過濾、聚合、查詢資料庫等操作，而且可以一級一級的進行處理。

最終，Topology 會被提交到 storm 叢集中運行；也可以通過命令停止 Topology 的運行，將 Topology 佔用的計算資源歸還給 Storm 叢集。

Storm 資料流程模型

資料串流 (Stream) 是 Storm 中對資料行進的抽象，它是時間上無邊界的 tuple 元組序列。在 Topology 中，Spout 是 Stream 的源頭，負責為 Topology 從特定資料來源發射 Stream；Bolt 可以接收任意多個 Stream 作為輸入，然後進行資料的加工處理過程，如果需要，Bolt 還可以發射出新的 Stream 給下級 Bolt 進行處理。

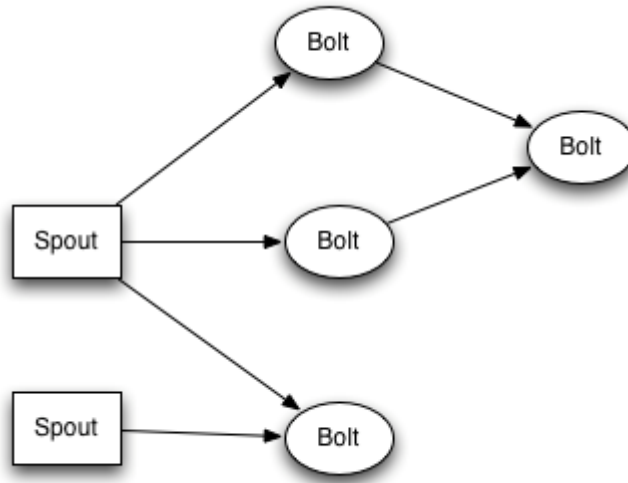


圖 2、Spout 和 Bolt 之間的資料串流關係

圖 2 所示是一個 Topology 內部 Spout 和 Bolt 之間的資料串流關係。Topology 中每一個計算元件（Spout 和 Bolt）都有一個並存執行緒，在建立 Topology 時可以進行指定，Storm 會在叢集內分配對應並行度個數的執行緒來同時執行這一元件。

那麼，有一個問題：既然對於一個 Spout 或 Bolt，都會有多個 task 執行緒來運行，那麼如何在兩個元件（Spout 和 Bolt）之間發送 tuple 元組呢？

Storm 提供了若干種資料串流群組（Stream Grouping）策略用來解決這一問題。在 Topology 定義時，需要為每個 Bolt 指定接收什麼樣的 Stream 作為其輸入（注：Spout 並不需要接收 Stream，只會發射 Stream）。

目前 Storm 中提供了以下 7 種 Stream Grouping 策略：Shuffle Grouping、Fields Grouping、All Grouping、Global Grouping、Non Grouping、Direct Grouping、Local or shuffle grouping，具體策略可以參考 <https://github.com/nathanmarz/storm/wiki/Concepts>。

Storm 不能支援的情境

以上介紹了一些 Storm 中的基本概念，可以看出，Storm 中 Stream 的概念是 Topology 內唯一的，只能在 Topology 內按照“發佈-訂閱”方式在不同的計算元件（Spout 和 Bolt）之間進行資料的流動，而 Stream 在 Topology 之間是無法流動的。如圖 3 所示為 Storm 不能支援的情境圖。

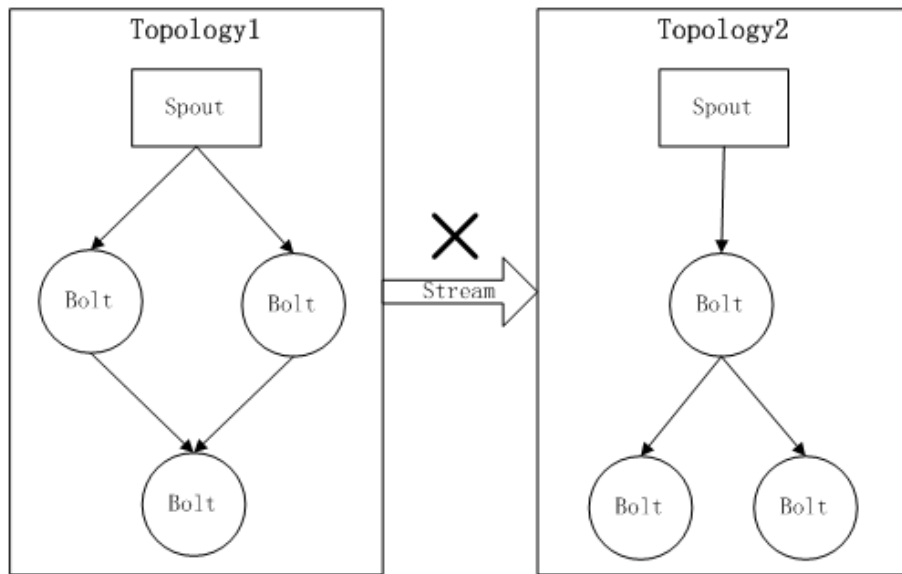


圖 3、Storm 不能支援的情境圖

這一點限制了 Storm 在一些應用，下面通過一個簡單的實例來說明。

假設現在有一個 Topology1 的結構如下：通過 Spout 產生資料流程後，依次需要經過 Filter Bolt，Join Bolt，Business1 Bolt，如圖 4 所示。其中，Filter Bolt 用於對資料進行過濾，Join Bolt 用於資料流程的聚合，Business1 Bolt 用於進行一個實際業務的計算邏輯。

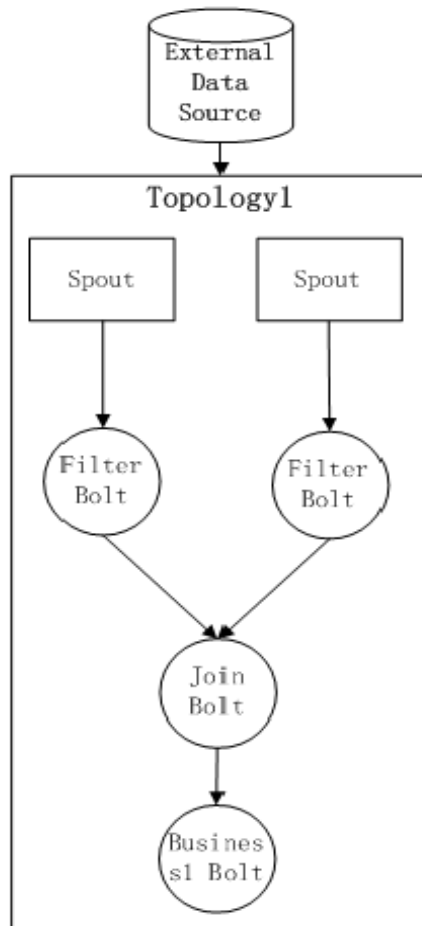


圖 4、實際業務的計算邏輯示意圖

目前這個 Topology1 已經被提交到 Storm 集群運行，而現在我們又有了新的需求，需要計算一個新的業務邏輯，而這個 Topology 的特點是和 Topology1 公用同樣的資料來源，而且前期的預處理過程完全一樣（依次經歷 Filter Bolt 和 Join Bolt），那麼這時候 Storm 怎麼來滿足這一需求？據個人瞭解，有以下幾種“曲折”的實現方式：

1) 第一種方式：首先 kill 掉已經在集群中運行的 Topology1 計算任務，然後實現 Business2 Bolt 的計算邏輯，並重新封包形成一個新的 Topology 計算任務 jar 包後，提交到 Storm 集群中重新運行，這時候 Storm 內的整體 Topology 結構如圖 5 所示：

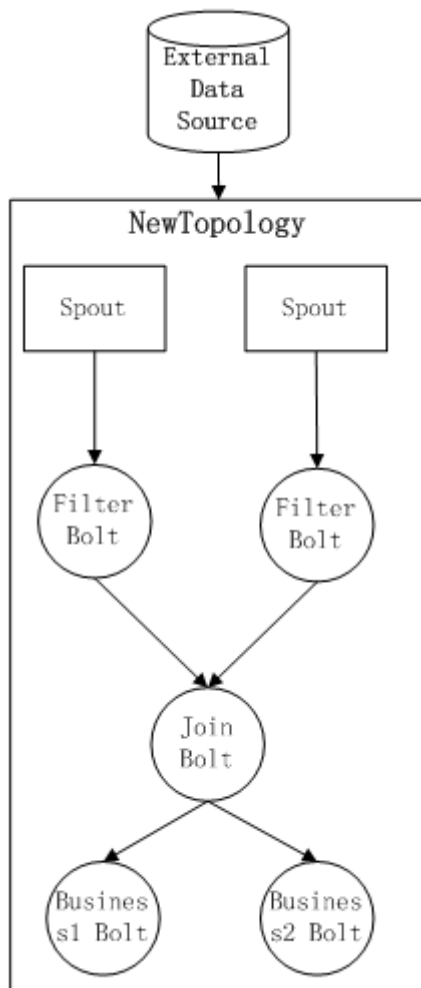


圖 5、新增 Business s2 Bolt 的 Topology

這種方式的缺點在於：由於要重啟 Topology，所以如果 Spout 或 Bolt 有狀態則會丟失掉；同時由於 Topology 結構發生了變化，因此重新運行 Topology 前需要對程式的穩定性、正確性進行驗證；另外 Topology 結構的變化也會帶來額外的運算時間。

2) 第二種方式：完全開發部署一套新的 Topology，其中前面的公共部分的 Spout 和 Bolt 可以直接複用，只需要重新開發新的計算邏輯 Business2 Bolt 來替換原有的 Business1 Bolt 即可。然後重新提交新的 Topology 運行。這時候 Storm 內的整體 Topology 結構如圖 6 所示：

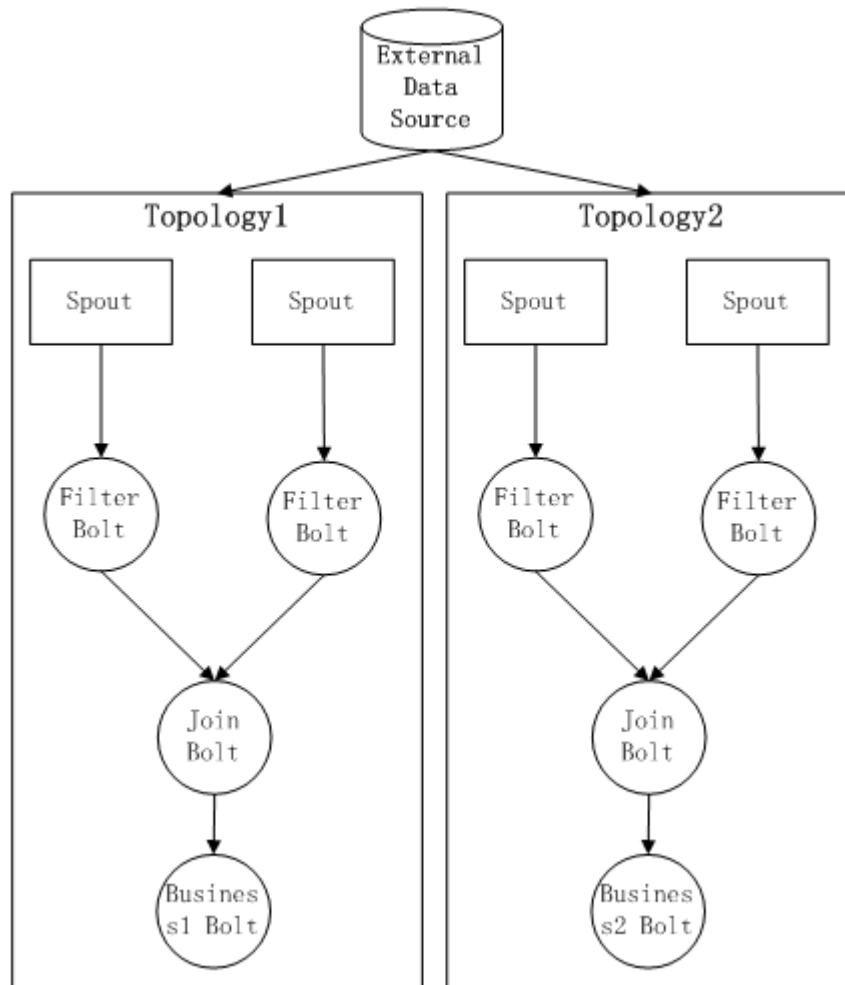


圖 6、複製一樣的 Topology

這種方式的缺點在於：由於兩個 Topology 都會從 External Data Source 讀取同一份資料，無疑增加了 External Data Source 的負載壓力；而且會導致同樣的資料在 Storm 集群內被傳輸相同的兩份，被同樣的計算單元 Bolt 進行處理，浪費了 Storm 的計算資源和網路傳輸頻寬。假設現在不止有兩個這樣的 Topology 計算任務，而是有 N 個，那麼對 Storm 的計算 Slot 的浪費很嚴重。

注意：上述兩種方式還有一個公共的缺點—系統可擴展性不佳，這意味著不管哪種方式，只要以後有這種新增業務邏輯的需求，都需要進行複雜的人工作業或線性的資源浪費現象。

3) 第三種方式：透過 Kafka 這樣的訊息中介軟體，實現不同 Topology 的 Spout 共用資料來源，而且這樣可以做到訊息可靠傳輸、訊息 rewind 回傳等，好處是對於 Storm 來說，已經有了 storm-kafka 外掛程式的支持。這時候 Storm 內的整體 Topology 結構如圖 7 所示：

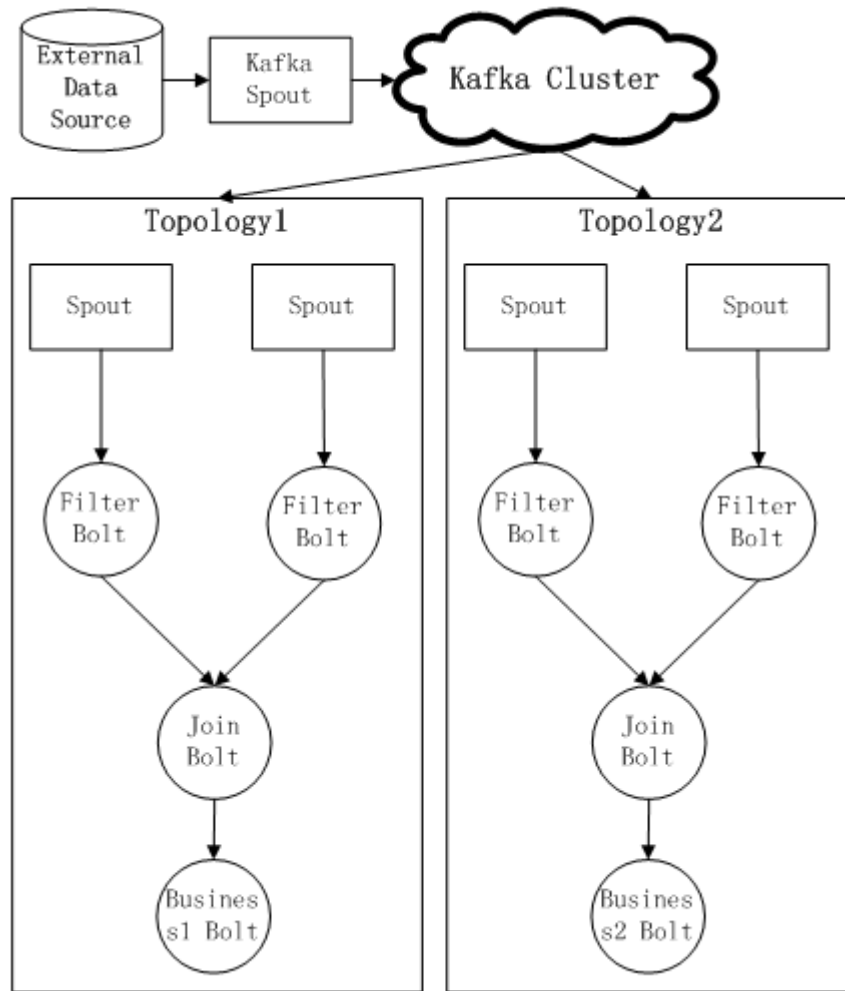


圖 7、導入 Kafka 訊息中介軟體的 Topology

這種實現方式可以通過引入一層訊息中介軟體減少對 External Data Source 的重複訪問的壓力，而且可以通過消息中介軟體層，遮罩掉 External Data Source 的細節，如果需要擴展新的業務邏輯，只需要重新部署運行新的 Topology，應該說是現有 Storm 版本下很好的實現方式了。不過消息中介軟體的引入，無疑將給系統帶來了一定的複雜性，這對於 Storm 上的應用開發來說提高了門檻。

值得注意的是，方案三中仍遺留有一點問題沒有解決：對於 Storm 叢集來說，這種方式還是沒有能夠從根本上避免資料在 Storm 不同 Topology 內的重複發送與處理。這是由於 Storm 的資料流程模型上的限制所導致的，如果 Storm 實現了不同 Topology 之間 Stream 的共用，那麼這一問題也就迎刃而解了。

一個流處理系統的資料流程模型

圖 8 所示是一個資料流處理框架的開發與應用所採用的資料流程模型範例：

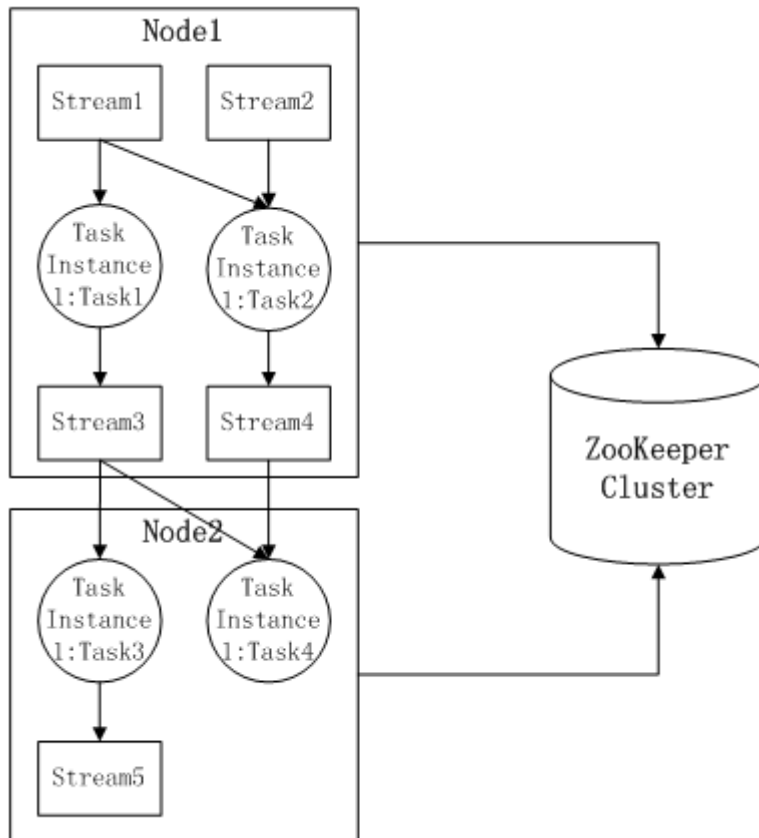


圖 8、料流處理框架的開發與應用所採用的資料流程模型範例

其中：

- 1) 資料流程 (data stream)：時間分佈和數量上無限的一系列資料記錄的集合體；
- 2) 資料記錄 (data record)：資料流程的最小組成單元，每條資料記錄包括 3 類資料：所屬資料流程名稱 (stream name)、用於路由的資料 (keys) 和具體資料處理邏輯所需的資料 (value)；
- 3) 資料處理任務定義 (task definition)：定義一個資料處理任務的基本屬性，無法直接被執行，必須特化為具體的任務實例。其基本屬性包括：
 - (選項) 輸入流 (input stream)：描述該任務依賴哪些資料流程作為輸入，是一個資料流程名稱列表；資料流程產生源不會依賴其他資料流程，可忽略該配置；
 - 資料處理邏輯 (process logic)：描述該任務具體的處理邏輯，例如由獨立進程進行的外部處理邏輯；
 - (選項) 輸出流 (output stream)：描述該任務產生哪個資料流程，是一個資料流程名稱；資料流程處理鏈末級任務不會產生新的資料流程，可忽略該配置；
- 4) 資料處理任務實例 (task instance)：對一個資料處理任務定義進行具體約束後，可推送到某個處理結點上運行的邏輯實體。附加下列屬性：

- 資料處理任務定義：指向該任務實例對應的資料處理任務定義實體；

- 輸入流過濾條件 (input filtering condition)：一個 boolean 運算式列表，描述每個輸入流中符合什麼條件的資料記錄可以作為有效資料交給處理邏輯；若某個輸入流中所有資料記錄都是有效資料，則可直接用 true 表示；

- (選項)強制輸出週期 (output interval)：描述以什麼頻率強制該任務實例產生輸出流記錄，可以用輸入流記錄個數或間隔時間作為週期；忽略該配置時，輸出流記錄產生週期完全由處理邏輯自身決定，不受框架約束；

5) 資料處理結點 (node)：可容納多個資料處理任務實例運行的實體機器，每個資料處理結點的 IPv4 位址必須保證唯一。

該分散式流處理系統由多個資料處理結點 (node) 組成；每個資料處理結點 (node) 上運行有多個資料任務實例 (task instance)；每個資料任務實例 (task instance) 屬於一個資料任務定義 (task definition)，任務實例是在任務定義的基礎上，添加了輸入流過濾條件和強制輸出週期屬性後，可實際推送到資料處理結點 (node) 上運行的邏輯實體；資料任務定義 (task definition) 包含輸入資料流程、資料處理邏輯以及輸出資料流程屬性。

該系統中，通過分散式應用程式協調服務 ZooKeeper 集群存儲以上資料流程模型中的所有配置資訊；不同的資料處理節點統一通過 ZooKeeper 集群獲取資料流程的配置資訊後進行任務實例的運行與停止、資料流程的流入和流出。

同時，每個資料處理任務可以接受流系統中已存在的任意資料流程 (data stream) 作為輸入，並產出新的任意名稱的資料流程 (data stream)，被其他結點上運行的任務實例訂閱。不同結點之間對於各個資料流程 (data stream) 的訂閱關係，通過 ZooKeeper 集群來動態感知並負責通知流系統做出變化。

二者在資料流程模型上的不同之處

至於兩個系統的實現細節，我們先不去做具體比較，下面僅列出二者在資料流程模型上的一些不同之處（這裡並不是為了全面對比二者的不同之處，只是列出其中的關鍵部分）：

1) 在 Storm 中，資料流程 Stream 是在 Topology 內進行定義，並在 Topology 內進行傳輸的；而在上面提到的流處理系統中，資料流程 Stream 是在整個系統內全域唯一的，可以在整個集群內被訂閱。

2) 在 Storm 中，資料流程 Stream 的發佈和訂閱都是靜態的，所謂靜態是指資料流程的發佈與訂閱關係在向 Storm 集群提交 Topology 計算任務時，被一次性生成的，這一關係在 Topology 的運行過程中是不能被改變的；而在上面提到的流處理系統中，資料流程 Stream 的發佈和訂閱都是動態的，即資料處理任務 task 可以動態的發佈 Stream，也可以動態的訂閱系統內已經生

成的任意 Stream，資料流程的訂閱關於通過分散式應用程式協調服務 ZooKeeper 集群的動態節點來維護管理。

有了以上的對比，我們不難發現，對於本文所舉的應用場景實例，Storm 的資料流程模式尚不能很方便的支援，而在這裡提到的這個流處理系統的全域資料流程模型下，這一應用場景的需求可以很方便的滿足。

Storm 有必要實現不同 Topology 之間 Stream 的共用，這個至少可以在不損失 Storm 現有功能的前提下，使得 Storm 在處理實際生產環境下的一些應用場景時更加從容應對。

至於如何在現有 Storm 的基礎上實現這一需求，可能的方式很多。一種簡單的方式是通過 Zookeeper 來集中存儲、動態感知 Topology 之間 Stream 的“發佈-訂閱”關係，同時在 Storm 的消息分發過程中對這種情況加以處理。

Storm 的優點：

- 簡單程式設計

在大資料處理方面相信大家對 hadoop 已經耳熟能詳，基於 Google Map/Reduce 來實現的 Hadoop 為開發者提供了 map/reduce 開發模型，使平行批次程式變得非常地簡單和優美。同樣，Storm 也為大資料的即時計算提供了一些簡單優美的開發模型，這大大降低了開發並行即時處理的任務的複雜性，幫助你快速、高效的開發應用。

- 多語言支援

除了用 java 實現 spout 和 bolt，你還可以使用任何你熟悉的程式設計語言來完成這項工作，這一切得益於 Storm 所謂的多語言協定。多語言協定是 Storm 內部的一種特殊協定，允許 spout 或者 bolt 使用標準輸入和標準輸出來進行消息傳遞，傳遞的消息為單行文本或者是 json 編碼的多行。

Storm 支援多語言程式設計主要是通過 ShellBolt, ShellSpout 和 ShellProcess 這些類來實現的，這些類都實現了 IBolt 和 ISpout 介面，以及讓 shell 通過 java 的 ProcessBuilder 類來執行腳本或者程式的協定。

可以看到，採用這種方式，每個 tuple 在處理的時候都需要進行 json 的編解碼，因此在輸送量上會有較大影響

- 支援水平擴展

在 Storm 集群中真正運行 topology 的主要有三個實體：工作程序、執行緒和任務。Storm 叢集中的每台機器上都可以運行多個工作程序，每個工作程序又可創建多個執行緒，每個執行緒可以執行多個任務，任務是真正進行資料處理的實體，我們開發的 spout、bolt 就是作為一個或者多個任務的方式執行的。

因此，計算任務在多個執行緒、進程和伺服器之間並行進行，支持靈活的水準擴展。

- 容錯性強

如果在消息處理過程中出了一些異常，Storm 會重新安排這個出問題的處理單元。Storm 保證一個處理單元永遠運行（除非你顯示殺掉這個處理單元）。

- 可靠的訊息保證

Storm 可以保證 spout 發出的每個訊息都能被“完全處理”，這也是直接區別於其他即時系統的地方，如 Yahoo S4 – distributed stream computing platform[25]。

- 快速的訊息處理

用 ZeroMQ 作為底層訊息佇列，保證訊息能快速被處理

- 本地模式，支援快速程式設計測試

Storm 有一種“本地模式”，也就是在進程中類比一個 Storm 集群的所有功能，以本地模式運行 topology 跟在集群上運行 topology 類似，這對於我們開發和測試來說非常有用。

1.2.2 複雜前景之移動性人物檢測與行為分析技術

由於科技的進步，監視系統的快速發展，從早期錄影帶，到現在透過電腦硬碟儲存，不僅可儲存的時間變長、影像也可以較以往清晰、甚至局部縮放，最重要的是可以減少因人工未更換影帶而造成未儲存到的結果。

然而現今大多路口都有加裝監視設備，若能加以利用這些設備，不光只是以往只有錄影的功能，可以在發生事故時，即時的發出警報、通知救援，就可以使這樣設備發揮出更大的功效，並且主要應用在行人方面，因為人與汽、機車在路上來說相對的比較危險，尤其是當有沒公德心的人將車子停在行人可走的範圍上，迫使行人往往需要走到機車道、甚至跨越到快車道，此時行人的生命安全將受到很大的威脅，然而當發生事故時，白天或許有目擊證人，當到了夜晚若是發生意外當下又沒目擊證人時，便可以藉由此系統對於此狀況加以判斷並且通報相關單位來處理。

電腦視覺的多年發展以來，應用的層面越來越廣闊，並有著許多相當有貢獻的成果，在電腦視覺中，要增加其正確的辨識率，首要的條件便是能完整的去除掉背景，完整的呈現出前景物，將其雜點盡可能的減少對於人的行為與姿態分析也發展許多年，例如即時人體追蹤[2]，透過前景物與背景物的分佈機率估算，再計算輸入以及已知場景的像素點之馬氏距離，即可獲得前景物。Haritaoglu 和 Harwood 等人提出的一套人和人的姿態即時監控系統[3]，利用輪廓定位出人體細部模組(頭、軀幹、手、腳)。對人的行為分析可能的姿態並加以分類[4]，不需要事先建立人體模型，直接利用特徵來進行辨識，並透過目標物的輪廓可能出現的機率進行動作的

辨識，以及 Aggarwal 等人提出的人體連續動作辨識[5]，透過取得身體及上下肢體角度的變化來辨識一連串連續的動作，而文獻[6]使用立體視鏡套上 3D 人體模型，優點可以分析前後空間行人的位置，但此方法必須透過較複雜的演算法才能達成。利用移動物資訊偵測[7]，利用人走路節奏成週期性當作行人辨識的特徵，必須搭配另外建立之模型，才可有效檢測出行人，也有利用隱藏馬可夫模型[8]，預測人的步伐及姿態追蹤失敗並且應用更強健的分析方法。在移動物體陰影偵測為 Cucchiara [9]等人所提出，利用像素點的變化以及物體的大小，將物體資訊分為移動物體、非真實物體(ghost)、移動物體的陰影與非真實物體的陰影等四個種類，再根據種類的做不同方式的背景更新，便能夠找到最適合目前環境的背景資訊，而建立完整的背景資訊。

檢測前景物使用的方法、背景影像相減法以及連續影像相減法介紹如下：

1. 背景影像相減法[3][4]：此方法是利用一張預先建立的背景，將當前影像與其做相減得到前景物之像素值，由於只有前景物部份與背景不相同，因此容易取得目標物，但缺點為容易受其他因素影響，會造成很多雜點的產生，例如建立背景時的光線與當前影像所遮避過後產生不同的像素值差。

2. 連續影像相減法[10][11]：此種方法則不需預先建立背景，採用當前影像與上一張影像的相減，即能直接獲得前景物的資訊，也因為是連續做影像的相減，所以受外在的影響不大，因而能有效的減少雜點的產生，然而最主要的缺點則是在於當物體停止不動時會無法正確的分離出前景物，因為兩張看起來幾乎一樣的影像相減後，會得不到所要的資訊。在物體的追蹤方面，幾位學者[12][13]提出將目標分成區塊，並統計色彩相似區塊中的色彩資訊，在追蹤時利用區塊的資訊與畫面做對應，好處是當目標物重疊有合併情形時，仍然可以正確的分割出目標物。

為了達成檢測移動性人物的目的，複雜前景之移動性人物檢測的系統流程如圖 9 所示。流程中的各步驟功能說明如下：

1. 影像輸入：經固定攝影機拍攝成影片或直接做即時處理。
2. 背景建立與更新：經由統計像素點出現的機率，並以所取樣之影像相似度較高者做為背景，或依照時間地點之不同做調整，例如可一段時間做定點之拍攝畫面當為背景。
3. 背景與連續影像相減：此為取出前景物的重要步驟。
4. 二值化與形態學運算：將其資料量變小並消除雜點。
5. 連接元標定：將相連之區塊與其他未相連的區域加以區隔。
6. 目標框定：將所要之資訊正確的框選出。
7. 影像輸出：完成所有步驟後之連續影像輸出。

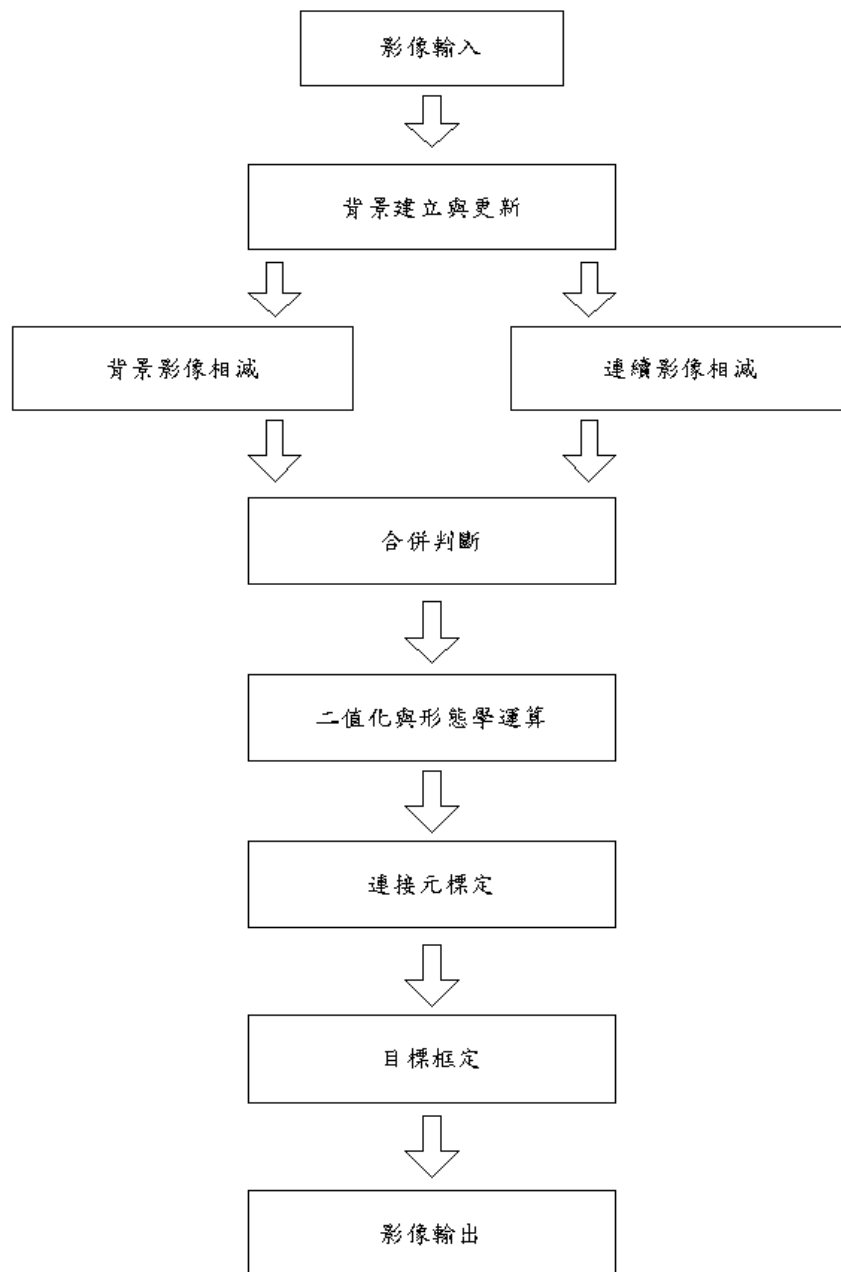


圖9、複雜前景之移動性人物檢測的系統流程

移動物檢測與追蹤

對於移動物的檢測以及追蹤，需要對所要檢測的影像施以一連串的处理才能夠準確的追蹤，以及完整的檢測出目標物。因此本章節首先敘述如何在非靜態背景中建立起背景，接著使用影像相減法，以及相關影像處理方法，當影像中出現移動性的人物時，可以即時處理並擷取出特徵。

建立一張完整的背景影像，對於移動物檢測及追蹤是非常重要的步驟。本論文在背景建立的過程中，根據不同環境條件採用了兩種方式來建立。第一種方式是根據連續影像之影像序列做像素點統計，將每張影像之相同位置上的像素值做統計，之後將重複出現次數較多者合併成為背景影像。如圖 10 原連續影像序列(一)及圖 11 原連續影像序列(二)所示分別為圖 12 原連續影像序列(一)經統計所得背景及圖 13 原連續影像序列(二)經統計所得背景。



圖10 原連續影像序列(一)

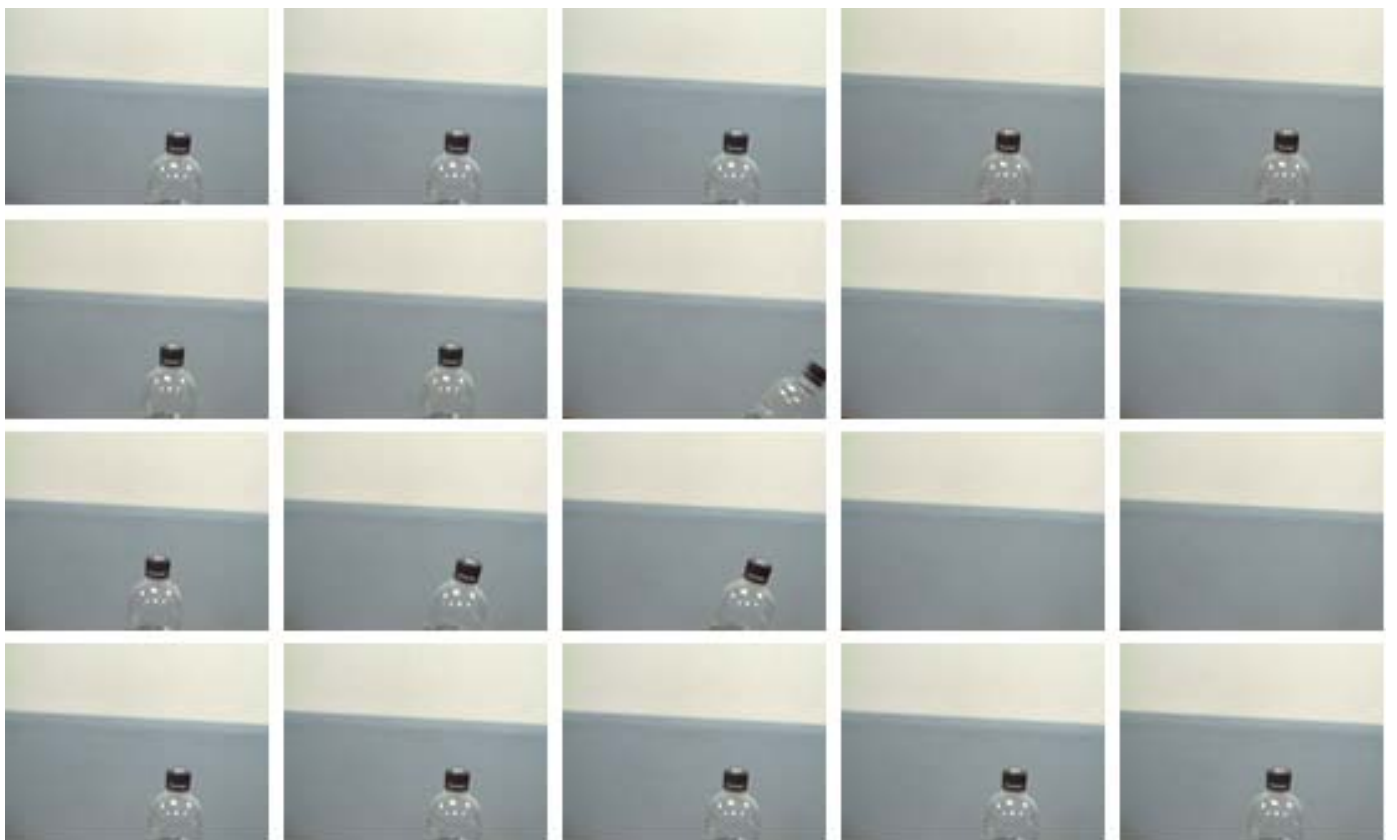


圖11 原連續影像序列(二)

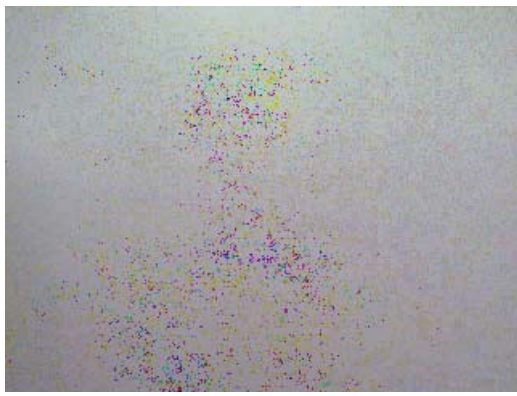


圖 12 原連續影像序列(一)經統計所得背景



圖 13 原連續影像序列(二)經統計所得背景

經由第一種方法所建立出來之背景，其結果常因用來建立背景的影像中有移動物體不在固定之地方時，會產生些許的雜點甚至是殘影。為了改進第一種方式會有雜點或殘影的缺點，本論文提出第二種建立背景的方式，本方式是將第一種方式所建立出來的背景再與影像序列相比較，其中最為相近者即被採用為真正的背景，如圖 14 及 15 所示分別為圖 12 及圖 13 經本方式修正後的結果。

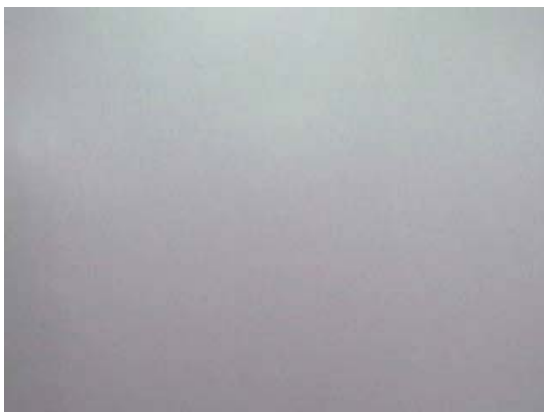


圖 14 統計圖背景(一)經比較選定後的背景



圖 15 統計圖背景(二)經比較選定後的背景

圖 14 與圖 15 結果顯示，利用此種方式可以有效的將背景依所設定的時間或是取樣的影像張數來加以更新。當物體移入到定點時經過一段時間，則會將它更新於背景中；或者當物體只是經過但並未停留太久時，則此方式會自動將它屏除於背景，因而得到一張乾淨的背景，有利於往後的人物追蹤處理。

背景影像相減法

一般背景影像相減法是利用連續影像對預先建立好之背景影像直接對像素值做相減，之後訂定一閾值當像素值大於閾值時，則給予當前影像之像素值，若小於閾值時即設為零，其完整公式如(1.1)~(1.3)式所示

$$\begin{aligned}
F_{R(n)}(i, j) &= |C_{R(n)}(i, j) - B_R(i, j)| \\
F_{G(n)}(i, j) &= |C_{G(n)}(i, j) - B_G(i, j)| \\
F_{B(n)}(i, j) &= |C_{B(n)}(i, j) - B_B(i, j)|
\end{aligned} \tag{1.1}$$

$$Total(i, j) = F_{R(n)}(i, j) + F_{G(n)}(i, j) + F_{B(n)}(i, j) \tag{1.2}$$

$$\begin{aligned}
Sub_R(i, j) &= \begin{cases} 0 & \text{if } Total(i, j) < Threshold \\ C_{R(n)}(i, j) & \text{otherwise} \end{cases} \\
Sub_G(i, j) &= \begin{cases} 0 & \text{if } Total(i, j) < Threshold \\ C_{G(n)}(i, j) & \text{otherwise} \end{cases} \\
Sub_B(i, j) &= \begin{cases} 0 & \text{if } Total(i, j) < Threshold \\ C_{B(n)}(i, j) & \text{otherwise} \end{cases}
\end{aligned} \tag{1.3}$$

其中 $F_{R(n)}(i, j)$ 、 $F_{G(n)}(i, j)$ 及 $F_{B(n)}(i, j)$ 分別為連續影像與背景影像相減之 R、G、B 像素值； $C_{R(n)}(i, j)$ 、 $C_{G(n)}(i, j)$ 及 $C_{B(n)}(i, j)$ 分別為原始連續影像之 R、G、B 像素值； $B_{R(n)}(i, j)$ 、 $B_{G(n)}(i, j)$ 及 $B_{B(n)}(i, j)$ 分別為背景影像之 R、G、B 像素值； n ：表示連續影像第 n 張影像； $Total(i, j)$ ：前景像素值總合； $Sub_R(i, j)$ 、 $Sub_G(i, j)$ 及 $Sub_B(i, j)$ 為經背景影像相減法分別所得之 R、G、B 像素值，當前景像素值小於門檻值時為零，否則等於影像前景像素值。例如圖 16~18 所示為背景影像相減之結果，由圖 18 可清楚的看到影像經背景影像相減法處理後，已將大部分之背景完全去除。



圖 16 目前影像



圖 17 背景影像

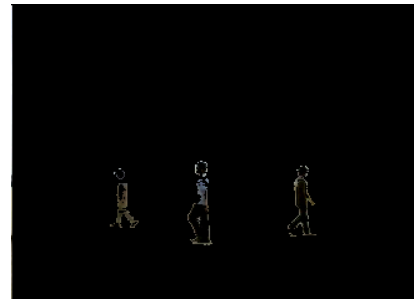


圖 18 前景物影像

連續影像相減法

連續影像相減法最大的優點就是在於不用事先建立背景，當連續影像中之物體持續移動時，透過此方法可以準確取得移動物之資訊，也因為是連續影像相減，光線的瞬間變化量小，所以不易受到光線之影響，其相減運算式如(1.4)式所示。

$$\begin{aligned}
CF_{R(n)}(i,j) &= |C_{R(n)}(i,j) - C_{R(n-1)}(i,j)| \\
CF_{G(n)}(i,j) &= |C_{G(n)}(i,j) - C_{G(n-1)}(i,j)| \\
CF_{B(n)}(i,j) &= |C_{B(n)}(i,j) - C_{B(n-1)}(i,j)|
\end{aligned} \tag{1.4}$$

其中 $CF_{R(n)}(i,j)$ 、 $CF_{G(n)}(i,j)$ 及 $CF_{B(n)}(i,j)$ 分別為第 n 張前景影像減去第 $n-1$ 張前景影像的 R、G、B 像素值； $C_{R(n)}(i,j)$ 、 $C_{G(n)}(i,j)$ 及 $C_{B(n)}(i,j)$ 分別為第 n 張連續影像之 R、G、B 像素值； $C_{R(n-1)}(i,j)$ 、 $C_{G(n-1)}(i,j)$ 及 $C_{B(n-1)}(i,j)$ 分別表示目前影像之前一張影像的 R、G、B 像素值。

移動物體之獲得

在移動物體檢測[19]及追蹤時，有時候往往只需要移動物體形狀，並不需要完整的形貌，因此經常利用二值化來處理，它不僅可以減少資料量，提升電腦對於影像處理的速度，更可以有效且明顯的將物體與背景做區隔。本論文於連續影像相減之後，再利用(1.5)及(1.6)式來取得二值化後之移動物資訊。

$$CTotal(i,j) = CF_{R(n)}(i,j) + CF_{G(n)}(i,j) + CF_{B(n)}(i,j) \tag{1.5}$$

$$\begin{aligned}
CSub_R(i,j) &= \begin{cases} 0 & \text{if } CTotal(i,j) < CThreshold \\ 255 & \text{otherwise} \end{cases} \\
CSub_G(i,j) &= \begin{cases} 0 & \text{if } CTotal(i,j) < CThreshold \\ 255 & \text{otherwise} \end{cases} \\
CSub_B(i,j) &= \begin{cases} 0 & \text{if } CTotal(i,j) < CThreshold \\ 255 & \text{otherwise} \end{cases}
\end{aligned} \tag{1.6}$$

其中 $CTotal(i,j)$ 表前景像素值總合； $CSub_R(i,j)$ 、 $CSub_G(i,j)$ 及 $CSub_B(i,j)$ 表經連續影像相減法分別所得之 R、G、B 影像二值化。例如圖 19 - 圖 21 所示為連續影像相減法的結果。



圖 19 前一張影像



圖 20 目前影像

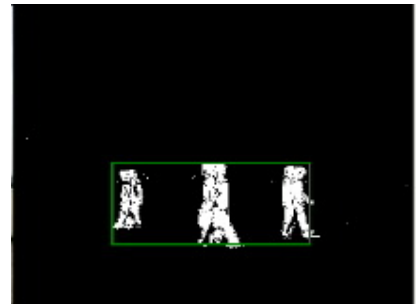


圖 21 經(3.4)~(3.6)式處理後的結果

在圖 21 中，往往須加以框選連續影像相減法所得之前景物資訊，以便確定框選區域為移動體之範圍(綠色框)，使得往後步驟皆根據此範圍加以處理，避免浪費時間來處理其他不存在移動物體之區域。

有時影像經過連續影像相減法與二值化處理後會有些許的雜點，以及影像較為不平滑或不完整，此時必須再經過形態學的處理。本論文使用兩次閉合運算後一次斷開運算，使得移動物目標更明顯。例如圖 22 所示為圖 21 影像形態學運算後的結果。

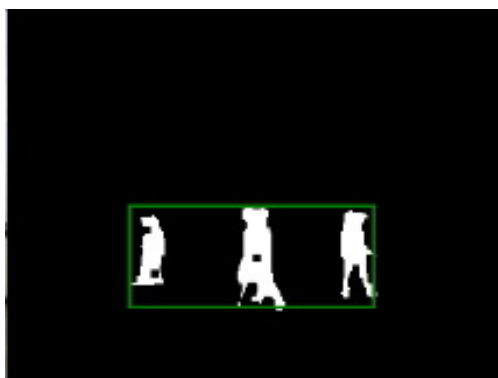


圖 22 為圖 21 影像經形態學處理後的結果

從圖 22 中清楚的看到，經形態學處理後之影像可完整的得到輪廓。完成形態學的運算後便要在畫面中框選出各別的移動物體，這時候就必須使用到連接成份的標示，編號出相連的區塊，再根據相同編號做區塊的框選。例如圖 23 所示為圖 22 經連接成份標示及框選的結果。



圖 23 連接成份標示後所框定之目標物

從圖 23 中清楚的看到將原本為畫面中移動物體的框選，經連接成份的標示過後，可以正確的框選出個別目標。

移動性人物檢測

移動性人物檢測主要在複雜的前景中針對移動性人物加以檢測，以及分析幾種姿態之行為，一般在複雜環境中，將前景物與背景分離後，通常不會只有移動性人物的資訊，往往還有

其它像是汽機車等的移動物體。因此在本論文中，為了正確尋找出移動中的人物，進一步利用人與其它移動物體之高寬比例不同的特性，從複雜的環境中將移動性人物檢測出來。

首先利用前 40 張的影像建立起完整的背景，再透過背景影像相減法即可完整的呈現出前景物。但此做法當背景物體有所變更或移動時，會造成畫面上不完整的缺點。因此本論文將連續影像相減法與背景影像相減法加以結合，以能獲得更佳的效果。在”移動物之獲得”一節中，已經在整個畫面中框選出個別的移動物體，本節則進一步在所有個別的移動性物體中找出移動性人物。本研究中可能出現的移動物體包括移動性人物以及汽機車，所以必須針對這些物體的特性加以探討。為了能順利檢測出移動性人物，本論文利用移動性人物與其它移動物體高與寬比例不同的特性來加以區別，舉例來說：移動性人物之比例與汽機車之比例明顯不同，且移動性人物的高明顯的大於寬，因此透過此方式設定好比例即可正確的辨識出移動性人物，本論文所使用判斷式(1.7)式及(1.8)式所示，例如圖 24 所示為高寬示意圖。

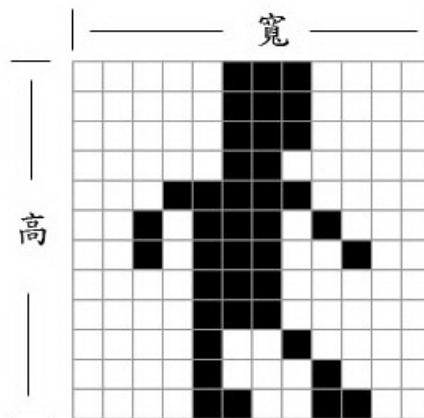


圖 24 高寬示意圖

$$\text{height} > \text{width} \tag{1.7}$$

$$\text{rate} = \frac{\text{width}}{\text{height}} \quad \text{rate} \leq 0.8 \tag{1.8}$$

其中(1.7)式表示高度大於寬度時為辨識移動性人物的必要條件；(1.8)式 rate 的門檻值是為了檢測當人為站立行走時的狀態。

當輸入影像經連續影像相減法、二值化、連通成份標示及加入(1.7)式與(1.8)式處理後，即可得到移動性人物的範圍，再經背景影像相減法處理後，與已去除大部分背景的前景影像相配合，並將連續影像相減法所得到的範圍去框選背景影像相減法的資訊，即可完整的框選到移動性人物。例如圖 25 - 圖 26 所示為移動性人物的框選，畫面中同時出現移動性人物以及車輛，經由測試可以正確的框選到人物。



圖 25 原始影像

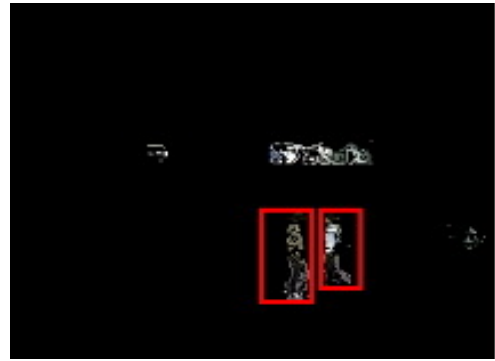


圖 26 框選到目標影像

移動性人物行為分析

在移動性人物行為分析方面，當移動性人物由行走變為蹲下時，所利用之方法與上一節有些許不同，主要以背景影像相減法為主，連續影像相減法為輔，並在這兩種影像相減法的架構下加入移動性人物為蹲下姿態時候的判斷式，如(1.9)式所示。

$$\text{rate} = \frac{\text{width}}{\text{height}} \quad 0.8 \leq \text{rate} \leq 1.1 \quad (1.9)$$

當人逐漸往下蹲時高寬比會開始增加，然而當高寬比落在(1.9)式的範圍時，即判斷為蹲下，如圖 27~28 所示為移動性人物及蹲下的檢測結果。



圖 27 原始影像

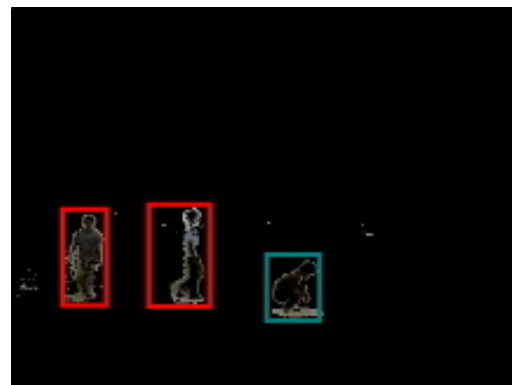


圖 28 框選到蹲下測試

圖 28 分別以不同顏色區分移動性人物(紅色)與蹲下(綠色)不同之姿態得知，利用比例來判斷人體姿勢有不錯之效果。

當移動性人物姿態為趴下或躺下狀態時，判斷的方法大致與蹲下相似，不同之處為判斷式的範圍，其寬度必定大於高度，判斷式如(1.10)式所示。

$$\text{rate} = \frac{\text{width}}{\text{height}} \quad 3 \leq \text{rate} \leq 4 \quad (1.10)$$

當高寬比滿足此 (1.10)式條件時，所框定之目標即為趴下或躺下。例如圖 29~30 所示為移動性人物以及趴下檢測結果。



圖 29 原始影像



圖 30 框選到趴下檢測

圖 30 分別以不同顏色區分移動性人物與趴下之姿態，其中黃色框為趴下的檢測結果，利用比例來判斷人體姿勢有不錯之效果。

移動性人物趴下的檢測技術可應用在人煙稀少之地區，透過攝影機的監視，當原本為移動性人物發生突發狀況，例如：在暗巷遭受飛車搶劫跌倒、在無目擊證人的情況下遭受撞擊跌倒，當類似情況發生時，有可能因跌倒撞擊到頭部或是嚴重受傷造成倒地不起，這個時候皆可以透過本論文的趴下、躺下偵測，對於倒地的目標進行時間判斷，當目標持續太久、或是直接倒地不動，就在第一時間進行通報。

1.3 系統平台建置

搭配系統的開發，需要整合一些雲端執行所需要的軟體，以下分別說明

a. 雲端運算環境

雲端作業系統	Xen Cloud Platform
虛擬機器	Ubuntu Linux
分散式即時雲端系統架構	Storm
通訊系統	XMPP(Openfire)

b. 硬體設備

雲端硬體機器	PC 2 部以上
PTZ 攝影機	2 部以上
手機	1 部以上

3. 計畫工作預定進度及查核點說明

計畫執行工作的預定進度將分月執行，並設定重大進度查核點，如下表所示：

表二、進度及查核點說明表

月份	進度	查核點
2013/11	雲端開發環境 Xen Cloud Platform 建置	完成開發環境建置
2013/12	完成 Storm 運算環境建置	完成雲端運算環境建置
2014/1	研究及發展本計畫所運用之影像處理的各種技術及方法	完成本計畫影像前處理技術開發
2014/2	研究 Storm 架構應於複雜前景之移動性人物檢測與行為分析的模型	完成複雜前景之移動性人物檢測與行為分析
2014/3	利用 Storm 進行背景影像相減法演算法開發	完成背景影像相減法演算法開發
2013/4	利用 Storm 進行連續影像相減法演算法開發	完成連續影像相減法演算法開發
2014/5-2014/6	利用 Storm 進行移動物體檢測及追蹤演算法開發	完成移動物體檢測及追蹤演算法開發
2014/7-2014/8	利用 Storm 進行移動性人物檢測及移動性人物行為分析開發	完成移動性人物檢測及移動性人物行為分析開發
2014/9	搭配警示機制開發網頁瀏覽系統及雲端 XMPP cluster 系統環境的建置	完成雲端系統環境的建置
2014/10	複雜前景之移動性人物檢測與行為分析技術系統的整合測試	完成複雜前景之移動性人物檢測與行為分析技術

(四) 預期完成之工作項目及成果：1.預期完成之工作項目及具體成果。2.對於學術研究、產業界、國家發展及其他應用方面預期之貢獻。3.本產學計畫之技術或知識服務應用等範圍（請以量化數據說明對合作企業的貢獻，如提升多少產值等）。4.對於參與之工作人員，預期可獲之訓練。

1. 預期完成之工作項目及具體成果：

- (1) 完成雲端 IaaS 平台建置。
- (2) 完成雲端 SaaS 分散式容錯即時運算技術-Storm 的建置
- (3) 完成 Storm 應用系統模型的研究與發展；
- (4) 完成 Storm 上背景影像相減法演算法研究及發展；
- (5) 完成 Storm 上連續影像相減法演算法的研究及發展；
- (6) 完成 Storm 上移動性人物檢測及移動性人物行為分析的研究及發展；
- (7) 完成運用雲端運算技術應用於複雜前景之移動性人物檢測與行為分析技術之監視系統。
- (8) 完成 XMPP 社群網路通訊系統之建置。

2. 對於學術研究、產業界、國家發展及其他應用方面預期之貢獻：

本產學計畫完成後預期之貢獻如下：

過去監視系統主要是在所監視的區域架設多台攝影機去擷取影像，監測者必須同時觀察由不同地方所擷取的畫面影像，以達到監控的目的。即使裝設了旋轉底座或架多台攝影機，也只能達到監視掃描的作用，並不能有效的去做即時追蹤的功能。

安全監控產業，雲端計算會把整個行業真正提升到智慧安防的時代。以現今大多路口都有加裝監視設備，若能加以利用這些設備，不光只是以往只有錄影的功能，可以在發生事故時，即時的發出警報、通知救援，就可以使這樣設備發揮出更大的功效，並且主要應用在行人方面，因為人與汽、機車在路上來說相對的比較危險，尤其是當有沒公德心的人將車子停在行人可走的範圍上，迫使行人往往需要走到機車道、甚至跨越到快車道，此時行人的生命安全將受到很大的威脅，然而當發生事故時，白天或許有目擊證人，當到了夜晚若是發生意外當下又沒目擊證人時，便可以藉由此系統對於此狀況加以判斷並且通報相關單位來處理。

3. 本產學計畫之技術或知識服務應用等範圍

目前柏力揚興業有限公司雖然在行動網路裝置、視訊會議、多媒體及監視系統等設備的開發整合方面，其市場經驗與上、下游的整合已有相當豐富的成果，但系統功能上大至與別的公司類同，因此較不具競爭力。若能與學界進行產學合作，積極研發行動網路裝置相關之智慧雲端監視技術與設備，加強行動網路裝置功能以及整合智慧雲端監視系統，則將可實現產、學雙方長短產生互補之優勢，亦能提高視訊會議、多媒體、雲端運算技術及監控系統產品的競爭力

及市場佔有率約 10%，並能結合台南與路竹科學園區新世代與高雄軟體園區雲端系統的研發能量，將南台灣產業導入未來數位生活產業中，以開創未來生活之即時視訊會議、多媒體及監控應用的產業聚落。

4. 參與之工作人員，預期可獲之訓練：

- (1) 雲端運算 IaaS 私有雲之建置。
- (2) 大量虛擬機器之建置與 Storm 即時分散式運算之基礎。
- (3) 複雜前景之移動性人物檢測與行為分析技術的研究及發展能力；
- (4) XMPP 社群網路通訊系統之建置。

參考文獻

- [1] 孫銘鐘, “複雜前景之移動性人物檢測與行為分析”, Thesis for Master of Science, Department of Electrical Engineering, Kun Shan University, July 2009.
- [2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: Real-time tracking of the human body”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp.780-785, 1997.
- [3] I. Haritaoglu, D. Harwood and L.S. Davis, “W4: Real-time surveillance of people and their activities”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 809-830, 2000.
- [4] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, “Probabilistic posture classification for human-behavior analysis”, IEEE Transactions on Systems, Man, and Cybernetics, Part A 35(1): 42-54, 2005.
- [5] A.Ali and J.K.Aggarwal, “Segmentation and recognition of continuous human activity”, in IEEE Workshop on Detection and Recognition of Events in Video, pp. 28-35, 2001.
- [6] S. Wachter, H. H. Nagel, “Tracking Persons in Monocular Image Sequences”, Computer Vision and Image Understanding, Volume 74 , Issue 3, pp. 174-192, 1999.
- [7] L.C. Fu, and C.Y.Liu, “Computer Vision Based Object Detection and Recognition for Vehicle Driving”, IEEE Proc. on Robotics & Automation, pp.2634-2641, 2001
- [8] Shiloh L. Dockstader, and Nikita S. Imennov, “Prediction for Human Motion Tracking Failures” IEEE Transactions on Image Processing, Vol. 15, No. 2, pp.411-421, 2006.
- [9] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, “Detecting Moving Objects, Ghost, and Shadows in Video Streams”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, pp. 1337-1342, 2003.

- [10] C. Stauffer, W. E. L. Grimson, “Presented an Adaptive Background Model for Real-time Tracking”, IEEE Proc. Computer Society Conf. on Computer Vision and Pattern Recognition, Vol 2, pp. 246-252, 1999.
- [11] M. Ekinici, E. Gedikli, “Sihouette Based Human Motion Detection and Analysis for Real-time Automated Video Surveillance”, Vision, Modeling, and Visualization 2000: Proceedings, pp. 22-24, 2000.
- [12] S.Khan and M.Shah, “Tracking people in presence of occlusion”,in Proc. Asian Conf. Computer Vision, Taipei, Taiwan, 2000.
- [13] A. Cavallaro, O. Steiger and T. Ebrahimi, “Tracking video objects in cluttered background,” IEEE Trans. Circuits and Systems for Video Technology, Vol. 15, No. 4, pp. 575-584 Apr. 2005.
- [14] 連國珍，數位影像處理，儒林出版，2007 年 9 月四版一刷
- [15] R.C.Gonzalez and R.E.Woods , Digital Image Processing , 2nd Edition, 2002.
- [16] 張穎華,“即時行人監控系統”,國立中央大學資訊工程研究所碩士論文, 2006.
- [17] 吳居穆,“利用轉換狀態圖及模型建立進行人體運動姿勢辨識”,國立中央大學資訊工程研究所碩士論文,2006.
- [18] Donald Hearn and M.Pauline Baker , Computer Graphics , 3th Edition , 2004.
- [19] P.Wayne Power, and Johann A.Schoonees, “Understanding Background Mixture Models for Foreground Segmentation”, Proceedings Image and Vision Computing New Zealand 2002, pp. 266-271, November 2002.
- [20] W. Niu, J. Long, D. Han, Y. F. Wang, “Human Activity detection and recognition for video surveillance”, IEEE Conf. Multimedia and Expo, Vol. 1, No. 3, pp. 719-722, June 2004.
- [21] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen,“Efficient moving object segmentation algorithm using background registration technique” IEEE Trans. on Circuits and Systems for Video Technology, Vol. 12, pp. 577-586, July 2002.
- [22] Storm - Distributed and fault-tolerant realtime computation, <http://storm-project.net/>.
- [23] Xen Cloud Platform, <http://www.xen.org/products/cloudxen.html>
- [24] Hadoop, <http://hadoop.apache.org/>.
- [25] Yahoo S4, <http://incubator.apache.org/s4/>