



國立中山大學資訊工程學系

碩士論文

Department of Computer Science and Engineering

National Sun Yat-sen University

Master Thesis

具視野分界之攝影機協同監控排程設計與實作

Cooperative monitoring scheduling of cameras with splitting
vision and its implementation

研究生：陳定延

Ding-Yan Chen

指導教授：王友群 博士

Dr. You-Chiun Wang

中華民國 107 年 8 月

August 2018

摘要

無線視覺感測網路(Wireless visual sensor network)是在三維空間中分布多個具聯網功能的方向性視覺感測節點(Directional visual sensor)所組成，其具有不同視角以擷取動態影像來達到監控之目的，有別於傳統全向性的無線感測網路(Omnidirectional wireless sensor network)，有向性的視覺感測器更注重於位置佈署與監控排程等議題，特別來講如何在具扇形區域限制的感測範圍下，達到最佳監控排程將是一大挑戰。

傳統的監控攝影機因其固定位置的特性，在佈署上有著諸多的限制，而當系統中所需的監控目標增加時，勢必要調整位置並且重新部署，從而提高整體成本，也因此，本論文藉由具有水平與垂直旋轉及變焦能力的攝影機 (Pan-tilt-zoom camera)，在已知攝影機及監視目標位置的前提下，設計出一套協同監控(Cooperative monitoring)排程演算法，利用其視野(Field-of-view)重疊的特性，來讓攝影機在旋轉時所產生的監控死角能夠降到最低，此外，我們也考量以視野分界(Splitting Vision)的現象，以貼近現實使用者觀測攝影機的情況，並且達到量化實驗結果之目的，最後，我們也實作出雛型系統，以驗證所設計方法的可行性。

關鍵字：協同監控、PTZ 攝影機、監控排程、監控系統、無線視覺感測網路。

Abstract

A wireless visual sensor network is a network of spatially distributed smart camera devices capable of processing and fusing images of a scene from a variety of viewpoints into some form more useful than the individual images. Different from omnidirectional wireless sensor network, directional wireless visual sensor focus on location deployment and monitoring scheduling. In particular, how to achieve the best monitoring scheduling under the sensing range limitation will be a big challenge.

Because of the characteristics of fixed position, there are many restrictions when traditional surveillance system is deploying. It will increase overall cost for redeploying if a new target needs to be monitored by system. Therefore, under the premise that all camera knows the position of targets and each other, this paper designs a cooperative monitoring scheduling algorithm with pan-tilt-zoom camera. We make use of the feature of field of view overlapping and avoid detection dead angle when cameras rotate. Furthermore, for closing to the reality situation, we consider splitting vision of camera and quantify the experiment result. In the last, we establish the prototype of system to verify the feasibility of our algorithm.

Keywords: cooperative monitoring, PTZ camera, monitoring scheduling, surveillance system, wireless visual sensor network

目次

摘要	i
Abstract.....	ii
目次	iii
圖次	v
表次	vii
第一章 導論	1
1.1 前言	1
1.2 研究動機	2
1.3 研究貢獻與章節架構	3
第二章 背景知識探討	4
2.1 監控視野	4
2.2 視線遮蔽性	6
2.3 與目標的視野角度	6
第三章 相關文獻探討	8
3.1 全向性感測網路之部署方法	8
3.2 有向性感測網路之部署與排程方法	9
第四章 問題定義	11
4.1 待測環境與攝影機之假設	11
4.2 監控視野與視野分界(Split vision)之定義	12
4.3 攝影機監控排程之的線性規劃問題	15
第五章 研究方法	17
5.1 VSS 架構	17
5.2 VSS 演算法細節	18
第六章 模擬結果分析與實作	25

6.1	模擬環境與參數設定	25
6.2	模擬結果	27
6.2.1	待測目標數量之影響	27
6.2.2	視野分界角度 φ 之影響	32
6.2.3	分數權重 ω 之影響	34
6.3	系統離型實作	36
第七章	結論與未來研究方向	39
參考文獻	40

圖次

圖 1-1：以錄影帶作為儲存媒介的類比式攝影機監控系統.....	1
圖 1-2：以數位影像作為儲存媒介的類比式攝影機監控系統.....	2
圖 1-3：類比式攝影機結合磁碟儲存模式並以網路傳輸的監控系統.....	2
圖 1-4：以網路傳輸的攝影機監控系統.....	2
圖 2-1：3D FoV 示意圖	5
圖 2-2：與目標的視野角度之影響.....	7
圖 4-1：攝影機排程序列	11
圖 4-2：協同監控攝影機之範例	13
圖 4-3：視野分界區域權重示意圖	14
圖 4-1：VSS 演算法流程圖	17
圖 5-2：尋找錨點並分割視野之範例	19
圖 5-3： c_1 與 c_2 的視野分布	22
圖 5-4：進行步驟二與步驟四判定之流程圖	24
圖 6-1：待測物間距示意圖	25
圖 6-2：以 5 台攝影機與 20 個待測目標為範例的部署示意圖，以藍色三角形表示 攝影機，黑點表示目標，綠色區域為感測範圍	26
圖 6-3：SSF 與 VSS-C 的視野配置	27
圖 6-4：目標數量對排程總分的影響(攝影機數量為 10)	29
圖 6-5：目標數量對排程總分的影響(攝影機數量為 20)	29
圖 6-6：目標數量對排程總分的影響(攝影機數量為 30)	30
圖 6-7：目標數量對排程成功率的影響(攝影機數量為 10)	30
圖 6-8：目標數量對排程成功率的影響(攝影機數量為 20)	31
圖 6-9：目標數量對排程成功率的影響(攝影機數量為 30)	31
圖 6-10： φ 角與排程總分之影響	33

圖 6-11： φ 角與中心視野監控的比例之影響	33
圖 6-12：兩側區域權重對排程總分變化影響	34
圖 6-13：兩側區域權重對 VSS 與 VSS-C 的排程總分變化影響	35
圖 6-14：兩側區域權重對 VSS 與 SSF 的排程總分變化影響	35
圖 6-15：系統架構	36
圖 6-16：使用者介面	37
圖 6-17：系統實作示意圖	38
圖 6-18：實作排程結果	38

表次

表 2-1：135 片幅以計算視角之範例.....	5
表 4-1：符號定義與說明.....	16
表 5-1：步驟二與步驟三的處理過程範例.....	23

第一章 導論

1.1 前言

監控系統的發展已將近有幾十年的歷史，一個完整的監控系統除了由多個攝影機鏡頭(Camera)所組成，另外還包含電纜(Cable)、監視主機(Monitor)、以及儲存媒介(storage media)等裝置，而早期的監控系統多半以錄影帶(Video cassette recorder，即 VCR)來做為儲存媒介而形成的類比式攝影機監控系統(Analog closed-circuit television) [1][2]，其概念如圖 1-1 所示，該系統由前端的類比攝影鏡頭來負責記錄影像，而每一台攝影機都擁有自己的同軸電纜(Coaxial cable)以連接到 VCR 主機上，然而，由於儲存的媒介為錄影帶，必需時常更換，另外，錄影帶也容易受到周遭環境濕度與溫度之影響，若保存不慎將造成影像的模糊而失真。

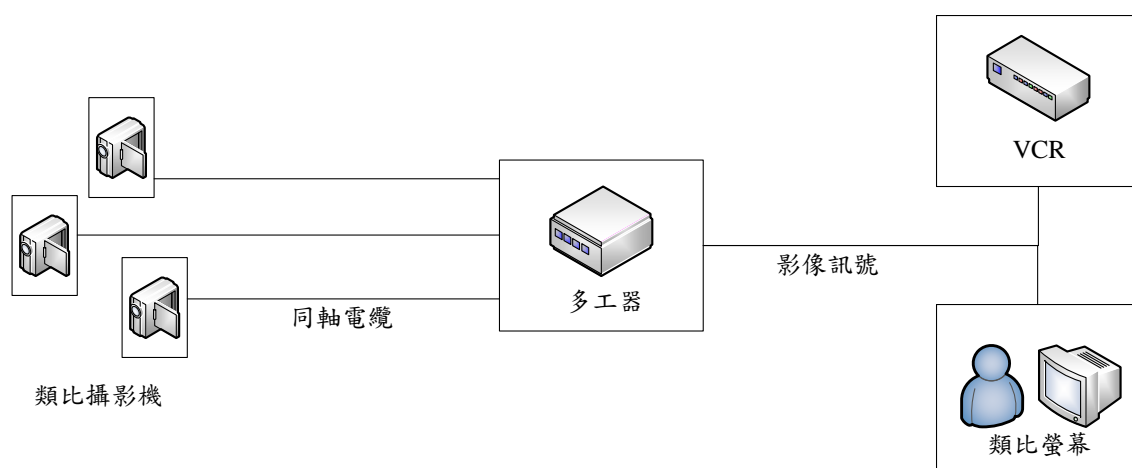


圖 1-1：以錄影帶作為儲存媒介的類比式攝影機監控系統

為了改善上述情況，1990 年代中期便發展出以數位影像(Digital video recorder，即 DVR)作為儲存媒介的類比式攝影機監控系統[3]，它以磁碟(Disk)來取代錄影帶以作為儲存媒介，其架構如圖 1-2 所示，這個時期的 DVR 仍透過類比螢幕來顯示影像，但是 DVR 本身已經能取得數位的影像資料，如此一來，藉由網路傳送影像也變得可行，於是發展出結合乙太網路(Ethernet)連接埠的 DVR 設備，如圖 1-3 所示，如此一來，接上網路交換器後便能在一般電腦上顯示攝影機的監控畫面。

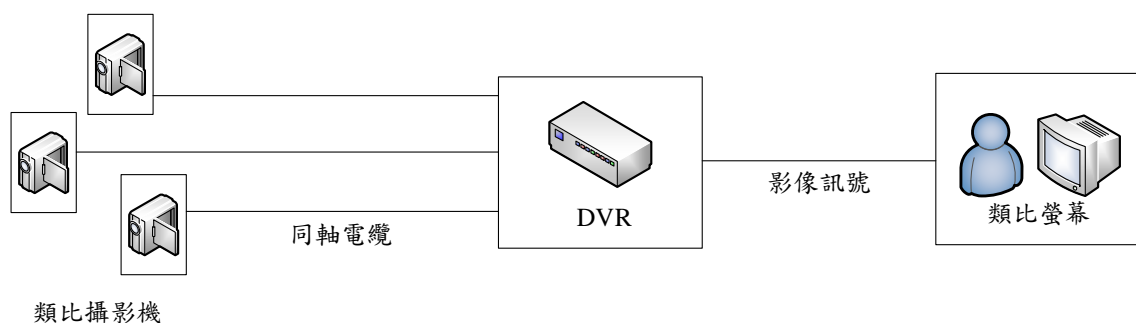


圖 1-2：以數位影像作為儲存媒介的類比式攝影機監控系統

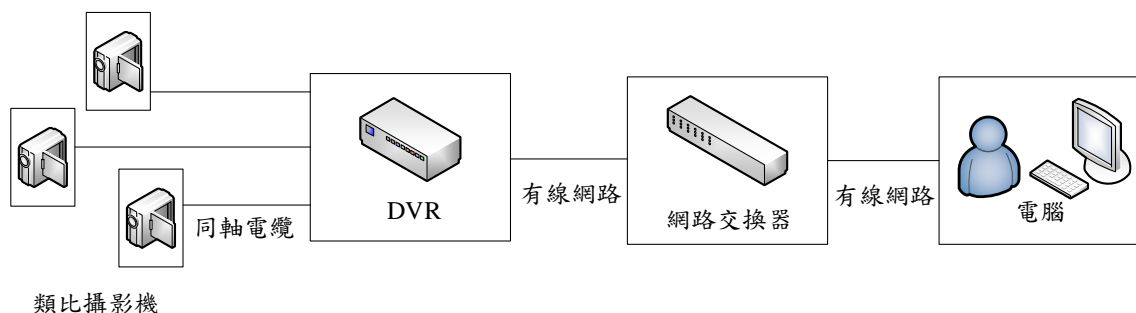


圖 1-3：類比式攝影機結合磁碟儲存模式並以網路傳輸的監控系統

現代的攝影機已不需要經由類比數位轉換器、多工器或放大器等其他設備幫助，它們可以透過有線或無線網路，直接傳輸影像至後端主機，如圖 1-4 所示，如此一來，不但可以大幅降低成本，影像品質也相對提升許多，也因此，近年來監控系統除了安全上的防護外，它也支援資料分析上的應用，諸如：事件偵測[4]、人臉辨識[5]、公共安全[6]、交通流量分析[7]，以及畜牧管理[8]等。

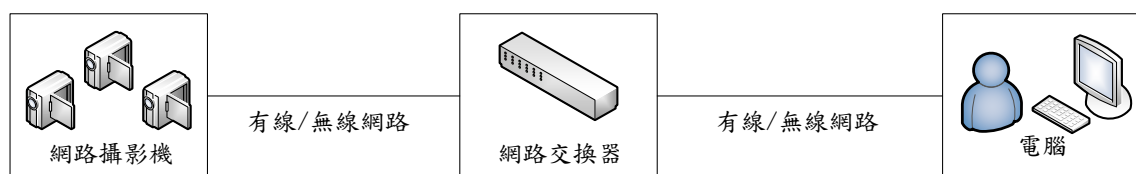


圖 1-4：以網路傳輸的攝影機監控系統

1.2 研究動機

隨著科技與社會的演進，人們對於生活的安全意識逐漸增加，依據馬斯洛的需求理論(Maslow's hierarchy of needs) [9]，當人們滿足基本的生理需求，例如：食物、水、空氣與睡眠後，接下來就需滿足人身的安全需求，也因此催生了監控系統，而隨著存取技術的發展，傳統的類比攝影機已演變成以網路傳輸影像訊號的視覺感

測器(Visual sensor)，這對於攝影機的佈署上有著更大的彈性，另外，我們也可以藉由網路傳輸，以便讓視訊資料更容易保存與備份。

一般常見的監視系統多半為定點監控(Fixed point monitoring)，也就是像路邊監視器般固定監視同一處，並且利用多台監視器，以不同視角來進行大範圍監控，然而，這種固定式監視器在使用上容易產生畫面死角，若是這些死角被有心人士加以利用，將會產生社會治安的風險；除此之外，倘若監控的目標或範圍增加時，勢必要增加攝影機數量或者更動位置來調整系統以符合實際需求，如此一來，將會耗費大量的資源成本，基於上述動機，本篇利用可旋轉的 PTZ 攝影機。以兼顧垂直與水平的視角，並藉以動態擴大監視範圍，這樣可以使用較少的攝影機。如何使用有限的可旋轉視覺感測器，以安排最佳的監控排程將是本論文所討論的研究重點。

1.3 研究貢獻與章節架構

基於前述動機，本論文在已知觀測物能被哪些攝影機偵測到的前提下，提出循環式監控排程策略，其中，攝影機有著固定的旋轉週期，且監測物都有最少的監測時間，此外，我們亦利用觀測物所在的視野位置來作為分數的劃分，以得分高低來比較排程策略的優劣性，總體而言本論文主要的貢獻如下：

- 攝影機能彼此協調並保持監控而不會有死角，此外，我們所提出的策略也能根據排程減少所需攝影機數量，從而降低佈建系統的成本。
- 我們考量人們在觀測的方式，並區分「中間區域」與「兩側區域」以設立成不同分數比例，如此一來，將更加貼近實際使用者在觀看監視器的情況。

本論文共分為七個章節：第二章會先講解攝影機的相關特性，第三章節將會討論視覺感測網路的相關文獻，接著，第四章將會說明問題定義與假設，而第五章會詳細介紹我們的排程策略，第六章則是系統實作與模擬實驗之分析，最後，我們於第六章總結本文並提出未來的研究方向。

第二章 背景知識探討

在三維空間(Three-dimensional space)的環境中，佈建監控系統時我們必須要考量攝影機的物理特性，以確保其監控品質並避免誤判的情形發生，因此，本章節將介紹這些特性，其中包含監控視野(Field of View, 即 FoV)、視線遮蔽性(Line of sight)以及目標視野角度(View-angle)。

2.1 監控視野

監控視野，簡單來講，就是使用者透過螢幕所能看到攝影機監控的範圍，其概念如圖 2-1 所示，當我們把攝影機鏡頭放置於三維中笛卡爾座標系統(Cartesian coordinate system)的原點 $O(0,0,0)$ ，並將攝影機的光軸(Optical axis)置於 Y 軸上，這會使得攝影機鏡頭的位置處於 $X=0$ 與 $Z=0$ 的向量上，而光軸即為攝影機視角的中心線，此時，攝影機的水平視角 θ_h (即圖 2-1 中的 $\angle AOB$)及垂直視角 θ_v (即圖 2-1 中的 $\angle BOC$)會影響 FoV 的寬廣程度，而這兩個角度又受到焦距(Focal length)的影響，當焦距越長時，其所產生的視角將越小，也就是 FoV 的範圍將越狹窄，反之，當焦距越短時，其所產生的視角將越大，而我們所能得到的 FoV 範圍就越寬廣，特別來講，我們可藉由以下公式來推算視角[10]：

$$\text{視角} = 2 \times \arctan\left(\frac{\text{片幅}}{2 \div \text{焦距}}\right) \times \frac{180}{\pi} \quad (1)$$

其中，焦距又會影響攝影機所能見到的清晰區域，舉例來講，表 2-1 為以 135 片幅(film)所計算出的焦距對應視角，其中，135 片幅是攝影中的一種底片格式[11]，自 1934 年推出後至今仍然廣泛的使用，因其普及性高，在這邊我們作為焦距與視角換算的範例。

我們所使用的 PTZ 攝影機，在其焦距最短時會產生一個可見區域所能辨別的最遠距離 R_{max} ，換句話說當某個待測物體的位置超過 R_{max} 時，將會造成該攝影機無法清晰辨識的狀況，也因此， θ_h 、 θ_v 與 R_{max} 這三項參數共同決定攝影機 FoV 的大小，其所產生的四角椎 OABCD(參考圖 2-1)即為攝影機在監控目標 O_2 產生的 FoV

之範圍。

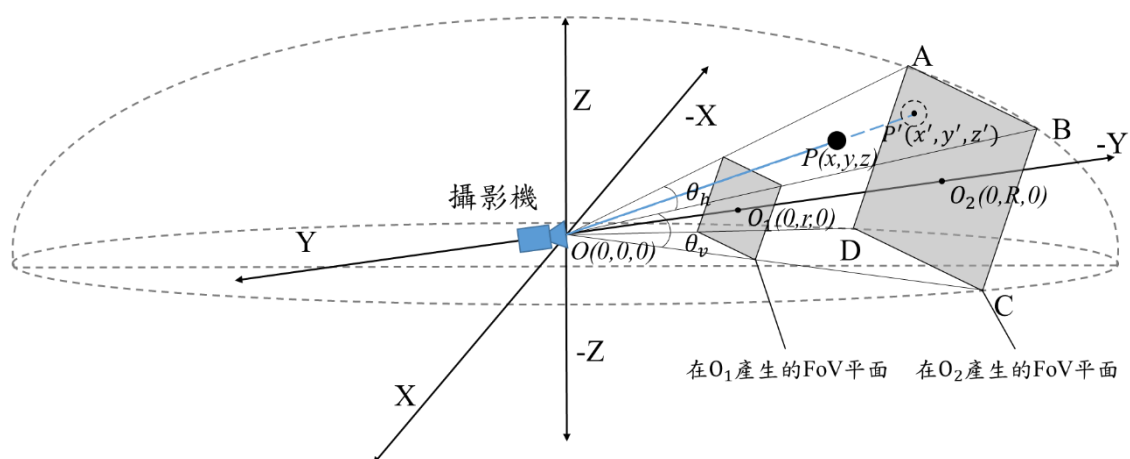


圖 2-1：3D FoV 示意圖

焦長(mm)	視角(度)	焦長(mm)	視角(度)	焦長(mm)	視角(度)
12	121.97	35	63.44	80	30.26
14	114.18	40	56.81	85	38.56
15	110.53	43	53.41	90	27.03
18	100.48	45	51.35	100	24.41
20	94.49	50	46.79	135	18.21
21	91.70	55	42.94	180	13.71
25	81.74	58	40.91	200	13.35
28	75.38	60	39.65	300	8.25
30	71.59	70	34.35	400	6.19
31	69.82	77	31.39	500	4.95

表 2-1：135 片幅以計算視角之範例

2.2 視線遮蔽性

視覺感測的成像是藉由光線所傳送，而光線在均勻介質(Uniform medium)的環境且不受其他重力場(Gravity field)干擾的前提下，擁有直線前進的特性，此外，光線亦無法穿越不透明的物體，因此，在光線傳遞的路徑上若存在障礙物，將會阻擋光線的前進，而造成攝影機無法偵測該目標，圖 2-1 給予一個範例，假設若有一個待測目標 $P'(x', y', z')$ 位於 ABCD 平面(即 O_2 的 FoV 範圍)上，此時若有一個不透明障礙物 $P(x, y, z)$ 位於座標 O 與 P' 連線上，此時會因為光線無法穿透該障礙物，而使得攝影機無法監控到目標 P' 。

2.3 與目標的視野角度

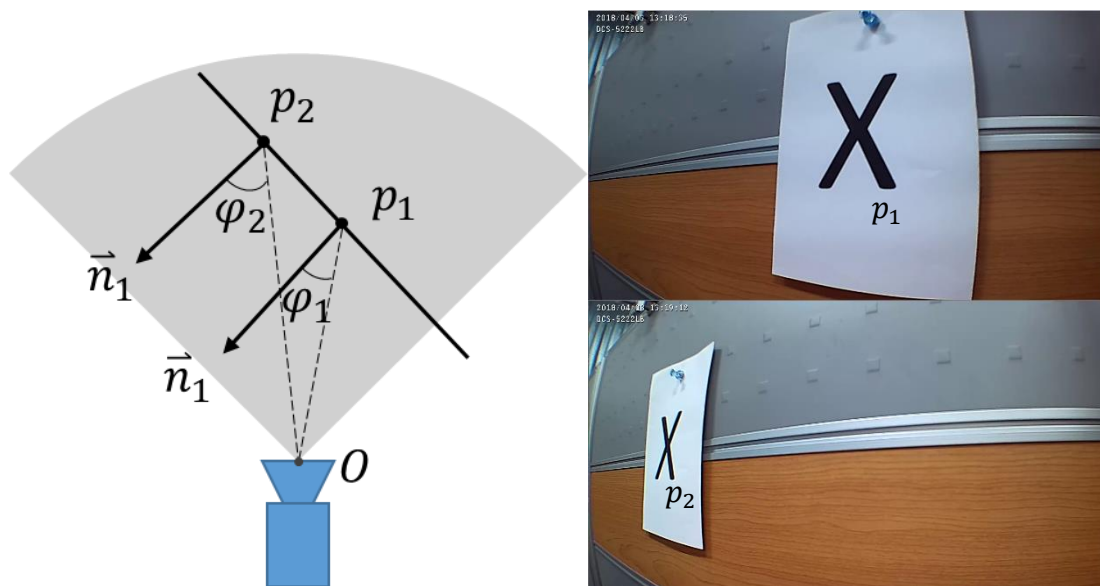
當攝影機監控某個目標時，其所呈現的影像會因為該目標所擺放的位置與方向，而產生不同的視覺角度(View angle)，此傾斜角度將會影響使用者辨別物體的清晰程度，其定義為物體正面之法向量(Normal vector)與攝影機和目標連線所形成之夾角，如圖 2-2 (a)所示，其中， p_1 與 p_2 分為不同位置的監控目標，而 $\overrightarrow{p_1O}$ 與 $\overrightarrow{p_2O}$ 則分別為 p_1 與 p_2 到位於原點 O 之攝影機所形成的向量，而 $\overrightarrow{n_1}$ 為監控目標所在平面之法向量， $|\overrightarrow{p_1O}|$ 、 $|\overrightarrow{p_2O}|$ 與 $|\overrightarrow{n_1}|$ 為向量長度，此時我們可以利用內積的公式(dot product)，來求出兩個向量之間的夾角：

$$\varphi_1 = \arccos\left(\frac{\overrightarrow{p_1O} \cdot \overrightarrow{n_1}}{|\overrightarrow{p_1O}| |\overrightarrow{n_1}|}\right) \quad (2)$$

$$\varphi_2 = \arccos\left(\frac{\overrightarrow{p_2O} \cdot \overrightarrow{n_1}}{|\overrightarrow{p_2O}| |\overrightarrow{n_1}|}\right) \quad (3)$$

因此，當監控目標與攝影機的傾角越大時，這表示該目標在影像中所顯示的面向角度也就越大，圖 2-2 (b)展示實際攝影機所拍攝的影像，這兩張影像分別對應到圖 2-2(a)的 p_1 與 p_2 之目標，而 p_2 的傾角由於較大，所以在辨識上會較 p_1 來的模糊，而同一個物體在不同攝影機上產生的監控角度不盡相同，可能會出現無法清晰辨識的情況，這可利用影像處理(Digital image processing)[12]或者機器學習(Machine

learning)能夠使得系統判別更加準確[13]。



(a)：不同位置的產生的視角

(b)：視角不同造成影像辨識上的差異

圖 2-2：與目標的視野角度之影響

第三章 相關文獻探討

對於視覺感測網路的研究來說，感測器的排程與覆蓋(Coverage)為重要的議題[14]，我們將會在本章節依序介紹全向性感測網路的部署方法，以及有向性感測網路的部署與排程方法。

3.1 全向性感測網路之部署方法

在全向性感測網路的部署研究中，文獻[15]在確保待測區域中的每個地點至少被 m 台感測器監測的條件下，如何安排兩種具不同成本與感測範圍的感測器進行部署，以使得整體系統成本花費達到最小；而文獻[16]則提出一個六邊形部署策略(hexagon-like deployment)，此部署方法可以讓感測區域滿足 k -coverage 需求(也就是待測區域的任何地方皆至少被 k 個感測節點所覆蓋)；文獻[17]則把感測區域分割成大小相同的子區域，並假設感測器節點能與周圍的節點彼此交換資料，以利用節點本身所偵測到的區域密度進行計算後再來移動節點，以使得所有區域密度達到平均，並且讓覆蓋面積最大化；此外，也有針對隨機分佈進行分析以符合系統需求的研究，例如，文獻[18]指出帕松隨機分布(Poisson distributed)無法滿足對於感測需求較高的熱點(hot spot)之服務質量(Quality of Service)，但高斯隨機分布(Gaussian distributed)又過於集中而使得遠處的覆蓋率在平均之下，因此[18]提出了混合式(Hybrid)隨機分布法，可以根據觀測節點位置及需求挑選上述兩種分布方式，以滿足最小覆蓋需求。

由於並非所有的應用都需要感測器的完全覆蓋(full coverage)，因此有部分文獻根據感測應用的需求提出不同的策略，舉例來講，文獻[19]為了增加感測網路的整體壽命，針對覆蓋率比較集中式(centralized)與分散式(distributed)兩種覆蓋演算法優劣；而文獻[20]則提出三維空間(Three-dimensional space)中可擴展之低延展性策略(scalable and low stretch routing scheme)，它對於不規則邊界之空間以切面(cut)的方式來取得最大感測範圍，再以貪婪式演算法(greedy algorithm)來部署感測器，這

種方式可適用於複雜的立體空間；另一方面，文獻[21]則是以部署空氣汙染偵測感測器為目標，將感測區域劃分為數個多邊形，並根據已知的汙染程度與部署成本來決定感測器位置和數量；此外，文獻[22]採用基於網格的網絡架構(Grid-based network architecture)，為了偵測入侵者的移動路徑，將感測區域劃分為長方形的子區域，再根據預測的路徑來部署感測器，以使得每個子區域的感測器密度達到適當的數值。

然而，以上研究方法皆屬於全向性感測網路，無法套用到本文所使用的視覺感測網路上，因為其感測模型與方向性感測網路不同，且全向性感測網路也不會因為時間改變而有不同的感測位置。

3.2 有向性感測網路之部署與排程方法

另一方面，針對指定感測方向的有向性感測網路部署之方法也有諸多研究，文獻[23]希望最小數量的攝影機，使得當入侵者進入待測區域時，感測網路的反應時間能夠最短，它採取最佳逼近放置法(Approximating optimal visual sensor placement)的策略[24]，並將攝影機的 FoV 視為三角形區域，在限制區域大小與攝影機角度後，以線性規劃(linear programming)的方式來部署覆蓋最大範圍的攝影機；而文獻[25]則針對不同區域所發生的事件定出不同的風險等級(risk level)，例如：區域是否偵測到入侵者，並根據此風險等級來安排攝影機的位置；文獻[26]則把監控區域分成數個子區域，並為每個子區域設定各自的最低覆蓋率，[26]在隨機部署攝影機後，即根據攝影機在不同角度給予在該子區域的覆蓋貢獻值，最後再依據貢獻值設定攝影機的方位；文獻[27]利用粒子群最佳化演算法(particle swarm optimization algorithm,即 PSO) [28]來解決攝影機最大化覆蓋區域的問題，它將所有感測器視為一個群體(Swarm)，而需要被偵測的物體視為粒子(particle)，在每一輪的運算會根據粒子在感測區域中的被覆蓋比例與粒子的速度來調整感測器的方向，以便達到覆蓋率的最大化；然而，上述文獻皆是針對固定方向的有向性感測器進行佈建或是將感測器部署到固定方向，而本論文所使用的 PTZ 攝影機則具有旋轉特性，相較

於上述方法能有更廣闊的覆蓋率。

部分文獻考慮可旋轉式感測器的排程議題，特別來講文獻[29]假設 PTZ 攝影機在動態移動時，其視野中的每一點都有著覆蓋機率，此外，待測物體也有覆蓋需求來以表示其監控優先度，在此假設中，[29]的目標是使得每個待測物體被感測器覆蓋機率可大於物體所給定的被監控機率，藉此來達到最有效率及最省成本的部署與排程；而文獻[30]採用賽局理論的概念，將每個感測器視為玩家(player)，在有限的能源下，找出在已佈建好的感測器中能夠達到最大覆蓋監控之排程；此外，文獻[31]先假設感測器的覆蓋為圓形，並藉以部署感測器以覆蓋所有目標，並利用簡易式扇形搜尋操作(simple sector finding)來找出每個感測器可以偵測目標的實際感測範圍，再將重複感測區域的部分移除；最後，文獻[32]則是將 PTZ 攝影機運用於運動場上，在廣大的運動場中針對感興趣的運動設施區域進行排程監控，並且將監控問題轉化為最大節點權重配對問題(maximum vertex weight matching)，其將攝影機與監控目標分成兩個獨立集合，並利用二分圖(bipartite graph)尋找監控所有物體所花費最少時間的配對(match)解答。

上述的文獻雖然考量到感測器具有旋轉的功能，卻未考慮到現實中感測物的位置對於感測範圍的影響，舉例來講，當物體出現於攝影機視野的邊緣時，雖然物體而位於感測範圍內，但是使用者卻不一定能夠仔細觀察待測物體，也因此，本論文將 PTZ 攝影機的視野劃分為兩側與中間區域，並且給予不同的權重值，在加入中間區域的考量後提出以中間區域為監控核心的監控排程演算法，以提升整體的監控品質。

第四章 問題定義

在本章節中，我們將會說明待測區域之環境與 PTZ 攝影機行為之假設，接著定義本論文所要研究的攝影機排程之問題。

4.1 待測環境與攝影機之假設

為了能夠有效地排程監控攝影機，我們採用具有水平與垂直旋轉功能，以及可變焦的 PTZ 攝影機來進行監控，而在監控的過程當中，個別攝影機會在已知的位置停留一至多個視野，同時根據給定的監控目標所需要的最小監控時間以及攝影機所能覆蓋的監控目標來建立協同排程。

我們假定在待測區域中已部署一組攝影機 $\hat{C} = \{c_0, c_1, c_2, \dots, c_n\}$ ，其中每台攝影機 $c_i \in \hat{C}$ 擁有相同的工作週期 T (包含旋轉與監控物品)，並且可以自身為中心進行 360° 旋轉，而攝影機在單一週期中因切換至不同視野產生轉動總時間為 δ ，也因此，我們可以用公式(4)來表示 c_i 於週期 T 的行為(如圖 4-1 所示)：

$$T = \delta + T' \geq \delta + \sum \{t_{i,k}^{cam} \mid t_{i,k}^{cam} \text{ 為 } c_i \text{ 在視野 } f_{i,k} \text{ 在停留時間, } \forall f_{i,k} \in \hat{F}_i\} \quad (4)$$

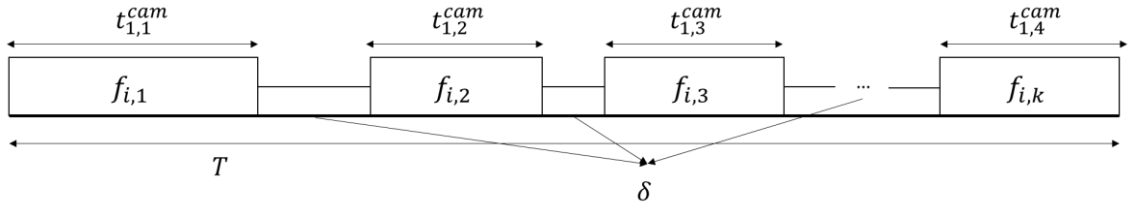


圖 4-1：攝影機排程序列

其中 T' 為 c_i 可用於監控物品的停留時間，而 \hat{F}_i 為 c_i 的視野之集合；值得注意的是，由於旋轉角度為 360° ，所有攝影機的轉動時間 δ 將為一個固定常數，假設攝影機的旋轉角速度為 V_A ，則 $\delta = 2\pi/V_A$ 。

由於 PTZ 攝影機在旋轉的過程需有一定的停留時間，才能讓攝影機捕捉到清晰的影像以便辨識物體[33]，另一方面，在文獻[34]中也有提到在攝影機監控排程時，對於兩個目標切換的時間要有最大的上限值，以避免目標不在攝影機監控範圍內的時間過長。基於上述的考量，我們假定待測環境中在一組監控目標物的集合 \hat{O} ，

此外，對於每項目標物 $o_j \in \hat{O}$ 我們也設定一個最小所需的監控時間(minimum monitoring time) t_j^{obj} ，也因此，假設有一子集合 $\hat{O}' \subseteq \hat{O}$ 的目標物可以被一子集合 $\hat{C}' \subseteq \hat{C}$ 的攝影機所涵蓋，則我們必須確保以下公式的成立：

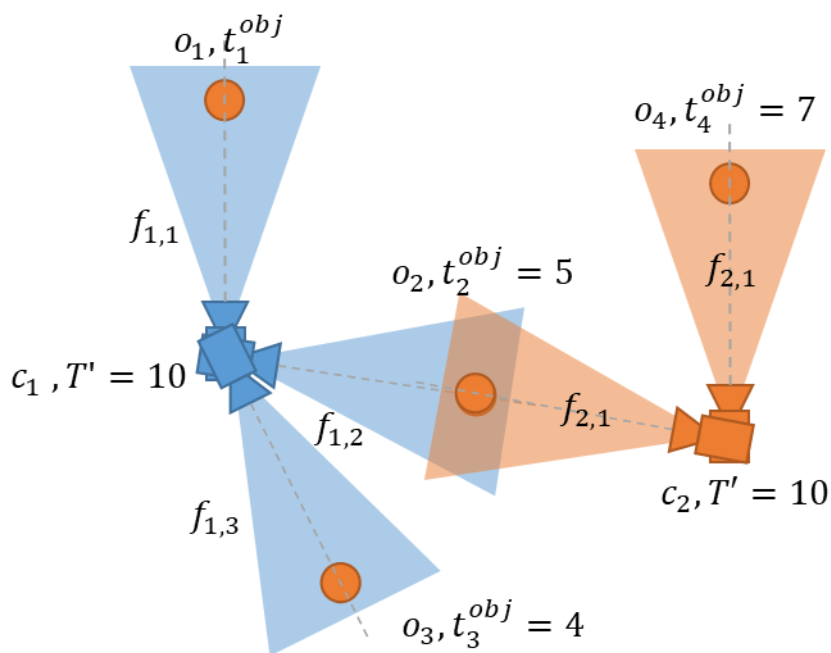
$$\sum_{c_i \in \hat{C}'} T' \geq \sum_{o_j \in \hat{O}'} t_j^{obj} \quad (5)$$

換言之，就是 \hat{C}' 集合中的攝影機之監控時間總和至少不小於 \hat{O}' 集合中的目標物所需的最小監控時間之總和，不然會有目標物無法滿足其 t_j^{obj} 之限制。

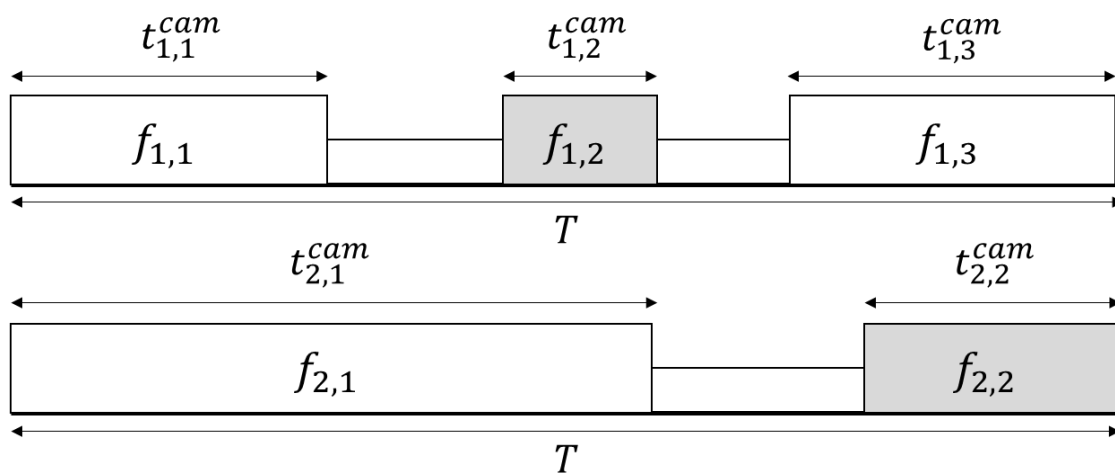
4.2 監控視野與視野分界(Split vision)之定義

由於攝影機可以旋轉，我們將其監控視野(monitored region) 定義為以攝影機光軸為中心所展開角度 θ 的扇形區域，並以變數 $f_{i,k}$ 來表示攝影機 c_i 的第 k 個監控視野，而其所形成的集合則以 \hat{F}_i 表示，另外，所有攝影機擁有相同的 θ 角度，也因此它們的視野大小是相同的。

當環境中具有多台攝影機時，可能會有部分視野相互重疊，我們可藉由此一特性來錯開重疊區域的監控時間，以達到分攤目標最小監控時間的效果[35]，以圖 4-2 (a)為例，攝影機 c_1 與 c_2 雖然都能監控到目標 o_2 ，由於 c_1 所負責的目標之最低監控時間總和超過 c_1 的週期，因此，我們可以利用 c_2 在週期內的時間接續監控目標 o_2 ，如同圖 4-2 (b)的排程，以保障所有目標皆能滿足其最小監控時間的需求。



(a) 攝影機 c_1 與 c_2 的監控視野



(b) 攝影機 c_1 與 c_2 的週期規劃

圖 4-2：協同監控攝影機之範例

另一方面，當監控目標座落於攝影機監控視野的邊界或中心時，其影像品質將會有所不同，為了反映這個現象，我們提出視野分界的概念，並以攝影機光軸為中心展開夾角 φ (其中 $\varphi < \theta$)，如圖 4-3 所示，將視野區分成三等分，並給予三個區域相對應的權重值，特別來講，當待測物體位於視野的夾角 φ 中時，其權重

為 1，而當待測物位於 φ 夾角之外的區域時，其權重為 $\omega(0 \leq \omega \leq 1)$ ，如此一來，我們以公式(6)表示權重的設定：

$$\omega = \begin{cases} 1, & \text{if } |\arccos\left(\frac{\vec{c_i o_j} \cdot \vec{v}}{|\vec{c_i o_j}| |\vec{v}|}\right)| \leq \frac{\varphi}{2} \\ \alpha, & \text{if } \frac{\varphi}{2} < |\arccos\left(\frac{\vec{c_i o_j} \cdot \vec{v}}{|\vec{c_i o_j}| |\vec{v}|}\right)| \leq \frac{\theta}{2}, 0 < \alpha < 1 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

我們以圖 4-3 為例，其中， \vec{v} 為攝影機 c_i 的光軸向量，而目標物 o_1 與 \vec{v} 的夾角小於 $\frac{\varphi}{2}$ ，所以其權重為 1，而目標物 o_2 與 \vec{v} 的夾角介於 $\frac{\varphi}{2}$ 至 $\frac{\theta}{2}$ 間，所以其權重值可設為介於[0,1]間之實數，此視野分界的權重將會運用於後續章節，並作為量化模擬結果之用。

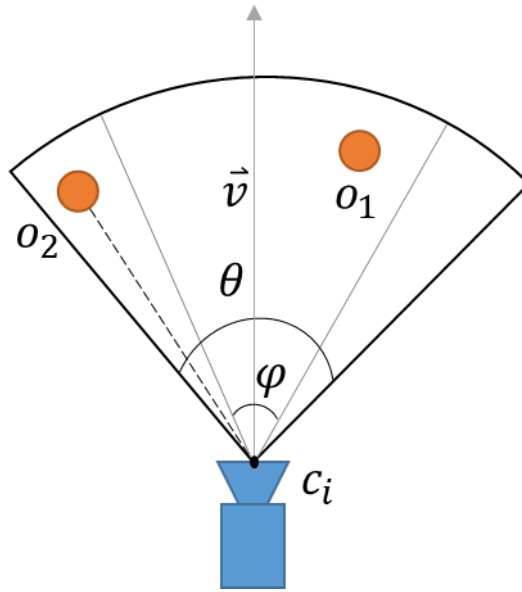


圖 4-3：視野分界區域權重示意圖

4.3 攝影機監控排程之的線性規劃問題

基於前兩子章節的環境假設，我們可以量化每台攝影機監控目標物的效能，特別來講，倘若一台攝影機 c_i 在某個視野 $f_{i,k}$ 停留 $t_{i,k}^{cam}$ 的時間，則我們可以計算 c_i 於 $f_{i,k}$ 所得到的分數為：

$$\sum_{o_j \in \hat{O}} \omega_{i,j} t_{i,k}^{cam} \quad (7)$$

其中， $\omega_{i,j}$ 為目標物 o_j 對於視野 c_i 的權重值，這可由公式(6)所求得，也因此，我們可以計算在每個週期中，所有攝影機所得到的分數總和：

$$P = \sum_{c_i \in \hat{C}} \sum_{f_{i,k} \in \hat{F}} t_{i,k}^{cam} \sum_{o_j \in \hat{O}} \omega_j \quad (8)$$

藉由將監控量化為分數，我們可以透過以下的線性規劃(linear programming)來描述攝影機監控排程之問題：

$$\text{目標：Maximize } P \quad (9)$$

限制：

$$t_j^{obj} > 0, \forall o_j \in \hat{O} \quad (10)$$

$$t_{i,k}^{cam} > 0, \forall f_{i,k} \in \hat{F}_i, \forall c_i \in \hat{C} \quad (11)$$

$$\hat{C} \neq \emptyset, \hat{O} \neq \emptyset \quad (12)$$

$$T = \delta + T' \geq \delta + \sum \{t_{i,k}^{cam} | t_{i,k}^{cam} \text{ 為 } c_i \text{ 在 } f_{i,k} \text{ 在停留時間}\} \quad (13)$$

$$\sum_{c_i \in \hat{C}} T' \geq \sum_{o_j \in \hat{O}} t_j^{obj} \quad (14)$$

顯而易見的，我們的目標是希望得到最高的監控分數(也就是公式(9)所表達的意義)，而公式(10)則表示每個目標都具有不為零的最小監控時間之需求，公式(11)則表示每台攝影機都必須在其每個監控視野停留正數的時間，公式(12)則代表攝影機與目標物的集合不為空集合，而公式(13)則反映攝影機的旋轉週期(也就是公式

(4)所表達之意義)，最後公式(15)即為公式(5)，用來滿足目標物的最小監控時間之需求；表 4-1 列出本論文常見之符號與其說明。

符號	意義
\hat{C}	所有攝影機之集合
T	攝影機的旋轉週期
\hat{F}_i	攝影機 c_i 之視野集合，其第 k 個視野為 $f_{i,k}$
\hat{O}	所有感測目標物之集合
$t_{i,k}^{cam}$	攝影機 c_i 在視野 $f_{i,k}$ 所停留的時間
δ	攝影機在單一週期中轉動切換視野時所花費的總時間
θ	攝影機的視角
φ	攝影機的視野分界之角度
ω	物體於 FoV 內的權重

表 4-1：符號定義與說明

第五章 研究方法

本論文所提出的方法，目標在於讓監控系統排成監控時能有更好的監控品質。

在定義出視野區域下，找出最適當的監控排程，並且符合公式(10)~(14)的限制。以下我們將會介紹考量視野分界情況的排程演算法(Vision Splitting Scheduling，即VSS)。

5.1 VSS 架構

在給定攝影機集合 \hat{C} 與待測目標集合 \hat{O} 的環境下，根據攝影機的視角，我們先切割出所有攝影機對於可偵測目標的視野 \hat{F}_i ，接著，將攝影機的時間分配給只被一台攝影機偵測到的目標，以確保所有物體都能被滿足最小時間，最後再根據最小監控時間由大至小分配，流程圖如圖 5-1 所示，演算法的細節將在 5.2 節中介紹。

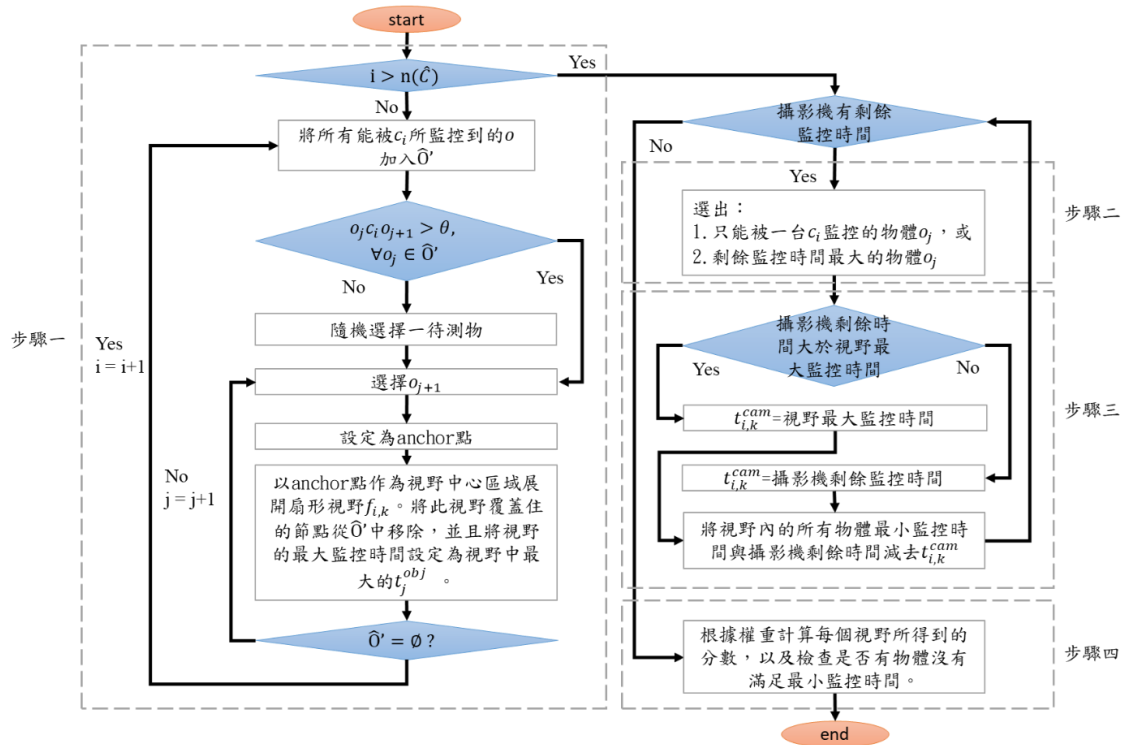


圖 4-1：VSS 演算法流程圖

5.2 VSS 演算法細節

- 步驟一：尋找視野中的錨點，並根據錨點劃分視野

根據[31]所提到的 SSF 演算法中，已經證明利用 SSF 能夠以最少的視野覆蓋範圍內所有物體。我們在每一次的迭代(iteration)中，將所有 c_i 可以偵測到的目標給加入 \hat{O}' ，再由 \hat{O}' 隨機挑選出一個目標並且編號為 o_j ，接著，以 c_i 為圓心， c_i 之感測距離為半徑，逆時針由 o_j 找尋下一個未被編號的目標標示為 o_{j+1} 如果角度 $\angle o_j c_i o_{j+1}$ 則將 o_{j+1} 標示為錨點(anchor)，否則由 o_{j+1} 繼續尋找下一個未被編號的目標，直到找到 $\angle o_j c_i o_{j+1}$ 大於 θ 時，將 o_{j+1} 設定為錨點，若無法在範圍內找到錨點，則隨機指定一個 o_j 作為錨點。

我們將錨點與 c_i 連線，向順時針展開角度 $(\theta - \varphi)/2$ 的扇形，以及向逆時針展開角度 $(\theta + \varphi)/2$ 的扇形，形成視野 $f_{i,k}$ ，並且將這個視野所覆蓋的目標物從 \hat{O}' 中移除，以避免再次被選成錨點。最後檢查 \hat{O}' 是否為空集合，如果不為空，表示範圍內還有目標沒有被標示，須要繼續重複步驟一，尋找下一個錨點並劃分視野，反之則換下一台攝影機。直到所有目標都被加入群組後進入步驟二。

接著我們透過範例來講解步驟一，在圖 5-2(a)中已部署了兩台攝影機 c_1 與六個監控目標。從所有目標隨機挑選出 o_1 後逆時針尋找 o_2 ，並且檢查 $\angle o_1 c_1 o_2$ 是否大於視角 θ ，如圖 5-2(b)，在這個例子中 $\angle o_1 c_1 o_2$ 小於視角 θ ，因此再由 o_2 開始逆時針尋找 o_3 ，如圖 5-2(c)，我們發現 $\angle o_2 c_1 o_3$ 大於視角 θ ，因此 o_3 被選為第一個錨點，根據公式(6)所提到的視野分界，向順時針展開角度 $(\theta - \varphi)/2$ 的扇形與逆時針展開角度 $(\theta + \varphi)/2$ 的扇形後形成第一個視野 $f_{1,1}$ ，再將視野底下的 o_3 和 o_4 由 \hat{O}' 中移除。若 t_3^{obj} 為 4 秒、 t_4^{obj} 為 6 秒，則此視野 $f_{1,1}$ 的 $t_{1,1}^{max}$ 即為 t_4^{obj} 也就是 6 秒。

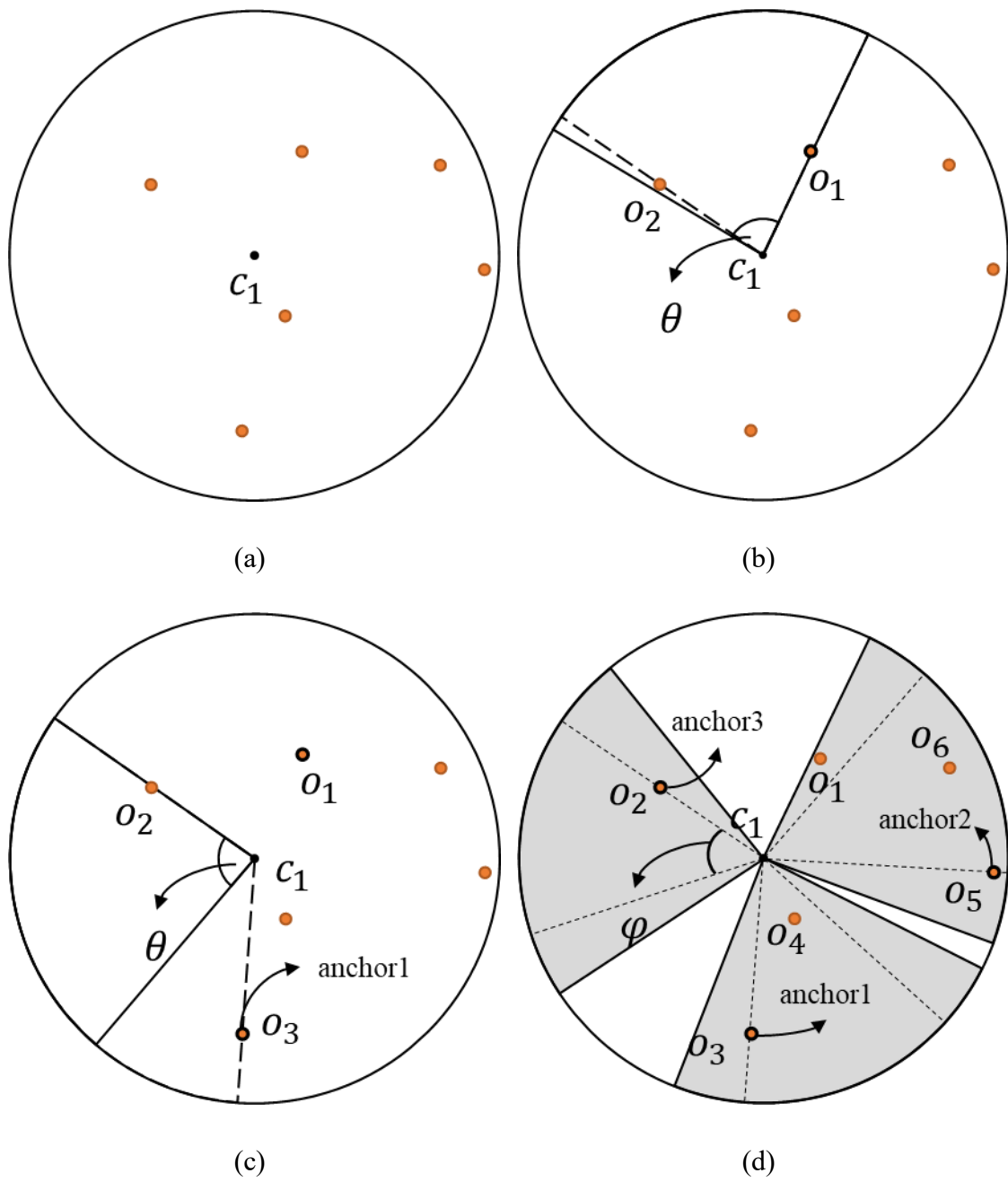


圖 5-2：尋找錨點並分割視野之範例

● 步驟二：選擇目標並設定監控時間

從步驟一中可以知道攝影機監控的位置與視野最大需求監控時間，接下來，我們將依序挑選適當的目標來分配監控時間。不同於[31]，為了盡可能滿足系統內所有目標的最小監控時間，我們有著以下的挑選順序：

1. 只被一台攝影機所監控的目標
2. 剩餘監控時間最大的目標

我們優先選擇只被一台攝影機所監控的目標，否則，當這類目標被分配時間之前，攝影機的剩餘時間將會不足以滿足該目標的最小監控時間，如此一來便違反了公式(14)，視為一個失敗的排程。若是所有的目標皆被兩台以上的攝影機所監控，則挑選剩餘監控時間最大的目標。在這裡我們假設 t_j^{remain} 為目標的剩餘監控時間，初始狀態 t_j^{remain} 等於 t_j^{obj} ，而隨著排程的進行，每當 o_j 再接下來的步驟三中被視野覆蓋並且減少監控時間時， t_j^{remain} 將會減去該次排程所覆蓋的時間。

選定完被排程的物體後，接著挑選能監控到該物體的攝影機中剩餘週期最大的攝影機。假設攝影機的剩餘週期為 T_i^{remain} ，同樣地在初始狀態 T_i^{remain} 等於 T_i' ，並且隨著步驟三分配視野監控時間時減去 $t_{i,k}^{cam}$ 。

● 步驟三：決定視野監控時間 $t_{i,k}^{cam}$

在步驟三中，我們將要決定每個視野監控時間 $t_{i,k}^{cam}$ ，從步驟二中，我們已經知道監控目標 o_j 、攝影機 c_i 與視野 $f_{i,k}$ ，然而 $f_{i,k}$ 中並非只有覆蓋 o_j ，所以我們也必須考量同一個視野下的其他目標。假設 \hat{O}' 為 $f_{i,k}$ 視野中目標物的集合， $t_{i,k}^{max}$ 為視野最長監控時間，其定義為 $t_{i,k}^{max} = \max(t_j^{remain})$ ，也就是所有視野下的目標中，剩餘的最長監控時間。當攝影機剩餘週期 T_i^{remain} 大於或等於 $t_{i,k}^{max}$ 時，視野的監控時間 $t_{i,k}^{cam}$ 即為 $t_{i,k}^{max}$ ，並且將攝影機的剩餘監控時間 T_i^{remain} 減去 $t_{i,k}^{max}$ ；反之，若攝影機剩餘週期 T_i^{remain} 小於 $t_{i,k}^{max}$ ，視野的監控時間 $t_{i,k}^{cam}$ 即為 T_i^{remain} ，我們把 T_i^{remain} 設為 0，表示這台攝影機將剩餘的監控時間都安排於這個視野，之後不再納入後續排程選擇考量。設定完 $t_{i,k}^{cam}$ 後，視野底下的物體 $o_j \in \hat{O}'$ 各自的剩餘時間 t_j^{remain} 也要減去 $t_{i,k}^{cam}$ ，如果 $t_{i,k}^{cam}$ 大於 t_j^{remain} ，則 t_j^{remain} 將會被設為 0，如此一來，下一輪的排程將不會考量此監控目標。我們用一小段虛擬碼來表示這一連串的段操作：

```

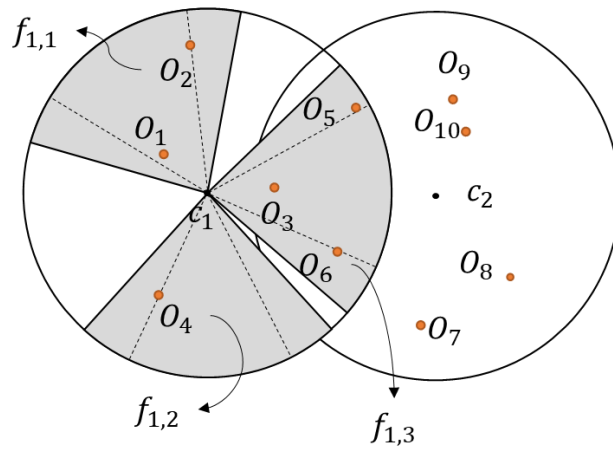
if  $T_i^{remain} > t_{i,k}^{max}$ 
     $t_{i,k}^{cam} = t_{i,k}^{max}$ 
     $T_i^{remain} = T_i^{remain} - t_{i,k}^{max}$ 
else
     $t_{i,k}^{cam} = T_i^{remain}$ 
     $T_i^{remain} = 0$ 

for  $o_j$  in  $\hat{O}'$ 
    if  $t_j^{remain} > t_{i,k}^{cam}$ 
         $t_j^{remain} = t_j^{remain} - t_{i,k}^{cam}$ 
    else
         $t_j^{remain} = 0$ 
end

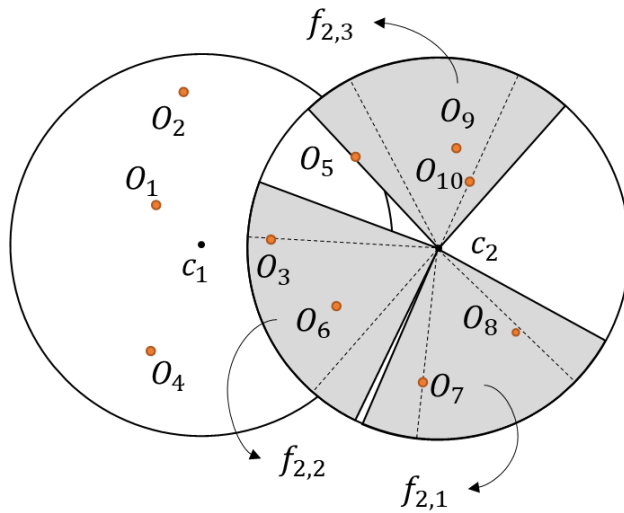
```

最後，我們以圖 5-3 和表 5-1 為範例說明步驟二與步驟三的流程，圖 5-3(a)與(b)為 c_1 和 c_2 視野的位置。表 5-1(a)顯示了各個物體剩餘時間的變化，假設環境中有十個目標物 $o_1 \sim o_{10}$ ，表 5-2(b)顯示了每一輪所選擇的攝影機與視野，表 5-2(c)則是攝影機對於目標的監控時間，假設兩台攝影機 c_1 與 c_2 週期皆為 20 秒，而 c_1 可以監

控到 o_1 、 o_2 、 o_3 、 o_4 、 o_5 、 o_6 ， c_2 可以監控到 o_3 、 o_5 、 o_6 、 o_7 、 o_8 、 o_9 、 o_{10} ，第一輪至第四輪都是選出了只被一台攝影機感測的物體 o_1 、 o_2 、 o_4 、 o_7 、 o_8 、 o_9 、 o_{10} ，由於 o_1 和 o_2 、 o_7 和 o_8 、 o_9 和 o_{10} 分別屬於同一個視野，所以在監控時直接取視野中最大的監控時間，值得注意的是， o_5 也和 o_9 與 o_{10} 屬於同一個視野，因此再第四輪同時也監控了 o_5 7 秒，再經過四輪排程後，剩下 o_3 和 o_6 ，取了剩餘時間較大的 o_6 與剩餘週期較大的 c_1 ，然而， c_1 所剩餘週期只剩下 5 秒，因此還需要第六輪以 c_2 來監控 o_6 ，如此一來，系統內的目標都滿足了最小監控時間。



(a)



(b)

圖 5-3： c_1 與 c_2 的視野分布

初始狀態		第一輪		第二輪		第三輪		第四輪		第五輪		第六輪	
目標	t_j^{remain}	t_j^{remain}		t_j^{remain}		t_j^{remain}		t_j^{remain}		t_j^{remain}		t_j^{remain}	
o_1	5	$f_{1,1}$	0	-	0	-	0	-	0	-	0	-	0
o_2	3	$f_{1,1}$	0	-	0	-	0	-	0	-	0	-	0
o_3	4	-	4	-	4	-	4	-	4	$f_{1,3}$	0	$f_{2,2}$	0
o_4	10	-	10	$f_{1,2}$	0	-	0	-	0	-	0	-	0
o_5	3	-	3	-	3	-	3	$f_{2,3}$	0	-	0	-	0
o_6	9	-	9	-	9	-	9	-	9	$f_{1,3}$	4	$f_{2,2}$	0
o_7	6	-	6	-	6	$f_{2,1}$	0	-	0	-	0	-	0
o_8	9	-	9	-	9	$f_{2,1}$	0	-	0	-	0	-	0
o_9	7	-	7	-	7	-	7	$f_{2,3}$	0	-	0	-	0
o_{10}	3	-	3	-	3	-	3	$f_{2,3}$	0	-	0	-	0

(a)目標物之剩餘時間變化

初始狀態		第一輪		第二輪		第三輪		第四輪		第五輪		第六輪	
攝影機	T_i^{remain}	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$
c_1	20	$f_{1,1}$	5	$f_{1,2}$	10	-	-	-	-	$f_{1,3}$	5	-	-
c_2	20	-	-	-	-	$f_{2,1}$	9	$f_{2,3}$	7	-	-	$f_{2,2}$	4

(b)每一輪所選擇的攝影機與視野

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
c_1	5	5	5	10	5	5	×	×	×	×
c_2	×	×	4	×	7	4	9	9	7	7

(c)攝影機偵測各目標的時間

表 5-1：步驟二與步驟三的处理過程範例

● 步驟四：根據權重與排程結果計算分數

在進入步驟四之前，我們要先檢查是否所有攝影機的時間都已經分配完畢，或者所有待測目標皆以滿足最小監控時間，如圖 5-4 所示，需要達成兩項條件才會進行步驟四結算分數，如此一來，能夠同時檢查本次排程是否滿足公式(4)，否則將回到步驟二繼續進行排程。

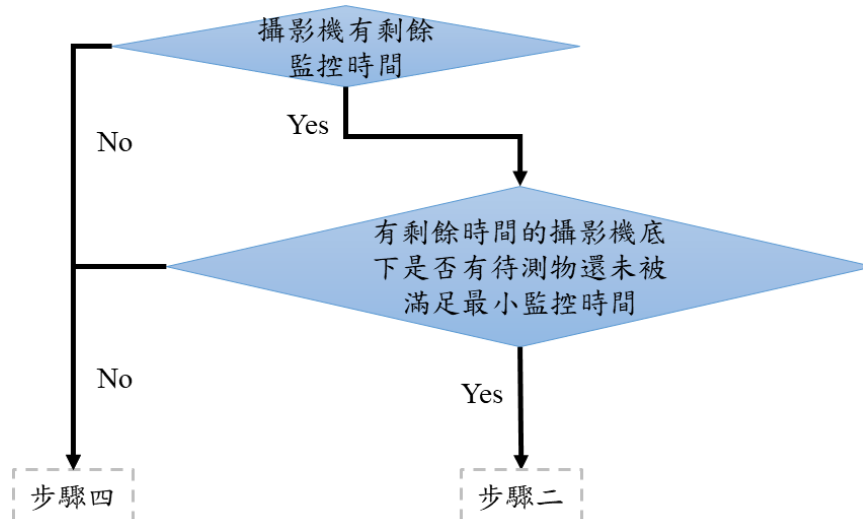


圖 5-4：進行步驟二與步驟四判定之流程圖

在步驟四中我們將計算 VSS 演算法所得到的監控分數，由圖 5-4 的判斷方式會出現三種情況：

1. 攝影機已無剩餘週期，且所有物體皆已滿足最小監控時間。
2. 攝影機尚有週期，且所有物體皆已滿足最小監控時間。
3. 攝影機已無剩餘週期，但仍有物體未滿足最小監控時間。

情況 1 為最佳的情況，即攝影機剛好滿足所有物體的排程，可以直接進行計算，情況 2 發生時，我們將攝影機多餘的時間拿來監控權重最大的視野，再進行計算，當情況 3 發生時，因為已經沒有多餘的時間可以監控，再進行計算時也必須要記錄這是一次未分配完全的排程。最後，根據公式(6)所提及之權重判定，我們可以將每一個視野下的目標權重相加再乘以視野的監控時間，即以公式(8)的方式計算。

第六章 模擬結果分析與實作

本章節將會介紹模擬環境及參數設定，並且比較在不同環境參數設置下，比較 SSF、VSS 以及將錨點設為中心光軸的 VSS-C 這三種演算法的結果，並且根據結果進行分析，最後再介紹以演算法為雛形設計的 PTZ 攝影機監控系統。

6.1 模擬環境與參數設定

在本論文中，我們以 Python 腳本語言作為模擬環境的建置與演算法開發工具，模擬環境為邊長 400 的正方形區域。為了讓環境能夠均勻分布攝影機與待測物，我們利用 Poisson-Disk Sampling Algorithm [36]來進行部署。設定環境時，我們先部署攝影機，讓每一台攝影機的間距保持在感測半徑至感測半徑的一半間($R \sim R/2$)，如此一來能夠避免攝影機之間視野重疊太高，也較符合實際情況，同樣地，為了避免待測物過於集中，我們希望待測物之間的間距能隨著攝影機的視角改變，因此，部署待測物時以攝影機的感測視角 θ 展開半徑為 R 的扇形，在其中畫出相切於兩條半徑的圓，如公式(15)與圖 6-1，我們便可以由三角函數算出該圓形的半徑 r ：

$$r = \frac{R \sin \frac{\theta}{2}}{1 + \sin \frac{\theta}{2}} \quad (15)$$

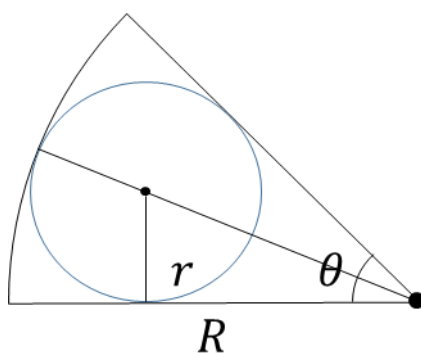


圖 6-1：待測物間距示意圖

以半徑為待測目標的間距，並且將部署範圍設定為所有攝影機的感測範圍，如果目標數量多到無法以此間距進行部署時再除去此間距之限制進行部署，部署完成後如圖 6-2 所示。其中綠色的點為攝影機，黑色的點為待測物體，綠色圓型表示攝影

機的監控範圍。我們設定了所有攝影機的感測週期為 10 秒，因此待測目標的最小監控時間總和必須小於 $10 \times$ 攝影機之數量，並且對每一組數據進行 100 次測量，以平均結果作為圖表進行分析。

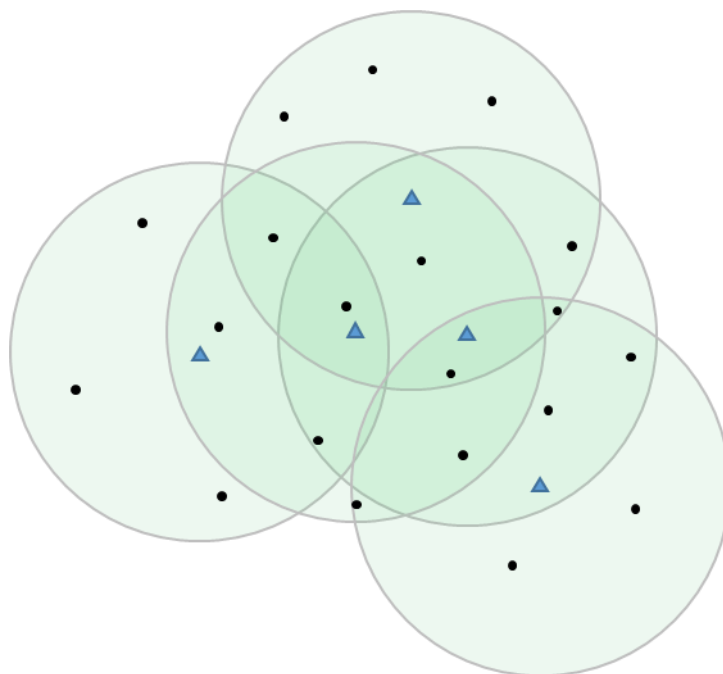


圖 6-2：以 5 台攝影機與 20 個待測目標為範例的部署示意圖，以藍色三角形表示攝影機，黑點表示目標，綠色區域為感測範圍

在本篇論文中，我們所比較的對象為文獻[31]提到的 SSF 演算法與 VSS-C，SSF 主要的目的為以最少的扇形找出感測範圍內能涵蓋所有目標的解，因此，在設置視野的步驟與 VSS 相同，差別在於找到錨點後 SSF 會直接以展開 θ 角的視野後找尋下一個錨點，而 VSS 展開角度則是 $(\theta + \varphi)/2$ ，如圖 6-3(a)所示。另一方面，文獻[31]對於監控時間的分配為權重(priority)高的目標優先，為了符合本論文的环境模型，此方法的排程設定為滿足視野內的最小監控時間為優先。至於，VSS-C 是將 VSS 方法中，錨點位置對其攝影機視野的光軸，如圖 6-3(b)所示，其排程方法則是與 VSS 相同。

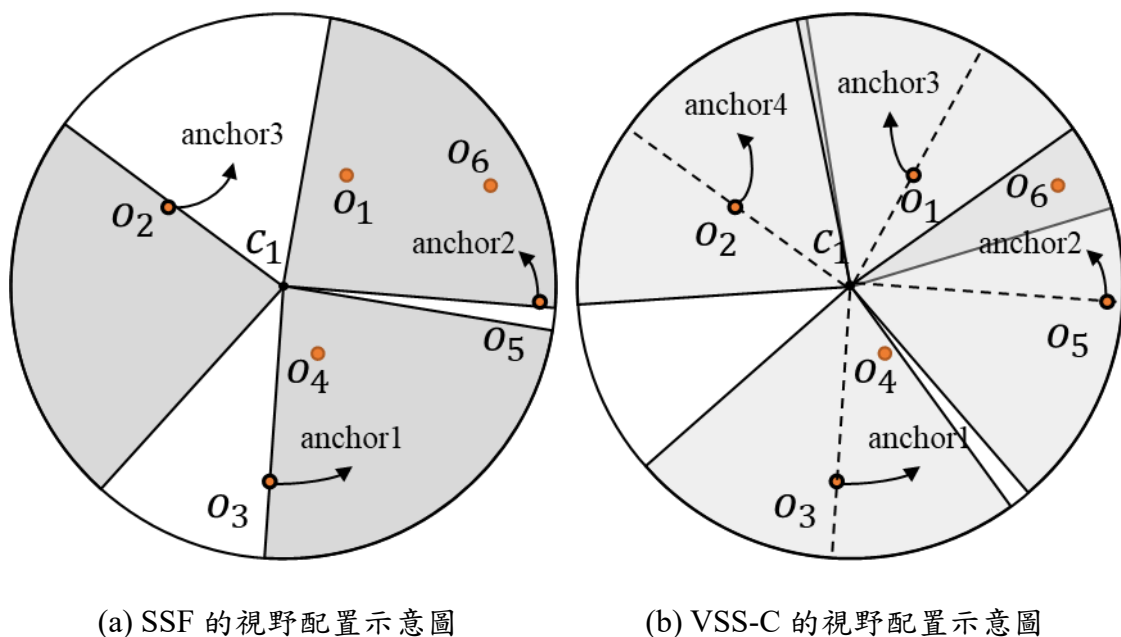


圖 6-3：SSF 與 VSS-C 的視野配置

6.2 模擬結果

6.2.1 待測目標數量之影響

我們先測量不同監控目標數量對於排程分數與排程失敗的比率所產生的影響，視野角度設定為 60 度，中心視野的部分則是 20 度，並且，根據公式(4)，我們的感測器最大數量為攝影機數量 \times 監控週期，也就是待測目標數量最大時，所有目標最小監控時間都是 1 秒的情況，計算結果時，位於視野中心區域的 ω 為 1，兩側邊緣區域的 ω 設定為 0.3。為了增加數據樣本，我們在分別對 90 種不同數量的攝影機進行排程比較，並且對每一種數量的攝影機產生 100 次排程。

圖 6-4 到圖 6-6 分別呈現了在 10 台、20 台與 30 台攝影機的情況下，VSS、SSF、VSS-C 三種方法在不同數量的感測目標所得到的排程結果。從圖中觀察可以知道，排程總分會隨著感測目標數量增加，因為在固定的範圍下，感測器的位置越密集，同一視野能夠覆蓋的感測器會越多。在分數方面，不過在哪種情況下皆為：VSS > VSS-S > SSF，這是因為 SSF 將錨點設置在視野邊緣，亦即每一個視野中，至少有一個目標位於邊緣區域，VSS 與 VSS-C 則是能夠確保至少一個目標位於中心

區域，但由於目標彼此間的距離限制，VSS-C 在確定有一個目標於中間的情況下，在兩側邊緣區域感測到其他目標的機率較高，而 VSS 將錨點設置於中心區域的側邊上，更容易於中心區域感測到其他目標。

圖 6-7 到圖 6-9 則是呈現了不同攝影機數量下，隨著感測目標增加之排程失敗的比率，也就是在公式(4)中，目標的最小監控時間沒有被滿足之情況。如同前面章節所述，SSF 對於視野的時間安排是以視野內剩餘時間最大的目標為主，這種方法也讓失敗率大幅提高。至於，VSS 與 VSS-C 在排程上的優勢便顯現出來，由於優先安排只被一台攝影機監視到的目標，也降低了無法滿足最小監控時間的可能。在感測目標數量達到 70%以上後，三種排程失敗率趨近於 0，這是因為隨著目標數量增加，產生出較小的最小監控時間機率高，同時，在同一個視野中感測到的目標也會增加，如此一來，。從這六張圖我們可以知道，VSS 排程能夠有較大的機會以視野中心監控目標並且同時保持較高的排程成功率。

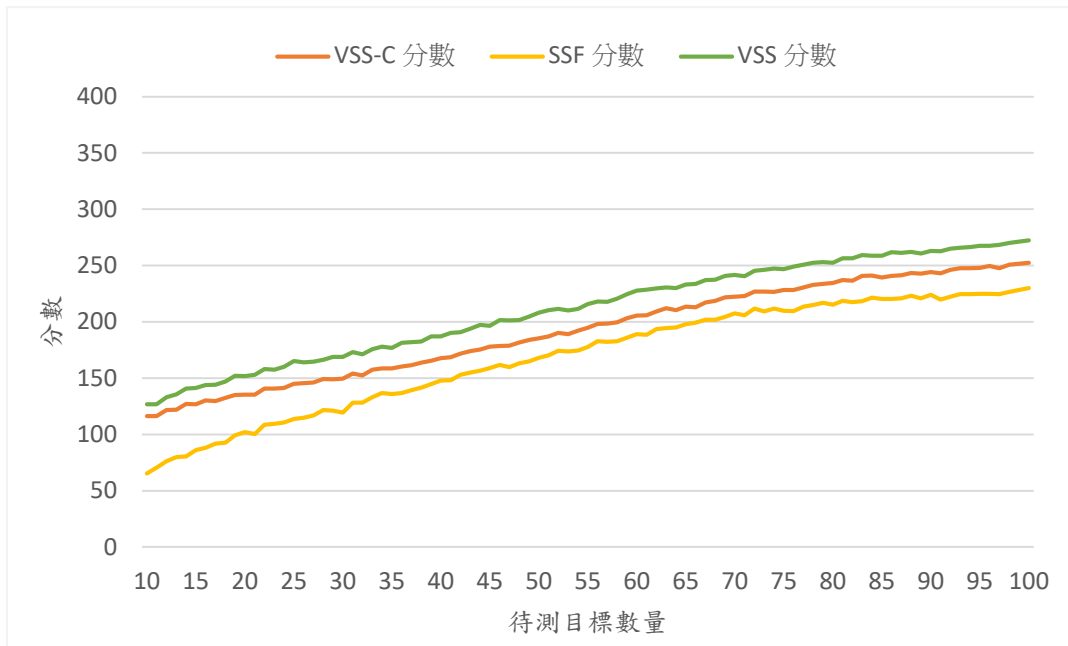


圖 6-4：目標數量對排程總分的影響(攝影機數量為 10)

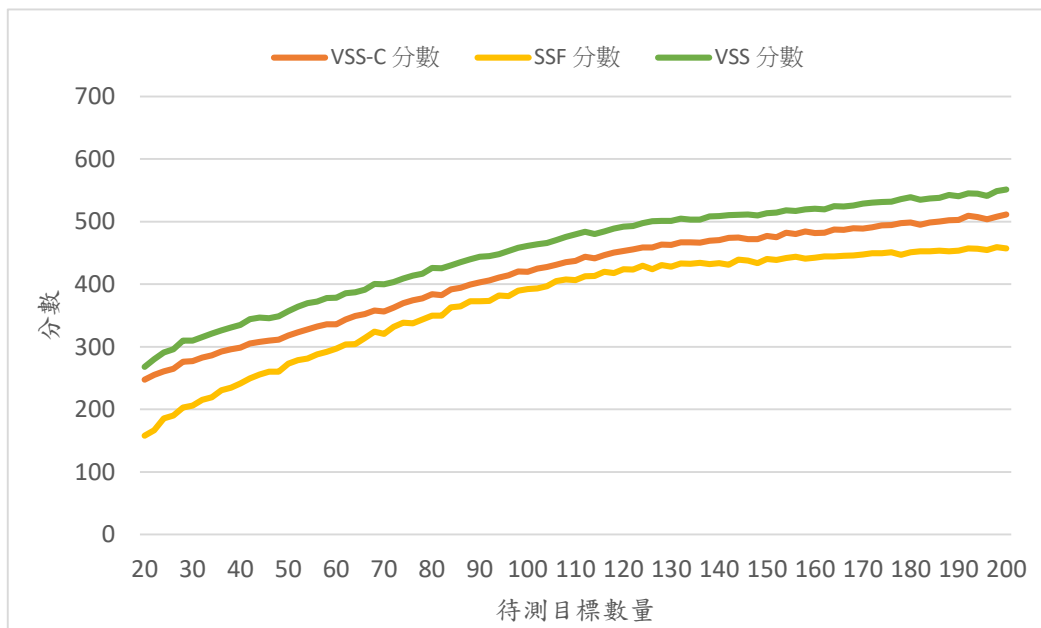


圖 6-5：目標數量對排程總分的影響(攝影機數量為 20)

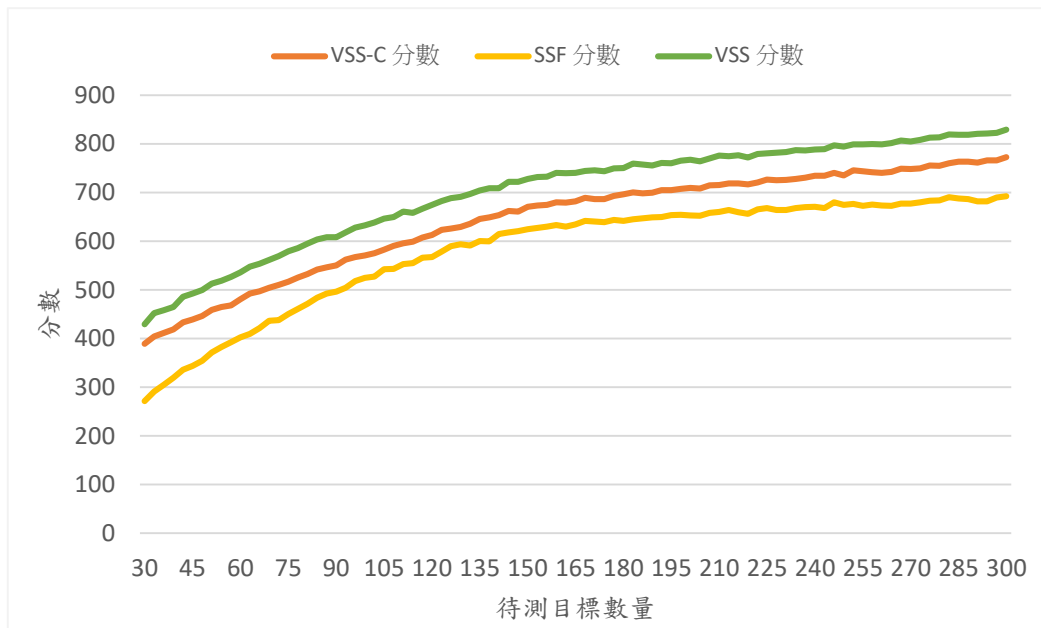


圖 6-6：目標數量對排程總分的影響(攝影機數量為 30)

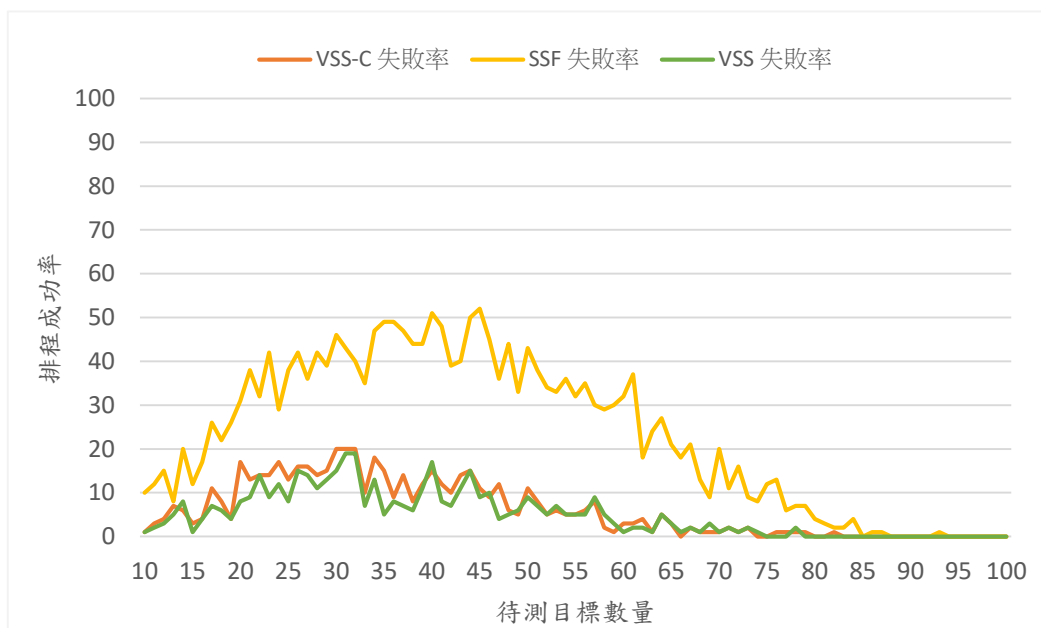


圖 6-7：目標數量對排程成功率的影响(攝影機數量為 10)

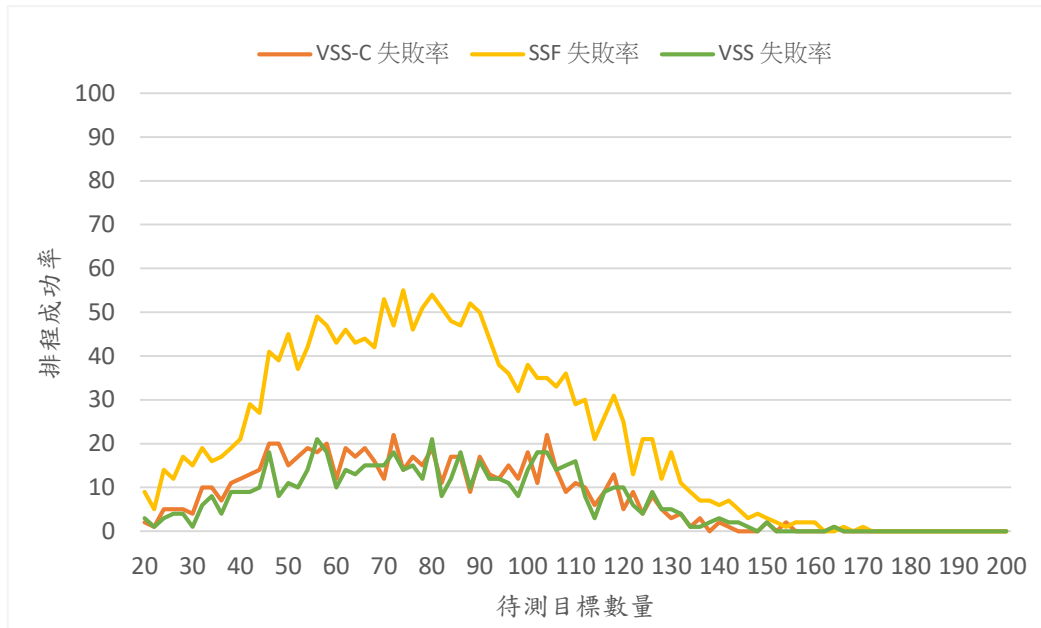


圖 6-8：目標數量對排程成功率的影響(攝影機數量為 20)

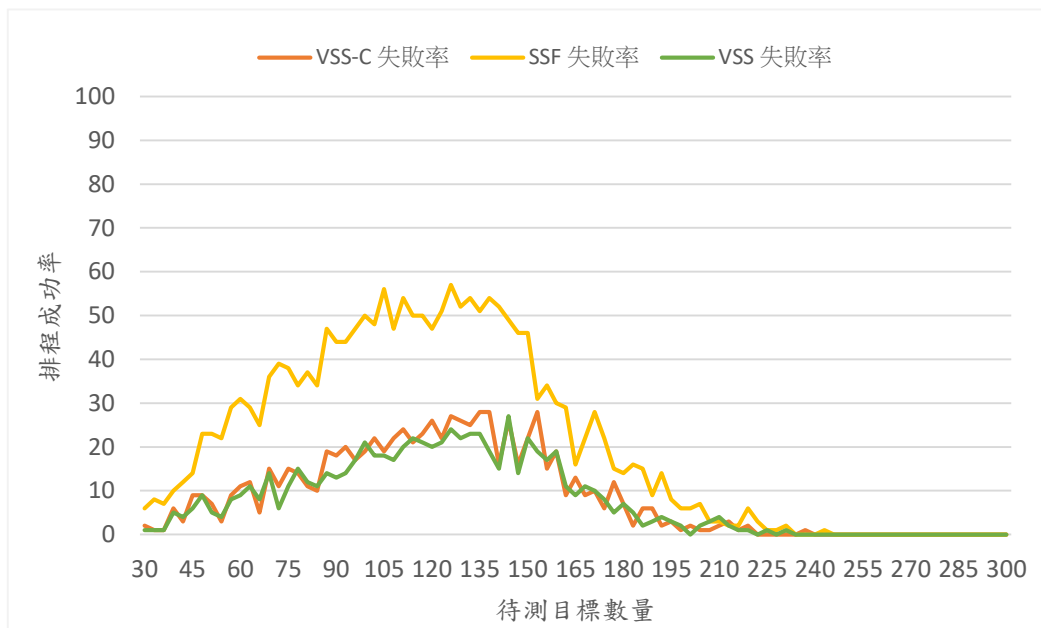


圖 6-9：目標數量對排程成功率的影響(攝影機數量為 30)

6.2.2 視野分界角度 φ 之影響

接著，觀察當視野分界角度 φ 不同時，對監控總分產生的影響。同樣在邊長 400 的正方形區域中，隨機部署了 20 台攝影機與 120 個監控目標，攝影機的視角為 60 度，視野角度由視角之 0.1 倍開始遞增至 0.9 倍，也就是從 6 度到 54 度，每一種角度皆測量 100 次取其平均。圖 6-10 呈現了三種演算法在不同的視野分界角度下所得到的分數，當 φ 值越大，代表該視野的中心覆蓋範圍越大，由此可知，即便再不同大小的 φ 角，VSS 仍然是總分最高的排程。圖 6-11 則是感測器在不同 φ 值下，在中心區域感測到目標次數的百分比。值得注意的是，VSS-C 與 VSS 在 $\varphi = 0.1$ 時，兩者差距並不大，而隨著 φ 角增加，VSS 的總分相較於 VSS-C 與 SSF 明顯上升了不少，這是因為監控目標之間有一定的距離，而 VSS 將錨點設置於中心區域的邊緣，當中心區域角度增加時，比起 VSS-C 能夠更容易於中心區域感測到目標，在圖 6-11 也驗證了這點，VSS 在不同角度下，以中心區域監控的比率皆高於另外兩種演算法。

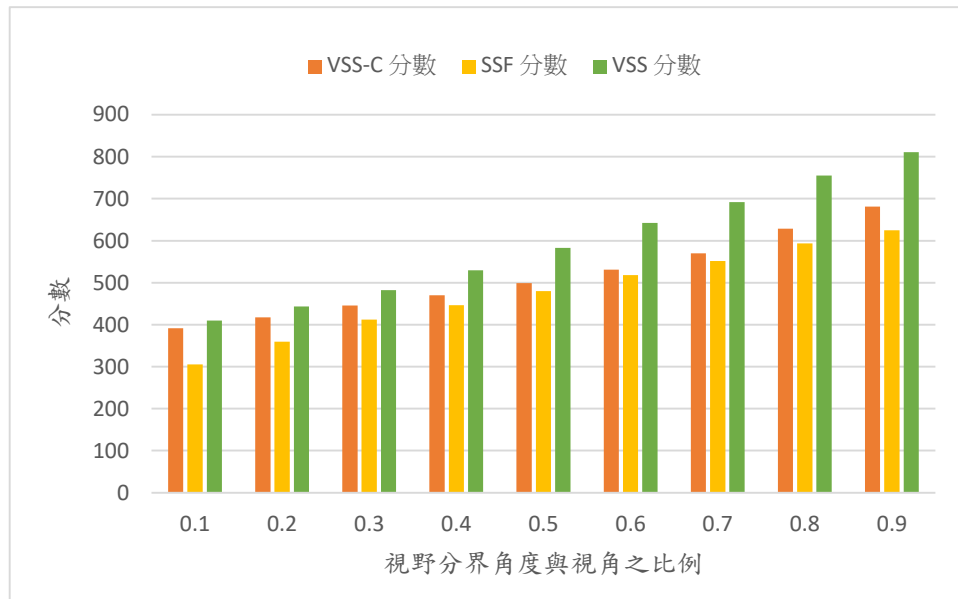


圖 6-10： ϕ 角與排程總分之影響

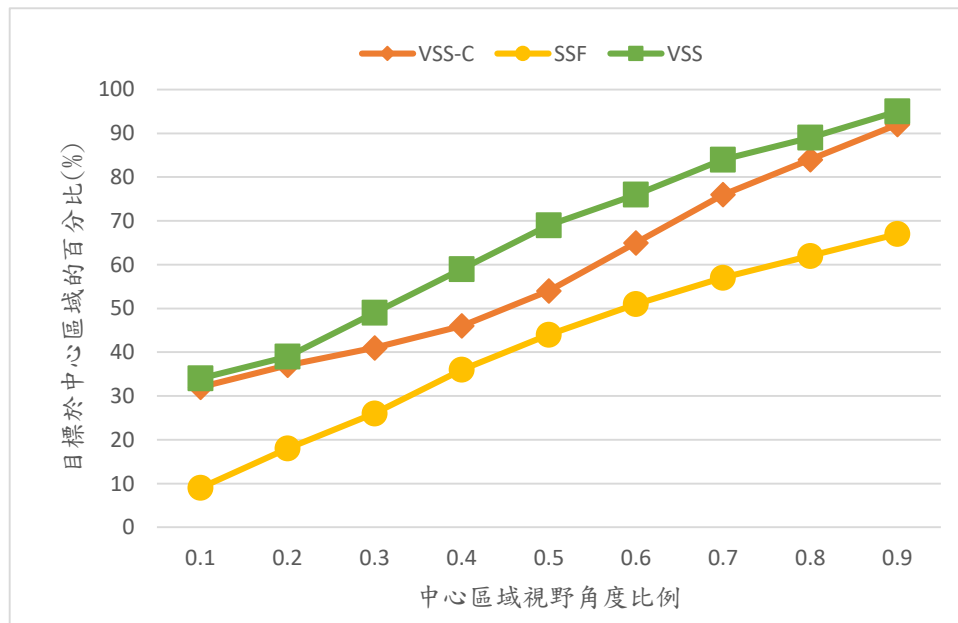


圖 6-11： ϕ 角與中心視野監控的比例之影響

6.2.3 分數權重 ω 之影響

最後是改變邊緣區域的權重時，對總分產生的影響。在同樣的環境下設定視角為 60 度，中心視野比例為 0.3，也就是 18 度，兩側區域的權重值由 0.1 遞增至 0.9，並且測量 100 次取其平均。圖 6-12 顯示了隨著兩側區域權重而改變的排程總分，其中，隨著權重的增加，SSF 的排程總分有著很明顯的成長，甚至超過了 VSS 與 VSS-C。我們將總分相差的百分比做為副座標軸比較，如圖 6-13 與 6-14，VSS-C 與 VSS 的差距由一開始的 9.7% 縮短到 1.2%，SSF 與 VSS 則是由 26% 縮短到 -9.7%，這也是如同前面所提到的，SSF 與 VSS-C 排程時錨點設定的關係，於兩邊視野所監控到目標的機會比 VSS 要來的多，因此，當兩側區域權重越高，區域間的差別減少，在這種情況下 SSF 才有機會得到比較高的分數。

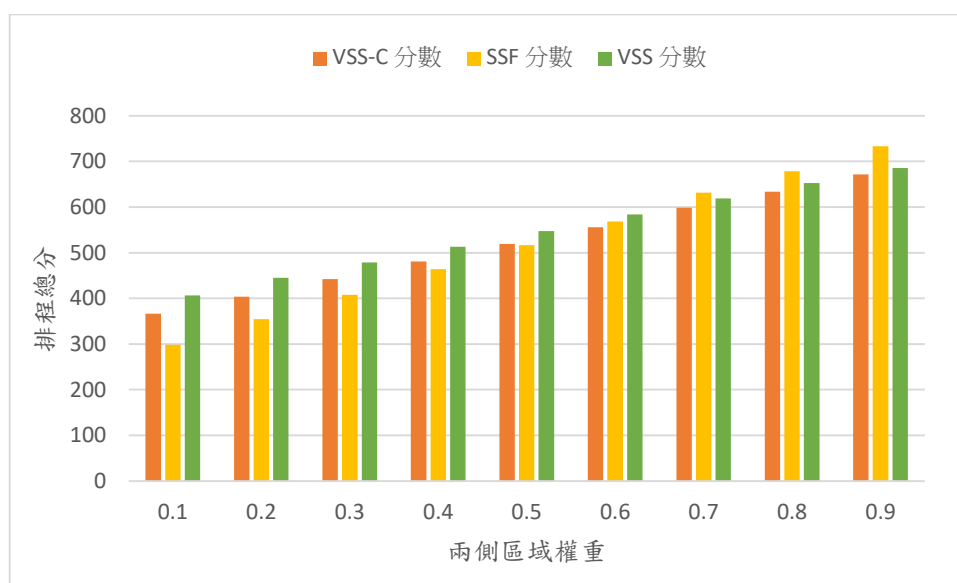


圖 6-12：兩側區域權重對排程總分變化影響

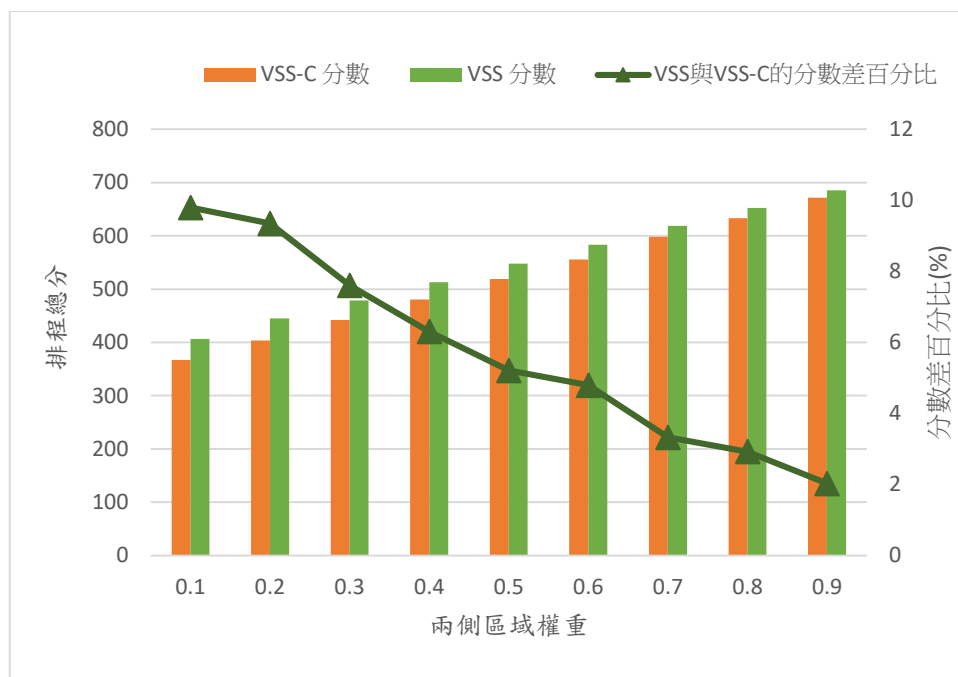


圖 6-13：兩側區域權重對 VSS 與 VSS-C 的排程總分變化影響

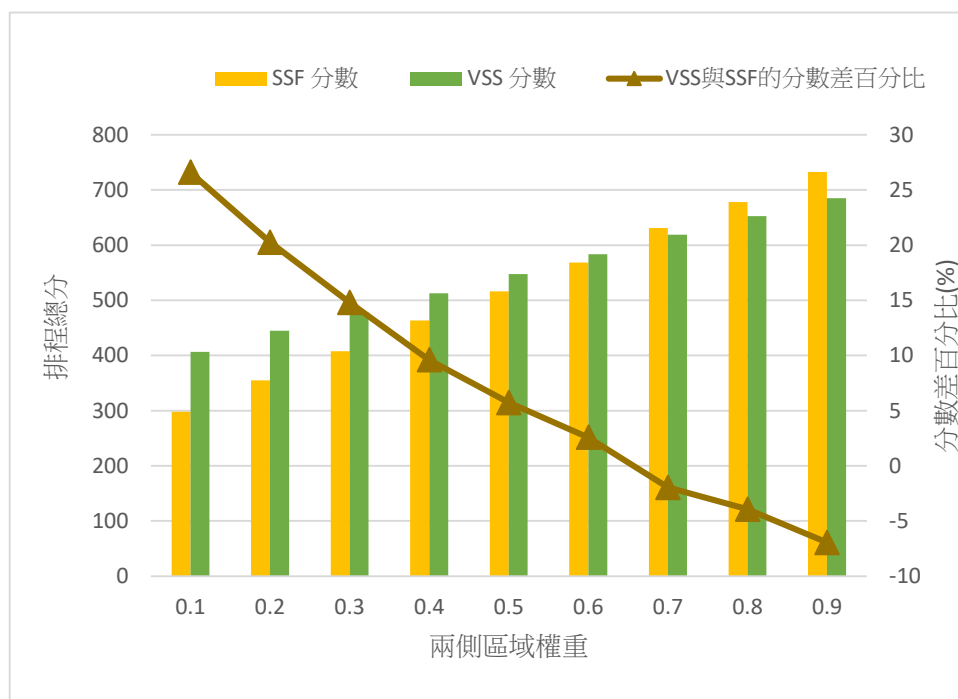


圖 6-14：兩側區域權重對 VSS 與 SSF 的排程總分變化影響

6.3 系統雛型實作

我們以 Python 腳本語言實作結合演算法的結果建立攝影機的雛型系統，系統元件如下：

1. 可運行 Python 3.7.0 或以上版本的使用者端電腦
2. PTZ 攝影機 D-Link 5222L 三台

系統架構與介面如圖 6-15 與圖 6-16，利用 Python 的 tkinter 元件製作 GUI 介面，並且以 multithread 方式搭配 http request 同步控制多台攝影機，在已知監控目標位置的情況下，針對給定的監控時間來控制攝影機的旋轉。使用者介面上有手動旋轉與排程功能，使用者可以自行設定旋轉幅度並且針對各個攝影機進行操控，排程功能則是將位置與時間預先設定好後再進行自動監控。圖 6-16 中，A~E 為待測目標，每個攝影機畫面下方的按鈕可以手動控制攝影機旋轉角度，Scheduling 按鈕則是能依照系統設定的排程對攝影機進行控制。

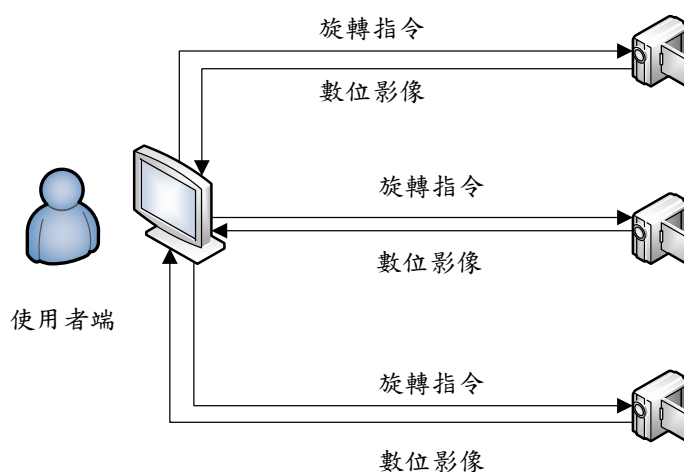


圖 6-15：系統架構

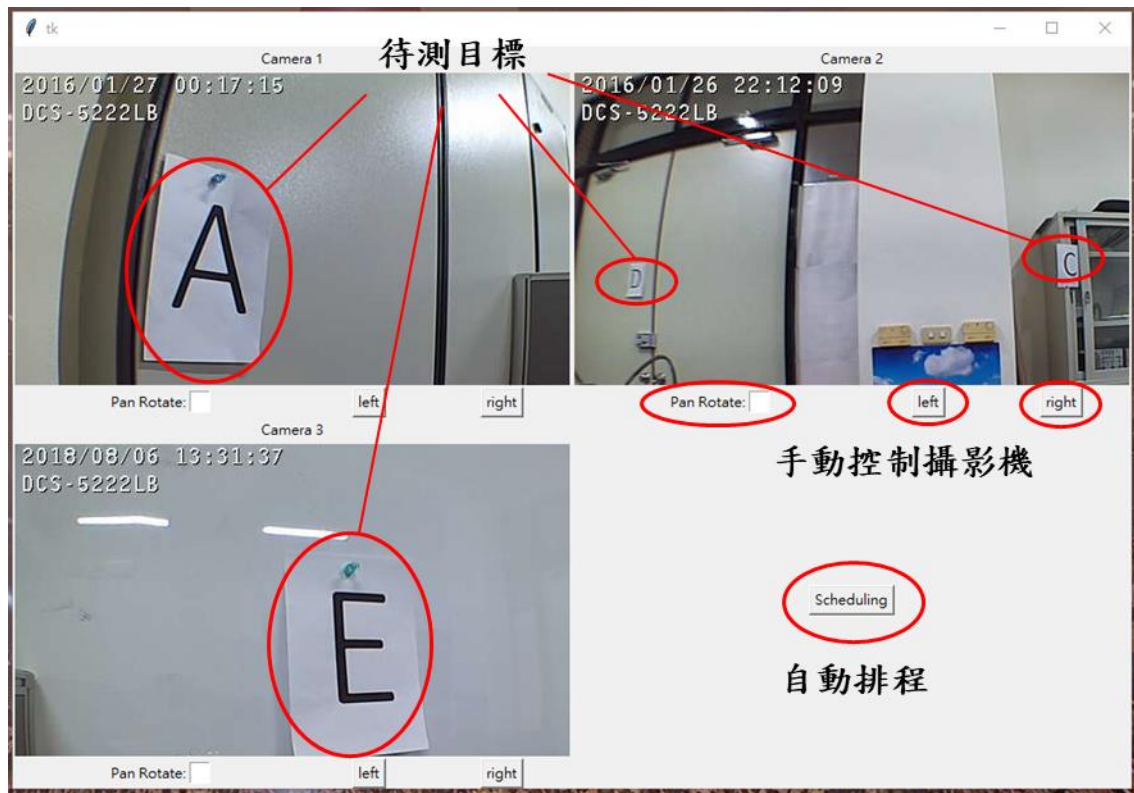


圖 6-16：使用者介面

由於 D-Link 5222L 最大旋轉角度為 340 度，我們在此系統設計了歸零的機制，當排程開始前，先順時針轉到底，將此時的位置作為 0 度，待排程監控結束後以逆時針轉到底後再進行歸零，然後進行下一輪排程，以這樣的方式來模擬 360 度旋轉並且同步攝影機，圖 6-17 與圖 6-18 為系統實作示意圖與排程結果。

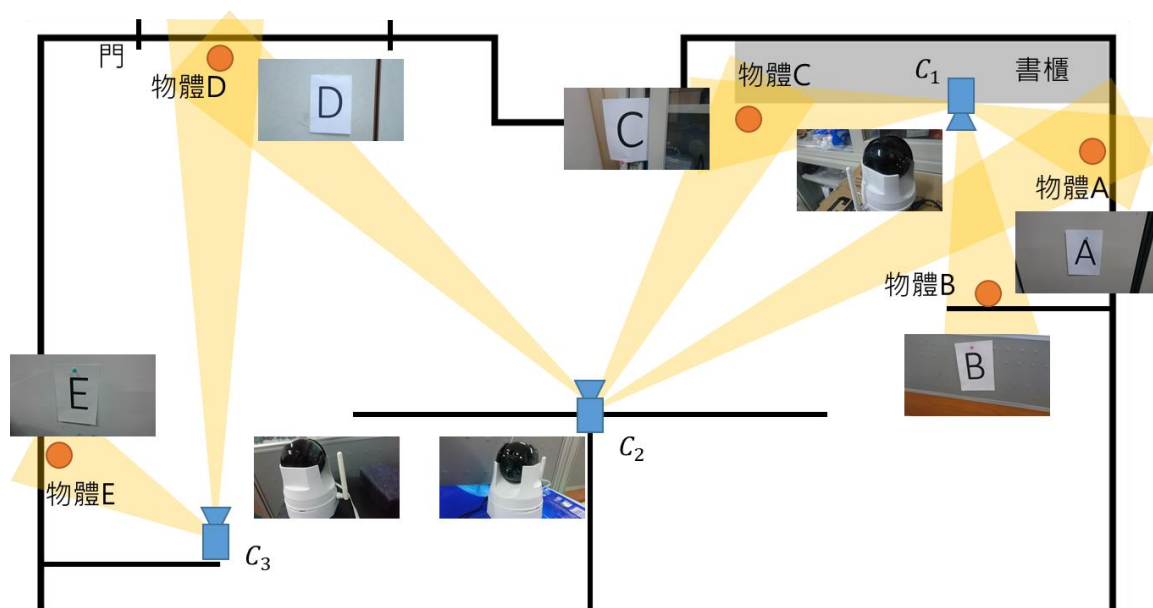


圖 6-17：系統實作示意圖

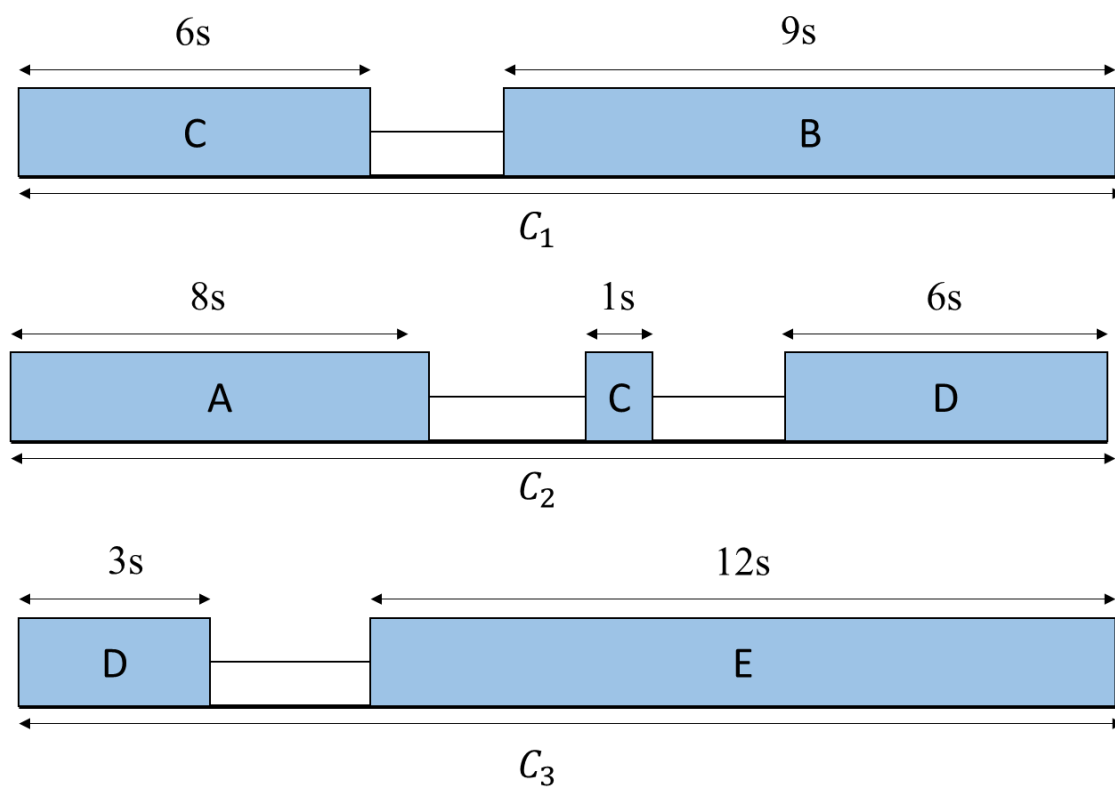


圖 6-18：實作排程結果

第七章 結論與未來研究方向

現有針對 PTZ 攝影排程的演算法中，大多沒有考慮到目標位置對於視野的差別，在同樣的範圍內生成的影像會因為監控目標的距離與角度而有不一樣的影像品質。在本論文中，我們以視野分界來模擬現實攝影機的情況同時做為一個量化實驗的標準，並且提出了有效利用中心視野進行排程的 VSS 方法，結合 SSF 尋找扇形並將錨點置於中心的側邊，如此一來，我們可以提升除了錨點之外的其他目標在中心視野被監控的機率，最後也由實驗結果驗證了 VSS 在滿足監控時間的條件下可以拿到最高的總分，我們也藉由比較不同的方法來證明 VSS 排程以中心視野來監控目標占了多數。

然而本論文主要以二維平面設計排程演算法，如果能考量三維空間，同時運用 PTZ 攝影機能夠水平與垂直旋轉的特性將會更加貼近實際情況。再者，我們所定義的視野分界只有將視野簡單分成兩旁與中間，未來可以將監控權重值根據距離與角度對比影像品質，以連續的函數表示權重，如此一來，能夠更精確的分析不同位置所帶來的影響，利用這點來改進排程方法。在實作方面，對於掌握了攝影機的影像後，利用影像處理或者深度學習等方法讓應用更加多元化，例如：排程的過程中發生已知位置的目標遺失能夠發出警訊；或者將通道作為目標，以人臉辨識分析出入人員的身分，並且持續追蹤未知的人員，這些應用都是未來可以思考的議題。

參考文獻

- [1] VCR: Webster's Timeline History, 1963 – 2007, ICON Group International, Inc. , 2010.
- [2] Information Security Today, “The evolution of video surveillance systems”. [Online].
- [3] S.J. Solari, “Architecture for a digital video recorder,” *Digest of Technical Papers. International Conference on Consumer Electronics*, pp. 32-33, 1994.
- [4] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” *IEEE Computer*, vol. 40, no. 6, pp. 60-66, Jun. 2007.
- [5] Y. Utsumi, and Y. Iwai, “Face tracking and recognition by using omnidirectional sensor network,” *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 2009
- [6] H. Cho, Y. Baek, and C. M. Kyung, “Wireless video sensor network platform and its application for public safety,” *IEEE International Conference on High Performance Computing and Communications, IEEE 6th International Symposium on Cyberspace Safety and Security, and IEEE 11th International Conference on Embedded Software and Systems*, pp. 743-746, 2014.
- [7] W. Balid, H. Tafish, and H. H. Refai, “Development of portable wireless sensor network system for real-time traffic surveillance,” *IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 201-205, 2015.
- [8] A. I. Mulla, A. P. Mulik, A. Prashant, and D. D. Gawai, “Continuous health surveillance system for cattle,” *International Conference on Intelligent Computing and Control Systems*, pp. 1192 -1195, 2017.
- [9] A. H. Maslow, “Motivation and Personality”, *Harper and Row*, New York, 1954

- [10] Edmund optics worldwide, “Understanding focal length and field of view,” [Online] <https://www.edmundoptics.com/resources/application-notes/imaging/understanding-focal-length-and-field-of-view/>.
- [11] Museum Of Obsolete Media, “135-film (1934–),” [Online] <http://www.obsoletemedia.org/135-film/>.
- [12] M. Valera, and S. A. Velastin, “Intelligent distributed surveillance systems: a review,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, pp. 192-204, 2015.
- [13] N. M. Oliver, B. Rosario and A. P. Pentland, “A Bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 831-843, 2000.
- [14] Y. Charfi, N. Wakamiya, and M. Murata, “Challenging issues in visual sensor networks,” *IEEE Wireless Communications*, vol. 16, pp. 44-49, 2009.
- [15] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Transactions on Computers*, vol. 51, pp. 1448-1453, 2002.
- [16] T. Sun, L. J. Chen, C. C. Han, and M. Gerla, “Reliable sensor networks for planet exploration,” *Networking, Sensing and Control*, pp. 816-821, 2005.
- [17] R. S. Chang, and S. H. Wang, “Deploying sensors for maximum coverage in sensor networks,” *International Conference on Wireless Communications and Mobile Computing*, pp. 220-224, 2007.
- [18] Y. Wang, M. Wilkerson, and X. Yu, “Hybrid sensor deployment for surveillance and target detection in wireless sensor networks,” *International Wireless Communications and Mobile Computing Conference*, pp. 326-330, 2011.
- [19] Y. Li, C. Ai, Z. Cai, and R. Beyah, “Sensor scheduling for p -percent coverage in wireless sensor networks,” *Cluster Computing*, pp. 27-40, 2011.

- [20] C. Wang, H. Jiang, T. Yu, and C. S. Liu, "SLICE: enabling greedy routing in high genus 3-D WSNs with general topologies," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2472–2484, 2016
- [21] A. Boubrima, W. Bechkit, and H. Rivano, "A new WSN deployment approach for air pollution monitoring," *IEEE Annual Consumer Communications & Networking Conference*, pp. 455-460, 2017.
- [22] I. Arfaoui, N. Boudriga, K. Trimeche, and W. Abdallah, "WSN-based border surveillance systems using estimated known crossing paths," *International Conference on Advances in Mobile Computing & Multimedia*, pp. 1-9, 2017
- [23] Y. Morsly, M. S. Djouadi, and N. Aouf, "On the best interceptor placement for an optimally deployed visual sensor network," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 43-51, 2010.
- [24] E. Horster, and R. Lienhart, "Approximating optimal visual sensor placement," *IEEE International Conference on Multimedia and Expo*, pp. 1257-1260, 2006.
- [25] C. Phama, A. Makhoulb, and R. Saadic, "Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 783-795, 2011..
- [26] C. Han, L. Sun, J. Guo, and C. Chen "Rotatable sensor scheduling for multi-demands of coverage in directional sensor networks," *International Conference on Computer Communication and Networks*, pp. 1-8, 2016.
- [27] S. Peng, Y. Xiong, M. Wu, and J. She, "A new method of deploying nodes for area coverage rate maximization in directional sensor network," *Annual Conference of the IEEE Industrial Electronics Society*, pp. 8452-8457, 2017.
- [28] Z. H. Zhan, J. Zhang, Y. Li, and S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 6, pp. 1362-1381, 2009.

- [29] P. Zhou, J. Wu, and C. Long, "Probability-based optimal coverage of PTZ camera networks", *IEEE International Conference on Communications*, pp. 218-222, 2012.
- [30] J. Li, K. Yue, W. Y. Liu, and Q. Liu, "Game-theoretic based distributed scheduling algorithms for minimum coverage breach in directional sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, pp. 1-12, 2014.
- [31] Y. C. Wang, and S. E. Hsu, "An efficient deployment heuristic to support temporal coverage of heterogeneous objects in rotatable and directional (R&D) sensor networks," *IEEE Vehicular Technology Conference*, pp 1-5, 2014.
- [32] L. Guo, Y. Zhu, D. Li, and D. Kim, "PTZ camera scheduling for selected area coverage in visual sensor networks," *IEEE 35th International Conference on Distributed Computing Systems*, pp. 379-388, 2015.
- [33] A. Mittal, L. S. Davis, and S. N. Lim, "Constructing task visibility intervals for a surveillance system," *ACM International Workshop on Video Surveillance & Sensor Networks*, pp. 141-148, 2005.
- [34] N. Basilico, N. Gatti, and F. Amigoni, "Developing a deterministic patrolling strategy for security agents," *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pp. 565-572, 2009.
- [35] M. Alaei, and J. M. Barcelo-Ordinas, "Node clustering based on overlapping FoVs for wireless multimedia sensor networks," *IEEE Wireless Communication and Networking Conference*, pp. 1-6, 2010.
- [36] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions," *SIGGRAPH sketches*, pp. 22, 2007