

國立交通大學

電子工程學系電子研究所

博士論文

貝氏階層式結構於視訊監控之研究與應用

A Study of Bayesian Hierarchical Framework
and Its Applications to Video Surveillance



研 究 生：黃敬群

指 導 教 授：王聖智 博士

中 華 民 國 九 十 九 年 九 月



貝氏階層式結構於視訊監控之研究與應用

A Study of Bayesian Hierarchical Framework and Its Applications to Video Surveillance


研究生：黃敬群

Student: Ching-Chun Huang

指導教授：王聖智 博士

Advisor: Dr. Sheng-Jyh Wang

國立交通大學
電子工程學系 電子研究所博士班
博士論文



A Dissertation Submitted to
Department of Electronics Engineering & Institute of Electronics
College of Electrical Engineering and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Electronics Engineering
September 2010
Hsinchu, Taiwan, Republic of China

中華民國九十九年九月



貝氏階層式結構於視訊監控之研究與應用

研究生：黃敬群

指導教授：王聖智博士

國立交通大學電子工程學系電子研究所 博士班

摘要

在本論文中，我們提出以貝氏階層式結構為基礎的分析方法，讓視訊監控系統得以用一致的架構，同時分析影像內容以及推論空間中場景的資訊。在真實的場景中，為了實現一套穩健的視訊監控系統，往往會面臨許多挑戰，諸如物體間相互遮蔽、前景物體與背景物體外貌相似而產生的混淆、透視投影所造成的物體形變、陰影的變化、還有外在光線變化造成的影像變異。在這篇論文中，我們發現，透過將空間場景適當的參數化，並同時依據場景模型和擷取到的影像資料來進行分析，系統將能更輕易地處理前面所提及的變異因素。在貝氏階層式架構中，我們透過階層式表示法將以像素特徵為基礎的資訊、以區域影像內容為基礎的資訊、與以物件特性為基礎的資訊，透過機率的方式進行有系統的整合，以支援影像內容的分析與場景資訊的推論。透過所提出的貝氏階層式架構，前面所提到的許多變異因素可以被有效地解決，除此之外，某些變異因素還可進一步變成有效的線索來協助三維場景資訊的推論。

在本論文中，我們將貝氏階層式架構實際應用在停車場空位偵測系統以及多攝影機視訊監控系統。在停車場空位自動偵測的系統上，實際的戶外停車場監控場景往往受到許多變因的影響，進而降低了系統的正確性，這些變因包含：(a)戶外變化劇烈的環境光源；(b)陰影的影響；(c)透視法上幾何投影所產生的變形；(d)停放車輛之間產生的相互遮蔽問題。藉由所提出的貝氏階層式結構，我們可以有

系統地將前述的許多變因加入停車空位的推論過程中，以降低這些變因對系統效能的影響。我們的貝氏階層式結構透過建立參數化的空間場景模型來描述空間中的遮蔽現象、幾何上的投影變形、以及陰影等變因所形成的影響，同時也將環境光線變化所造成的色彩變動視為一種色彩分類的問題，並藉由分類程序的建立來描述光線的變化。實驗結果顯示，我們的系統可以穩定地偵測空位的位置、有效地標記並區分影像中屬於地面或車輛的區域、確切地標記屬於陰影的區域、以及克服光線變化所衍生的問題。

另一方面，在多攝影機視訊監控系統中，我們自動地定位、標記、與對應在不同攝影機監控範圍內的多個物體，同時有效壓抑因為幾何深度上的不確定性所產生的假物體。多攝影機視訊監控系統在真實的應用場景中，往往面臨一些具挑戰性的議題：(a) 場景中未知物體的數量；(b) 物體間的相互遮蔽；以及(c) 假物體的出現。有別於過去的方法，我們提出了一套包含資訊整合與場景推論的兩步驟策略。在資訊整合的步驟中，我們整合來自多攝影機的資訊以建立一機率分佈，藉以描述物體出現於地面某一位置的可能性。在場景推論的步驟中，我們應用貝氏階層式結構將場景模型納入考量，透過此結構，我們將物件在影像內的標記議題、物件在多攝影機間的對應議題、以及假物件的消除議題整合為單一的最佳化問題。此外，我們進一步採用期望-最大化架構來調整出更好的物體三維模型，透過貝氏階層式結構與期望-最大化架構的結合，我們可以得到更好的系統效能。實驗結果顯示，我們的系統可以自動地決定場景中的運動物體數量、有效地標記並對應出不同攝影機影像中的多個物體、準確地定位物體在三維場景中的位置、並且能有效地清除假物件。

在本論文中，我們驗證了以貝氏階層式結構為基礎的影像分析架構可以有效地應用到視訊監控的分析與應用上。透過此架構，我們將像素層級的色彩資訊、像素間的區域層級資訊、以及以物體為基本單位的物件層級資訊有系統地整合在一起，這樣的整合讓系統可以擁有更多的資訊，並可以針對較複雜的影像內容進行準確的推論分析。

A Study of Bayesian Hierarchical Framework and Its Applications to Video Surveillance

Student : Ching-Chun Huang

Advisor : Dr. Sheng-Jyh Wang

Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University

Abstract

In this dissertation, we present a Bayesian hierarchical framework (BHF) to simultaneously deal with 3-D scene modeling and image analysis in a unified manner. In practice, to develop a robust video surveillance system, many challenging issues need to be taken into account, such as occlusion effect, appearance ambiguity between foreground and background, perspective effect, shadow effect, and lighting variations. In this dissertation, we find a way to handle these challenging issues by modeling 3-D scene in a parametric form and by integrating scene model and image observation together in the inference process. In the proposed hierarchical framework, we systematically integrate pixel-level information, region-level information, and object-level information in a probabilistic way for the semantic inference of image content and 3-D scene status. Under this BHF framework, occlusion effect, appearance ambiguity, perspective effect, shadow effect, and lighting variations can be well handled. Actually, in the BHF framework, occlusion effect, perspective effect, and shadow effect may even provide useful clues to support 3-D scene inference.

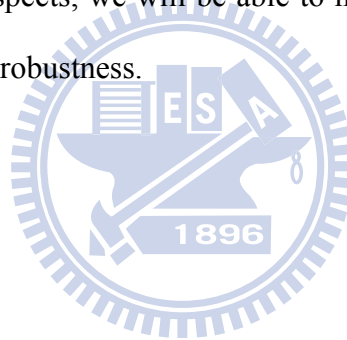
In this dissertation, the BHF framework is applied to two video surveillance systems: a vacant parking space detection system and a multi-camera surveillance

system. In the vacant parking space detection system, the challenges come from dramatic luminance variations, shadow effect, perspective distortion, and the inter-occlusion among vehicles. With the proposed BHF, those issues can be well modeled in a systematic way and can be effectively handled. In detail, the proposed BHF scheme depicts the occlusion pattern, perspective distortion, and shadow effect by building a parametric scene model. On the other hand, the color fluctuation problem caused by luminance variation is treated as a color classification problem. With the BHF scheme, the detection of vacant parking spaces and the labeling of scene status are regarded as a unified Bayesian optimization problem subject to a shadow generation model, an occlusion generation model, and an object classification model. The system accuracy was evaluated by testing over a few outdoor parking lot videos captured from morning to evening. Experimental results showed that the proposed framework can systematically detect vacant parking spaces, efficiently label ground and car regions, precisely locate shadowed regions, and effectively handle luminance variations.

On the other hand, in the application of multi-target detection and tracking over a multi-camera system, the main goal is to locate, label, and correspond multiple targets with the capability of ghost suppression over a multi-camera surveillance system. In practice, the challenges of this kind of system come from the unknown target number, the inter-occlusion among targets, and the ghost effect caused by geometric ambiguity. Instead of directly corresponding objects among different camera views, the proposed framework adopts a fusion-inference strategy. In the fusion stage, we formulate a posterior distribution to indicate the likelihood of having some moving targets at certain ground locations. In the inference stage, the scene model is inputted into the proposed BHF, where the target labeling, target correspondence, and ghost removal are regarded as a unified optimal problem subject to 3-D scene priors, target priors,

and image observations. Moreover, the target priors are iteratively refined based on an expectation-maximization (EM) process to further improve the system performance. The system accuracy is evaluated via both synthesized videos and real videos. Experimental results showed that the proposed system can systematically determine the target number, efficiently label and correspond moving targets, precisely locate their 3-D locations, and effectively tackle the ghost problem.

With simulations over these two applications, we verified that the proposed BHF scheme can be well applied to various kinds of video surveillance applications. This BHF framework provides the flexibility to properly integrate pixel-level, region-level, and object-level information into a unified inference process. With the integrated information from multiple aspects, we will be able to handle more complicated tasks with improved accuracy and robustness.



Acknowledgements

從小學、中學、一直到博士的學習過程，我需要感謝每一位無私而慷慨指導我的導師。在這些敬愛的老師中，影響我最深，教導我最多，支持我最久，就是我的指導教授王聖智博士。在我的腦海中，老師對我的指導，早在高中推薦甄試後，收到老師寄來一本“電腦系統”的原文書便已開始；在大學專題實作與研究所學習時，老師幾乎是亦步亦趨地扶著學生，教我如何做研究；當我開始研習博士學程時，老師總是不遲辛苦地牽著學生的手，一字一句地教導如何寫論文。十多年的學習歲月，我何等幸運可以得到王老師無數的關懷與包容，以及數不盡的疼愛與教誨。印象中的老師，堅毅、認真、嚴謹、而且仁慈。學者的風範，深刻地影響著我，在無所適從的時刻，正因著老師的精神與我同伴，得以尋得正確的方向。真的非常感謝老師的栽培，謝謝老師。

能夠完成學業，我還要感謝我的家人，特別是我的母親——傅玉女女士。感謝母親的教養與督促，總是在我洩氣時給我重新站起來的力量；在我做決定時給予我適時的建議與全部的信任；感謝母親在生活與精神上全力的支持。也要感謝珮婷，在最艱苦的時刻，給我絕對的肯定與滿滿的歡笑。

此外，我要感謝實驗室的許多好伙伴。沒有您們的支持與陪伴，研究生活中必然缺少調色而黯淡非常。感謝熱心的奕安與瑞男，可愛又聰明的晴駿，貼心又搞笑的維辰和庭璋，認真負責的博凱、瑋國、周節、禎宇，以及老是聽我訴苦的慈澄。還有許許多多學弟妹，您們真是太棒啦！

另外，工研院以及遠在美國卡內基大學的許多師長與朋友，感謝您們讓我的生命更寬廣。感謝陳祖翰教授、張耀仁博士的指導；感謝余孝先副所長、張森嘉博士給我工作的機會，發揮所學的舞台，並在我犯錯時包容與教導我。感謝鴻欣、正一、博超 … 等工作上合作無間的夥伴。

我還要感謝投稿過程中給予我寶貴意見的許多匿名的論文評論委員。我也要感謝敬愛的博士班口試委員們，洪一平博士、戴顯權博士、黃仲陵博士、張文鍾博士、莊仁輝博士、許秋婷博士、林嘉文博士以及王聖智博士，很感謝您們的愛護及教誨。

最後，我要感謝主一路的保守與眷顧，沒有主的陪伴與引領，許多事情都無法憑自己的力量克服，感謝主，願這得來不易的喜悅，可以彰顯主的光輝與榮耀。

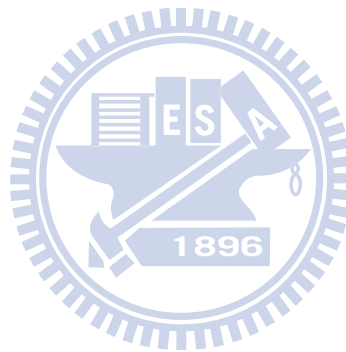
黃敬群 2010 年 9 月

Contents

摘要	i
Abstract	iii
Acknowledgements	vi
Contents	vii
List of Tables	x
List of Figures	xi
List of Notations	xvi
Introduction	1
1.1 Overview	1
1.2 Contribution	4
1.3 Organization	6
Backgrounds	7
2.1 Image Analysis Techniques	8
2.1.1 Pixel-level Methods	8
2.1.2 Region-level Methods	13
2.1.3 Object-level Methods	18
2.2 Connection between Image Analysis and 3D Scene Modeling	24
Bayesian Hierarchical Framework	28
3.1 The Structure of BHF	28
3.2 The Property of BHF	31
3.2.1 Differences to Data-driven and Model-driven Methods	31

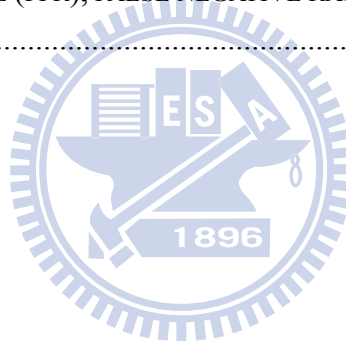
3.2.2	Differences to Existing Hybrid Methods	33
3.3	The Modeling of BHF	35
3.4	The Inference of BHF	41
3.5	The Application of BHF	44
A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection		46
4.1	Introduction of Parking Space Detection	46
4.2	Overview of Vacant Space Detection	50
4.3	Top-Down Knowledge From Scene Layer	54
4.3.1	3-D Scene Parameters	55
4.3.2	Generation of Expected Labeling Maps	56
4.3.3	Estimation of Sunlight Direction	60
4.4	Bottom-Up Messages From Observation Layer	62
4.4.1	Classification Energy Model	63
4.4.2	Adjacency Energy Model	69
4.5	Vacant Parking Space Detection	70
4.5.1	Optimal Inference of Parking Space Status	70
4.5.2	Refinement of Classification Energy Model	72
4.5.3	System Setup and Online Vacant Space Detection	73
4.6	Experiment Results and Discussion	75
4.6.1	Experiment Setup and Test Data	75
4.6.2	Object/Shadow Labeling and Accuracy of Vacant Space Detection	76
4.6.3	Discussion and Future Works	83
Multi-Target Correspondence and Labeling with Ghost Suppression over Multi-Camera System		85
5.1	Introduction	85
5.2	System Overview	89
5.2.1	System Property	90
5.2.2	System Flow	91
5.3	Information Fusion and Summarization	93
5.3.1	Foreground Detection on Single Camera	93
5.3.2	Information Fusion	93
5.3.3	Representation of TDP and Information Summarization	97
5.3.4	Ghost Object	99
5.4	Bayesian Inference and Ghost Suppression	100

5.4.1	System Modeling	102
5.4.2	Multi-Target Labeling and Tracking.....	107
5.5	Results and Discussion	114
5.5.1	Experimental Datasets	114
5.5.2	Foreground Detection and Information Fusion.....	115
5.5.3	Accuracy of Target Location.....	119
5.5.4	Detection and Labeling with Ghost Removal.....	121
5.5.5	Multi-target Tracking on the Ground Plane.....	123
5.5.6	System Complexity.....	124
5.5.7	Future Works.....	125
	Conclusions.....	126
	Bibliography	133
	Curriculum Vita.....	140



List of Tables

TABLE 1. PERFORMANCE COMPARISON OF FOUR VACANT SPACE DETECTION ALGORITHMS.	82
TABLE 2. ACCURACY OF TARGET LOCATION IN THREE DIFFERENCE ZONES FOR FLEURET'S SEQUENCE.	120
TABLE 3. FALSE POSITIVE RATE (FPR), FALSE NEGATIVE RATE (FNR).....	122
TABLE 4. RUNTIME LIST	125



List of Figures

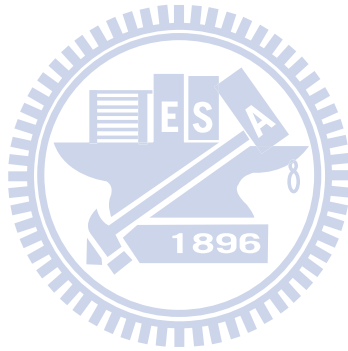
FIG. 1. AN EXAMPLE OF HUMAN DETECTION AND HUMAN IDENTITY LABELING. (A) TEST IMAGE. (B) HUMAN DETECTION RESULT. (C) HUMAN LABELING RESULT, WITH DIFFERENT COLORS INDICATING DIFFERENT PERSONS.....	2
FIG. 2. THE PROBABILITY DISTRIBUTION OF A PIXEL WITH GAUSSIAN MIXTURE MODEL [2].	10
FIG. 3. BACKGROUND SUBTRACTION RESULTS BASED ON GAUSSIAN MIXTURE MODEL.	11
FIG. 4. THE BACKGROUND SUBTRACTION RESULTS BASED ON THE METHOD PROPOSED BY HEIKKILÄ ET AL. [14]. THE FIRST AND THIRD ROWS ARE THE TEST IMAGES. THE SECOND AND FOURTH ROWS ARE THE DETECTION RESULTS. (FIGURES COURTESY OF MARKO HEIKKILÄ [14]).....	14
FIG. 5. A BACKGROUND SUBTRACTION RESULT BASED ON THE METHOD OF ELGAMMAL ET AL. [10]. (A) A TEST IMAGE. (B) PER-PIXEL DETECTION RESULT. (C) PER-PIXEL DETECTION RESULT WITH NEIGHBORING CONSIDERATION. (FIGURES COURTESY OF A. ELGAMMAL [10])	15
FIG. 6. THE PROCEDURE FOR THE FOREGROUND MODELING IN [17] BASED ON THE SPATIAL STATISTICS. (FIGURES COURTESY OF Cs. BENEDEK [17]).....	17
FIG. 7. A TYPICAL OBJECT-BASED DETECTION PROCEDURE WITH SLIDING WINDOW. HERE, WE USE FACE DETECTION AS AN EXAMPLE.	19
FIG. 8. ILLUSTRATION OF SVM CLASSIFICATION WITH A HYPERPLANE THAT SEPARATES POSITIVE EXAMPLES (“+”S) FROM NEGATIVE EXAMPLES (“O”S) WITH THE MAXIMUM MARGIN. SUPPORT VECTORS ARE PARTS OF THE TRAINING EXAMPLES THAT LIE ON THE BOUNDARY.	20
FIG. 9. THE PICTORIAL STRUCTURE FRAMEWORK PROPOSED BY FISCHLER AND ELSCHLAGER [29].	21
FIG. 10. THE PART-BASED OBJECT DETECTION REPORTED IN [32]. (A) A TESTED IMAGE WITH THE DETECTED HUMAN AND ITS PARTS. (B) THE HOG HUMAN MODEL. (C) THE HOG MODELS OF EACH BODY PARTS. (D) THE DEFORMABLE MODELS	

DEPICTING THE POSSIBLE VARIATION OF EACH PART. (FIGURES COURTESY OF P. FELZENSZWALB [32]).....	22
FIG. 11. FOUR REPRESENTATION METHODS FOR THE HUMAN MODEL. (FIGURES COURTESY OF JK AGGARWAL [39])	23
FIG. 12. ILLUSTRATE THE PROCESS OF AUTOMATIC PHOTO POP-UP [54]. (A) AN INPUT IMAGE. (B) THE SURFACE LAYOUT WITH GREEN, RED, AND PURPLE REPRESENTING SUPPORT SURFACES, VERTICAL SURFACES, AND SKY. (C) ONE SYNTHESIZED IMAGE VIEW. (D) ANOTHER SYNTHESIZED IMAGE VIEW. (FIGURES COURTESY OF D. HOIEM [54]).....	25
FIG. 13. HUMAN DETECTION BASED ON SCENE KNOWLEDGE [53][56]. (A) AN INPUT IMAGE. (B) THE SURFACE LAYOUT WITH GREEN, RED, AND BLUE REPRESENTING SUPPORT SURFACES, VERTICAL SURFACES, AND SKY. (C) DETECTION WITHOUT SCENE INFORMATION. THE DETECTION WINDOWS ARE UNIFORMLY DISTRIBUTED IN IMAGE. (D) DETECTION WITH THE PRIOR OF SURFACE LAYOUT. THE DETECTION WINDOWS ARE MAINLY DISTRIBUTED IN THE “VERTICAL” SURFACES. (E) DETECTION WITH THE PRIOR OF DEPTH AND CAMERA VIEWPOINT. THE DETECTION WINDOWS ARE LARGER IN THE NEAR DISTANCE. (F) DETECTION WITH THE PRIOR OF SURFACE LAYOUT, DEPTH AND CAMERA VIEWPOINT. THE DETECTION WINDOWS ARE FEWER AND MORE ACCURATE. (FIGURES COURTESY OF D. HOIEM [56])	26
FIG. 14. (A) THE PROPOSED BAYESIAN HIERARCHICAL FRAMEWORK (BHF). (B) BHF FOR THE VACANT PARKING SPACE DETECTION SYSTEM. (C) BHF FOR THE MULTI-TARGET MULTI-CAMERA SURVEILLANCE SYSTEM.	29
FIG. 15. EXAMPLES OF SIGM(U) WITH $P=0.05$ AND $C_{TH}=100$	40
FIG. 16. ILLUSTRATE THE INFERENCE PROCESS OF BHF. (A) A STANDARD INFERENCE PROCESS. (B) AN EXAMPLE OF BHF INFERENCE PROCESS FOR THE MULTI-TARGET MULTI-CAMERA SURVEILLANCE SYSTEM.	40
FIG. 17. THE GRAPH SETTING FOR THE GRAPH CUTS ALGORITHM.	43
FIG. 18. IMAGE SHOTS OF A PARKING LOT. (A) CAPTURED IN A NORMAL DAY. (B) CAPTURED IN A DAY WITH STRONG SUNLIGHT. (C) CAPTURED IN A CLOUDY DAY. ...	47
FIG. 19. THE CONCEPT OF BAYESIAN HIERARCHICAL FRAMEWORK FOR VACANT SPACE DETECTION.....	52
FIG. 20. ILLUSTRATION OF THE 3-LAYER BHF FOR VACANT SPACE DETECTION.....	53
FIG. 21. (A) A 3-D CAR MODEL. (B) EXPECTED CAR LABELING MAP OF A PARKED CAR. (C) EXPECTED CAR LABELING OF ALL PARKED CARS. (D) EXPECTED GROUND LABELING OF ALL PARKED CARS.	57
FIG. 22. (A) SHADOW FORMATION. (B) EXPECTED SHADOW LABELING MAP.	59
FIG. 23. (A) A 3-D CAR MODEL. (B) EXPECTED CAR LABELING MAP OF A PARKED CAR. (C) EXPECTED CAR LABELING OF ALL PARKED CARS. (D) EXPECTED GROUND	

LABELING OF ALL PARKED CARS.	59
FIG. 24. ILLUSTRATION OF SOLAR MOVEMENT AND SUNLIGHT DIRECTION.	60
FIG. 25. (A) A PARKING LOT IMAGE WITH THREE MANUALLY SELECTED IMAGE PIXELS, MARKED IN RED, GREEN, AND BLUE. (B) THE INTENSITY PROFILES (BLUE) OF THE GREEN PIXEL, OVERLAPPED WITH THE FITTED SKYLIGHT PROFILE (GREEN) AND THE FITTED SKYLIGHT+SUNLIGHT PROFILE (RED).....	62
FIG. 26. THE COLOR DISTRIBUTIONS (A) OF SHADOWED GROUND PIXELS, (B) OF UN-SHADOWED GROUND PIXELS, (C) OF SHADOWED CAR PIXELS, AND (D) OF UN-SHADOWED CAR PIXELS.....	63
FIG. 27. (A) THE REFERENCE GROUND PATCH (RED) AND THE GROUND PATCHES (PINK) FOR THE LEARNING OF GROUND REFLECTANCE FUNCTION. (B) THE CAR PATCHES (PINK) FOR THE LEARNING OF CAR REFLECTANCE FUNCTION.	67
FIG. 28. ILLUSTRATION OF PARKING SPACE STATUS INFERENCE.	71
FIG. 29. COMPARISON OF CAR PIXEL LABELING. (A) TEST IMAGES. (B) REGIONS LABELED AS CAR PIXELS BASED ON [13]. (C) REGIONS LABELED AS CAR PIXELS BASED ON THE PROPOSED METHOD.	78
FIG. 30. COMPARISONS OF GROUND PIXEL LABELING. (A) TEST IMAGES. (B) REGIONS LABELED AS GROUND PIXELS BASED ON [11]. (C) REGIONS LABELED AS GROUND PIXELS BASED ON OUR METHOD.	78
FIG. 31. THE DETECTION AND LABELING RESULTS AT THREE DIFFERENT TIME INSTANTS. FOR EACH CASE, THE IMAGES FROM THE TOP ARE THE TEST IMAGE, THE CAR LABELING WITHOUT SCENE KNOWLEDGE, THE CAR LABELING WITH SCENE KNOWLEDGE, THE SHADOW LABELING WITHOUT SCENE KNOWLEDGE, AND THE SHADOW LABELING WITH SCENE KNOWLEDGE.	79
FIG. 32. THE RECEIVER OPERATING CHARACTERISTIC (ROC) CURVES OF OUR METHOD, HUANG’S METHOD [46], WU’S METHOD [77], AND DAN’S METHOD [76], WITH THE VALUES OF THE AREA UNDER ROC (AUC) FOR (A)“DAY 1” (B)“DAY 2”, AND (C)“DAY 3” IMAGE SEQUENCES.	81
FIG. 33. THE PROPOSED DETECTION AND LABELING RESULTS AT THREE DIFFERENT TIME INSTANTS IN ANOTHER PARKING SPACE. FOR EACH CASE, THE IMAGES FROM THE LEFT ARE THE TEST IMAGE, THE PARKING SPACE DETECTION RESULTS, AND THE CAR LABELING RESULTS.	83
FIG. 34. SYSTEM FLOW OF THE PROPOSED SYSTEM.	92
FIG. 35. (A) VISUAL HULL CONSTRUCTED FROM THE FOREGROUND IMAGES OF TWO CAMERA VIEWS. (B) THE VOXEL HISTOGRAM BASED ON THE VISUAL HULL IN (A). (C) VISUAL HULL CONSTRUCTED FROM FRAGMENTED FOREGROUND IMAGES. (D) THE VOXEL HISTOGRAM BASED ON THE VISUAL HULL IN (C). (E) THE PROPOSED PILLAR MODEL IN THE 3-D SPACE. (F) THE ESTIMATED TDP DISTRIBUTION BASED ON THE	

FOREGROUND IMAGES IN (E). (THE RED BAR IN (B)(D)(F) REPRESENTS THE TRUE TARGET POSITION.).....	95
FIG. 36. (A) THE TDP OF FOUR MOVING TARGETS IN THE SURVEILLANCE ZONE.....	97
FIG. 37. AN ILLUSTRATION OF THE GHOST PROBLEM WHEN TRYING TO RECONSTRUCT A 3-D SCENE BASED ON TWO CAMERA VIEWS.	99
FIG. 38. (A) AN EXAMPLE OF TDP DISTRIBUTION FUSED FROM FOUR CAMERA VIEWS..	101
FIG. 39. (A) THE SCENE LAYER IN FIGURE 36 AND TWO OF THE FOUR CAMERA VIEWS. (B) THE COMBINATION $\{s_1, s_2, s_3, s_4, s_5\}=\{1,0,1,1,1\}$ AND THE EXPECTED FOREGROUND IMAGES OVERLAID WITH THE DETECTED FOREGROUND IMAGES. (C) THE COMBINATION $\{1,1,1,1,1\}$ AND THE EXPECTED FOREGROUND IMAGES OVERLAID WITH THE DETECTED FOREGROUND IMAGES.	103
FIG. 40. EXAMPLES OF $P(H_i(M,N) = T_k S)$	106
FIG. 41. ILLUSTRATION OF THE LABELING RESULTS. (A) TWO CAMERA VIEWS. (B) WITHOUT AND (C) WITH TARGET MODEL REFINEMENT.....	111
FIG. 42. ONE EXPERIMENT RESULT OF OUR LAB SEQUENCE. (A) FOUR CAMERA VIEWS. (B) FOREGROUND DETECTION IMAGES. (C) TDP DISTRIBUTION. (D) THE VOXEL HISTOGRAM BASED ON THE VISUAL HULL. (E) BIRD-EYE VIEW OF TARGET LOCATION. (F) LABELING AND CORRESPONDENCE OF TARGETS IN PSEUDO-COLOR.....	116
FIG. 43. ONE EXPERIMENT RESULT OF THE M2TRACKER SEQUENCE. (A) FOUR CAMERA VIEWS. (B) FOREGROUND DETECTION IMAGES. (C) TDP DISTRIBUTION. (D) THE VOXEL HISTOGRAM BASED ON THE VISUAL HULL. (E) BIRD-EYE VIEW OF TARGET LOCATION. (F) LABELING AND CORRESPONDENCE OF TARGETS IN PSEUDO-COLOR.	117
FIG. 44. ONE EXPERIMENT RESULT OF THE FLEURET'S SEQUENCE. (A) FOUR CAMERA VIEWS. (B) FOREGROUND DETECTION IMAGES. (C) TDP DISTRIBUTION. (D) THE VOXEL HISTOGRAM BASED ON THE VISUAL HULL. (E) BIRD-EYE VIEW OF TARGET LOCATION. (F) LABELING AND CORRESPONDENCE OF TARGETS IN PSEUDO-COLOR.	118
FIG. 45. THE MEAN DEVIATION PER FRAME FOR THE FLEURET'S DATASET.	119
FIG. 46. ONE EXAMPLE OF EXTENDED SURVEILLANCE ZONE. (A) FOUR CAMERA VIEWS. (B) THE TDP DISTRIBUTION. (C) BIRD-EYE VIEW OF TARGET LOCATION.....	120
FIG. 47. A COMPARISON OF THE MEAN DEVIATION OF EACH FRAME OVER THE M2TRACKER DATASET.	121
FIG. 48. THE DISTRIBUTIONS OF THE NUMBER OF DETECTED TARGET PER FRAME FOR THE 5-PERSON LAB DATASET. (A) RESULTS WITHOUT GHOST REMOVAL. (B) RESULTS WITH GHOST REMOVAL.	123
FIG. 49. MULTI-TARGET TRACKING RESULTS (A) M2TRACKER DATASET (4 PERSON). (B) LAB DATASET (5 PERSON).....	124

FIG. 50. THREE NORMAL VECTORS IN THE USN COORDINATE SYSTEM. 130



List of Notations

BHF	Bayesian hierarchical framework
I_L	Observation layer of BHF
H_L	Hidden labeling layer of BHF
S_L	Scene layer of BHF
H_L^*, S_L^*	The optimal solution pair of image content labeling and the 3-D scene parameters
$p(S_L)$	The prior knowledge of the 3-D scene statuses
$p(H_L S_L)$	3-D scene model representing the object-level constraints from scene layer
$p(I_L H_L)$	The data constraints from image observation
$E_D[I_L(m,n), H_L(m,n)]$	Classification energy model
$E_A[I_L(m,n), H_L(m,n); N_p]$	Adjacency energy model
N_p	Neighborhood around (m,n)
$G_S(U)$	Adaptive function for preserving the discontinuity
$Sigm(U)$	Logistic sigmoid function
(m,n)	Pixel coordinates
C and G	Car label and ground label

S and US	Shadowed label and unshadowed label
N_s	Number of parking spaces
$h^O(m,n)$	Object label at (m,n)
$h^L(m,n)$	Light label at (m,n)
$C_i(m,n)$	Expected car labeling map at (m,n) given the i th parking space being occupied
$G_i(m,n)$	Expected ground labeling map at (m,n) given the i th parking space being occupied
$S_i(m,n)$	Expected shadow labeling map at (m,n) given the i th parking space being occupied
$US_i(m,n)$	Expected non-shadow labeling map at (m,n) given the i th parking space is occupied
\mathbf{I}_{RGB}	RGB color features of a pixel
$\mathbf{I}_{\text{RGB}}^N$	Normalized \mathbf{I}_{RGB}
\mathbf{R}	A 3×3 matrix depending on surface reflectance
\mathbf{I}	A vector depending on illumination
$(D_X(t), D_Y(t), D_Z(t))^T$	The direction of sunlight
$G(X)$	Target Detection Probability (TDP)
F_i	Foreground detection result of the i th camera view
M_i	Projection image on the i th camera view
Ω_i	Normalized overlapping area between F_i and M_i
μ^k	Mean vector of the k th cluster
\mathbf{C}^k	Covariance matrix of the k th cluster
$p(R)$	Probability density function of target width
$p(H)$	Probability density function of target height

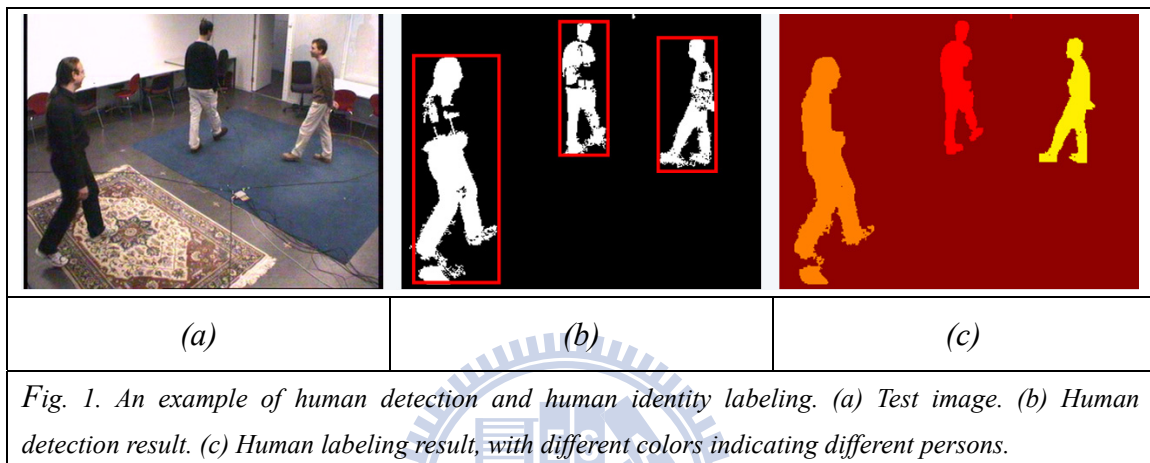
CHAPTER 1

Introduction

1.1 Overview

Recently, computer vision technology for video surveillance applications has made tremendous progress. Using an intelligent surveillance system to manage parking lots or to monitor security zones is becoming practical. To add more values to existing surveillance systems, various kinds of vision-based intelligent functionalities have been explosively proposed. For example, some algorithms provide user-friendly ways to help operators in the control room to monitor tens of, or even hundreds of, cameras; while a few others provide the capability to automatically detect unusual events in the surveillance zone. These vision-based algorithms may be roughly classified into single-camera based methods and multi-camera based methods. Among those methods, object detection and object labeling are two essential processes for subsequent analyses, like behavior modeling and scene modeling. Object detection, such as face detection and vehicle detection, is an object-level classification that tells

whether and where a specific object is inside an image. On the other hand, object labeling is an identity-level (ID-level) classification that determines the identity of each object region in the image. An example of human detection and human identity labeling is shown in Fig. 1. Even though it seems very easy and straightforward for human eyes to perform object detection and labeling, a robust computational algorithm for these two operations is actually not trivial at all.



For a single camera system, the captured 2-D image lacks the depth information and the detection of moving targets usually suffers from the occlusion problem, which makes it difficult to correctly label or segment connective targets. To deal with occlusion, some methods adopt multi-camera approaches. Even though the cross reference of multiple camera views may ease the occlusion problem and provide a more reliable way for object detection and labeling, the object correspondence among multiple cameras may become another thorny problem.

On the other hand, to detect foreground objects, the appearance ambiguity between the foreground objects and the surrounding background is a challenging issue that may fail many widely-used object detection algorithms. For example, some background subtraction algorithms, like [1][2], focus mainly on the modeling of background information. These algorithms work pretty well for scenes with stationary

background. However, they may detect incomplete foreground regions while the appearance of foreground objects happens to be similar to that of the background. To overcome this appearance ambiguity problem, simply relying on pixel-level image data would not be enough. Some other information, such as region-level messages and object-level messages, should be taken into consideration.

Besides occlusion and appearance ambiguity, the perspective distortion in 2-D images is also a challenging issue. An object far away from the camera and an object close to the camera would have quite different scales and shapes in the camera views. To overcome the perspective effect, some researches focused on invariant feature descriptors. In their approaches, they detect reliable feature points first and design appropriate feature descriptors for object classification. For example, difference of Gaussian (DoG) [3] and Harris-Laplace [4] operators are popular feature extraction operators. The SIFT (Scale Invariant Feature Transform) [5] descriptor is another widely-used operator that is invariant to illumination variation and affine transformation. Even though these operators perform quite well in detecting prominent features, they are still incapable of handling object labeling in complicated scenes.

Shadow effect and lighting variations are another two troublesome issues that degrade the robustness of present surveillance systems. Plentiful works have been proposed to solve these two problems. For example, Finlayson et al. [6] proposed an entropy minimization method to extract from an image the intrinsic image that is shadow-free. Matsushita et al. [7] proposed an illumination normalization method based on an off-line learned eigenspace to eliminate shadows. On the other hand, a few methods have been proposed to maintain reliable color appearance under varying illumination conditions. A review of these color constancy algorithms could be found in [8]. Moreover, in the last decade, the Bayesian approach and some learning-based

methods for color constancy have gotten great attention. A complete survey of Bayesian color constancy methods could be found in [9].

To overcome these aforementioned problems, like occlusion, appearance ambiguity, perspective effect, shadow effect, and lighting variations, we found most existing methods rely more on image observation but less on 3-D scene knowledge. In this dissertation, we focus on the inclusion of 3-D scene knowledge in object detection and object labeling. In our study, we found the usage of 3-D knowledge could be very helpful in handling these complicated issues. Moreover, from the aspect of system functionality, an important role of a practical surveillance system is to dynamically reveal the 3-D status of the surveillance zone. To achieve this functionality, a major task of an intelligent surveillance system would be to automatically infer the unknown 3-D status based on the observed images. In this dissertation, we propose a Bayesian hierarchical framework to realize the integration of 2-D image information and 3-D scene model in a unified and efficient manner for scene inference. The optimal inference of BHF provides a systematic way to resolve the image labeling problem and to find out the 3-D scene unknowns simultaneously. We also apply the framework to two real applications of video surveillance. By using the hierarchical framework to represent the image generation model in a probabilistic manner, our systems can systematically integrate useful information from pixel level, region level, and object level to achieve semantic inference of the 3-D environments.

1.2 Contribution

In this dissertation, by using a parametric form to represent the 3-D scene model with unknown variables, we propose a unified framework, named as Bayesian hierarchical framework (BHF), to accomplish object detection, object labeling, and 3-D scene inference, simultaneously. Based on the BHF framework, it becomes easier

to handle these aforementioned issues, like occlusion, appearance ambiguity, perspective effect, shadow effect, and lighting variations. Actually, under the BHF framework, occlusion effect, perspective effect, and shadow effect may even provide useful clues to support 3-D scene inference.

Moreover, the proposed BHF framework has been applied to two video surveillance systems: a vacant parking space detection system and a multi-camera surveillance system. In the vacant parking space detection system, the challenges come from dramatic luminance variations, shadow effect, perspective distortion, and the inter-occlusion among vehicles. With the proposed BHF, those challenging issues can be well modeled in a systematic way and can be effectively handled. Experimental results over a few outdoor parking lot videos show that the proposed framework can systematically detect vacant parking spaces, efficiently label ground and car regions, precisely locate shadowed regions, and effectively handle luminance variations. On the other hand, in the application of multi-target detection and tracking over a multi-camera system, the challenges come from the unknown target number, the inter-occlusion among targets, and the ghost effect caused by geometric ambiguity. Similarly, with the proposed BHF, the target labeling, target correspondence, and ghost removal are regarded as a unified optimal problem subject to 3-D scene priors, target priors, and image observations. Experimental results show that the proposed system can systematically determine the target number, efficiently label and correspond moving targets, precisely locate their 3-D locations, and effectively tackle the ghost problem.

1.3 Organization

The following chapters of this dissertation are organized as follows.

- ◆ In Chapter 2, we introduce various kinds of messages that have been commonly used for image analysis and scene modeling in video surveillance. Based on the coverage of the information, we classify these messages as pixel-level messages, region-level messages, and object-level messages.
- ◆ In Chapter 3, we detail the main idea of the proposed BHF framework and how we integrate various kinds of messages under this framework. In this chapter, we first introduce the modeling process in the proposed framework. After that, we depict the inference stage of the BHF framework which determines the optimal estimates of the system unknowns.
- ◆ In Chapter 4 and Chapter 5, we present the applications of the BHF framework to two different applications. In Chapter 4, we present how we develop a vacant parking space detection system based on the BHF framework. In Chapter 5, we present how we develop a multi-camera surveillance system to perform multi-target detection and tracking. In both systems, we explain how the BHF framework integrates the top-down information from 3-D scene models with the bottom-up message from image observations. The inference procedure of each system is also presented, together with a few experimental results over real scenes to demonstrate the feasibility of the proposed BHF framework.
- ◆ In Chapter 6, conclusions are drawn.

CHAPTER 2

Backgrounds

Object detection and object labeling have played an important role in the development of video systems. Some examples, like face detection, human detection, and vehicle detection, have been widely applied to various applications. Right now, a lot of digital cameras can perform automatic face detection while capturing photos. A few intelligent video surveillance systems can count the number of people in the scene based on human detection techniques. For modern intelligent transportation systems, automatic vehicle detection is also prevalent. In the literature, many image analysis works have been proposed to detect or label interested objects. In Section 2.1, we illustrate a few representative algorithms for object detection and labeling. According to the type of information used, these algorithms can be categorized into pixel-level methods, region-level methods, and object-level methods. Since the proposed BHF framework is designed to integrate pixel-level, region-level, and object-level information together, we will briefly review these three types of image analysis methods for object detection and labeling.

On the other hand, with the rapid development of computer vision techniques, scene modeling has attracted more and more attentions. In recent years, the concept of contextual analysis, which physically connects image analysis with scene knowledge, has been intensively studied to achieve improved detection performance. For instance, if we know a car is parked at a certain place and we also know the direction of sunlight, we would expect a shadowed pattern caused by the parked car. This kind of scene knowledge can be helpful in object detection and labeling. Hence, in this dissertation, another focus is to study the way to combine image analysis with the inference of unknown factors in the scene model. In Section 2.2, we will review a few relevant works that discuss the connections between image analysis and 3-D scene modeling.

2.1 Image Analysis Techniques

2.1.1 Pixel-level Methods

In most video surveillance systems, cameras are fixed. This static camera setting relaxes the difficulty of foreground object detection. Ideally, if we collect the color/intensity feature of a pixel over a temporal period, we may find, in most cases, the statistical property of the foreground color/intensity is somewhat different from that of the background color/intensity. Moreover, most of the period, the color/intensity feature at a pixel belongs to the background color/intensity. These two observations are the fundamental assumptions of many pixel-based background subtraction methods. Since background subtraction methods are simple and effective, this background modeling approach has become one of the popular tools in video surveillance applications.

The basic operation of background modeling is to dynamically learn the

temporal statistical property of every pixel. Based on the learned model, a pixel is classified as either a “background pixel” or a “foreground pixel” based on the current color/intensity observation at that pixel. Besides, the current observation is fed back to update the background model. By on-line learning the statistical property of the background color/intensity, this background modeling method can efficiently extract foreground regions from the background. Currently, several efforts have been proposed for the modeling of time-varying background. Some simpler methods used the 1st order and 2nd order statistics to model the temporal property of a pixel [104]. In these simple approaches, a pixel with its color/intensity feature far away from the mean value is classified as a foreground pixel.

On the other hand, some methods used more complicated parametric forms to model the dynamic statistics of the color/intensity feature at a pixel. Among those methods, the Gaussian mixture model (GMM) has been widely studied and has been proved to be a useful form for background modeling [2]. In principle, the distribution of a pixel value (x) over the temporal (t) direction is formulated as

$$p(x(t)) = \sum_{i=1}^K w_i(t) \times g_{au}(x(t), \mu_i, \sigma_i), \quad (1)$$

where $p(x(t))$ is the probability of observing the current pixel value $x(t)$, $w_i(t)$ is an estimate of the weight of the i th Gaussian function $g_{au}(\cdot)$ in the mixture model at Time t . μ_i and σ_i are the mean value and the standard deviation value of the i th Gaussian in the mixture at Time t . An example of the probability distribution of a pixel with a Gaussian mixture model is shown in Fig. 2.

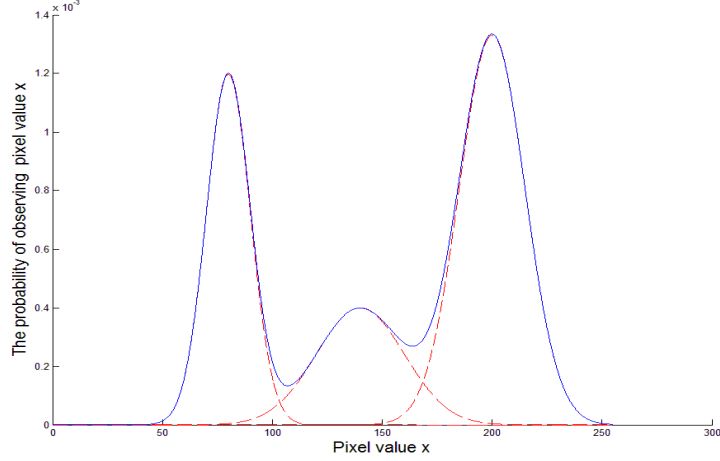


Fig. 2. The probability distribution of a pixel with Gaussian mixture model [2].

To classify a pixel into either a background pixel or a foreground pixel, Stauffer-Grimson [1] suggested firstly separating the K Gaussian distributions into background Gaussians and foreground Gaussians. Those pixels belonged to background Gaussians are determined as background pixels, and vice versa. To separate the K Gaussian distributions, the ratio w_i / σ_i of each Gaussian distribution are calculated and is used to rank the K Gaussian distributions from small to large. The first B Gaussian distributions, whose summation of their probability weights exceeds a threshold T , are treated as background Gaussians. This is formulated as

$$B = \arg \min_b \left(\sum_{i=1}^b w_i > T \right). \quad (2)$$

On the other hand, the parameter sets $\{ \mu_i, \sigma_i, w_i \}$ are dynamically updated over time to adapt to the environmental variation. By using a recursive filter to approximate the online Expectation-maximization (EM) algorithm [1], the parameter sets are updated based on the following formulation:

$$\beta(t) = (1 - \lambda(t))\beta(t-1) + \lambda(t)Q(x(t), \beta(t-1)). \quad (3)$$

Here, $\beta(t)$ could be any model parameter of $\{ \mu_i, \sigma_i, w_i \}$ at Time t , $\lambda(t)$ is the parameter learning rate, and $x(t)$ is the new observation at Time t . The function $Q(\cdot)$ is a prediction of the model parameter $\beta(t)$ at Time t based on $x(t)$ and the previous

parameter $\beta(t-1)$. In Fig. 3, we show the detection results based on the Gaussian mixture method.

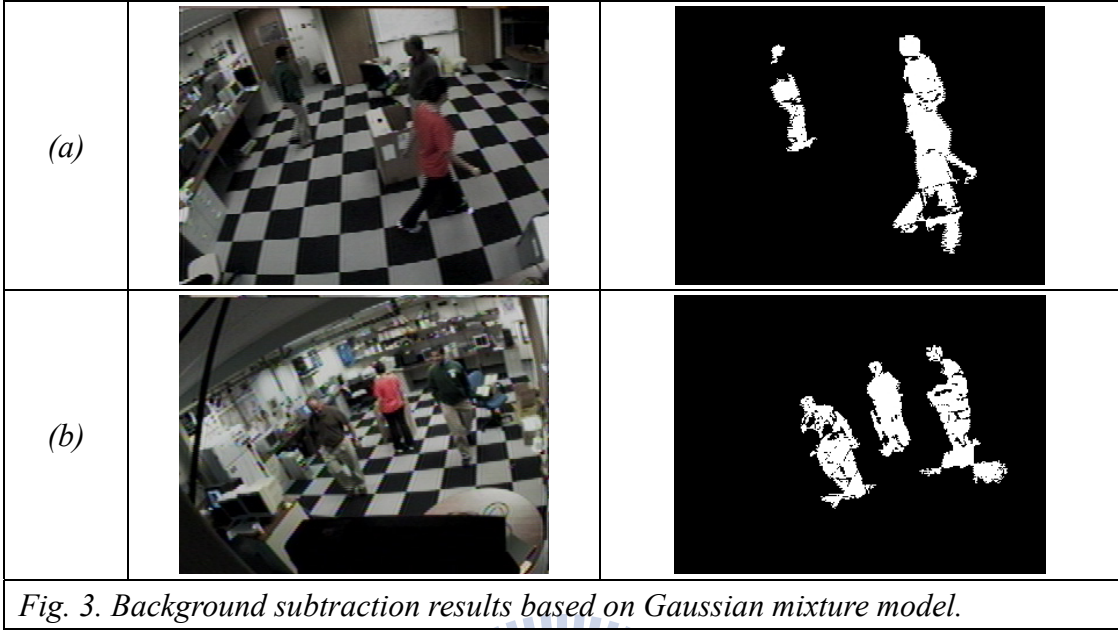


Fig. 3. Background subtraction results based on Gaussian mixture model.

Instead of using a parametric form to model the statistical property of a background pixel, Elgammal et al. [10] proposed the description of a background model based on non-parametric kernel density estimation. In their method, the pixel-wise statistical property along the temporal direction is modeled by a kernel density function. Given N successive intensity values $B_x = \{x_1, x_2, \dots, x_N\}$ along a temporal period at a pixel, they estimate the probability density function (pdf) to be

$$p(x_t | B_x) = \frac{1}{N} \sum_{i=1}^N K_{BW}(x_t - x_i). \quad (4)$$

Here, x_t represents an intensity value. K_{BW} is the kernel function with bandwidth BW . By assuming that most of the intensity values inside the observed time period belong to the background, a pixel with a smaller probability value $p(x_t)$ is more likely to be a foreground pixel. To adapt to the environmental variation over time, this algorithm simply shifts the time window to update samples for the estimation of the pdf function.

To overcome the appearance variations caused by surrounding lighting, a few

researchers try to record all possible forms of the background images and then dynamically select the most suitable background image from the stored background image database. Obviously, it would be inefficient to directly store all possible background images in a large database. Hence, Funck et al. [11] assumed that the background images would form a Euklidian subspace within the space formed by all image pixels. By applying the Principal Component Analysis (PCA) technique to calculate the major principal components, any background image could be represented as a linear combination of the derived eigen-backgrounds. With this eigen-background representation, any input image is firstly projected onto the background subspace to find the most matched background image. By subtracting the matched background image from the input image, foreground objects are identified.

Even though the detection of foreground objects based on pixel-level background modeling works pretty well for a scene with stationary background, this approach has difficulty in handling the occasional appearance ambiguity between a foreground object and its surrounding background [12]. When a foreground object happens to have an appearance similar to that of the surrounding background, the background model may not be enough for foreground/background discrimination. Hence, instead of focusing on the background model, some other researchers proposed the learning of the foreground target model. For instance, Tsai et al. [13] developed a probabilistic method to model a pixel-level car model in the chromatic domain. In their method, the RGB color features of many “car” pixels are collected and converted to a new color domain based on the following transformation.

$$\begin{aligned}
 Z &= (R + G + B) / 3 \\
 u &= (2Z - G - B) / Z \\
 p &= \text{Max}\{Z - G / Z, (Z - B) / Z\}
 \end{aligned}
 \tag{5}$$

To combat the luminance variation problem, only the chromatic information (u, p) is used. The brightness value Z is ignored. Based on the finding in [13], the chromatic

values of the “car” pixels cluster compactly in the u - p color space. This cluster can be approximated by a Gaussian function:

$$P(x_c | car) = \frac{1}{2\pi\sqrt{|\Sigma_c|}} \exp\left(-\frac{1}{2}(x_c - m_c)\Sigma_c^{-1}(x_c - m_c)^t\right) \quad (6)$$

where $x_c = (u, p)$ is the chromatic feature of a pixel x , m_c is the estimated chromatic mean based on the training set of “car” pixels, and Σ_c is the estimated chromatic covariance matrix. Based on the car probability model in (6), the probability of being a “car” pixel at a pixel with the chromatic feature x_c can be evaluated.

2.1.2 Region-level Methods

Because of its abilities to adapt to the background variations over time and to cope with multi-modal background distributions, the aforementioned pixel-level modeling has achieved its success in foreground object detection and labeling. Besides, the background modeling approach can handle the situations of new comers and the leave of existing objects. However, in an outdoor scene, occasional camera shaking and the swinging trees caused by strong wind may sometimes seriously degrade the performance of object detection and labeling. In order to improve the performance, some region-level methods have been proposed for image analysis in the literature.

In region-level methods, some researchers extended the concept of GMM to develop new background subtraction methods that incorporate region-level information. For example, Heikkilä et al. [14] tried to model the temporal statistics of a small region to capture the textural information. In [14], local binary patterns were proposed to efficiently extract the texture features of a small region which are invariant to lighting changes. By modeling the dynamic variation of those texture features along the temporal direction, their system outperforms the traditional GMM

background subtraction in an outdoor scene, where the trees were swinging and the camera was shaking. In Fig. 4, we show the detection results based on their method.

[14].

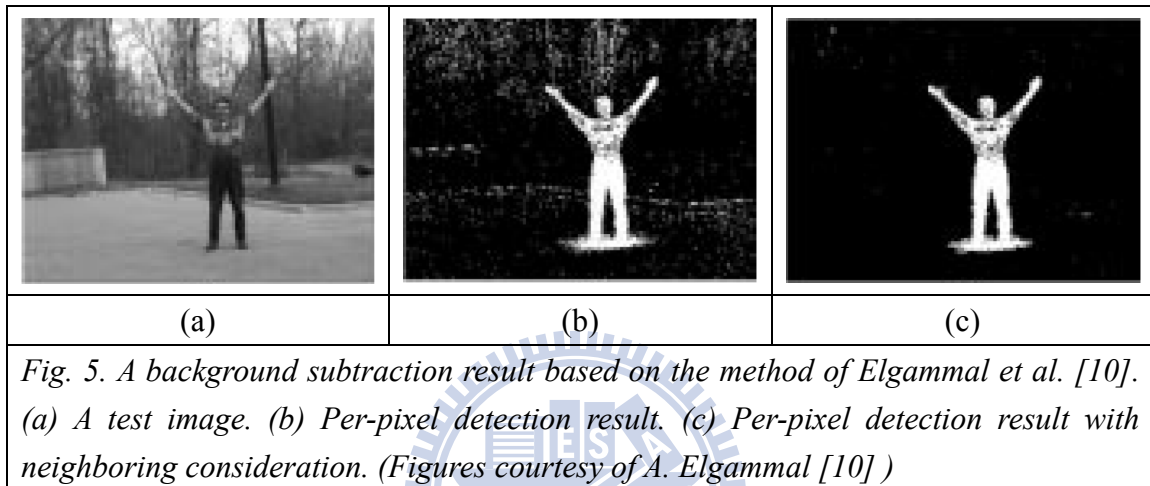


Fig. 4. The background subtraction results based on the method proposed by Heikkilä et al. [14]. The first and third rows are the test images. The second and fourth rows are the detection results. (Figures courtesy of Marko Heikkilä [14])

Compared with GMM background modeling, non-parametric kernel based modeling relaxes the constraint of a GMM pdf function and may sometimes provide a more compact match with the true distribution. However, the original non-parametric method is still a pixel-based approach and may suffer from the aforementioned non-stationary effect like camera shaking and swinging trees. In [10], Elgammal et al. suggested an approach that takes into account the background models of neighboring pixels. This is due to the thinking that the intensity value x_t at the current pixel may actually belong to a neighboring pixel at the previous moment. In their approach, they calculated the following probability

$$p_N(x_t) = \max_{y \in N(x)} p(x_t | B_y), \quad (7)$$

where y is a pixel belongs to the neighborhood of the target pixel x , x_t is the intensity value at x , and B_y is the intensity set for the pdf estimation of Pixel y . The distribution $p(x_t | B_y)$ is estimated by the non-parametric formula in (4). By comparing $p_N(x_t)$ with a pre-defined threshold, foreground pixels are determined. A detection result based on the method of [10] is shown in Fig. 5.



Some methods suggested maintaining a region-based foreground model and a background model at the same time for object detection and labeling. A simplest setting is to use a uniform distribution over the feature domain to model the foreground model, as used in [15]. Obviously, the uniform foreground model cannot well capture the foreground property. Hence, Sheikh and Shah [16] expended the original non-parametric kernel density modeling in [10] with some modifications. First, both the foreground model and the background model are dynamically maintained in order to reduce the effect of appearance ambiguity. In their approach, it was assumed that foreground objects tend to have consistent appearance and high spatial correlation in successive frames as long as the video frame rate is high enough. With this assumption, the foreground detection results of the previous frames can be used to establish the foreground model of the current frame. Moreover, in their hybrid

modeling, the background and foreground models compete with each other for a better detection without the need of a manually selected threshold. Second, a new non-parametric kernel density estimation of the probability model over the domain (location) space and the range (color) space is proposed. Rather than modeling the color space only, the integration of the color space and the location space makes it easier to handle non-stationary background in an outdoor scene. In their method, by combining the spatial location x and the pixel color values x_{rgb} into a random vector $\vec{d}=(x, x_{rgb})$, the joint domain-range probability is defined as

$$p(\vec{d}|\Omega_C)=\frac{1}{N}\sum_{i=1}^N\phi_{BW}(\vec{d}-\vec{d}_C^i). \quad (8)$$

Here, $\Omega_C=\{\vec{d}_C^1,\vec{d}_C^2,\dots,\vec{d}_C^N\}$ is the training set with N domain-range training data of some class C . In [16], the class C could be foreground (C_F) or background (C_B). ϕ_{BW} is the domain-range kernel function with bandwidth BW . While calculating the class probability of a pixel x , Ω_C directly embedded the information from neighboring pixels to contribute the support of the class C . With this design, the non-stationary statistical properties caused by winds or other factors can be overcome.

In some surveillance systems, the video frame rate is low and unstable due to the limited transmission bandwidth or the limited storage. In this kind of surveillance systems, the temporal persistence property required in [16] becomes unreliable. This fact makes foreground modeling difficult. One possible way to model the foreground model would be to exploit the region-level information in the current image. Based on the spatial statistics of the neighboring regions of a pixel, Benedek et al. proposed a method in [17] to build the foreground model of that pixel. They assumed a foreground pixel shares a similar appearance with the other foreground pixels around it. The procedure of the foreground modeling in [17] is illustrated in Fig. 6. To model $p_f(X_S|S)$, the foreground probability of a pixel S with the color intensity X_S , a

manually-defined window V_S centered at S is selected as shown in Fig. 6(a). A rough foreground region F is extracted by background subtraction, as shown in Fig. 6(b). In Fig. 6(c), the intersection region of F and V_S is denoted as F_S . The histogram of F_S is presented in Fig. 6(d). Those pixels whose intensities are within the range $[X_S - \tau, X_S + \tau]$ are collected for the training of a Gaussian foreground model, as shown in Fig. 6(e). Compared with the uniform foreground model, which gets a likelihood value 2.71 for X_S in this example, the spatial statistics based foreground modeling gives a likelihood value 4.03 for X_S which apparently better represents the foreground property. In Fig. 6(f) and Fig. 6(g), the final detection results are compared based on the uniform foreground model and the spatial statistics based foreground model, respectively. Note that the gray color represents the shadow regions.

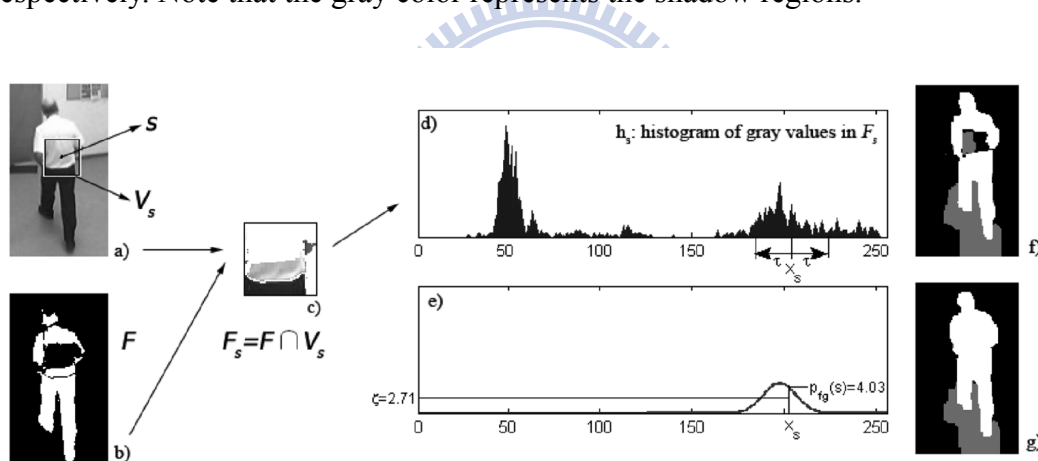


Fig. 6. The procedure for the foreground modeling in [17] based on the spatial statistics. (Figures courtesy of Cs. Benedek [17])

Another kind of region-level information is the expansion of spatial similarity. Statistically, adjacent points tend to belong to the same class, especially when the adjacent points share similar appearance. This property is sometimes named the “smooth constraint” of neighboring regions in the literature. To consider spatial similarity while doing image analysis, a popular way is to adopt Markov random field (MRF) model [18][19][20]. In MRF, the smooth constraint is modeled by the clique

potential among neighboring sites. Unlike many previous works which directly assign a suitable class to a pixel, the clique potential only requires the labels of neighboring pixels to fulfill the smooth constraint. Hence, a typical form of MRF usually involves an extra constraint (the data term), which defines the cost of assigning different labels for a pixel, to cooperate with the clique potential. By combining the data term and smoothness term, the MRF provides a flexible framework to integrate pixel-level information and region-level information for image analysis.

2.1.3 Object-level Methods

Instead of using pixel-level information to classify local pixels or using region-level information to group neighboring pixels through the use of MRF technique or some other grouping techniques like connected component analysis [23], a few other methods suggest to directly learn the unique object-level property of an object class for detection and labeling. Once the properties of an object class are well learned and modeled, a popular way to detect the interested target is to scan through the image by using a sliding window, as shown in Fig. 7. The discriminative properties of an object class are used to verify whether a target is inside the sliding window. Instead of merging local information to reach the final decision, those object-level methods use the object-level information as a whole for detection and labeling. A typical face detection procedure is illustrated in Fig. 7 as an example. A test window is first selected and the object features inside the window are calculated. By comparing the object features with respect to the object model and the non-object model, we decide whether an object could be found inside the window.

A crucial step for object-level detection is the extraction of object-level information. A systematic way to find the discriminative features of an object class is to analyze a labeled training dataset based on a learning process. Through the learning

process, the object-level information, usually named as trained object model, is extracted and used for the detection task.

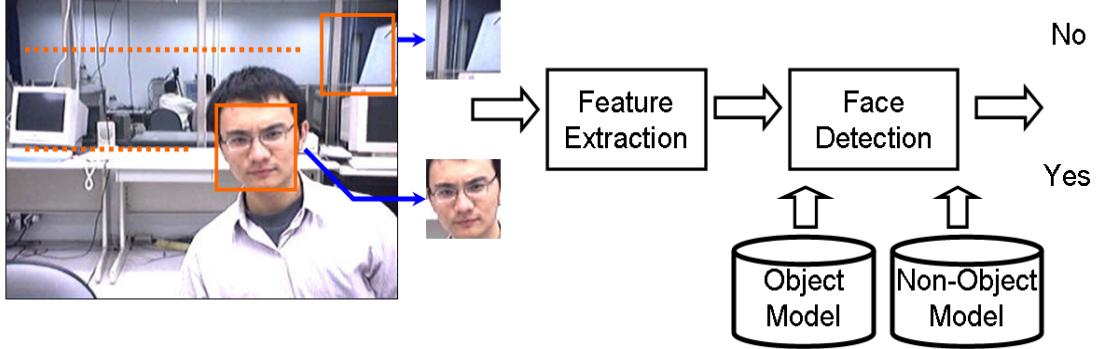


Fig. 7. A typical object-based detection procedure with sliding window. Here, we use face detection as an example.

Support vector machine (SVM) is a popular technique to train object models in the field of machine learning [24][25]. Given a set of training examples composed of positive examples and negative examples, the main operation of SVM is to search an optimal hyperplane that separates positive examples from negative examples with a maximum margin. The optimal hyperplane could be expressed as

$$f(x_o) = w^T \phi(x_o) + b. \quad (9)$$

In (9), x_o is the features calculated from an image patch (window). $\phi(\cdot)$ is a nonlinear mapping function to map the input features into a higher dimensional space \mathbf{H} . (w, b) are the major parameters controlling the direction and shift of a hyperplane. In SVM, (w, b) are the major factors to learn. They are only determined by the support vectors, which are the borderline training examples in the dataset. Those support vectors represent almost all the information of the training dataset. We may treat these support vectors as the extracted object-level information learned from the SVM training process. The positive support vectors define the object model while the negative support vectors define the non-object model. In Fig. 8, we illustrate the concept of

SVM classification.

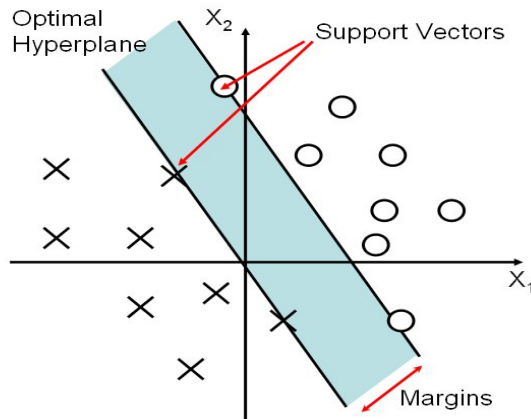


Fig. 8. Illustration of SVM classification with a hyperplane that separates positive examples (“+”s) from negative examples (“O”s) with the maximum margin. Support vectors are parts of the training examples that lie on the boundary.

In the literature, the feature x_0 in (9) calculated from an image window has played an important role in the performance of object detection. In general, a good feature is required to be invariant to illumination variations. Recently, a few features are commonly used, including the cascaded raw color pixel over the window [26], the wavelet-like features [27], and the histogram of oriented gradients (HOG) [28].

In object detection, a major difficulty is the need to deal with various kinds of variations, like appearance variation or shape variation. In a practice system, variations mainly come from intra-class difference, environmental illumination, and object deformation. To achieve robust object detection, a sophisticated but flexible object model is needed. However, an object-level model learned from the typical SVM procedure is more like to be a rigid template. While dealing with non-rigid object detection, the typical SVM-based object model may not be a proper solution. To overcome non-rigid deformation, the pictorial structure framework was first proposed in [29] and then extended by [30][31][32]. As illustrated in Fig. 9, the pictorial structure framework represents an object model by a set of parts that are

located in a deformable manner. Each part captures some local appearance properties of an object. Deformable models is also learned to characterize the spring-like connections between each pair of individual parts.

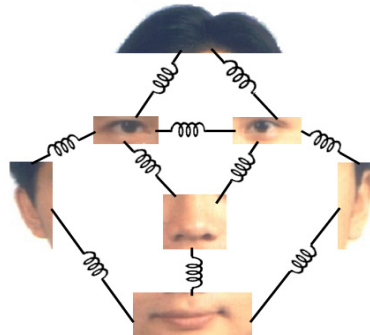


Fig. 9. The pictorial structure framework proposed by Fischler and Elschlager [29].

To learn a part-based object model based on a typical training dataset, where the positive examples are only selected by bounding boxes without any training information of the object parts, the SVM learning procedure would not be suitable. This is because the locations of object parts in each positive example are latent and unknown for training. In [32], Felzenszwalb et al. adopted the latent SVM [33] to handle latent factors. In latent SVM, the first step of the learning procedure is to maximize over latent part locations to find out the optimal part locations for each positive example based on a learned object model in the current iteration. The second step is to refine the object model based on the training dataset and the optimal part locations found in Step one. These two steps are iteratively performed until the final object model converges. As an example, we show a human model with its part models and deformable models in Fig. 10(c)(d). Here, HOG is used as the feature in this example. The deformable models allow each part to deviate from a reference location and can adapt to the variation caused by deformation. In Fig. 10(a), a result of human detection shows the deviation of each body part.

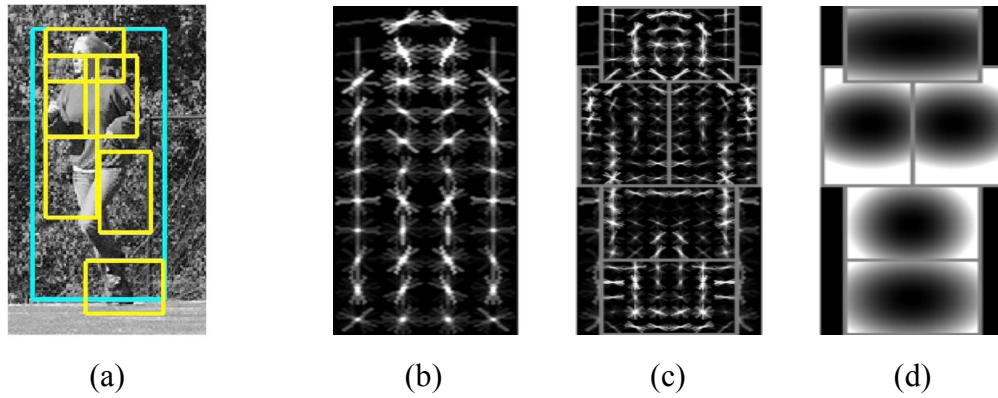


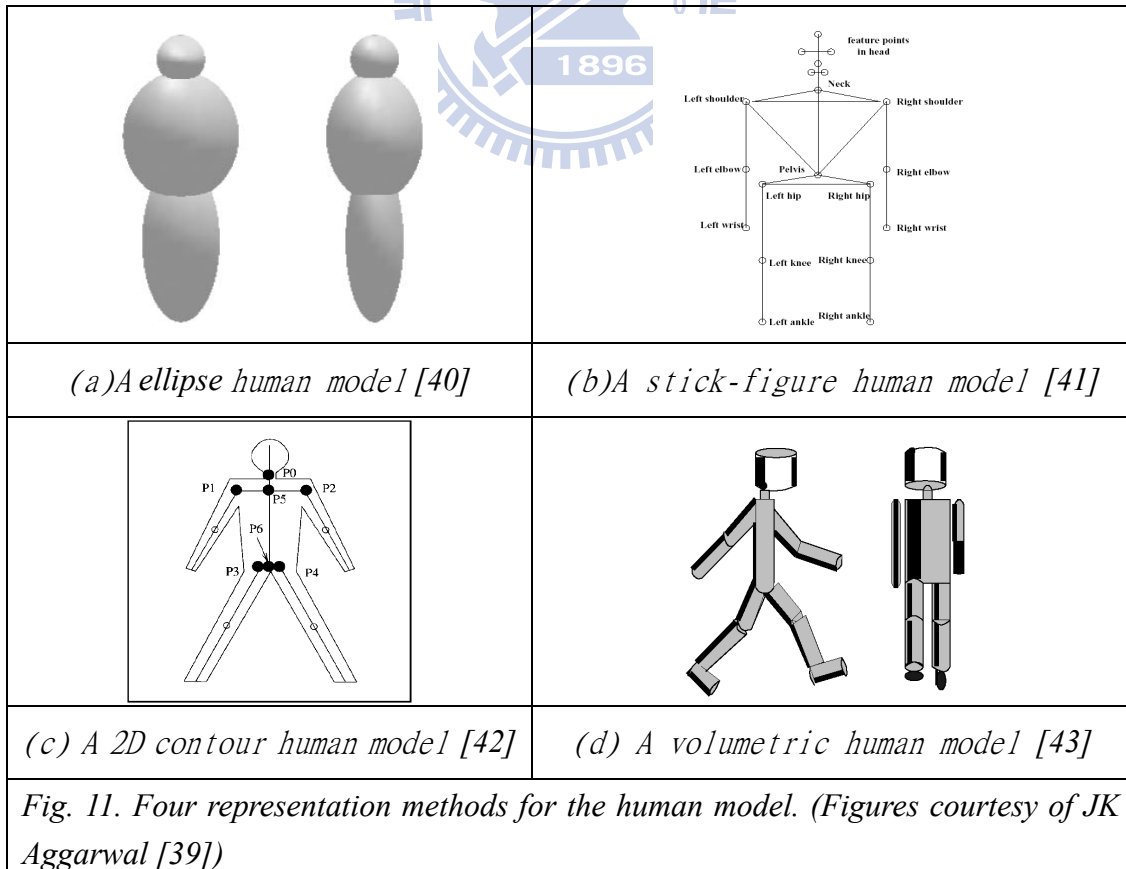
Fig. 10. The part-based object detection reported in [32]. (a) A tested image with the detected human and its parts. (b) The HOG human model. (c) The HOG models of each body parts. (d) The deformable models depicting the possible variation of each part. (Figures courtesy of P. Felzenszwalb [32])

If looking into the SVM learning procedure, we may find the SVM procedure “equally” takes into account the entire local feature space to maximize the margin while minimizing the number of incorrectly classified examples. Hence, the object model learned by the SVM process gives an equal weight to each local property of the object. However, different local area may have different degrees of discriminability. This brings the idea of feature selection while learning an object model. The AdaBoost technique [34][35][36] is a successful method, which incorporates feature selection into object model learning with a unified training procedure. Instead of combining many features with an equal weight like SVM, the AdaBoost procedure selects a few but important features to represent object information and creates a sparse classification rule for object detection.

A main feature of AdaBoost is its ability to select the discriminative features. This is achieved by dynamically adjust the weights of each training sample. However, a typical AdaBoost algorithm does not put too much effort on the combination of local features. On the contrary, SVM method put more effort on the combination of local features through the use of different kernels. Recently, a few research works [37][38]

focus on the integration of AdaBoost algorithm and SVM method. The AdaBoost algorithm is used to select discriminative features for the object detection while the SVM process is used to determine the final classifier by fusing the selected features.

Besides utilizing a learning-based method to obtain object-level information, some previous works directly design the specific target body structure for detection. For instance, human is a very important class for video surveillance systems. Hence, many works have been proposed to design a suitable representation for human detection. Basically, the proposed human model is composed of some simple elements, such as blobs, pillars, ellipses, and cylinders. The conventional human representations include the ellipse model [40], the stick figure model [41], the 2-D contour model [42], and the volumetric model [43]. After the body structure is defined, object detection is accomplished by fitting the target structure model to the image observation. In Fig. 11, we illustrate a few commonly used representations for the human model.



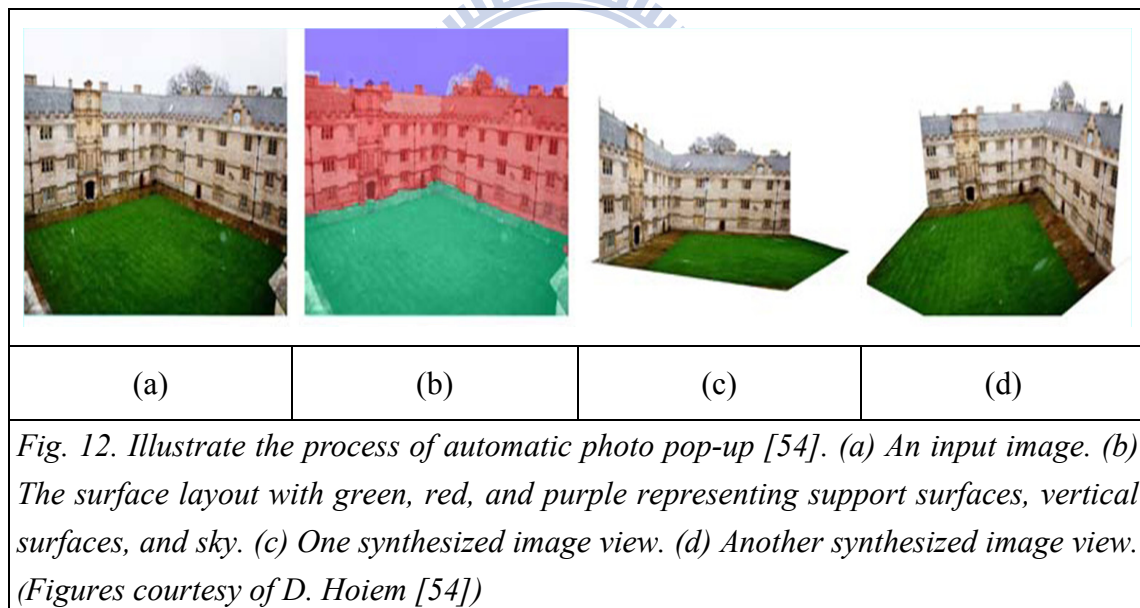
2.2 Connection between Image Analysis and 3D Scene Modeling

Besides using pixel-level, region-level, and object-level information for image analysis, another useful clue is to rely on the prior knowledge of 3-D scene. For instance, in a lobby, we would expect a few people walking on the ground plane. Based on this prior knowledge plus an appropriate 3-D human model, object detection may become more stable, as reported in [44] and [45]. On the other hand, for a typical parking lot, we may know the 3-D layout of the parking spaces. Based on this prior knowledge plus suitable 3-D car models, the detection of parked cars may become more robust and reliable [46].

The study of the connection between vision analysis and scene modeling has a long history. In the 19th century, James Gibson [47] proposed that scene surfaces constitute the fundamental of human vision. Human vision can perceive the depth and distance mainly depending on the perception of longitudinal surfaces. Warren [48] also believed that human vision can fully understand the 3-D scene not only based on image observation but also based on lots of visual experiences in daily life. The visual experiences drive human beings to utilize clues, such as horizontal line, shadow, and some familiar objects, to infer the status of the 3-D scene. Moreover, Koenderink et al. [49][50] found that the participants of their experiments could not measure the depth order of two points in the scene unless there is a scene surface connecting these two points. Those findings suggested that physical surfaces provide valuable information for scene interpretation.

In the recent study of video surveillance techniques, an example of utilizing surface information to improve system accuracy is the use of the 3-D prior that human stands on the ground plane [40]. Based on this assumption, Object detection and

tracking become more robust. Moreover, Hoiem et al. [51][52] believed the extraction of the surface layout in an image is a right way to interpret the 3-D scene. Hence, they proposed a learning based method to assign each image pixel a geometric class. To find out the surface layout, Hoiem et al. [52] firstly over-segmented an image observation. Each segmented region was named as a super-pixel. By merging similar super-pixels based on some local features, like color, texture, location and shape, their algorithm generated a large set of segmented regions. The learned surface models were utilized to assign a surface class to each segmented region. Once the surface layout is extracted, Hoiem et al. [53][54] used the surface knowledge to reconstruct 3-D view based on a single image. In Fig. 12., we illustrate the automatic photo pop-up with the help of the extracted surface layout.



On the other hand, 3-D depth knowledge and camera viewpoint are also valuable information for object detection. In general, the camera viewpoint is available if the intrinsic and extrinsic parameters of the camera are available. Furthermore, for a practical video surveillance system, the inter-object occlusion would be a challenge issue. If the depth order of objects could be known in advance, it becomes easier to

handle the inter-object occlusion problem. In [55], Sudderth et al. integrated the depth information to achieve high detection performance. In [56], Hoiem et al. proposed to combine the information of surface layout, depth order, and camera viewpoint to support object detection. The results are shown in Fig. 13. By using the scene knowledge, lots of unlikely detection results are removed.

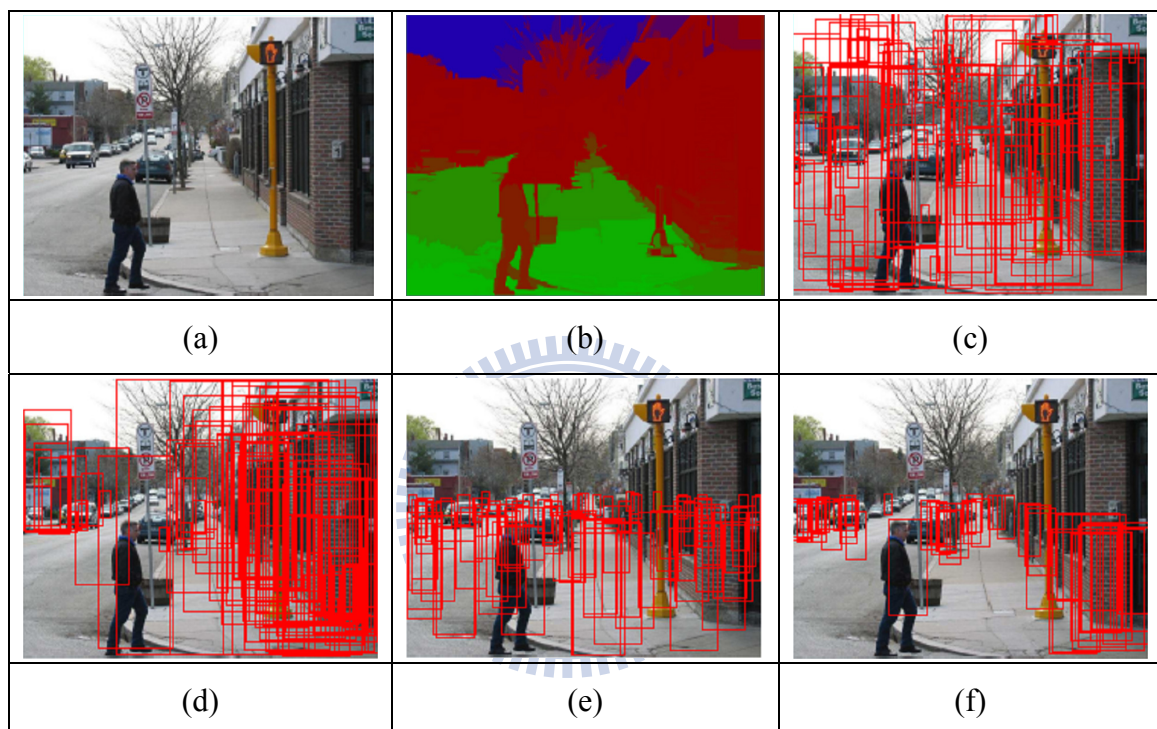


Fig. 13. Human detection based on scene knowledge [53][56]. (a) An input image. (b) The surface layout with green, red, and blue representing support surfaces, vertical surfaces, and sky. (c) Detection without scene information. The detection windows are uniformly distributed in image. (d) Detection with the prior of surface layout. The detection windows are mainly distributed in the “vertical” surfaces. (e) Detection with the prior of depth and camera viewpoint. The detection windows are larger in the near distance. (f) Detection with the prior of surface layout, depth and camera viewpoint. The detection windows are fewer and more accurate. (Figures courtesy of D. Hoiem [56])

Some researches tried to estimation the depth map from a single image. Oliva and Torralba [57] found some image local properties, such as naturalness, openness,

roughness, expansion, and ruggedness, are directly relevant to the 3-D depth. By measuring those local properties from a single image, a rough depth map could be estimated. Moreover, Saxena [58] proposed an MRF-based framework to integrate local image properties to infer the depth map. The extracted depth order is then utilized to help image analysis.

On the other hand, instead of using scene knowledge, like scene surfaces or depth order, to help object detection, some researchers began to think in the opposite way. Sudderth et al. [59] suggested that by understanding the relations among multi-targets, the depth information can be derived.

In this dissertation, we study another possibility to combine image analysis and scene modeling in a unified framework. According to the findings of these previous works, image analysis and scene modeling are highly relative and are complementary to each other. However, in a practical video surveillance system, we usually have some unknowns in both 3-D scene model and 2-D image contents. This drives us to propose the Bayesian Hierarchical Framework (BHF) to simultaneously infer the status of 3-D scene model and label objects in the image domain.

CHAPTER 3

Bayesian Hierarchical Framework

3.1 The Structure of BHF

The proposed BHF has a 3-layer graphical structure as illustrated in Fig. 14. In order to perform the inference of 3-D scene status based on image observations, we include a scene layer and an observation layer in the structure. However, if only using a 2-layer graphical structure, it would be difficult for our system to clearly depict the generation of image appearance based on a parametric 3-D scene model. This is due to the fact that a parametric 3-D scene models can only generate geometric patterns and labeling layout rather than image color appearance. For this reason, we introduce a hidden labeling layer between the scene layer and the observation layer. With the insertion of the labeling layer, the prior knowledge of the 3-D scene can be propagated down to the labeling layer, while the information from the image observation can be propagated upward to efficiently affect the labeling layer. As a crucial medium, the labeling layer not only enables the communication between the

scene layer and the observation layer but also facilitates the integration of 2-D image information and 3-D scene model in a unified manner.

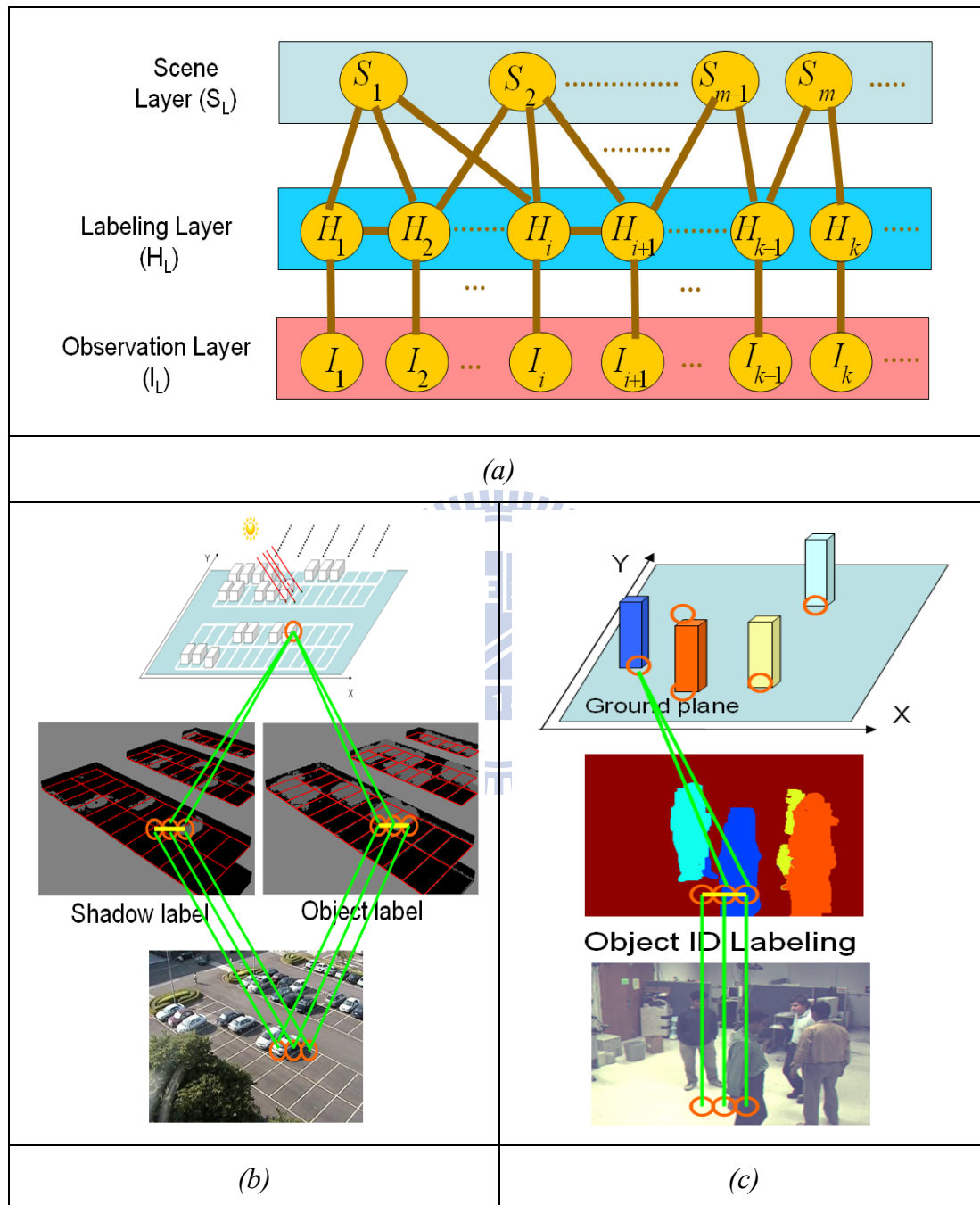


Fig. 14. (a) The proposed Bayesian hierarchical framework (BHF). (b) BHF for the vacant parking space detection system. (c) BHF for the multi-target multi-camera surveillance system.

In Fig. 14 (b)(c), we use “the vacant parking space detection system” and “the

multi-target multi-camera surveillance system” as two examples to illustrate the application-oriented definition of each node in the proposed BHF as shown in Fig. 14 (a). In detail, in the observation layer (I_L), each node indicates a local feature. The local feature can be either region-based, like a gradient feature, or pixel-based, like a color feature. In Fig. 14 (b)(c), each observation node in both systems represents the color feature of a corresponding pixel in the observation image. In the hidden labeling layer (H_L), each node represents the semantic status of a local region or an image pixel. Here, for “the vacant parking space detection system”, each labeling node represents a labeling pixel with four possible statuses, (“car pixel”, “shadowed pixel”), (“car pixel, un-shadowed pixel”), (“ground pixel”, “shadowed pixel”), and (“ground pixel, un-shadowed pixel”). For “the multi-target multi-camera surveillance system”, each labeling node represents the object identity which the corresponding image pixel belongs to. In Fig. 14 (c), the labeling statuses could be “object 1 (marked by blue color)”, “object 2 (marked by orange color)”, “object 3 (marked by yellow color)”, “object 4 (marked by light blue color)”, and “background object (marked by red color)”. On the other hand, the scene layer (S_L) indicates the unknown 3-D scene statuses that are to be inferred. For “the vacant parking space detection system”, each node in the scene layer represents the status of a corresponding parking space. It could be “vacant space” or “occupied space”. For “the multi-target multi-camera surveillance system”, each node in the scene layer represents if a moving target exists at a specific position. The links between the observation layer (I_L) and the hidden labeling layer (H_L) convey the bottom-up information from the image observation, while the links between the scene layer (S_L) and the hidden labeling layer (H_L) convey the top-down messages from the scene knowledge and the trainable target models. In the middle layer, the links between adjacent nodes convey a smooth constraint to model the high correlation in a local neighborhood. Based on this three-layer

framework, we are able to construct the generation models from the scene level to the labeling level and from the labeling level to the observation level.

3.2 The Property of BHF

In the literature, commonly used methods for object detection, object labeling and scene modeling can be roughly divided into three categories --- data-driven methods, model-driven methods, and hybrid methods. In general, data-driven methods directly use region-level and pixel-level information from the image data to support image analysis and the inference of the 3-D scene; while model-driven methods use a few object-based models pre-learned from training data to infer the scene statuses and to detect interested objects. On the other hand, hybrid methods are proposed to combine both image information and object knowledge for image analysis.

In this dissertation, the proposed BHF framework is a hybrid method. As shown in Fig. 14, the message stream propagated upward from the observation layer is considered as data-driven information; while the message stream propagated downward from the scene layer is considered as model-driven knowledge. This BHF framework has quite different properties if compared with either data-driven methods or model-driven methods. On the other hand, if compared with existing hybrid methods, the BHF framework proposes a new way to integrate pixel-level, region-level and object-level information under a unified framework. A few distinctive properties of the proposed BHF framework are to be explained as follows.

3.2.1 Differences to Data-driven and Model-driven Methods

Compared with data-driven methods and model-driven methods, a distinctive feature of the proposed BHF is the integration of object-level information from 3-D scene, region-level constraints in 2-D image patches, and pixel-level features from image pixels in a unified framework, as presented in Fig. 14. The main characteristics of BHF has two aspects: (a) a unified framework to combine pixel-level, region-level, and object-level information together to represent the generation process from 3-D scene to 2-D image; and (b) a systematic procedure to simultaneously analyze 2-D images and infer 3-D scene statuses.

For most bottom-up methods, the process usually begins at the classification of each pixel into a target pixel or a non-target pixel. Since the pixels of a target usually share similar appearance, these methods merge target pixels into target regions based on region-level information in the image. However, when the appearance of a target region happens to be similar to that of the background, the appearance ambiguity causes the extracted target regions to be fragmental and incomplete. If the incomplete target regions are used to infer the 3-D scene statuses, the system accuracy will be deteriorated. Without using object-level information, data-driven methods usually suffer from poor accuracy in object detection and labeling.

On the other hand, for most top-down methods, the process usually begins at the training of a suitable object-based classifier. After the setting of the classifier is learned, the process can detect interested targets via the classification of image patches. Those object-based detection methods can obtain a complete detection result without fragments, but may lose the accurate silhouette of the interested targets. Furthermore, when there are multiple targets inside the 3-D scene, the occlusions among targets could be crucial and may cause difficulty in object detection and labeling.

In this dissertation, the proposed 3-layer BHF includes a scene layer for

object-based information, an image observation layer for pixel-based information, and a labeling layer in the middle. This framework efficiently integrates top-down information with bottom-up messages. Based on the integration, top-down information and bottom-up messages cross-reference each other to support more robust and accurate inference. Moreover, the scene layer may also systematically model the interaction among multiple targets so that the proposed framework can effectively deal with the inter-target occlusion while doing the inference. This can further boost the system performance.

3.2.2 Differences to Existing Hybrid Methods

In recent years, a few hybrid frameworks that combine data-driven messages and model-driven information have been proposed to improve the performance of image labeling and object detection. In [60], the authors integrated image contexts and local appearance into a hybrid framework to provide improved image labeling results. However, the detection problem has not been addressed in their method. In [61], a hierarchical conditional random field framework was proposed to model the interaction between image labeling and object detection. In this approach, the interaction is described based on scene-context relationship. However, the adopted segmentation process is mainly based on local features without taking into account the global shape layout constraints. In [62], a located-hidden-random-field framework has been proposed to label and detect objects simultaneously. This method mainly focuses on the detection of a single object and adopts an object labeling template that is treated as the global shape knowledge for object detection. Extra efforts are needed to identify the absence of objects or the presence of multiple objects. In [63], an extended work of located-hidden-random-fields framework, named layout-consistent random field framework, was proposed to further deal with inter-object occlusion. In

this method, inter-object occlusions are assumed to be unexpected and are handled by defining asymmetric pair-level potentials between adjacent labels.

Even though these aforementioned methods also integrate pixel-level, region-level, and object-level information for image content analysis, there are distinctive differences between our BHF-based modeling and theirs. In our approach, we couple the object-level information with the 3-D scene inference based on a unified parametric scene model. In the proposed BHF framework, since the camera parameters have been calibrated beforehand, we can fully utilize the geometric knowledge in the monitored scene. Unlike previous methods which learn the object-level information from a bunch of training data, our BHF framework adopts the 3-D parametric scene model to synthesize geometric patterns for model learning. In other words, we do not simply rely on training data for the learning of the object models. Moreover, in BHF modeling, the use of the parametric scene model has greatly reduced the dimension of the solution space. Since the possible status of each 3-D scene parameter is usually limited and can be quantized into a few choices, the possible solutions of image content labeling are well bounded.

Furthermore, since the 3-D scene is properly modeled, the occlusion effect, the perspective effect, and the shadow effect can be theoretically analyzed. To deal with the variations of the surrounding illumination and to integrate the geometric scene knowledge with image observation, a hidden labeling layer is included in the structure. With the hidden layer between the observation layer and the scene layer, our framework provides a systematic structure that is very suitable for solving luminance variations, shadow effect, perspective effect, and occlusion.

In BHF, image labeling is modeled as a pixel-level classification process. By dynamically training the pixel-level classification models to adapt to luminance variations, luminance-varying observations are converted into more consistent

labels. On the other hand, to handle the occlusion and shadow effect, the target number, target location, target size, and a few necessary scene factors are modeled as scene parameters. During the inference process, the statuses of those scene parameters are all inferred at the same time so that the occlusion effect and the shadow effect can be well handled.

Furthermore, for occlusions and shadows, the BHF framework can explicitly model their generation processes from 3-D scene to 2-D images. This makes occlusions and shadows a portion of the global knowledge. Hence, another distinctive feature of BHF is that occlusion and shadow effects may actually be used to offer useful and structured information to support scene inference. The occlusion effect tells how the 3-D objects in the scene interact with each other; while the shadow effect conveys the existing of certain objects. In BHF, these two effects are well modeled as parts of global knowledge. This kind of global knowledge may deduce expected labeling configuration when the scene parameters in the scene layer are specified. Under the BHF framework, scene modeling and image labeling processes are linked in an interactive manner. The labeling of image pixels adopts some global knowledge from the scene layer, while the scene layer makes a global inference based on local messages passed from the labeling process.

3.3 The Modeling of BHF

For different video surveillance systems, the system unknowns, the available physical constraints, and the available observations are application-dependent. In BHF, we treat the system observations and unknowns as random variables and represent them as nodes in the BHF structure. Through a learning procedure, we train appropriate probability models to model the physical constraints which are the links in the BHF structure. With the integration of system unknowns, system observations,

bottom-up constraints, and top-down constraints under the hierarchical framework, the analysis of image contents and the inference of the scene statuses are formulated as an optimization problem. By finding the optimal inference, the system can make a semantic understanding of the monitored scene.

In BHF, the inter-layer links and intra-layer links represent the message propagations that should be properly modeled. As illustrated in Fig. 14, observation nodes are assumed to be conditionally independent when the statuses of the labeling layer is given. This implies no connections among observation nodes. On the other hand, one labeling node represents a local decision based on a local observation. Hence, there is a link connecting each labeling node and its corresponding observation node. Moreover, the local decisions of two adjacent labeling nodes are usually highly correlated. This property is modeled by connecting the labeling nodes as a four-neighbor Markov random field (MRF) [18]. To model the interactions between the labeling layer and the scene layer, each scene node that represents one kind of 3-D scene status is connected to related labeling nodes. Through those connections, the global information of geometric arrangement may influence the classification of local labeling nodes. In BHF, the topology of the inter-layer connection is flexible and application-oriented. In Chapter 4 and Chapter 5, we will apply the BHF framework to two different applications, a parking space detection system and a multi-camera surveillance system, to demonstrate how to define the nodes and how to model the links of the BHF structure in real applications.

In principle, we can formulate the scene inference problem as a status decision process based on image observations. Since the process of image content analysis and the inference of the scene status are highly correlated, the proposed BHF is developed to combine the image labeling problem $p(H_L|I_L)$ and the scene inference problem $p(S_L|I_L)$ into a joint-inference problem $p(H_L, S_L|I_L)$. That is, our BHF always formulates

the system goal as to simultaneously find the optimal image content labeling and the 3-D scene parameters based on the image observation and some model constraints. By unifying these two problems under a single framework, the connections among pixel-level features, region-level constraints, and object-level knowledge are well-constituted in a hierarchical form. This structure enables the proper use of the information embedded among layers and provides an efficient way to deal with scene inference and image content analysis simultaneously rather than to solve them individually. To find out a suitable classification label H_L and the best scene inference S_L under the given observation I_L , an MAP optimization problem is defined as

$$\begin{aligned}
H_L^*, S_L^* &= \arg \max_{H_L, S_L} \ln p(H_L, S_L | I_L) \\
&= \arg \max_{H_L, S_L} \ln [p(I_L | H_L, S_L) p(H_L | S_L) p(S_L)] \\
&= \arg \max_{H_L, S_L} \ln [p(I_L | H_L) p(H_L | S_L) p(S_L)] \\
&= \arg \max_{H_L, S_L} [\ln p(I_L | H_L) + \ln p(H_L | S_L) + \ln p(S_L)]
\end{aligned} \tag{10}$$

where (H_L^*, S_L^*) denotes the optimal solution pair of image content labeling and 3-D scene parameters. Here, $p(S_L)$ represents the prior knowledge of the 3-D scene status and $p(H_L|S_L)$ stands for the object-level constraints propagated from the 3-D parametric scene model to the labeling layer. In the graphical structure of our BHF, we use the links between the scene layer and the labeling layer to represent $p(H_L|S_L)$. On the other hand, we assume $p(I_L|H_L, S_L) = p(I_L|H_L)$. That is, we assume the probabilistic property of the observed image data is conditionally independent of the scene model once if the pixel labels are determined. Moreover, $p(I_L|H_L)$ links the image observation data with the labeling results. In detail, $p(I_L|H_L)$ is composed of a pixel classification model for pixel-level information and an adjacency model for region-level information. As mentioned above, for the pixel classification model, we assume the observation nodes in Fig. 14 are conditionally independent when the status

of the labeling layer is given. In addition, we assume the connections between the observation layer and the labeling layer are one-to-one and these connections can be modeled in terms of a “classification energy” $E_D[I_L(m,n),H_L(m,n)]$. This classification energy conveys the property that the labeling result should be consistent with the feature values of the observed image. On the other hand, for the adjacency model, since the local labeling results of adjacent nodes are usually highly correlated, we define an “adjacency energy” $E_A[I_L(m,n),H_L(m,n);N_p]$ to depict the assumption that the labels of adjacent pixels should follow some kind of smoothness constraint. By combining these two energy models, we have

$$p(I_L | H_L) = K \cdot \prod_m \prod_n e^{-E_D[I_L(m,n),H_L(m,n)]} e^{-E_A[I_L(m,n),H_L(m,n);N_p]}. \quad (11)$$

Here, N_p denotes a neighborhood around the pixel location (m,n) and K is a normalization term.

In our system, $p(I_L|H_L)$ and $p(H_L|S_L)$ need to be explicitly determined in order to completely model the system goal as an optimization problem in (10). Once the models of BHF are defined, an optimal inference procedure is performed to obtain the results. In our BHF, the definition of the 3-D parametric scene model $p(H_L|S_L)$ and the pixel classification model $E_D[I_L(m,n),H_L(m,n)]$ are highly application-dependent. In order to explain the modeling of $p(H_L|S_L)$ and $E_D[I_L(m,n),H_L(m,n)]$, two examples will be demonstrated in Chapter 4 and Chapter 5, respectively.

On the other hand, the adjacency model $E_A[I_L(m,n),H_L(m,n);N_p]$ defined in the BHF framework is more generic. Usually, the local decisions of two adjacent labeling nodes are highly correlated especially when their corresponding image pixels share similar color features. In our system, by taking the observed image $I_L(m,n)$ into consideration, we define the adjacency energy of labeling nodes as a Markov random field [18] to provide a smoothness constraint between adjacent labeling nodes. Here,

we define

$$\begin{aligned}
& E_A[I_L(m,n), H_L(m,n); N_p] \\
& \equiv \beta \times \sum_{\Delta m=-p}^p \sum_{\Delta n=-p}^p C_A[I_L, H_L, m, n, \Delta m, \Delta n]
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
& C_A[I_L, H_L, m, n, \Delta m, \Delta n] \\
& \equiv (1 - \delta[H_L(m,n), H_L(m + \Delta m, n + \Delta n)]) \\
& \quad \times G_S(\|I_L(m,n) - I_L(m + \Delta m, n + \Delta n)\|)
\end{aligned} \tag{13}$$

and

$$\delta[p_a, q_a] = \begin{cases} 1 & \text{if } p_a = q_a \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

In (12), N_p denotes the $(2p+1) \times (2p+1)$ neighborhood around (m,n) and β is a pre-selected penalty constant. In (13), the function G_S is an adaptive function designed to preserve the intensity/color discontinuities in the original image. In our system, we design function G_S to be a function similar to a logistic sigmoid function:

$$G_S(U) = \text{Sigm}(U) + 1 = (1 - e^{\rho(U-C_{th})}) / (1 + e^{\rho(U-C_{th})}) + 1. \tag{15}$$

An example of $\text{Sigm}(U)$ is shown in Fig. 15. In principle, $\text{Sigm}(U)$ works like a soft thresholding function, with C_{th} and ρ controlling its zero-crossing point and shape, respectively. Both C_{th} and ρ are application-dependent and are determined empirically. $\text{Sigm}(U)$ outputs a positive value if U is smaller than C_{th} , and outputs a negative value otherwise. With this design, $C_A[\cdot]$ is equal to zero when $H_L(m,n)$ and $H_L(m+\Delta m, n+\Delta n)$ are the same. If $H_L(m,n)$ and $H_L(m+\Delta m, n+\Delta n)$ are different, $C_A[\cdot]$ gives a larger penalty if the difference between $I_L(m,n)$ and $I_L(m+\Delta m, n+\Delta n)$ is smaller than C_{th} , while gives a smaller penalty otherwise. Hence, to reduce the adjacency energy, $H_L(m,n)$ and $H_L(m+\Delta m, n+\Delta n)$ tend to share the same label when the difference between $I_L(m,n)$ and $I_L(m+\Delta m, n+\Delta n)$ is small, and tend to have different labels otherwise.

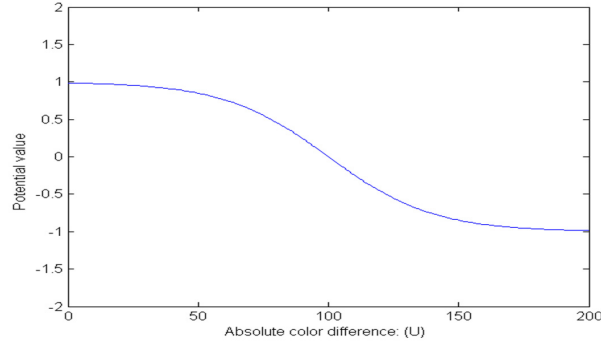


Fig. 15. Examples of $\text{Sigm}(U)$ with $\rho=0.05$ and $C_{th}=100$

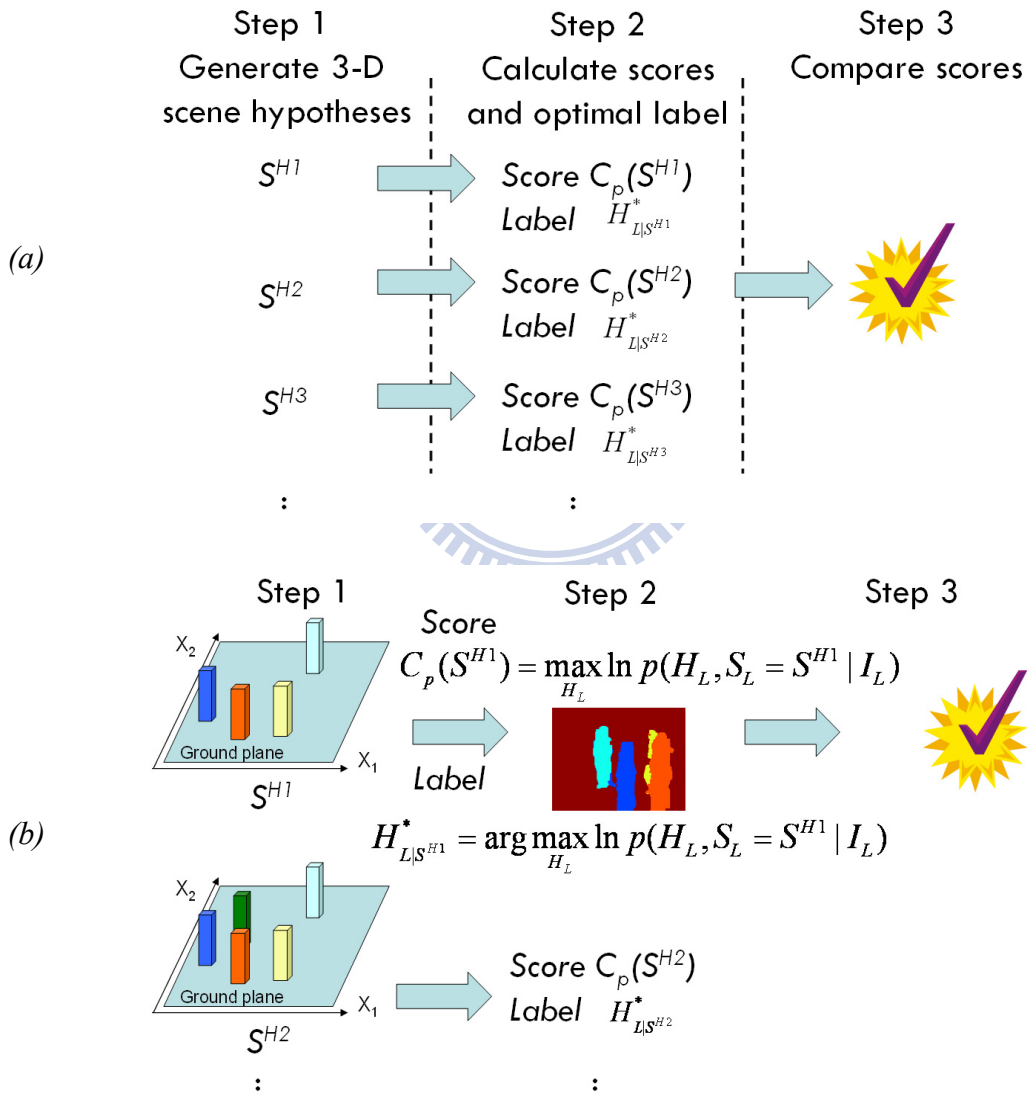


Fig. 16. Illustrate the inference process of BHF. (a) A standard inference process. (b) An example of BHF inference process for the multi-target multi-camera surveillance system.

3.4 The Inference of BHF

To solve (10), an inference procedure is needed for the determination of the optimal solution pair (H_L^*, S_L^*) . Since the undetermined variables include the optimal label of each pixel and the optimal status of scene parameters, this inference process is non-trivial at all. In our system, to find the status of each scene parameter, we first generate the possible status hypotheses of scene parameters. The status hypothesis that achieves the maximum posterior probability in (10) is picked. Here, we use Fig. 16 to illustrate the inference steps. In detail, to implement this idea, our inference process for BHF is composed of three major steps:

Step1: Generate the possible status hypotheses of scene parameters with the consideration of the independency among parameters. Assume there are P_a scene parameters in the system model and each parameter has S_N statuses. While generating the possible status hypotheses, there would be totally $S_N^{P_a}$ status hypotheses if we ignore the possible independency among parameters. By considering the independency, the number of eligible status hypotheses could be greatly reduced. To the best case, the number of eligible status hypotheses could be as small as the product of P_a and S_N . In real systems, the number of unknown scene parameters P_a are usually large but with some levels of independency. By properly taking into account these independency properties, the computational complexity of our inference process may grow much slower than the expected exponential growth.

Step 2: Given a possible status hypothesis S^H , find out the optimal labeling $H_{L|S^H}^*$ and compute the corresponding posterior probability of S^H , denoted as $C_p(S^H)$. In our approach, $H_{L|S^H}^*$ and $C_p(S^H)$ are defined as

$$H_{L|S^H}^* = \arg \max_{H_L} \ln p(H_L, S_L = S^H | I_L), \text{ and} \quad (16)$$

$$C_p(S^H) = \max_{H_L} \ln p(H_L, S_L = S^H | I_L) . \quad (17)$$

In (16) and (17), the energy function $\ln p(H_L, S_L = S^H | I_L)$ is defined as

$$\begin{aligned} & \ln p(H_L, S_L = S^H | I_L) \\ &= - \sum_m \sum_n E_D[I_L(m, n), H_L(m, n)] \\ & \quad - \sum_m \sum_n E_A[I_L(m, n), H_L(m, n); N_p] \\ & \quad + \sum_m \sum_n \ln p(H_L(m, n) | S = S^H) + \ln p(S = S^H) \end{aligned} \quad (18)$$

In our system, the maximization of $\ln p(H_L, S_L = S^H | I_L)$ under the given status hypothesis S^H has a form much like the canonical MRF optimization formulation frequently used in some early-vision problems [18][19][20]. For a canonical MRF optimization formulation, the energy function E^{MRF} , usually viewed as the log likelihood of the posterior distribution of an MRF [20][21][22], is composed of two parts, part one E_D^{MRF} and part two E_S^{MRF} , with a constant λ controlling the weighting between the part one and the part two. That is,

$$E^{MRF} = E_D^{MRF} + \lambda \times E_S^{MRF} . \quad (19)$$

To fit (16) and (17) into the canonical MRF optimization formulation, we combine $E_D(I_L(m, n), H_L(m, n))$, $p(H_L(m, n) | S = S^H)$, and the prior $p(S = S^H)$ in (18) to build the part one in (19); and treat $E_A[I_L(m, n), H_L(m, n); N_p]$ as the part two. With this formulation, (16) and (17) can be solved by many practical optimization algorithms, such as the graph cuts algorithm, the loopy belief propagation algorithm, the tree-reweighted algorithm, and the iterated conditional mode algorithm. Based on a recent study of those methods [20], the graph cuts algorithm [64][65][66] has been found to perform better in terms of runtime.

Hence, in our system, we apply the graph cuts algorithm to the maximization of (16) and (17) under the status hypothesis S^H . Here, the optimal image labeling under S^H are achieved by assigning a suitable label to each pixel. To explain how we apply the graph cuts algorithm to our system, we assume each pixel has a label from the terminal (label) set $\{T_0, T_1, \dots, T_M\}$. To setup the graph cuts method, we form a graph as shown below in Fig. 17 to represent our optimization problem. In this graph, a possible terminal connects to a portion of labeling nodes in the labeling image. Their relations are represented by the collections named as “t-links”. In our system, we use data term to define the weight of each t-link. On the other hand, the “n-links” in the labeling image is defined by the smoothness term. With this graph representation, our optimization problem is equal to cutting the t-links and n-links with the minimal cost so that all terminals are separated and each labeling node $H_L(m,n)$ only connects to one terminal through a t-link.

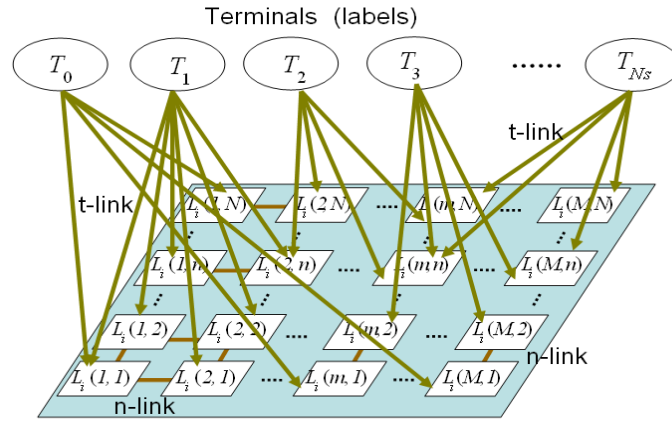


Fig. 17. The graph setting for the graph cuts algorithm.

Step 3: Compare the values of posterior probability over all possible status hypotheses.

The status hypothesis that achieves the maximal value of posterior probability is picked as the optimal status S_L^* . The corresponding optimal image labeling under S_L^* defines the optimal labeling H_L^* .

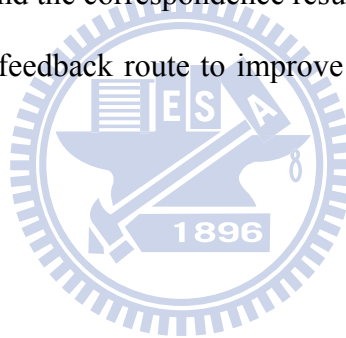
3.5 The Application of BHF

To further explain how to apply the framework to practical video surveillance applications under different scene conditions, we will discuss two real systems as examples in the following two chapters. In Chapter 4, we firstly apply the BHF to the design of a vacant parking space detection system over an outdoor parking lot, which is a scene with well-structured and predictable 3-D model. Next in Chapter 5, we apply the BHF to the tracking of multiple targets over a multi-camera system, whose scene model is dynamically changing and unpredictable. Below, we briefly explain the roles of the proposed BHF in these two systems.

In the first application, we apply the BHF framework to a system for vacant parking space detection. Based on the 3-layer BHF, the bottom-up messages from image observation and the top-down knowledge from the scene model are effectively integrated. In BHF, the illumination variations in the outdoor scene are overcome by transferring the fluctuating RGB observations into meaningful labels. To adapt to the time-varying lighting condition, we online build the color classification models for object type and lighting condition. On the other hand, some global knowledge of the 3-D scene, like the direction of sunlight and the 3-D car model, offers useful information for the labeling of image pixels. The top-down knowledge is propagated downward to influence the labeling process via the generation of an “expected object map” and an “expected shadow map”. By compromising between the expected labeling maps and the labeling from image observation, the status hypotheses of each parking space are evaluated. Under the proposed BHF, the vacant parking space detection problem and the optimal image content labeling problem are integrated in a unified manner.

On the other hand, in the application of multi-target tracking with ghost

suppression over a multi-camera system, we propose a new approach to efficiently integrate, summarize, and infer video messages from multiple client cameras. The main concept is to fuse detection results from many client cameras, summarize consistent 2-D messages into a 3-D space, and do the inference for the scene model so that the operators in the control room can monitor the surveillance zone in an easier and more intuitive way. Here, we proposed a fusion-inference procedure to preserve the accuracy of target location without dramatically increasing the computational cost. In our fusion-inference procedure, the data fusion stage is used to detect possible targets and their 3-D locations. Based on the 3-D priors, target identification, labeling, and inter-occlusion are then analyzed under the proposed BHF in the inference stage. The optimal target labeling and the correspondence result are further used to refine the 3-D target model through a feedback route to improve the accuracy of the inference stage.



CHAPTER 4

A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection



4.1 Introduction of Parking Space Detection

In this chapter, we introduce how the proposed BHF is adopted to detect the vacant parking spaces in a typical outdoor parking lot. Nowadays, using an intelligent surveillance system to manage parking lots has become practical. A recent technology review about smart parking system can be found in [67]. To assist users to efficiently find a vacant parking space, an intelligent parking space management system can not only provide the total number of vacant spaces in the parking lot but also explicitly identify the location of vacant parking spaces. In addition, a vision-based system may provide many value-added services, like parking space guidance and video surveillance.

In practice, the major challenges of vision-based parking space detection come

from occlusion effect, shadow effect, perspective distortion, and the fluctuation of lighting condition. In Fig. 18, we show several parking lot images in our dataset. In these images, some environmental factors are mixed together in a sophisticated way. For instance, the illumination in a sunny day is quite different from that in a cloudy day; a parked car may occlude or cast a shadow over the parking space next to it; a shadowed region may be mistakenly recognized as a dark-colored vehicle; and a light-colored vehicle under strong sunlight may look very similar to a vacant parking space.

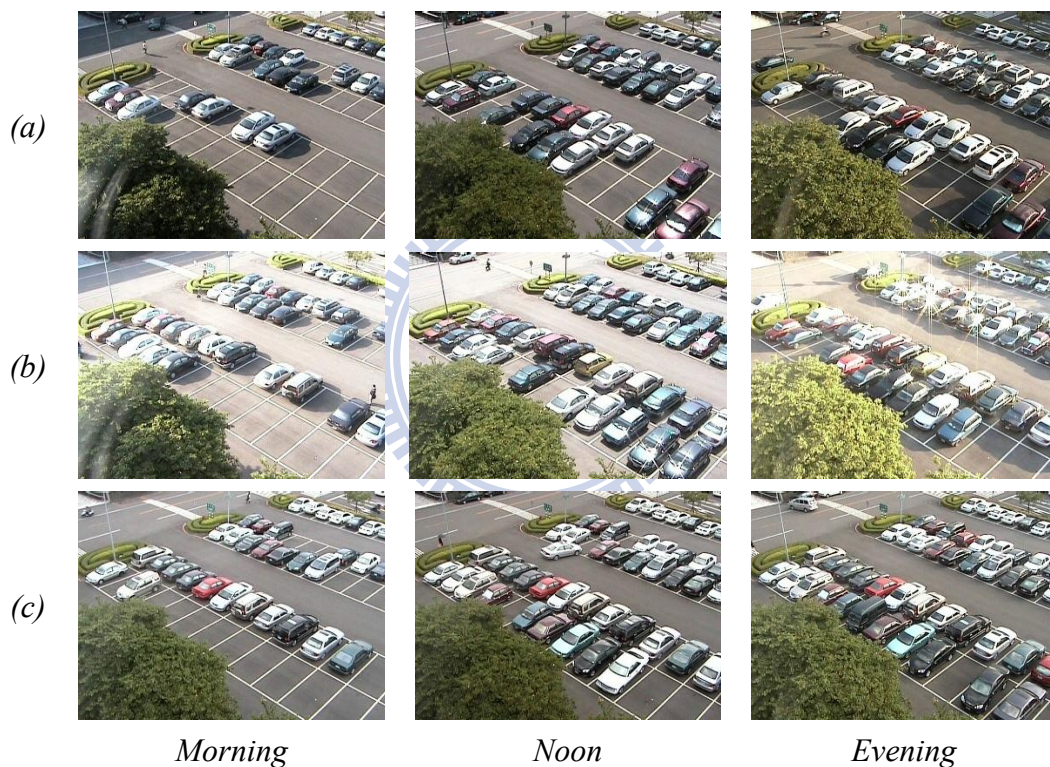


Fig. 18. Image shots of a parking lot. (a) Captured in a normal day. (b) Captured in a day with strong sunlight. (c) Captured in a cloudy day.

Up to now, several methods have been proposed to overcome the aforementioned difficulties. These methods can be roughly classified into two major categories: car-driven and space-driven. For a car-driven method, cars are the major target and algorithms are developed to detect cars. Based on the result of car detection, vacant parking spaces are determined. To detect objects of interest, plentiful object detection

algorithms can be used. For example, the object detection method proposed in [68] by Schneiderman and Kanade is a trainable detector based on the statistics of localized parts. The adaboosting-based detection algorithm [69] is another widely used technique for the detection of specific objects in 2-D images. The method proposed by Felzenszwalb et al. [32] offered an efficient way to match objects based on a part-based model that well represents an object by pictorial structures. A global color-based model had been proposed by Tsai et al. [13] to efficiently detect vehicle candidates. On the other hand, Lee et al. [70] and Masaki [71] kept tracking and recording the movement of vehicles to identify empty parking spaces. Even though these object detection based frameworks had gained impressive achievement in many circumstances, such as highway and roadway, most of these algorithms are not specifically designed for vacant parking space detection in a typical parking lot. For example, as shown in Fig. 18, the captured images may include some cars with unclear details. Besides, due to the perspective distortion, a car far away from the camera only occupies a small area in the captured image. This perspective distortion may also affect the performance of car detection.

For a space-driven method, the property of a vacant parking space is the major focus and available parking spaces are detected directly. When the camera is static, several background subtraction algorithms, like [2], can be used to detect foreground objects. Typically, these algorithms assume the variation of the background is statistically stationary within a short period. Unfortunately, this assumption is not always true for an outdoor scene. For example, a passing cloud that block the sunlight may suddenly change the lightness. To handle the dynamic variation of an outdoor environment, a possible solution is to build a complete background reference set under all kinds of lighting conditions. Funck et al. [11] proposed an eigen-space representation that models a huge set of background models with much less memory

space and computational cost. With a suitable background model, a typical way to determine the status of a parking space is to check the ratio of foreground pixel number to background pixel number. However, even if the background model is well learned, this kind of method still suffers from the occlusions and shadows caused by neighboring cars. To improve the performance of detection, Huang et al. [46] proposed a Bayesian detection framework to take into account both ground plane model and car model. Both occlusion effect and illumination variation were modeled under that framework. Recently, Bong et al. [72] proposed a Car Park Occupancy Information System (COINS) by using a “bi-stream” detector to overcome the shadow effect. In their approach, one stream used the background subtraction method to perform car detection, while the other stream adopted edge information to achieve shadow-insensitive detection. By using an “And” operator to combine both detection results, detection performance was improved.

On the other hand, some other space-driven methods assume a vacant parking space possesses homogeneous appearance and use this property to detect vacant spaces. For example, Yamada and Mizuno [73] designed a homogeneity measure by calculating the area of fragmental segments. In principle, a vacant space has fewer but larger segments, while the area of a parked car has an opposite property. Lee et al. [74] suggested an entropy-based metric to determine the status of each parking space. However, these two systems ignored the shadow and occlusion caused by adjacent cars. In [75], Fabian used a segment-based homogeneity measure similar to that in [73] and proposed a method for occlusion handling. By pre-training a weighting map to indicate the image regions that may get occupied by neighboring cars, the influence of the occlusion effect can be reduced. Even though their homogeneity measure is effective for most parking spaces, the environmental variations, especially the shadow effect and the over-exposure effect caused by strong sunlight, may fail the assumption

of homogeneous appearance. In practice, the shadow effect makes a parking space less homogeneous while the over-exposure effect makes the appearance of a car more homogeneous.

Some other authors tried to detect vacant parking spaces via classification. For example, Dan [76] trained a general support vector machine (SVM) classifier by directly using the cascaded color vectors inside a parking space as the classification feature. However, the occlusion patterns were not well modeled in their approach. On the other hand, Wu et al. [77] grouped three neighboring spaces as a unit and define the color histogram across three spaces as the feature in their SVM classifier. With this arrangement, the inter-space correlation can be learned beforehand to overcome the inter-occlusion problem. However, the performance of classification is greatly affected by the environmental variations. In general, the lighting changes may cause the variations of object appearance in both brightness and chromaticity. This effect may dramatically degrade the accuracy of classification-based detection.

The rest of this chapter is organized as follows. In Section 4.2, we present the main idea of our algorithm. The top-down information from the 3-D scene model is detailed in Section 4.3, while the message from image observation is presented in Section 4.4. The whole inference procedure is explained in Section 4.5. Experimental results and discussions are presented in Section 4.6.

4.2 Overview of Vacant Space Detection

In our system, the scene modeling and vacant parking space detection are accomplished based on the integration of scene prior and image observation in the BHF. By treating the status of each parking space as a part of the scene parameters, the vacant space detection is achieved via the process of scene inference. The general concept of the proposed system is illustrated in Fig. 19. Based on the BHF, the

bottom-up messages from image observation and the top-down knowledge from the scene model are integrated. In BHF, the illumination variations are overcome by transferring the fluctuating RGB observations into meaningful labels. The labeling process is treated as a color classification process between content labeling and image observation. Since the observation difference is mainly caused by the object type and the lighting condition, we decompose the image observation into an object component and a lighting component. The object type is either “car” or “ground”, while the lighting condition is either “shadowed” or “unshadowed”. To adapt to the time-varying lighting condition, we online build the color classification models for object type and lighting condition. On the other hand, some global knowledge of the 3-D scene offers useful information for the labeling of image pixels. The top-down knowledge is propagated downward to influence the labeling process via the generation of an “expected object map” and an “expected shadow map”. Here, we explicitly define a generative model that takes into account the inter-occlusion effect, the expected shadow effect, and the perspective distortion. The relationships among these effects and the status of parking spaces are explicitly modeled via a Bayesian probabilistic model. By compromising between the expected labeling maps and the labeling from image observation, the status hypotheses of each parking space are evaluated. Finally, to avoid incorrect inference caused by unexpected occlusions, the global status hypotheses from the scene model provides useful constraints to handle partially inconsistent labels. In principle, we can formulate the vacant space detection problem as a status decision process based on image observations from a single camera. Since the status of a parking space may actually affect the inference of neighboring spaces, we analyze the status of neighboring parking spaces at the same time. Moreover, the vacant parking detection process is regarded as a Bayesian inference problem and is solved by finding the most reasonable parking space status

that fits both scene prior and image observation.



Fig. 19. The concept of Bayesian hierarchical framework for vacant space detection.

In Fig. 20, we show a simplified 3-layer structure to explain the BHF framework for vacant space detection. Here, we define the image observation layer as I_L , where each node $I_L(m,n)$ indicates the RGB color feature at the (m,n) pixel of an image of size $M \times N$. On the other hand, we define the labeling layer as H_L , where each node $H_L(m,n)$ represents the categorization of the image pixel at (m,n) . The labeling result of $H_L(m,n)$ could be (C,S) , (G,S) , (C,US) , or (G,US) , where C denotes “Car”, G denotes “Ground”, S denotes “Shadowed”, and US denotes “Unshadowed”. Moreover, we define the scene layer as S_L , which indicates the status hypotheses of the parking

spaces. The node $S_L(i)$ in S_L denotes the status of the i th parking space. Its value can be either 1 (occupied) or 0 (vacant).

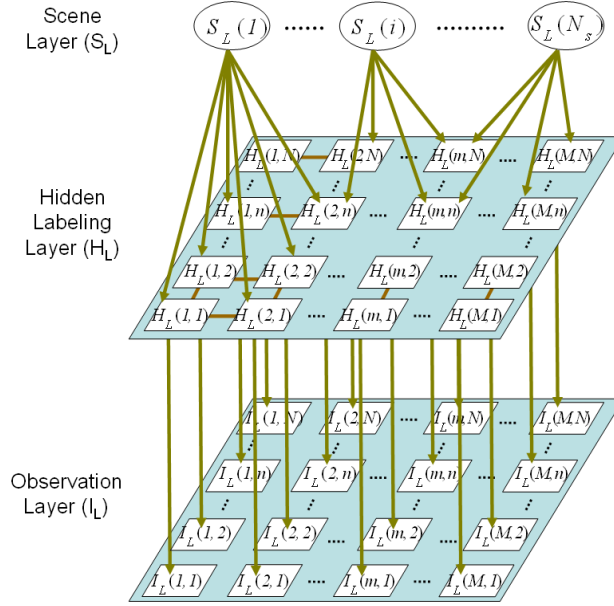


Fig. 20. Illustration of the 3-layer BHF for vacant space detection.

In BHF, the topology of the inter-layer connections represents the probabilistic constraints between nodes as illustrated in Section 3.3. Given the observation I_L , the status of the parking spaces is determined by finding the optimal pair (H_L^*, S_L^*) such that

$$\begin{aligned} H_L^*, S_L^* &= \arg \max_{H_L, S_L} \ln p(H_L, S_L | I_L) \\ &= \arg \max_{H_L, S_L} [\ln p(I_L | H_L) + \ln P(H_L | S_L) + \ln p(S_L)] \end{aligned} \quad (20)$$

The detail deduction of (20) is the same as that of (10). In the parking space detection system, $p(I_L|H_L)$ constrains that the labeling results should be consistent with the RGB values of the observed image. Moreover, the labels of adjacent pixels should follow some kind of smoothness constraint. On the other hand, $p(H_L|S_L)$ constrains that the labeling of parked cars and shadowed regions should match the expected inter-occlusion pattern and shadow pattern in a probabilistic sense. Finally, $p(S_L)$

represents the prior knowledge of the parking space status. In our system, we assume the “occupied” status and the “available” status are equally possible for every parking space. With this assumption, the $\ln p(S_L)$ term in can be ignored. Moreover, to find the optimal solution in (20), we adopt the graph-cuts technique as mentioned in Section 3.4.

4.3 Top-Down Knowledge From Scene Layer

Since the parking spaces in a parking lot are well structured, we can synthesize an expected object map once if we have the 3-D car model and have a hypothesis about the status of parking spaces. On the other hand, if we know the lighting condition (sunny or cloudy) and have the direction of sunlight, we may also synthesize an expected shadow pattern. In our system, both expected object map and expected shadow map are created to help the labeling of image pixels. In our approach, $p(H_L|S_L)$ is reformulated as

$$p(H_L | S_L) = \prod_m \prod_n p(H_L(m, n) | S_L), \quad (21)$$

in which we assume the labeling nodes $H_L(m, n)$ are conditionally independent of each other once if the knowledge from the scene layer S_L is given. Since the object type and the lighting type are physically independent, we formulate $p(H_L(m, n)|S_L)$ as

$$p(H_L(m, n) | S_L) = p(h^o(m, n) | S_L) p(h^l(m, n) | S_L). \quad (22)$$

In physics, the object labeling model $p(h^o(m, n)|S_L)$ includes the expected car mask and the inter-occlusion effect among neighboring cars; while the light labeling model $p(h^l(m, n)|S_L)$ includes the expected shadow mask to indicate shadowed pixels. To define these two labeling models, we first introduce a parametric model to define the 3-D structure of a parking lot. Based on the parametric scene model, we propose a generation process to generate the expected object labeling map and the expected shadow labeling map.

4.3.1 3-D Scene Parameters

In our system, the number of parking space (N_s) and their locations on the 3-D ground plane are defined and learned in advance. In a normal situation, a car is parked inside a parking space. To simulate a parked car, we assume each car is a cube in the 3-D world. The length (l), width (w), and height (h) of the cube are modeled as three independent Gaussian random variables, with the probability density functions $p(l)$, $p(w)$, and $p(h)$. Besides, the random vector $(l, w, h)^T$ is assumed to be identically and independently distributed at different parking spaces. Here, the probability density functions $p(l)$, $p(w)$, and $p(h)$ are pre-learned based on 120 parked cars. On the other hand, the 3-D ground plane of the parking lot is defined as a 2-D plane $(X, Y, 0)$. Inside the i th parking space, we assume the projection of the car center on the ground plane is represented by $(X_i, Y_i, 0)$, where X_i and Y_i are modeled as two randomly distributed Gaussian random variables with the probability density functions $p(X_i)$ and $p(Y_i)$. The mean values of $p(X_i)$ and $p(Y_i)$ are set to be the center of the i th parking space on the ground plane. Moreover, we assume the location pattern of parked cars at difference parking spaces is similar. That is, we assume the variances of $p(X_i)$ and $p(Y_i)$ are independent of i . To train the variance values of $p(X_i)$ and $p(Y_i)$, we measured for each of these 120 cars the deviation of the car center from the center of the parked space.

To predict the shadowed regions, we model the lighting condition in the 3-D scene. In general, we may assume there are two major types of illumination in an outdoor environment: direct illumination from the Sun and ambient illumination from the sky. For each image pixel, it may be lighted by the skylight only, or lighted by both skylight and sunlight. Basically, shadow reflects the contrast of brightness for regions illuminated by different types of lighting. If the sunlight exists in the environment, the regions lighted by skylight only appear to be shadowed. On the

other hand, when sunlight is absent, we assume there is no shadowed region. Moreover, when sunlight is present, we assume the direction of sunlight is represented by a three dimensional vector $(D_X(t), D_Y(t), D_Z(t))^T$, which is a function of time t . In our approach, the 3-D scene model of a parking lot is determined by the parameter set Φ , where

$$\Phi = \{D_X(t), D_Y(t), D_Z(t), \{S_L(i), l_i, w_i, h_i, X_i, Y_i, \text{ for } i = 1, 2, \dots, N_s\}\}. \quad (23)$$

In Φ , $\{S_L(i)\}$ is the main unknown variable in scene model. The detailed deduction of the sunlight direction $(D_X(t), D_Y(t), D_Z(t))^T$ is to be explained later.

4.3.2 Generation of Expected Labeling Maps

4.3.2.1 Object Labeling Model

In our system, once the 3-D scene parameters Φ are given, the expected object labeling and the expected shadow labeling on the captured images are automatically generated. Based on the projection matrix of the camera, a synthesized car parked at $(X_i, Y_i, 0)$ inside the i th parking space, with length l_i , width w_i , and height h_i , is projected onto the camera view to get the projection image $M_i(m, n | X_i, Y_i, l_i, w_i, h_i)$, which has the value 1 if the pixel (m, n) is within the projected region, and 0 otherwise. Since the size parameters (l_i, w_i, h_i) and the parked location (X_i, Y_i) may vary from car to car, we take into account the prior probabilities $p(l_i)$, $p(w_i)$, $p(h_i)$, $p(X_i)$, and $p(Y_i)$ and define the expected car labeling map to be a probabilistic map $C_i(m, n)$, which is the expectation value of $M_i(m, n | X_i, Y_i, l_i, w_i, h_i)$. That is,

$$C_i(m, n) = E_{X_i, Y_i, l_i, w_i, h_i} [M_i(m, n | X_i, Y_i, l_i, w_i, h_i)]. \quad (24)$$

On the other hand, since the object type of an image pixel is either ‘‘Car’’ or ‘‘Ground’’, the expected ground labeling map is defined as

$$G_i(m, n) = 1 - E_{X_i, Y_i, l_i, w_i, h_i} [M_i(m, n | X_i, Y_i, l_i, w_i, h_i)]. \quad (25)$$

In our system, we numerically calculate the expectation in (24) and (25) based on the Monte Carlo approach. Here, based on the prior probabilities $p(l_i)$, $p(w_i)$, $p(h_i)$, $p(X_i)$, and $p(Y_i)$, we draw a large set of sample tuples. For each sample tuple, say $(l_k, w_k, h_k, X_k, Y_k)$, we synthesize a projection image. By averaging all projection images for all sample tuples, we get a probability map that approximates $C_i(m, n)$. In Fig. 21(b), we show the expected car labeling map of the car in Fig. 21(a).

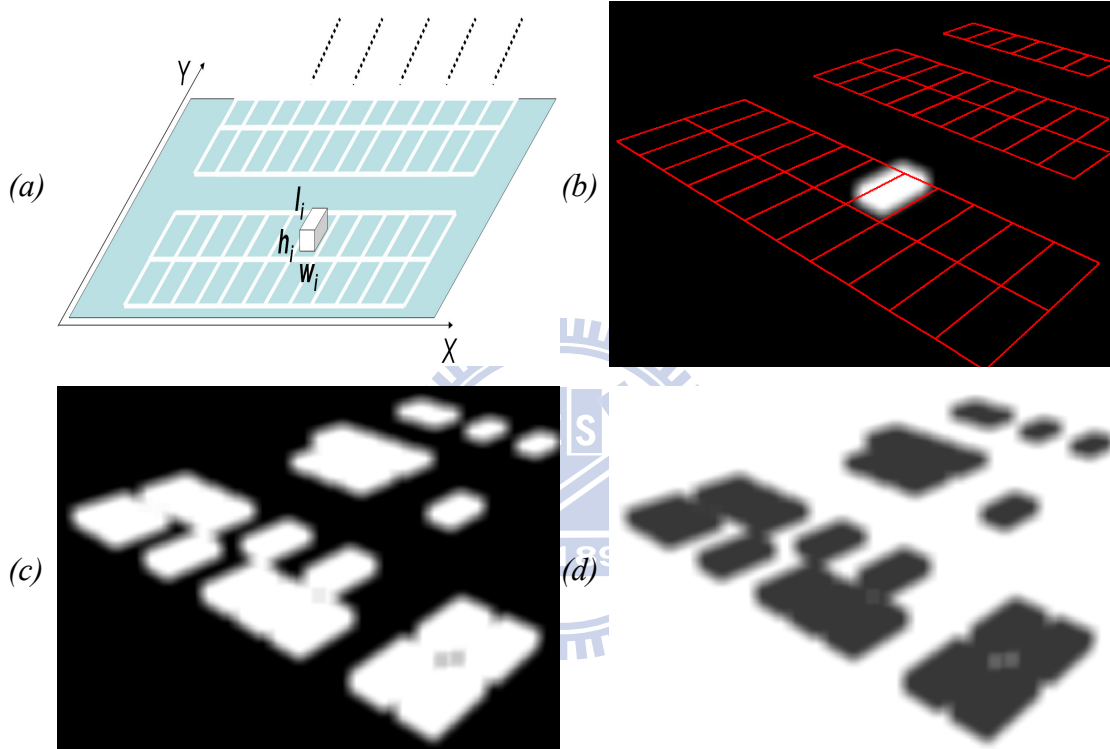


Fig. 21. (a) A 3-D car model. (b) Expected car labeling map of a parked car. (c) Expected car labeling of all parked cars. (d) Expected ground labeling of all parked cars.

While taking all parking spaces into consideration, an image pixel at (m, n) in the i th parking space may get occluded not only by a car parked at that parking space but also by a car parked at an adjacent parking space. To model the inter-occlusion effect in the object labeling model, we define the probability

$$p(h^o(m, n) = 0 | S_L) = \prod_{i=1}^{N_s} [G_i(m, n)^{S_L(i)}] \quad (26)$$

where $S_L(i)$ is the status of the i th parking space. With (26), the probability of car

labeling at (m,n) given the status of all parking spaces can be formulated as

$$p(h^o(m,n) = 1 | S_L) = 1 - \prod_{i=1}^{N_s} [G_i(m,n)^{S_L(i)}] \quad (27)$$

In Fig. 21(c) and (d), we show the examples of $p(h^o(m,n) = 1 | S_L)$ and $p(h^o(m,n) = 0 | S_L)$, respectively.

4.3.2.2 Shadow Labeling Model

Similarly, by using a cube model for a parked car, the expected shadowed regions on the ground plane can be quickly determined in the 3-D space whenever the sunlight direction is known and the status of parking spaces are determined. An example is illustrated in Fig. 22. Here, we define $T_i(m,n | X_i, Y_i, l_i, w_i, h_i)$ to be the projected shadow labeling image generated by a car parked at $(X_i, Y_i, 0)$ inside the i th parking space, with length l_i , width w_i , and height h_i . Similarly, by taking into account the prior probabilities $p(l_i)$, $p(w_i)$, $p(h_i)$, $p(X_i)$, and $p(Y_i)$, we define the expected shadow labeling map $S_i(m,n)$ in a probabilistic sense:

$$S_i(m,n) = E_{X_i, Y_i, l_i, w_i, h_i} [T_i(m,n | X_i, Y_i, l_i, w_i, h_i)]. \quad (28)$$

Similarly, the expected non-shadow labeling map is defined as $US_i(m,n) = 1 - S_i(m,n)$.

In Fig. 22(b), we show the expected shadow labeling map of the car in Fig. 22(a).

To model the shadow labeling model $p(h^L(m,n) | S_L)$ with the consideration of all parking spaces, we define

$$p(h^L(m,n) = 0 | S_L) = \prod_{i=1}^{N_s} [US_i(m,n)^{S_L(i)}]. \quad (29)$$

With (29), the probability of shadow labeling at (m,n) given S_L is modeled by

$$p(h^L(m,n) = 1 | S_L) = 1 - \prod_{i=1}^{N_s} [US_i(m,n)^{S_L(i)}]. \quad (30)$$

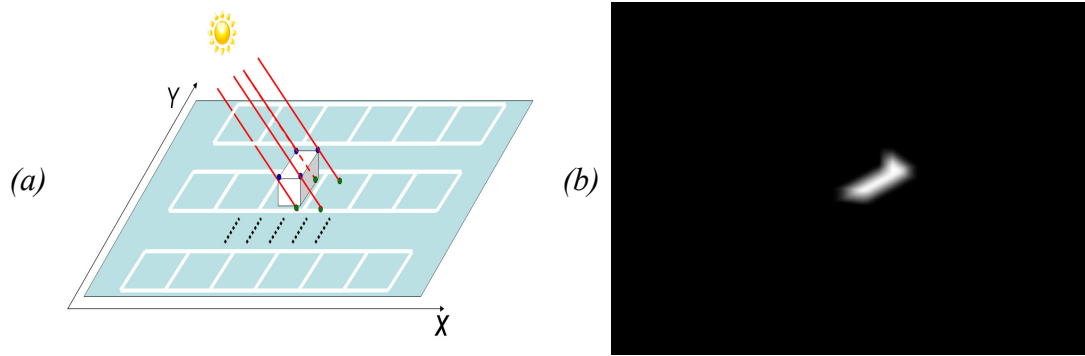


Fig. 22. (a) Shadow formation. (b) Expected shadow labeling map.

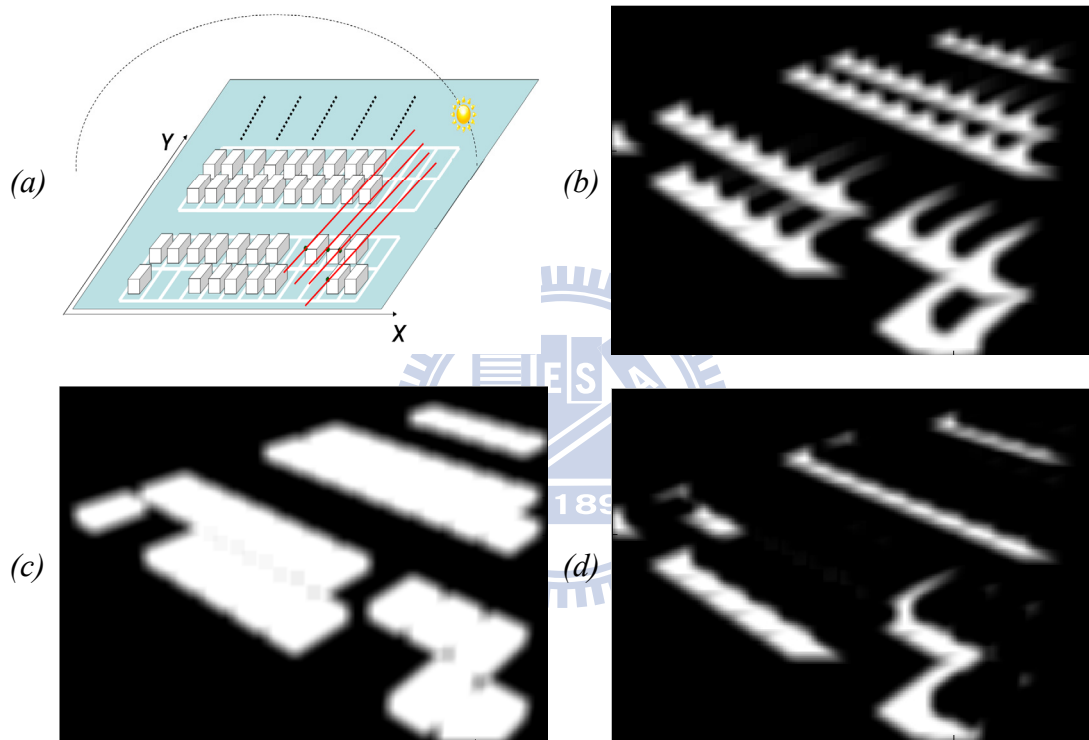


Fig. 23. (a) A 3-D car model. (b) Expected car labeling map of a parked car. (c) Expected car labeling of all parked cars. (d) Expected ground labeling of all parked cars.

In Fig. 23(a) and (b), we show an example of the 3-D parking lot model and its expected shadow labeling map. To simplify the problem, we ignore the shadows cast upon the parked cars and only consider the shadows cast on the ground plane. With this assumption, a pixel with a higher probability of car labeling is less likely to be shadowed. Hence, we refine the probabilistic shadow labeling map to be

$$p(h^L(m,n)=1|S_L) = (1 - p(h^O(m,n)=1|S_L)) \times (1 - \prod_{i=1}^{N_s} [US_i(m,n)^{S_L(i)}]). \quad (31)$$

A refined shadow labeling map is shown in Fig. 23(d).

4.3.3 Estimation of Sunlight Direction

To generate the expected shadow labeling map, we need the direction of sunlight. The information of sunlight parameters is available on the internet, like the U.S. Naval Observatory website [78]. By providing the date and the geo-location of the parking lot, including longitude and latitude from a global position system (GPS), the web service can provide samples of sunlight direction for every 10 minutes.

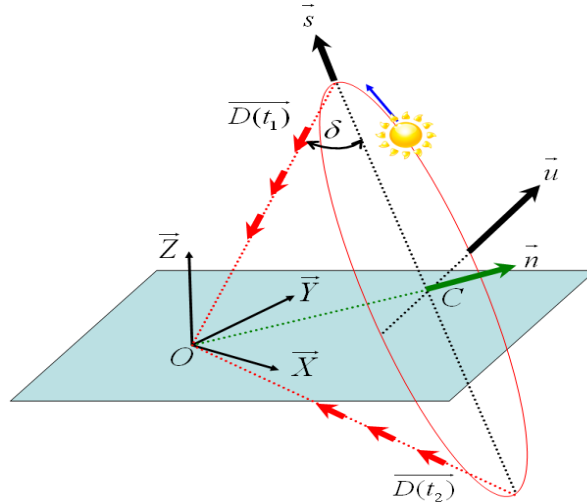


Fig. 24. Illustration of solar movement and sunlight direction.

In our system, we adopt the concept proposed in [79] to calculate the sunlight direction. In principle, the solar motion model and the sunlight direction can be estimated based on the variations of intensity values in a day. In a single day, the solar motion follows a circle on the solar plane in the 3-D space, with a constant angular frequency ω_s , as illustrated in Fig. 24. The angular frequency depends mainly on the self rotation of the Earth and is known in advance. The whole set of sunlight directions in a day form a conical surface and the cone aperture is equal to $\pi-2\delta$,

where δ is the Sun declination angle approximated as

$$\delta = -23.45^\circ \cdot \cos\left[\left(\frac{360}{365}\right) \cdot (N_d + 10)^\circ\right] \quad (32)$$

In (32), N_d is the number of days counted from January 1 to the current date. With this cone model, the sunlight direction over time can be parametrically represented by

$$\overline{D}(t) = -\{\sin(\delta)\vec{n} + \cos(\delta)[\cos(\omega_s(t - t_\theta))\vec{u} + \sin(\omega_s(t - t_\theta))\vec{s}]\}, \quad (33)$$

where \vec{u} is a unit reference vector on the solar plane at time t_θ , \vec{n} is the normal vector of the solar plane, and $\vec{s} = \vec{n} \times \vec{u}$.

On the other hand, we assume the scene surfaces are mainly Lambertian surfaces. Hence, the intensity value reflected from a surface is proportional to the incident angle of the incident light with respect to the surface normal. The intensity value at an image pixel will climb to its maximum when the subtended angle between the corresponding surface normal vector and the sunlight direction reaches the minimum. As explained in appendix section A, if \vec{P} is the normal vector of a surface patch in the 3-D scene, the intensity value at the image pixel can be approximated as

$$I_{sun}(m, n, t) = B(m, n) \cos(\omega_s t - \theta_p(m, n)) + C(m, n), \quad (34)$$

which is a scaled cosine function plus a constant offset. Moreover, if θ represents the angle subtended by \vec{u} and the projection of \vec{P} on the solar plane, the phase shift θ_p of the cosine function is equal to θ up to a constant offset. In principle, if we pick up three image pixels, whose 3-D scene points lie on different surfaces with linearly independent normal vectors, we can deduce the geometric relationship between the solar plane and these three surface normal vectors [79]. For detailed deduction, please refer to Appendix A.

In Fig. 25(a), we show three manually selected image pixels in the parking lot scene, one from the driveway and two from the bushes. These image pixels locate at three mutually orthogonal planes. The intensity profile of a pixel in green region is

shown in Fig. 25(b) as an example. By identifying the phase shift θ_p from each of these three intensity profiles, we can determine the sunlight direction $\overline{D(t)}$ at any time instant t . Moreover, if a parking lot cannot provide these three mutually independent planes, an artificial cube is recommended to be set up in the parking lot scene.

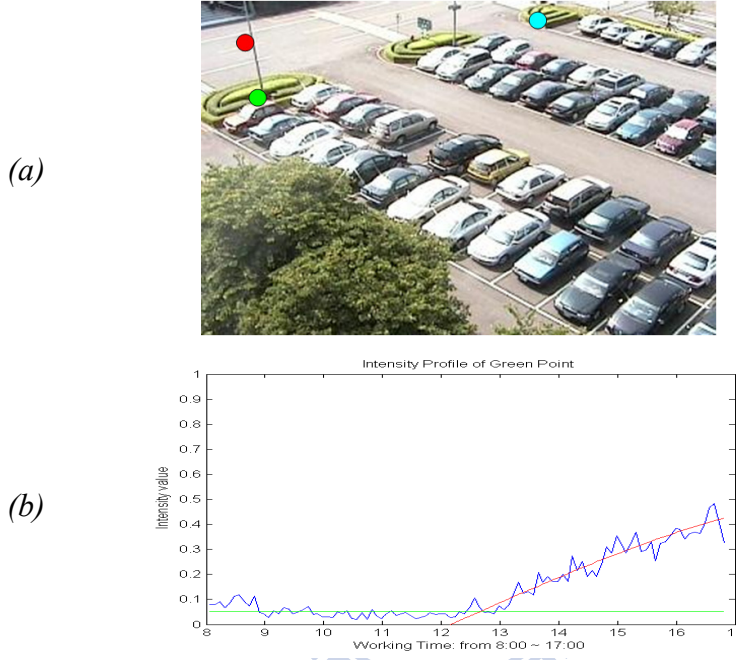


Fig. 25. (a) A parking lot image with three manually selected image pixels, marked in red, green, and blue. (b) The intensity profiles (blue) of the green pixel, overlapped with the fitted skylight profile (green) and the fitted skylight+sunlight profile (red).

4.4 Bottom-Up Messages From Observation Layer

In our parking space detection system, the bottom-up messages are embedded in the likelihood function $p(I_L|H_L)$, which links the observation data with the labeling results. As mentioned in Section 3.3, $p(I_L|H_L)$ is composed of a “classification energy” $E_D[I_L(m,n),H_L(m,n)]$ and an “adjacency energy” $E_A[I_L(m,n),H_L(m,n);N_p]$. That is, we have

$$p(I_L | H_L) = K \cdot \prod_m \prod_n e^{-E_D[I_L(m,n),H_L(m,n)]} e^{-E_A[I_L(m,n),H_L(m,n);N_p]} \quad (35)$$

In (35), N_p denotes a neighborhood around (m,n) and K is a normalization term. In the following subsections, we will explain the design of these energy models.

4.4.1 Classification Energy Model

4.4.1.1 Energy Model

In our approach, we convert the RGB color features \mathbf{I}_{RGB} of each pixel into a semantic labeling. Here, we model the classification energy as

$$E_D[I_L(m,n), H_L(m,n)] = -\ln(p(\mathbf{I}_{\text{RGB}}(m,n) | h^O(m,n), h^L(m,n))), \quad (36)$$

where $p(\mathbf{I}_{\text{RGB}} | h^O, h^L)$ is the conditional probability distribution of \mathbf{I}_{RGB} given the semantic labeling (h^O, h^L) . In (36), $h^O(m,n)$ could be C or G , and $h^L(m,n)$ could be S or US . For more detail, in Fig. 26, we show an example of color distributions in the RGB color space under the four different labeling statuses --- (C,S) , (C,US) , (G,S) , and (G,US) .

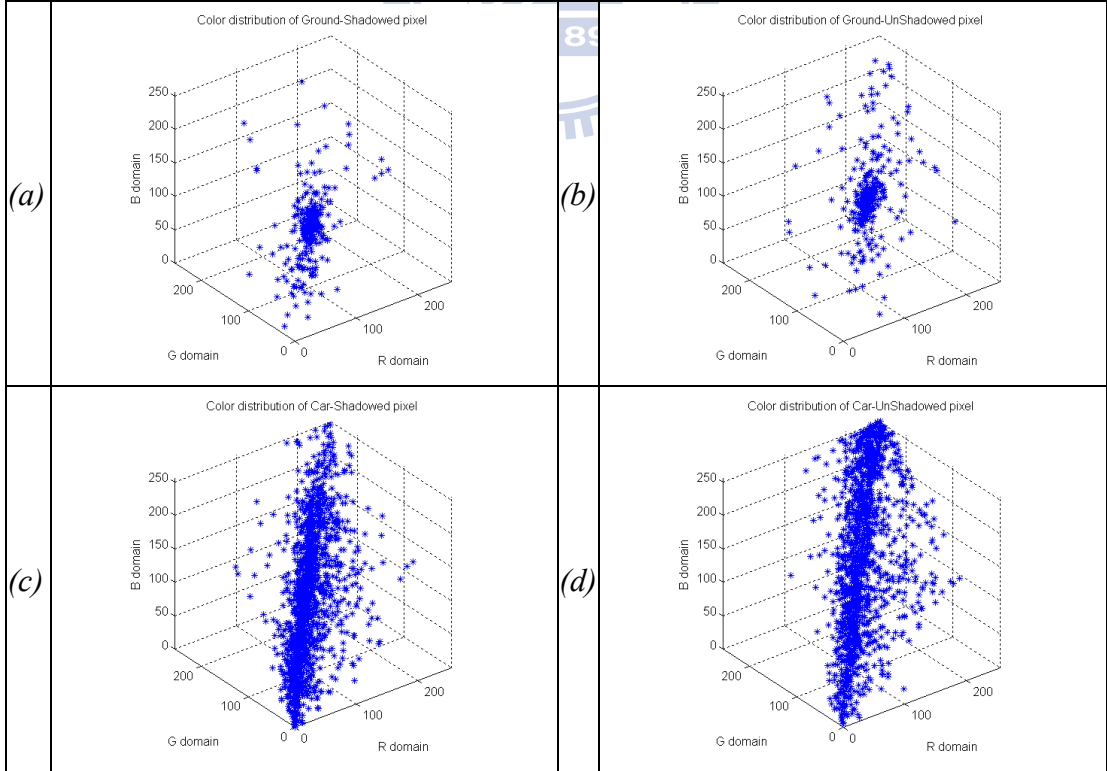


Fig. 26. The color distributions (a) of shadowed ground pixels, (b) of un-shadowed ground pixels, (c) of shadowed car pixels, and (d) of un-shadowed car pixels.

Since the lighting condition changes from time to time, we need to dynamically adjust $p(\mathbf{I}_{\text{RGB}}|h^O, h^L)$. Based on the image formation model explained in Appendix B, the trichromatic color vector \mathbf{I}_{RGB} at an image pixel can be represented as $\mathbf{I}_{\text{RGB}} = \|\mathbf{I}_{\text{RGB}}\| \mathbf{R} \mathbf{i}$, where $\|\mathbf{I}_{\text{RGB}}\|$ is the norm of \mathbf{I}_{RGB} , \mathbf{R} is a 3×3 matrix depending on surface reflectance, \mathbf{i} is a vector depending on illumination, and $\|\mathbf{R} \mathbf{i}\| = 1$. With this image formation model, we formulate $p(\mathbf{I}_{\text{RGB}}|h^O, h^L)$ as

$$p(\mathbf{I}_{\text{RGB}} | h^O, h^L) = p(\|\mathbf{I}_{\text{RGB}}\| | h^O, h^L) p(\mathbf{R} | h^O) p(\mathbf{i} | h^L). \quad (37)$$

Since the reflectance of target objects (ground or cars) can be learned beforehand but the lighting condition is varying over time, $p(\mathbf{R}|h^O)$ is learned off-line while $p(\mathbf{i}|h^L)$ and $p(\|\mathbf{I}_{\text{RGB}}\||h^O, h^L)$ are determined dynamically. Here, we build those probability models similar to the approach of [80] with a few modifications. First, instead of training the reflectance functions of only two objects (grass and ground in [80]) based on a single singular value decomposition (SVD) over one set of data, our application needs to collect the reflectance functions of various cars at different positions and at different time instants. This requires multiple SVD's over different sets of data. Hence, we need to register the solutions of different SVD's to deal with the ambiguity in SVD decomposition. Second, instead of clustering the daylight spectrums into only three classes, we determine $p(\mathbf{i}|h^L)$ dynamically to deal with the continuously changing lighting condition. Third, in [80], the trained chromaticity values of different classes are used to initialize the classification of image content. Their intensity model is then on-line determined. However, owing to the wide range of car appearance, some cars may get confused with the ground in the chromaticity space. In our approach, we add in the scene knowledge to dynamically determine the intensity model $p(\|\mathbf{I}_{\text{RGB}}\||h^O, h^L)$. Basically, given an image, there are two types of light: skylight and sunlight. Moreover, the ratio of reflectance between any two scene

patches can be well learned in advance. These two facts offer a possibility to on-line determine the intensity model of scene patches based on a few reference patches. Below, we explain the details of our approach.

4.4.1.2 Learning of $p(\mathbf{R}|h^O)$

In our experiments, we collected 5000 training samples of ground and cars to learn $p(\mathbf{R}|h^O=G)$ and $p(\mathbf{R}|h^O=C)$, respectively. Since the camera pose in our system is fixed, the captured images can be easily registered. To get the reflectance function of an object, we select a small surface patch with uniform illumination. To simplify the problem, we normalized \mathbf{I}_{RGB} by its norm to get the normalized RGB $\mathbf{I}_{\text{RGB}}^N = \mathbf{I}_{\text{RGB}} / \|\mathbf{I}_{\text{RGB}}\|$. Assume there are P pixels inside the patch and we collect the samples for F registered frames. The illumination condition is the same for the whole patch at a certain time instant, but could be different at different time instants. On the contrary, the reflectance function could be different at different image pixels but is temporally invariant at the same pixel. Hence, for an image pixel at the spatial location \bar{p} , its normalized RGB value at time instant k can be expressed as

$$\mathbf{I}_{\text{RGB}}^N(\bar{p}, k) = \mathbf{R}(\bar{p})\mathbf{i}(k). \quad (38)$$

By arranging the normalized RGB values of all pixels inside the surface patch over F frames into a $3P \times F$ matrix, we obtain the following formula

$$\mathbf{M}_{\text{RGB}}(\bar{\mathbf{p}}, \mathbf{k}) \equiv \begin{bmatrix} \mathbf{I}_{\text{RGB}}^N(\bar{p}_1, k_1) & \dots & \mathbf{I}_{\text{RGB}}^N(\bar{p}_1, k_F) \\ \vdots & \ddots & \vdots \\ \mathbf{I}_{\text{RGB}}^N(\bar{p}_P, k_1) & \dots & \mathbf{I}_{\text{RGB}}^N(\bar{p}_P, k_F) \end{bmatrix}_{3P \times F} = \begin{bmatrix} \mathbf{R}(\bar{p}_1) \\ \vdots \\ \mathbf{R}(\bar{p}_P) \end{bmatrix}_{3P \times 3} [\mathbf{i}(k_1) \dots \mathbf{i}(k_F)]_{3 \times F} \equiv \mathbf{M}_{\mathbf{R}}(\bar{\mathbf{p}})\mathbf{M}_{\mathbf{i}}(\mathbf{k}), \quad (39)$$

where $\bar{\mathbf{p}} = \{\bar{p}_1, \dots, \bar{p}_P\}$ is the spatial locations of the P pixels and $\mathbf{k} = \{k_1, \dots, k_F\}$ is the temporal indexes of the F frames.

Given \mathbf{M}_{RGB} , we can decompose it into a reflectance matrix $\mathbf{M}_{\mathbf{R}}$ and an illumination matrix $\mathbf{M}_{\mathbf{i}}$, up to a 3×3 non-singular matrix \mathbf{Q} . That is, if $\mathbf{M}_{\mathbf{R}\mathbf{I}}$ and $\mathbf{M}_{\mathbf{i}\mathbf{I}}$ is

a pair of matrices that decompose \mathbf{M}_{RGB} , then $\mathbf{M}_{\text{R2}}=\mathbf{M}_{\text{R1}}\mathbf{Q}$ and $\mathbf{M}_{\text{i2}}=\mathbf{Q}^{-1}\mathbf{M}_{\text{i}}$ is another decomposition pair. Fortunately, in the detection of vacant parking spaces, we only care about the difference in the surface reflectance matrix \mathbf{R} but not the true value of \mathbf{R} . As long as we fix the matrix \mathbf{M}_{i} , two surface patches with different \mathbf{R} will always have different \mathbf{M}_{R} .

To decompose \mathbf{M}_{RGB} , we applied the SVD process over several planar patches to collect samples for the ground reflectance function and car reflectance function. For the car samples, we select the car roof as the planar patch, which is usually parallel to the ground plane. To deal with the ambiguity in matrix decomposition, we collected a set of image frames and manually selected a ground region in the parking lot scene as the reference patch, shown as the red patch in Fig. 27(a). By performing the SVD decomposition over the reference patch, we got the reference truth \mathbf{M}_{R0} and \mathbf{M}_{i0} . The reference truth \mathbf{M}_{i0} is used to register the illumination matrix of another spatial patch that are under the same lighting condition in the same set of image frames. On the other hand, the reference truth \mathbf{M}_{R0} is used to register the reflectance matrix of the reference ground patch in another set of image frames. Based on SVD, with enough reflectance samples of cars and ground, we can construct the reflectance probability model $p(\mathbf{R}|h^O)$.

4.4.1.3 Learning of $p(\mathbf{i}|h^L)$

The illuminant probability model $p(\mathbf{i}|h^L)$ is determined based on the pre-trained model and the current image observation. Given an image, there are two types of regions: shadowed regions and unshadowed regions. By collecting many illumination samples \mathbf{i} 's in shadowed and unshadowed regions, we can approximate $p(\mathbf{i}|h^L=\text{'S'})$ and $p(\mathbf{i}|h^L=\text{'US'})$. Since the reflectance matrix \mathbf{R} of a scene patch can be learned in advance, we extract the illuminant component of some manually selected shadowed

and unshadowed regions to learn the off-line models $p_{\text{off}}(\mathbf{i}|h^L='S')$ and $p_{\text{off}}(\mathbf{i}|h^L='US')$. On the other hand, to deal with the continuously changing lighting condition, we also build the on-line models $p_{\text{on}}(\mathbf{i}|h^L='S')$ and $p_{\text{on}}(\mathbf{i}|h^L='US')$ based on the current image observation. The illuminant probability model is then determined based on a weighted combination of off-line and on-line models. That is,

$$p(\mathbf{i}|h^L='S') = \omega_1 p_{\text{on}}(\mathbf{i}|h^L='S') + (1-\omega_1) p_{\text{off}}(\mathbf{i}|h^L='S') \quad \text{and} \quad (40)$$

$$p(\mathbf{i}|h^L='US') = \omega_2 p_{\text{on}}(\mathbf{i}|h^L='US') + (1-\omega_2) p_{\text{off}}(\mathbf{i}|h^L='US'). \quad (41)$$

Here, ω_1 and ω_2 are determined by the ratio of the on-line training samples to the total training samples.

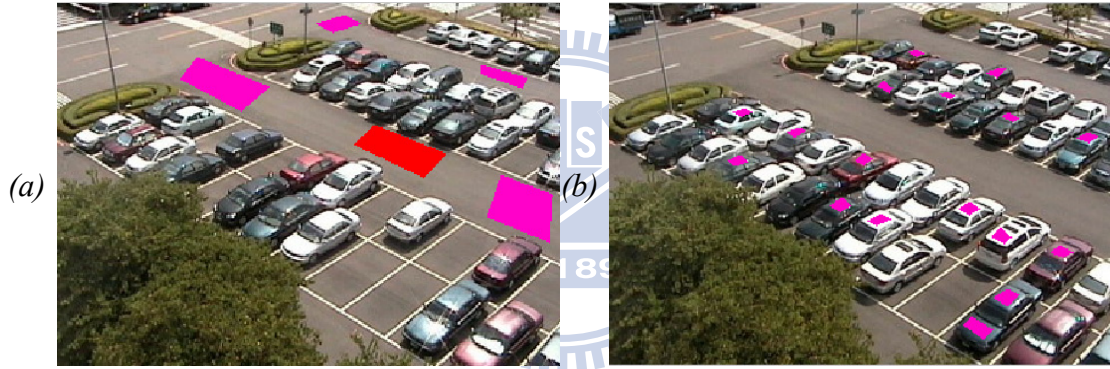


Fig. 27. (a) The reference ground patch (red) and the ground patches (pink) for the learning of ground reflectance function. (b) The car patches (pink) for the learning of car reflectance function.

During on-line modeling, we need to determine whether a given illuminant sample is shadowed or unshadowed. Here, for the period from 10:30 to 14:00, we suppose all samples are unshadowed. For the other periods, the lighting situation is more complicated. In our parking lot scene, we identified a few regions that are always unshadowed, like some regions in the driveway. These driveway regions can be used as the reference regions for the ‘unshadowed’ case for both skylight-plus-sunlight case and skylight-only case. On the other hand, as shown in Fig.

25(b), the green region in the bush in Fig. 25(a), together with all the other planes parallel to that green region, is only lighted by skylight in the morning; while the blue region in Fig. 25(a), together with all the other planes parallel to that blue region, is only lighted by skylight in the afternoon. These two types of regions can be used as the reference regions for the ‘shadowed’ case when both sunlight and skylight are present. In Section 4.5.1, we will further explain how we check the presence of sunlight in the current image.

4.4.1.4 Learning of $p(\|\mathbf{I}_{\text{RGB}}\| \mid h^O, h^L)$

The intensity information $\|\mathbf{I}_{\text{RGB}}\|$ is crucial in distinguishing cars from ground, especially when some cars may get confused with the ground in the chromaticity space. Unfortunately, $\|\mathbf{I}_{\text{RGB}}\|$ is affected by the lighting source, the object reflectance, the object geometry, and even some unknown factors in the imaging pipeline such as automatic gain control and white balance. Hence, the modeling of the intensity model $p(\|\mathbf{I}_{\text{RGB}}\| \mid h^O, h^L)$ is more difficult. To build an adaptive intensity model based on current image observation, we propose a simplified linear model as expressed in (42) to model the intensity mapping from one object type (O_1) in a scene patch to another object type (O_2) in another scene patch, under the same illumination type (L).

$$g_{O_2, L} = a_{O_2, O_1, L} \cdot g_{O_1, L} + n_{O_2, O_1, L}. \quad (42)$$

In (42), $g_{O, L}$ denotes an intensity sample from the object type O under the illumination type L . Note that $g_{O, L}$ value is equal to the norm $\|\mathbf{I}_{\text{RGB}}\|$ of a color pixel. $a_{O_1, O_2, L}$ represents the intensity ratio between objects O_2 and O_1 under illumination type L . $n_{O_1, O_2, L}$ is defined as a zero mean Gaussian noise that expresses the uncertainty in modeling the intensity ratio. Even though $a_{O_1, O_2, L}$ is actually a random variable, we

found a deterministic setting works very well in our experiments. Here, we learn $a_{o_1, o_2, L}$ and the variance of $n_{o_1, o_2, L}$ based on the following equations.

$$\hat{a}_{o_2, o_1, L} = \overline{g_{o_2, L}} / \overline{g_{o_1, L}}, \text{ and} \quad (43)$$

$$\hat{\sigma}_{n_{o_2, o_1, L}}^2 = \hat{\sigma}_{g_{o_2, L}}^2 - \hat{a}_{o_1, o_2, L}^2 \hat{\sigma}_{g_{o_1, L}}^2 \quad (44)$$

In (43) and (44), $\overline{g_{O, L}}$ and $\hat{\sigma}_{g_{O, L}}^2$ are the sample mean and sample variance of the intensity training samples. The training samples are manually collected from training image patches, with classified light type L and object type O .

In our system, a few transformation models were pre-learned to generate the intensity model $p(\|\mathbf{I}_{\text{RGB}}\| | h^O, h^L)$ dynamically. Here, we adopt the aforementioned reference regions, like the driveway regions that are always unshadowed and the bush regions that are always lighted by the skylight only. By using these reference regions, in which the lighting condition is already known, we learned the transformation models from each of these reference regions to the parking space ground and to the cars, respectively. After that, based on the learned transformation models and the current intensity values at these reference regions, we dynamically construct the intensity model $p(\|\mathbf{I}_{\text{RGB}}\| | h^O, h^L)$. Similar to the deduction of the sunlight direction, if the parking lot scene cannot provide such reference regions, an artificial cube is suggested to be set up in the scene to form reference regions.

4.4.2 Adjacency Energy Model

In the parking lot scene, the local decisions of two adjacent labeling nodes are usually highly correlated. In this system, with the use of the original intensity image $I_L(m, n)$, we define the adjacency energy $E_A[I_L(m, n), H_L(m, n); N_p]$ by using the smooth constraint explained in Section 3.3. Here, we briefly explain the design of the

adjacency energy model again.

In our system, the adjacency energy $E_A[I_L(m,n), H_L(m,n); N_p]$ is defined as

$$E_A[I_L(m,n), H_L(m,n); N_p] \equiv \beta \times \sum_{\Delta m=-p}^p \sum_{\Delta n=-p}^p C_A[I_L, H_L, m, n, \Delta m, \Delta n]$$

With this definition, if two neighboring sites are set to different labels, our system will give a larger penalty if we find the color difference between two sites is small. Otherwise, our system will give a smaller penalty. That is, two neighboring sites tend to share the same label when the difference between their color features is small, and tend to have different labels otherwise.

4.5 Vacant Parking Space Detection

4.5.1 Optimal Inference of Parking Space Status

With the top-down knowledge and the bottom-up message, we can infer the optimal H_L^* and S_L^* by solving the optimization problem in (10). In our approach, we get the initial guess of $H_L(m,n)$ by finding the labeling that minimizes the classification energy in (36). That is, we find the labeling image $H_L^i(m,n)$ such that

$$H_L^i(m,n) = \arg \min_{H_L} E_D[I_L(m,n), H_L(m,n)]. \quad (45)$$

On the other hand, since the status inference of a parking space depends on its neighboring parking spaces, we need to take into account relevant parking spaces when we infer the status of a parking space. In our experiments, a parked car casts a shadow to the right in the morning and to the left in the afternoon. Hence, we sequentially infer the status of each parking space from the bottom row to the top row and from left to right in the morning, and reverse the order in the afternoon. In Fig. 28, we show an example in the status determination of a parking space. Due to the

direction of sunlight, we check the parking spaces from left to right and from bottom to top. The red regions indicate those parking spaces whose status has already been inferred. The yellow circle indicates the parking space to be inferred at this moment. The green triangles indicate the relevant parking spaces. In this case, by trying different status combination of A and B spaces, four status hypotheses are to be tested. For each status hypothesis, we deduce the optimal $H_L(m,n)$ by using the graph-cuts algorithm, with the initial guess $H_L^i(m,n)$. The status hypothesis that achieves the maximum posterior probability is picked to infer the status of the current parking space. In our process, since the status of a parking space is only affected by its adjacent spaces, the system complexity grows linearly as the number of parking spaces increases.

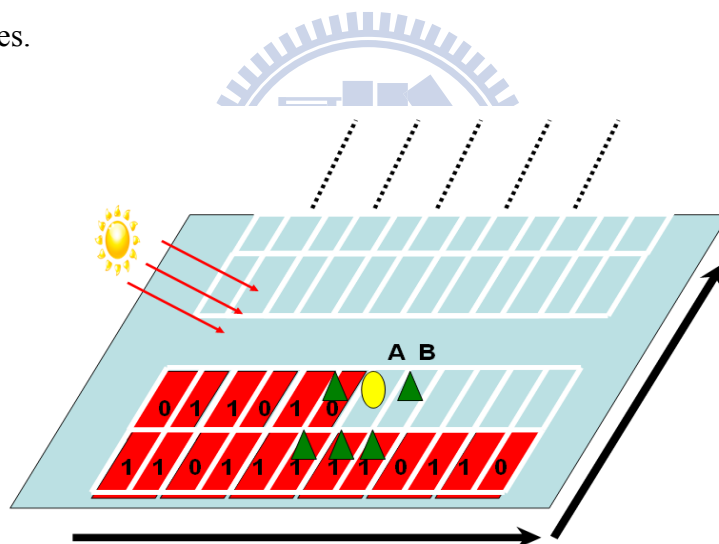


Fig. 28. Illustration of parking space status inference.

Moreover, in an outdoor environment, the sunlight does not always exist. In the inference of parking space status, we need to determine whether the sunlight is present or not. In our approach, we first perform the optimal labeling based on the assumption that sunlight is present. After the optimal inference for the whole image, we divide those “ground” pixels into shadowed pixels and unshadowed pixels. In principle, if the sunlight is present, the RGB values of these two pixel groups should

reveal obvious difference. Hence, by calculating the Davies-Bouldin index (DBI) [81], which is defined as

$$DBI = (S_S + S_{US}) / (\|\mu_S - \mu_{US}\|), \quad (46)$$

we can decide whether to accept the “presence” hypothesis or not. In (46), μ_S and μ_{US} are the mean RGB values of the shadowed cluster and the unshadowed cluster. S_S and S_{US} are the centroid distance of these two clusters defined as

$$S_c = \left(\sum_{i=1}^{n_k} \|f_i - \mu_c\| \right) / n_k, \quad (47)$$

where $c \in \{S, US\}$, n_k is the total pixel number of the cluster, and f_i is the RGB value of the i th pixel. When the DBI is smaller than a pre-defined threshold, we accept the “presence of sunlight” hypothesis. Otherwise, we take the “absence of sunlight” hypothesis and perform the optimal inference over the whole image again to get the final detection result.

4.5.2 Refinement of Classification Energy Model

In our system, after performing the optimal inference over an image, we obtain a semantic labeling (h^O, h^L) of the image that may provide useful information for the refinement of $p(\mathbf{I}_{\text{RGB}}|h^O, h^L)$. The inferred semantic labeling (h^O, h^L) includes not only the bottom-up information but also the top-down knowledge. With the inclusion of the top-down knowledge, some pixels, which would be incorrectly labeled if only based on the classification models, can be correctly labeled. Those pixels usually correspond to non-Lambertian surfaces, like the car windows. Hence, based on the inferred optimal labeling (h^O, h^L), we re-compute the classification model $p(\mathbf{I}_{\text{RGB}}|h^O, h^L)$ by checking the distribution of \mathbf{I}_{RGB} in the current image over different object types and different lighting types. The new model is then merged into the existing model for refinement:

$$P_{refind}(\mathbf{I}_{\text{RGB}} | h^O, h^L) = w_{new} \cdot P_{new}(\mathbf{I}_{\text{RGB}} | h^O, h^L) + w_{old} \cdot P_{old}(\mathbf{I}_{\text{RGB}} | h^O, h^L). \quad (48)$$

In (48), w_{old} and w_{new} determine the weights of the existing model and the new model. In our system, we empirically select (w_{old}, w_{new}) to be (0.2,0.8). Based on the refined model, the optimal labeling is re-estimated again. This optimization-refinement process is iteratively performed until the status inference of the parking spaces becomes stable. In our experiments, the refinement process usually converges in one or two iterations.

4.5.3 System Setup and Online Vacant Space Detection

To implement the whole system, several preparatory processes are required, as listed below.

1. Calibration Steps

- a. Define a 3-D coordination system for the parking lot. Measure the 3-D location of each parking space. Here, we record the 3-D information in a blueprint.
- b. Perform camera calibration to compute the camera projection matrix.

2. Offline learning of 3-D information

- a. Estimate the parameters of solar direction model based on the method introduced in Section 4.3.3.
- b. Collect 3-D training samples of vehicle length, width, and height to train the priors $p(l)$, $p(w)$, $p(h)$.
- c. Collect 3-D location deviation samples to train $p(X)$, and $p(Y)$.

3. Offline learning of 2-D information

- a. Collect reflectance samples to train the reflectance models of ground and cars, based on the method mentioned in Section 4.4.1.2.

- b. For different time period, manually select unshadowed and shadowed reference regions in the image.
- c. Collect illuminant samples to train the offline illuminant probability model of the shadowed regions and unshadowed regions, based on the method mentioned in Section 4.4.1.2.
- d. Based on the method mentioned in Section 4.4.1.2, learn the intensity mapping models from each of these reference regions to the ground and to the cars.

In our experiments, it took about five days to finish the above system setup processes for each parking lot. After system setup, the following processes are performed to dynamically detect vacant parking spaces.

- a. Determine the current sunlight direction based on the pre-learned solar movement model. This solar movement model is updated for every few days.
- b. Based on the learned 3-D information, the sunlight detection, and the projection matrix, generate the expected object and shadow labeling models.
- c. Extract illuminant samples from pre-selected reference regions to update the illuminant probability model.
- d. Based on the pre-learned intensity mapping models, establish the intensity model of different classes.
- e. Combine object reflectance models, illuminant probability models, and intensity models to build the classification models.
- f. Incorporate classification models, expected labeling models, and adjacency model into the BHF to detect vacant parking spaces.

4.6 Experiment Results and Discussion

4.6.1 Experiment Setup and Test Data

In our experiments, we tested two different parking lots for performance evaluation. In each test, we set up an IP camera on the roof of a building near the parking lot. The camera was geometrically calibrated beforehand and monitored the status of parking spaces from morning to evening. Both experiments report similar detection accuracy. To avoid confusion, we mainly present the results and the analysis over the first parking lot. At the end of this section, we briefly present the detection performance over the second parking lot.

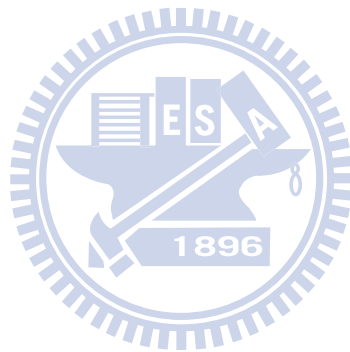
Fig. 18 shows a few image shots of the first parking lot. Within the image view, there are 46 parking spaces in total. To evaluate the performance of our system, we tested three image sequences under different weather conditions. The first sequence was captured in a normal sunny day. The second sequence was captured in a day with very strong sunlight so that there were plentiful over-exposed regions in the images. The third sequence was captured in a day with unstable lighting condition. In this sequence, the lighting condition dramatically switched between sunny and cloudy. For each sequence, the recording time was from 8:00am to 5:00pm. Since the status of the parking condition was slowly changing, we performed vacant parking space detection for every five minutes. In total, we tested the status of 14766 spaces. In these three sequences, the shadow patterns varied from morning to evening. Sometimes, the shadowed regions suddenly disappeared when the sunlight was blocked by a cloud. The variations of illumination caused apparent drifts in color and brightness. These three sequences with vacant space detection results and ground truth are available at our website [82].

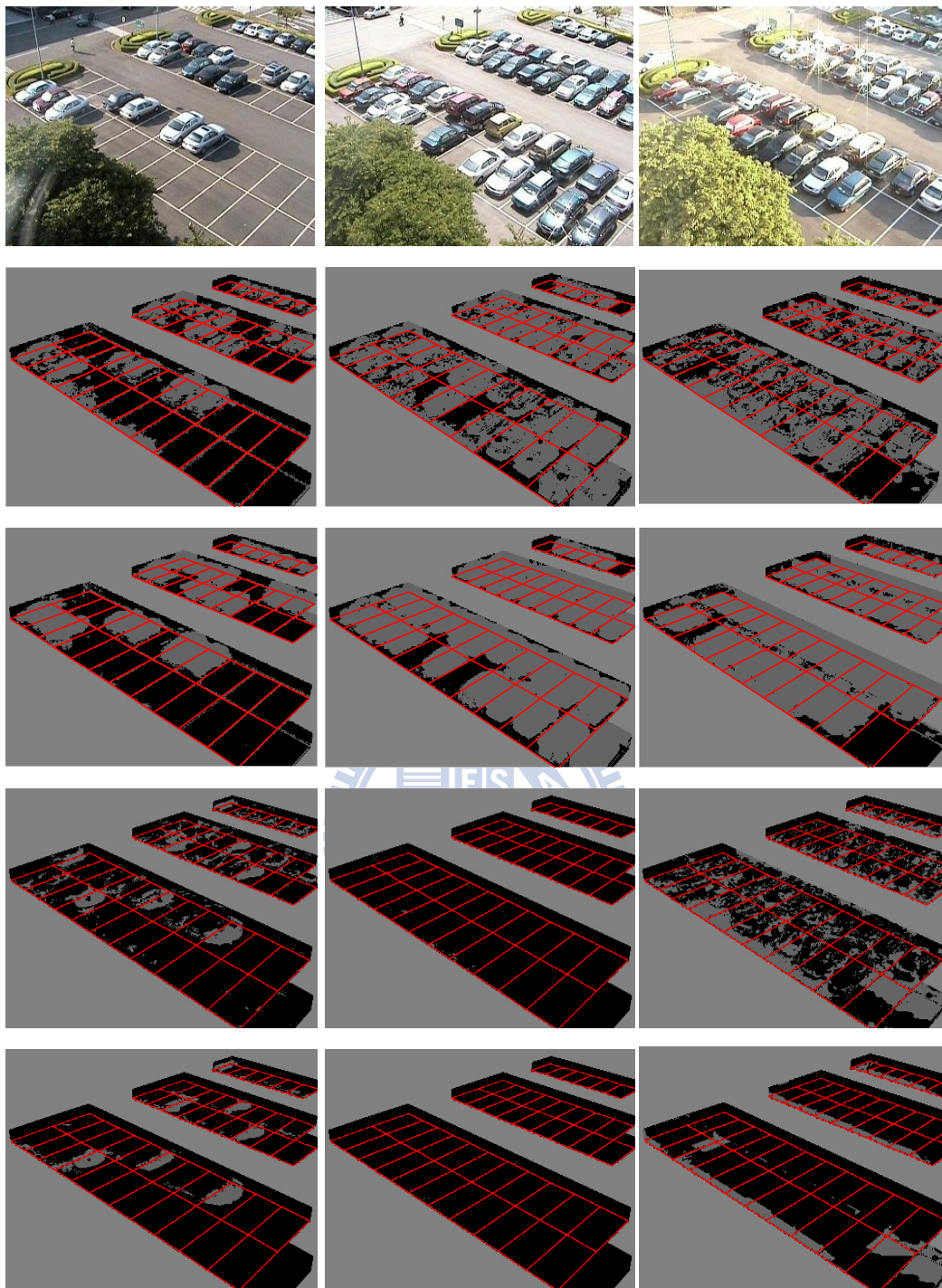
4.6.2 Object/Shadow Labeling and Accuracy of Vacant Space Detection

Many previous studies suggested the vacant spaces be detected by labeling the car pixels, such as Tsai et al. [13], or by labeling the ground pixels, such as Funck et al. [11]. In our method, we modeled both cars and ground plane for object labeling. In Fig. 29, we compare the results of car pixel labeling based on Tsai’s method [13] and ours. Here, we show the image portions that were labeled as “car”. Based on Tsai’s method, many shadowed ground regions were labeled as car pixels, many over-exposed car regions were labeled as ground pixels, and some car regions were mistakenly labeled as ground pixels. In comparison, our parking space detection system provided more accurate car regions and was less sensitive to the shadow effect. In Fig. 30, we compare the results of ground pixel labeling based on [11] and our method. Both [11] and our method used adaptive models for labeling. However, the method in [11] did not take into account the shadow effect and many shadowed ground regions were classified as car pixels. In comparison, most shadowed ground regions are correctly identified by our method.

Even though the proposed adaptive models can better handle the shadow effect, many pixels were still misclassified if the scene knowledge was not involved. An example is presented in Fig. 31, where we show the labeling results with and without the scene knowledge. Especially, there were some pepper-like errors inside the car regions as shown in Fig. 31(c) which were caused by the ambiguity in color appearance. It is difficult to remove those errors if we only rely on color models. In our system, the scene information in the expected labeling maps provides constraints to remove that kind of errors. To deal with the color ambiguity between dark cars and shadowed ground, the expected shadow labeling map clearly constrains the location

of shadowed regions. On the other hand, if a region is to be occupied by a car, the expected object labeling map reveals the probable regions of car pixels and disfavors the occurrence of pepper-like labeling. Moreover, the expected object labeling map also reveals the expected occlusion effect and the perspective distortion. By taking into account these kinds of scene knowledge, more accurate and reliable detection results were obtained, as shown in Fig. 31.





(a)

(b)

(c)

Fig. 31. The detection and labeling results at three different time instants. For each case, the images from the top are the test image, the car labeling without scene knowledge, the car labeling with scene knowledge, the shadow labeling without scene knowledge, and the shadow labeling with scene knowledge.

To assess the detection accuracy of our system, we manually built the ground truth of 14766 parking spaces. To evaluate our system from different aspects of environmental variations, we assessed the detection performance over a day, over different periods of a day, and over different regions of the parking lot. To quantitatively evaluate the performance, the false positive rate (FPR), false negative rate (FNR), and system accuracy (ACC) were calculated. In our simulation, the methods proposed by Dan [76], Wu et al. [77], and Huang et al. [46] were tested for comparison. The Receiver Operating Characteristic (ROC) curves of the four methods are also plotted in Fig. 32 for comparison. Here, we consider three test image sequences. For each image sequence and each method, the area under the ROC curve (AUC) is also calculated and provided in the Fig. 32 for reference.

As listed in

Table 1, the proposed method worked well in all three test sequences. We further divide a day into three periods: morning (8:00~11:00), noon (11:00~14:00), and afternoon (14:00~17:00). Generally, the afternoon period has the most serious shadow effect, while the noon period has almost no shadow at all. By calculating the ACC of those three periods, we found the ACC is inversely proportional to the degree of shadow effect. Moreover, we also evaluated the performance of detection over different regions to evaluate the influence of perspective distortion. As shown in

Table 1, perspective distortion does not cause serious degradation in our experiments. Moreover, even though some portions of the 1st row were occluded by the trees, the proposed system still accurately inferred the status of the parking spaces.

We also implemented our system in another parking lot. For each 320×240 tested image, there are 64 spaces inside. In total, we tested the statuses of 6912 spaces in that parking lot. In Fig. 33, we show some detection results in the second parking lot. The ACC , FPR , and FNR are 0.988, 0.0185, and 0.0097 respectively. The complete

detection results of the second parking lot are also available at our website.

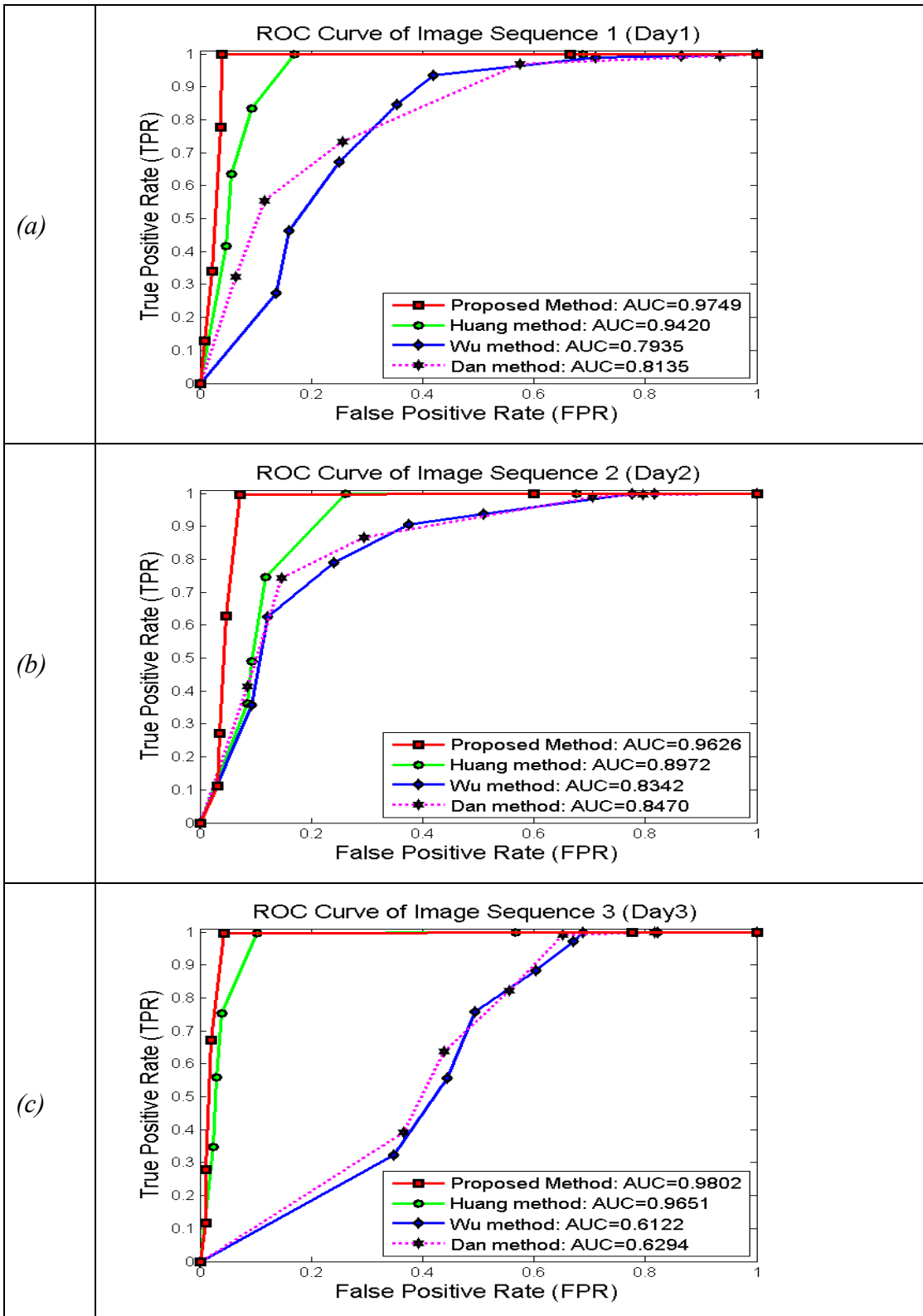


Fig. 32. The Receiver Operating Characteristic (ROC) curves of our method, Huang's method [46], Wu's method [77], and Dan's method [76], with the values of the area under ROC (AUC) for (a) "Day 1" (b) "Day 2", and (c) "Day 3" image sequences.

Table 1. Performance comparison of four vacant space detection algorithms.

Test Data	# of tested spaces		Proposed method			Huang [46]			Wu [77]			Dan [76]		
	vacant	parked	FPR	FNR	ACC	FPR	FNR	ACC	FPR	FNR	ACC	FPR	FNR	ACC
Image Seq. 1 (Day 1)	491	4431	0.0004	0.0081	0.9988	0.0004	0.1690	0.9827	0.0111	0.7115	0.9193	0.0307	0.5748	0.9153
Image Seq. 2 (Day 2)	278	4644	0.0024	0.0324	0.9959	0.0002	0.2626	0.9850	0.0016	0.7837	0.9577	0.0101	0.7061	0.9537
Image Seq. 3 (Day 3)	206	4716	0.0040	0.0437	0.9943	0.0042	0.1019	0.9917	0.0018	0.7012	0.9739	0.0073	0.6524	0.9703
Morning period of 3 Seq.	380	4588	0.0031	0.0105	0.9964	0.0011	0.2026	0.9835	0.0004	0.4955	0.9646	0.0097	0.4478	0.9594
Noon period of 3 Seq.	367	4601	0.0015	0.0082	0.9980	0.0015	0.0381	0.9958	0.0045	0.8632	0.9360	0.0179	0.7629	0.9306
Afternoon period of 3 Seq.	228	4602	0.0024	0.0658	0.9946	0.0024	0.3772	0.9799	0.0091	0.8920	0.9502	0.0195	0.6948	0.9494
1 st & 2 nd rows of 3 Seq.	644	6739	0.0019	0.0233	0.9962	0.0025	0.1770	0.9823	0.0068	0.6960	0.9377	0.0179	0.5641	0.9381
3 rd & 4 th rows of 3 Seq.	98	5359	0.0015	0.0306	0.9980	0.0009	0.3163	0.9934	0.0028	0.6933	0.9871	0.0059	0.6933	0.9840
5 th row of 3 Seq.	233	1693	0.0065	0.0172	0.9922	0.0006	0.1373	0.9829	0.0024	0.8240	0.8982	0.0366	0.7554	0.8764

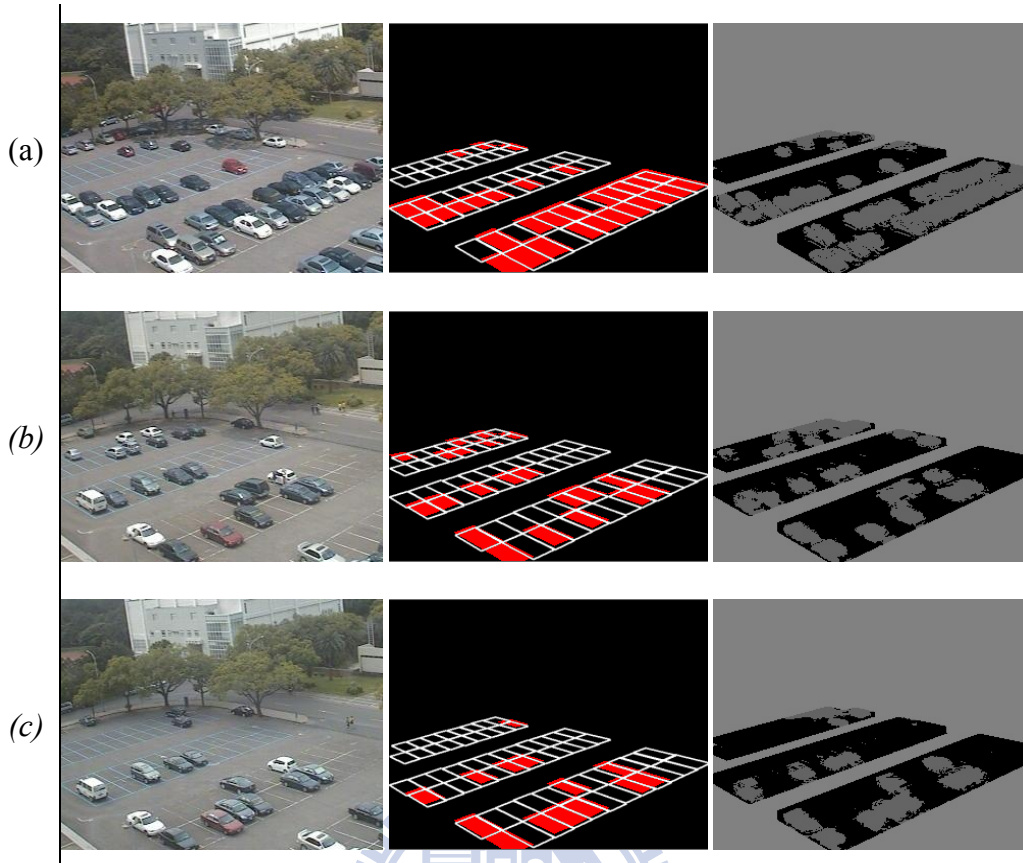


Fig. 33. The proposed detection and labeling results at three different time instants in another parking space. For each case, the images from the left are the test image, the parking space detection results, and the car labeling results.

4.6.3 Discussion and Future Works

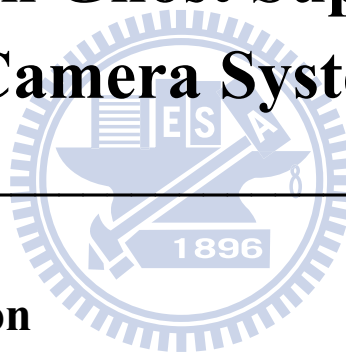
The whole system has been implemented in the Visual C++ environment on a PC with a 2.0GHz Pentium-4 CPU. It takes about 30 seconds to perform the space detection and labeling of parking spaces for a 320×240 color image with 46 spaces inside. The major CPU time is spent on building the online models, including the expected object labeling model, the expected shadow labeling model, and the color classification model. Even though the execution time takes a little while, the speed of the proposed system is still reasonably fast to support practical parking space detection systems. Although the complexity of our system is already affordable for practical applications, the speed can be further boosted if we either adopt parallel

programming techniques, such as Open Multi-Processing (OpenMP), to fully use the computing power of a multi-core processor, or to adopt General-purpose computing on graphics processing (GPGPU).

In our system, people in the parking lot may affect the detection of vacant parking spaces. However, people tend to dynamically move in the scene. By taking the temporal information into consideration, the problem of walking pedestrians can be relieved. On the other hand, even though our system works very well in an outdoor parking area during the daytime, there exist still several challenging issues, like how to manage an indoor parking area, how to detect vacant spaces in an outdoor parking lot during the night, and how to handle the unexpected shadow caused by other environmental objects. For an indoor parking area, the severe occlusion and the limited camera field of view could be the major challenges. Considering cost and efficiency, a possible solution is to build a low-cost camera sensor network. To detect vacant spaces in evening, we may need to consider multiple lighting sources while generating the expected shadow maps. We also require a new mechanism to handle the unpredictable lighting change caused by car headlights. All these discussions would be the future works of our vacant parking space detection system.

CHAPTER 5

Multi-Target Correspondence and Labeling with Ghost Suppression over Multi-Camera System



5.1 Introduction

In recent years, plentiful vision based techniques have been investigated to boost intelligent functionalities of modern surveillance systems. Among those technologies, object detection and labeling are especially crucial. For a single-camera system, these two processes are the fundamental steps for advanced analyses, like object tracking and behavior understanding. Up to now, many frameworks have been used to detect and label targets of interest. For example, Schneiderman and Kanade [68] proposed a trainable object detector for the detection of faces and cars, based on the statistics of localized parts. Adaboosting detection algorithm [69] is another widely used technique for the detection of specific objects in 2-D images. However, since a 2-D image lacks 3-D depth information, the detection of targets usually suffers from the

occlusion problem, especially when multiple targets appear in a complicated scene.

An alternative way to deal with the occlusion problem is to use a multi-camera system. The cross reference of multiple camera views can effectively handle the occlusion problem and provide a reliable way for object labeling and correspondence. Up to now, several multi-camera surveillance systems have been proposed for multi-target correspondence. These approaches can be roughly classified into two major categories – “direct correspondence” and “indirect correspondence”. For a “direct correspondence” approach, moving objects are detected in each 2-D camera view first. After that, object correspondences are built among 2-D camera views and 2-D detection results in different camera views are fused together to support surveillance over the 3-D space. For instance, In [83], Khan et al. found the overlapped fields of view among cameras. Whenever a moving object enters an overlapped region, the correspondence of this object with respect to its counterparts in other camera views can be established. In [84], Hu et al. proposed a principal axis-based correspondence among multiple camera views. This method offers robust results and can tolerate a certain level of defects in the motion detection and segmentation of each camera view. Moreover, the typically required camera calibration step is not a necessity in their system. In [85], Black and Ellis established the correspondence by comparing the distance between the projected epipolar lines and the detected objects in each 2-D image. For a multi-camera system with a narrow baseline setup, the use of epipolar constraint provides an efficient way to establish the correspondence.

Basically, most “direct correspondence” approaches require the foreground regions of each target be correctly extracted in each camera view to ensure reliable correspondence. However, with the presence of occlusion, this requirement cannot be easily achieved. In [86][87], Mittal and Davis launched the correspondence of objects

by matching the color appearance of segmented regions along epipolar lines in pairs of camera views. In their approach, the mid-points of the matched regions are projected onto the 3-D space to yield a 3-D probability distribution map for the description of object position. Although this method may relax the need of accurate foreground extraction, it has the extra requirement of color calibration among multiple cameras. Incorrect correspondence may also occur while matching objects with similar color appearance.

In the “indirect correspondence” category, a multi-camera system fuses multi-view information onto a pre-selected data-fusion space. The fused information is then projected back to each camera view to build object correspondence. Typically, the 3-D space is chosen as the space for data fusion. For example, Utsumi et al. [88] proposed the adoption of intersection points, which are the intersections of the 3-D lines emitted from the 2-D tracking results of different camera views. In that approach, a mixture of Gaussian functions was used to describe the possible positions of moving objects in the 3-D space. By projecting these 3-D Gaussian distributions back to individual 2-D image plane, the object correspondence among camera views is derived in a probabilistic manner. On the other hand, Fleuret et al. [89][90][91] adopted a simple blob detector in 2-D analysis and introduced a generative model to fuse data from multiple views. In their system, a discrete occupancy map is designed to describe whether an individual target is standing at a specific ground location in the 3-D space. After that, the most likely trajectory of each individual over the 3-D ground plane is traced via the Viterbi algorithm. In [92][93], Huang and Wang proposed a model-based approach to efficiently fuse consistent 2-D foreground detection results from multiple camera views. A probabilistic method is further proposed to simultaneously label and map multiple targets based on a Markov network.

Instead of fusing multi-view information onto the 3-D space, Khan and Shah [94] chose one of the 2-D camera views as the reference view for data fusion. In their approach, without relying on complicated camera calibration, they built a few homography matrices to map the projected ground planes in multiple camera views. After that, they fused the foreground likelihood information from multiple views to the scene plane in the reference camera view in order to generate a probability map of the target location. Owing to the geometric consistence, the fused target location probability map, named the “synergy map” in [94], would indicate a higher probability for a true target location. The synergy map was finally rectified so that the target location on the reference image is remapped to the relative ground plane location in the 3-D space. Since the fused synergy map is built over a 2-D image space, the spatial resolution of the target location is influenced by the perspective projection and is non-uniform in the 3-D space. A target far away from the reference camera would have a lower location resolution, while a target close to the reference camera would have a higher resolution. In addition, it is a little complicated to utilize the prior knowledge of the 3-D targets into this 2-D fusion framework.

For these aforementioned “indirect correspondence” approaches, certain geometric ambiguity may cause “ghost objects” in the 3-D space. The ghost effect is another form of the inter-occlusion problem and is a classic problem in 3-D object reconstruction. Owing to the limited number of cameras around the surveillance zone, some ghost objects may occasionally fulfill the geometric consistency and appear in the reconstructed 3-D scene. These fake targets could severely affect the accuracy in building object correspondence. In recent years, several approaches have been proposed to suppress ghost objects in multi-camera applications. Including the aforementioned method in [94], most methods used the temporal consistency to remove ghost targets. For example, given a limited number of 2-D camera views,

Otsuka and Mukawa [95] proposed a framework of multi-view occlusion analysis to track objects. Once if occlusion patterns are detected, some occlusion hypotheses are launched to indicate the uncertainty caused by occlusion. Since an occlusion structure usually lasts only for a short period, those hypotheses are tested recursively based on the temporal consistency to suppress fake detection. In [96], on the other hand, Guan et al. suppressed ghost targets by considering the consistency of color appearance. By projecting 3-D objects onto different image views, they identify ghost objects based on dissimilarity of colors. Moreover, their approach may automatically learn the appearance models for different objects in different camera views during the tracking process. This eliminates the requirement of color calibration among different cameras.

In this dissertation, we propose a new approach to efficiently integrate, summarize, and infer video messages from multiple client cameras. Even though we only use a simple foreground object detector to obtain imperfect foreground detection results, our system can still efficiently determine the number of moving targets inside the surveillance zone and accurately track the 3-D trajectories of the tracked targets. Besides, our approach can perform image labeling in a pixel-level manner and match targets among multiple camera views. The rest of this chapter is organized as follows. In Section 5.2, we present the main idea of the proposed framework, which is composed of a data fusion stage and an inference stage for multi-target labeling and correspondence. In Sections 5.3 and 5.4, we explain the details of the fusion stage and the inference stage, respectively. Experimental results and discussions are presented in Section 5.5.

5.2 System Overview

In this system, we focus on a client-server surveillance setting, which monitors a zone with multiple client cameras. The main goal of our system is to detect, locate,

correspond, and label multiple targets, especially for walking people in the zone. Without knowing the number of targets in advance, it would be a challenge to efficiently analyze the inter-occlusion situation among targets while locating and labeling targets.

To handle the inter-occlusion problem, previous works [89][90][91][96] checked the possible points over a discrete domain, like a lattice of discrete ground locations or a set of 3-D voxels. At each point, a random variable is attached to represent the probability of having a target at that point. By considering the joint probability among random variables and the relative position among targets, the inter-occlusion situation can be well modeled and the moving targets can be detected. Basically, those previous works couple the detection of candidate locations with the analysis of inter-occlusion. This coupling leads to a trade-off between location accuracy and computational cost.

In our approach, we decouple the detection of target locations from the analysis of inter-occlusion. The basic idea is to detect the candidate target locations in the first stage and then spend computations only over those candidate locations for inter-occlusion analysis. This two-stage procedure may preserve the accuracy of target location without dramatically increasing the computational cost.

5.2.1 System Property

In our system, we adopt an “indirect correspondence” approach that fuses 2-D information from a set of calibrated cameras to perform labeling and correspondence of multiple targets in the surveillance zone. The proposed scheme has two major features. First, to suppress the ghost targets caused by geometric ambiguity, the 3-D scene model in our framework is defined in a probabilistic manner. Second, instead of applying a fixed 3-D target model to all tracked targets, we use the BHF (Bayesian Hierarchical Framework) structure with an expectation-maximization mechanism to

on-line refine the 3-D target model for each individual target. Moreover, our system can locate, correspond, and label multiple targets over a multi-camera surveillance system, with the capability of ghost suppression and target model refinement.

If compared with other relevant works, the proposed system includes three major contributions. First, we introduce a fusion-inference procedure to decouple the detection of target locations from the analysis of inter-occlusion so that the trade-off between location accuracy and computational cost are relieved. Second, in the fusion stage, we suggested a model-driven approach to achieve more robust fusion under imperfect foreground detection. Third, in the inference stage, the labeling, correspondence, and inference of 3-D target model, together with the suppression of ghost targets, are modeled in a unified framework and are resolved via an optimization process. Under the proposed system, we can systematically estimate the target number and tackle the inter-target occlusion problem. Moreover, the proposed system requires neither accurate foreground/background separation nor color calibration among multiple cameras.

5.2.2 System Flow

In our fusion-inference scheme, we design a data fusion stage to detect candidate targets and their 3-D locations. After that, target identification, image labeling, and inter-occlusion are analyzed under the proposed BHF framework in the inference stage. The inferred target labeling and correspondence results are further used to refine the 3-D target model. In Fig. 34, we illustrate the system flow of the proposed system.

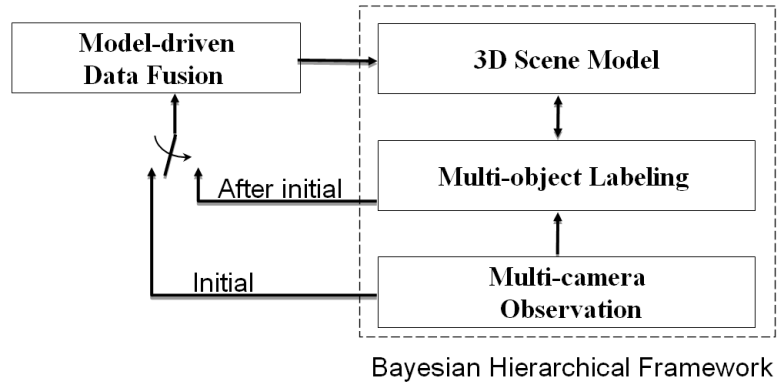


Fig. 34. System flow of the proposed system.

In the data fusion stage, a model-based approach is used to efficiently fuse consistent 2-D foreground detection results from multiple camera views. Here, we formulated a posterior distribution, named target detection probability (TDP), as the fused message pool to indicate the probability of having a moving target at a certain ground location. Based on the TDP distribution, the candidate targets and their locations can be identified in a probabilistic manner, which combines a sample-based representation of TDP and Mean-Shift clustering [97]. Moreover, with the use of 3-D target model, our fusion scheme may work reasonably well even with imperfect foreground extraction.

After data fusion, a set of candidate targets are detected that include both true targets and ghost targets. Since the occurrence of ghost targets is geometrically consistent with the 2-D foreground detection results, existing methods attempt to suppress ghosts by checking some other properties, like photometric consistency and temporal consistency. In our system, we use a few 3-D priors about the surveillance scenario, such as the assumption that human stands on the ground plane, the probability distribution of the target height, and the probability distribution of the target location, to distinguish true targets from ghost targets. By properly integrating these 3-D priors into the scene knowledge, we can greatly simplify the ghost problem.

Moreover, in this system, we used the BHF framework to unify the processes of target labeling, target correspondence, and ghost suppression into a Bayesian inference process. Here, the labeling layer in BHF not only plays an intermediate role in the hierarchical framework but also provides a feedback route to refine the scene knowledge based on an EM (Expectation-Maximization) mechanism. In the following sections, we will explain in detail how we design the fusion stage and the inference stage of our system.

5.3 Information Fusion and Summarization

5.3.1 Foreground Detection on Single Camera

To fulfill the speed requirement of a real-time multi-camera system, we only consider the 2-D foreground detection results as the observation data. In our system, the intrinsic and extrinsic parameters of all cameras are well calibrated beforehand. For each camera, we build its reference background based on the Gaussian mixture model (GMM) [98]. The foreground image is determined by checking the frame difference between the current image and the reference background in a pixel-level manner. Besides, to remove shadows, the frame difference operation is performed over the chromatic domain, rather than the achromatic domain. However, although the GMM background subtraction method can deal with gradually changing illumination through on-line background learning, it may still falsely reject some foreground pixels whose appearance happens to be similar to that of the reference background. As shown in Fig. 42(b), Fig. 43(b), and Fig. 44(b), the detected foreground objects are usually neither perfectly silhouetted nor well connected.

5.3.2 Information Fusion

In the fusion step, we integrate the 2-D foreground detection results from a set of camera views to offer global 3-D information. To fuse 2-D information, most existing methods adopt a data-driven approach to back-project the 2-D foreground regions into a 3-D visual hull, as plotted in blue in Fig. 35(a). By accumulating the number of voxels of the visual hull along the normal direction of the ground plane, we can build a histogram that indicates the likelihood of having a candidate target on the ground plane, as illustrated in Fig. 35(b). However, since the extracted 2-D foreground silhouettes are usually fragmental and far from perfect, the reconstructed visual hull could be very different from the original 3-D target and the deduced voxel histogram could be seriously biased from the true location, as illustrated in Fig. 35(c) and (d).

To improve the accuracy in the estimation of target location, we adopt a model-driven approach to fuse 2-D information. In the proposed method, a so-called Target Detection Probability (TDP) distribution is defined to estimate the probability of having a moving target at a ground location. In Fig. 35(f), we show the estimated TDP distribution based on the incomplete foreground images in Fig. 35(e). It can be seen that the model-based approach provides a more reliable estimation of the target location. The detail of this model-driven approach is to be explained as follows.

In our approach, the TDP distribution is formulated as a posterior distribution, which is expressed below based on the Bayes rule:

$$G(X) \equiv p(X | F_1, \dots, F_N; \Theta) \sim p(X)p(F_1, \dots, F_N | X; \Theta). \quad (49)$$

In (49), X represents a location (x_1, x_2) on the ground plane of the 3-D space. N is the total number of cameras in the multi-camera system. F_i denotes the foreground detection result of the i th camera view. Θ defines the set of camera parameters of all N cameras. To simplify the formulation, we'll ignore Θ in the following deductions. Moreover, $p(X)$ is used to define the prior information about the targets' possible

locations in the surveillance zone. If there is no specific knowledge about the possible locations of the moving targets, we can simply define $p(X)$ to be uniformly distributed over the ground plane of the surveillance zone.

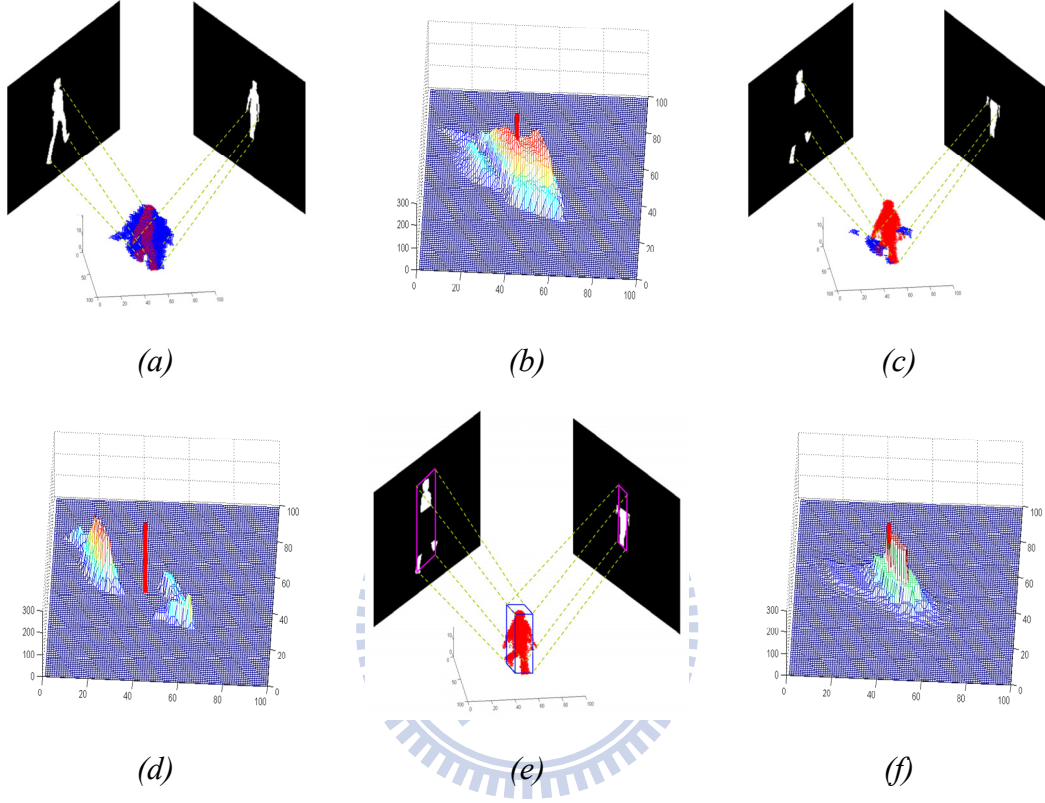


Fig. 35. (a) Visual hull constructed from the foreground images of two camera views. (b) The voxel histogram based on the visual hull in (a). (c) Visual hull constructed from fragmented foreground images. (d) The voxel histogram based on the visual hull in (c). (e) The proposed pillar model in the 3-D space. (f) The estimated TDP distribution based on the foreground images in (e). (The red bar in (b)(d)(f) represents the true target position.)

To define F_i , we use (m,n) to represent the 2-D coordinate system of the i th camera. If this camera has the image size $M_s \times N_s$, we define the image view V of the i th camera to be the set of (m,n) with $0 \leq m \leq (M_s-1)$ and $0 \leq n \leq (N_s-1)$. With this notation, based on the foreground detection result on the i th camera view, we define F_i as

$$F_i(m,n) = \begin{cases} 1 & \text{if } (m,n) \in V \text{ and } (m,n) \in \text{foreground regions} \\ 0 & \text{if } (m,n) \in V \text{ and } (m,n) \in \text{background regions} \\ P_L & \text{if } (m,n) \notin V \end{cases} \quad (50)$$

In (50), P_L is a trainable constant designed to indicate the possibility that there could be some other foreground objects out of the field of view of the i th camera.

Moreover, given the location X , we assume the foreground detection results are conditionally independent of each other. With this assumption, we rewrite (49) as

$$p(X)p(F_1, \dots, F_N | X) = p(X) \prod_{i=1}^N p(F_i | X). \quad (51)$$

To formulate $p(F_i|X)$, we model a moving person at the ground position X as a rectangular pillar, as shown in Fig. 35(e). The height H and width R of the rectangular pillar are modeled as independent Gaussian random variables, with their priors $p(H)$ and $p(R)$ being pre-trained based on the training data collected from the health center of our university. Based on the pre-calibrated projection matrix of the i th camera, a target at X with height H and width R can be projected onto the image plane of the i th camera to obtain the projection regions. Here we define the projection image M_i on the i th camera view as

$$M_i(m,n | H, R, X) = \begin{cases} 1 & \text{if } (m,n) \in \text{projected regions} \\ 0 & \text{if } (m,n) \notin \text{projected regions} \end{cases} \quad (52)$$

Please note that the projected regions in (52) could be out of the image view V of the i th camera.

With F_i and M_i , the normalized overlapping area, Ω_i , is defined as

$$\Omega_i(H, R, X) \equiv \frac{\iint F_i(m,n)M_i(m,n | H, R, X)dmdn}{\iint M_i(m,n | H, R, X)dmdn}. \quad (53)$$

By taking into account the prior probabilities $p(H)$ and $p(R)$, an estimate of $p(F_i|X)$ is defined as

$$p(F_i | X) \equiv \iint \Omega_i(H, R, X)p(H)p(R)dHdR. \quad (54)$$

In our approach, (54) is calculated numerically based on the Monte Carlo approach. Here, we draw a set of sample pairs (H,R) based on the prior models $p(H)$ and $p(R)$. For each sample pair (H,R) and a target location X , we evaluate its correlation value Ω_i . By averaging the correlation values over all sample pairs, we estimate $p(F_i|X)$ in a statistical manner.

5.3.3 Representation of TDP and Information Summarization

To numerically calculate TDP, we calculate $G(X)$ over a K_n by K_n lattice on the ground plane. For each node X_i of the lattice, its value $W_i=G(X_i)$ indicates the probability of having an object at that location. The sample set $\{X_i, W_i\}_{i=0 \sim S-1}$, with $S = K_n^2$, is then used to approximate the TDP distribution. In our experiments, we set $K_n = 100$ and $S = 10000$.

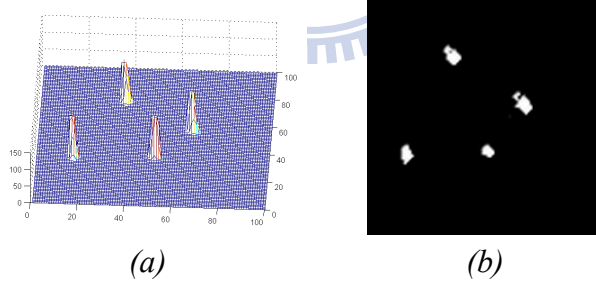


Fig. 36. (a) The TDP of four moving targets in the surveillance zone.
(b) The bird-eye view of (a).

Based on the TDP distribution, we summarize some useful information about the 3-D scene, including the number of candidate targets, the most likely position of each candidate target, and the unique ID of each candidate target. Typically, the TDP distribution contains several clusters, with each cluster indicating a moving target on the ground plane. Hence, the detection of multiple moving targets can be treated as a

clustering problem over the TDP distribution. In Fig. 36(a), we show an example of the TDP distribution, which are fused from the foreground detection results of four cameras. To perform clustering over the TDP distribution, we apply the Mean-Shift clustering algorithm [99] over the sample set $\{X_i, W_i\}_{i=1 \sim S}$. This mean-shift clustering method is efficient in mode searching and does not require the prior knowledge of the cluster number. By iteratively calculating the next position y_{j+1} based on the following equation

$$y_{j+1} = \frac{\sum_{i=0}^{S-1} X_i W_i \exp\left(\left\|\frac{y_j - X_i}{h}\right\|^2\right)}{\sum_{i=0}^{S-1} W_i \exp\left(\left\|\frac{y_j - X_i}{h}\right\|^2\right)}, \quad (55)$$

we can identify a few converging points [99]. In (55), h is a parameter that controls the kernel size. With the mean-shift algorithm, those samples that converge to the same converging point are grouped as the same candidate target and are assigned the same ID.

Based on the clustered groups, we determine the number of candidate targets. Moreover, assume we have identified M candidate targets on the ground plane with the ID's: T_1, T_2, \dots, T_M . If we denote the R_s samples that belong to T_k as $\{X_{k,0}, X_{k,1}, \dots, X_{k,R_s-1}\}$ with the corresponding weights as $\{W_{k,0}, W_{k,1}, \dots, W_{k,R_s-1}\}$, we can estimate the position distribution function $p(X|T_k)$ for T_k . Here we model $p(X|T_k)$ as a Gaussian function. The mean vector and covariance matrix of $p(X|T_k)$ are estimated based on (56) and (57).

$$\mu^k = \left(\sum_{j=0}^{R_s-1} W_{k,j} X_{k,j} \right) / \left(\sum_{j=0}^{R_s-1} W_{k,j} \right) \quad (56)$$

$$\mathbf{C}^k = \left(\sum_{j=0}^{R_s-1} W_{k,j} (X_{k,j} - \mu^k)(X_{k,j} - \mu^k)^T \right) / \left(\sum_{j=0}^{R_s-1} W_{k,j} \right) \quad (57)$$

Under the assumption that $p(X|T_k)$ is a Gaussian distribution, the location of T_k is estimated to be μ^k , which is the minimum-variance unbiased estimate of the

location.

5.3.4 Ghost Object

From time to time, ghost clusters may occur in the TDP distribution. Geometrically, the ghost effect happens when the projection of a rectangular pillar at an incorrect location accidentally matches the foreground detection results on the camera views. In Fig. 37, we present an illustration of the ghost problem when trying to reconstruct the 3-D scene based on two camera views. In this case, there are four reconstructed targets while only two of them are true. As a result of the limited camera views, two extra ghost objects occur even based on perfect 2-D silhouettes.

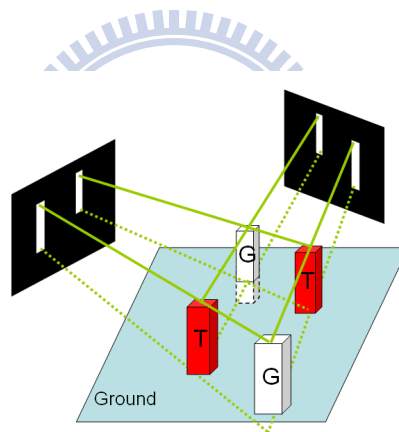
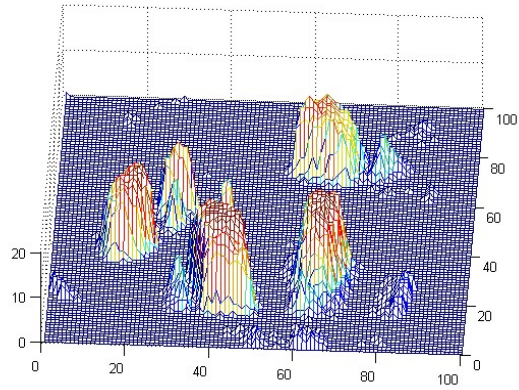


Fig. 37. An illustration of the ghost problem when trying to reconstruct a 3-D scene based on two camera views.

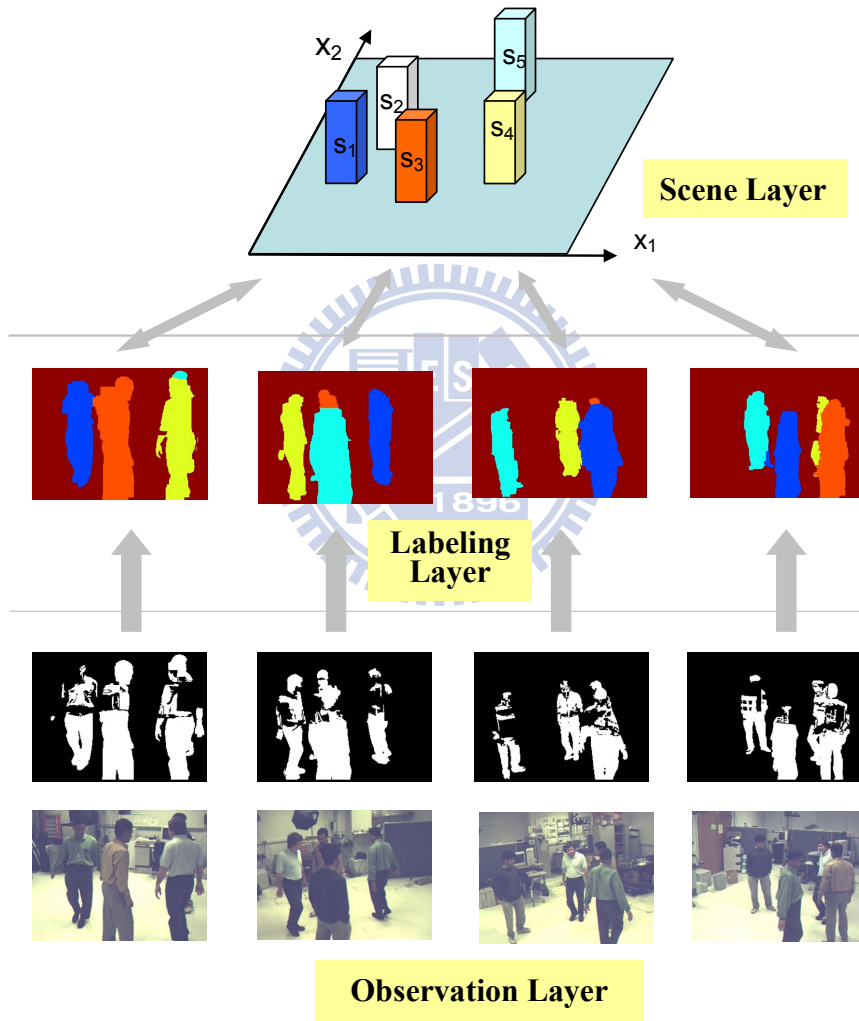
5.4 Bayesian Inference and Ghost Suppression

After information summarization, we have identified a few candidate targets and their possible locations. For each candidate, we have to decide its status to be either a true target or a ghost target. However, owing to the inter-occlusion among candidate targets, the status of a candidate target may actually affect the inference of other candidates. Hence, in our approach, the statuses of all candidate targets are to be inferred simultaneously, rather than being decided individually.

To determine the status of candidate targets, we consider not only the foreground observations and geometric consistence but also some helpful prior knowledge about the targets. For example, as illustrated in Fig. 37, in the perspective back-projection from the 2-D camera view to the 3-D space, the farther the candidate target is away from the camera, the larger the reconstructed object would be. Since the 3-D size of a walking person actually distributes over a specific range, the prior information of human height may offer useful information to exclude targets with unreasonable height.



(a)



(b)

Fig. 38. (a) An example of TDP distribution fused from four camera views.
 (b) The corresponding Bayesian hierarchical framework.

5.4.1 System Modeling

5.4.1.1 Bayesian Hierarchical Framework

In this system, we adopt the BHF framework to simultaneously infer the status of candidate targets. In Fig. 38, without loss of generality, we consider an example of TDP distribution fused from four camera views. The top layer of the BHF architecture is the scene layer S_L that indicates the 3-D scene knowledge built at the fusion stage. Here, we treat the scene model as a knowledge pool collecting message from all cameras. The bottom layer is the observation layer I_L , which contains both the captured images and the corresponding foreground detection results. We define $I_i(m,n)$ and $F_i(m,n)$ as the captured image and the foreground detection result of the i th camera view, respectively. The value of $F_i(m,n)$ is defined as in (50). Between the scene layer and the observation layer, a labeling layer H_L is added to deal with image labeling, target correspondence, and ghost removal. Here, we define $L_i(m,n)$ as the labeling image of the i th camera view.

5.4.1.2 Problem Formulation

In the “five candidate targets” case in Fig. 39, the scene layer $S_L = \{s_1, s_2, s_3, s_4, s_5\}$ corresponds to the status of five candidate targets, with each status node being either “true” (1) or “ghost” (0). With five candidate targets, we have 2^5 status combinations in total. For each combination, we generate the expected foreground occlusion pattern by approximating each “true” target as a rectangle pillar on the ground. By projecting the 3-D rectangle pillars onto each camera view, we form the expected foreground image. Ideally, the optimal status combination would lead to the best match between the expected foreground image and the detected foreground image. In Fig. 39, we show two status combinations based on the example in Fig. 38. In Fig. 39(a), the

scene layer with five candidate targets, together with two of the four camera views, is shown for reference. In Fig. 39(b), we show the combination $\{s_1, s_2, s_3, s_4, s_5\} = \{1,0,1,1,1\}$, which assumes the second candidate is a ghost while the remaining are true. By projecting the four 3-D pillars onto the camera views, we compare the expected foreground image with the detected foreground image. In Fig. 39(c), we show another combination $\{1,1,1,1,1\}$, which assumes all candidates are true targets. By checking the projected foreground images, it appears that the latter combination is less likely than the former combination.

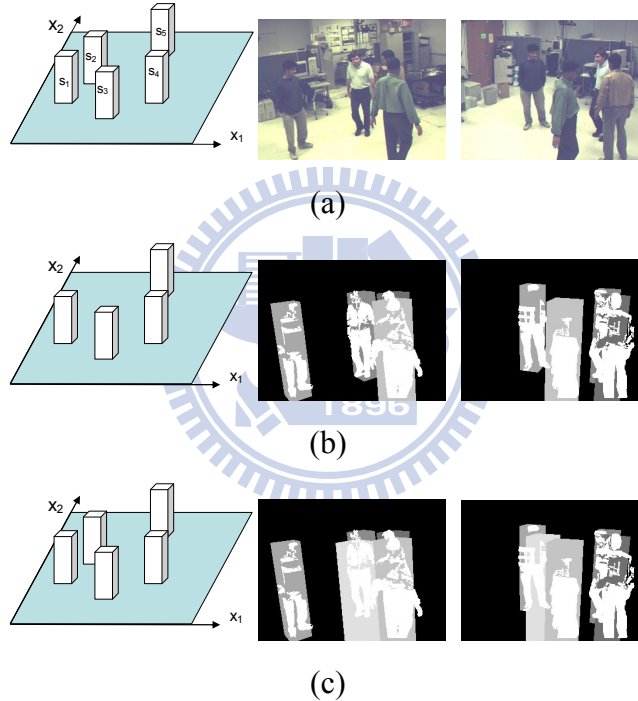


Fig. 39. (a) The scene layer in Figure 36 and two of the four camera views. (b) The combination $\{s_1, s_2, s_3, s_4, s_5\} = \{1,0,1,1,1\}$ and the expected foreground images overlaid with the detected foreground images. (c) The combination $\{1,1,1,1,1\}$ and the expected foreground images overlaid with the detected foreground images.

Assume there are N camera views and we have identified M candidate targets based on the fused TDP distribution. In our system, targets correspondence and image labeling are achieved by assigning a suitable ID from the set $\{T_0, T_1, \dots, T_M\}$ to each pixel of the N labeling images. Note that T_k is the ID of the k th target and T_0 represents the “background” object. Labeling and ghost suppression is achieved by

searching the optimal status combination that fits the foreground detection results. Here, we denote the observation layer as $I_L = (I, F)$, where I indicates the set of N original images and F indicates the set of N foreground detection images. Moreover, we denote the labeling layer H_L as the set of N labeling images, and the scene layer S_L as a status combination. With those definitions, we may combine the target labeling problem and the ghost suppression problem into a single MAP (Maximum A Posteriori) problem as introduced in Section 3.3. In this MAP problem, given the observation $I_L = (I, F)$, we seek the optimal status combination S_L^* and the optimal target labeling H_L^* such that,

$$\begin{aligned}
H_L^*, S_L^* &= \arg \max_{H_L, S_L} \ln p(H_L, S_L | I_L) \\
&= \arg \max_{H_L, S_L} [\ln p(I_L | H_L) + \ln P(H_L | S_L) + \ln p(S_L)] \quad . \quad (58) \\
&= \arg \max_{H, S} [\ln p(I, F | H_L) + \ln P(H_L | S_L) + \ln p(S_L)]
\end{aligned}$$

In (58), $\ln[p(I, F | H_L)]$ describes the relation between the labeling images and the observation data, $\ln[p(H_L | S_L)]$ describes the relation between the 3-D scene model and the 2-D labeling images, and $\ln[p(S_L)]$ describes the prior information about the 3-D scene model.

5.4.1.3 Learning of $p(I, F | H_L)$

As illustrated in Section 3.3, $p(I_L | H_L)$ is composed of a ‘‘classification energy’’ $E_D[I_L(m, n), H_L(m, n)]$ and an ‘‘adjacency energy’’ $E_A[I_L(m, n), H_L(m, n); N_p]$. Hence, we formulate $p(I, F | H_L)$ as

$$p(I, F | H_L) = K \cdot \prod_i \prod_m \prod_n \exp(-E_D[F_i(m, n), H_i(m, n)]) \exp(-E_A[I_i(m, n), H_i(m, n); N_p]). \quad (59)$$

In (59), $E_D[F_i(m, n), H_i(m, n)]$ denotes the ‘‘classification energy’’ that relates the i th foreground detection image with the i th labeling image; $E_A[I_i(m, n), H_i(m, n); N_p]$ denotes the ‘‘adjacency energy’’ that relates the i th original image with the i th labeling

image by checking the adjacent property within the neighborhood N_p ; and K is a normalization term.

Ideally, if the foreground detection results are perfect, we expect $H_i(m,n)$ to be T_0 if $F_i(m,n)$ is 0, and to be an element of $\{T_1, T_2, \dots, T_M\}$ if $F_i(m,n)$ is 1. Once a labeling violates this expectation, an empirically selected constant α is added onto the detection energy to penalize this inference. Hence, we define $E_D[F_i(m,n), H_i(m,n)]$ as

$$E_D(F_i(m,n), H_i(m,n)) \equiv \alpha \times \{1 - \delta[F_i(m,n), T(H_i(m,n))]\} \quad (60)$$

with $T(H_i(m,n))$ being defined as

$$T(H_i(m,n)) = \begin{cases} 0 & \text{if } H_i(m,n) = T_0 \\ 1 & \text{otherwise} \end{cases} \quad (61)$$

and $\delta[p_a, q_a]$ being defined as

$$\delta[p_a, q_a] = \begin{cases} 1 & \text{if } p_a = q_a \\ 0 & \text{otherwise} \end{cases} \quad (62)$$

On the other hand, the local decisions of two adjacent labeling nodes are usually highly correlated, especially when their corresponding image pixels share similar color features. Hence, we define the adjacency energy $E_A[I_i(m,n), H_i(m,n); N_p]$ by using the same smooth constraint presented in Section 3.3. Here, we briefly explain the design of the adjacency energy model again.

In our system, the adjacency energy $E_A[I_i(m,n), H_i(m,n); N_p]$ is defined as.

$$E_A[I_i(m,n), H_i(m,n); N_p] \equiv \beta \times \sum_{\Delta m=-p}^p \sum_{\Delta n=-p}^p C_A[I_i, H_i, m, n, \Delta m, \Delta n]$$

With this definition, if two neighboring sites are set to different labels, our system will give a larger penalty if we find the color difference between two sites is small. Otherwise, our system will give a smaller penalty. That is, two neighboring sites tend to share the same label when the difference between their color features is small, and tend to have different labels otherwise.

5.4.1.4 Learning of $p(H_L|S_L)$

Given a status combination S_L , we define a conditional probability $p(H_i(m,n)=T_k|S_L)$ to express the likelihood of having a label T_k at the pixel (m,n) of the i th labeling image. Here, we construct the probability model in a Monte Carlo manner. With the status combination S , we define a few rectangular pillars on the ground. The height and width of each pillar are sampled based on the probability density functions $p(H)$ and $p(R)$. The locations of the pillars are sampled from $p(X|T_k)$, where T_k indicates the k th target. With the camera projection parameters, the expected foreground patterns for each target can be generated by projecting these rectangular pillars onto each camera view. Occasionally, more than two targets may project onto the same image region and cause occlusion. The inter-occluded patterns can be determined by checking the distance from the camera to the mean location of the targets. In Fig. 40, we demonstrate the occlusion effect by plotting $p(H_i(m,n)=T_k|S_L)$ individually for each of the four targets in Fig. 38 (b).

Based on the definition of $p(H_i(m,n)=T_k|S_L)$, we have

$$p(H_L | S_L) \equiv \prod_i \prod_m \prod_n p(H_i(m,n) | S_L) \quad (63)$$

and we define the log probability function $\ln[p(H_L|S_L)]$ as

$$\ln p(H_L | S_L) = \sum_i \sum_m \sum_n \ln p(H_i(m,n) | S_L). \quad (64)$$

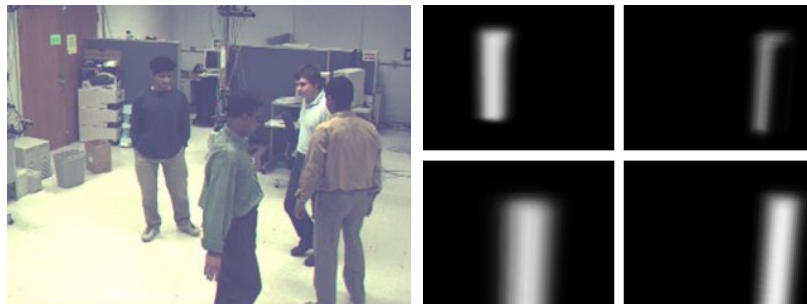


Fig. 40. Examples of $p(H_i(m,n) = T_k|S)$

On the other hand, the prior knowledge $p(S_L)$ is also used in the determination of

the optimal status combination. In our system, if M_t true targets are identified at the previous time instant, we assume it is more likely to have a similar number of true targets at the current moment. That is, if we denote S_o^{t-1} as the optimal status combination at the previous time instant ($t-1$) and S^t as a status combination at the current time instant t , we define the prior probability of S^t as

$$p(S^t) = \begin{cases} W_1, & \text{if } |N(S^t) - N(S_o^{t-1})| \leq 1 \\ W_2, & \text{otherwise} \end{cases}, \quad (65)$$

where W_1 and W_2 are two constants with $W_1 \geq W_2$. In (65), $N(S_L)$ denotes the number of true targets in the status combination S_L . In detail, if we know the ratio between W_1 and W_2 , we could determine W_2 such that the probability summation equals to 1. For example, we assume $W_1 = 2W_2$, the number of candidate targets at Time t is 5, and the number of true targets in the previous optimal combination S_o^{t-1} is 4. For this case, we have $2W_2 \cdot (C_3^5 + C_4^5 + C_5^5) + W_2 \cdot (C_0^5 + C_1^5 + C_2^5) = 1$. Hence, we choose $W_2 = 1/48$ and $W_1 = 1/24$.

5.4.2 Multi-Target Labeling and Tracking

5.4.2.1 Optimal Inference of Target Labeling

With the above deduction, the labeling of targets and the suppression of ghost targets can be solved by finding the optimal labeling images (H_L^*) and the optimal status combination (S_L^*) that maximize the following potential function $C_p(H_L, S_L)$:

$$\begin{aligned}
H_L^*, S_L^* &= \arg \max_{H_L, S_L} C_p(H_L, S_L) \\
&= \arg \max_{H_L, S_L} \left\{ - \sum_i \sum_m \sum_n E_D[F_i(m, n), H_i(m, n)] \right. \\
&\quad - \sum_i \sum_m \sum_n E_A[I_i(m, n), H_i(m, n); N_p] \\
&\quad \left. + \sum_i \sum_m \sum_n \ln p(H_i(m, n) | S_L) + \ln p(S_L) \right\} . \tag{66}
\end{aligned}$$

Basically, the problem of target labeling and ghost suppression is treated as a maximum a posterior (MAP) problem from the viewpoint of Bayesian generative model. Here, we incorporate four constraint terms: classification energy E_D , adjacency energy E_A , likelihood function $p(H_L|S_L)$, and prior probability $p(S_L)$. As illustrated in Fig. 38, the classification energy $E_D[F_i(m, n), H_i(m, n)]$ represents the bottom-up constraint between the foreground detection images and the labeling images. To model the interaction between the labeling layer and the scene layer, the likelihood function $p(H_L|S_L)$ represents the expected labeling layout based on the status combination S_L . The expected inter-occluded patterns among candidate targets are also modeled in $p(H_L|S_L)$ to influence the classification of local labeling nodes. By introducing the adjacency energy $E_A[I_i(m, n), H_i(m, n); N_p]$, the proposed framework can not only infer the labeling based on the fusion of scene knowledge and foreground detection results, but also refine the labeling results based on the original image data. Last, the prior probability $p(S_L)$ includes the temporal prediction based on the previous decision.

Moreover, due to the inter-occlusion among targets, the status inference of a candidate target may depend on some other candidate targets. Hence, we need to take into account relevant candidate targets when we infer the status of a candidate target. A brute-force way is to evaluate all possible status combination and pick the optimal one as S_L^* . However, this leads to exponentially growing computational complexity as the number of candidate targets increases. Fortunately, in general, there could be

some kind of separateness among candidate targets that can be used to reduce the number of status hypotheses. In our system, if the projection of a candidate target on a camera view does not overlap with the projection of other targets, that candidate target is thought to be a true target. By excluding those targets with isolated projections, we only need to check the status combinations of the remaining targets. For example, in Fig. 38, the target S_5 corresponds to the left target in the third camera view. Since this target has an isolated projection in the third camera view, it is treated as a true target. For this case, we only generate 2^4 status combinations for S_1, S_2, S_3 and S_4 , instead of generating 2^5 combinations for all five targets.

In principle, the best configuration of labels depends on image data, foreground detection result, and scene model. In our experiments, even though plentiful false alarms and false rejection may appear in the foreground detection results, these errors have little influence on the final inference result. Based on the proposed BHF, the inter-occlusion problem can be effectively analyzed, the connected foreground regions can be well separated, and the ghost targets can be correctly identified.

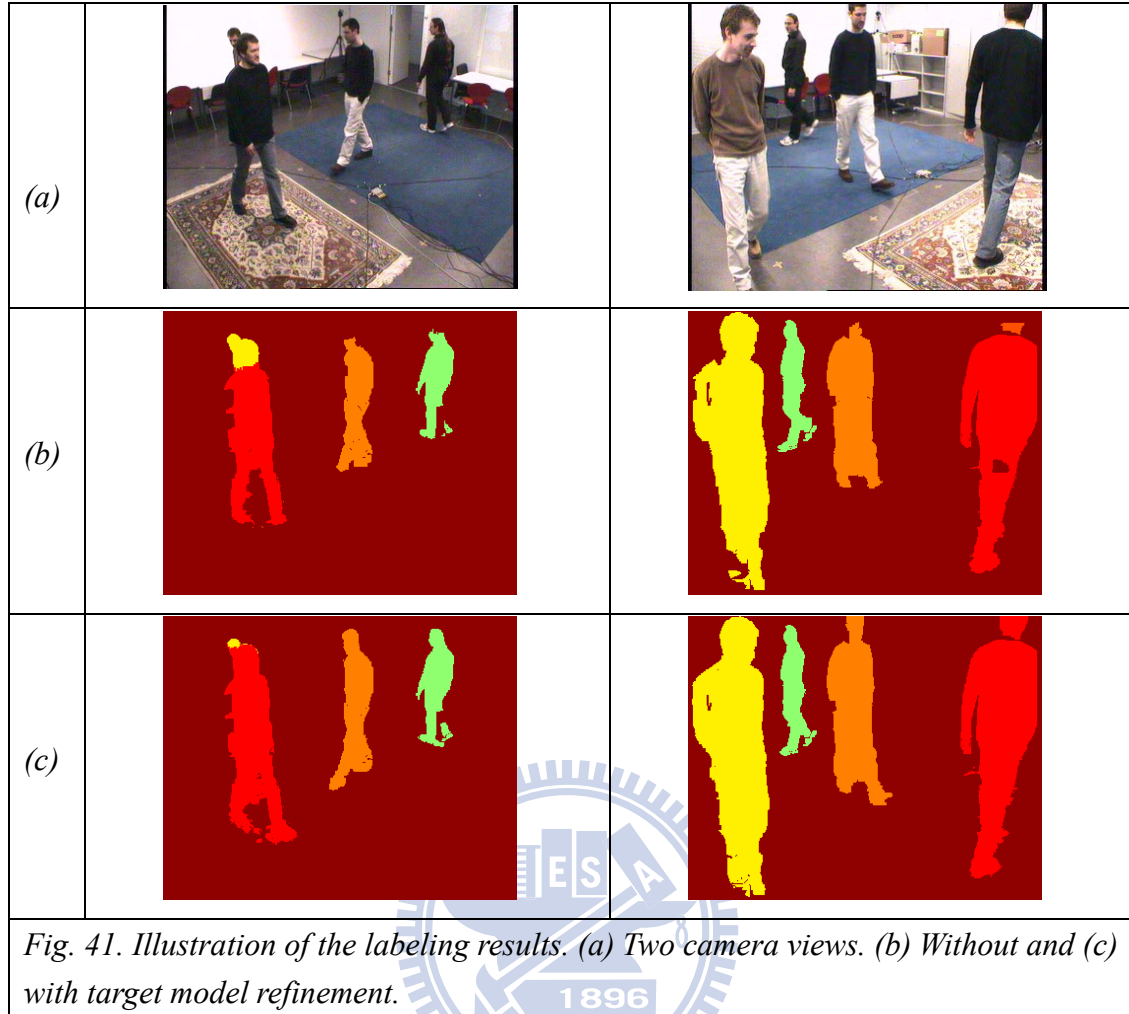
5.4.2.2 3-D Target Model Refinement

Usually, the moving targets in the surveillance zone may have different model parameters, such as the target height and width. If the personalized target models can be obtained, the performance of the proposed inference framework can be further boosted. In real situations, however, it is impractical to obtain the personalized 3-D model parameters in advance. Hence, in our system, we achieve personalized 3-D modeling by treating the model parameters as latent random variables and introduce an EM based algorithm to iteratively refine the model parameters. The basic idea is to update the 3-D model parameters in the Expectation step based on the labeling results derived from the optimization procedure in (66). Next, in the Maximization step, by

using (54) to consider the refined statistics of the 3-D model parameters in an expectation sense, the optimization procedure in (66) is re-executed to boost the inference performance. The operation is repeated until the updated parameters converge or the maximum iteration number is met.

In Fig. 41, we show an example of the labeling results with and without the target model refinement. Since each target has obvious height difference, the labeling results with a unified target model generate wrong labeling around the head regions as shown in Fig. 41(b). After the refinement of target model, more accurate labeling results are achieved, as shown in Fig. 41(c).

In our system, the major 3-D target model of each target is a pillar model standing at a location X on the ground plane, with parameters height (H) and width (R). Initially, the proposed EM algorithm uses the pre-trained probability distributions $p(H)$ and $p(R)$ to model the uncertainty of each target height and width. With this initial setting, the proposed BHF generates the optimal inference of target labeling. Since the BHF combines not only the 3-D scene priors and target priors but also the observed image data and the corresponding foreground detection results, the optimal target labeling actually reveals the personal property of each detected target. Hence, based on the labeling results in multiple image views, we further update the probability distributions of H and R to establish personalized probability models. In practice, we found the target width has less uncertainty among targets and the pre-trained probability $p(R)$ can well model the uncertainty in target width. Hence, in our system, only the model of target height is recursively refined.



In the Expectation step of the proposed EM procedure, the main goal is to refine the posterior probability of each target height given the multi-view labeling results. In our system, based on the Bayesian rule, the refinement of the posterior probability is defined as follows

$$p(H_k^r | L^r) \equiv C \cdot p(L^r | H_k^r) \cdot p(H_k^r). \quad (67)$$

where

$$p(H_k^r) = \begin{cases} p(H) & \text{if } r = 1 \\ p(H_k^{r-1} | L^{r-1}) & \text{otherwise} \end{cases}.$$

In (67), L^r indicates the labeling results of multiple image views at the r th Iteration of

our EM procedure. H_k^r is the height of the k th target at Iteration r , C is a normalization constant, $p(L^r | H_k^r)$ is the likelihood term which will be defined later, and $p(H_k^r)$ is the prior term of H_k^r . In our system, we directly treat $p(H_k^{r-1} | L^{r-1})$ as the prior information propagated from the previous iteration to the current iteration to set the prior $p(H_k^r)$. Initially, $p(H_k^1)$ is set to be the pre-trained target height probability $p(H)$.

To formulate the likelihood term $p(L^r | H_k^r)$, we project the pillar model at the ground position of the k th target, with height H_k^r and width R_k , onto multiple camera views and we verify the projected regions with the labeling results. Since the variables H and R are assumed to be statistically independent, we assign the width of all targets to be the mean value of $p(R)$ during the computation of $p(L^r | H_k^r)$. Ideally, if a more precise target height is chosen, the projected region will better fit the labeling result. Hence, we define the likelihood term as

$$p(L^r | H_k^r) = \left(\prod_i \prod_{m,n \in A_i^k} (p_{m,n}^i(l)) \right)^{1/N}. \quad (68)$$

In (68), A_i^k is the projected region of the k th target in the i th camera view. $p_{m,n}^i(l)$ is the probability of the labeling pixel at (m,n) with the label ID “ l ”. N is the total number of pixels within the projected regions. Since different H_k^r may generate different projected regions, we use the function $(\cdot)^{1/N}$ for normalization. Moreover, we assume the statuses of different labeling pixels are independent of each other and we evaluate only those pixels inside the projected regions of the k th target. In principle, the label ID “ l ” tends to be T_k . Hence, $p_{m,n}^i(l)$ has a higher probability if

“ L ” equals to T_k and has a lower probability if “ L ” equals to T_0 . Occasionally, owing to occlusion, “ L ” may equal to some foreground target other than T_k . In this case, we do not have the information about T_k and we assign $p_{m,n}^i(l)$ to be an intermediate value.

In summary, we define $p_{m,n}^i(l)$ as

$$p_{m,n}^i(l) = \begin{cases} \lambda \cdot e^x & \text{if } l = T_k \\ \lambda \cdot e^y & \text{if } l = T_0 \\ \lambda \cdot e^z & \text{otherwise} \end{cases}, \quad (69)$$

where λ is a normalization term to make the probability summation equal to 1. Moreover, x , y , and z are empirically pre-selected parameters, with $x > z > y$. If we rewrite (68) based on (66), we get a likelihood form as below

$$p(L^r | H_k^r) = \lambda \cdot \exp\left\{\frac{1}{N}(x \cdot N_k + y \cdot N_0 + z \cdot N_{other})\right\}, \quad (70)$$

where N_k , N_0 , and N_{other} are the number of T_k -labeled pixels, the number of T_0 -labeled pixels, and the number of other pixels inside the projected regions in all camera views. Basically, (70) simply measures the matching level by accumulating the weighted sum of different labeling pixels inside the projected regions with the weighting parameters (x,y,z) . Once the likelihood term $p(L^r | H_k^r)$ is determined, the refined probability distribution of the k th target height at the current iteration can be obtained based on (67). The refined model $p(H_k^r | L^r)$ is fed back to the proposed BHF to find the optimal object labeling again. In our experiments, 2~3 iterations are enough for the convergence of the EM algorithm.

5.4.2.3 Multi-target Tracking

In our system, by associating the temporal succession, we also extend the detection results to perform 3-D tracking over the ground plane. Basically, the object

tracking is treated as a dynamic system problem. Based on the proposed Bayesian detection framework, the major observation of the dynamic system comes from the estimated target location on the ground plane. In principle, to deal with the dynamic system problem, several Bayesian filter techniques can be used. For instance, we can use a Monte Carlo based framework to track multiple targets on the ground plane, as proposed in [92]. However, for the sake of computational simplicity, we adopt the Kalman filter to track each target in the scene.

5.5 Results and Discussion

5.5.1 Experimental Datasets

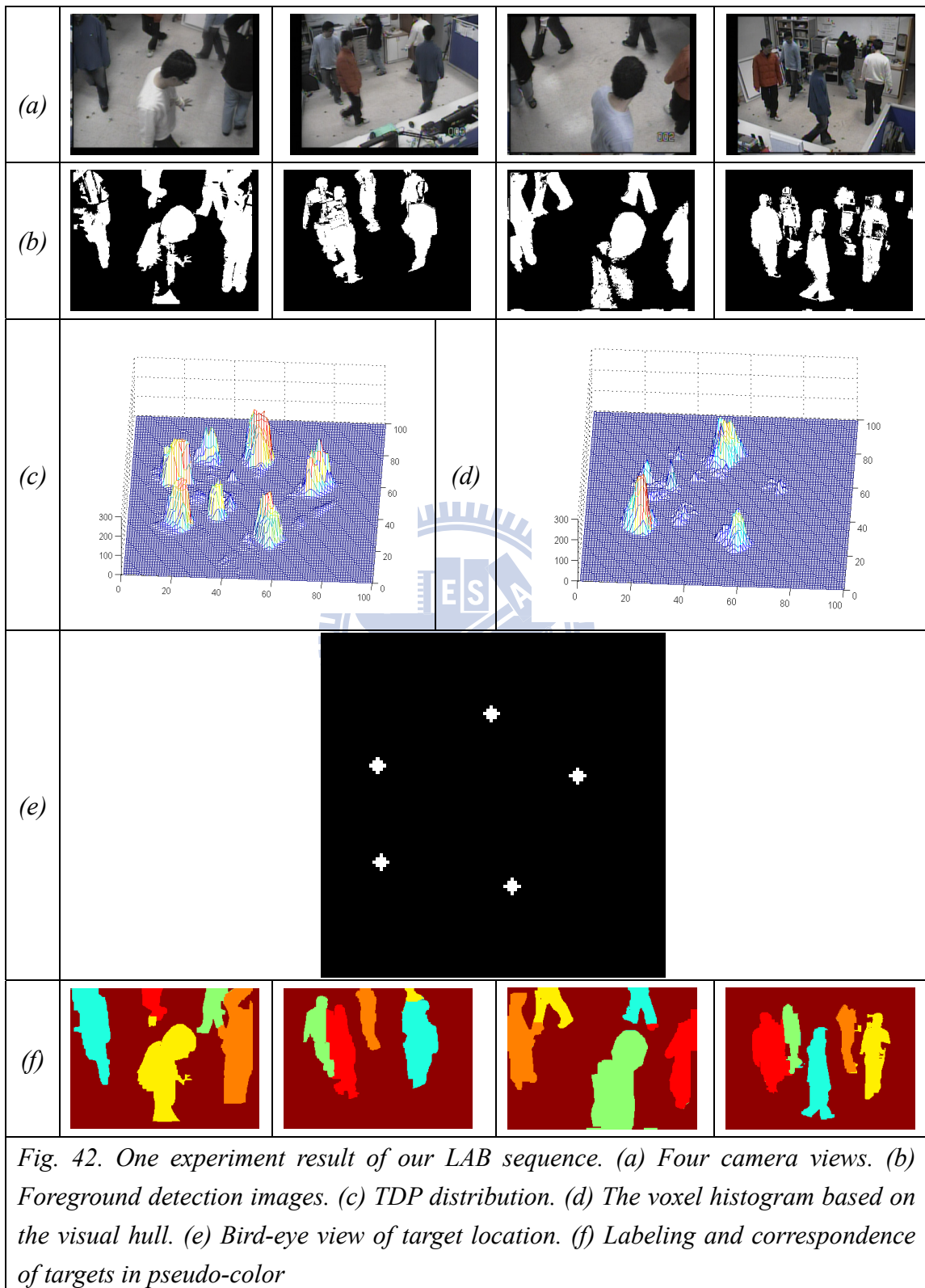
To test our system over real video sequences, we set up four static cameras in our lab to capture test sequences. In our sequences, the coverage is about 4.5m by 4.5m, with 3 to 5 moving targets within the zone. A set of snap shots with 5 persons inside the scene are shown in Fig. 42(a). On the other hand, we also tested our system over the video sequences provided by the M2Tracker project [87] and the dataset used in Fleuret's papers [89][90][91]. The M2Tracker sequence was captured by 15 synchronized cameras over a 3.0m by 3.0m area, while Fleuret's sequence was captured by 4 synchronized cameras in a 12.88m² room. For each sequence, four camera views are used to evaluate our system. If more camera views are used, the performance of our system can be further boosted. In Figure 11(a) and Figure 12(a), we show four snap shots obtained from each of these two sequences.

For each sequence, the cameras have been geometrically calibrated with respect to a world coordinate system. Except the M2Tracker sequence, each video sequence contains more than 300 frames. Especially, Fleuret's video sequence contains 3900 frames with many interesting events, such as people moving into surveillance zone,

people approaching and occluding each other, and some people only monitored by a portion of the cameras. For the evaluation of object ground location, we acquired the ground truth of the M2Tracker sequence from Dr. Li [96]. To establish the ground truth of Fleuret’s sequence, we manually identified the image position of human necks and built the correspondence among camera images. By backprojecting the corresponding image points, the object locations on the 3-D ground plane were obtained. For this sequence, we established the ground truth for every 25 frames. To see the details of our experimental results, please visit our website [100].

5.5.2 Foreground Detection and Information Fusion

For each video sequence, foreground blobs are detected based on the popular GMM (Gaussian Mixture Model) background subtraction algorithm [101]. Shadow removal [102] is also included to suppress false detection. In Fig. 42(b), Fig. 43(b), and Fig. 44(b), we show the detected foreground images, where plentiful false detections occur due to the appearance similarity between the foreground objects and the background environment. In Fig. 42(c)(d), Fig. 43(c)(d), and Fig. 44(c)(d), we compare the fusion results based on the proposed model-based method and the conventional data-driven method. It can be seen that the model-based approach generates more reasonable fusion results, especially for the person in white shirt in Fig. 42 whose bottom part and upper part cannot be observed in the first and the third camera views, respectively.



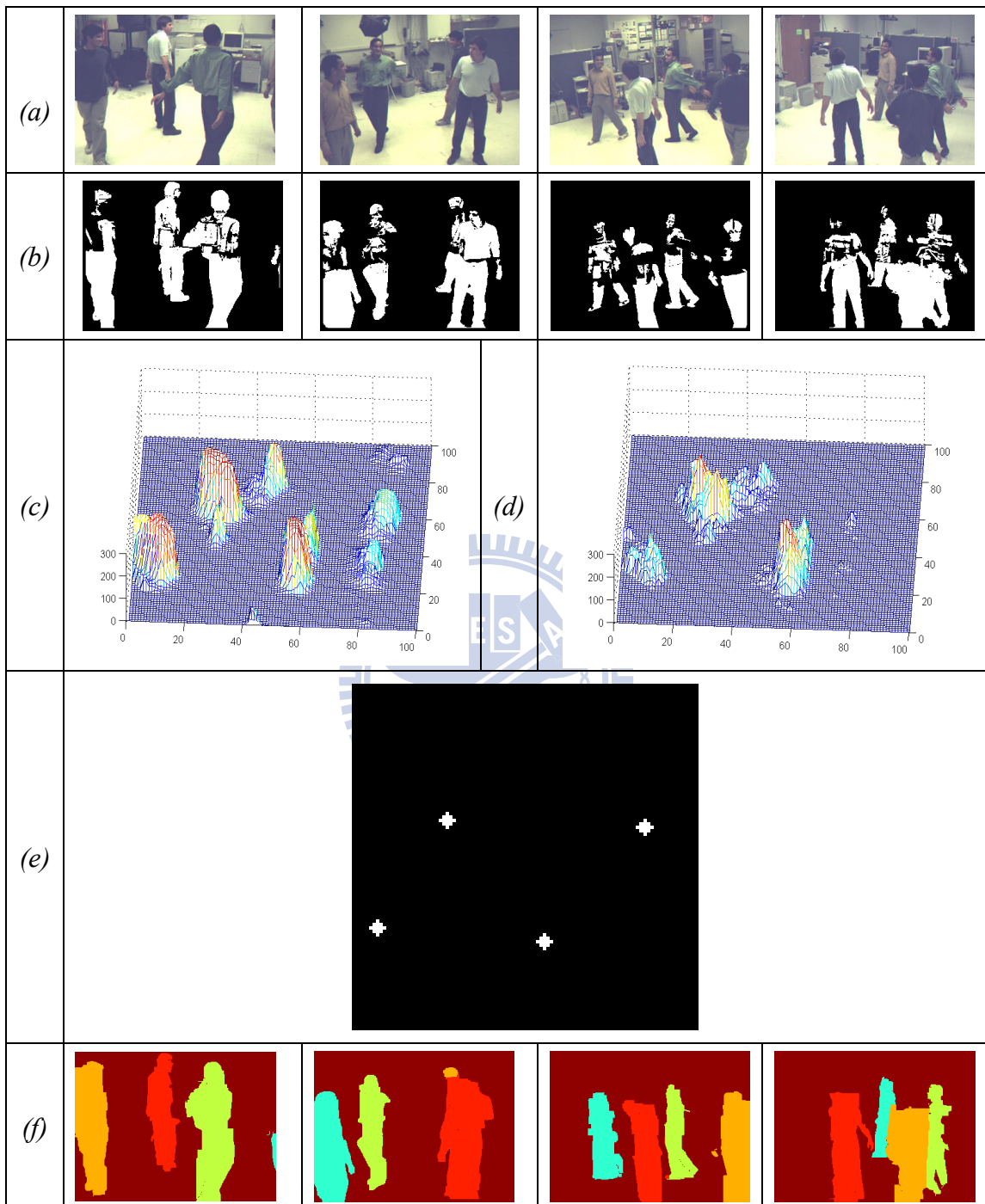
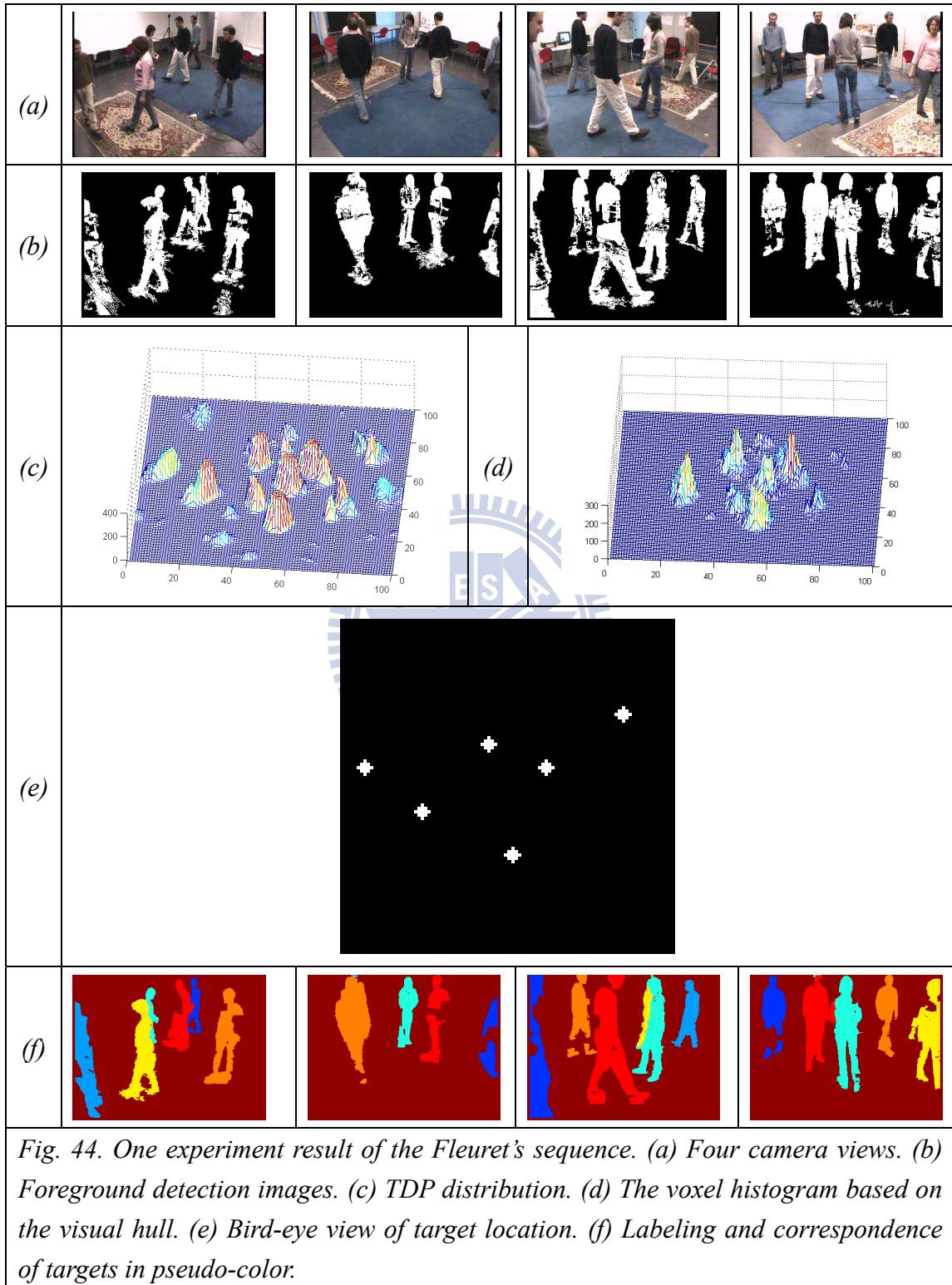
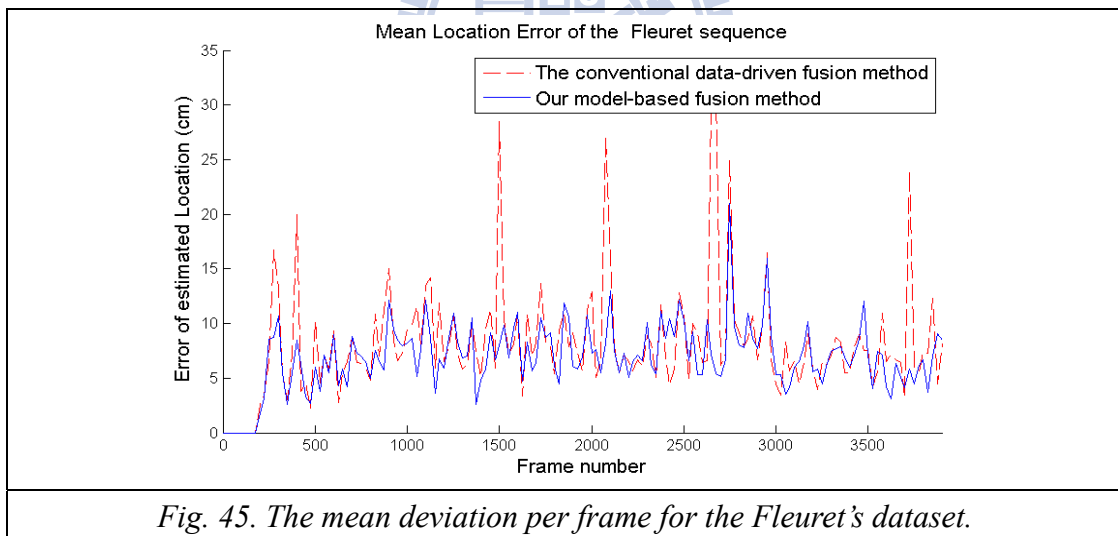


Fig. 43. One experiment result of the M2Tracker sequence. (a) Four camera views. (b) Foreground detection images. (c) TDP distribution. (d) The voxel histogram based on the visual hull. (e) Bird-eye view of target location. (f) Labeling and correspondence of targets in pseudo-color.



5.5.3 Accuracy of Target Location

In our experiments, the object locations on the ground plane were estimated and displayed on the bird-eye view image, as shown in Fig. 42(e), Fig. 43(e), and Fig. 44(e). To evaluate the performance of our system, we calculate the deviation of the estimated location with respect to the ground truth. First, we compare the performance between the model-based fusion method and the conventional data-driven fusion method by measuring the mean location deviation per frame for Fleuret’s video sequence. In this comparison, we use the same inference process for the estimation of target location. The profiles of the mean location deviation are plotted in Fig. 45. Numerically, the averaged mean deviations over Fleuret’s sequence are 0.087m and 0.073m respectively for the data-driven method and our model-based method.



From time to time, some targets in the scene may not be monitored by all cameras. An example is shown in Fig. 46(a), where the person in blue jean can only be observed by two of the four cameras. For this case, the corresponding cluster in the TDP distribution is smaller but still detectable, as shown at the upper-right corner of the distribution in Fig. 46(b). Some ghost clusters also exist in this TDP distribution.

With the mean-shift algorithm for clustering and the BHF inference for ghost removal, all four targets are successfully detected, as shown in Fig. 46(c). In this figure, we use different gray levels to indicate different surveillance zones. From bright-gray to dark-gray, they are the 4-camera zone, 3-camera zone, and 2-camera zone. We also numerically evaluate the location accuracy of our method inside each of these three zones. For Fleuret's sequence, 76% of the moving targets are monitored by four cameras, 21% of the moving targets are monitored by three cameras, and 3% of the moving targets are monitored by two cameras. In Table 2, we list the accuracy of target location in these three zones. We may find the accuracy goes down when the number of cameras decreases.

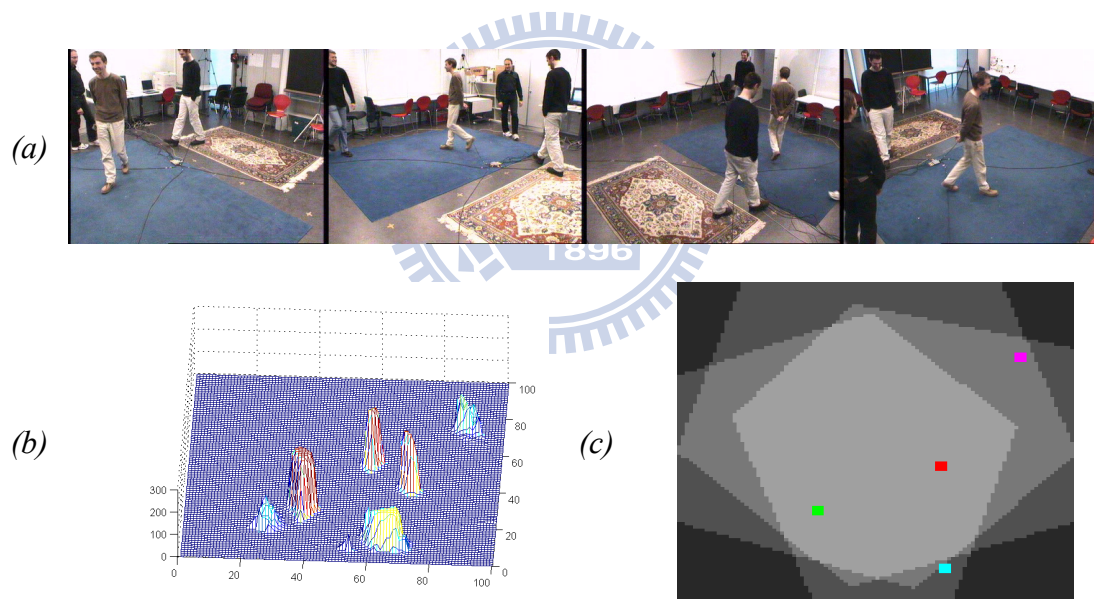


Fig. 46. One example of extended surveillance zone. (a) Four camera views. (b) The TDP distribution. (c) Bird-eye view of target location.

Table 2. Accuracy of target location in three difference zones for Fleuret's sequence.

Surveillance Zone	4-camera Zone	3-camera Zone	2-camera Zone
Mean deviation	0.069 m	0.079 m	0.147 m
Max deviation	0.178 m	0.257 m	0.391 m

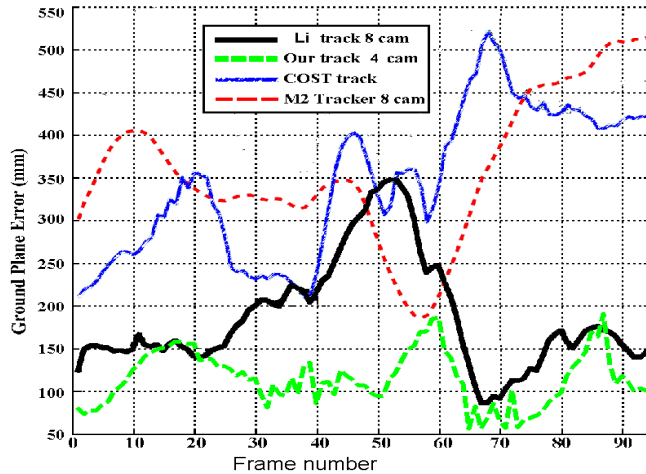


Fig. 47. A comparison of the mean deviation of each frame over the M2Tracker dataset.

Moreover, we adopt the widely-used M2Tracker sequence as the benchmark to compare the accuracy of target location. In detail, we calculate the deviation of the estimated locations based on the M2Tracker sequence, for which the experimental results of a few other systems are available. For the M2Tracker sequence, the averaged mean deviation of our system is about 0.108m. In Fig. 47, we compare the mean deviation of each frame over four different systems: M2Tracker [87], Cost track [103], Li’s algorithm [96], and ours. Please note that only four camera views are used in our system, rather than the eight camera views used in the other three methods

5.5.4 Detection and Labeling with Ghost Removal

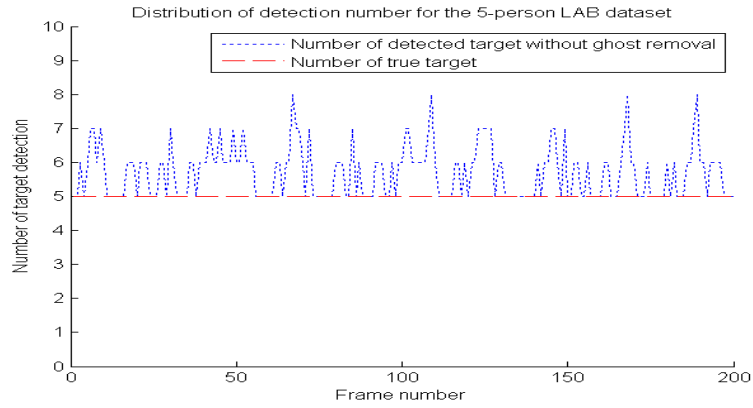
As shown in Fig. 42, Fig. 43, and Fig. 44, the computed TDP distribution reveals distinguishable clusters for candidate target identification and localization. The number and the location of the candidate targets can be decided by mean-shift clustering. With the presence of ghost objects, the number of candidate targets is usually larger than the true target number. After the inference stage, the results of ghost suppression, labeling, and correspondence are presented in Part (f) of Fig. 42,

Fig. 43, and Fig. 44. The results demonstrated that the scene knowledge is very helpful in the labeling process even under severe inter-target occlusion, especially for those connected foreground regions. We may also find that ghost targets are correctly removed under the proposed BHF framework.

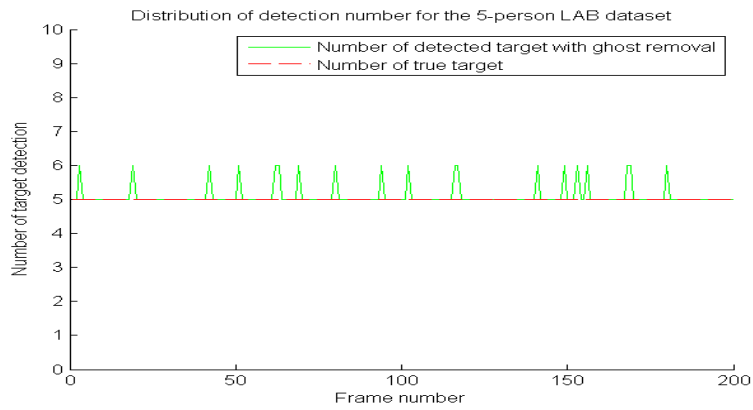
To quantitatively evaluate the detection and correspondence performance, false positive rate (FPR) and false negative rate (FNR) are used. In our system, the target detection and correspondence are defined as “correct” when the projected regions of the detected target in all camera views intersect the same individual. Based on this definition, the FPR and FNR of all tested datasets are calculated and listed in Table 3. Here, the performance before and after ghost removal are provided for comparison. The results depict the FPR before ghost suppression is higher, while the FNR is very low for all test sequences. After applying the BHF to detect and remove ghost targets, the ghost effect is suppressed and the FPR is decreased. Moreover, if we compare with Fleuret’s results [91], whose FPR and FNR are 0.0399 and 0.0614 respectively, our method achieves even lower FPR and FNR with values 0.021 and 0.013 for the same dataset. On the other hand, for the Lab dataset, we show in Fig. 48 the number of detected targets at each time frame. With ghost removal, the identified target number is much closer to the true target number

Table 3. False positive rate (FPR), false negative rate (FNR).

Video datasets	Without ghost removal		With ghost removal	
	FPR	FNR	FPR	FNR
OVVV 3 persons	0.033	0.000	0.000	0.000
OVVV 4 persons	0.023	0.000	0.000	0.000
OVVV 5 persons	0.040	0.000	0.000	0.000
Lab 3 persons	0.053	0.000	0.003	0.001
Lab 4 persons	0.045	0.000	0.010	0.003
Lab 5 persons	0.042	0.000	0.017	0.000
M2tracker	0.183	0.000	0.027	0.000
Fleuret	0.219	0.000	0.021	0.013



(a)

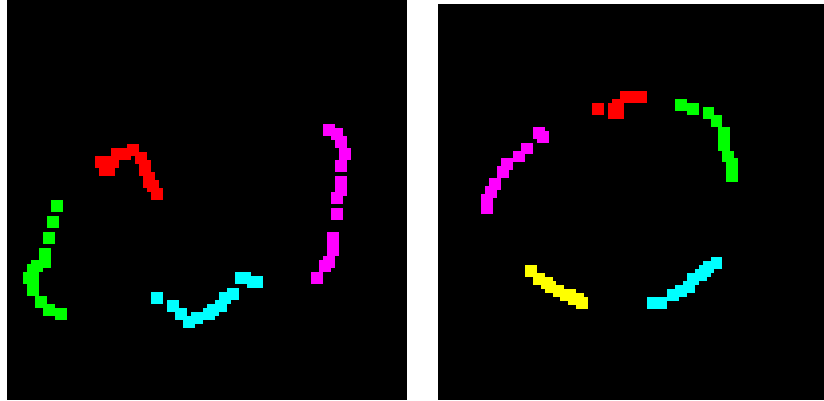


(b)

Fig. 48. The distributions of the number of detected target per frame for the 5-person Lab dataset. (a) Results without ghost removal. (b) Results with ghost removal.

5.5.5 Multi-target Tracking on the Ground Plane

In our system, the multi-target detection results across a few successive frames are associated to establish temporal target tracking. In Fig. 49, we show the bird-eye view of our tracking results for both the M2tracker and Lab datasets. Different colors correspond to different targets. It can be seen that the proposed system can be easily extended to handle the task of multi-target tracking.



(a)

(b)

Fig. 49. Multi-target tracking Results (a) M2tracker dataset (4 person). (b) Lab dataset (5 person).

5.5.6 System Complexity

The whole system is implemented in the Visual C++ environment on a PC with 3.0GHz Core 2 Duo CPU. To evaluate the computational complexity of our system, we analyze the execution time of our system based on the M2Tracker sequence. In Table 4, we list the major processes of our system and the averaged runtime of each process at one time instant with four camera views. It can be seen that the major computations are spent over background subtraction, mean-shift clustering, and graph cut optimization. In practice, the background subtraction process can be executed at the camera side with a client-server surveillance architecture and our algorithm is mainly implemented at the server side for data integration. If excluding the background subtraction process, it takes about 3 to 6 seconds to perform the positioning, labeling, correspondence, 3-D target model refinement, and ghost suppression processes over four image shots with 320×240 resolution. Basically, it takes longer time if there are more candidate targets in the scene. Moreover, if we simplify the inference process to perform 3-D positioning and ghost suppression only, the whole computation time can be shortened down to around 0.2 seconds for every 4-camera image shot.

Table 4. Runtime list

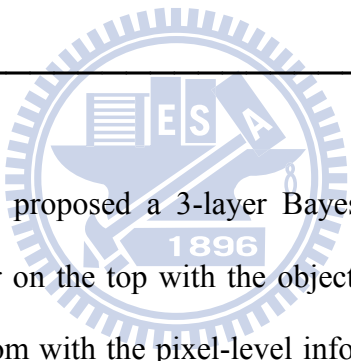
Process	Detailed Operations	Averaged Runtime (sec.)
Foreground Detection (4 camera views)	Background Subtraction	0.25
	Shadow Removal	~ 0.001
Information fusion (4 camera views)	Sample Generation	< 0.00001
	Mean-shift Clustering	0.13
Bayesian Inference (4 camera views)	Hypothesis Generation	<0.00001
	Graph Cuts Optimization	3.75
	Target Model Refinement	0.0002

5.5.7 Future Works

Currently, the proposed system could efficiently determine the number of moving clusters inside the surveillance zone and accurately track the 3-D trajectories of the tracked targets. However, an extra target counting analysis for groups is needed in order to estimate the target number if there are groups inside the surveillance zone. In the future, we plan to utilize the target width as possible prior information to roughly estimate the target number of a group. Also, we attempt to integrate a robust face detection algorithm into current system so that we can have more precise target counting. On the other hand, the face view of a target inside the surveillance zone is also important 3-D scene information for a modern surveillance system. Therefore, we will expand our system with the capability of multi-view multi-face detection in the near future.

CHAPTER 6

Conclusions



In this dissertation, we proposed a 3-layer Bayesian hierarchical framework, which includes a scene layer on the top with the object-level information, an image observation layer at the bottom with the pixel-level information, and a labeling layer in the middle to interconnect these two layers. The proposed framework can efficiently integrate both the top-down information and the bottom-up messages. With the integration in a unified framework, the top-down information and the bottom-up messages cross reference each other to support a more robust and accurate system inference. Moreover, the scene layer offers a systematic representation to depict the 3-D scene model in a parametric fashion. With the parameterized scene model, many troublesome issues, such as shadow effect and occlusion, now become easier to handle. In fact, shadow and occlusion are nature phenomena caused by objects in the 3-D scene. In our approach, the proposed BHF framework models the generation of those scene effects so that shadow and occlusion may even provide useful clues for

scene inference. This BHF framework is designed to simultaneously perform image analysis and scene modeling. By having calibrated the surveillance cameras in advance, the BHF framework builds the physical connection between the 3-D scene and the captured 2-D images. This connection enables scene knowledge and image observation to cross reference each other so that the unknown parameters in the scene model and the labeling of the image contents are inferred simultaneously under a unified framework.

For the application of vacant parking space detection, we adopted the proposed BHF to simultaneously detect vacant parking spaces and interpret the image content through labeling. In practice, the challenges of vacant space detection come from the shadow effect, the occlusion effect, the appearance ambiguity, the perspective distortion, and the dramatic luminance variations. In our system, we explicitly define a scene model of the parking lot. Based on the model, the generation of shadow, the generation of occlusion, the variation of lighting, and the perspective distortion are closely coupled with the status of the parking spaces. By utilizing the proposed BHF framework, the scene generation process is well modeled and the optimal inference of the parking space status is deduced. Our results showed that this system can achieve up to 99% accuracy in vacant parking space detection under different lighting conditions.

In the application of multi-target tracking with ghost suppression over a multi-camera system, the proposed BHF provides an efficient way to simultaneously detect, locate, and label targets across multiple cameras. The ghost effect is also analyzed and suppressed. In principle, the system algorithm consists of two major steps: information fusion and Bayesian inference. The model-based information fusion step collects consistent information from multiple camera views and couples with 3-D priors to establish scene knowledge. Furthermore, the scene knowledge is

treated as extra information and is used in labeling, correspondence, and ghost suppression. The whole process is well modeled and resolved in the Bayesian inference step under the proposed BHF framework. Based on the proposed algorithm, many troublesome issues, like fragmental foreground detection results, inter-target occlusion, ghost targets, and the determination of target number, can be effectively handled in a systematic manner. Moreover, the proposed EM-based mechanism can iteratively refine target models and further boost the system performance. The experimental results show our system can successfully label objects and build correspondence even under severe occlusion. In addition, our system requires neither isolated foreground extraction nor color calibration among cameras.

In summary, in this dissertation, we present a BHF framework for image analysis and 3-D scene modeling. We also apply the BHF framework to two applications of video surveillance. By using the hierarchical framework to represent the image generation model in a probabilistic manner, we have demonstrated how to systematically integrate useful information from pixel-level, region-level, and object-level for a semantic inference of the 3-D environment.

In the future, we plan to expand the proposed BHF so that the temporal information from previous frames could be further utilized. With temporal information, the image constraint, and the scene constraint, the modified BHF would have more flexibility to boost the system performance. In addition, we will apply BHF to other applications such as scene understanding. As most of our knowledge, to achieve better scene understanding, the contextual scene information is proved to be useful. Since the proposed BHF could model contextual scene information in a natural sense, we believe BHF offers a possible solution to scene understanding.

Appendix A: Estimation of Sunlight Direction

Based on the vectors \vec{u} , \vec{s} , and \vec{n} shown in Fig. 24, we define a USN coordinate system and represent the sunlight direction as

$$(-\cos(\delta)\cos(\omega_s(t-t_\theta)), -\cos(\delta)\sin(\omega_s(t-t_\theta)), -\sin(\delta))_{\text{USN}}. \quad (\text{A-1})$$

On the other hand, any unit vector \vec{P} in the 3-D scene can be represented as $(\cos\phi\cos\theta, \cos\phi\sin\theta, \sin\phi)_{\text{USN}}$. Here, ϕ represents the angle between \vec{P} and the solar plane, and θ represents the angle subtended by \vec{u} and the projected vector of \vec{P} on the solar plane. In our system, we assume the scene surfaces are mainly Lambertian. Hence, if \vec{P} is the normal vector of a surface patch in the 3-D scene, the intensity value at the corresponding image pixel can be approximated as

$$I_{\text{sun}} \propto \langle \vec{D}(t), \vec{P} \rangle \propto -\cos(\delta)\cos(\phi)\cos(\omega_s t - \omega_s t_\theta - \theta) - \sin(\delta)\sin(\phi). \quad (\text{A-2})$$

Based on (A-2), I_{sun} can be modeled as

$$I_{\text{sun}}(m, n, t) = B(m, n)\cos(\omega_s t - \theta_p(m, n)) + C(m, n), \quad (\text{A-3})$$

where the angular frequency of the cosine function is equal to the angular frequency of Earth's self-rotation.

Assume we denote \vec{P}_1 , \vec{P}_2 , and \vec{P}_3 as the unit normal vectors of three selected surface patches in the parking lot. Since we manually select these three surface patches, the relative relationship among \vec{P}_1 , \vec{P}_2 , and \vec{P}_3 can be obtained beforehand. Suppose \vec{P}_1' , \vec{P}_2' , and \vec{P}_3' are the unit vectors along the projections of these three normal vectors onto the solar plane, and θ_1 , θ_2 , and θ_3 are the angles subtended by \vec{u} and each of these three projected vectors, as illustrated in Fig. 50. Since the phase shift θ_p in (A-3) is equal to θ up to a constant offset, the angles between these three projected vectors can be estimated by $(\theta_{p1}-\theta_{p2})$, $(\theta_{p2}-\theta_{p3})$, and $(\theta_{p1}-\theta_{p3})$.

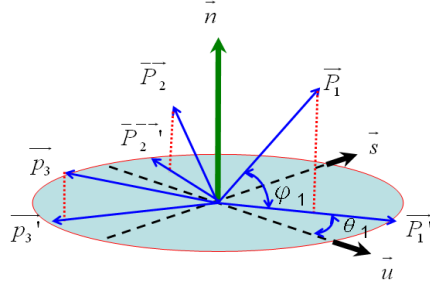


Fig. 50. Three normal vectors in the USN coordinate system.

Assume we represent \vec{n} as a linear combination of \vec{P}_1 , \vec{P}_2 , and \vec{P}_3 . That is, $\vec{n} = a\vec{P}_1 + b\vec{P}_2 + c\vec{P}_3$. If we take the inner product of \vec{n} and \vec{P}_i , where $i = 1, 2, 3$, we

obtain three equations to solve a , b , and c :

$$\begin{aligned} \langle \vec{P}_1, \vec{n} \rangle &= a \langle \vec{P}_1, \vec{P}_1 \rangle + b \langle \vec{P}_1, \vec{P}_2 \rangle + c \langle \vec{P}_1, \vec{P}_3 \rangle = \sin(\varphi_1) \\ \langle \vec{P}_2, \vec{n} \rangle &= a \langle \vec{P}_1, \vec{P}_2 \rangle + b \langle \vec{P}_2, \vec{P}_2 \rangle + c \langle \vec{P}_2, \vec{P}_3 \rangle = \sin(\varphi_2) \\ \langle \vec{P}_3, \vec{n} \rangle &= a \langle \vec{P}_1, \vec{P}_3 \rangle + b \langle \vec{P}_2, \vec{P}_3 \rangle + c \langle \vec{P}_3, \vec{P}_3 \rangle = \sin(\varphi_3) \end{aligned} \quad (\text{A-4})$$

In (A-4), the inner products $\langle \vec{P}_i, \vec{P}_j \rangle$, with $i, j = 1, 2, 3$, are known beforehand. To estimate $\{\varphi_1, \varphi_2, \varphi_3\}$, we formulate the vector \vec{P}_i' as $\vec{P}_i' = (\vec{P}_i - \sin \varphi_i \vec{n}) / \cos \varphi_i$. As

we take the inner products among \vec{P}_1' , \vec{P}_2' , and \vec{P}_3' , we have

$$\begin{aligned} \langle \vec{P}_1', \vec{P}_2' \rangle &= (\langle \vec{P}_1, \vec{P}_2 \rangle - \sin \varphi_1 \sin \varphi_2) / (\cos \varphi_1 \cos \varphi_2) = \cos(\theta_{p1} - \theta_{p2}) \\ \langle \vec{P}_2', \vec{P}_3' \rangle &= (\langle \vec{P}_2, \vec{P}_3 \rangle - \sin \varphi_2 \sin \varphi_3) / (\cos \varphi_2 \cos \varphi_3) = \cos(\theta_{p2} - \theta_{p3}) \\ \langle \vec{P}_3', \vec{P}_1' \rangle &= (\langle \vec{P}_3, \vec{P}_1 \rangle - \sin \varphi_3 \sin \varphi_1) / (\cos \varphi_3 \cos \varphi_1) = \cos(\theta_{p3} - \theta_{p1}) \end{aligned} \quad (\text{A-5})$$

Hence, with $\{(\theta_{p1} - \theta_{p2}), (\theta_{p2} - \theta_{p3}), (\theta_{p1} - \theta_{p3})\}$, the geometric direction of \vec{n} with respect to $\{\vec{P}_1, \vec{P}_2, \vec{P}_3\}$ can be deduced.

After the determination of \vec{n} , the choice of $\{\vec{n}, t_\theta\}$ is rather arbitrary. In our approach, we simply align \vec{n} with one of $\{\vec{P}_1', \vec{P}_2', \vec{P}_3'\}$. The reference time t_θ is defined to be the time when the corresponding intensity profile has the maximum value.

Appendix B: Image Formation Model

We assume the surfaces in the 3-D scene of a parking lot are mainly Lambertian. and the trichromatic RGB features at a pixel can be formulated as

$$I_c = g \int_{\lambda} l(\lambda) r(\lambda) f_c(\lambda) d\lambda. \quad (\text{A-6})$$

Here, g is a geometric factor that depends on the included angle between the incident radiant flux and the normal vector of the corresponding surface, $l(\lambda)$ denotes the illuminant spectrum, $r(\lambda)$ represents the spectral reflectance function, and $f_c(\lambda)$ represents the filter sensitivity function of the c channel with $c \in \{R, G, B\}$. To discretize (A-6) for computational analysis, several research works adopted finite-dimensional linear models to approximate both spectral reflectance function and illuminant spectrum. In our approach, we adopted a three-dimensional linear model and (A-6) is reformulated as

$$\begin{aligned} I_c &= g \int_{\lambda} \left(\sum_{i=1}^3 \beta_i l_i(\lambda) \right) \left(\sum_{j=1}^3 \alpha_j r_j(\lambda) \right) f_c(\lambda) d\lambda \\ &= g \sum_{i=1}^3 \beta_i \sum_{j=1}^3 \alpha_j \int_{\lambda} l_i(\lambda) r_j(\lambda) f_c(\lambda) d\lambda \\ &= g \boldsymbol{\beta}^T \mathbf{M}_c \boldsymbol{\alpha} = g \boldsymbol{\beta}^T \boldsymbol{\alpha}_c, \end{aligned} \quad (\text{A-7})$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T$ is the vector of illuminant coefficients, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^T$ is the vector of reflectance coefficients, \mathbf{M}_c is a 3×3 matrix with its entries defined as

$$\mathbf{M}_c(i, j) = \int_{\lambda} l_i(\lambda) r_j(\lambda) f_c(\lambda) d\lambda, \quad (\text{A-8})$$

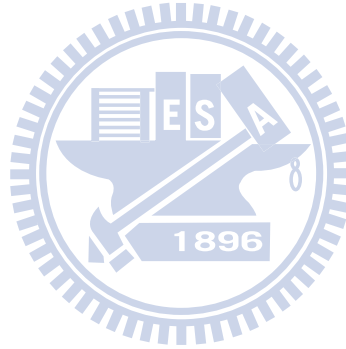
and $\boldsymbol{\alpha}_c = \mathbf{M}_c \boldsymbol{\alpha}$. With (A-8), the trichromatic color vector \mathbf{I}_{RGB} is represented as

$$\mathbf{I}_{\text{RGB}} = \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = g \begin{bmatrix} \boldsymbol{\alpha}_R^T \\ \boldsymbol{\alpha}_G^T \\ \boldsymbol{\alpha}_B^T \end{bmatrix} \cdot \boldsymbol{\beta} = g \mathbf{A} \boldsymbol{\beta}, \quad (\text{A-9})$$

where $\mathbf{A} = [\boldsymbol{\alpha}_R \quad \boldsymbol{\alpha}_G \quad \boldsymbol{\alpha}_B]^T$ is a 3×3 matrix.

In an outdoor parking lot, the lighting condition is varying over time. This makes both g and $\boldsymbol{\beta}$ change accordingly. To simplify the detection process, we focus mainly on the chromatic information. Since the absolute magnitudes of $\boldsymbol{\alpha}_c$ and $\boldsymbol{\beta}$ do not affect the chromatic information, we arbitrarily rescale \mathbf{A} and $\boldsymbol{\beta}$ by two constants a and b so that (A-9) can be reformulated as

$$\mathbf{I}_{\text{RGB}} = g\mathbf{A}\boldsymbol{\beta} = gab\left(\frac{1}{a}\mathbf{A}\right)\left(\frac{1}{b}\boldsymbol{\beta}\right) = (gab)\mathbf{Ri} = \|\mathbf{I}_{\text{RGB}}\|\mathbf{Ri}. \quad (\text{A-10})$$



Bibliography

- [1] Stauffer, C. and Grimson, W., “Adaptive Background Mixture Models for Real Time Tracking,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.
- [2] P. Power and J. A. Schoonees, “Understanding Modeling Background Mixture Models for Foreground Segmentation,” *Image and Vision Computing*, pp. 267-271, 2002.
- [3] D. Lowe, “Distinctive Image Features from Scale Invariant Key Points,” *International Journal of Computer Vision*, pp. 91-110, 2004.
- [4] K. Mikolajczyk and C. Schmid., “Scale and Affine Invariant Interest point Detectors,” *International Journal of Computer Vision*, pp. 63-86, 2004.
- [5] David Lowe and David G., “Object Recognition from Local Scale-invariant Features,” *International Conference on Computer Vision*, pp. 1150–1157, 1999.
- [6] Graham D. Finlayson, Mark S. Drew, and Cheng Lu, “Entropy Minimization for Shadow Removal,” *International Journal of Computer Vision*, pp. 13-30, 2009.
- [7] Y. Matsushita, K. Nishino, K. Ikeuchi, and S. Masao, “Illumination Normalization with Time-dependent Intrinsic Images for Video Surveillance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1336-1347, 2004.
- [8] Vivek Agarwal, Besma R. Abidi, Andreas Koschan, and Mongi A. Abidi, “An Overview of Color Constancy Algorithms,” *Journal of Pattern Recognition Research*, pp. 42-54, 2006.
- [9] Peter Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp, “Bayesian Color Constancy Revisited,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2008
- [10] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, pp. 1151-1163, 2002.
- [11] S. Funck, N. Mohler, and W. Oertel, “Determining Car-Park Occupancy from Single Images,” *IEEE Intelligent Vehicles Symposium*, pp. 325-328, 2004.
- [12] B. Bose, X. Wang, and E. Grimson, “Multi-class Object Tracking Algorithm that Handles Fragmentation and Grouping,” *International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
- [13] Luo-Wei Tsai, Jun-Wei Hsieh, and Kao-Chin Fan, “Vehicle Detection Using Normalized Color and Edge Map,” *IEEE Transactions on Image Processing*, pp. 850-864, 2007.
- [14] Marko Heikkilä and Matti Pietikäinen, “A Texture-Based Method for Modeling the Background and Detecting Moving Objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 657-662, 2006.

- [15] Y. Wang, K. F. Loe, and J. K. Wu, "A Dynamic Conditional Random Field Model for Foreground and Shadow Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 279-289, 2006.
- [16] Y. Sheikh and M. Shah, "Bayesian Modeling of Dynamic Scenes for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1778-1792, 2005.
- [17] Cs. Benedek and T. Szirányi, "Bayesian Foreground and Shadow Detection in Uncertain Frame Rate Surveillance Videos", *IEEE Transactions on Image Processing*, pp. 608-621, 2008,
- [18] T. Boykov, O. Veksler, and R. Zabih, "Markov Random Fields with Efficient Approximations," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 648-655, 1998.
- [19] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *International Journal of Computer Vision*, pp. 41-54, 2006.
- [20] R. S. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1068-1080, 2008.
- [21] S. Geman and C. Graffigne, "Markov Random Field Image Models and Their Applications to Computer Vision," *International Congress of Mathematicians*, pp. 1496-1517, 1986.
- [22] S. Li, "Markov Random Field Modeling in Computer Vision," *European Conference on Computer Vision*, pp. 361-370, 1995.
- [23] Kenji Suzuki, Isao Horiba, and Noboru Sugie, "Linear-time Connected-component Labeling Based on Sequential Local Operations," *Computer Vision and Image Modeling*, pp. 1-23, 2003.
- [24] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Proceedings of Data Mining and Knowledge Discovery*, pp. 1-43, 1998.
- [25] T. Evgeniou, M. Pontil, and T. Poggio., "A Unified Framework for Regularization Networks and Support Vector Machines," *Technical Report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology*, 1999.
- [26] Theodoros Evgeniou, Massimiliano Pontil, Constantine Papageorgiou, and Tomaso Poggio, "Image Representations for Object Detection Using Kernel Classifiers," *Asian Conference on Computer Vision*, pp. 687-692, 2000.
- [27] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, pp. 15-33, 2000.
- [28] N. Dalaland and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.
- [29] M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Transactions on Computers*, pp. 67-92, 1973.
- [30] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *International Journal of Computer Vision*, pp. 55-79, 2005.
- [31] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale,

- Deformable Part Model,” *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [32] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1627-1645, 2009.
- [33] Y. LeCun, S. Chopra, R. Hadsell, R. Marc’Aurelio, and F. Huang, “A Tutorial on Energy-based Learning,” *Predicting Structured Data*, MIT Press, 2006.
- [34] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, “Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods,” *Annals of Statistics*, pp. 1651-1686, 1998.
- [35] Y. Shi, A. Bobick, and I. Essa, “A Bayesian View of Boosting and its Extensions,” ser. GVU Technical Report; GIT-GVU-05-22, Georgia Institute of Technology, 2005.
- [36] Schapire, R.E. and Singer, Y., “Improved Boosting Algorithms Using Confidence-rated Predictions,” *Machine Learning*, pp. 297–336. 1999.
- [37] X. Li, L. Wang, E. Sung., “AdaBoost with SVM-based Component Classifiers,” *Engineering Applications of Artificial Intelligence*, pp. 785–795, 2008.
- [38] Laetitia Leyrit, Thierry Chateau, Christophe Tournayre, and Jean-Thierry Lapresté, “Association of AdaBoost and Kernel Based Machine Learning Methods for Visual Pedestrian Recognition,” *IEEE Intelligent Vehicles Symposium*, pp. 67 - 72, 2008.
- [39] JK Aggarwal and Q Cai, “Human Motion Analysis: A Review,” *Computer Vision and Image Understanding*, pp. 428-440. 1999.
- [40] T.Zhao and R. Nevatia, “Tracking Multiple Humans in Crowded Environment,” *International Conference on Computer Vision and Pattern Recognition*, pp. 406-413, 2004.
- [41] H. J. Lee and Z. Chen, “Knowledge-guided Visual Perception of 3-D Human Gait from a Single Image Sequence”, *IEEE Transactions on Systems, Man and Cybernetics*, pp. 336-342, 1992.
- [42] M. K. Leung and Y. H. Yang, “First Sight: A Human Body Outline Labeling System”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 359-377, 1995.
- [43] D. Hogg, “Model-based vision: A Program to See a Walking Person”, *Image and Vision Computing*, pp. 5-20, 1983.
- [44] T. Zhao and R. Nevatia, “Bayesian Human Segmentation in Crowded Situations,” *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 459-466, 2003.
- [45] T. Zhao and R. Nevatia, “Tracking Multiple Humans in Complex Situations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1208-1221, 2004.
- [46] C.C. Huang, S. J. Wang, Y. J. Chang, and T. Chen, “A Bayesian Hierarchical Detection Framework for Parking Space Detection,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2097 – 2100, 2008
- [47] Gibson J., “The Perception of the Visual World,” *Houghton Mifflin*, 1950.
- [48] Warren R.M. and Warren R.P., “*Helmholtz on Perception: Its Physiology and Development*,” *John Wiley & Sons*, 1968.

- [49] Koenderink J.J., “Pictorial Relief,” *Philosophical Transactions of the Royal Society*, pp. 1071-1086, 1998.
- [50] Koenderink J.J., Doorn, A.J.V., and Kappers, A.M.L., “Pictorial surface Attitude and Local Depth Comparisons,” *Perception and Psychophysics*, pp. 163–173, 1996.
- [51] D. Hoiem, A. A. Efros, and M. Hebert, “Geometric Context from a Single Image,” *International Conference on Computer Vision*, pp. 654 - 661, 2005
- [52] D. Hoiem, A. A. Efros, and M. Hebert, “Recovering Surface Layout from an Image,” *International Journal of Computer Vision*, pp. 151–172, 2007.
- [53] D. Hoiem, A. A. Efros, and M. Hebert, “Putting Objects in Perspective,” *International Journal of Computer Vision*, pp. 3-15, 2008.
- [54] D. Hoiem, A. A. Efros, and M. Hebert, “Automatic Photo Pop-up,” *ACM SIGGRAPH*, pp. 577-584, 2005.
- [55] Sudderth, E., Torralba, A., Freeman, W.T., and Wilsky, A., “Learning Hierarchical Models of Scenes, Objects, and Parts,” *International Conference on Computer Vision*, pp. 1331-1338, 2005.
- [56] Hoiem D., Efros, A.A., and Hebert, M., “Closing the Loop in Scene Interpretation,” *International Conference on Computer Vision and Pattern*, pp. 1-8, 2008.
- [57] Oliva, A. and Torralba, A., “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope,” *International Journal of Computer Vision*, pp. 145-175, 2001.
- [58] Saxena, A., Chung, S., and Ng, A.Y., “Learning Depth from Single Monocular Images,” *International Conference on Neural Information Processing Systems*, pp. 1161-1170, 2005.
- [59] Sudderth, E., Torralba, A., Freeman, W.T., and Wilsky, A., “Depth from Familiar Objects: A Hierarchical Model for 3D Scenes,” *International Conference on Computer Vision and Pattern Recognition*, pp. 2410 – 2417, 2006.
- [60] Toyoda, T. Tagami, K. Hasegawa, O. “Integration of Top-down and Bottom-up Information for Image Labeling,” *International Conference on Computer Vision and Pattern Recognition*, pp. 1106-1113, 2006.
- [61] Kumar, S., Hebert, M. “A Hierarchical Field Framework for Unified Context-based Classification,” *International Conference on Computer Vision*, pp. 1284–1291, 2005.
- [62] A. Kapoor, J. Winn, “Located Hidden Random Fields: Learning Discriminative Parts for Object Detection,” *European Conference on Computer Vision*, pp. 302-315, 2006.
- [63] J. Winn, J. Shotton. “The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects”, *International Conference on Computer Vision and Pattern Recognition*, pp. 37- 44, 2006.
- [64] Y. Boykov, O. Veksler and R. Zabih, “Efficient Approximate Energy Minimization via Graph Cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.1222-1239, 2001.
- [65] Vladimir Kolmogorov and Ramin Zabih, “What Energy Functions can be Minimized via Graph Cuts?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 147-159, 2004.
- [66] Y. Boykov and V. Kolmogorov, “An Experimental Comparison of Min-Cut/Max-Flow

- Algorithms for Energy Minimization in Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1124 -1137, 2004.
- [67] M.Y.I. Idris, Y.Y. Leng, E.M. Tamil, N.M. Noor and Z. Razak, “Car Park System: A Review of Smart Parking System And Its Technology,” *Information Technology Journal*, pp. 101-113, 2009.
- [68] Henry Schneiderman and Takeo Kanade, “Object Detection Using the Statistics of Parts,” *International Journal of Computer Vision*, pp. 151-177, 2004.
- [69] P. Viola and M. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision*, pp.137-154, 2004.
- [70] C. H. Lee, M. G. Wen, C. C. Han and D. C. Kou, “An Automatic Monitoring Approach for Unsupervised Parking Lots in Outdoors,” *International Conference on Security Technology*, pp.271-274, 2005.
- [71] I. Masaki, “Machine-Vision Systems for Intelligent Transportation Systems,” *IEEE International Conference on Intelligent Transportation System*, pp. 24-31, 1998.
- [72] D.B.L. Bong, K.C. Ting, and K.C. Lai, “Integrated Approach in the Design of Car-Park Occupancy Information System,” *International Journal of Computer Science*, pp. 1-8, 2008.
- [73] K. Yamada, M. Mizuno, “A Vehicle Parking Detection Method Using Image Segmentation,” *Electronics and Communications*, pp. 25-34, 2001.
- [74] C.H. Lee, M. G. Wen, C. C. Han, and D. C. Kuo, “An Automatic Monitoring Approach for Unsupervised Parking Lots in Outdoor,” *IEEE International Conference on Security Technology*, pp. 271-274, 2005.
- [75] Tomas Fabian, "An Algorithm for Parking Lot Occupation Detection," *IEEE Computer Information Systems and Industrial Management Applications*, pp. 165-170, 2008.
- [76] Noah Dan, “Parking Management System and Method,” *US patent*, Pub. No.: 20030144890A1, Jul 2003.
- [77] Q. Wu, C. C. Huang, S. Y. Wang, W. C. Chiu, and T. H. Chen, “Robust Parking Space Detection Considering Inter-Space Correlation,” *IEEE International Conference on Multimedia and Expo*, pp. 659-662, 2007.
- [78] U.S. Naval Observatory. (2010). *Naval Oceanography Portal* [Online]. Available at <http://www.usno.navy.mil/USNO/>
- [79] K. Sunkavalli, F. Romeiro, W. Matusik, Y. Zickler, and H. Pfister, “What do Color Changes Reveal about an Outdoor Scene?” *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [80] Yanghai Tsin, Robert Collins, Visvanathan Ramesh, and Takeo Kanade, “Bayesian Color Constancy for Outdoor Object Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1132- 1139, 2001.
- [81] Davies, D.L., Bouldin, D.W., “A Cluster Separation Measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 224-227, 1979.
- [82] Ching-Chun Huang. (2010). *Huang’s Projects* [Online]. Available at <http://140.113.238.220/~>

chingchun/projects.html.

- [83] S. Khan and M. Shah, "Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1355-1360, 2003.
- [84] Weiming Hu, Min Hu, Tieniu Tan, Jianguang Lou, and Steve Maybank, "Principal Axis-based Correspondence between Multiple Cameras for People Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 663-671, 2006.
- [85] J. Black and T. Ellis, "Multi Camera Image Measurement and Correspondence," *Measurement - Journal of the International Measurement Confederation*, pp. 61-71, 2002.
- [86] A. Mittal and L. Davis, "Unified Multi-camera Detection and Tracking Using Region-matching," in *Proceedings of IEEE Workshop on Multi-Object Tracking*, pp. 3-10, 2001.
- [87] A. Mittal and L. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene", *International Journal of Computer Vision*, pp. 189-203, 2003.
- [88] A. Utsumi, H. Mori, J. Ohya and M. Yachida, "Multiple-human Tracking Using Multiple Cameras," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 498-503, 1998.
- [89] F. Fleuret, R. Lengagne, and P. Fua, "Fixed Point Probability Field for Complex Occlusion Handling," *IEEE International Conference on Computer Vision*, pp. 694-700, 2005.
- [90] J. Berclaz, F. Fleuret, and P. Fua, "Robust People Tracking with Global Trajectory Optimization," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 744-750, 2006.
- [91] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multi-Camera People Tracking with a Probabilistic Occupancy Map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 267-282, 2008.
- [92] Ching-Chun Huang, and Sheng-Jyh Wang, "A Monte Carlo Based Framework for Multi-Target Detection and Tracking Over Multi-Camera Surveillance System," *European Conference on Computer Vision Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.
- [93] Ching-Chun Huang, and Sheng-Jyh Wang, "Moving Targets Labeling and Correspondence over Multi-Camera Surveillance System Based on Markov Network," *IEEE International Conference on Multimedia and Expo*, pp. 1258-1261, 2009.
- [94] Saad M Khan, and Mubarak Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 505-519, 2009.
- [95] Kazuhiro Otsuka, and Naoki Mukawa, "Multiview Occlusion Analysis for Tracking Densely Populated Objects Based on 2-D Visual Angles," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 90-97, 2004.
- [96] Li Guan, Jean-Sebastien Franco, and Marc Pollefeys, "Multi-Object Shape Estimation and

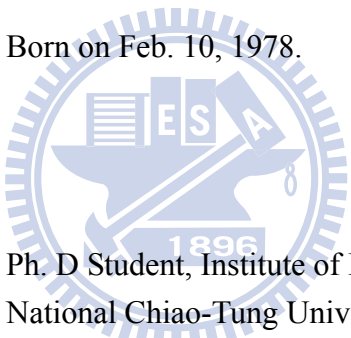
- Tracking from Silhouette Cues”, *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [97] B. Georgescu, I. Shimshoni, P. Meer, “Mean Shift Based Clustering in High Dimensions: A Texture Classification Example,” *IEEE International Conference on Computer Vision*, pp. 456-463, 2003.
- [98] P. KaewTraKulPong, R. Bowden, “An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection,” *European Workshop on Advanced Video-based Surveillance Systems*, pp. 135-145, 2001.
- [99] Fajie Li and Reinhard Klette, “A Variant of Adaptive Mean-Shift Based Clustering,” *International Conference on Neural Information Processing*, pp. 1002-1009, 2009.
- [100] Ching-Chun Huang. (2010). *Huang’s Projects* [Online]. Available at <http://140.113.238.220/~chingchun/projects.html>.
- [101] Chris Stauffer, W. E. L. Grimson, “Adaptive Background Mixture Models for Real-time Tracking,” *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.
- [102] T. Horprasert, D. Harwood, L. A. Davis, “A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection,” *IEEE International Conference on Computer Vision*, pp. 1-19, 1999.
- [103] Abhinav Gupta, Anurag Mittal and Larry S. Davis, “COST: An Approach for Camera Selection and Multi-Object Inference Ordering in Dynamic Scenes,” *IEEE International Conference on Computer Vision*, pp. 14-21, 2007.
- [104] 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*, pp. 780-785, 1997.

Curriculum Vita

Name:

Ching-Chun Huang (黃敬群) Born on Feb. 10, 1978.

Education:

- 
- Sep. 2002 ~ June 2010 Ph. D Student, Institute of Electronics,
National Chiao-Tung University, Hsin-chu, Taiwan
- Sep. 2000 ~ June 2002 M.S. Student, Institute of Electronics,
National Chiao-Tung University, Hsin-chu, Taiwan
- Sep. 1996 ~ June 2000 B.S. Student, Institute of Electronics,
National Chiao-Tung University, Hsin-chu, Taiwan

Work Experience:

- Oct. 2002 ~ Mar. 2008 Work in Industrial Technology Research Institute -
Advanced Technology Center (工業技術研究院 - 前
瞻技術中心)
- Aug. 2005 ~ Dec. 2005 Visiting Researcher at Carnegie Mellon University (美
國卡內基美隆大學)
- Aug. 2006 ~ Dec. 2006 Visiting Researcher at Carnegie Mellon University (美

國卡內基美隆大學)

July 2000 ~ Sep. 2000 Summer Job in Texas Instrument Corp. (德州儀器)

July 1998 ~ Sep. 1999 Summer Job in D-Link Corp. (友訊科技)

Publications:

Dissertation:

Ph. D Dissertation: A Study of a Bayesian Hierarchical Framework for Video Surveillance Systems
(貝氏階層式結構於視訊監控系統之研究)

M.S. thesis: Textured Segmentation Based on Spatial-Frequency Domain Analysis
(基於空間頻率域分析之紋理切割研究)

Journal Papers:

- [1] Ching-Chun Huang, Sheng-Jyh Wang, “A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection,” *IEEE Transactions on Circuits and Systems for Video Technology*. (Accepted)
- [2] Ching-Chun Huang, Sheng-Jyh Wang, “A Bayesian Hierarchical Framework for Multi-Target Labeling and Correspondence with Ghost Suppression over Multi-Camera Surveillance System,” *IEEE Transactions on Automation Science and Engineering*. (Under Revision)

International Conference Papers:

- [1] Ching-Chun Huang, Sheng-Jyh Wang, “A Cascaded Hierarchical Framework for Moving Object Detection and Tracking,” *IEEE International Conference on International Conference Image Processing (ICIP)*, Hong Kong, Sep. 26-29, 2010.
- [2] Ching-Chun Huang, Wei-Chen Chiu, Sheng-Jyh Wang, and Jen-Hui Chuang, “Probabilistic Modeling of Dynamic Traffic Flow across Non-overlapping Camera Views,” *IEEE International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, August 23-26, 2010.
- [3] Ching-Chun Huang and Sheng-Jyh Wang, “Moving Targets Labeling and Correspondence over Multi-Camera Surveillance System Based on Markov

- Network,” *IEEE International Conference on Multimedia and Expo (ICME)*, Cancun, Mexico, June 28-July 3, 2009.
- [4] Arvind Kandhalu, Anthony Rowe, Ragunathan Rajkumar, Ching-Chun Huang, and Chao-Chun Yeh, “Real-Time Video Surveillance over IEEE 802.11 Mesh Networks,” *IEEE International Real-Time and Embedded Technology and Applications Symposium*, San Francisco, CA April 13-16, 2009.
- [5] Ching-Chun Huang and Sheng-Jyh Wang, “A Monte Carlo Based Framework for Multi-Target Detection and Tracking Over Multi-Camera Surveillance System,” *accepted by the 10th European Conference on Computer Vision (ECCV) Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, October 12-18, 2008.
- [6] Ching-Chun Huang, Sheng-Jyh Wang, Yao-Jen Chang, and Tsuhan Chen, “A Bayesian Hierarchical Detection Framework for Parking Space Detection,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2008)*, Las Vegas, NV, USA, Mar. 30-Apr. 4, 2008.
- [7] Ming-Yu Shih, Yao-Jen Chang, Bwo-Chau Fu, and Ching-Chun Huang, “Motion-based Background Modeling for Moving Object Detection on Moving Platforms,” *Proceedings of 2007 First International Workshop on Multimedia Analysis and Processing (IMAP-2007), in conjunction with the 16th International Conference on Computer Communications and Networking (ICCCN-2007)*, Honolulu, Hawaii, USA, Aug. 15-16, 2007.
- [8] Q. Wu, Ching-Chun Huang, S.Y. Wang, W.C. Chiu, and T. Chen, “Robust Parking Space Detection Considering Inter-space Correlation,” *IEEE International Conference on Multimedia and Expo (ICME)*, Beijing, China, July 2-5 2007.
- [9] Ching-Chun Huang and Cheng Yi Liu, “Novel Illumination-Normalization Method Based on Region Information,” *Proceedings of SPIE*, San Jose, USA, March 2005, pp. 339-348.
- [10] Y. H. Tsai and Ching-Chun Huang, “Handheld Person Verification System Using Face Image,” *Digital Image Computing - Techniques and Applications*, 2003.

Domestic Papers:

- [1] Wei-Chen Chiu, Ching-Chun Huang, Jen-Hui Chuang, and Sheng-Jyh Wang, “Probabilistic Modeling of Dynamic Traffic Flow between Non-overlapping FOVs,” *IPPR Conference on Computer Vision, Graphics, and Image Processing*, Taiwan, Aug, 2009. (學會佳作論文)
- [2] Ching-chun Huang and Yao-Jen Chang, “Vision-Based Parking Spaces Detection,” *CCL Technical Journal*, v. 120, pp. 72-80, 2007.

- [3] Ching-Chun Huang and Bwo-Chau Fu, "Vision-Based Parking Lot Surveillance Services," *CCL Technical Journal*, v. 116, pp. 81-88, 2006.
- [4] Yi-Tsung Chien, Ting-Wu Ho, and Ching-Chun Huang, "Framework for Manageable Multi-Event Detection, Alarming and Monitoring," *CCL Technical Journal*, v. 112, pp. 84-92, 2005.
- [5] Ching-Chun Huang and Cheng-Yi Liu, "A Novel Region based Illumination normalization Method for Face Recognition," *IPPR Conference on Computer Vision, Graphics, and Image Processing*, Taiwan, Aug, 2004.
- [6] Cheng-Yi Liu, Ching-chun Huang, and Yao-Hong Tsai, "Image Retrieval Technology Based on MPEG-7 Still Image Descriptors," *CCL Technical Journal*, v. 104, pp. 86-95, 2003. (最佳論文獎第三名)
- [7] Yao-Hong Tsai, Ching-chun Huang, Raymond B.C.Fu, Li-Wu Huang, and Cheng-Yi Liu, "Handheld Person Verification System Using Face Image," *IPPR Conference on Computer Vision, Graphics, and Image Processing*, Taiwan, Aug, 2003.

Filed U.S. Patents:

- [1] Chingchun Huang and Sheng-Jyh Wang, "Multi-Target Detection and Tracking Over Multi-Camera Surveillance System," Filing No.: 12/481,910, Filing Date: Sep. 14, 2009.
- [2] Ching-Chun Huang, Yao-Jen Chang, Ruei-Cheng Wu, and Cheng-Peng Kuan, "System and Method of Image-Based Space Detection," Filing No.: 12/111,190, Filing Date: Apr. 28, 2008.
- [3] Ching-Chun Huang, and Yao-Jen Chang, "Method and System for Object Detection and Tracking," Filing No.: 11/959,462, Filing Date: Dec. 19, 2007.
- [4] Ching-Chun Huang, and Yao-Hong Tsai, "Region based illumination-normalization method and system," Issued Patent #7263241.

Filed Taiwan Patents:

- [1] Ching-chun Huang and Sheng-Jyh Wang, "Multi-Target Detection and Tracking Over Multi-Camera Surveillance System," Filing No.: 097150635, Filing Date: Dec. 25, 2008.
- [2] Ching-chun Huang, Yao-Jen Chang, Ruei-Cheng Wu, and Cheng-Peng Kuan, "System and Method of Image-Based Space Detection," Filing No.: 096136320, Filing Date: Sep. 28, 2007.
- [3] Ching-chun Huang, and Yao-Jen Chang, "Method and System for Object Detection and Tracking," Filing No.: 096140534, Filing Date: Oct. 29, 2007.

- [4] Ching-chun Huang, and Yao-Hong Tsai, “Region based illumination-normalization method and system,” Issued Patent #I245229.

Filed China Patents:

- [1] Ching-chun Huang, Yao-Jen Chang, Ruei-Cheng Wu, and Cheng-Peng Kuan, “System and Method of Image-Based Space Detection,” Filing No.: 200710182315.2, Filing Date: Oct. 17, 2007.
- [2] Ching-chun Huang, and Yao-Jen Chang, “Method and System for Object Detection and Tracking,” Filing No.: 200710169282.8, Filing Date: Nov. 8, 2007.

