



Feature-based Image Patch Classification for Moving Shadow Detection

Mosin Russell

A thesis submitted in fulfillment of the requirements for the
degree of Doctor of Philosophy in Engineering

SCHOOL OF COMPUTING, ENGINEERING AND MATHEMATICS
WESTERN SYDNEY UNIVERSITY
AUSTRALIA

July 28, 2019

© Mosin Russell, 2019

ABSTRACT

Moving object detection is a first step towards many computer vision applications, such as human interaction and tracking, video surveillance, and traffic monitoring systems. Accurate estimation of the target object's size and shape is often required before higher-level tasks (e.g., object tracking or recognition) can be performed. However, these properties can be derived only when the foreground object is detected precisely.

Background subtraction is a common technique to extract foreground objects from image sequences. The purpose of background subtraction is to detect changes in pixel values within a given frame. The main problem with background subtraction and other related object detection techniques is that cast shadows tend to be misclassified as either parts of the foreground objects (if objects and their cast shadows are bonded together) or independent foreground objects (if objects and shadows are separated). The reason for this phenomenon is the presence of similar characteristics between the target object and its cast shadow, i.e., shadows have similar motion, attitude, and intensity changes as the moving objects that cast them.

Detecting shadows of moving objects is challenging because of problematic situations related to shadows, for example, chromatic shadows, shadow color blending, foreground-background camouflage, nontextured surfaces and dark surfaces.

Various methods for shadow detection have been proposed in the literature to address these problems. Many of these methods use general-purpose image feature descriptors to detect shadows. These feature descriptors may be effective in distinguishing shadow points from the foreground object in a specific problematic situation; however, such methods often fail to distinguish shadow

points from the foreground object in other situations. In addition, many of these moving shadow detection methods require prior knowledge of the scene conditions and/or impose strong assumptions, which make them excessively restrictive in practice.

The aim of this research is to develop an efficient method capable of addressing possible environmental problems associated with shadow detection while simultaneously improving the overall accuracy and detection stability.

In this research study, possible problematic situations for dynamic shadows are addressed and discussed in detail. On the basis of the analysis, a robust method, including change detection and shadow detection, is proposed to address these environmental problems. A new set of two local feature descriptors, namely, binary patterns of local color constancy (BPLCC) and light-based gradient orientation (LGO), is introduced to address the identified problematic situations by incorporating intensity, color, texture, and gradient information. The feature vectors are concatenated in a column-by-column manner to construct one dictionary for the objects and another dictionary for the shadows. A new sparse representation framework is then applied to find the nearest neighbor of the test image segment by computing a weighted linear combination of the reference dictionary. Image segment classification is then performed based on the similarity between the test image and the sparse representations of the two classes.

The performance of the proposed framework on common shadow detection datasets is evaluated, and the method shows improved performance compared with state-of-the-art methods in terms of the shadow detection rate, discrimination rate, accuracy, and stability. By achieving these significant improvements, the proposed method demonstrates its ability to handle various problems associated with image processing and accomplishes the aim of this thesis.

DECLARATION

Date: July 28, 2019

Author: Mosin Russell

Title: Feature-based Image Patch Classification for Moving Shadow
Detection

Degree: Ph.D.

I certify that the work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text, and that the material has not been submitted, either in full or in part, for a degree at this or any other institution.

I certify that I have complied with the rules, requirements, procedures and policy relating to my higher degree research award of Western Sydney University.

Author's Signature

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. Ju Jia (Jeffrey) Zou for his useful comments, remarks and engagement throughout the learning process of this Ph.D. thesis. Without his encouragement and guidance, this research would not have materialized. Furthermore, I would like to thank my associated supervisor, Professor Gu Fang, for his assistance and support.

I thank Western Sydney University for the financial support in the form of a Western Sydney University Postgraduate Research Award Program scholarship. I may not have embarked on this incredible journey otherwise.

Finally, and most importantly, I would like to thank my wife, Kazhal, who has supported me throughout the entire process; without her I would never have had the strength to succeed. I will be grateful forever for her love.

LIST OF PUBLICATIONS

Journal Papers

- M. Russell, J.J. Zou, G. Fang, and W. Cai, “Feature-based Image Patch Classification for Moving Shadow Detection”, IEEE Transactions on Circuits and Systems for Video Technology, 2017, DOI: 10.1109/TCSVT.2017.2763181.
- M. Russell, J.J. Zou, and G. Fang, “An evaluation of moving shadow detection techniques”, Computational Visual Media, 2(3), 195-217, 2016.
- M. Russell, J.J. Zou, and G. Fang, “Real-time vehicle shadow detection”, Electronics Letters, 51(16), 1253-1255, 2015.
- M. Russell, J.J. Zou, and G. Fang, “Background Patch Matching for Moving Object Detection”, currently in draft.

Conference Papers

- M. Russell, J.J. Zou, G. Fang, and W. Cai, “Feature-based Patch Matching for Moving Object Detection”, In 2019 IEEE Visual Communications and Image Processing (VCIP), **accepted**.
- M. Russell, J.J. Zou, and G. Fang, “A Novel Region-Based Method for Moving Shadow Detection”, In IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1-6, 2016.
- M. Russell, J.J. Zou, and G. Fang, “Human shadow detection for real-time applications”, In IEEE Region 10 Conference (TENCON), pp. 1610-1613, 2016.

- M. Russell, J.J. Zou, and G. Fang, “Moving shadow detection: a survey of recent methods”, In Proceedings of the International Conference on Computer Science and Information Engineering (CSIE2015), pp. 253, 2015.

Contents

ABSTRACT	I
DECLARATION	III
ACKNOWLEDGEMENTS	IV
List of Publications	V
List of Figures	XXIII
List of Tables	XXVI
List of Algorithms	XXVII
Abbreviations	XXVIII
1 Introduction	1
1.1 Background	1
1.2 Existing Shadow Detection Methods	3
1.3 Problem Statement	6
1.4 Research Aim	7
1.5 Scope of the Research Study	8
1.6 Method Outline	9

1.7	Contributions	11
1.8	Thesis Outline	12
2	Literature Review	15
2.1	Introduction	15
2.2	Existing Shadow Detection Methods	15
2.2.1	Shape-based Methods	18
2.2.2	Light-direction-based Methods	20
2.2.3	Color-based Methods	22
2.2.4	Texture-based Methods	26
2.2.5	Gradient-based Methods	29
2.2.6	Segmentation-based Methods	33
2.2.7	Block-based Methods	36
2.2.8	Drawbacks	38
2.3	Summary	41
3	Problematic Situations	44
3.1	Understanding Shadows	44
3.1.1	Cast Shadow Model	45
3.1.2	Shadow Properties	47
3.2	Problematic Situations	49
3.2.1	Achromatic Shadow	49
3.2.2	Chromatic Shadow	52

3.2.3	Shadow Color Blending	53
3.2.4	Foreground-background Camouflage	54
3.2.5	Nontextured Surfaces	57
3.2.6	Dark Surfaces	58
3.3	Dataset Classification	61
3.4	Summary	66
4	Change Detection	68
4.1	Introduction	68
4.2	Existing Change Detection Methods	70
4.3	Proposed Change Detection Framework	71
4.3.1	Background Initialization	72
4.3.2	Feature Descriptors	78
4.3.3	Background Modeling	87
4.3.4	Pixel-level Classification	89
4.3.5	Background Updating	90
4.3.6	Region-level Classification	91
4.3.7	Outputs	93
4.4	Summary	94
5	Moving Shadow Detection	96
5.1	Introduction	96
5.2	Candidate Shadow Segments	98

5.3	Dictionary Construction	101
5.4	Feature Extraction	106
5.4.1	Local Structure Descriptor	107
5.4.2	Gradient Descriptor	113
5.4.3	Intensity Descriptor	120
5.4.4	Feature Descriptor	120
5.5	Segment Classification	123
5.6	Region-level Classification	126
5.7	Summary	129
6	Results and Discussion	133
6.1	Introduction	133
6.2	Datasets	136
6.3	Evaluation Metrics	138
6.4	Parameter Settings	140
6.5	Evaluation of the Proposed Framework	145
6.5.1	State-of-the-art of Comprehensive Methods	145
6.5.2	Qualitative Performance	146
6.5.3	Quantitative Performance	160
6.6	Evaluation of Features	162
6.6.1	State-of-the-art Features	163
6.6.2	Qualitative Performance	168

6.6.3	Quantitative Performance	170
6.7	Evaluation of Shadow Detection Methods	171
6.7.1	Qualitative Performance	174
6.7.2	Quantitative Performance	176
6.7.3	Stability	182
6.8	Summary	184
7	Conclusion and Future Work	190
7.1	Conclusion	190
7.2	Future Work	194
	References	195
	Appendices	215
	Appendix A: MTM Matching	215
	Appendix B: Other Common Datasets	220

List of Figures

1.1	Examples of shadow applications; (first column) detection of moving shadows in a frame of the sequence <i>HwyI</i> using the method reported in [1], (second column) manual detection of shadows of a building in still image, (third column) manual detection and correction of shadows for a still image, and (fourth column) manual detection and mapping of the shadow for an outdoor still image.	2
1.2	Examples of problematic situations for shadow detection; (a) chromatic shadow (cast shadow and background surface are shown in the normalized rgb colour space for better comparison), (b) shadow colour-blending (parts of the moving person are reflected by the background surface), (c) foreground-background camouflage, (d) non-textured surfaces, and (e) dark surfaces. A frame is shown on the left where regions with the problem are highlighted by a red rectangle. A magnified section in the given frame is shown on the top-right corner, and the same section in the background surface is shown on the bottom-right corner.	6
1.3	Block diagram of proposed moving shadow detection.	10
2.1	Classification of moving shadow detection.	17

2.2	Illustration of shape-based moving shadow detection as reported in [31]; (a) the center of gravity and orientation of a shaded object R, and (b) object and its contour information.	18
2.3	Illustration of shape-based moving shadow detection as reported in [33]. Six possible shadow locations are shown.	19
2.4	Illustration of light-direction based for moving shadow detection as reported in [36]; (a) eight directions of illumination, and (b) direction of illumination, the green arrow, is determined by the model.	21
2.5	Different colour spaces used for moving shadow detection.	23
2.6	Illustration of texture feature information; (a) a small patch of the background image extracted and magnified before shadow occur, and (b) the patch is occluded by shadow at a different time. . . .	27
2.7	Visualization of gradient orientation and edges detection obtained using different operators; (a1 - a2) same image under two different illumination conditions, (b1 - b2) edge detection using Perwitt operator, (c1 - c2) edge detection using Sobel operator, (d1 - d2) edge detection using Canny operator, (a3 - b3) gradient magnitude and direction of image (a1), and (c3 - d3) gradient magnitude and direction of image (a2).	31
3.1	Shadow model	46

3.2	An example of linear attenuation of achromatic shadows; (a) a frame in the sequence (top) and the corresponding background (bottom) with an area (highlighted in blue) under the test, and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel) for a number of frames.	50
3.3	Discriminating the two classes, shadows and moving objects, for achromatic shadows for an area highlighted in blue; (a) the background frame, (b) a current frame in the sequence, (c) the ground truth image of (b), (d) red-green joint distribution in normalized RGB space, (e) red-blue joint distribution in normalized RGB space, and (f) Green-Blue joint distribution in normalized RGB space.	51
3.4	An example of non-linear attenuation of chromatic shadows; (a) a frame in the sequence (top), the corresponding background (bottom) with an area (highlighted in blue) under the test, and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel) for a number of frames.	52
3.5	Discriminating the two classes, shadows and moving objects, for chromatic shadows for an area highlighted in blue; (a) the background frame, (b) a current frame in the sequence, (c) the ground truth image of (b), (d) red-green joint distribution in normalized RGB space, (e) red-blue joint distribution in normalized RGB space, and (f) green-blue joint distribution in normalized RGB space.	53

3.6	An example of shadow colour blending; (a) a frame in the sequence with an area (highlighted in white) under the test and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel)for a number of frames.	54
3.7	An example of RGB intensity differences for a foreground patch (highlighted in black) affected by camouflage; (a) a given image under the test, (b) background image, (c) ground truth image of (a), (d) intensity histogram, calculated for red channel, for all points belong to the patch in the background image (blue bars) and the current frame (red bars), (e) intensity histogram for green channel, and (f) intensity histogram for blue channel. Euclidean distance (ED) is used to measure the differences.	55
3.8	An example of foreground-background camouflage; (a) a given image under the test masked with CDM , (b) background image masked with CDM , (c) ground truth image (camouflage area for the foreground object is highlighted in blue, non-camouflage area is highlighted in red and the shadow cast is highlighted in green), (d) the red-green joint distribution in normalized RGB space, (e) the red-blue joint distribution in normalized RGB space, and (f) the green-blue joint distribution in normalized RGB space.	56

3.9	An example of non-textured surfaces; (a) the image under the test masked with CDM , (b) the background image masked with CDM , (c) ground truth image (non-textured area for the foreground object is highlighted in blue, textured area is highlighted in red and shadow cast is highlighted in green), (d) the red-green joint distribution in normalized RGB space, (e) the red-blue joint distribution in normalized RGB space, and (f) the green-blue joint distribution in normalized RGB space.	59
3.10	An example of dark surfaces; (a) the frame under the test, (b) the background image, (c) ground truth image (dark area for the foreground object is highlighted in blue, bright area is highlighted in red and shadow cast is highlighted in green), (d) the Red-Green joint distribution in normalized RGB space, (e) the Red-Blue joint distribution in normalized RGB space, and (f) the Green-Blue joint distribution in normalized RGB space.	60
3.11	Labelling the two classes, objects and shadows, for dataset classification; (a) a CDM , (b) ground truth image of (a) consists of moving object O and its shadow S , (c) background image BG masked with CDM , and (d) current frame CF masked with CDM	61
3.12	Computing the texture descriptor T for a small patch P of a moving object O^{CF} in the current frame; (a) a patch P in the moving object is taken and magnified, (b) intensity values for all pixels in the patch are given, (c) computing pixel-level texture descriptor $T_o(p)$ for three pixels with different spatial locations, and (d) computing pixel-level texture descriptor for all pixels belong to the patch. The patch-level texture descriptor $T(P)$ is calculated for the patch.	65

4.1	Examples of change detection masks for four different scenarios; (a1 - a4) background image, (b1 - b4) given frame, and (c1 - c4) change detection masks. The foreground objects are highlighted in white and the backgrounds are in black.	69
4.2	Examples of various challenges for change detection. The object of interests are highlighted in red, high frequency objects are highlighted in blue, objects caused changes in background geometry are highlighted in cyan, areas directly affected by global change in illumination are highlighted in yellow, and cast shadows are highlighted in green.	70
4.3	Block diagram of the proposed change detection method. The current frame, background image, and change detection mask are labelled as CF , BG , and CDM , respectively.	72
4.4	Computing background image for a busy traffic scene using the proposed block-based background initialization method; (a) a frame at $t=5$ divided into 12×16 blocks, (b) a frame at $t=10$ divided into 12×16 blocks, (c) initial background model obtained by applying Equation (4.4) with estimated area of changes highlighted in black, (d) the background scene image computed using 20 frames, (e) histogram of intensity mean of a foreground block highlighted in green in (d), and (f) visualization of the 20 image patterns used to compute the histogram shown in (e).	77

- 4.5 Illustration of global intensity consistency for two points, one is a background point p_b and the other is an object point p_o ; (a) the current frame contains three different background surface shapes A, B, and C (highlighted in green) and one target object (shape D highlighted in blue) and (b) the background image contains only the background surface shapes. The reference point q is highlighted in yellow in both images. Note that $p_o \in D$ in (a) and $p_o \in C$ in (b). 82
- 4.6 Computation process of **GIC** feature descriptor; (a) a background image in **RGB** colour space, (b) a current frame in **RGB** colour space, (c) the current frame in gray-scale. These images are divided into 9 large blocks. The best match pixels for the reference blocks are highlighted in red. The test pixel is highlighted in green, (d) indexes are assigned for each block for simplification, (e) intensity values for the best match pixels of the reference blocks, taken from (c), are shown, and (f) computing processes of the **GIC** feature descriptor for the test pixel with intensity value of 162 (highlighted in green in (e)). **GIC**(162)=62. 85
- 4.7 Illustration of *HSV*T feature comparison for three points belong to different surface normals; (a) the current frame with an object point *obj* (highlighted in red), a shaded point *sh* (highlighted in green), and a background point *bg* (highlighted in blue) are selected for testing, (b) the corresponding points in the background image, (c) the feature bar chart for the background point *bg* in the two images, (d) the feature bar chart for the object point *obj* in the two images, (e) the feature bar chart for the shadow point *sh* in the two images. 86

4.8	Background modelling using neuronal networks; (a) an image patch of 2×3 , (b) the enlarged image by 3×3 , and (c) weight vector updating for the best match pixel b_9 and its neighbouring pixels. These pixels are highlighted inside the red square.	88
5.1	Overview of the proposed shadow detection method.	97
5.2	Comparison of intensity means of four segments taken from a current frame and its background image; (a) current frame masked with CDM and divided into segments, (b) the corresponding backgrounds of (a), (c) intensity means of a bright object segment (OL) and a dark background segment, (d) intensity means of a shadow-like object segment (SL) and a background segment having two different reflection properties, (e) intensity means of two dark segments (DS) taken from two different surfaces, and (f) intensity means of a shadowed segment (SL) and a background segment with high reflectance properties.	100
5.3	Examples of MTM matching results for a number of frames and their backgrounds. Best match segments are highlighted in green and most distant segments in red.	106
5.4	Comparison of popular local texture descriptors, $CLBP_{P=8}^{\tau=0}$, $LTP_{P=8}^{\tau=5}$, $SILLTP_{P=8}^{\tau=0.1}$, $LDP_{P=8}^{\tau=0}$, and the proposed $BPLCC_{P=8}^{\tau=5}$, under: (a) local illumination change, and (b) non-textured surfaces.	109
5.5	Illustration of proposed inter-channel spatial structure feature descriptor. (top) a layer with 3×3 neighbouring pixels is shown for each channel (Red, Green, and Blue) and (bottom) accumulated histogram for all pixels in an image segment with 32-dimensions. .	112

5.6	Comparison of various feature descriptors for two patches under the test, one occluded by foreground object (highlighted in red in the current frame (a1) and blue in the background image (b1)), and the other by shadow (highlighted in green in the current frame and in yellow in the background image). Euclidean distances are computed and shown for each method. (a2)-(b2) $LBP_{8,1}$, (a3)-(b3) $LTP_{8,1}^5$, (a4)-(b4) $SILTP_{8,1}^{0,1}$, and (a5)-(b5) the proposed $BPLCC$	114
5.7	Luminance ratio analysis for a line scanned along light direction. .	117
5.8	The process of proposed LGO feature extraction; (a) an estimation of angular direction of the light source L , (b) a magnified section of image segment S highlighted in yellow with different gradient orientations (white arrows) and matching angle θ , (c) 8-radial section with a starting position of histogram, and (d) the LGO histogram of S	118
5.9	An example of segment-level classification for a test image segment occluded by shadow in the current frame.	126
5.10	An example of the process of region-level classification. (a) a test image segment S_t surrounded by a two object segments S_o , three shadow segments S_{sh} , and two dark segments S_d , (b) the classification result after first iteration, and (c) the classification result after second iteration.	129

6.1	Shadow detection rate and shadow discrimination rate versus various parameter settings; (top) Shadow detection rate and shadow discrimination rate versus various color constancy threshold TH_{cc} , (bottom) shadow detection rate and shadow discrimination rate versus various number of radial sections Ω	143
6.2	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>Campus</i> . Fore-ground objects and shadows are highlighted in red and green, respectively.	148
6.3	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>Cubicle</i>	150
6.4	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>WBS</i>	151
6.5	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>PPL</i>	153
6.6	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>Seam</i>	154
6.7	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>HwyI</i>	156

6.8	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>HwyIII</i>	157
6.9	Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence <i>Bungalows</i>	158
6.10	Qualitative shadow detection results using different features for a frame taken from <i>Campus</i> , <i>Cubicle</i> , <i>WBS</i> , and <i>PPL</i> . Foreground objects are marked in red and shadows are marked in green. Results are best viewed in color.	166
6.11	Qualitative shadow detection results using different features for a frame taken from <i>Seam</i> , <i>HwyI</i> , <i>HwyIII</i> , and <i>Bungalows</i>	167
6.12	The performance of popular features and the proposed features versus shadow detection rate, shadow discrimination rate, and accuracy.	170
6.13	Qualitative moving shadow detection results of all compared methods for a frame taken from the first four datasets in Table 6.1. . .	172
6.14	Qualitative moving shadow detection results of all compared methods for a frame taken from the last four datasets in Table 6.1. . .	173
6.15	Qualitative moving shadow detection results of the proposed method.	177
6.16	Qualitative moving shadow detection results of the proposed method.	178

6.17	The pixel-based classification accuracy, in terms of (a) Shadow detection rate (η %) and (b) Shadow discrimination rate (ξ %), of AP [138], JH [139], LCC [62], SL [134], SVM [131], and the proposed moving shadow detection with respect to all sequences provided in Table 6.1.	179
6.18	The pixel-based classification accuracy, in terms of (a) Shadow detection rate (η %) and (b) Shadow discrimination rate (ξ %), of AP [138], JH [139], LCC [62], SL [134], SVM [131], and the proposed moving shadow detection with respect to shadow problematic situations.	181
6.19	Stability of shadow detection methods using a number of samples from the sequences; (a) <i>Campus</i> , (b) <i>PPL</i> , and (c) <i>HwyI</i>	183
A.1	Formation of the slice transform SLT matrix for a small segment with 4 gray values. The column vector V is obtained by scanning the image segment S row-by-row. The column vector is then mapped into a collection of 4 binary slices, each represents one gray-level (one bin). These binary slices are then used to form the SLT matrix with a dimension of 11 samples and 4 features (bins).	217
A.2	Construction steps of an image segment W' using the SLT matrix of S and a different weight vector β	218
B.1	Other common sequences used to evaluate moving shadow detection methods. The sequences <i>backdoor</i> , <i>busStation</i> , <i>highway</i> , and <i>PETS2006</i> can be downloaded from [25] and the sequences <i>hallway</i> , <i>highwayII</i> , <i>laboratory</i> , and <i>intelligent room</i> can be downloaded from [26].	220

List of Tables

2.1	Summary of shape-based methods for shadow detection.	20
2.2	Summary of light direction-based methods for shadow detection.	22
2.3	Summary of color-based methods for shadow detection.	25
2.4	Summary of texture-based methods for shadow detection.	28
2.5	Summary of gradient-based methods for shadow detection.	32
2.6	Summary of segmentation-based methods for shadow detection.	35
2.7	Summary of block-based methods for shadow detection.	37
2.8	Quantitative comparison of the methods (- means that the method is either not tested or not affected by the listed problematic situation).	39
3.1	Summary of dataset classification methods	62
4.1	Similarity between two patterns for possible conditions.	74
5.1	Possible conditions associated with region growing.	128
6.1	Technical details of used sequences.	135
6.2	Methods quantitative performance for the sequence <i>Campus</i>	148

6.3	Methods quantitative performance for the sequence <i>Cubicle</i>	150
6.4	Methods quantitative performance for the sequence <i>WBS</i>	151
6.5	Methods quantitative performance for the sequence <i>PPL</i>	153
6.6	Methods quantitative performance for the sequence <i>Seam</i>	154
6.7	Methods quantitative performance for the sequence <i>HwyI</i>	156
6.8	Methods quantitative performance for the sequence <i>HwyIII</i>	157
6.9	Methods quantitative performance for the sequence <i>Bungalows</i> . .	158
6.10	Overall accuracy performance and execution time for the compre- hensive methods.	160
6.11	Average quantitative performance for all shadow detection meth- ods being compared in this experiment.	180

List of Algorithms

5.1	Candidate shadow segments	99
5.2	MTM matching and dictionary sample collection	105
5.3	The proposed segment feature extraction method.	122
5.4	Region-level classification	130

Abbreviations

BG	Background image
BPLCC	Binary Patterns of Local Colour Constancy
CDM	Change Detection Mask
CF	Current Frame
CLPB	Colour Local Binary Pattern
CSR	Candidate Shadow Regions
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DS	Dark Segment
DSM	Dark Segment Mask
ED	Euclidean Distance
FAR	False Alarm Rate
FN	False Negative
FPR	False Positive Rate
GIC	Global Intensity Consistency
GICC	Global Intensity Consistency Classifier
GSCN	Gradient-Space-Connected Neighbourhoods
GT images	Ground-Truth images
H	Hausdorff distance
HOG	Histogram of Oriented Gradients
HSI	Hue, Saturation, Intensity colour space
HSV	Hue, Saturation, Value colour space
IH	Intensity Histogram
LBP	Local Binary Pattern
LDP	Local Derivative Pattern

LGHP	Local Gradient Hexa-Pattern
LGO	Light-based Gradient Orientation
LTP	Local Ternary Patterns
MBG	Masked Background image
MCF	Masked Current Frame
MAP	Maximum A Posteriori probability
MRF	Markov Random Field
MSR	Multi-independent Sparse Representation
MTM	Matching by Tone Mapping
NED	Normalized Euclidean Distance
Nrgb	Normalized red, green, blue
OL	Object-Like segment
OLM	Object-Like Mask
OMP	Orthogonal Matching Pursuit
PCA	Principal Component Analysis
PR	Precision Rate
RGB	Red, Green, Blue colour space
ROC graph	Receiver Operating Characteristic graph
ROI	Region Of Interest
SILTP	Scale Invariant Local Ternary Patterns
SL	Shadow-Like segment
SLM	Shadow-Like Mask
SLT	Slice Transform
SOBS	Self-Organization approach to Background Subtraction

SPD	Spectral Power Distribution
SVD	Singular Value Decomposition
SVMs	Support Vector Machines
TP	True Positives

Chapter 1

Introduction

1.1 Background

Shadows play an important role in our understanding of the world and provide rich visual information about the properties of objects, scenes and lights. The human vision system is capable of recognizing and extracting shadows from complex scenes and uses shadow information to automatically perform various tasks, such as perception of the position, size and shape of the objects and understanding the structure of 3D scene geometry and the location and intensity of light sources.

In past decades, researchers working in computer vision and related fields have been trying to develop a mechanism for machines to mimic the human vision system in handling visual data and performing associated tasks. However, the problem is far from being solved, and all the tasks remain challenging.

Shadows are involved in many low-level computer vision applications and image processing tasks, such as shadow detection, removal, extraction, correc-

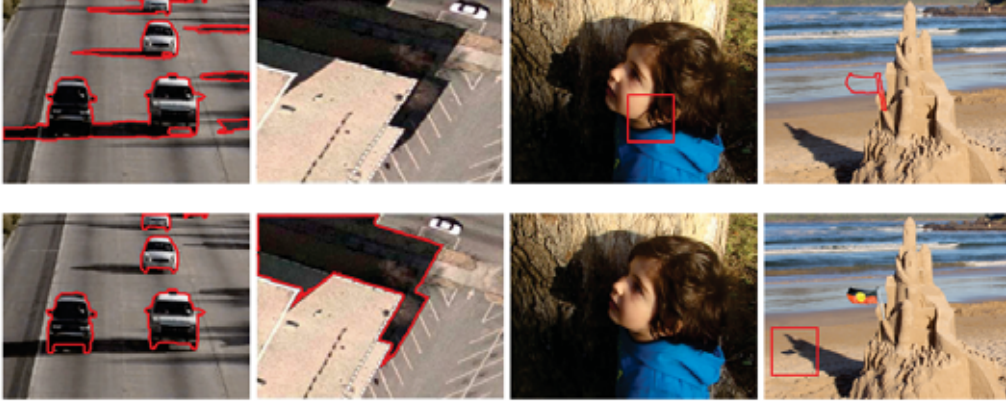


Figure 1.1: Examples of shadow applications; (first column) detection of moving shadows in a frame of the sequence *HwyI* using the method reported in [1], (second column) manual detection of shadows of a building in still image, (third column) manual detection and correction of shadows for a still image, and (fourth column) manual detection and mapping of the shadow for an outdoor still image.

tion and mapping. In many video applications, shadows must be detected and removed for the purpose of object tracking [2], classification [3][4], size and position estimation [5], behavior recognition [6] and structural health monitoring [7]. In a still image processing, shadow feature extraction is applied to obtain features that will be useful in object shape estimation [8], 3D object extraction [9], building detection [10], illumination estimation [11] and direction [12], and camera parameter estimation [13]. Shadow detection and correction (i.e., shadow compensation or deshadowing) involves complex image processing techniques to produce a shadow-free image that can be useful in many applications, including reconstruction of surfaces [14], illumination correction [15] and face detection and recognition [16]. In contrast to shadow detection, some applications, such as rendering soft shadows for 3D objects [17] and creating shadow mattes in cel animation [18], require rendering shadows to add additional spatial detail within and among objects and to produce images with a natural realistic look. Shadow detection and mapping must also be considered in some recent image processing

applications, such as PatchNet [19], Timeline editing [20] and many other visual media processing applications [21]. Examples of these applications are shown in Figure 1.1.

1.2 Existing Shadow Detection Methods

The detection of moving shadows in videos has been the subject of considerable research, discussion and debate among the computer vision research community. Many algorithms have been developed to address these two problematic situations by utilizing either color information or taking advantage of geometric information of the foreground and/or shadow orientations. Recently, Sanin et al. [22] divided the existing cast shadow detection methods into four main categories: geometry-based, chromaticity-based, physical-property-based and texture-based methods. This classification is based on the primary features that are used to identify shaded points. As stated in [22], this type of classification results in a high quality influence on the shadow detection results compared to the selection of algorithms [23].

In this thesis, the existing methods are divided into two main categories: object-shape-property-based and shadow-property-based methods. Shadow-property-based methods are further subdivided into light-direction-based and image-feature-based methods.

Light-direction-based methods analyze geometric information, such as location and direction, of cast shadows and the light source to find useful geometric features to detect possible shadows in the background.

By contrast, image-feature-based methods analyze 2D images to extract various image features, such as color, texture, and edges, independent of the

scene type, object type, and other geometric features. Image-feature-based methods are further subdivided into pixel-based and region-based methods with respect to the spatial features used in their final classification.

Regardless of the type of image features, such as color, edge, and texture, used in the analysis stage, the final classification in both cases is performed on individual pixels. The three image-feature-based methods, i.e., color-based, gradient-based, and texture-based methods, are studied separately.

Region-level methods effectively utilize contextual information and segment images into regions. These methods can be broadly subdivided into segmentation-based methods and block-based methods.

In the following, shape-based, light-direction-based, color-based, gradient-based, texture-based, segmentation-based, and block-based methods are discussed briefly along with their advantages and disadvantages. These methods are discussed in detail in Chapter 2.

Shape-based methods assess the geometric features, such as shape and size, of the foreground moving objects to eliminate their cast shadows. These methods are designed to detect shadows in specific applications and can provide accurate results when all the geometric features, along with their assumptions, are valid. In addition, these methods can be used to detect shadows of foreground objects in still images as such methods do not depend on the background. However, the applications of these methods are limited, and they may fail when the geometric relationships are changed.

Light-direction-based methods rely heavily on the shadow/light direction to detect shadows. These methods can provide good results when a strong single light source is present in the scene. However, these methods are not reliable in other situations, especially when multiple light sources exist in the scene.

Color-based methods utilize color information to define the change in intensity and appearance of a pixel when a shadow occurs. These methods usually perform well for simple achromatic shadows and smooth surfaces; however, they may fail in other problematic shadow situations and suffer from image pixel noise.

Texture-based methods explore the texture formation of the regions under the cast shadow and compare these regions to their corresponding backgrounds. These methods are the best choice among pixel-level analysis methods for detecting shadows of moving objects. Additionally, such methods provide reasonable results in situations with chromatic shadows and camouflage.

Gradient-based methods identify background patches using gradient and/or edge information. Gradient and/or edge information can be useful when a pixel in the current frame has similar brightness or intensity values to that in the corresponding background. These methods can provide good shadow detection results in situations with chromatic shadows; however, they may fail in other situations.

Segmentation-based methods perform classification on the region level. These regions are formed using the similarity measurements among various image features, including intensity, color, and texture. In general, the performance of these methods is better than that of other methods for all possible scenarios in a scene.

Block-based methods perform classification on independent image blocks. For each block, features are extracted in the current frame and compared to the corresponding block in the background image. Block-based methods perform better than pixel-level methods in scenes with nontextured and dark surfaces. These methods can be used to detect shadows in low levels of illumination.

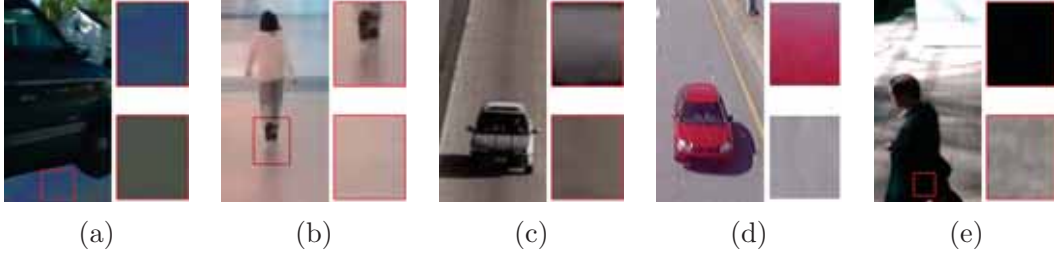


Figure 1.2: Examples of problematic situations for shadow detection; (a) chromatic shadow (cast shadow and background surface are shown in the normalized **rgb** colour space for better comparison), (b) shadow colour-blending (parts of the moving person are reflected by the background surface), (c) foreground-background camouflage, (d) non-textured surfaces, and (e) dark surfaces. A frame is shown on the left where regions with the problem are highlighted by a red rectangle. A magnified section in the given frame is shown on the top-right corner, and the same section in the background surface is shown on the bottom-right corner.

1.3 Problem Statement

Shadows are problematic because they cause local intensity/color changes on background surfaces. When a 3D object is placed between a direct light source and the background surface, the object partially or totally blocks light from reaching adjacent regions of the background surface, which causes a change in illumination in that region. The darker region of the shadow, where the direct light is completely blocked and only ambient light is available, is called the umbra. The lighter region of the shadow, where both ambient light and part of the light are available for illumination, is known as the penumbra.

Detecting shadows of moving objects using low-resolution surveillance cameras is challenging because of problematic situations associated with shadows, including chromatic shadows, shadow color blending, foreground-background camouflage, nontextured surfaces and dark surfaces, as shown in Figure 1.2. The first two problems relate to regions of shadows that can be easily misclassified as moving objects. The latter three problems relate to regions of objects that can be misclassified as shadows. Each problematic situation is addressed briefly

in the following.

- A chromatic shadow occurs when parts of the shadow in the umbra are illuminated by colored light (mainly blue for outdoor environments), which is different from direct light and causes color bleeding to the underlying region.
- Color blending is another problem in shadow detection where the reflectance of the background surface is high, which causes some parts of the foreground object to be reflected by the background.
- The problem of camouflage arises when parts of the foreground object have similar intensity and color to their corresponding backgrounds.
- Nontextured surfaces, where texture information is not available, are another common problem in moving shadow detection.
- Dark surfaces, where some parts of the foreground have low reflection properties, are a major problem in many computer vision tasks, including shadow detection. As a result, these foreground parts are often misclassified as parts of shadows.

1.4 Research Aim

The **aim** of this research effort is to develop an efficient method to solve these environmental problems in shadow detection while simultaneously improving the overall accuracy and detection stability. To measure the effectiveness, the proposed methods, including the proposed feature descriptors, will be assessed in terms of common quantitative metrics, including shadow detection rate, shadow discrimination rate, accuracy, and stability. A large number of ground truth

images have been either manually segmented or obtained from different resources to make this comparison possible (e.g., [24, 25, 26, 27]).

In the following, the four common quantitative metrics are defined as they will be used to evaluate moving shadow detection methods in Chapter 2.

Shadow detection rate (η) refers to the percentage of shadow points correctly detected and classified as shadows by a method. This metric assesses the robustness of a method under the presence of two related problematic situations, namely, chromatic shadow and shadow color blending.

Shadow discrimination rate (ξ) indicates the percentage of object points successfully detected as objects. This metric investigates problems associated with object parts, in particular, the problems of foreground-background camouflage, nontextured object surfaces, and dark surfaces.

Accuracy evaluates the overall performance of a moving shadow detection method in detecting moving foreground objects. This metric is used to test the method under all the problematic situations discussed earlier.

Stability determines whether the method is stable in producing a constant detection result over time. The difference between the maximum and minimum accuracy for a number of individual frames is used as a measure of stability. This metric is used to evaluate the performance of only the proposed moving shadow detection method.

1.5 Scope of the Research Study

The scope of this thesis is summarized below.

- This research addresses moving shadow detection and removal in image sequences for both outdoor and indoor environments.
- The research focuses on studying various effects of cast shadows and reflections from images obtained by static low-angle cameras. Detecting moving shadows from top-down view images (sometimes called aerial images) is considered a different study field and is not considered in this study.
- Detecting static shadows or those shadows that are caused by objects of no interest (such as shadows from waving trees or clouds) is beyond the scope of this study.
- Due to its importance, detecting changes in background scenes is a subject of this thesis. A background subtraction method is developed to obtain change detection masks for most of the sequences used in this thesis. A review of change detection methods is not considered to be a main part of this thesis; thus, the performance of the proposed change detection algorithm will be evaluated only as part of the complete framework proposed for shadow detection in this thesis.

1.6 Method Outline

In this thesis, a new feature-based image patch approximation and multi-independent sparse representation technique is presented to address the environmental problems listed in Section 1.3. The method has been designed to detect shadows of various moving objects in outdoor and indoor environments.

Figure 1.3 shows a general block diagram for the proposed moving shadow detection method developed in this thesis. The method consists of two main stages, namely, change detection and shadow detection. In the first stage, the

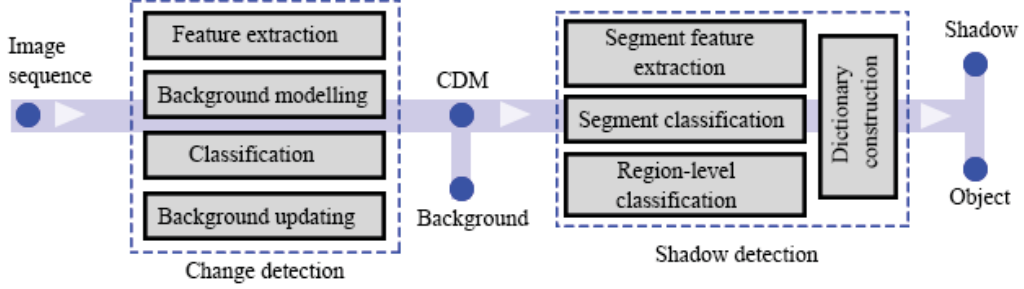


Figure 1.3: Block diagram of proposed moving shadow detection.

moving object is detected using a new background subtraction technique. A new set of pixel-level feature descriptors, global intensity consistency combined with normalized hue, saturation, value color space **HSV**, is proposed to successfully cope with global illumination changes in the background scene. The classification is first made on the pixel level using Euclidean distance measurements. Region-level classification is then performed using the total number of foreground pixels detected from the pixel-level classification. The output from the first stage is a binary image, called the change detection mask (**CDM**), that contains the moving objects and their cast shadows.

The second stage consists of two phases: training and running. In the training phase, two overcomplete reference dictionaries, one for objects and the other for shadows, are constructed using sample patches taken from four different sequences. In the running phase, the current frame and the background image are divided into nonuniform segments. Then, a set of three features, binary patterns of local color constancy (**BPLCC**), light-based gradient orientation (**LGO**), and intensity reduction histogram (**IRH**), are extracted for each segment. Given a new image segment, the best approximation for a number of iterations is found from each dictionary. For each iteration, an independent class assignment is performed by finding its distances from the reference dictionaries. The patch is then assigned to a class based on its probability of occurrence.

1.7 Contributions

A number of contributions have been made throughout this research, as summarized below:

- Problematic situations within shadow detection are addressed, and a new way of analyzing these problems is described in detail.
- A new taxonomy of moving shadow detection, which includes a review of published moving shadow detection methods, is introduced.
- Two new illumination-invariant features, namely, **BPLCC** and **LGO**, are proposed to efficiently address problematic situations associated with shadows.
- A generalized model of the standard sparse representation-based classifier, namely, the multi-independent sparse representation-based classifier (**MSR**), is proposed to deliver confident class assignments to the segments.
- A novel method for moving shadow detection, which is capable of accurately distinguishing two classes, objects (people/vehicles) and shadows under possible scenarios occurring in real video sequences, is developed.
- An extensive qualitative and quantitative evaluation is performed to prove the capability of the proposed framework in addressing the identified problematic situations associated with shadows. In addition, the proposed feature descriptors are evaluated and compared with state-of-the-art features.

1.8 Thesis Outline

In this study, the proposed research is presented in a thesis consisting of seven chapters, as outlined below.

Chapter 2 presents a literature review of the various methods for moving shadow detection and briefly discusses the different types of change detection algorithms. The shadow detection methods are analyzed and tested to determine their advantages, disadvantages, limitations and assumptions. The main drawbacks of the existing methods are outlined at the end of this chapter.

Chapter 3 discusses various problematic situations for moving shadow detection. These problematic situations are analyzed in detail, and examples are given. Methods for classifying datasets with respect to these problematic situations are discussed at the end of this chapter.

Chapter 4 presents the proposed change detection and background modeling method. The method is described in detail, and the underlying assumptions and limitations are discussed.

Chapter 5 introduces a new general-purpose moving shadow detection method. The method can be used to distinguish moving objects, including humans and vehicles, from their shadows. In this chapter, details of the two image feature descriptors, **BPLCC** and **LGO**, are provided. In addition, a proposed classification technique is discussed.

Chapter 6 evaluates the experimental results of the proposed shadow detection framework. In the first part of this chapter, the shadow detection datasets used for evaluation, the parameter settings, and the evaluation metrics are discussed in detail. In the second part, the results from the proposed framework and the proposed feature descriptors are compared with the results of current state-of-

the-art algorithms.

Chapter 7 includes a summary, conclusions and recommendations of further study.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a detailed background review of the existing moving shadow detection methods and their assumptions and limitations. In addition, these methods are classified based on properties used for classification and are evaluated with respect to possible problematic situations in complex scenes. These methods are discussed in Section 2.2.

2.2 Existing Shadow Detection Methods

Many shadow detection and removal algorithms, in which different techniques are used to accurately extract the foreground object from its shadow, have been proposed in the literature. Prati et al. [23] surveyed shadow detection methods based mainly on the type of algorithm used. They organized the contributions reported in the literature into four main classes, namely, parametric, nonparametric, model-based and non-model based. Al-Najdawi et al. [28] proposed

a four-layer taxonomy survey complementary to that in [23]. The survey is based mainly on the object/environment dependency and the implementation domain of the algorithms. Sanin et al. [22] stated that the selection of features has a strong influence on the shadow detection results compared to the selection of algorithms. Thus, they classified shadow detection methods into four main categories: chromaticity-based, geometry-based, physical-property-based and texture-based methods.

This research study introduces a different systematic method to classify existing shadow detection algorithms based on the type of properties used for classification. Since the properties of the two main components in the CDM, i.e., the moving object and the cast shadow, have important features that separate them, the existing methods can be divided into two main categories (Figure 2.1): object-shape-property-based and shadow-property-based methods. Shadow-property-based methods can be further subdivided into two groups depending on the main type of features used: light-direction-based and image-feature-based methods. Light-direction-based methods use the geometric formation of cast shadows and the light source to determine useful geometric features, such as the location and direction of the light source and the location of the shadow cast in the background. Meanwhile, image-feature-based methods directly analyze 2D images and extract color and texture independent of the scene type, object type or other geometric features.

Because the vast majority of the work belongs to this category, the image-feature based methods are further subdivided into pixel-based and region-based methods based on the spatial features used in their final classification. Regardless of the type of image features, such as color, gradient, edge, and texture, used in the analysis stage, the final classification in both cases is performed on individual pixels. Due to the importance of the three features, the color-based,

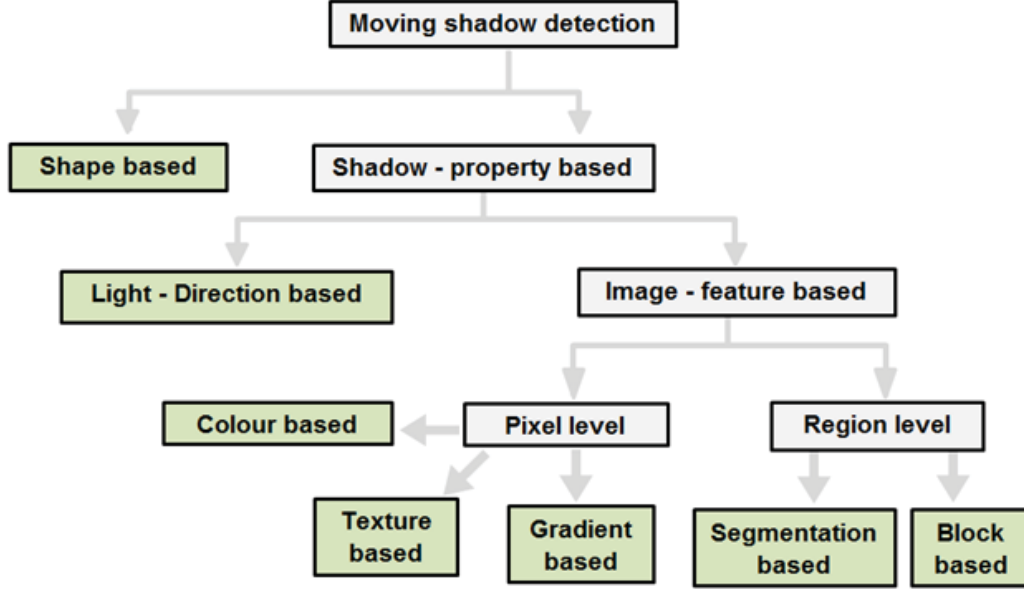


Figure 2.1: Classification of moving shadow detection.

gradient-based and texture-based methods are studied separately. Region-level methods effectively take advantage of contextual information to segment the image into regions. These methods can be further subdivided into segmentation-based methods and block-based methods.

When compared to other types of classification, the proposed classification provides a good understanding of the existing shadow detection methods by considering more features in the classification to cover more papers in the literature and by analyzing and evaluating the methods under all major problematic situations in shadow detection. In the following, the shape-based, light-direction-based, color-based, gradient-based, texture-based, segmentation-based and block-based methods are discussed in detail.

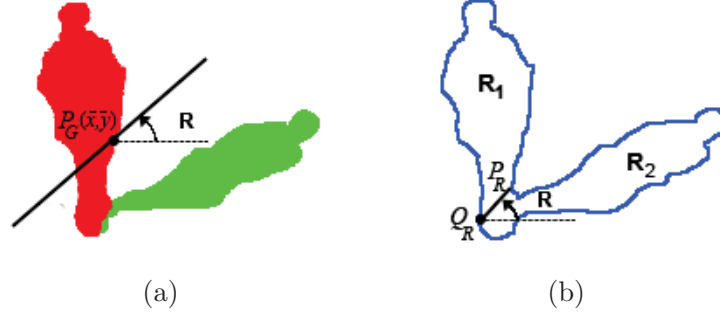


Figure 2.2: Illustration of shape-based moving shadow detection as reported in [31]; (a) the center of gravity and orientation of a shaded object R , and (b) object and its contour information.

2.2.1 Shape-based Methods

Shape-based methods utilize the properties of the foreground objects, such as shape and size, to detect cast shadows. These methods model foreground objects using various object-geometric features that can be obtained by having a priori knowledge about the foreground object or extracted from the input images without depending on the background reference. Such methods are designed to detect shadows cast by a specific foreground object, such as a human [29] or vehicle [30]. Typical shape-based methods are summarized in Table 2.1.

Hsieh et al. [31] proposed a coarse-to-fine Gaussian shadow algorithm to eliminate the shadows of pedestrians. As illustrated in Figure 2.2, several geometric features are utilized in this model, including the object orientation, center position of the shadow region and the intensity mean.

Yoneyama et al. [30] utilized vehicle-shadow orientations to distinguish shadows from moving vehicles. The method is based on a joint 2D vehicle/shadow model that is projected onto a 2D image plane. They explicitly divided the 2D vehicle/shadow model into six types, as shown in Figure 2.3. Each type refers to one location of the shadow in the foreground mask. The geometric properties

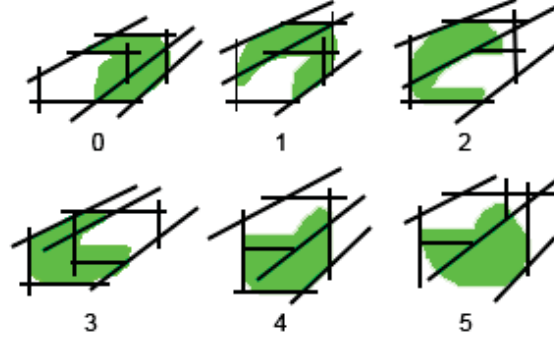


Figure 2.3: Illustration of shape-based moving shadow detection as reported in [33]. Six possible shadow locations are shown.

are then estimated from input frames without a priori knowledge of the light source and the camera calibration information.

Sheng et al. [32] introduced a shadow detection method based on human body geometrical existence and its approximate location. In the first step, the human body shape property is analyzed and used to determine the location of the cast shadow. In the second step, an image orientation information measure is used to classify the image pixels into smooth and edge regions. The two measurements, shape analysis and the ratio of pixels, are then fused in the final classification.

Fang et al. [35] exploited spectral and geometrical properties to detect shadows in video sequences. In the first stage, candidate shadow points are segmented using the spectral properties of shadows. Feature points of the occluding function are then detected using wavelet transform. In the last stage, the occluding line, formed by the feature points, is detected to separate objects from their shadows.

Chen et al. [29] proposed a 3-stage algorithm to detect cast shadows of pedestrians who are posed vertically. In the first stage, a support vector machine (**SVM**) classifier is trained and applied to the foreground mask to

Table 2.1: Summary of shape-based methods for shadow detection.

Paper	Year	Main feature	Other feature(s)	Results	
				Methods compared with	Summary
Hsieh et al. [31]	2003	Object orientation	Intensity histogram	[34]	Quantitative analysis are based on the results from 12 frames using Precision Rate (PR) and False-Alarm Rate (FAR) as follows: PR (average)=95.76% and FAR=1.76%
Yoneyama et al. [30]	2005	2D joint vehicle-shadow model	-	-	False-Alarm Rate is given for four different situations with an average of 2.17%
Sheng et al. [32]	2007	Ellipse-shape fitting	Intensity ratio	-	Some visual results are given
Fang et al. [35]	2008	1D Wavelet transform	Intensity reduction	-	Some visual results are given
Chen et al. [29]	2010	Log-polar coordinates	Colour, pix location, HOG transform	-	ROC graph is given for the proposed method with various features

compute possible shadow points. A linear classifier is then adopted to divide the foreground mask into human and shadow subregions. In the last stage, the shadow region is reconstructed with the aid of the background image.

2.2.2 Light-direction-based Methods

Some methods utilize geometric information, such as location and direction, of the light source(s) and shadows to detect shadows cast by moving objects. These geometric measures can be extracted from the input images or from prior information about the scene. These methods depend on geometric features and use other image features to enhance the detection results. Typical light-direction-

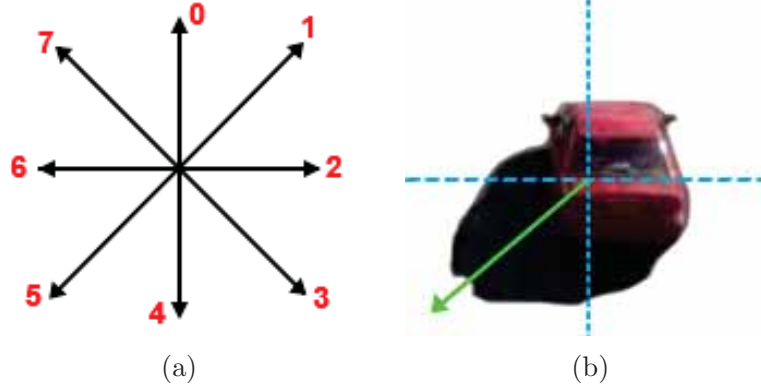


Figure 2.4: Illustration of light-direction based for moving shadow detection as reported in [36]; (a) eight directions of illumination, and (b) direction of illumination, the green arrow, is determined by the model.

based methods are summarized in Table 2.2.

Nicolas et al. [37] stated that estimating the position of a light source can improve the detection results. Thus, they proposed a method that enables joint estimation of the light source projection on the image plane and segmentation of moving cast shadows in natural video sequences. The light source position is estimated by exploring the geometric relations between the light source and the object/shadow regions on the 2D image plane. For each incoming frame, the shadow-foreground discrimination is performed based on the estimation of the light source position and the video object contours. This method is based on two assumptions: (i) the light source is unique and (ii) the surface of the background is flat.

Wang et al. [36] presented a method for detecting and removing shadows that is based on the detection of the cast shadow direction. In the method, the shadow direction is computed using a number of sampling points taken from shadow candidature. An edge map is then used to isolate the foreground object from its shadows. The researchers applied rules to recover parts of the vehicles that are (i) darker than their corresponding backgrounds and (ii) located in the

Table 2.2: Summary of light direction-based methods for shadow detection.

Paper	Year	Feature used for light direction estimation	Other feature(s)	Results	
				Methods compared with	Summary
Wang et al. [36]	2004	Boundary pixels	Intensity reduction, edge information	-	No quantitative results are given
Nicolas et al. [37]	2006	Geometric information	Intensity reduction, object contours	-	Shadow detection rate (η), False Positive Rate (FPR) and Shadow discrimination Rate (ξ) are given for the method in three situation: Light source estimation ($\eta=91.62\%$, FPR =6.32%, $\xi=2.12\%$), no light source estimation ($\eta=90.82\%$, FPR =8.55%, $\xi=2.12\%$) and no modification of initial segmentation ($\eta=81.25\%$, FPR =6.5%, ξ =not given)
Meher et al. [4]	2013	Principal component analysis (PCA)	Image segmentation	[38]	The quantitative comparison is performed using Average Accuracy measurement with 90.2% for the method and 90.52% (for three datasets). No qualitative results are given for comparison

self-shadow regions. This method is illustrated in Figure 2.4.

Meher et al. [4] used light source direction estimation to detect the cast shadow of a vehicle for classification purposes. In the first step, image segmentation is performed on the moving regions via a mean-shift algorithm. Principal component analysis (**PCA**) is then used to determine the direction of shadow region movement and to separate shadow regions from vehicle regions.

2.2.3 Color-based Methods

Color-based methods use color information to describe the change in the value and appearance of a pixel when a shadow occurs. In these methods, two fea-

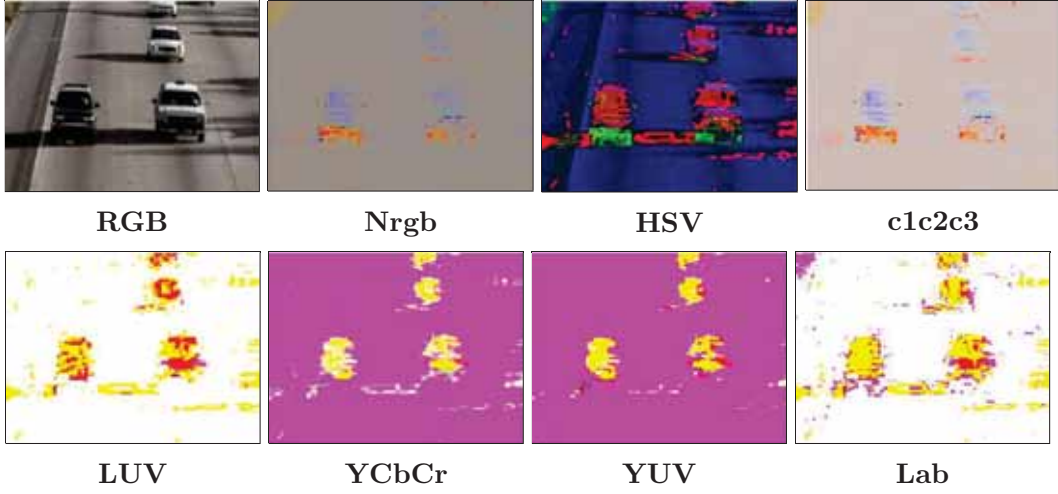


Figure 2.5: Different colour spaces used for moving shadow detection.

tures, namely, the intensity and invariant measures, are combined to identify those points that become darker than their corresponding background while maintaining their color consistency. Algorithms based on color techniques attempt to use suitable color spaces for separating the brightness of each pixel from its chromaticity. Examples of common color spaces used for moving shadow detection are shown in Figure 2.5. Comparative surveys on different color spaces used for shadow detection can be found in [39], [40] and [41]. Table 2.3 shows the common color spaces used in shadow detection algorithms. The formulas used for calculating the two terms, the shadow detection rate (η) and the shadow discrimination rate (ξ), are given in Chapter 6.

Cucchiara et al. [42] introduced the **HSV** color space as a good choice for shadow detection compared to the **RGB** color space. Their method is based on the observation that shadows lower the pixel's value (**V**) and saturation (**S**) but barely change its hue component (**H**).

Guan [43] proposed a shadow detection method for color video sequences using multiscale wavelet transforms and temporal motion analysis. The method

exploits the **HSV** color space instead of introducing complex color models.

Similarly, Salvador et al. [44] adopted a new color space model, **c1c2c3**, for detecting shadow points in both still images and videos. In the method, the features of the invariant color **c1c2c3** of each candidate point from a predefined set are compared with the features of the reference point. Thus, a candidate point is labeled as shadow if the value of its **c1c2c3** has not changed with respect to the reference.

Melli et al. [45] proposed a shadow-vehicle discrimination method for traffic scenes. They asserted that the **YCbCr** color space is more suitable for shadow-foreground discrimination, namely, for separating the road surface from shadow regions.

Cavallaro et al. [46] used normalized rgb (**Nrgb**) color space to obtain shadow-free images. The main idea of using this color space is that the values of the normalized components (usually labeled as rgb) do not change substantially for points under local or global illumination changes.

Similar to **Nrgb**, normalized r-g (**nR-G**) was proposed by Kuo-Hua et al. [47] to separate brightness and color for each pixel in the foreground mask region. The researchers stated that the normalized values of the two channels (red and green) remain roughly the same under different illumination conditions.

Sun et al. [48] proposed a method for detecting the cast shadows of vehicles using combined color spaces. In the method, hue, saturation, intensity **HSI** and **c1c2c3** color spaces are used to detect possible shadow points. An approximate result is then obtained by synthesizing these two results. In the final step, morphological operations are applied to improve the accuracy of the detection result.

Ishida et al. [49] used the **UV** components of the **YUV** color space and

Table 2.3: Summary of color-based methods for shadow detection.

Paper	Year	Colour space	Results			
			Quantitative performance (%)		Methods compared with	Summary
			η	ξ		
Cucchiara et al. [42]	2003	HSV	-	-	-	False negatives and false positives are compared with ground-truth (GT) using ROC graphs.
Salvador et al. [44]	2004	c1c2c3	-	-	[50] [51] [52] [53]	Only one dataset is used for quantitative comparison via ROC graph and accuracy mean of 86.6 %, qualitative results are shown for 3 datasets.
Meli et al. [45]	2005	YCbCr	63.19	67.51	[51]	Four traffic datasets are used in the experiment, quantitative comparison are shown for two datasets.
Cavallaro et al. [46]	2005	Nrgb	-	-	[50] [51] [52] [53]	Only accuracy mean given (88.0 %) for two datasets along with qualitative results on 5 datasets.
Kuo-Hua et al. [47]	2006	nR-G	-	-	[54]	Qualitative comparison results are shown, quantitative comparison are shown using ROC graph
Guan [43]	2010	HSV	80.31	95.36	[50] [51] [52] [53]	Only one dataset is used for quantitative comparison, qualitative results are shown for 5 datasets
Sun et al. [48]	2010	HSI & c1c2c3	89.35	90.45	[55][52]	Qualitative results are shown for three frames.
Ishida et al. [56]	2013	YUV	91.56	95.68	[50] [51] [52] [53]	Qualitative results are shown for all datasets, only one video sequence is used for quantitative comparison.
Dai et al. [57]	2013	Multi-colour	89.82	95.37	[58] [59] [42] [44] [60]	Eight datasets are used for performance evaluation, qualitative and quantitative comparison results are shown.
Wang et al. [61]	2014	HSI	87.75	90.5	[62] [2] [63]	Qualitative and quantitative results are shown on 6 datasets.

the normalized vector distance, peripheral increment sign correlation and edge information to detect shadows from image sequences. They stated that the differences in the \mathbf{U} and \mathbf{V} components of each shaded pixel and the corresponding backgrounds are small, whereas the differences in a moving object region are large.

Dai et al. [57] introduced a method to detect shadows using multiple color spaces and multiscale images. Their color features include chromaticity difference in \mathbf{HSV} , invariant photometric color in $\mathbf{c1c2c3}$ and salient color information in \mathbf{RGB} .

Wang et al. [61] proposed a method for shadow detection using online subscene shadow modeling and object inner-edge analysis. In the method, accumulating histograms are computed by means of the chromaticity differences in hue, saturation and intensity \mathbf{HSI} between the foreground and background regions.

2.2.4 Texture-based Methods

Textures in the background do not change under varying illumination. Moreover, the foreground object produces different patterns and edges than that of its shadow or the corresponding background. Texture can provide helpful information to recognize background regions when shadows occur. This powerful image feature descriptor is illustrated in Figure 2.6, where two test patches, before and after a shadow occurs, are extracted and magnified for visualization. The intensity values of the center pixel p and 8 other pixels in each patch are shown to test the intensity consistency of the patch. The intensity value of the center pixel is larger than the intensity values of q_1, q_2, q_3, q_6, q_7 , and q_8 and is less than the intensity values of q_4 and q_5 , before and after a shadow occurs.

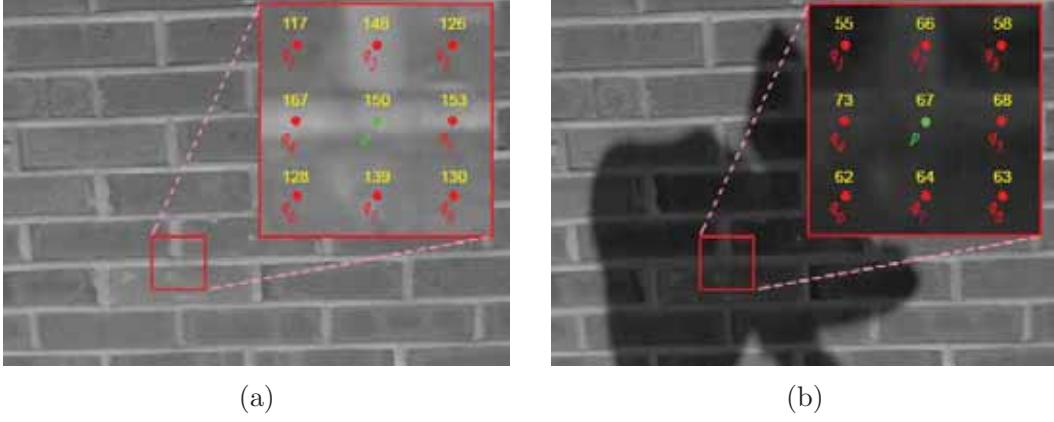


Figure 2.6: Illustration of texture feature information; (a) a small patch of the background image extracted and magnified before shadow occur, and (b) the patch is occluded by shadow at a different time.

This model of the intensity value comparison can be formulated using advanced feature extraction techniques. In this section, various texture-based methods are discussed. A summary of these methods is given in Table 2.4.

Leone et al. [64] presented an approach for shadow detection of moving objects in visual surveillance environments. Potential shadow points are detected based on adaptive background difference. The similarities between small textured patches are then measured using the Gabor function to improve the detection results.

Yang et al. [65] proposed a method to detect shaded points by exploiting color constancy among and within pixels, as well as temporal consistency between adjacent frames. The method has good performance compared to other pixel-based methods in which the interpixel relationship is used as an additional metric to support classification.

Qin et al. [66] proposed a shadow detection method using local texture descriptors called scale invariant local ternary patterns (**SILTP**). Texture and color features are learned and modeled via a mixture of Gaussian distributions.

Table 2.4: Summary of texture-based methods for shadow detection.

Paper	Year	Texture descriptor	Other feature(s)	Results			Summary
				Quantitative performance (%)		Methods compared with	
				η	ξ		
Leone et al. [64]	2007	2D Gabor filter	Intensity ratio	-	-	-	Some qualitative results are shown, quantitative comparison are performed on different dimensions of the Gabor filters.
Yang at al. [65]	2008	Colour constancy	Invariant colour	-	-	[54]	Qualitative results are shown for the three datasets, quantitative comparison are shown in ROC graphs.
Qin et al. [66]	2010	SILTP	Luminance ratio and angle variation	76.73	83.36	[67] [68] [69]	Performance comparisons are shown using quantitative analysis. No qualitative results are shown for the methods.
Yiyang et al. [70]	2010	Gradient confidence weight	Luminance reduction	92.34	84.55	[68] [69] [67] [71]	quantitative comparison tables are provided based on results obtained from 4 datasets.
Khare et al. [72]	2014	DWT	HSV differences	90.13	96.34	[2] [43] [42] [31] [58] [50] [73] [74]	Eight datasets are used for comparison and the results for multiple frames in each dataset are shown for qualitative comparison.
Dai at al. [75]	2015	LBP	Intensity ratio, colour distortion	79.54	88.59	[42] [44] [60] [58] [59] [48] [76]	The quantitative performances is based on final classification of the method without using morphological operations.
Huerta at al. [77]	2015	Gradient detection	Intensity reduction	53.00	92.00	[42] [31] [64] [2] [67] [69] [68] [70]	Quantitative results are shown for 8 datasets.

The contextual constraint from Markov random field (**MRF**) modeling is applied to obtain the maximum a posteriori (**MAP**) probability estimation of the cast shadows.

Yiyang et al. [70] proposed a texture-based method to detect shadow points in video. Potential shadow points are first detected using intensity reduction features. A gradient confidence weight is used to describe the texture formation within a window of 3×3 pixels (centered at the point under the test).

Khare et al. [72] used the discrete wavelet transform (**DWT**) to describe the texture information in horizontal and vertical image dimensions. The shadow points are detected through computing several wavelet decompositions in the **HSV** color space.

Local binary pattern (**LBP**) is used in [75] as a local texture descriptor for detecting shadows of surveillance scenarios. In addition to **LBP**, features such as the intensity ratio and color distortion are utilized in the statistical learning framework to enhance the detection results.

Huerta et al. [77] proposed a multistage texture-based approach to detect shadows in video. In the first stage, candidate shadow regions are formed by means of intensity reduction. Chromatic shadow detection is then performed using gradients and chrominance angles.

2.2.5 Gradient-based Methods

The gradient distribution is another useful image feature complementary to texture that is potentially very effective in detecting cast shadows in still images and videos since the structure of the background surface does not change under varying illumination. The gradient indicates how much the gray levels in an im-

age change in the horizontal and vertical directions. Based on that information, two gradient components can be extracted, the gradient magnitude $GM(p)$ and the gradient direction $GD(p)$, as computed below:

$$GM(p) = \sqrt{\Delta_X(p)^2 + \Delta_Y(p)^2} \quad (2.1)$$

$$GD(p) = \arctan \frac{\Delta_Y(p)}{\Delta_X(p)} \quad (2.2)$$

where Δ_X and Δ_Y are the differences in intensity between two pixels in the horizontal and vertical directions, respectively.

Gradient information has been used to design various edge detection algorithms for images, including the Prewitt edge operator, Sobel operator, and Canny operator. The two closely related feature descriptors, gradient and edges, are capable of identifying a foreground object pixel in the current frame that has a similar brightness or intensity value to that in the corresponding background. Figure 2.7 shows the edge detection results using the Prewitt edge operator, Sobel operator, and Canny operator. The gradient magnitudes and directions of the two images, obtained by applying Equations (2.1) and (2.2), are also shown.

In this section, a number of moving shadow detection methods that use gradient and edge features are discussed. These methods are summarized in Table 2.5.

Xu et al. [78] used static edge correlation and seed region formation to detect shadow regions for indoor sequences. The method implements various techniques, including (a) generation of the initial CDMs, (b) application of Canny edge detection on the given frame, (c) detection of moving edges using multiframe integration, and (d) the use of morphological dilation to enhance the

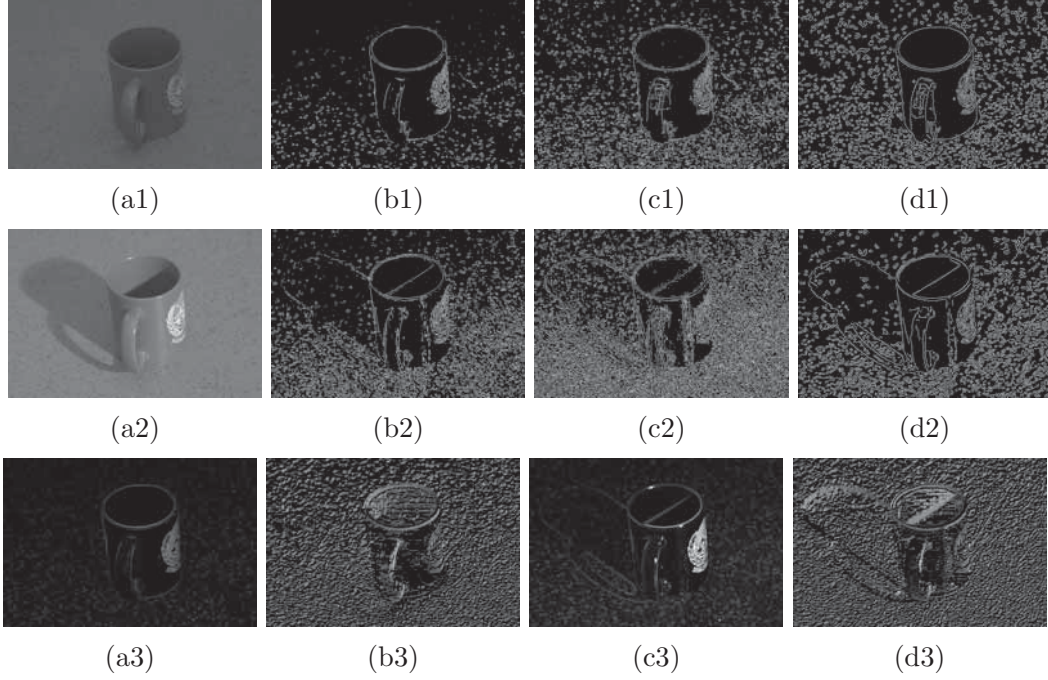


Figure 2.7: Visualization of gradient orientation and edges detection obtained using different operators; (a1 - a2) same image under two different illumination conditions, (b1 - b2) edge detection using Perwitt operator, (c1 - c2) edge detection using Sobel operator, (d1 - d2) edge detection using Canny operator, (a3 - b3) gradient magnitude and direction of image (a1), and (c3 - d3) gradient magnitude and direction of image (a2).

output results.

Zhang et al. [79] proved that the ratio edge is illumination invariant. In the first stage, the possible shadow points are modeled in a mask based on the intensity constraint and the physical properties of shadows. The ratio edge between the intensity of a pixel and its neighboring pixels is then computed for the given frame and the background image. In the final stage, geometric heuristics are imposed to improve the quality of the results.

Xiao et al. [80] used Sobel edge detection to eliminate the shadows of the moving vehicles. The Sobel edge detector is applied to the binary CDM (to detect the boundary of the whole mask) and the given frame masked with the

Table 2.5: Summary of gradient-based methods for shadow detection.

Paper	Year	Edge/ Gradient model	Other feature(s)	Results			
				Quantitative performance (%)		Methods compared with	Summary
				η	ξ		
Xu et al. [78]	2005	Canny operator	Morphological dilation	-	-	[81]	No quantitative results of the method are given.
Zhang et al. [79]	2007	Ratio edge	Intensity and geometry constrains	82.50	92.37	[58] [82] [50] [51] [52] [53]	ROC graphs are provided for further quantitative comparison.
Xiao et al. [80]	2007	Sobel operator	Spatial verifications	-	-	[83] [36]	No quantitative results of the method are given.
Joshi et al. [38]	2008	Gradient	Colour and Intensity	90.4	96.2	[23]	3 datasets are used. Classification accuracy graph is provided along with quantitative results.
Panicker et al. [84]	2010	Sobel operator	-	70.38	84.92	[50] [51] [52] [53]	No results are shown for qualitative comparison.
Huang et al. [85]	2011	SUSAN algorithm	Histogram analysis	-	-	-	No quantitative results of the method are given.
Elham et al. [86]	2013	Canny operator	DWT	86.25	85.45	[67] [31] [64] [42]	No results are shown for qualitative comparison.

CDM (to detect inner edges of the vehicles). The edges from the vehicles are then extracted from the two results. In the final step, spatial verifications are applied to reconstruct the vehicle's shape.

Similarly, Panicker et al. [84] proposed a method that uses edge information to detect moving shadows for traffic sequences. In the first stage, the edge information for both the foreground and the background masks is extracted by using the Sobel operator. The two edge maps are then correlated to eliminate the boundary of the cast shadow, thereby preserving the internal edge of the object. In the final stage, the object shape is reconstructed by applying horizontal

and vertical gradient operations.

Joshi et al. [38] proposed a semisupervised learning technique to solve the problem of achromatic and chromatic shadows. Their method can be divided into two phases: the feature phase and the training phase. In the feature phase, a set of useful features is extracted from the input image sequence by exploiting the pixel gradient distribution and color and intensity information. In the training phase, an **SVM** is trained from a set of manually labeled images. A co-training algorithm is employed with the **SVM** for shadow-foreground classification.

Huang et al. [85] proposed a simple edge-pixel statistic histogram analysis to detect and segment the shadow area for traffic sequences. The statistic characteristics of edge pixels, detected using SUSAN's algorithm [87], are analyzed to detect shadow pixels.

Elham et al. [86] introduced a shadow detection method base mainly on edge information. In the first step, static edges of the CDM are detected using the Canny operator. In the second step, a wavelet transform is applied to obtain a noise-free image followed by the watershed transform to segment different parts of the object, including shadows. The segmented parts are then marked as shadows or foreground based on the chromaticity of the background.

2.2.6 Segmentation-based Methods

Segmentation-based methods attempt to find similarity in intensity, color or texture among neighboring pixels to form independent regions. In general, these methods consist of two main stages: candidate shadow points and region correlations. Usually, the selection of candidate shadow points is performed on individual pixels by employing spectral features, such as intensity reduction [88], chromaticity [2], luminance ratio [62], and intensity-color [1]. These candidate

points often form one or more independent candidate shadow regions. In the next stage, region correlation is performed based on various measurements, including texture, intensity, and color. Typical segmentation-based methods are summarized in Table 2.6.

Javed et al. [88] proposed a five-stage algorithm for detecting shadow points in **RGB** color space. First, a shadow mask containing all the pixels with their intensity values reduced significantly is created. In the second stage, the vertical and horizontal gradients of each pixel are computed. Shadow candidate regions are then formed based on color segmentation. In the fourth stage, the gradient direction of each region in the current frame is correlated with that of the background. The classification is conducted in the final stage by comparing the results of the correlation with a predetermined threshold. Regions with a high gradient correlation are classified as shadows.

Toth et al. [54] proposed a shadow detection method based mainly on color and shading information. The foreground image is first divided into subregions using a mean-shift color segmentation algorithm. Then, a significance test is performed to classify each pixel as foreground or shadow. The final classification is then obtained based on whether the majority of the pixels inside each subregion, in the previous stage, are classified as shadows. The subregion is considered to be shadow if the total number of shaded points exceeds 50% of the total number of the pixels inside the subregion.

Sanin et al. [2] stated that selecting a larger region, which ideally contains all the shadow points, provides better texture information than selecting smaller regions. Therefore, chromaticity information is used to select possible shadow points. Connected components are then extracted to form candidate regions, and gradient information is computed to remove foreground regions that are incorrectly detected as shadows. Some assumptions are made in this method:

Table 2.6: Summary of segmentation-based methods for shadow detection.

Paper	Year	Segmentation type	Colour space -feature(s)	Results			
				Quantitative performance (%)		Methods compared with	Summary
				η	ξ		
Javed et al. [88]	2002	Mean-shift	RGB	-	-	-	Some qualitative results are shown.
Toth et al. [54]	2004	Mean-shift	LUV	-	-	-	Some qualitative results are shown.
Sanin et al. [2]	2010	Connected components	HSV	92.05	97.85	[88][89]	Visual results are shown for one dataset.
Amato et al. [62]	2011	GSCN	RGB Intensity ratio	84.00	89.66	[67][68]	Qualitative results are not given for other compared methods.
Russell et al. [1]	2013	Mean-shift	Nrgb	91.52	92.50	[44][65]	4 datasets are used for evaluation. A frame from each dataset is shown for qualitative comparison.

(i) the candidate shadow regions are assumed to be isolated from each other and do not have common boundaries, and (ii) each region contains either shadow points or foreground object points.

Amato et al. [62] proposed a method to detect moving achromatic and chromatic shadows. Their method is based on the fact that a local constancy exists for any pair of pixels belonging to the shadow region, while foreground pixels do not have this property. In the method, the intensities of the background pixels are divided by the intensity of the given frame in the **RGB** space, and regions are formed using the gradient-space-connected neighborhoods algorithm (**GSCN**). Gradient constancy is then applied to detect possible shadow regions. In the final stage, regions with low gradient constancy are considered to be shadows. Similar to other approaches, this method assumes that the foreground object has a different texture than that of the shadow region.

Russell et al. [1] used color segmentation to divide the detected moving

regions into sub-regions. Three features, namely, intensity, spatial color constancy and temporal color constancy, are used to distinguish shaded regions from objects. An initial clustering of the CDM is used to divide the mask into subregions; then, three quantities, i.e., intensity mean, invariant color constancy measurement and temporal color constancy measurement, are computed for each region. The initial classification is based on these three measurements, and the inter-region relationships among neighboring regions are used to enhance the final detection result.

2.2.7 Block-based Methods

In contrast to segmentation-based methods, regions in block-based methods are manually formed by fixed-equal-size blocks without relying on color or texture information. The color and texture information among the pixels and their corresponding backgrounds is exploited to determine whether the block is located under a cast shadow. Typical block-based methods are summarized in Table 2.7.

Zhang et al. [90] assumed that the normalized coefficients of the orthogonal transform of image blocks are illumination invariant. Based on this assumption, they used the normalized coefficients of five orthogonal transforms, namely, discrete Fourier transform (**DFT**), discrete cosine transform (**DCT**), singular value decomposition (**SVD**), Haar transform, and Hadamard transform, to distinguish between a moving object and its cast shadow. The intensity and geometry information is utilized to refine the detection results.

Song et al. [60] developed a shadow-region-based statistical nonparametric approach to construct a new model for shadow detection of all pixels in an image frame. The color ratio between the illuminated regions and the shaded regions is utilized as an index to establish the model for different shadow pixels.

Table 2.7: Summary of block-based methods for shadow detection.

Paper	Year	Block size/ No. of blocks	Techniques -feature(s) used	Results			
				Quantitative performance (%)		Methods compared with	Summary
				η	ξ		
Zhang et al. [90]	2006	8×8	Orthogonal transform	86.27	95.53	-	Quantitative comparison is performed using different orthogonal transforms including: DFT ($\eta=84.85\%$, $\xi=95.94\%$), DCT ($\eta=86.27\%$, $\xi=95.53\%$), SVD ($\eta=80.27\%$, $\xi=92.38\%$), Haar transform ($\eta=83.62\%$, $\xi=97.29\%$), and Hadamard transform ($\eta=84.46\%$, $\xi=96.18\%$). Best result is recorded for DCT .
Song et al. [60]	2007	Various	Colour ratio, boundary detection	77.50	72.20	[52] [42]	Quantitative comparison results are given, no qualitative comparison results are given.
Celik et al. [91]	2008	8 blocks	Brightness ratio histogram	72.87	91.42	[53] [52] [50] [51]	Two visual results from the method are shown, quantitative comparison results are given for some datasets.
Bullkich et al. [76]	2012	7×8	Tone mapping	-	-	[50] [92]	Precision rate (average) = 93.47%, Recall rate (average) = 93.20% and F-measure (average) = 93.35% are given.
Cogun et al. [93]	2013	8×8	Cepstrum analysis	77.65	88.40	[50] [51] [52] [53]	No visual results are given for comparison between the methods.
Dai et al. [94]	2015	Various	Affinity propagation	71.28	96.00	[42] [44] [60] [76]	Qualitative results from all methods are given using a frame in each dataset.

Celik et al. [91] divided images into 8 nonoverlapped homogeneous blocks. The brightness ratio histogram of each block is used to determine whether the block is part of the moving object or shadow.

Bullkich et al. [76] assumed that shadow pixels are associated with background pixels through a nonlinear tone mapping. A model of matching by tone mapping (**MTM**) is developed to evaluate the distances between suspected foreground and background pixels. Regions with low **MTM** distance are considered to be shadows.

In [93], the CDM is divided into 8×8 blocks, and the 2D cepstrum

is applied to check whether the current frame preserves the background texture and color. Pixel-based analysis is performed within each shaded block for further classification.

Dai et al. [94] used affinity propagation to detect moving cast shadows in videos. In the first stage, the foreground image is divided into nonoverlapping blocks, and the color information in the **HSV** color space is extracted from each block. Affinity propagation is then utilized to cluster foreground blocks adaptively, and subregions are generated after coarse segmentation. In the last stage, texture features from irregular subregions are extracted and compared with the corresponding backgrounds to detect those with similarity.

2.2.8 Drawbacks

Table 2.8 presents a quantitative performance evaluation in terms of the average shadow detection rate (η) or average shadow discrimination rate (ξ) for each class of method with respect to problematic shadow situations. These rates are calculated according to the stated results of the original publications. Clearly (as indicated in Table 2.8), the shadow detection rate is affected by the presence of achromatic shadow, chromatic shadow, foreground-shadow camouflage, shadow color blending and multiple shadows. Furthermore, problems of foreground-background camouflage, nontextured surfaces, dark surfaces and shadow affect the shadow discrimination rate.

The first two classes, shape-based methods and light-direction-based methods, rely mainly on the geometric relationships of the objects and shadows in a scene. These methods can provide accurate results when the geometric features and their assumptions are valid (maximum shadow detection rate (η)=92.3% and maximum shadow discrimination rate (ξ)=93.5% are reported for

the light-direction-based method in Table 2.8). However, these methods may fail when the geometric relationships change. Moreover, these methods impose strong geometric assumptions that limit their applicability to specific situations and they may require human interaction or prior knowledge about the scene and the moving objects.

Color-based methods provide a reasonably high shadow detection rate ($\eta = 86.5\%$) for indoor environments in which only achromatic shadows are present; however, color-based methods fail to recognize most shadow points in other problematic situations. For example, from Table 2.8, the average shadow detection rates (η) obtained for the color-based method are 63.1% and 71.9% in cases of chromatic shadows and background color blending. Further quantitative results for the shadow discrimination rate are obtained to examine the performance of the method in situations with other problems that are directly related to the foreground objects. Except for nontextured surfaces, color-based methods fail to cope with the problems of foreground-background camouflage ($\xi = 64.1\%$) and dark surfaces ($\xi = 61.2\%$). In general, color-based methods are easy to implement and are applicable in real-time applications due to their low computational complexity.

Gradient-based methods can provide better results than color-based methods in cases of chromatic shadows ($\eta = 73.6\%$) and foreground-background camouflage ($\xi = 78.9\%$). However, these methods are not suitable for other situations, such as dark surfaces ($\xi = 68.9\%$) and nontextured surfaces ($\xi = 70.1\%$).

Among the pixel-based methods, the performance of the texture-based methods is better than that of the color-based and edge-based methods in terms of the shadow detection rate and discrimination rate for all the problematic situations, except nontextured ($\xi = 69.8\%$) and dark surfaces ($\xi = 70.9\%$).

Region-based methods are designed to address noise, camouflage and dark surfaces by taking advantage of the spatial image features and forming independent regions. Region-based methods are computationally expensive and are generally not suitable for real-time applications. Compared to pixel-based methods, region-based methods provide good results in situations with achromatic shadow ($\eta=90.7\%$), nontextured surfaces ($\xi=85.4\%$), and dark surfaces ($\xi=74.2\%$).

Among all the methods, segmentation-based methods achieve the highest performance in all scenarios except achromatic shadows ($\eta=86.2\%$) and nontextured surfaces ($\xi=77.0\%$). The results illustrate the ability of these methods to handle the problems of chromatic shadows ($\eta=87.7\%$), foreground-background camouflage ($\xi=89.5\%$), shadow color blending ($\eta=84.2\%$), and dark surfaces ($\xi=77.9\%$).

In general, many of the abovementioned moving shadow detection methods use general-purpose image feature descriptors to detect shadows. These feature descriptors may be effective in distinguishing shadow points from the foreground object in specific problematic situations; however, the methods often fail to distinguish shadow points from foreground objects in other situations. In addition, many of these moving shadow detection methods require prior knowledge of the scene conditions and/or impose strong assumptions, which make the methods excessively restrictive in practice.

2.3 Summary

Various moving shadow detection methods are reviewed in this chapter. These methods are classified into several categories according to the type of properties used for final classification. The different techniques include shape-based,

light-direction based, color-based, texture-based, gradient-based, segmentation-based, and block-based methods. The performance of each category is evaluated under various environments and problematic situations associated with moving shadows.

The first two types of methods, shape-based and light-direction-based methods, impose strong geometric assumptions that make them applicable in only specific situations, such as foreground objects posed vertically and scenes with a single light source. These assumptions are not valid in many cases and may result in failed shadow-foreground discrimination.

The major weakness of the existing image-feature-based methods is that they do not consistently perform well in different scenarios. For instance, segmentation-based methods, with an average of $\eta=86\%$ and $\xi=81.4\%$, perform well compared to all other methods; however, they may fail to identify dark and nontextured object regions and to distinguish them from their backgrounds.

Chapter 3

Problematic Situations

Detecting shadows of moving objects is difficult due to several environmental problems related to shadow detection. These problems can cause significant degradation in overall performance. Therefore, analyzing image sequences under various environments and identifying these problems can help in developing effective algorithms to discriminate between objects and shadows.

This chapter addresses problematic situations present in real-world scenes and discusses their impacts on classification. These problems are discussed in further detail in Section 3.2. Examples are provided along with the data analysis to clarify the problems. In addition, the general concept of shadow formation and modeling and the properties of shadows are discussed in detail in Section 3.1.

3.1 Understanding Shadows

Understanding the physical formation of shadows is an essential key to solving many problems in the applications mentioned in the previous section. The basic

idea of shadow formation and modeling is discussed in Section 3.1.1, and the main properties and assumptions of moving shadows are summarized in Section 3.1.2.

3.1.1 Cast Shadow Model

Shadows are considered to be a case of local or regional illumination changes. In other words, when an object is placed between a light source and a background surface, the object blocks the light from reaching the adjacent region(s) of the foreground object, causing a change in illumination in that region. Due to the multilighting effects, changes in illumination (with respect to the background) are more substantial in the center regions of the shadow (areas linked to the self-shadow of the foreground object) than in the outer boundaries. In this case, shadows can be further classified into two regions, namely, umbra and penumbra. The umbra is the darker region of the cast shadow where direct light (dominant light) is completely blocked and ambient light illuminates the region. The penumbra is the lighter region where both light sources (dominant and ambient lights) illuminate the area. An example of cast shadow analysis is illustrated in Figure 3.1.

These parts of the detected foreground mask can be analyzed in terms of the illumination of the light sources and the surface reflections of these regions. Thus, Kubelka-Munk theory [95] can be used to express the intensity of each pixel $S^t(p)$ in an image plane obtained by a camera as:

$$S^t(p) = i^t(p)r^t(p) \quad (3.1)$$

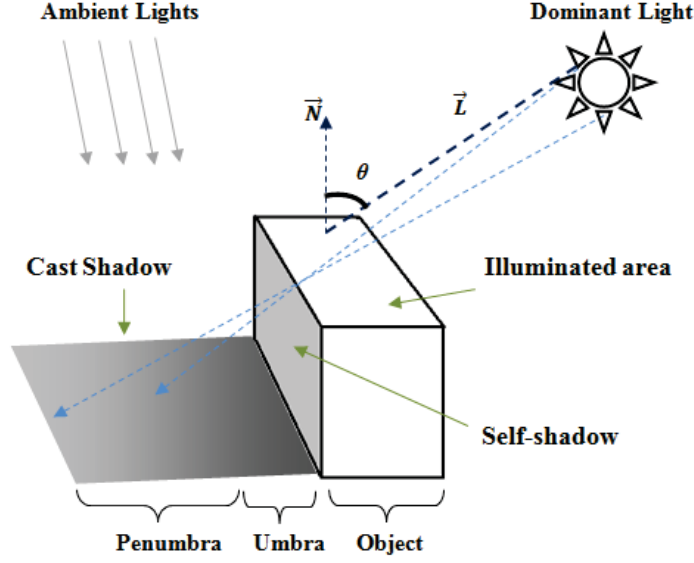


Figure 3.1: Shadow model

where $0 < i^t(p) < \infty$ is the irradiance term, which indicates the amount of source illumination received by point p on surface S at time t , and $0 < r^t(p) < 1$ is a coefficient measuring the amount of illumination reflected by the same point.

Based on the Phong model [96] and assuming that the light source is far from the object, the irradiance term $i^t(p)$ can be further expressed with respect to the incident angle θ of the dominant light source (the angle between the direction of the light source $\vec{L} = (l_x, l_y, l_z)$ and the point surface normal $\vec{N} = (n_x, n_y, n_z)$), the intensity of the dominant light source c_D , and the intensity of the ambient light c_A :

$$i^t(p) = c_A + T(p) \cdot T_D \cdot c_D \cdot \cos(\theta) \quad (3.2)$$

The coefficient $0 < T_D < 1$ measures the amount of light energy available from

the dominant light at time t . This value represents the global change in illumination of the dominant light source over time. $T(p)$ determines the amount of available light energy of the dominant light source received by point p at time t . This value represents the local change in illumination at point p over time. Theoretically, point p belongs to a cast shadow when the value of the coefficient $T(p) < 1$. Furthermore, point p is considered to belong to the penumbra region of the cast shadow when $0 < T(p) < 1$ or to the umbra region when $T(p) = 0$.

3.1.2 Shadow Properties

Many properties are associated with shadows; however, only properties that are directly related to the field of moving shadow detection are discussed here. Each method in the literature relies on at least one of the following properties to detect shadows:

- **Intensity reduction:** The intensity of a background point is reduced when a shadow occurs because the irradiance term $i^t(p)$ in Equation (3.1) receives less light from the dominant light source. The strength of the ambient light sources around the object determines how much darker the point will be.
- **Linear attenuation:** When the spectral power distributions (**SPDs**) of the dominant and ambient light sources are similar, the background color will be maintained when a shadow occurs. In other words, linear attenuation exists for the three color channels (R, G, B) ; that is,

$$[R_{SH} < R_{NSH}] \cup [G_{SH} < G_{NSH}] \cup [B_{SH} < B_{NSH}] \quad (3.3)$$

where $[R_{NSH}, G_{NSH}, B_{NSH}]$ and $[R_{SH}, G_{SH}, B_{SH}]$ are color vectors for the same point in nonshadowed (NSH) and shadowed (SH) background regions, respectively.

- Nonlinear attenuation: When the **SPDs** for the dominant and ambient light sources are different, the background colors, depending on the color of the ambient light, will be changed when a shadow occurs.
- Reflectance constancy: The textures and patterns of background regions do not change over time, i.e., the object reflectance term $r^t(p)$ in Equation (3.1) does not change when a shadow occurs.
- Size property: The size of the shadow depends mainly on the direction of illumination, the size of the moving object, and the number of available light sources.
- Shape property: The shape of the shadow depends on the shape of the object that casts the shadow and the direction of the illumination.
- Shadow direction: For a single point light source, only one shadow directions exist; however, multiple shadow directions occur if more than one light source is present in the scene.
- Motion attitude: Shadows follow the same motion as the objects that cast them.

3.2 Problematic Situations

Many problematic situations of shadow detection must be considered when developing an algorithm for shadow detection. These problematic situations can be defined by one of two factors: (i) the color and intensity of the lights that illuminate the shadow and nonshadow regions and (ii) the reflectance properties of foreground objects, shadows and background regions. Examples of problematic situations include chromatic shadows, shadow color blending, dark surfaces, foreground-background camouflage, and nontextured surfaces. The first two problems relate to regions of shadows that can be easily misclassified as moving objects. The latter three problems relate to regions of objects that can be misclassified as shadows.

Careful consideration should be given to these problematic situations when developing an algorithm for shadow detection because these scenarios often cause large intraclass variations and lead to misclassification.

In the following subsections, the most common problematic cases for moving cast shadows are addressed. Although not considered to be a problem in recognizing shadows, achromatic shadows are discussed first to understand the concept of a chromatic shadow.

3.2.1 Achromatic Shadow

Achromatic shadow occurs when two light sources, the dominant light and the ambient light, have similar **SPDs** for both the penumbra and umbra regions. This type of shadow is common for almost all indoor environments, where the

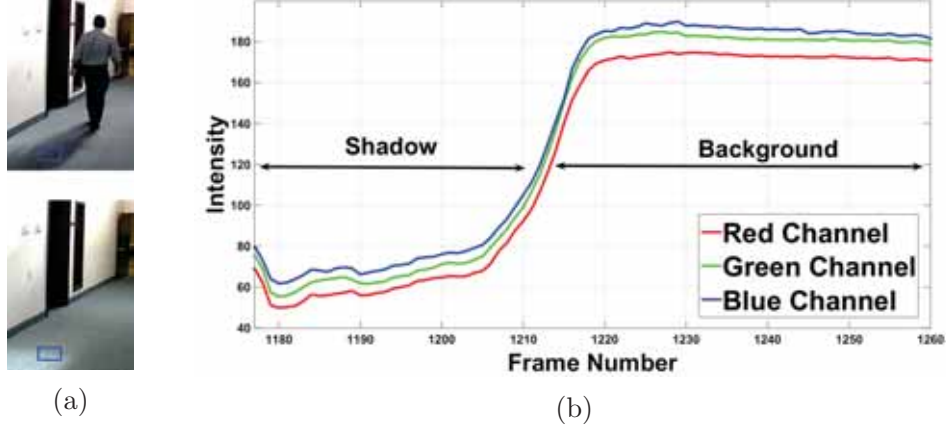


Figure 3.2: An example of linear attenuation of achromatic shadows; (a) a frame in the sequence (top) and the corresponding background (bottom) with an area (highlighted in blue) under the test, and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel) for a number of frames.

colors of the lights are similar. An example data analysis of an achromatic shadow is given for an indoor sequence in Figure 3.2. The intensity of the **RGB** channels of a selected area (highlighted in blue) of a given frame is monitored for a period of time, and clearly (as indicated in Figure 3.2-(b)), the intensity of the three channels (**RGB**) decreases when a shadow occurs; however, the order of the three channels is maintained. On the basis of the ground truth shown in Figure 3.2-(b), the real distribution of the points belonging to the foreground object (highlighted in red) and the shadow (highlighted in green) is plotted in Figure 3.3. The figure shows the joint distributions of two channels. The distribution for each color is calculated in normalized **RGB** color space using:

$$\begin{cases} R = \left| \frac{R_{BG}}{R_{BG} + G_{BG} + B_{BG}} - \frac{R_{CF}}{R_{CF} + G_{CF} + B_{CF}} \right| \\ G = \left| \frac{G_{BG}}{R_{BG} + G_{BG} + B_{BG}} - \frac{G_{CF}}{R_{CF} + G_{CF} + B_{CF}} \right| \\ B = \left| \frac{B_{BG}}{R_{BG} + G_{BG} + B_{BG}} - \frac{B_{CF}}{R_{CF} + G_{CF} + B_{CF}} \right| \end{cases} \quad (3.4)$$

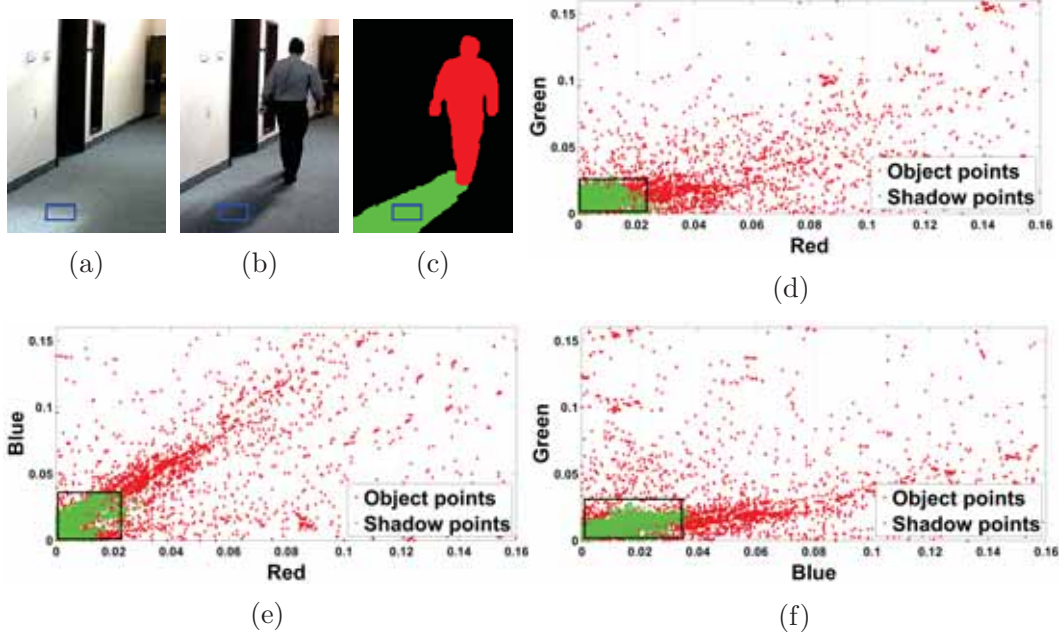


Figure 3.3: Discriminating the two classes, shadows and moving objects, for achromatic shadows for an area highlighted in blue; (a) the background frame, (b) a current frame in the sequence, (c) the ground truth image of (b), (d) red-green joint distribution in normalized **RGB** space, (e) red-blue joint distribution in normalized **RGB** space, and (f) Green-Blue joint distribution in normalized **RGB** space.

where $[R_{BG}, G_{BG}, B_{BG}]$ and $[R_{CF}, G_{CF}, B_{CF}]$ are **RGB** color vectors for the same point in the background image and the current frame, respectively. Notably, most of the shadow points (green dots inside the black rectangle) are distributed near the origin of the three planes (red-green, red-blue, and green-blue), where an optimal threshold can be estimated to eliminate these shadow points from the foreground objects.

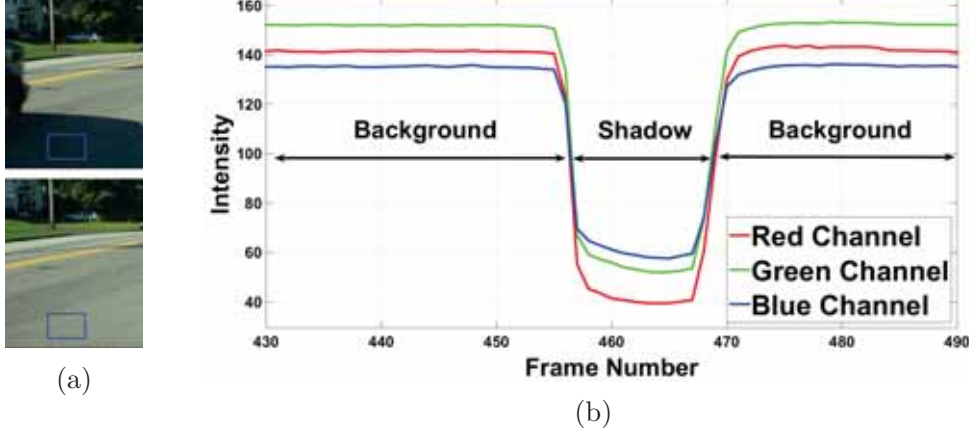


Figure 3.4: An example of non-linear attenuation of chromatic shadows; (a) a frame in the sequence (top), the corresponding background (bottom) with an area (highlighted in blue) under the test, and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel) for a number of frames.

3.2.2 Chromatic Shadow

A chromatic shadow occurs when some parts of the shadow are illuminated by light of a different color than that of the dominant light. As a result, the colors of the **RGB** channels for the shadows are different from those in the corresponding backgrounds. Chromatic shadows occur mainly for outdoor sequences, where the **SPD** of the ambient light (blue color reflected from the sky) is different from that of the dominant light (the sun), causing color blending to shadow region points.

An example of this case is shown in Figure 3.4. Figure 3.4-(b) shows the intensity of the three channels (**RGB**) for an area of the background (highlighted in blue in Figure 3.4-(a)) for 60 frames in a traffic scene. Clearly (as indicated in Figure 3.4-(b)), the background color becomes more bluish when a shadow occurs.

Additional analysis of chromatic shadows is visualized in Figure 3.5-(d)-

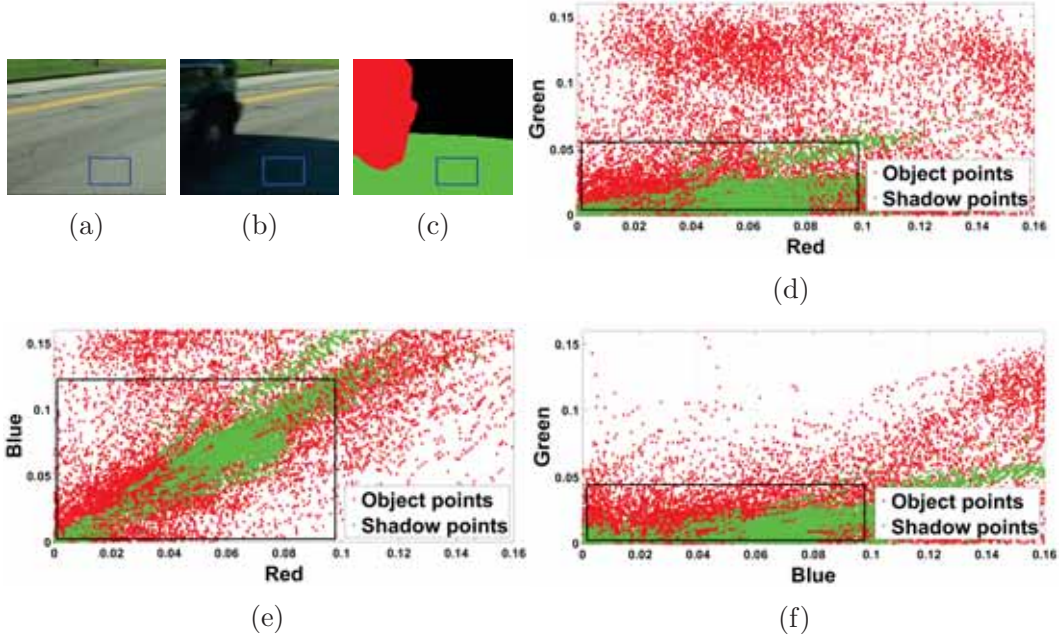


Figure 3.5: Discriminating the two classes, shadows and moving objects, for chromatic shadows for an area highlighted in blue; (a) the background frame, (b) a current frame in the sequence, (c) the ground truth image of (b), (d) red-green joint distribution in normalized **RGB** space, (e) red-blue joint distribution in normalized **RGB** space, and (f) green-blue joint distribution in normalized **RGB** space.

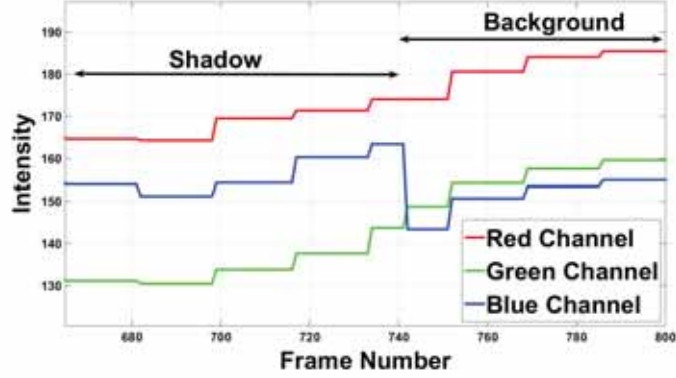
(f), where the distribution of each channel is calculated using Equation (3.4). Most shadow points (highlighted in green) are spread over the two axes and have a distribution similar to the object points (highlighted in red). In contrast to chromatic shadows, even an optimal threshold (highlighted in black rectangle) cannot differentiate between the two classes (the object and its shadow).

3.2.3 Shadow Color Blending

When the reflectance of the background surface is high, some parts of the foreground object reflect off the background, causing color blending in the background. Figure 3.6 shows an example case of shadow blending where an area (highlighted in white in Figure 3.6-(a)) has been tested over time. Clearly (as



(a)



(b)

Figure 3.6: An example of shadow colour blending; (a) a frame in the sequence with an area (highlighted in white) under the test and (b) the intensity of the three channel for the tested area with red (red channel), green (green channel) and blue (blue channel) for a number of frames.

indicated in Figure 3.6-(b)), the intensity of the blue channel increases when a shadow occurs because parts of the foreground object are reflected by the background.

3.2.4 Foreground-background Camouflage

Foreground-background camouflage is a serious problem in all image processing and computer vision applications, including moving object detection [97]. Some moving object detection methods have limited their scope to detect moving objects from the background in scenarios with foreground-background camouflage [98, 99].

The problem occurs when parts of the foreground region have similar appearance in terms of color and intensity to the corresponding background. That is, the intensity difference between the foreground and background for all channels of the color image is very small, and the foreground cannot be accurately detected. An example of this case is illustrated in Figure 3.7, where a camouflaged

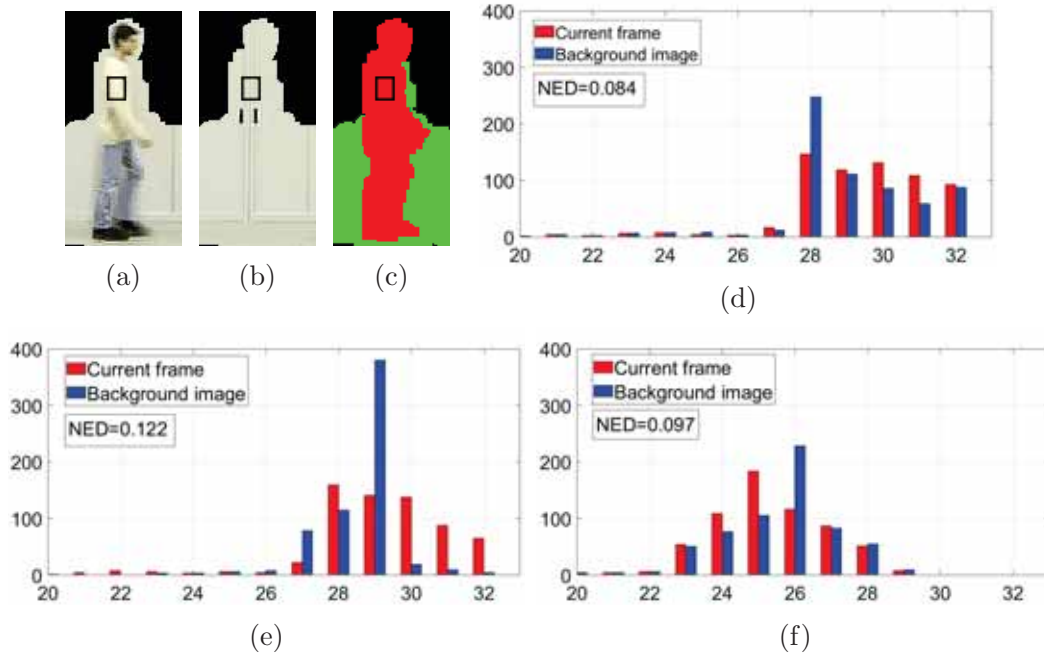


Figure 3.7: An example of **RGB** intensity differences for a foreground patch (highlighted in black) affected by camouflage; (a) a given image under the test, (b) background image, (c) ground truth image of (a), (d) intensity histogram, calculated for red channel, for all points belong to the patch in the background image (blue bars) and the current frame (red bars), (e) intensity histogram for green channel, and (f) intensity histogram for blue channel. Euclidean distance (**ED**) is used to measure the differences.

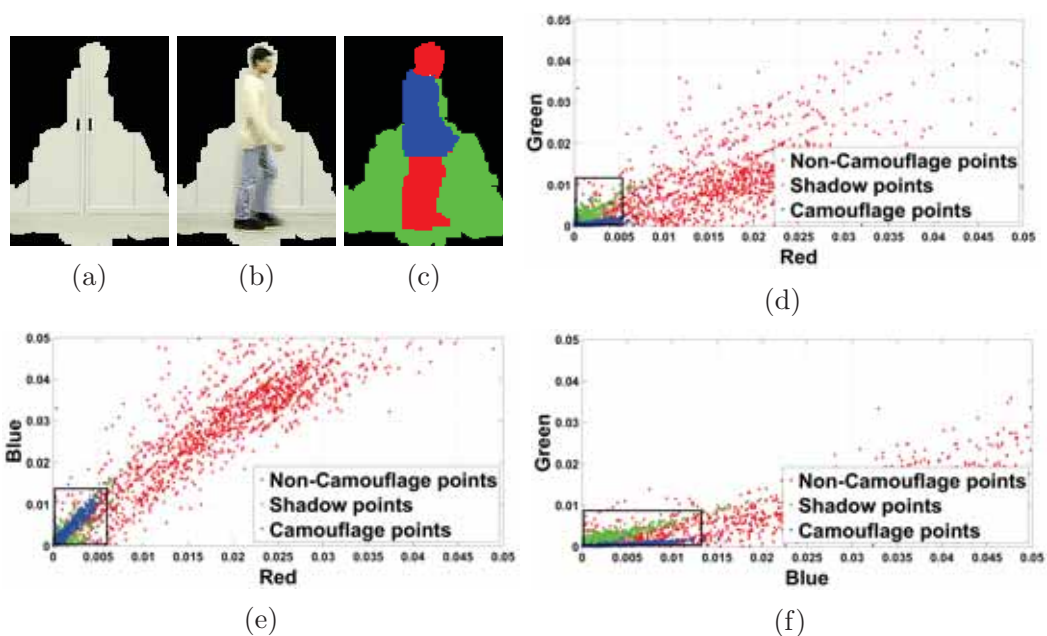


Figure 3.8: An example of foreground-background camouflage; (a) a given image under the test masked with **CDM**, (b) background image masked with **CDM**, (c) ground truth image (camouflage area for the foreground object is highlighted in blue, non-camouflage area is highlighted in red and the shadow cast is highlighted in green), (d) the red-green joint distribution in normalized **RGB** space, (e) the red-blue joint distribution in normalized **RGB** space, and (f) the green-blue joint distribution in normalized **RGB** space.

patch is taken from the current frame (the points belonging to the yellow shirt) and the background image (the points belonging to the yellow cabin) and analyzed using **RGB** intensity histograms. Since yellow is a combination of mainly red and green, the intensity values for the red channel and green channels of most points fall into high bins (bin numbers 27 to 32), as shown in Figure 3.7-(d)-(e). The intensity difference for each channel is calculated using the normalized Euclidean distance (**NED**)¹ to measure the similarity between the two patches. The foreground patch, with an **NED**=0.084 for the red channel, **NED**=0.122 for the green channel, and **NED**=0.097 for the blue channel, has high similarity in intensity and color distribution to the corresponding background; therefore, the patch may be misclassified as part of the background.

The previous example is considered to study the effect of camouflage on moving shadow detection, as shown in Figure 3.8. The camouflaged region of the yellow shirt is highlighted in blue in the ground truth image. As the two camouflaged surfaces have similar color and intensity, most of these points (highlighted in blue) are distributed near the origin of the three plots (Figure 3.8-(d)-(f)). Although optimal thresholds are used to distinguish the shaded points (highlighted in green), the camouflaged points are misclassified as shadows.

3.2.5 Nontextured Surfaces

Textures can be a useful tool to detect and eliminate shadows of moving objects. However, in many cases, texture information is not available because some internal parts of the foreground object or the background surface are nontextured or flat. Figure 3.9 shows the distributions of points belonging to three classes (textured surface in red, nontextured surface in blue, and shadowed surface in

¹The normalized squared euclidean distance gives the squared distance between two vectors where there lengths have been scaled to have unit norm

green). Most nontextured points are located near the origin and are spread in a similar way as the shadowed points. The axis of the histogram is calculated using:

$$\begin{cases} R(p) = \frac{1}{N} \sum_{n=1}^N \left| \frac{R_{CF}(p)}{R_{BG}(p)} - \frac{R_{CF}(p-n)}{R_{BG}(p-n)} \right| \\ G(p) = \frac{1}{N} \sum_{n=1}^N \left| \frac{G_{CF}(p)}{G_{BG}(p)} - \frac{G_{CF}(p-n)}{G_{BG}(p-n)} \right| \\ B(p) = \frac{1}{N} \sum_{n=1}^N \left| \frac{B_{CF}(p)}{B_{BG}(p)} - \frac{B_{CF}(p-n)}{B_{BG}(p-n)} \right| \end{cases} \quad (3.5)$$

where $[R_{BG}, G_{BG}, B_{BG}]$ and $[R_{CF}, G_{CF}, B_{CF}]$ are **RGB** color vectors for the same point in the background image and the current image, respectively. $(p-n)$ is the neighboring point of p , and the total number of neighboring points is N .

3.2.6 Dark Surfaces

Dark surfaces occur when parts of the foreground have low reflection. As a result, these foreground parts are often misclassified as shadows. An example of this case is shown in Figure 3.10, where a person wearing a black suit is walking under a large shaded area. Due to having similar intensities to those of the shadow points, these dark points (highlighted in blue in Figure 3.10-(c)) are detected as shadows, even though an optimal threshold is used (area under the black rectangle in Figure 3.10-(d)-(f)). The axis of the histogram is calculated using Equation (3.4).

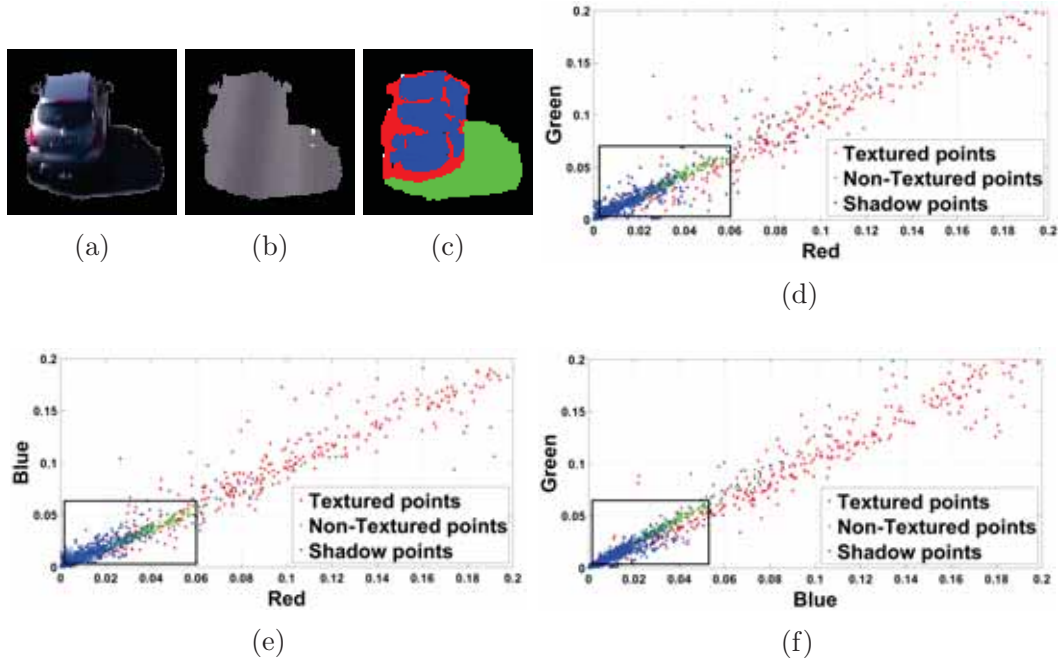


Figure 3.9: An example of non-textured surfaces; (a) the image under the test masked with **CDM**, (b) the background image masked with **CDM**, (c) ground truth image (non-textured area for the foreground object is highlighted in blue, textured area is highlighted in red and shadow cast is highlighted in green), (d) the red-green joint distribution in normalized **RGB** space, (e) the red-blue joint distribution in normalized **RGB** space, and (f) the green-blue joint distribution in normalized **RGB** space.

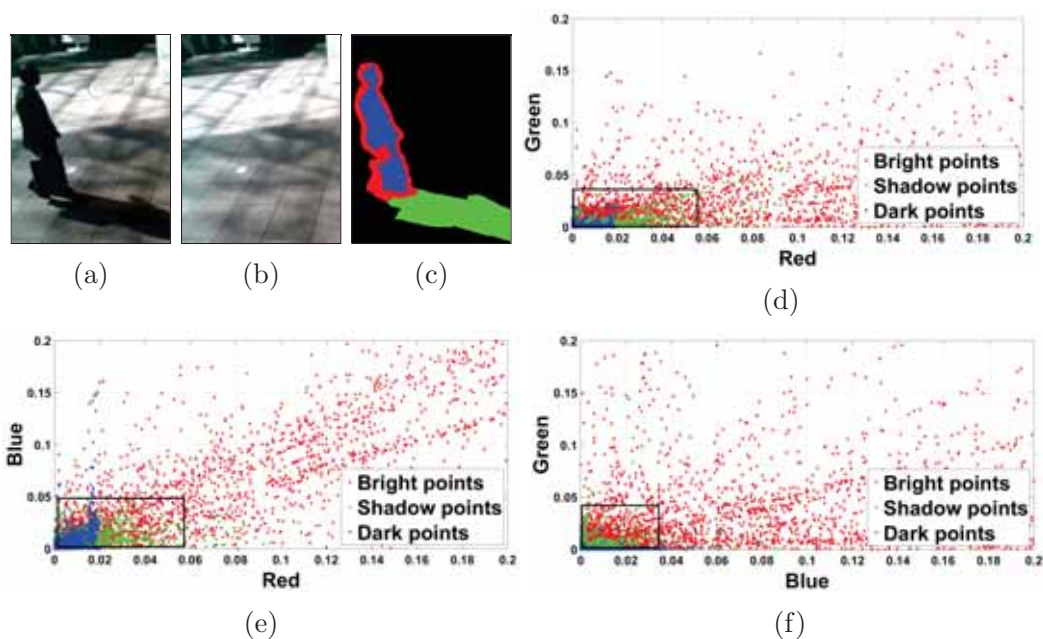


Figure 3.10: An example of dark surfaces; (a) the frame under the test, (b) the background image, (c) ground truth image (dark area for the foreground object is highlighted in blue, bright area is highlighted in red and shadow cast is highlighted in green), (d) the Red-Green joint distribution in normalized **RGB** space, (e) the Red-Blue joint distribution in normalized **RGB** space, and (f) the Green-Blue joint distribution in normalized **RGB** space.

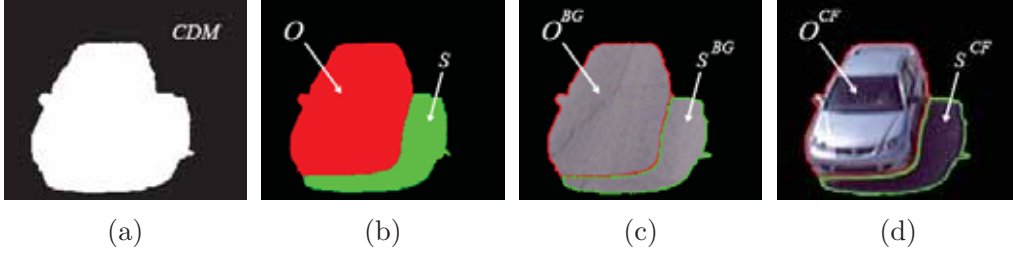


Figure 3.11: Labelling the two classes, objects and shadows, for dataset classification; (a) a **CDM**, (b) ground truth image of (a) consists of moving object O and its shadow S , (c) background image **BG** masked with **CDM**, and (d) current frame **CF** masked with **CDM**.

3.3 Dataset Classification

The ability to identify problematic situations within datasets (or a sequence) is extremely useful since moving shadow detection methods are analyzed and evaluated using these datasets. In general, any dataset may contain at least one of the defined problems.

To determine whether a dataset (or a sequence) contains any of the problematic situations discussed in the previous section, a number of samples from the dataset is taken, and further investigation is performed. Each sample consists of the **CDM**, ground truth image, background image, and current frame. These images, along with the class labels, are shown in Figure 3.11. After analyzing the samples, the sequence is considered to contain a problem if at least 75% of the selected samples are affected by that problem. All parameters in this section are set using Otsu's method [100].

Table 3.1 summarizes the methods used to identify these problems in the datasets. For simplicity, let O^{CF} , S^{CF} , O^{BG} , and S^{BG} be four blobs representing the moving object O and its cast shadow S in the current frame CF and background image BG , respectively. The whole procedure for identifying each problem is explained in the following.

Table 3.1: Summary of dataset classification methods

Problem case	Affected class	Colour space	Method used
Chromatic shadows	Shadow	Nrgb	Comparison of mean values of blue channel in the current frame and background image.
Colour blending	Shadow	Nrgb	Comparison of mean values of red and green channels in the current frame and background image.
Camouflage	Object	Gray-scale	Euclidean distance is measured for all pixels in the current frame and background image.
Non-textured surfaces	Object	Gray-scale	Pixel-level and object-level local texture descriptor are used.
Dark surfaces	Object	Gray-scale	Intensity means for all pixels in the object-blob are calculated.

A Chromatic shadow

The shadow blob S^{CF} in the current frame is compared to that of the corresponding background S^{BG} . The comparison is performed by calculating the mean intensity of the blue channel in normalized **RGB** color space.

$$\begin{cases} \bar{b}_{CF} = \frac{1}{N} \sum_{i=1}^N b_{CF}(i) & \forall i \in S^{CF} \\ \bar{b}_{BG} = \frac{1}{N} \sum_{i=1}^N b_{BG}(i) & \forall i \in S^{BG} \end{cases} \quad (3.6)$$

where b is the blue channel of normalized **RGB** color space and N is the total number of pixels belonging to shadow blob S . If the intensity of the *blue* channel

is increased with respect to the background, i.e., $\bar{b}_{CF} > \bar{b}_{BG}$, the shadow blob is considered to be under chromatic shadows.

B Colour blending

Shadow blob S^{CF} in the current frame is compared to its corresponding background S^{BG} . Using red r and green g channels of normalized **RGB** color space, the mean intensities \bar{r}_{CF} , \bar{g}_{CF} , \bar{r}_{BG} , and \bar{g}_{BG} are calculated using Equation (3.6). Shadow blob S^{CF} is then considered to be affected by color blending if the intensity of the red and/or green channel is increased with respect to the background, i.e., $\bar{r}_{CF} > \bar{r}_{BG}$ or $\bar{g}_{CF} > \bar{g}_{BG}$.

C Foreground-background camouflage

To determine whether object blob O^{CF} is in camouflage with respect to the corresponding background O^{BG} , the intensity differences of the two patches are calculated in grayscale images using Euclidean distance.

$$d(O^{CF}, O^{BG}) = \left\{ \frac{1}{N} \sum_{i=1}^N [O^{CF}(i) - O^{BG}(i)]^2 \right\}^{\frac{1}{2}} \quad \forall i \in O \quad (3.7)$$

where $d(\cdot)$ is the Euclidean distance between the two blobs. The patch is then considered to be affected by camouflage if $d(O^{CF}, O^{BG}) \leq 10$.

D Nontextured object surfaces

To determine whether object blob O^{CF} in the current frame has a uniform and smooth surface, texture information is extracted on two levels: pixel level and object level. Any local texture feature descriptor in the literature, such as **LBP** [101], **LTP** [102], **SILTP** [103], or a modified descriptor, as described in Equation (3.5), can be used to extract pixel-level texture information. In this thesis, a simplified version of the **LBP** described in [101] is used.

For each pixel p in foreground object O^{CF} , its local texture descriptor $T(p)$ is calculated by comparing the intensity value of p with each of its direct neighboring pixels q_j with $j \in \{0, 1, 2, \dots, J\}$.

$$T(p) = \frac{1}{J} \sum_{j=1}^J |O^{CF}(p) - O^{CF}(q_j)|, \quad \forall p \in O^{CF} \quad (3.8)$$

where J is the maximum number of neighboring pixels, which can have a value within the range of 1 to 8, depending on the location of the center pixel. $T(p)$ is expected to have a small value if the intensities of the neighboring pixels are similar.

Object-level texture descriptor $T(O^{CF})$ is then obtained by averaging the pixel-level texture descriptors obtained for all pixels in the previous step. Moving object O^{CF} is considered to have smooth and nontextured surfaces if $T(O^{CF}) \leq 5$.

Figure 3.12 shows an example of calculating the pixel-level and patch-level local texture descriptors for a small patch taken from a moving object. The steps of computing the local texture descriptors for the three pixels highlighted in green, cyan, and gray are shown in Figures 3.12-(b)-(c). These pixels have

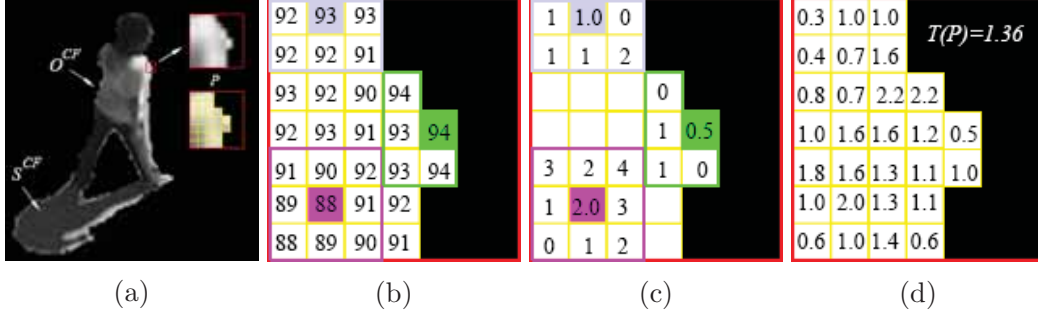


Figure 3.12: Computing the texture descriptor T for a small patch P of a moving object O^{CF} in the current frame; (a) a patch P in the moving object is taken and magnified, (b) intensity values for all pixels in the patch are given, (c) computing pixel-level texture descriptor $T_o(p)$ for three pixels with different spatial locations, and (d) computing pixel-level texture descriptor for all pixels belong to the patch. The patch-level texture descriptor $T(P)$ is calculated for the patch.

different numbers of neighbors. The pixel-level texture descriptors $T(p)$ are calculated for all pixels using Equation (3.8). With $T(P) = 1.36 \leq 5$, patch P is considered to have smooth and nontextured surfaces.

E Dark object surfaces

On the basis of grayscale images with an intensity range of $[0, 255]$, object-blob O^{CF} in the current frame with mean intensity below 25 is considered to be a dark object.

3.4 Summary

This chapter highlighted possible problematic situations present in real-image sequences and discussed their influences on distinguishing moving objects from shadows. These problematic situations are analyzed based on the color and intensity of the light illuminating the two main regions, i.e., objects and shadows, and the reflectance properties of these regions. These problems may cause substantial misclassification problems between shadows and objects.

In the second part of this chapter, various methods were presented to classify moving shadow detection datasets with respect to these problematic situations. This dataset classification is extremely useful in evaluating the proposed shadow detection methods. In fact, the proposed dataset classification is used in Chapter 6 to organize and classify common datasets for moving shadow detection.

Chapter 4

Change Detection

4.1 Introduction

Change detection identifies parts of the foreground objects where the appearances differ from the corresponding background. From the perspective of change detection, an image obtained from a video sequence can be divided into two complementary sets of pixels. The first set contains the pixels corresponding to foreground objects (people, vehicles, etc.) while the second set complements the foreground objects and contains the background pixels.

The aim of change detection systems is to distinguish the two sets from each other and represent the output results as binary masks. The output binary mask is often referred to as the **CDM**. Examples of CDMs for various image sequences are shown in Figure 4.1.

Many issues may cause a failure in any change detection system. These problems should be considered when developing background subtraction algorithms. Examples of such challenges are shown in Figure 4.2. Such challenges

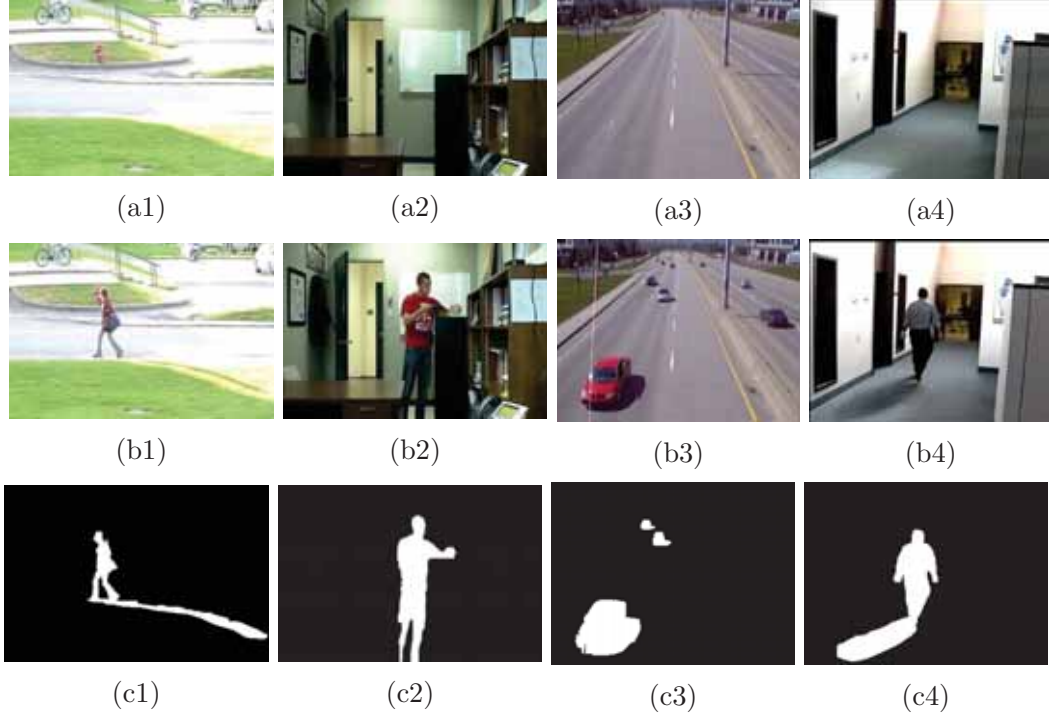


Figure 4.1: Examples of change detection masks for four different scenarios; (a1 - a4) background image, (b1 - b4) given frame, and (c1 - c4) change detection masks. The foreground objects are highlighted in white and the backgrounds are in black.

include global illumination changes, background geometry changes, dynamic background, background initialization, and noise. In many applications, an accurate change detection system that is capable of responding to the above problems quickly is extremely important. In addition, the system should be fast and computationally inexpensive and have low memory requirements.

This chapter consists of two main parts. In the first part, three common types of change detection methods are briefly discussed, along with their limitations, assumptions, advantages, and disadvantages. In the second part, a new framework for change detection in image sequences is proposed. More details of the two parts are given in Sections 4.2 and 4.3.

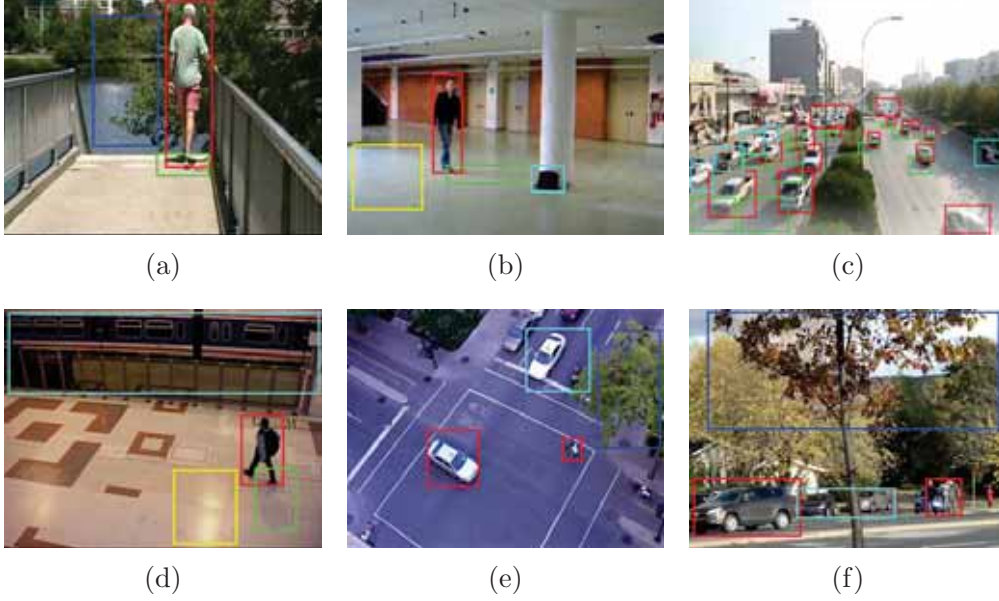


Figure 4.2: Examples of various challenges for change detection. The object of interests are highlighted in red, high frequency objects are highlighted in blue, objects caused changes in background geometry are highlighted in cyan, areas directly affected by global change in illumination are highlighted in yellow, and cast shadows are highlighted in green.

4.2 Existing Change Detection Methods

A wide variety of change detection techniques have been reported in the literature [104]. These methods can be broadly divided into three main types: optical flow, frame differencing, and background subtraction.

Optical flow methods [105, 106, 107], which are based on the optical flow distribution characteristics of the image under the test, compute the image optical flow field. This process can provide accurate detection results under various scenarios; however, the large number of required calculations and the sensitivity to noise limit the real-time application.

In frame differencing methods [108, 109, 110], the simplest of the three types, the preceding frame(s) is set as the background image(s) for the subse-

quent frame. The foreground object in the current frame is then detected by calculating the difference between consecutive frames. The method effectively eliminates most static pixels from an image but always gives unsatisfactory results in extracting most pixels belonging to a moving object. In addition, high false positives and false negatives occur in some cases, such as sudden changes in illumination and fast moving of objects of no interest (e.g., waving trees on a windy day).

Background subtraction [111, 112, 113] is a common technique used to detect changes in the background scene for a stationary camera. In background subtraction methods, the current frame in the video is compared to a static image (background scene free from moving objects) to extract the pixels whose intensities have changed significantly. In general, the computational effort of background modeling and subtraction algorithms is slightly higher than that of frame differencing, but the performance, in terms of accuracy, is considerably better.

4.3 Proposed Change Detection Framework

In this thesis, a new change detection method based on background subtraction is proposed to overcome the limitations and disadvantages of the methods discussed in Section 4.2 and to address possible scenarios occurring in real scenes. Figure 4.3 shows a block diagram of the proposed change detection method. A combination of two feature descriptors, namely, color and global intensity consistency **GIC**, is introduced to effectively identify background points in image sequences under global illumination changes and in the presence of image noise. On the basis of these features, a new adaptive background model is constructed to automatically adapt to scene changes in a self-organizing manner. The frame-

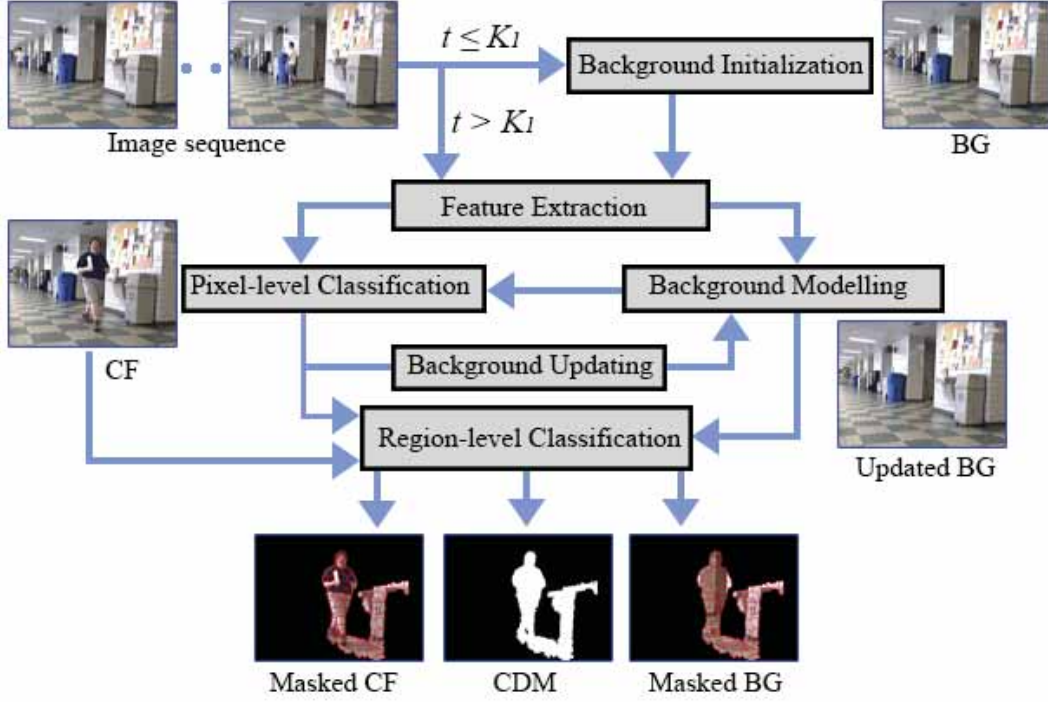


Figure 4.3: Block diagram of the proposed change detection method. The current frame, background image, and change detection mask are labelled as CF , BG , and CDM , respectively.

work provides a strong ability to detect most changes (moving objects and their cast shadows) in the background scene. The processing steps involved in the proposed method are discussed in greater detail in the following sections.

4.3.1 Background Initialization

Background initialization (also called bootstrapping) is a major problem in background subtraction. A background representation free from moving objects is often impractical to obtain in real scenarios. A considerable amount of work has been conducted on problems related to background modeling and updating; however, little attention has been given to background model representation [114] [115]. As a common approach in many background subtraction methods,

a background is initialized by applying a mean/median filter to a number of frames reserved for that purpose at the beginning of the sequence. In order for the background to be estimated precisely, each pixel location should be occluded by objects by no more than 10% when using the mean filter and by less than 50% when using the median filter. On the basis of these assumptions, the mean/median filter is a good choice for estimating the background when a scene is uncongested; however, problems with using the two filters arise when the scene is congested (e.g., traffic scene). In this case, a large number of frames is required to filter out the foreground objects. A fast and accurate technique is proposed to address cluttered and congested scenes to overcome this problem. The method requires fewer frames for estimating the background image and incorporates spatial and temporal consistency to obtain an improved intensity distribution over time. The steps of the proposed background initialization are explained in the following.

Let W_i be a block of size $X \times Y$ obtained by dividing a given frame into a number of nonoverlapping, equal-sized blocks. The size of a block can vary and depends on the scene complexity of the sequence. For congested scenes, a small size is preferable to avoid including both classes, i.e., background and foreground objects, in one block. By contrast, a larger size can be selected for uncongested scenes to accelerate the initialization process. Let P_i^{t1} and P_i^{t2} be two associated patterns of W_i taken from two consecutive frames at two different time instants $t1$ and $t2$, with $t2 = t1 + \Delta t$. Four possible conditions can occur between P_i^{t1} and P_i^{t2} , as shown in Table 4.1. For the first condition, the distance between the two blocks is expected to be too small since the intensity values of the background pixels do not change substantially in a short period of time. For the remaining three cases, the differences in intensity values are expected to be high as the blocks are occluded by different classes (*BG* or *FG*) or by

Table 4.1: Similarity between two patterns for possible conditions.

Condition	P^{t1}	P^{t2}	Possibility
1	$\in BG$	$\in BG$	$P^{t1} \simeq P^{t2}$
2	$\in BG$	$\in FG$	$P^{t1} \neq P^{t2}$
3	$\in FG$	$\in BG$	$P^{t1} \neq P^{t2}$
4	$\in FG_1$	$\in FG_2$	$P^{t1} \neq P^{t2}$

* BG : Background object, FG : Foreground object, $FG_1 \neq FG_2$

two foreground objects (FG_1 and FG_2) with different patterns. The condition where the two blocks belong to the same foreground object is invalid since the two frames are taken at two different times with difference Δt . Considering the speed of the foreground objects, Δt can be set to a large value for surveillance scenes and to a small value in highway traffic scenes.

Based of the above analysis, a small distance between P_i^{t1} and P_i^{t2} indicates a strong possibility that W_i belongs to the background in both frames. To take advantage of this fact, the Hausdorff distance \mathbf{H} is used to find the difference in intensities between the two patterns:

$$H(P^{t1}, P^{t2}) = \max\{h(P^{t1}, P^{t2}), h(P^{t2}, P^{t1})\} \quad (4.1)$$

where the two distances $h(P^{t1}, P^{t2})$ and $h(P^{t2}, P^{t1})$ are the forward and the backward Hausdorff distances between the two patterns, defined as:

$$h(P^{t1}, P^{t2}) = \max_{p^{t1} \in P^{t1}} \{ \min_{p^{t2} \in P^{t2}} \{d(p^{t1}, p^{t2})\} \} \quad (4.2)$$

$$h(P^{t2}, P^{t1}) = \max_{p^{t2} \in P^{t2}} \{ \min_{p^{t1} \in P^{t1}} \{d(p^{t2}, p^{t1})\} \} \quad (4.3)$$

where $d(p^{t1}, p^{t2})$ and $d(p^{t2}, p^{t1})$ measure the distance between any element in one pattern to the nearest point in the other.

Given the \mathbf{H} distances between every pair of patterns in the two frames, an initial background model mask of the same size as the sequence frame can be obtained:

$$BGM(W_i) = \begin{cases} 0, & \text{if } H(P^{t1}, P^{t2}) \geq TH_{BG} \\ P_i^{t2}, & \text{otherwise} \end{cases} \quad (4.4)$$

where TH_{BG} is a distance threshold for distinguishing the two classes. Note that more the recent pattern P_i^{t2} , instead of P_i^{t1} , is used to model the background.

The initial background model BGM obtained from Equation (4.4) contains a number of foreground blocks whose distance values are above the distance threshold TH_{BG} . These blocks should be replaced by patterns belonging to the background scene to produce a final model of the background scene that is free from moving objects. This process can be performed by computing the average of the intensity distribution $\overline{P}_i^t = \frac{1}{X \cdot Y} \sum_{x,y} P_i^t(x, y) \forall x, y \in P_i$ for a number of frames $t = t2 + 1, t2 + 2, \dots + K1$, represented as a histogram with J bins. Note that $K1$ is the total frames required for background initialization. To obtain an optimal histogram representation with a minimum number of frames, the weighted votes between adjacent bins j and $j + 1$ are interpolated linearly, as described in [116]:

$$\begin{cases} v(j) = \sum_t |\overline{P}_i^t - c_{j+1}|, & \text{if } l_j \leq P_i^t < u_j \quad \forall j = \{1, 2, \dots, J-1\} \\ v(j+1) = \sum_t |\overline{P}_i^t - c_j|, & \text{if } l_{j+1} \leq P_i^t < u_{j+1} \end{cases} \quad (4.5)$$

where $c_j = (u_j + l_j)/2$ is the value of the center bin j , u and l are the upper bound and lower bound of the bin, $v(j)$ is the sum of the weighted votes of the bin, and $|\cdot|$ measures the distance between the mean intensity of the block to the center bin (a weighted vote of the bin).

Regardless of the scene conditions and complexity, the peak of the weighted votes in the constructed histogram consistently corresponds to the bin with the same intensity range as that of the background patterns. Since more than one background pattern is available, the background pattern P_i^m that best represents block W_i in the background model BGM is the one that gives the minimum distance of the mean intensity to the value c_m of the associated center bin with the highest score in the histogram:

$$P_i^m = \min_t |\overline{P}_i^t - c_m| \quad (4.6)$$

The final background image BG can then be produced by updating Equation (4.4) with the new patterns obtained from Equation (4.6):

$$BG(W_i) = \begin{cases} P_i^m, & \text{if } BGM(W_i) = 0 \\ P_i^{t2}, & \text{otherwise} \end{cases} \quad (4.7)$$

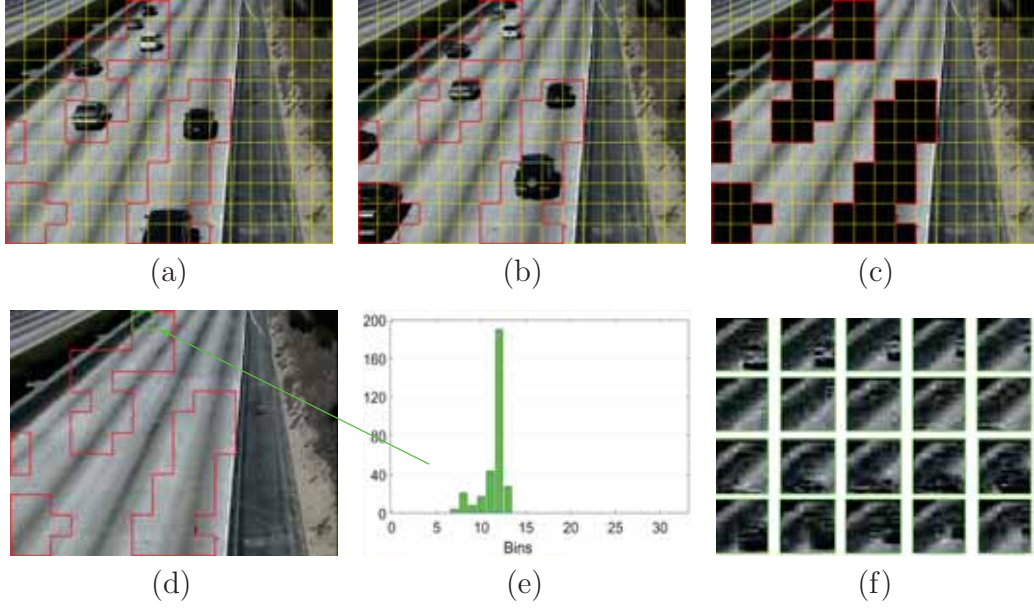


Figure 4.4: Computing background image for a busy traffic scene using the proposed block-based background initialization method; (a) a frame at $t=5$ divided into 12×16 blocks, (b) a frame at $t=10$ divided into 12×16 blocks, (c) initial background model obtained by applying Equation (4.4) with estimated area of changes highlighted in black, (d) the background scene image computed using 20 frames, (e) histogram of intensity mean of a foreground block highlighted in green in (d), and (f) visualization of the 20 image patterns used to compute the histogram shown in (e).

The total number of frames, $K1$, used to estimate the background scene is based on the scene's characteristics, including the speed and size of foreground objects, the scene conditions (clear or congested), weather conditions, and illumination conditions. A larger number of frames $K1$ is preferable in cases where the scene is congested, while a smaller number of frames is required to estimate the background scene in other cases.

Figure 4.4 shows an example of computing the background image for a busy highway traffic scene using the proposed background initialization method. Given the original frame size of 380×480 and setting the size of each block to 30×30 , the frames $K1$ of the sequence are divided into 12×16 equal-sized blocks. The other major parameters are set carefully to obtain a clean background image.

For example, the distance threshold TH_{BG} is set to 30 to minimize the risk of including foreground patterns in the initial background model. Furthermore, setting $t1$ to 5, $t2$ to 10, and $K1$ to 30 provides a good background estimation since the target objects (vehicles) are moving fast and do not rest in a block location for more than 40% of the entire length of $K1$. Finally, the number of bins J is set to 32 to accommodate 8 gray-levels from 256 ($[0,255]$) discrete intensity levels. With the above parameter settings, Equation (4.4) is applied to detect background patterns and mask foreground patterns, as highlighted in black in Figure 4.4-(c), for further processing. The background patterns are estimated for all masked foreground blocks using histograms of the mean intensity for 20 frames, as shown in Figure 4.4-(d). A histogram of one of the foreground blocks is shown along with all 20 patterns (used to compute the histogram) in Figures 4.4-(e)(f) for illustration. The background image is successfully estimated using the proposed method.

4.3.2 Feature Descriptors

Many image feature descriptors can be used to detect changes in the background scene. A set of two pixel-level feature descriptors, namely, **HSV** color information and global intensity consistency **GIC** information, are used for the background subtraction method introduced in this chapter. Both features are extracted using normalized **HSV** color space and **RGB** color space. The two feature descriptors are discussed in greater detail in the following.

A - Color Descriptor

The normalized values of the three channels in **HSV** color space are used to extract the color information for pixel p . Compared to other common color

spaces, such as **RGB** and **YCbCr**, **HSV** color space can successfully cope with changes in global illumination. In fact, using the normalized values of the three components in **RGB** color space to obtain **HSV** color format can result in increased capability to recognize background points in scenarios of global illumination changes. In the following, the process of transforming an **RGB** image into **HSV** color format is discussed, as described in [117].

The **RGB** image obtained from the incoming frame is transformed into **HSV** color space using a \mathcal{T} operator, i.e., $I_{HSV} = \mathcal{T}[I_{RGB}]$. The values for each component, i.e., H , S , and V , are computed as below.

$$H = \begin{cases} \cos^{-1} \left\{ \frac{0.5 \cdot [(r - g) + (r - b)]}{\sqrt{[(r - g)^2 + (r - b)(g - b)]}} \right\}, & \text{if } b \leq g \\ 2\pi - \cos^{-1} \left\{ \frac{0.5 \cdot [(r - g) + (r - b)]}{\sqrt{[(r - g)^2 + (r - b)(g - b)]}} \right\}, & \text{if } b > g \end{cases} \quad (4.8)$$

$$S = 1 - \frac{3}{(r + g + b)}[\min(r, g, b)] \quad (4.9)$$

$$V = \frac{1}{3}(r + g + b) \quad (4.10)$$

where $H \in [0, 2\pi]$, $S \in [0, 1]$, and $V \in [0, 1]$. The values of components r , g , and b are within the range of $[0, 1]$ and are calculated using **RGB** color format, as given below.

$$\begin{cases} r = \frac{R}{255} \\ g = \frac{G}{255} \\ b = \frac{B}{255} \end{cases} \quad (4.11)$$

B - GIC Feature Descriptor

Many traditional feature descriptors have been proposed in the literature to detect changes in images by analyzing the local texture properties. The most popular texture descriptors include **LBP** [101], modified **LBP** and **SILTP** [113]. These *local* texture descriptors often fail to distinguish object patches from the background when no texture information is available. In addition, these methods cannot accurately handle sudden changes in global illumination and are often sensitive to image noise.

To overcome these problems, a new pixel-level texture feature descriptor, namely, global intensity consistency **GIC**, is proposed by taking background candidate points into account. Rather than being conditionally chosen from the local neighborhood, background reference points are selected globally (discussed below) and used to assess the intensity consistency of the center pixel over time. In this way, whether the intensity of a test pixel (center pixel) is maintained under varying conditions can be investigated. In the following, the **GIC** feature descriptor and the technique used to extract this feature are discussed in detail.

Recall the pixel model formula introduced in Equation (3.1) in Section 3.1.1. The intensity of each pixel $S^t(p)$ in an image plane is a product of the irradiance term i^t (the amount of source illumination received by the point) and the reflectance term r^t (the amount of illumination reflected by the same point):

$$S^t(p) = i^t(p)r^t(p) \quad (4.12)$$

Let $CF(p)$, $BG(p)$, $CF(q)$, and $BG(q)$ be the intensity values of a test point, p , and a reference point, q , belonging to a background surface in the current frame CF and background image BG , respectively. Using Equation (4.12), the intensity ratio of the two points can be expressed as:

$$\begin{cases} \Delta_{p,q}^{CF} = \frac{r^{CF}(p) - \Lambda_1 \cdot r^{CF}(q)}{r^{CF}(p)} \\ \Delta_{p,q}^{BG} = \frac{r^{BG}(p) - \Lambda_2 \cdot r^{BG}(q)}{r^{BG}(p)} \end{cases} \quad (4.13)$$

where $\Lambda_1 = \frac{i^{CF}(q)}{i^{CF}(p)}$ and $\Lambda_2 = \frac{i^{BG}(q)}{i^{BG}(p)}$ are the illumination ratios of the two points in the current frame and background image.

Since the illumination ratio is consistent over time, i.e., $\Lambda_1 = \Lambda_2 = \Lambda$, and the reflectance property of a background point does not change over time, i.e., $r^{CF}(p) = r^{BG}(p)$ and $r^{CF}(q) = r^{BG}(q)$, then $\Delta_{p,q}^{CF} = \Delta_{p,q}^{BG}$. Thus, Equation (4.13) can be rewritten as:

$$r^{CF}(p) - \Lambda \cdot r^{CF}(q) = r^{BG}(p) - \Lambda \cdot r^{BG}(q) \quad (4.14)$$

where

$$\Lambda = \frac{i^{CF}(q)}{i^{CF}(p)} = \frac{i^{BG}(q)}{i^{BG}(p)}, \quad \text{or} \quad \Lambda = \frac{i^{CF}(p)}{i^{BG}(p)} = \frac{i^{CF}(q)}{i^{BG}(q)} \quad (4.15)$$

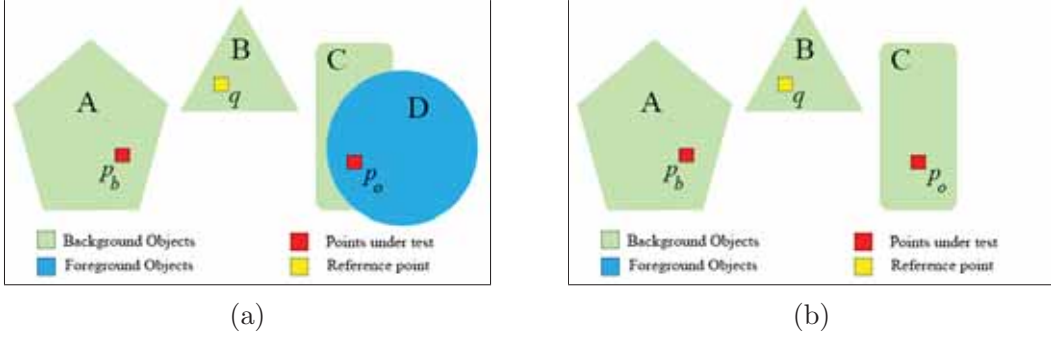


Figure 4.5: Illustration of global intensity consistency for two points, one is a background point p_b and the other is an object point p_o ; (a) the current frame contains three different background surface shapes A, B, and C (highlighted in green) and one target object (shape D highlighted in blue) and (b) the background image contains only the background surface shapes. The reference point q is highlighted in yellow in both images. Note that $p_o \in D$ in (a) and $p_o \in C$ in (b).

which leads to:

$$i^{CF}(p) \cdot r^{CF}(p) - i^{CF}(q) \cdot r^{CF}(q) = i^{BG}(p) \cdot r^{BG}(p) - i^{BG}(q) \cdot r^{BG}(q) \quad (4.16)$$

or:

$$d^{CF}(p, q) = d^{BG}(p, q) \quad (4.17)$$

where $d^{CF}(p, q) = CF(p) - CF(q)$ and $d^{BG}(p, q) = BG(p) - BG(q)$ measures the **GIC** between two image points over time. According to Equation (4.17), the intensity difference between two background points is constant over time, regardless of their positions in the image or the surface materials they belong to. The points can be far from each other in 2D image space and belong to two types of background object material with different reflectance surface properties.

Due to difference in surface reflectance of the foreground object and the background (i.e., $r^{CF}(p) \neq r^{BG}(p)$), **GIC** does not hold for a background point occluded by the target object in the current frame. Furthermore, **GIC** does not

hold for a background point occluded by a cast shadow because of the dissimilarity in the illumination ratios of the two points in the current frame and the background image (i.e., $\Lambda_1 \neq \Lambda_2$).

An example of global intensity consistency is given in Figure 4.5. The two points, $p_b \in A$ (highlighted in red) and $q \in A$ (highlighted in yellow), belong to two different background surfaces A and B in the current frame and the background image. p_o (highlighted in red) belongs to foreground object D in the current frame and to background surface C in the background image. Based on Equation (4.17), the **GIC** for the background point p_b , with respect to reference point q , is constant over time, i.e., $d^{CF}(p_b, q) = d^{BG}(p_b, q)$. For the object point p_o , the **GIC** varies due to differences in the surface properties of D and C , i.e., $d^{CF}(p_o, q) \neq d^{BG}(p_o, q)$.

Eight background candidate points are considered to extract global texture feature descriptor **GIC**. These background points are selected globally (discussed below) and used to check the intensity consistency of the test pixel over time. In this way, whether the intensity of a test pixel is maintained under varying conditions can be checked.

The image is first divided into 3×3 equal-sized blocks, as shown in Figure 4.6. Let Q_s and Q_r represent the selected block with index s and the reference block with index r , respectively. For each of the remaining reference blocks, a pixel is selected to represent each block and is used to compute **GIC** of the test pixel $p \in Q_s$. This selection is based on the minimum distance measurements of the **RGB** weight components between a given frame and the background image. Thus, the background reference point q_r for each block can be computed as:

$$q_r = \min \| Q_r^{CF}(q) - Q_r^{BG}(q) \|_2 \quad (4.18)$$

where q is the index of a pixel in the current frame CF and the background image BG and q_r is the index of the pixel within Q_r with the minimum distance.

The intensity distance $d(j)$ between pixel $p \in Q_s$ and each of the selected eight background candidate points q_r is measured using grayscale images as:

$$d(j) = |G(p) - G(q_r)|, \quad \forall q_r \in Q_r \quad (4.19)$$

where $G(\cdot)$ is the gray value of the point and $j \in \{1, 2, 3, \dots, 8\}$. The feature descriptor **GIC** for each pixel p is then computed using the ceiling operator $\lceil \cdot \rceil$ according to the following:

$$GIC(p) = \lceil \frac{1}{8} \sum_{j=1}^8 d(j) \rceil \quad (4.20)$$

Since eight background reference points are considered for the **GIC** operator, the possible values of the feature descriptor of any pixel **GIC**(p) is within the range $[0, 255]$.

Figure 4.6 shows an example of the block labeling and computational process of the **GIC** feature descriptor for a test pixel with a gray value of $G(p) = 162$ in the current frame. The **GIC** is computed by comparing the intensity value of the test pixel (highlighted in green in Figure 4.6 (e)) to that of the reference points. The average of these intensity comparisons is then rounded up to the next intensity value level using the ceiling function.

C - Feature Descriptor

The two feature descriptors extracted in Section 4.3.2 - A and B have different scales. Therefore, they must be normalized to have values within the same range.

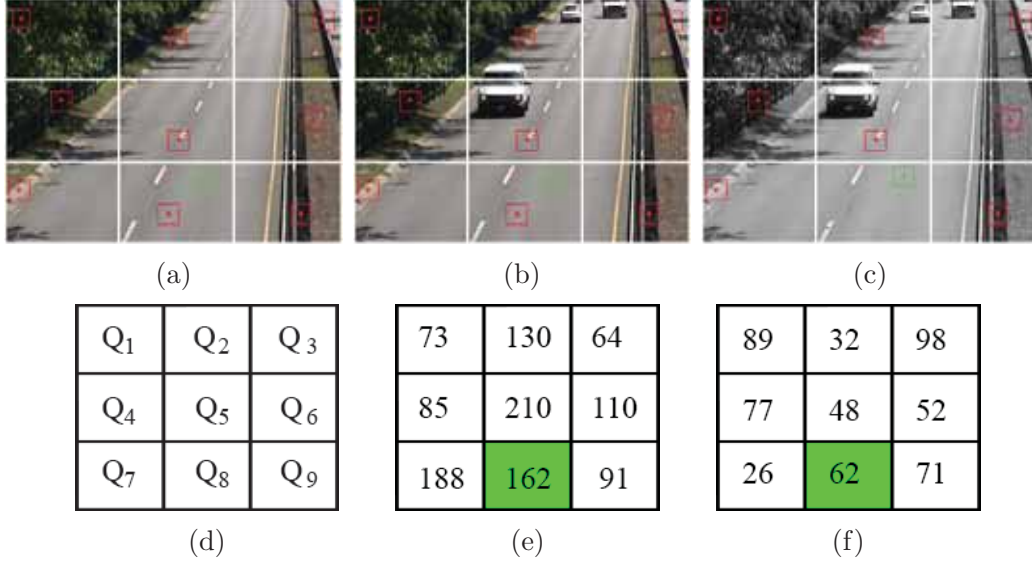


Figure 4.6: Computation process of **GIC** feature descriptor; (a) a background image in **RGB** colour space, (b) a current frame in **RGB** colour space, (c) the current frame in gray-scale. These images are divided into 9 large blocks. The best match pixels for the reference blocks are highlighted in red. The test pixel is highlighted in green, (d) indexes are assigned for each block for simplification, (e) intensity values for the best match pixels of the reference blocks, taken from (c), are shown, and (f) computing processes of the **GIC** feature descriptor for the test pixel with intensity value of 162 (highlighted in green in (e)). **GIC**(162)=62.

Thus, the three components of the color feature descriptor are normalized as below:

$$\begin{cases} H' = \frac{255}{2\pi} \cdot H \\ S' = 255 \cdot S \\ V' = 255 \cdot V \end{cases} \quad (4.21)$$

Both features are then combined to form a 4-dimensional feature vector with a common range of $[0, 255]$. For simplicity and hereinafter, let HSV_T represent the combined background change detection feature descriptor proposed in this chapter. Thus:

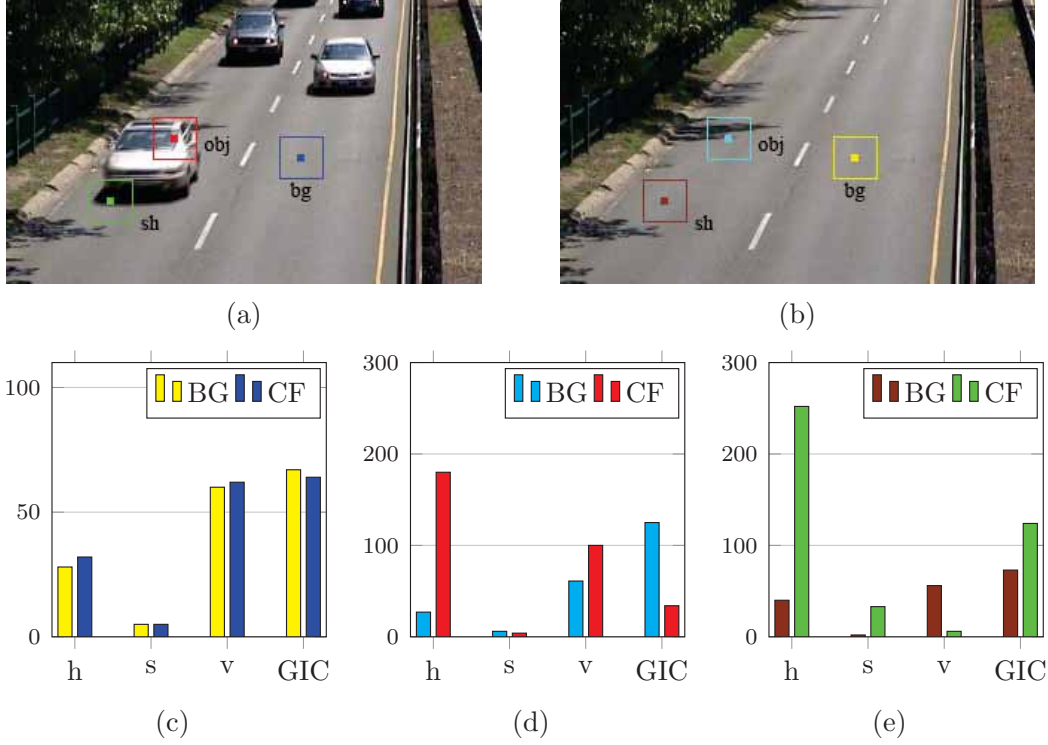


Figure 4.7: Illustration of *HSV**T* feature comparison for three points belong to different surface normals; (a) the current frame with an object point *obj* (highlighted in red), a shaded point *sh* (highlighted in green), and a background point *bg* (highlighted in blue) are selected for testing, (b) the corresponding points in the background image, (c) the feature bar chart for the background point *bg* in the two images, (d) the feature bar chart for the object point *obj* in the two images, (e) the feature bar chart for the shadow point *sh* in the two images.

$$HSV\mathcal{T}(p) = \{H'(p), S'(p), V'(p), GIC(p)\}. \quad (4.22)$$

Figure 4.7 tests the performance of the *HSV**T* feature descriptor under various conditions. Three test points, a background point *bg*, an object point *obj*, and a shadow point *sh*, are selected at three different locations in the scene. The *HSV**T* feature descriptors of these points are extracted in the current frame and the background image and are represented in three bar charts, as shown in Figures 4.7-(c)(d)(e). In these bar charts, the feature dimensions *h*, *s*, *v*, and

GIC are given on the x-axis, and the feature values are shown on the y-axis. The background points *bg*, highlighted in blue and yellow in Figures 4.7-(a)(b)(c), have similar *HSV* feature values in the two images, which indicates consistency in color and intensity for the two background points over time. For the *HSV* feature descriptors of the object point *obj*, highlighted in red and cyan in Figures 4.7-(a)(b), the derived feature bars presented in Figure 4.7-(c) show significant differences in values for *h*, *v* and **GIC**. Meanwhile, they both have a small amount of saturation due to the large percentage of gray in each background surface they belong to. The last bar chart, shown in Figure 4.7-(e), compares the *HSV* features of a point occluded by shadows in the current frame and one belonging to a road surface in the background image. The *h* component significantly increases when shadows occur due to a temporary color shift towards blue (this type of shadow is called a chromatic shadow, as discussed in Chapter 3). In addition, the bar graph shows an increase in the **GIC** feature descriptor for the shadow point with respect to its corresponding background. This increase is expected since the intensity of a background point drops when the point is occluded by shadow.

4.3.3 Background Modeling

The performance of a background subtraction method relies strongly on how the background scene is constructed and updated. In this section and Section 4.3.5, background modeling and updating are discussed in detail, as described in [112].

By means of the *HSV* feature descriptor, the background is modeled by mapping each pixel in the background image *BG* into $n \times n$ weight vectors of the neuronal network¹. Thus, the complete set of weight vectors for all pixels

¹The same term used as in [112] to avoid confusion in this thesis.



Figure 4.8: Background modelling using neuronal networks; (a) an image patch of 2×3 , (b) the enlarged image by 3×3 , and (c) weight vector updating for the best match pixel b_9 and its neighbouring pixels. These pixels are highlighted inside the red square.

of background image BG is represented as neuronal map A with $n \times Y$ rows and $n \times X$ columns. Let $C \times R$ be the size of the neuronal map and $Z_p = (z_1, z_2, \dots, z_{n,n})$ be the model for pixel p , which is located at position (x_1, y_1) in the background image BG . The assigned weight vector for neuronal map A is calculated as follows:

$$A(Z_p) = BG_{HSV T}(x_1, y_1) \quad (4.23)$$

where $Z_p = c_1 : c_2, r_1 : r_2$, $c_1, c_2 \in C$ and $r_1, r_2 \in R$. The index values for c_1, c_2, r_1 , and r_2 are calculated as below:

$$\begin{cases} c_1 = (n \cdot x_1) - 2 \\ c_2 = n \cdot x_1 \\ r_1 = (n \cdot y_1) - 2 \\ r_2 = n \cdot y_1 \end{cases} \quad (4.24)$$

where $C = n \cdot X$ and $R = n \cdot Y$.

Figure 4.8 shows an example of the neuronal map structure for a small background patch with a spatial size $Y = 2$ rows and $X = 3$ columns using $n = 3$. The value for each pixel in the background image has been mapped into 3×3 elements in the neuronal map. As shown in Figure 4.8- (b), the structure of background image BG appears to be enlarged 3×3 times. In this way, the spatial feature is incorporated into the classification process by establishing the spatial dependencies between neighboring pixels.

4.3.4 Pixel-level Classification

In this stage, samples from image sequence are fed to the network. The objective is to find the best match among the corresponding $n \times n$ elements in the neuronal map (background model) to each sample pixel in the incoming frame. In other words, each pixel in the incoming frame is compared to its model Z to determine whether an acceptable matching weight vector exists. Based on the Euclidean distance, the best $HSVT$ weight vector match between pixel p_f (belonging to the incoming frame with index (x_f, y_f)) and p_z (belonging to the background model with $p_z \in Z_f$) can be found as:

$$d(p_f, p_z) = ||HSVT_f - HSVT_z||_2 \quad (4.25)$$

where $p_f = (H_f, S_f, V_f, T_f)$ and $p_z = (H_z, S_z, V_z, T_z)$ are the $HSVT$ component values of pixels p_f and p_z , respectively.

The best matching weight vector candidate z_m with an index of p_m from model Z_f is the vector that gives the minimum distance d :

$$z_m = \arg \min_z d(p_f, p_z) \quad (4.26)$$

where $z = z_1, z_2, \dots, z_n$.

The next step is to determine whether p_f belongs to the background scene or the foreground moving objects. The classification is made at the pixel level by comparing the best matching weight vector candidate z_m with a predetermined threshold TH_p :

$$p_f = \begin{cases} \text{Foreground}, & \text{if } z_m \geq TH_p \\ \text{Background}, & \text{otherwise} \end{cases} \quad (4.27)$$

4.3.5 Background Updating

The updating process in the background model depends on whether p_f is classified as part of the moving objects. If p_f belongs to the moving object and is labeled as foreground, then the background model A remains unchanged. If p_f is classified as belonging to the background, then the weight vectors in the background model A are updated according to a selective weighted running average.

To provide more details on how to update the background model, let us consider a case where the best matching weight vector from model Z_f is found to be z_m . If the index for z_m in background model A is p_m with column and row indexes c_m and r_m , the background is updated according to the following formula:

$$A_t(u) = \alpha_{u,t} \cdot v_f + (1 - \alpha_{u,t}) \cdot A_{t-1}(u) \quad (4.28)$$

where

$$u = (i, j) : [i = c_m - \lfloor \frac{n}{2} \rfloor, \dots, c_m, \lfloor \frac{n}{2} \rfloor], [j = r_m - \lfloor \frac{n}{2} \rfloor, \dots, r_m, \lfloor \frac{n}{2} \rfloor] \quad (4.29)$$

$$p_f = \{(x_f, y_f) : 1 \leq y_f \leq Y, 1 \leq x_f \leq X\} \quad (4.30)$$

where $\alpha_{u,t}$ is the learning rate and $\lfloor \cdot \rfloor$ is the floor operator.

Consider the case shown in Figure 4.8-(c) to illustrate the weight vector updating procedure for a best match pixel and its neighborhoods. In this example, the mapping size is $n = 3$, and the best match for current pixel b is weight vector b_9 . The weight vectors within background model A that must be updated are then $b_5, b_6, c_4, b_8, b_9, c_7, e_2, e_3$, and f_1 ; that is, the parts of Z_b, Z_c, Z_e , and Z_f belonging to background model A and representing models for pixels b, c, e , and f in the current frame are updated.

The learning rate $\alpha_{u,t}$ in Equation (4.28) determines how fast the neuronal network responds to changes in the scene. Based on experiments, the value for $\alpha_{u,t}$ is set to:

$$\alpha_{u,t} = \frac{TH_\alpha}{\max(\omega_u)} \quad (4.31)$$

where ω_u represents the weights in the $n \times n$ neighborhood of u and TH_α is a predefined learning rate constant.

4.3.6 Region-level Classification

The change detection results obtained from pixel-level classification are usually not accurate. Some false positives (pixels that belong to background scene) appear, and further processes are needed to remove them. In addition, false negatives (pixels that belong to the moving object) must be recovered. These misclassifications are often caused by image noise and other varying conditions

discussed earlier. A new framework is developed to perform the final classification at the region level to overcome this problem. The following describes the steps involved in the final region-level classification process.

In the first step, an optimized implementation of the superpixel image segmentation described in [118] is applied to divide the current frame CF into nonuniform \mathcal{V} segments, i.e., $CF(S_v) = \{S_v : v = 1, 2, \dots, \mathcal{V}\}$. Any image segmentation method can be used here; however, due to its effectiveness, superpixel image segmentation is selected in this thesis. The parameters for this segmentation process are set automatically based on the size of the frames. The goal is to segment the current frame CF into two main regions, namely, foreground objects and background objects.

Let S_v be a segmented region of the current frame CF that contains N pixels, where $S_v = \{p_i : i = 1, 2, \dots, N\}$. A test is then performed to count the number of foreground pixels, M , detected in the pixel-level classification stage that belong to each segment S_v :

$$M(S_v) = \frac{1}{N} \sum_{i=1}^N [p_i = 1], \quad \forall p_i = \text{foreground} \quad (4.32)$$

The final classification is then performed for each segment as follows:

$$CDM(S_v) = \begin{cases} 1, & \text{if } M(S_v) \geq TH_S \\ 0, & \text{otherwise} \end{cases} \quad (4.33)$$

The detected foreground pixels are formed in a binary CDM (1-bit binary image) of the same size as the input frame. The CDM mask contains two sets of complementary pixels, black and white: black pixels represent background objects, while white pixels represent foreground (or moving) objects. In this

way, each image segment is expected to contain only parts of the foreground object or parts of its shadow.

4.3.7 Outputs

In addition to the **CDM** obtained from the previous stage, a set of two other frames is required for further processing to detect shadows cast by moving objects. The frames are the current frame segmented into regions and masked with the **CDM** and the updated background images segmented into regions and masked with the **CDM**. The current frame is in **RGB** color format and can be obtained directly from the image sequence. Since the main feature descriptors for the proposed moving shadow detection are extracted in **RGB** color format, the updated background images must be converted back into **RGB** color space and have the same size as the input images.

The enlarged background image, i.e., the background model $A_{HSV T}$, at each instant time t is mapped back to *RGB* color space using the color transformation operator \mathcal{T} :

$$A_{RGB}^t = \mathcal{T}\{A_{HSV}^t\} \quad (4.34)$$

Only the first three components of the feature descriptor *HSV T* are considered in this transformation. The original size of the background image can then be obtained via local spatial averaging:

$$BG(x, y) = \frac{1}{n^2} \sum_{r, c} A(r, c) \quad (4.35)$$

where $(y \cdot n) - 2 \leq r \leq (y \cdot n)$ and $(x \cdot n) - 2 \leq c \leq (x \cdot n)$.

In this way, the updated background image holds the pixel values that are last considered as background pixels. This configuration helps the shadow detection algorithm to compare similarities among background and foreground pixels, which in turn improves the shadow-object discrimination ability.

4.4 Summary

The structure of a scene may differ over time throughout the day for many reasons, including the movement of objects consisting of high-frequency components (e.g., trees, clouds) and changes in the background geometry. These problems, along with gradual or sudden changes in illumination, should be considered when developing background subtraction algorithms.

In this chapter, a new and effective approach to background subtraction was presented. The background was modeled using a combination of color information and global intensity consistency measures to tolerate possible changes in global illumination and noise in natural scenes. Both features are extracted in **HSV** and **RGB** color spaces for each individual pixel. On the basis of the pixel-level classification process, the background model was updated in a self-organization manner according to a selective weighted running average.

The results of the proposed change detection framework consist of a set of three frames, including the masked current frame (**MCF**) in **RGB** color format, the **RGB** updated background image masked (**MBG**) with the **CDM**, and a binary **CDM** image. In addition, the output of the segmentation process for the first two images is included in the result. These images, along with their segmentation process, are used for further analysis in the proposed moving shadow detection method discussed in Chapter 5.

Chapter 5

Moving Shadow Detection

5.1 Introduction

In this chapter, a new local feature descriptor is introduced to address all the problematic situations discussed in Chapter 3 by incorporating color and texture information. Compared to traditional features, incorporating color and texture information yields a good invariant feature descriptor when problems of local illumination variation and nontextured object surfaces exist. On the basis of the new feature descriptors and a reliable model of a sparse representation classifier, a shadow detection framework that effectively extracts moving objects from their shadows is proposed. Figure 5.1 shows an overview of the proposed method. In this method, two overcomplete dictionaries, one for objects and the other for shadows, are constructed from **CDM** segments. Feature vectors are generated for image segments using the proposed feature descriptors. The sparse representation classifier finds the nearest neighbor by computing a linear combination of elements from each dictionary. Classification is then performed at the segment level using the similarity between the test image feature vector

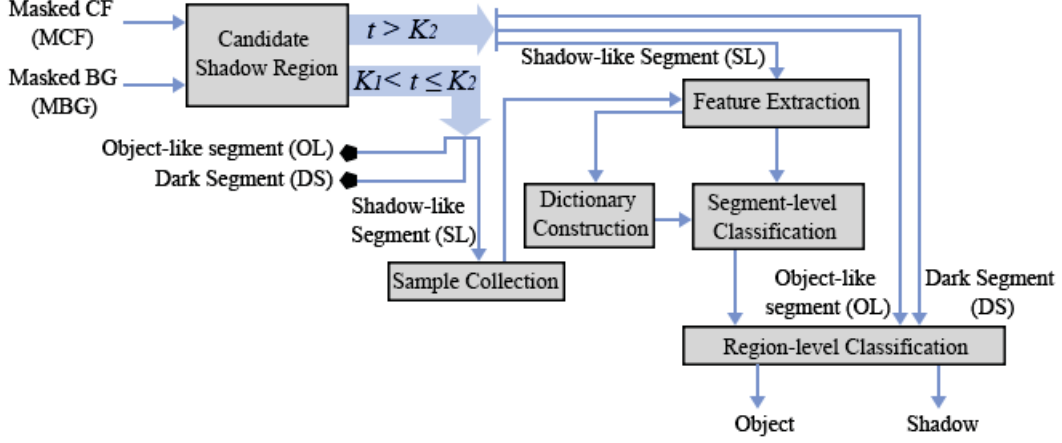


Figure 5.1: Overview of the proposed shadow detection method.

and its best local estimate from each reference dictionary. In the final stage of the classification process, the inter-region dependencies among all neighboring segments are established to include all dark segments in the labeling process.

The remainder of this chapter is organized as follows. In Section 5.2, the initial labeling process is performed to select all possible shadow-like segments for further processing. A novel matching technique and distance measure are deployed to divide the best match segments and the most distant segments among the shadow-like segments into two reference dictionaries (shadows and objects). The sample collection and reference dictionary construction are discussed in Section 5.3. The method of extracting a set of useful features is discussed in detail in Section 5.4. Sections 5.5 and 5.6 provide details of the image segment classification based on the proposed model of sparse representation and inter-region dependencies. This chapter is summarized in Section 5.7.

5.2 Candidate Shadow Segments

Regions under shadows have lower intensity values than other regions in the background: the strengths of the light sources around the object determine how much darker the shadow region is. By contrast, well-illuminated regions usually have higher intensity values than the background.

To take advantage of these differences, the foreground segments obtained from the change detection stage are examined to determine whether their intensity values have increased or decreased with respect to the background. This initial segment labeling can help accelerate the overall labeling process and is thus more efficient for real-time applications. The test segment is assigned to one of two initial categories, namely, object-like segments and shadow-like segments, based on the intensity measures in the current frame and the background image. Object-like segments are bright segments for which the intensity levels of most of the pixels are increased. These segments are directly classified as objects and are discarded from the future labeling process. Meanwhile, shadow-like segments contain relatively dark pixels for which the intensity values have decreased with respect to the background. These segments are labeled as candidate shadows and are considered further in the subsequent labeling process.

In addition to the above two initial categories, segments with low intensity values (below a certain level of the corresponding background) are also labeled as dark segments. As discussed in Chapter 3, these dark segments often cause large intraclass variation and lead to misclassification due to difficulties in extracting valuable feature descriptors. Dark segments are temporarily discarded from the segment-level classification process and are assessed later in the final stage of the region-level classification process. The detailed procedure for determining candidate shadow segments is given in Algorithm 5.1.

Algorithm 5.1: Candidate shadow segments

Input : Masked current frame MCF , masked background image MBG , and index v of image segment S_v

Output: Object-like mask OLM , shadow-like mask SLM , and dark-segment mask DSM

- 1 **For** $v = 1, 2, \dots, V$
- 2 Compute intensity means of \bar{S}_v^{CF} and \bar{S}_v^{BG}
- 3 **If** $\bar{S}_v^{CF} \geq \bar{S}_v^{BG}$
- 4 $S_v^{CF} \in OLM$
- 5 **Else If** $\bar{S}_v^{CF} \leq TH_d$ and $\bar{S}_v^{BG} \leq TH_d$
- 6 $S_v^{CF} \in DSM$
- 7 **Else**
- 8 $S_v^{CF} \in SLM$
- 9 **End For**
- 10 **Return**

To assign a category label $\mathcal{L} \in \{OL, SL, DS\}$ to an image segment S^{CF} , the mean intensities of the segment in the current frame (\bar{S}^{CF}) and its corresponding background (\bar{S}^{BG}) are computed and compared using grayscale images. The segment \bar{S}^{CF} is then labeled according to the following formula:

$$\mathcal{L}(S^{CF}) = \begin{cases} OL, & \text{if } \bar{S}^{CF} \geq \bar{S}^{BG} \\ DS, & \text{if } \bar{S}^{CF} \leq TH_d \text{ and } \bar{S}^{BG} \leq TH_d \\ SL, & \text{otherwise} \end{cases} \quad (5.1)$$

where OL , SL , and DS represent the object-like, shadow-like, and dark segment categories, respectively. TH_d is a predetermined threshold used to identify dark

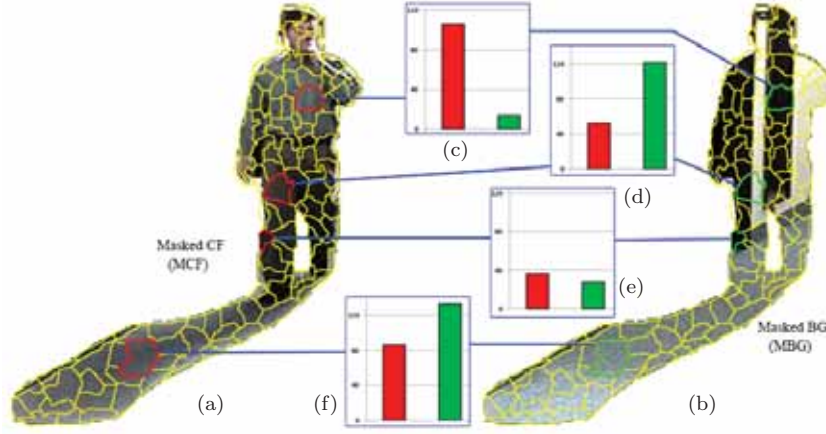


Figure 5.2: Comparison of intensity means of four segments taken from a current frame and its background image; (a) current frame masked with **CDM** and divided into segments, (b) the corresponding backgrounds of (a), (c) intensity means of a bright object segment (**OL**) and a dark background segment, (d) intensity means of a shadow-like object segment (**SL**) and a background segment having two different reflection properties, (e) intensity means of two dark segments (**DS**) taken from two different surfaces, and (f) intensity means of a shadowed segment (**SL**) and a background segment with high reflectance properties.

segments and distinguish them from candidate shadows. TH_d is set to 50 in all the experiments. The output results from this stage are formed in three binary masks. The first binary mask, the object-like mask **OLM**, contains all the segments labeled as objects. The second binary mask, the dark-segment mask **DSM**, contains segments with low-reflection properties. The final binary mask, the shadow-like mask **SLM**, accommodates all segments labeled as shadows.

Figure 5.2 shows a current frame and its background representation masked with the **CDM** and divided into segments. Four segments are randomly taken at different locations of the current frame for demonstration. For each segment, the mean intensity is computed and presented in a bar graph. The first segment belongs to a bright part of the walking person in the current frame and to a dark part of the background surface. Since the mean intensity of the foreground segment (in the current frame) shown in Figure 5.2-(c) is higher than that of

the background segment (in the background image), the segment is directly labeled as belonging to the target object. For the two segments shown in Figures 5.2-(d)(f), the mean intensities of the foreground segments are lower than their corresponding backgrounds and slightly higher than 50. Based on Equation (5.1), these segments - one belonging to the walking person and the other to the cast shadow - are labeled as shadow-like segments and are passed to the next stage for further processing. The last segment is occluded by low-reflectance surfaces in the current frame and the background image. The segment is considered to be dark since its mean intensity is lower than 50 in both images. This segment is assessed in the final classification stage.

5.3 Dictionary Construction

In the proposed method, two reference dictionaries, one for objects and the other for shadows, are involved in the classification process. The samples for these dictionaries are collected from detected image segments in an offline phase calibration. A single segment is assumed to contain either shadow points or object points since superpixel image segmentation is used. The process of constructing the reference dictionaries, using the collected samples and proposed set of feature descriptor, is discussed in detail in this section.

To collect the samples for the reference dictionaries, matching by tone mapping (**MTM**), as described in [119] and [120], is used to effectively find the best matches under nonlinear tone mapping. Compared to other template matching techniques introduced in [121], **MTM** provides high discriminative capability performance under varying illumination conditions and in the presence of noise. The **MTM** pattern matching scheme was originally proposed to

evaluate the minimum normed distance between a test patch and a candidate image window under all possible tone mappings. In this thesis, a normed distance measure is applied at the segment level using a modified version of the **MTM** scheme. In Appendix A, the **MTM** matching scheme among image segments is described in detail.

In the beginning of the sample collection process, shadow-like segments (**SL**) formed in shadow-like masks **SLM**s and detected from the previous stage, as described in Section 5.2, are fed into the sample collection network. The total number of samples used to construct the dictionaries is based on the number of **CDM**s available for each dataset. For simplicity, an additional two terms, namely, foreground segments and background segments, are used to represent segments in the current frame and background image, respectively. Using grayscale format and for all frames with $K_1 < t \leq K_2$, each foreground segment S^{CF} is compared with its background segment S^{BG} using the **MTM** distance measure:

$$d_u(S^{CF}, S^{BG}) = \frac{1}{N \cdot \text{var}(V^{BG})} \left[\|V^{BG}\|^2 - \sum_j^k \frac{1}{|V^{j,CF}|} \{V^{j,CF} \cdot V^{BG}\}^2 \right], \quad (5.2)$$

$\forall u \in U$

where $u = \{1, 2, \dots, U\}$ is the index of the shadow-like segment in **SLM**, and V^{CF} and V^{BG} are $N \times 1$ column vectors for foreground and background segments. The **MTM** distance measurements are obtained for all image segments belonging to the **SLM** at each time instant. Since the segments in **SLM** are automatically divided into nonuniform image segments (as discussed in Chapter 4), the total number of pixels N belonging to each segment differs. The process

of computing the **MTM** matching and finding the best match and distant segment for each **SLM** is presented in Algorithm 5.2. The two notations, \odot and \oslash , denote element-wise multiplication and division, respectively.

The index of the best match, u_m , among all segments within the **CDM** can be determined by finding the minimum **MTM** distance measure obtained from Equation (5.2). Moreover, the index of the most distant segment u_d with respect to its background segment can also be determined by finding the maximum **MTM** distance measure. The two indexes for the minimum and maximum distances are calculated as below:

$$u_m = \arg \min_u d_u(S^{CF}, S^{BG}) \quad (5.3)$$

$$u_d = \arg \max_u d_u(S^{CF}, S^{BG}) \quad (5.4)$$

Given two segment categories, one for shadows C_{SH} and the other for objects C_O , image segment S_u^{CF} is assigned to a category as follows:

$$S_u^{CF} = \begin{cases} C_{SH}, & \text{if } u = u_m \\ C_O, & \text{if } u = u_d \end{cases} \quad (5.5)$$

Based on Equations (5.3), (5.4), and (5.5), a large quantity of frame sequences may be needed to form a complete reference dictionary since only two foreground segments per **SLM**, with indexes u_m and u_d , are collected to construct the two reference dictionaries. In many cases, it is impossible to have

many frames available for training. In addition, taking a large amount of frames into consideration will result in slowing the process of dictionary construction and the overall detection process. To overcome these problems, an additional four segments (two segments with minimum **MTM** distances and the other two segments with maximum **MTM** distances) are collected per **SLM** to form the two reference dictionaries.

The result of this stage is the collection of a number of segment samples for each reference dictionary. These samples are assessed later to extract a set of valuable feature descriptors per sample, as described in Section 5.4, and are concatenated to construct the two feature dictionaries.

Algorithm 5.2: MTM matching and dictionary sample collection

Input : Current frame masked with SLM , background image
masked with SLM , and index u of image segment S_u

Output: $L(S_u) : S_u^{CF} \in C_{SH}$ or $S_u^{BG} \in C_O$

```

1 For  $u = 1, 2, \dots, U$ 
2   Convert  $S_u^{CF}$  to 1D array  $V^{CF}$ 
3   Convert  $S_u^{BG}$  to 1D array  $V^{BG}$ 
4   Compute intensity sum of  $V^{BG} \Rightarrow V_1 := 1 * V^{BG}$ 
5   Compute intensity sum of squares of
       $V^{BG} \Rightarrow V_2 := 1 * \{V^{BG} \odot V^{BG}\}$ 
6   Compute variance of  $V^{BG} \Rightarrow \text{var}(V^{BG}) := V_2 - (V_1 \odot V_1)$ 
7   For  $j = 1, 2, \dots, k$ 
8     Generate binary vectors  $V^{j,CF}$ 
9   End For
10  Set accumulator  $A := 0$ 
11  For  $j = 1, 2, \dots, k$ 
12    Compute sum of  $V^{j,CF} \Rightarrow |V^{j,CF}| = 1 \cdot V^{j,CF}$ 
13    Convolve  $V^{BG}$  with  $V^{j,CF} \Rightarrow C1 := V^{j,CF} * V^{BG}$ 
14    Compute  $C2 \Rightarrow C2 := (C1 \odot C1) / |V^{j,CF}|$ 
15    Update accumulator  $A := A + C2$ 
16  End For
17  Compute  $d_u(S_u^{CF}, S_u^{BG}) \Rightarrow d_u := (V_1 - A) \odot V_2$ 
18 End For
19  Find best match with an index  $u_m \Rightarrow u_m := \text{minimum}(d_u)$ 
20  Find distant segment with an index  $u_d \Rightarrow u_d := \text{maximum}(d_u)$ 
21   $S_{u_m}^{CF} \in C_{SH}$ 
22   $S_{u_d}^{CF} \in C_O$ 
23 Return
```

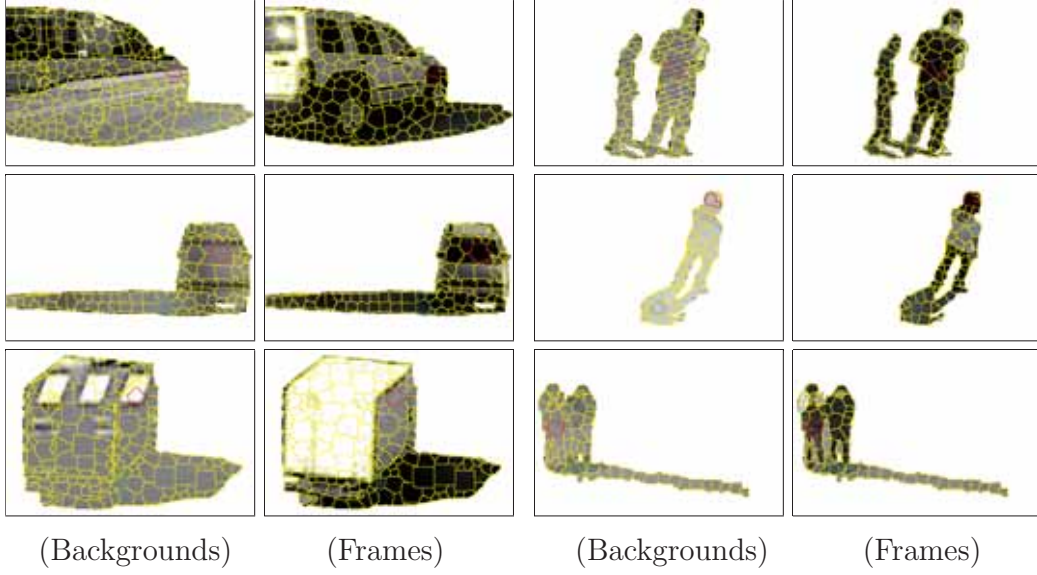


Figure 5.3: Examples of **MTM** matching results for a number of frames and their backgrounds. Best match segments are highlighted in green and most distant segments in red.

Examples of **MTM** matching results for a number of frames and their backgrounds are shown in Figure 5.3. Different sizes of objects, shadows, and image segments are used to test the ability of the **MTM** matching scheme to distinguish between the two classes. Only **SLs** are considered in the matching test. The best match segments, highlighted in green, are successfully detected by the proposed **MTM** matching scheme as parts of cast shadows. In addition, segments with high **MTM** distance measures, highlighted in red, are labeled as parts of the foreground objects.

5.4 Feature Extraction

A number of illumination-invariant features can be used to detect shadows in images. Three powerful features are considered for patches that are occluded by shadows: local color constancy, gradient orientation, and intensity histogram.

a new set of feature descriptors that is capable of distinguishing shadows from objects is constructed by taking advantage of these features. The features are learned and classified at the region level to minimize the execution time without affecting the final results. Since superpixel image segmentation is used [118], the produced image segments contain only parts of the foreground object or parts of its shadow.

At each time instant, t with $t > K_1$, the masked background image MBG_t and the masked current frame MCF_t are available. For simplicity, the general notations of Bg and Fg are used to represent the background frame and the foreground frame at each time instant, respectively. Note that the shadow detection process is discarded if the current frame does not contain any moving object (i.e., no **CDM** is available). The goal is to segment the foreground frame Fg into two main regions, namely, the moving objects and their cast shadows.

Let $S = S_v$ be a segmented region of the shadow-like mask **SLM** in the foreground frame Fg containing N pixels. Let p be a pixel in S , i.e., $S(p) : \{p = 1, 2, \dots, N\}$. Furthermore, let $I_c^k(p)$ be the intensity value of the pixel p within a range of $[0, 255]$ for a primary color $c = \{R, G, B\}$ in either the current foreground frame Fg or the background Bg as determined by $k = \{Fg, Bg\}$. Since the segmentation process produces segments with different sizes, the total number of pixels N may vary by segment. A feature vector $\mathcal{F}(S)$ is then derived for the image segment S , as described below.

5.4.1 Local Structure Descriptor

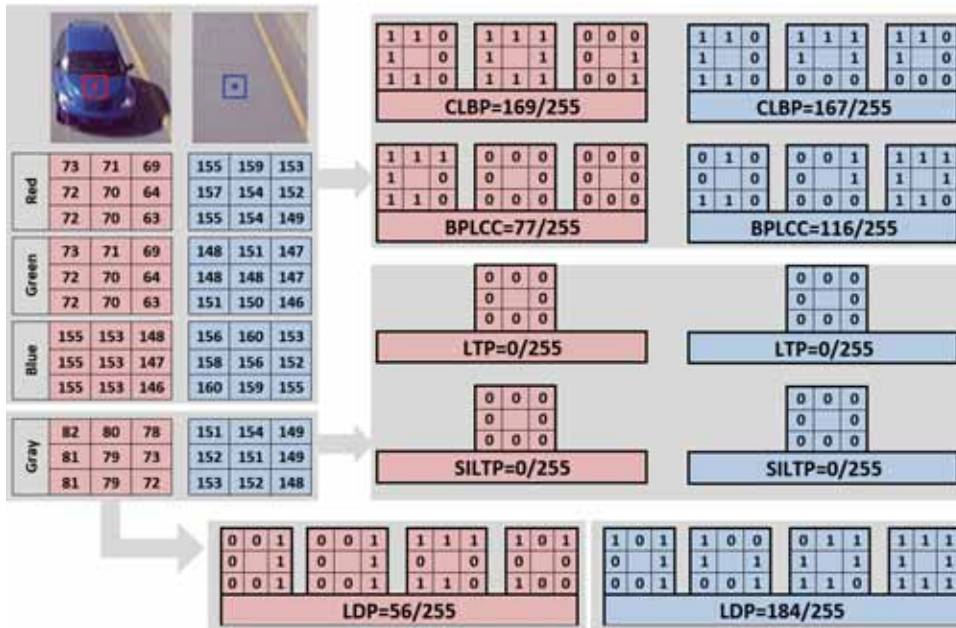
The textures of background surfaces do not change when shadows occur; however, the textures of the foreground objects are often different from those in the corresponding background. The most effective method to describe the texture

properties of a background surface is to determine the intensity differences among neighboring pixels. Local binary patterns (**LBP**) [101] are among the first local texture descriptors designed to model the local texture of object surfaces. **LBP** is simple to implement and has proven to be effective in many computer vision applications, including classification. In addition, **LBP** can easily be extended to multichannel image applications [122]. However, the method is sensitive to noise and is applicable only when texture information is available. Local ternary patterns (**LTP**) [102], local derivative pattern (**LDP**) [123], and local gradient hexa pattern (**LGHP**) [124] were designed and used for face recognition to overcome these problems. **LTP** provides a robust texture descriptor by introducing a threshold among the local intensity comparisons; however, **LTP** is not invariant under linear intensity transformation. Therefore, scaled invariant local ternary patterns (**SILTP**) [103] was designed to address global and local illumination changes.

The problem with these local texture descriptors is that they work only when texture information is available; moreover, local texture descriptors often produce undesirable results when neighboring pixels have similar and/or low reflectance properties. Figure 5.4 shows an overall performance comparison of traditional local texture descriptors under local illumination changes and non-textured surfaces. In Figure 5.4, two small test patches of 3×3 pixels from two different locations of the road surface are taken before and after occlusion by parts of the vehicle and parts of the shadow. The corresponding local texture descriptors, color-LBP (**CLBP**) [122], **LTP** [102], **SILTP** [103], the proposed **BPLCC**, and **LDP** [123], are shown for comparison. All the values have been normalized to 255 for a better comparison. In Figure 5.4 (a), CLBP produces different local texture values (with a score of 196 for shadow and 225 for background), which indicates that the feature descriptor is not efficient under varying



(a)



(b)

Figure 5.4: Comparison of popular local texture descriptors, $CLBP_{P=8}^{\tau=0}$, $LTP_{P=8}^{\tau=5}$, $SILTP_{P=8}^{\tau=0.1}$, $LDP_{P=8}^{\tau=0}$, and the proposed $BPLCC_{P=8}^{\tau=5}$, under: (a) local illumination change, and (b) non-textured surfaces.

illumination. In addition, **CLBP** fails to distinguish the test patch that belongs to the vehicle in the given frame from that of the background (**CLBP**=169 for the object patch and **CLBP**=167 for the background in Figure 5.4 (b)). The **LDP** feature descriptor performs better than **CLBP** in the scenario of nontextured surfaces (**LDP**=56 for the object part and **LDP**=184 for the corresponding background in Figure 5.4 (b)). However, **LDP** extracts unnecessary information, which makes it inefficient under changes in illumination. For example, **LDP** produces two different local texture values, 191 for shadow and 248 for the corresponding background, for the test patch in Figure 5.4 (a). With a score of **LTP**=0 and **SILTP**=0, the two texture descriptors successfully produce the same local texture values for the central pixel, which indicates their efficient performance under varying illumination, as shown in Figure 5.4 (a). However, the **LTP** and **SILTP** feature descriptors fail to distinguish nontextured object patches from their corresponding background (**LTP**=0, **SILTP**=0 in Figure 5.4 (b)).

Therefore, **BPLCC**, a new joint model of spatial structure feature descriptors based on a simplified version of **LBP** combined with interchannel color constancy, is designed to solve the misclassification problems discussed above and to sustain the classification criterion. The proposed **BPLCC** is effective under both cases and produces the same pattern for background patches under various illumination conditions (**BPLCC**=85 for shadow and the corresponding background in Figure 5.4 (a)) and a different one for patches that are occluded by objects (**BPLCC**=77 for object and **BPLCC**=116 for the corresponding background in Figure 5.4 (b)).

Let r , g and b represent three new values of pixel p for the red, green, and blue channels, respectively. The rgb values of p are calculated as

$$\begin{cases} r(p) = \frac{I_R^{FG}(p)}{I_G^{BG}(p) + 1}, \\ g(p) = \frac{I_G^{FG}(p)}{I_B^{BG}(p) + 1}, \\ b(p) = \frac{I_B^{FG}(p)}{I_R^{BG}(p) + 1}. \end{cases} \quad (5.6)$$

Note that 1 is added to the denominator to avoid division by zero. This image transformation takes advantage of the color distribution among the channels and cancels the effect of the reflectance properties. In addition, Equation (5.6) monitors the changes in the irradiance of pixel p in the three channels over time. Since the surface structure of the background does not change when a shadow occurs, the new transformed image gives a luminance ratio in which all neighboring pixels have similar values. The property of color constancy among neighboring pixels does not hold for target objects due to the inequality of the surface reflectance of the foreground object and the background.

The next step is to determine whether two neighboring pixels are consistent in intensity, color and texture. This step is accomplished by comparing the intensity value of the center pixel from one channel with that of the neighboring pixels from the other channel, as shown in Figure 5.5:

$$M_1(p) = \begin{cases} 1, & \text{if } |r(p) - g(q)| \leq TH_{cc} \\ 0, & \text{otherwise} \end{cases} \quad (5.7)$$

$$M_2(p) = \begin{cases} 1, & \text{if } |g(p) - b(q)| \leq TH_{cc} \\ 0, & \text{otherwise} \end{cases} \quad (5.8)$$

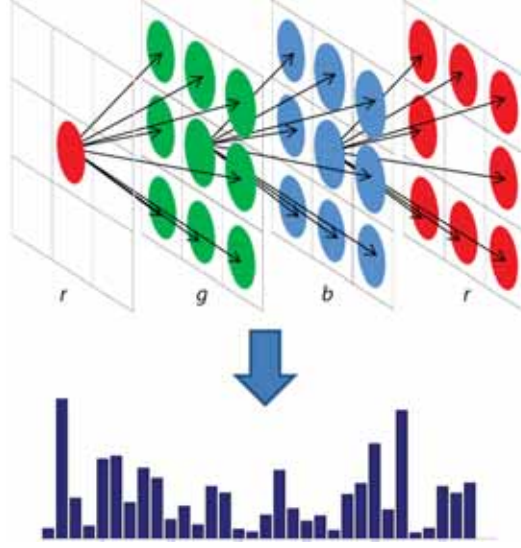


Figure 5.5: Illustration of proposed inter-channel spatial structure feature descriptor. (top) a layer with 3×3 neighbouring pixels is shown for each channel (Red, Green, and Blue) and (bottom) accumulated histogram for all pixels in an image segment with 32-dimensions.

$$M_3(p) = \begin{cases} 1, & \text{if } |b(p) - r(q)| \leq TH_{cc} \\ 0, & \text{otherwise} \end{cases} \quad (5.9)$$

where q is a direct neighboring pixel of the center pixel p . The color constancy threshold TH_{cc} determines whether the two neighboring pixels, p and q , are similar in color and intensity under illumination changes. The value for the color constancy threshold TH_{cc} is learnt in Chapter 6.

The **BPLCC** spatial structure feature descriptor can then be applied to the center pixel p as follows:

$$cc_{J,\Theta}(p) = \sum_{l=1}^3 \sum_{j=0}^{J-1} \{M_l(p) \cdot 2^j\}. \quad (5.10)$$

where $M_l(p) : l = \{1, 2, 3\}$ checks the consistency between the two points in different layers, as modeled in conditional Equations (5.7), (5.8), and (5.9), and $j \in \{0, 1, 2, \dots, J-1\}$ is one of the eight direct neighbors of the center pixel p , equally spaced on a circle of radius Θ .

The segment-level **BPLCC** feature descriptor is thus constructed by concatenating the pixel spatial structure feature descriptor $cc(p)$ of all pixels in image segment S , accumulated as a histogram feature:

$$BPLCC(S) = \{cc(p) : p = 1, 2, \dots, N\}. \quad (5.11)$$

The total number of possible values of pixel p in Equation (5.10) is 766 (within a range of $[0, 755]$); hence, the histogram of BPLCC can be constructed by setting the number of bins to 32. Each of the first 31 bins accommodates 24 levels of possible intensity values, and the last bin accommodates 22 levels.

Figure 5.6 shows the performances of the **LBP**, **LTP**, and **SILTP** feature descriptors and the proposed **BPLCC** using two different image patches, one occluded by an object and the other occluded by shadow. The Euclidean distances **ED** are computed between the two feature vectors of the patch in the current frame and the background for comparison. The proposed **BPLCC** clearly outperforms the other three operators, with a small ED=0.13 when a shadow occurs and a large ED=1.12 when the background is occluded by a foreground object.

5.4.2 Gradient Descriptor

The local gradient distribution is a powerful image feature that has been used in many computer vision applications, including medical image classification [125],

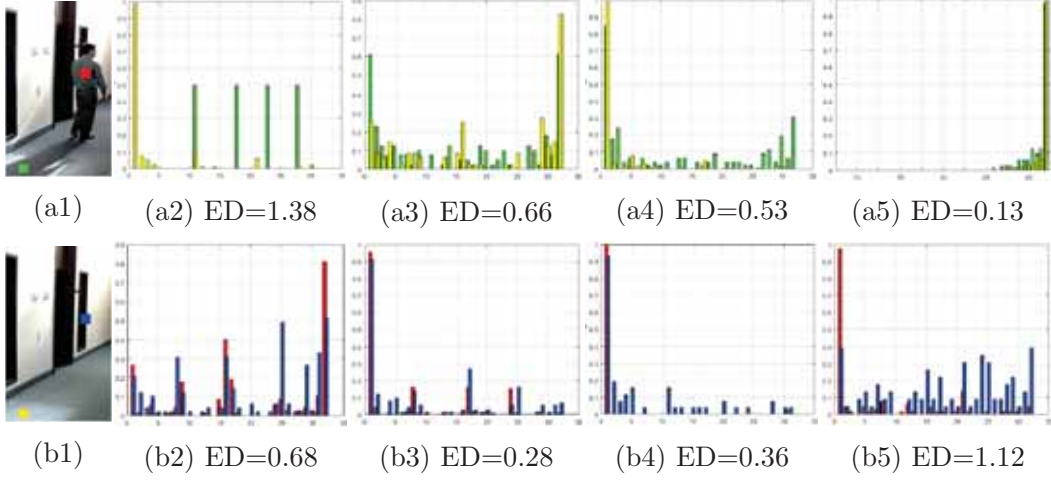


Figure 5.6: Comparison of various feature descriptors for two patches under the test, one occluded by foreground object (highlighted in red in the current frame (a1) and blue in the background image (b1)), and the other by shadow (highlighted in green in the current frame and in yellow in the background image). Euclidean distances are computed and shown for each method. (a2)-(b2) $LBP_{8,1}$, (a3)-(b3) $LTP_{8,1}^5$, (a4)-(b4) $SILTP_{8,1}^{0,1}$, and (a5)-(b5) the proposed $BPLCC$.

human detection [126] and shadow detection [127]. The main limitation of the standard gradient descriptors, such as scale-invariant feature transform **SIFT** [128] and [129], is that a pixel's orientation (magnitude and direction) can easily be affected by varying the illumination. Furthermore, the effect is severe when the underlying area of the pixel is illuminated by a directional light source. On the other hand, the geometric formation of the cast shadow, based on the location of the light source, can provide additional information to design better gradient descriptors.

To overcome the drawbacks mentioned above, a new model for local gradient distributions, called **LGO**, is introduced as a gradient feature descriptor under varying illumination. In the following, the theoretical basis of **LGO** and the feature extraction are explained.

On the basis of the pixel model formula introduced in Equation (3.1) in

Section 3.1.1, the intensity $S(p)$ of a background pixel p in the underlying scene can be expressed as a product of the irradiance $i(p)$ and the surface reflection $r(p)$ of the point:

$$S(p) = i(p)r(p) \quad (5.12)$$

As discussed in Chapter 3, the irradiance term $i(p)$ can be modeled as:

$$i(p) = c_A + T(p)T_D c_D \cos(\theta) \quad (5.13)$$

where θ is the incident angle of the dominant light source, c_D is the intensity of the dominant light source, c_A is the intensity of the ambient light, $0 \leq T_D \leq 1$ is the global light coefficient measuring the amount of light energy available at each time instant, and $0 \leq T(p) \leq 1$ is the local light coefficient determining the amount of light received by the point.

On the basis of Equations (5.12) and (5.13) and under the assumption that (1) the reflectance property $r(p)$ of the point does not change over time and (2) the light source is far from the object (i.e., $\theta = 0$), the luminance ratio $\Delta(p)$ can be obtained for point p at two time instants:

$$\Delta(p) = \frac{c_A + T'(p) \cdot T_D \cdot c_D}{c_A + T(p) \cdot T_D \cdot c_D} \quad (5.14)$$

Note that the three components c_A , T_D , and c_D are independent of the location of pixel p . Equation (5.14) can be further simplified by considering $c_D \gg c_A$:

$$\Delta(p) \approx \frac{T'(p)}{T(p)} \quad (5.15)$$

where $T(p)$ and $T'(p)$ are two local light coefficients determining the percentage of the energy received from the dominant light by point p at two different times.

Now, let q be a neighboring point to p and assume that both are located on a shaded line parallel to the direction of illumination when shadows occur. Based on Equation (5.15), the gradient $\Delta(p, q)$ between the two shaded points can be found:

$$\Delta(p, q) = \frac{T'(p)}{T(p)} - \frac{T'(q)}{T(q)} \quad (5.16)$$

Since the two points are close to each other in the background scene, p and q are expected to receive the same amount of dominant light energy before shadows occur, that is, $T(p) \approx T(q) = T$. Equation (5.16) then becomes:

$$\Delta(p, q) = \frac{1}{T} \cdot [T'(p) - T'(q)] \quad (5.17)$$

The geometric location of the shaded point p at the current time determines whether the point receives a higher amount of light energy (from the dominant and the ambient light sources) than q . The intensity of the dominant light source within a cast shadow region smoothly increases from the object-shadow boundary to the outer boundary [108]. In other words, if point p is located on the outer part of the penumbra and point q is located on the inner part of the penumbra, then $\Delta(p, q) > 0$, which leads to:

$$T'(p) > T'(q) \quad (5.18)$$

Similarly, the difference between any two parts of the shadow from the inner part of the penumbra to the inner part of the umbra gives a positive gradient. Figure 5.7 illustrates this property of the shadow, where the relationship

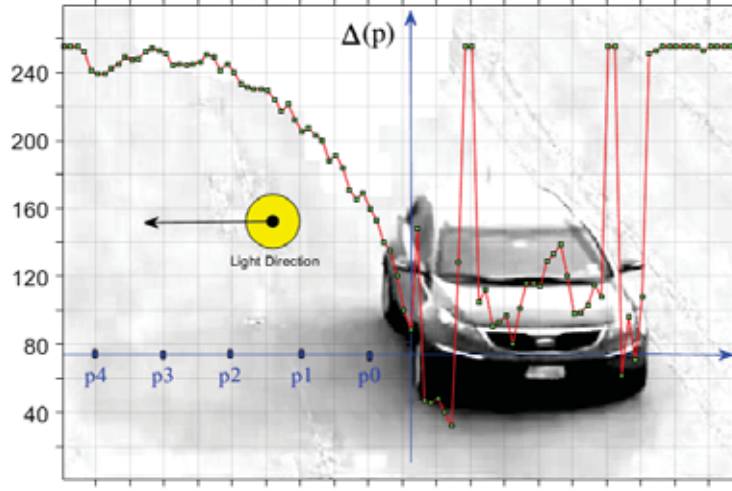


Figure 5.7: Luminance ratio analysis for a line scanned along light direction.

among neighboring points within the shadow region can be established according to Equation (5.18):

$$T'(p_4) > T'(p_3) > T'(p_2) > T'(p_1) > T'(p_0) \quad (5.19)$$

where p_4 , p_3 , p_2 , p_1 and p_0 are points located in the outer part of the penumbra, inner part of the penumbra, penumbra-umbra boundary, outer part of the umbra, and inner part of the umbra, respectively. The decreasing property of the shadow transition does not exist among object points due to the inequality of their reflectance with the corresponding background.

The location of the light source in the image sequence is first estimated using [130] to extract this valuable feature model. The gradient features are then extracted for each color channel independently using **RGB** format and then concatenated to form one **LGO** feature descriptor. An example of the **LGO** feature extraction is shown in Figure 5.8.

On the basis of Equation (5.14), the luminance ratio image ΔI is com-

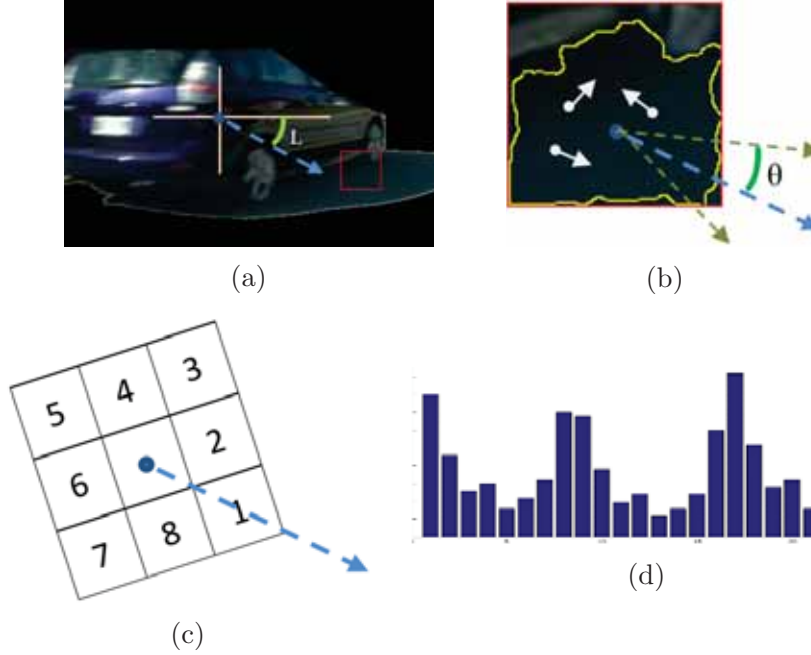


Figure 5.8: The process of proposed **LGO** feature extraction; (a) an estimation of angular direction of the light source L , (b) a magnified section of image segment S highlighted in yellow with different gradient orientations (white arrows) and matching angle θ , (c) 8-radial section with a starting position of histogram, and (d) the **LGO** histogram of S .

puted for each pixel as follows:

$$\Delta I(p) = \frac{I^{FG}(p)}{I^{BG}(p) + 1}. \quad (5.20)$$

$\Delta I(p)$ in the resultant image can have a value within one of two ranges, $0 \leq \Delta I(p) < 1$ or $1 \leq \Delta I(p) \leq 255$. Since the first range corresponds to a very low gradient magnitude compared to the second range, the values in each image segment are normalized with a scaler to have the same global mean (the mean for all the image segments). In this way, we ensure a fair comparison of the gradient magnitudes and directions.

Next, for each pixel in image segment S , the gradient direction $GD(p)$

and gradient magnitude $GM(p)$ are computed as

$$GM(p) = \sqrt{\Delta_X^2 + \Delta_Y^2} \quad (5.21)$$

$$GD(p) = \arctan\left\{\frac{\Delta_Y}{\Delta_X}\right\} \quad (5.22)$$

where Δ_X and Δ_Y are the horizontal gradient (the difference in intensity between pixels in the previous column and the next column) and the vertical gradient (the difference in intensity between pixels in the previous row and the next row) of the central pixel p . The angle of the gradient direction $GD(p)$ varies from 0° to 360° , i.e., $0^\circ \leq GD(p) < 360^\circ$.

Segment S is then divided into Ω radial sections based on the direction of the light source L , each with a directional angle of 2θ , where $\theta = \pi/\Omega$. The first radial section is aligned with the direction of the light source L to cover angles ranging from $L - \theta$ to $L + \theta$ and to accommodate the magnitude values of all the pixels (mainly shadow pixels) that have a similar gradient direction to that of L . The value of each radial direction is calculated as

$$Dr(h) = \sum_{p \in S} GM(p) \quad \text{if } L + (2h - 3)\theta \leq GD(p) \leq L + (2h - 1)\theta \quad (5.23)$$

where h is the index of the radial direction.

The histogram for the **LGO** features is finally constructed by concatenating all the angular directions $Dr(h)$, and the first bin of the histogram contains the magnitude values of all the pixels that have a direction similar to that of L :

$$LGO(S) = \{Dr(h) : h = 1, 2, \dots, \Omega\}. \quad (5.24)$$

Because all three primary colors are used in constructing the gradient model, the total number of bins is $3 \times \Omega$.

5.4.3 Intensity Descriptor

The proposed **BPLCC** and **LGO** feature descriptors discussed above are extracted by establishing a spatial relationship between a pixel and its surroundings without considering the actual intensity values in the image segment. However, pixel intensity can provide additional information for distinguishing the two classes. Therefore, an intensity histogram of the segments $IH(S)$ is computed along with **BPLCC** and **LGO** as an additional feature descriptor. For grayscale format with a range of $[0, 255]$, the total number of bins is empirically set to 16 to accommodate all 16 levels of intensity.

5.4.4 Feature Descriptor

The three features discussed above have varying dynamic ranges; therefore, the three feature vectors are normalized to a common mean. A linear scaler, which is computed by dividing the sum of the feature elements of $BPLCC(S)$ by its feature dimension, is used to align the means of $LGO(S)$ and $IH(S)$ with that of $BPLCC(S)$. All three features are then combined as the final descriptor for the image segment S :

$$\mathcal{F}(S) = \{BPLCC(S), LGO(S), IH(S)\} \quad (5.25)$$

The dimension of the above segment feature descriptor is $H = (32+3\times\Omega+16)$. $\mathcal{F}(S)$ is extracted for all shadow-like segments initially formed in the **SLMs** at $t > K_1$, including all the shadow-like segments collected for constructing the reference dictionaries. The process of computing the three feature descriptors for segment S is given in Algorithm 5.3.

Algorithm 5.3: The proposed segment feature extraction method.

Input : A segment $S \in \mathbf{SLM}$, foreground frame FG , and background image BG in **RGB** colour format.

Output: $\mathcal{F}(S)$

- 1 Obtain luminance ratio of S between two layers of **RGB** using Equation (5.6): $S_{rgb} \leftarrow \mathcal{T}(S_{RGB}^{FG}, S_{RGB}^{BG})$
 - 2 Obtain luminance ratio of S using the same layer in **RGB** using Equation (5.20): $S_{\Delta} \leftarrow \mathcal{T}(S^{FG}, S^{BG})$
 - 3 Divide the segment S into Ω radial sections using Equation (5.23)
 - 4 Obtain gray-scale format for S : $S_{GR}^{FG} \leftarrow \mathcal{T}(S_{RGB}^{FG})$
 - 5 **For** $p = 1, 2, \dots, N$
 - 6 $cc(p) \leftarrow$ Compute pixel spatial structure feature descriptor in a
 - 7 3×3 window centred at p using Equation (5.10)
 - 8 $GD(p) \leftarrow$ Compute pixel gradient direction using Equation (5.22)
 - 9 $GM(p) \leftarrow$ Compute pixel gradient magnitude using Equation
 - 10 (5.21)
 - 11 **End For**
 - 12 Compute histogram of BPLCC for S using Equation (5.11):
 $BPLCC(S) \leftarrow cc(p) : p = 1, 2, \dots, N$
 - 13 Compute histogram of LGO for S using Equation (5.24):
 $LGO(S) \leftarrow D_r(h) : h = 1, 2, \dots, \Omega$
 - 14 Compute histogram of IH for S using 16 bins:
 $IH(S) \leftarrow S_{GR}^{FG}(p) : p = 1, 2, \dots, N$
 - 15 Combine all feature descriptors for S :
 $\mathcal{F}(S) \leftarrow \{BPLCC(S), LGO(S), IH(S)\}$
 - 16 **Return**
-

5.5 Segment Classification

The second stage of shadow detection is classification, where the decision is based on the derived features. In many shadow detection methods, the classification process uses adaptive thresholds [62] without the aid of a classifier, while other methods use standard classifiers to improve the detection results. Widely used classifiers include SVMs [131], Bayesian classifiers [132], nearest neighbor classifiers [133], and statistical learning classifiers [134]. Compared with other methods, SVM often performs well for discriminating features of multiclass. However, SVM is a monolithic classifier whose performance is adversely affected by large intraclass variation [135].

The objective of segment classification is to assign each image segment S to one of two image categories: shadow or object. The sparse approximation method for classification, which has proven to be effective in addressing a variety of classification issues [125], is adopted.

Let C_{SH} and C_O represent the image categories of shadow and object, respectively. The image segment S is intended to be assigned a category label $\mathcal{L}(S) \in \{C_{SH}, C_O\}$. Q image segments of category l are assumed to have been collected and saved in each dictionary reference (as discussed in Section 5.3) to form the corresponding training set $\{S_q : q = 1, 2, \dots, Q\}$. A feature dictionary matrix D_l is then constructed by concatenating the feature vectors in a column-by-column manner:

$$D_l = \{\mathcal{F}(S_q) : q = 1, 2, \dots, Q\} \in \mathcal{R}^{H \times Q} \quad (5.26)$$

where $\mathcal{F}(S_q)$ is the feature vector with dimension H ($H < Q$) for image segment S_q . Two dictionaries are constructed for categories C_{SH} and C_O . The number

of feature vectors Q in a dictionary depends on the size of the training set; therefore, the dictionaries can have different values of Q .

Given image segment S , a sparse-regularized linear model is used to find the best approximation vector $\mathcal{F}'_l(S)$ for the feature vector $\mathcal{F}(S)$ from the feature dictionary D_l :

$$\omega_l = \arg \min_{\omega_l} \| \mathcal{F}(S) - D_l \omega_l \|_2^2 \quad \text{s.t.} \quad \| \omega_l \|_0 \leq E \quad (5.27)$$

$$\mathcal{F}'_l(S) = D_l \omega_l \quad (5.28)$$

where ω_l is a sparse coefficient vector with E nonzero elements. The above optimization problem can be solved by the orthogonal matching pursuit (**OMP**) algorithm¹ [136].

The difference between the actual feature vector $\mathcal{F}(S)$ and its best approximation $\mathcal{F}'_l(S)$ in each category $l \in \{C_{SH}, C_O\}$ is calculated to determine a category label for the image segment S :

$$d_l(S) = \| \mathcal{F}(S) - \mathcal{F}'_l(S) \|_2 \sigma(\omega_l) \quad (5.29)$$

where $\sigma(\omega_l)$ is the standard deviation of the nonzero elements in ω_l . The above optimization penalizes a large value of $\sigma(\omega_l)$ because it indicates disagreement among image segments in a feature dictionary.

S is assigned a category label corresponding to the minimum difference

¹The OMP package can be downloaded from: <http://www.cs.technion.ac.il/~ron-rubin/software.html>

obtained:

$$\mathcal{L}(S) = \arg \min_l d_l(S) \quad (5.30)$$

The standard form of the sparse-representation-based classification discussed above and used in [125] is often affected by intraclass variation, which can lead to misclassification. A new model of sparse-representation-based classification that accounts for more than a single class estimate from the dictionary of each class is introduced to overcome this problem and to improve the classification results. A multi-independent sparse representations is used instead of a single representation; thus, the class of the segment under test is assigned by deriving a standard model of the probability of occurrence among the independent class assignments.

In practice, the sparse-regularized linear model used in Equation (5.27) and Equation (5.28) is iterated $j \in \{1, 2, \dots, J\}$ times to find the J best approximation vectors $\mathcal{F}'_{l,j}(S)$ for the feature vector $\mathcal{F}(S)$ from each dictionary. After each iteration, the two dictionaries are updated by removing the elements that have been selected as a class feature representation by the previous iterations. The classification is then performed using Equation (5.29) and Equation (5.30), and the result is mapped to $\{0, 1\}$ for each class, where $C_l(S) = 1$ if $S \in C_l$ and $C_l(S) = 0$ otherwise. The final classification of S can thus be obtained by maximizing the number of occurrences of each independent class:

$$\mathcal{L}(S) = \arg \max_l \sum_{j=1}^J C_{l,j}(S) \quad (5.31)$$

The proposed multi-independent sparse-representation-based classification (**MSR**) reduces the risk of misrepresentation of feature similarities among categories and provides a robust class estimate of the segment under test. Figure

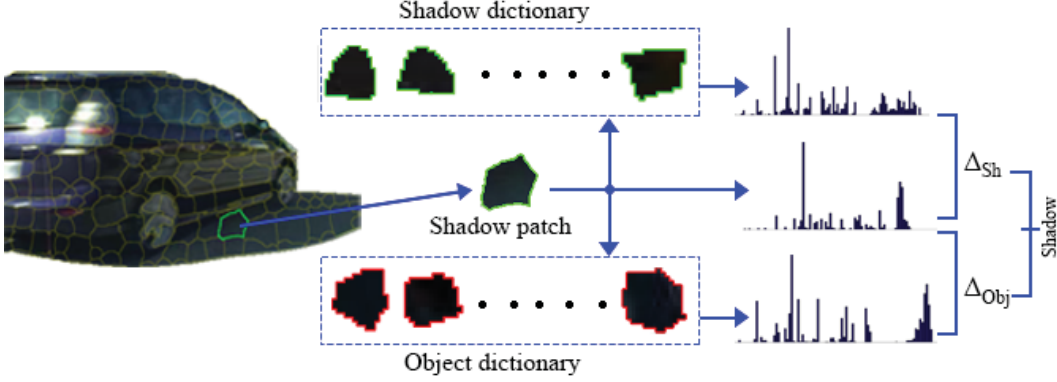


Figure 5.9: An example of segment-level classification for a test image segment occluded by shadow in the current frame.

5.9 illustrates the overall classification process for a shadow patch taken from the current frame. Two classes of image segments, one for objects (highlighted in red) and the other for shadows (highlighted in green) are shown along with a test image (belonging to the shadow class). Feature vectors are generated for image segments using the proposed feature descriptors. The classification is performed using similarity measures between the test image feature vector and its best local estimate from each reference dictionary.

5.6 Region-level Classification

The final step is to establish inter-region dependencies among all neighboring segments to assign a class to each segment labeled as dark in Section 5.2. The idea of inter-region dependencies is based on the process of region-growing segmentation. The process is repeated until no more region dependencies exist among neighboring segments. In many region-growing methods, a homogeneity test is performed to determine whether the features (such as mean intensity, edges, and textures) of any two neighboring regions are similar. The differences in feature

descriptors of the two regions are compared with the standard deviations of the two regions to determine whether to merge or split the two regions. Since useful features cannot be extracted from a dark segment, the feature-based test fails to assign correct labels to the segment. Alternatively, examining the spatial connectivity (also called connected components) between adjacent segments is a good way to achieve higher labeling accuracy. Spatial connectivity can be applied to the internal boundary points that the region shares with surrounding regions.

Let S_d be the dark segment under test, and let S_o and S_{sh} be segments belonging to the moving object and shadow, respectively. The aim is to find all possible paths (if any exist) from S_d to S_o and S_{sh} through the boundary shared by the two regions. The total number of boundary points that S_d shares with each class is then used to determine whether the region belongs to a class:

$$\mathcal{L}(S_d) = \begin{cases} C_O, & \text{if } |B_o| \neq 0 \text{ and } |B_o| \geq |B_{sh}| \\ C_{SH}, & \text{if } |B_{sh}| \neq 0 \text{ and } |B_{sh}| > |B_o| \\ \text{Undefined}, & \text{otherwise} \end{cases} \quad (5.32)$$

where $|B_o|$ is the total number of boundary points between the test segment S_d and the object segment(s), i.e., $|B_o| = S_o \cup S_d$. $|B_{sh}|$ is the total number of boundary points between the test segment S_d and the shadow segment(s), i.e., $|B_{sh}| = S_{sh} \cup S_d$. The classification process for a segment with no shared boundaries with S_o and S_{sh} is skipped to proceed to the next segment.

To detect boundary pixels between two adjacent segments, morphological dilation followed by the logical AND operation can be performed between the two segments to determine the region of intersection. The number of boundary points between the test segment and all neighboring segments belonging to the

Table 5.1: Possible conditions associated with region growing.

$ B_{sh} $	$ B_o $	$ B_d $	Case	$\mathcal{L}(S_t)$
=0	=0	> 0	S_t is surrounded by S_d s only	Undefined (skip)
=0	> 0	=0	S_t is surrounded by S_o s only	Object
=0	> 0	> 0	S_t is surrounded by S_o s & S_d s	
> 0	=0	=0	S_t is surrounded by S_{sh} s only	Shadow
> 0	=0	> 0	S_t is surrounded by S_{sh} s & S_d s	
> 0	> 0	=0	S_t is surrounded by S_{sh} s & S_o s	Object if $ B_o \geq B_{sh} $
> 0	> 0	> 0	S_t is surrounded by S_{sh} s, S_o s, & S_d s	Shadow if $ B_o < B_{sh} $

* $|B_d| = S_d \cap S_t$ is the total number of boundary points between the test segment and a dark segment S_d .

same class can then be computed. The whole process of labeling dark segments is performed iteratively and is terminated when all undefined segments are labeled. The labeling process of the dark segments is given in Algorithm 5.4.

Table 5.1 presents all possible conditions that can occur in the proposed region-growing method. A category label is assigned to the test segment S_t based on the given conditions.

Figure 5.10 shows a magnified section of the segmentation classification of a frame obtained in the previous stage of the proposed method. The section shows a case in which a test segment S_t is surrounded by more than one labeled segment from each class. The segment also shares some boundary points with two other dark segments S_{d1} and S_{d2} . In the first iteration, the test segment S_t is classified as part of the shadow since it shares most of its boundary points with shadow segments S_{sh1} , S_{sh2} and S_{sh3} (the results are shown in Figure 5.10-(b)). In the second iteration, S_{d1} is labeled as an object since it shares some boundary points with S_{o1} , while S_{d2} is labeled as shadow as it is surrounded by the connected border of shadow segments S_t and S_{sh3} (the results are shown in

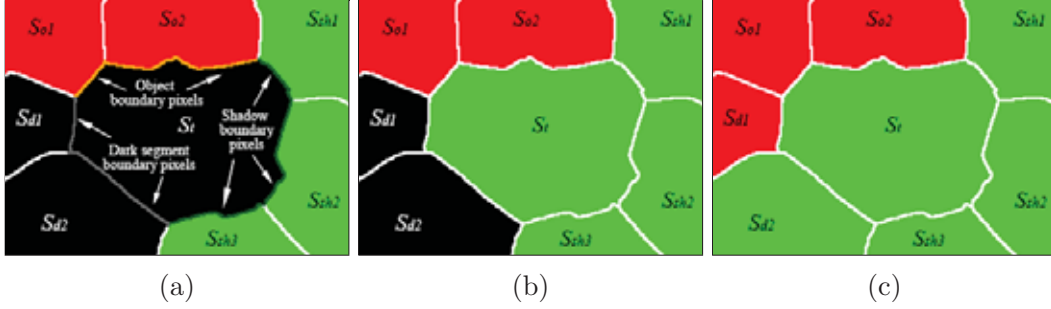


Figure 5.10: An example of the process of region-level classification. (a) a test image segment S_t surrounded by a two object segments S_o , three shadow segments S_{sh} , and two dark segments S_d , (b) the classification result after first iteration, and (c) the classification result after second iteration.

Figure 5.10-(c)).

5.7 Summary

This chapter presents a novel method for detecting moving shadows under possible problematic situations. In this method, two illumination-invariant features, namely, **BPLCC** and **LGO**, are introduced along with the intensity histogram **IH**. These features are extracted from image patches and are used to construct two overcomplete dictionaries for objects and shadows, respectively. Samples for the two reference dictionaries are collected by applying a matching by tone mapping **MTM** pattern matching scheme on shadow-like segments. Given a new image patch, its best approximation for a number of iterations is found from each dictionary. For each iteration, an independent class assignment is performed by finding the distance from the reference dictionaries. The patch is then assigned to a class based on its probability of occurrence. In the final stage of the proposed method, spatial connected component labeling is applied to assign a class to dark segments.

The three proposed feature descriptors are selected carefully to extract

Algorithm 5.4: Region-level classification

Input : Dark segments S_d in DSM mask, object segments S_o in object segment mask OSM , and shadow segments S_{sh} in shadow segment mask SSM

Output: $\mathcal{F}(S_d)$

- 1 Set: number of iterations = number of dark segments D
- 2 **For** $Itr = 1, 2, \dots, D$
- 3 Initialize: $z = 0$
- 4 **For** $d = 1, 2, \dots, D$
- 5 Compute $Mask(S_d)$ with the same size of DSM containing
- 6 only dark segment S_d
- 7 $Mask(S_d) \leftarrow$ Apply morphological dilation on $Mask(S_d)$
- 8 Compute $|B_o|$: $|B_o| \leftarrow Mask(S_d) \cap OSM$
- 9 Compute $|B_{sh}|$: $|B_{sh}| \leftarrow Mask(S_d) \cap SSM$
- 10 **If** $|B_o| = 0$ and $|B_{sh}| = 0$
- 11 $z \leftarrow z + 1$
- 12 **Else If** $|B_o| \geq |B_{sh}|$
- 13 $\mathcal{F}(S_d) = C_O$
- 14 **Else** $|B_o| < |B_{sh}|$
- 15 $\mathcal{F}(S_d) = C_{SH}$
- 16 **End If**
- 17 **End For**
- 18 **If** $z = 0$
- 19 Break the **For** loop
- 20 **End If**
- 21 **End For**
- 22 **Return**

valuable information under possible problematic situations that occur for shadows. For example, **BPLCC** detects micro patterns of the underlying surface structure using a new joint model of **LBP** and interchannel color constancy. Thus, the **BPLCC** feature descriptor can distinguish the two classes, objects and shadows, in situations where texture information is not available. On the other hand, the **LGO** feature descriptor can distinguish the two classes in cases of chromatic shadows or color blending. In addition to **BPLCC** and **LGO**, the intensity of individual pixels in the segment, **IH**, provides additional support for

the labeling process and overcomes the problem of camouflage.

In Chapter 6, the overall performance of the proposed shadow detection method and the feature descriptors is evaluated individually with respect to the problem domain and compared with the performance of other state-of-the-art algorithms. Furthermore, the performance of the overall proposed framework is evaluated and compared with that of a number of well-known algorithms proposed for shadow detection.

Chapter 6

Results and Discussion

6.1 Introduction

Chapter 4 presented a background subtraction method for creating **CDMs** for further processing. Chapter 5 presented a method to detect moving shadows from these **CDM** under possible problematic situations discussed in Chapter 3.









In this chapter, overall results from both methods are presented in relation to the research aims developed in Chapter 1. These results are analyzed and evaluated using manually segmented ground truth images. Three main experiments are conducted to evaluate the methods and features proposed in this thesis. The first experiment tests the accuracy of the proposed complete framework (change detection and shadow detection) in the presence of problematic situations. In addition, the results from the proposed framework are compared with those obtained by three popular state-of-the-art moving object detection methods. In the second experiment, the proposed moving shadow detection features, **BPLCC** and **LGO**, are examined and compared with the most popular image feature descriptors. In the final experiment, the proposed shadow detec-

tion method (the proposed feature descriptors and the multi-independent sparse representation classifier) is evaluated in terms of shadow detection rate, shadow discrimination rate, accuracy, and stability. In addition, the quantitative performance of the proposed shadow detection method is compared with that of the current state-of-the-art moving shadow detection algorithms.

All the experiments in this chapter, including the implementation of all the methods, features, and the production of the results, were conducted using a PC running MATLAB on an AMD FX(tm)-8150 Eight-Core Processor at 3.6 GHz with 16 GB RAM.

The remainder of this chapter is organized as follows. First, technical details of the datasets are briefly discussed in Section 6.2. The five main evaluation metrics used to evaluate the performance are introduced in Section 6.3. Section 6.4 explains the methods involved in determining the major parameters used for the proposed method. In Section 6.5, pixel-level qualitative and quantitative results from the proposed framework are analyzed and evaluated. In Section 6.6, the segment-level qualitative and quantitative performance of the proposed *BPLCC* and *LGO* feature descriptors is analyzed, evaluated, and compared with that of the most popular feature descriptors. Section 6.7 provides an extensive qualitative and quantitative analysis of the results of the proposed moving shadow detection method and the most common and recent works in the area. Finally, a summary of the results is presented in Section 6.8.

Table 6.1: Technical details of used sequences.

Sample								
Name	<i>Campus</i> [24]	<i>Cubicle</i> [25]	<i>WBS</i> [24]	<i>PPL</i> [25]	<i>Scam</i> [27]	<i>HwyI</i> [24]	<i>HwyIII</i> [24]	<i>Bungalows</i> [25]
Length (frames)	2947	7400	2358	1199	459	733	2227	1700
Original frame size	288×352	240×352	288×384	244×380	240×320	240×320	240×320	240×360
Scene type	Outdoor	Indoor	Indoor	Outdoor	Outdoor	Outdoor	Outdoor	Outdoor
Object class	Variable	People	People	People	People	Vehicle	Vehicle	Vehicle
Object size	Variable	Medium	Small	Large	Small	Variable	Variable	Large
Size	Large	Small	Small	Large	Large	Large	Small	Large
Strength	Weak	Strong	Weak	Strong	Strong	Strong	Strong	Strong
Self-shadow	✓	✓	-	✓	✓	-	✓	✓
Umbra	✓	✓	-	✓	✓	✓	-	✓
Penumbra	✓	✓	✓	✓	✓	✓	✓	✓
Achromatic shadow	✓	✓	-	-	-	✓	✓	✓
Chromatic shadow	-	-	-	✓	-	✓	✓	✓
Shadow color blending	-	-	✓	-	-	-	-	-
Camouflage	-	✓	✓	✓	✓	✓	✓	✓
Non-textured surfaces	✓	-	✓	✓	✓	✓	✓	✓
Dark surfaces	-	-	-	✓	✓	✓	-	-

Shadow

6.2 Datasets

Three publicly available datasets containing eight challenging sequences for indoor and outdoor environments are used in the experiment. The technical details of these sequences are shown in Table 6.1. These sequences are classified with respect to the associated problematic situations discussed in Chapter 3.

To initialize the background model, construct the reference dictionaries, and learn the parameters, the datasets are divided into three groups, with each group having a different number of frames. The first group contains the first K_1 video frames from the whole datasets and is used to obtain the background model. The second group of datasets contains $K_2 - K_1$ video frames taken from the whole datasets and is used to construct the two reference dictionaries. A number of frames is taken from the first K_2 video frames of each dataset to learn the major parameters. The last group of datasets, which contains most of the frames from the entire datasets, is used to test the overall proposed method. More details about these sequences are given below.

- *Campus*:

This sequence shows an outdoor environment with various moving object classes and sizes. The sequence represents a condition where shadows are relatively large and weak due to the presence of multilight effects. *Campus* is publicly available to download from [24].

- *Cubicle*:

Cubicle shows people walking in a quiet room with stable light conditions. The sequence represents a strong camouflage problem between parts of the walking people and the background. *Cubicle* is chosen to assess the ability of the proposed method under this problematic situation. This sequence,

including input frames and ground truth, is available to download [25].

- *WBS*:

WBS is a video of a shopping mall corridor where people are walking alone and in groups, entering and exiting side shops. The sequence is challenging because of shadow color blending, which results from strong background reflection. This sequence, including input frames and associated ground truth images, is available to download [24].

- *PPL*:

The sequence *PPL* is an outdoor sequence that contains people walking under a large shaded area. This sequence is due to the presence of strong chromatic shadows, foreground-background camouflage, and nontextured dark surfaces. This sequence can be downloaded from [25].

- *Seam*:

Seam shows another outdoor environment where a camera is mounted in front of a building to record people's activities during a sunny day. Distinguishing moving objects from their shadows is a challenging task in this sequence due to foreground-background camouflage, nontextured surfaces, and dark surfaces. This sequence, including input frames and ground truths, is available to download [27].

- *HwyI*:

This sequence represents noisy and heavy traffic conditions where large and strong shadows cause multiple vehicles to be combined into a single *CDM*. The sequence contains all the problematic situations except shadow color blending. This sequence can be downloaded from [24].

- *HwyIII*:

HwyIII provides another example of a traffic scene with various problematic situations. The sizes of the vehicles and their shadows are relatively smaller than in *HwyI*. This sequence, including input frames and a ground truth, is available to download [24].

- *Bungalows*:

This sequence is recorded with a low-angle camera mounted on the side of the road. The sequence contains large vehicles passing in front of the camera. *Bungalows* is a good example of nontextured surfaces and chromatic shadows. This sequence can be downloaded from [25].

6.3 Evaluation Metrics

Five quantitative metrics, namely, recall, precision, shadow detection rate (η), shadow discrimination rate (ξ), and accuracy, are used in the performance evaluation [23][112]. The first two quantitative metrics are adopted to measure the accuracy of the proposed change detection method:

$$\text{Recall} = \frac{TP_{CD}}{TP_{CD} + FN_{CD}} \quad (6.1)$$

$$\text{Precision} = \frac{TP_{CD}}{TP_{CD} + FP_{CD}} \quad (6.2)$$

where recall measures the percentage of detected true positives TP_{CD} compared to the total number of true positives in the ground truth, that is, the sum of all the pixels correctly detected as changes in the background (TP_{CD}) and the pixels

incorrectly detected as background (FN_{CD}). Precision gives the percentage of detected true positives TP_{CD} compared to the total number of pixels detected by the method, i.e., the sum of all the pixels correctly detected as changes in the background (TP_{CD}) and the background pixels incorrectly detected as changes (FP_{CD}).

The last three quantitative metrics evaluate the proposed moving shadow detection method with respect to the ground truth. The metrics are computed as below:

$$\eta = \frac{TP_S}{TP_S + FN_S} \quad (6.3)$$

$$\xi = \frac{TP_O}{TP_O + FN_O} \quad (6.4)$$

$$\text{Accuracy} = \frac{TP_O}{TP_O + FN_O + FN_S} \quad (6.5)$$

where the shadow detection rate (η) measures the percentage of shadow points (TP_S - number of true positive shadow points and FN_S - number of false negative shadow points) that are correctly detected as shadows (TP_S), and the shadow discrimination rate (ξ) measures the percentage of object points (TP_O - number of true positive object points, and FN_O - number of false negative object points) that are correctly detected as objects (TP_O). Accuracy is adopted in conjunction with the shadow detection rate and shadow discrimination rate to evaluate the overall performance of the proposed moving shadow detection method with respect to foreground moving objects. This evaluation metric is necessary since the main aim of change detection and shadow detection is to extract target objects from their shadows and from the background scene. Accuracy is measured

as the ratio of correctly detected object points (TP_O) to the sum of the number of actual object points (TP_O) (taken from ground truth), the number of object points incorrectly detected as shadows (FN_O), and the number of shadow points incorrectly detected as objects (FN_S).

In addition to the above five quantitative metrics, another important metric, namely, detection stability, is considered for each shadow detection method. Stability monitors the detection accuracy over time and determines whether the accuracy is relatively constant. In terms of moving shadow detection, high detection accuracy for certain conditions (or frames) and low accuracy for others is unacceptable. Detection stability is computed as below:

$$\text{Stability} = 1 - \frac{\max (\text{Accuracy}) - \min (\text{Accuracy})}{100} \quad (6.6)$$

where $\max (\text{Accuracy})$ and $\min (\text{Accuracy})$ are the maximum and minimum detection accuracies obtained for a number of frames in the sequence. Stability is used to evaluate the shadow detection methods proposed in Section 6.7.3.

6.4 Parameter Settings

Eleven major parameters are used to determine the overall accuracy of the proposed moving shadow detection: number of frames K_1 used to model initialization, distance threshold TH_{BG} , size of the neuronal network mapping n , best match threshold TH_p , learning rate TH_α , segment level threshold TH_S , color constancy thresholds TH_{cc} , number of radial sections Ω , dictionary size Q , sparse constant E , and number of iterations J . Various settings of these parameters are assessed in the experiments. Note that selecting inadequate parameters may

result in low performance of the proposed method. The main parameters are discussed below.

- Number of frames K_1 :

The number of frames K_1 used to initialize the background image in the proposed change detection method is set based on the scene's characteristics and the number of static initial frames available for each sequence. To estimate the reference background BG , K_1 is set to 60 for *Campus*, *Cubicle*, *PPL*, and *Seam* since these scenes are uncongested. Sequence *WBS* is a crowded scene; hence, K_1 is set to 260 to ensure that the background is clear from moving people. For the three traffic scenes, *HwyI*, *HwyIII* and *Bungalows*, K_1 is set to 40, 45, and 120, respectively.

- Distance threshold TH_{BG} :

Based on the analysis, the distance threshold TH_{BG} used in Equation (4.4) (Section 4.3.1) is set to 30 for all sequences to avoid including foreground patterns in the background initialization.

- Size of the neuronal network n :

The size of the neuronal network mapping n used in Equation (4.24) (Section 4.3.3) is set to 3×3 for all sequences to minimize the computational complexity.

- Best match threshold TH_p :

The threshold TH_p used in Equation (4.27) (Section 4.3.4) determines whether the best matching pixel p_m with weight vector z_m belongs to a foreground object or background scene. Since the *HSV* feature descriptor is normalized to a common range of $[0, 255]$, z_m can have any value within the range of $[0, 510]$. In the experiment, lower values of TH_p are

chosen for congested sequences to allow the method to detect more foreground points in the current frame. By contrast, higher values of TH_p are chosen for uncongested sequences to ensure that the background model is maintained. Thus, TH_p is set to 50 for *WBS*, *HwyI*, *HwyIII*, and *Bungalows* and to 120 for the remaining uncongested sequences.

- Learning rate TH_α :

The learning rate TH_α used in Equation (4.31) (Section 4.3.5) is set to 0.05 for sequences *Campus*, *WBS*, and *Bungalows* since the background points are occluded by the same (or similar) foreground pattern for more than a frame. Meanwhile, TH_α is set to 0.01 for the remaining sequences.

- Segment level threshold TH_S :

The segment level threshold TH_S used in Equation (4.33) (Section 4.3.6) in the final classification is set to 0.6. Thus, a segment is classified as foreground if more than 60% of the total pixels are labeled as foregrounds in the previous stage of the algorithm.

- Color constancy threshold TH_{cc} :

In Figure 6.1 (top), the main parameter for **BPLCC**, namely, the color constancy threshold TH_{cc} (used in Equations (5.7), (5.8), and (5.9) in Section 5.4.1), is learned separately using 1890 image segments from the first K_2 frames of the datasets. The selected parameter is considered to be global since different scene conditions are considered. A TH_{cc} to 0.2 maintains a good balance between the shadow detection rate and the discrimination rate in all the datasets. Higher values than $TH_{cc} = 0.2$ provide better shadow detection results (more pixels are detected as shadows); however, the higher values lead to more misclassification because most object pixels with dark surface reflectance and low texture properties are classified as

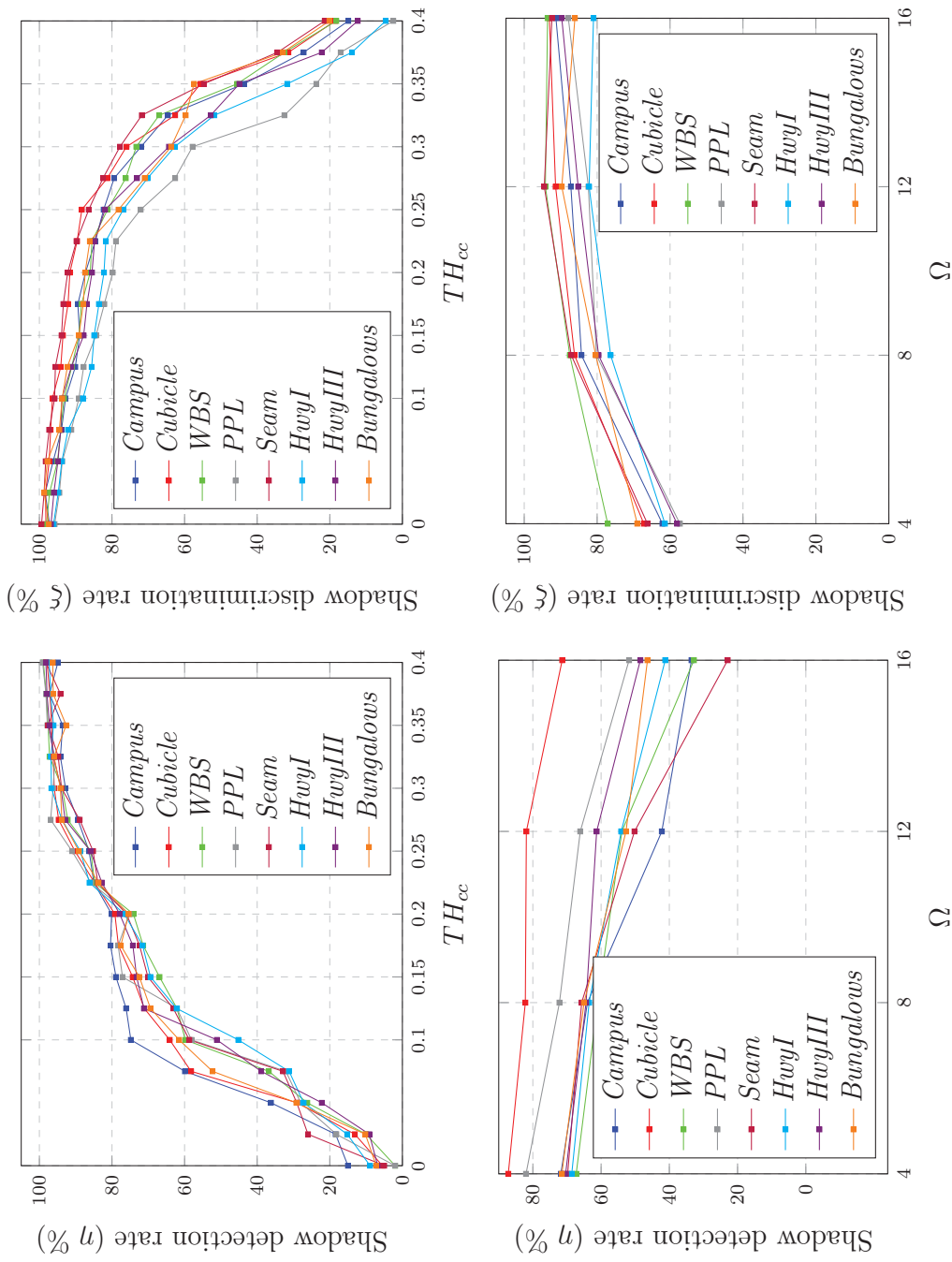


Figure 6.1: Shadow detection rate and shadow discrimination rate versus various parameter settings; (top) Shadow detection rate and shadow discrimination rate versus various color constancy threshold TH_{cc} , (bottom) shadow detection rate and shadow discrimination rate versus various number of radial sections Ω .

shadows (see shadow discrimination rate in Figure 6.1(top)).

- Number of radial sections Ω :

Figure 6.1 (bottom) shows the effects of using different numbers of directional angles Ω for computing the gradient features, as formulated in Equations (5.23) and (5.24) (Section 5.4.2). The results indicate that setting Ω to 8 (i.e., $\theta = \frac{\pi}{\Omega}$) provides satisfactory detection results for all the datasets.

- Dictionary size Q :

A total of $K_2 - K_1$ frames are taken from each dataset to construct the reference dictionaries. Taking small segments into consideration, a total of 816 object segments ($C_O=816$) and 672 shadow segments ($C_{SH}=672$) were collected from each dataset. For both classes, the size of the dictionary is much larger than the feature dimension ($Q \gg H$).

- Sparse constant E :

Based on the experimental analysis, setting the sparse constant E , which is used in Equation (5.27) in Section 5.5, to 10 achieves a good balance between the classification accuracy and computational efficiency. Note that the size of each dictionary should be sufficiently large such that at least 10 image segments can be sparsely selected.

- Number of iterations J :

The number of iterations J used in Equation (5.31) (Section 5.5) for the final classification is set to 3 to provide good classification accuracy between the two categories. A larger number of iterations could be selected; however, $J=3$ is adequate to achieve good detection results with low computational effort.

6.5 Evaluation of the Proposed Framework

The performance of the overall proposed change detection method and proposed shadow detection method is evaluated under various environments using all the datasets listed in Table 6.1. Three well-known and comprehensive shadow detection methods are examined: the method proposed by Lenone et al. [137] (GABOR), the method proposed by Maddalena et al. [112] (SOBS), and the method proposed by Amato et al. [62] (LCC). These methods are described in greater detail in Section 6.5.1. The qualitative performance and quantitative performance on each dataset are discussed in Section 6.5.2. Finally, the quantitative performance of all the methods with respect to all problematic situations is analyzed in Section 6.5.3.

6.5.1 State-of-the-art of Comprehensive Methods

Many moving object detection methods have been proposed; however, only a few include moving shadow detection as the final main stage of the overall system. Three well-known methods that provide all the required stages for detecting shadows are selected for comparison in this thesis. The recommended parameters specified in each reference paper are used to ensure a fair comparison.

The first method selected for comparison was proposed by Leone et al. [137]. In the method, the intensity ratio and adaptive background difference are used to detect changes in background scenes. To detect shadow points in the

CDM, the similarities between texture feature descriptors, initially extracted using standard Gabor functions, are examined. This method is selected for comparison since standard Gabor functions are used instead of special-purpose feature descriptors.

The second method selected for comparison was proposed by Maddalena et al. [112]. The method works at the pixel level to detect changes in the background and eliminate shadows. In this method, self-organization through neuronal networks is used to find the best match of the current sample pixel belonging to the incoming frame. The method is a good example from the pixel-based category of methods and, hence, is used for comparison in this thesis.

Amato et al. [62] proposed a moving shadow detection method based on local constancy detection. Four similarity measures, namely, angular similarity, intensity similarity, angular neighborhood similarity, and intensity neighborhood similarity, are used to detect changes in the background. Moreover, the gradient constancy and local color consistency are analyzed to detect shadow points. This method is selected for comparison as it belongs to the region-based category of shadow detection methods.

6.5.2 Qualitative Performance

In this section, the qualitative performance of the methods is analyzed and compared using frames taken from all the sequences given in Table 6.1. Two types of morphological operation [117], namely, 2D hole filling and area opening, are applied to enhance the final change detection results obtained from all the methods. The 2D hole filling morphological operation is applied to fill small holes within the detected foreground objects. Meanwhile, the area opening morphological operation is used to remove all small blobs detected as foreground.

Figure 6.2 shows a frame containing a soft achromatic shadow and relatively stationary background scene from the *Campus* dataset. The selected frame tests the ability of each method under global illumination changes, nontextured surfaces, and camouflage. The associated background frame, the change detection ground truth image (CDM Ground Truth), and the ground truth (GT) image for shadow detection are given for comparison. The detection results of the GABOR method [137], the SOBS method [112], the LCC method [62], the proposed change detection method and the shadow detection method are shown for comparison. The frame-level quantitative performance, including the number of true positives for shadow points (TP_s), the number of false negatives for shadow points (FN_s), the number of true positives for object points (TP_o), the number of false negatives for object points (FN_o), and the accuracy, are also presented.

The performance of all the methods is very good for this frame since the frame contains only one problematic situation (e.g., some parts of the red vehicle are smooth and textureless). With an accuracy of 95.4%, the performance SOBS is higher than GABOR (accuracy of 85.1%) and LCC (accuracy of 88.4%) for this frame as the method strongly relies on color information (**HSV** color format) to distinguish the three classes: foreground objects, background, and shadows. By incorporating texture information with color, the proposed method achieves high accuracy (95.4% of the total object points are correctly detected as foreground objects). Some background surfaces (e.g., parts above the car in Figure 6.2 (h)) are detected as foreground by the proposed change detection method. However, these misclassified parts are correctly detected as shadows in a later stage of the proposed shadow detection method (see Figure 6.2-(i)).

Table 6.2 shows the quantitative performance of the methods in terms of recall, precision, shadow detection rate, shadow discrimination rate, accu-

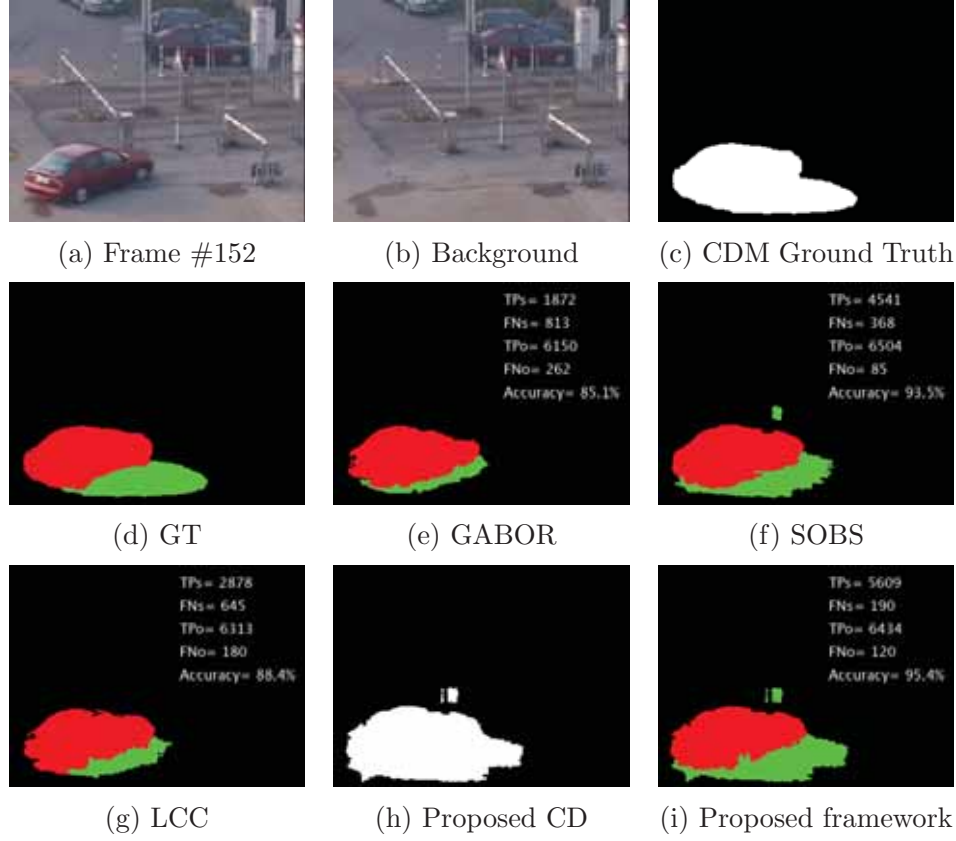


Figure 6.2: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *Campus*. Foreground objects and shadows are highlighted in red and green, respectively.

Table 6.2: Methods quantitative performance for the sequence *Campus*.

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	88.1	90.7	69.7	95.9	84.9	8
SOBS [112]	96.0	96.7	92.5	98.7	93.4	18
LCC [62]	91.7	93.4	81.7	97.2	88.7	14
Proposed	97.4	98.5	96.7	98.2	95.1	10

racy, and execution time for 230 frames from the sequence *Campus*. The results shown in Table 6.2 indicate that the proposed framework, with an overall average accuracy of 95.1%, outperforms state-of-the-art algorithms in the case of nontextured surfaces. In terms of execution time, the proposed framework produces 10 frames per second, which is slower than the SOBS and LCC methods. However, the proposed method can still be considered to be efficient for real-time applications. One way to increase the method’s capability is to reduce the frame size (the original frame size is 288×352) without affecting the final results.

Figure 6.3 shows the results of the methods for a frame taken from the second dataset, *Cubicle*. In this example, the capability of the methods is tested under the existence of foreground-background camouflage (strong camouflage between parts of the walking person and the background surfaces). The performance of GABOR is higher than that of SOBS and LCC since the method depends heavily on extracting microtextures via standard Gabor functions. With a total of 2427 object points correctly detected and an accuracy of 92.5%, the GABOR method is the closest to the proposed framework, which has an accuracy of 92.8%.

Table 6.3 presents the average quantitative performance of the methods for 308 frames from the sequence *Cubicle*. The proposed framework achieved high overall accuracy, correctly detecting 92.9% of the foreground objects. With an execution time of 9 frames per second, the run-time efficiency of the proposed method is slightly better than that of the GABOR method but lower than that of the other two methods.

The performance of the methods is further examined and analyzed on another challenging dataset affected by the problem of shadow color blending. Figure 6.4 shows the results of the methods for a frame taken from the *WBS* dataset. In this scene, the background surface has high reflectance that, in turn,

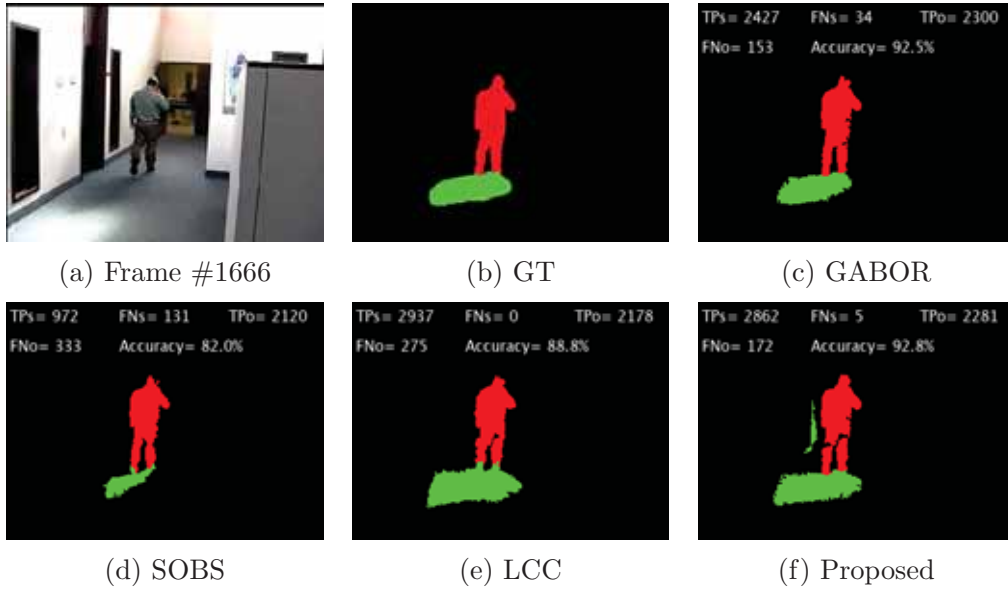


Figure 6.3: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *Cubicle*.

Table 6.3: Methods quantitative performance for the sequence *Cubicle*.

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	96.2	99.2	98.3	93.2	92.5	7
SOBS [112]	86.8	95.9	87.7	86.5	82.4	17
LCC [62]	94.9	100	100	88.7	88.6	12
Proposed	96.7	99.9	99.8	93.1	92.9	9

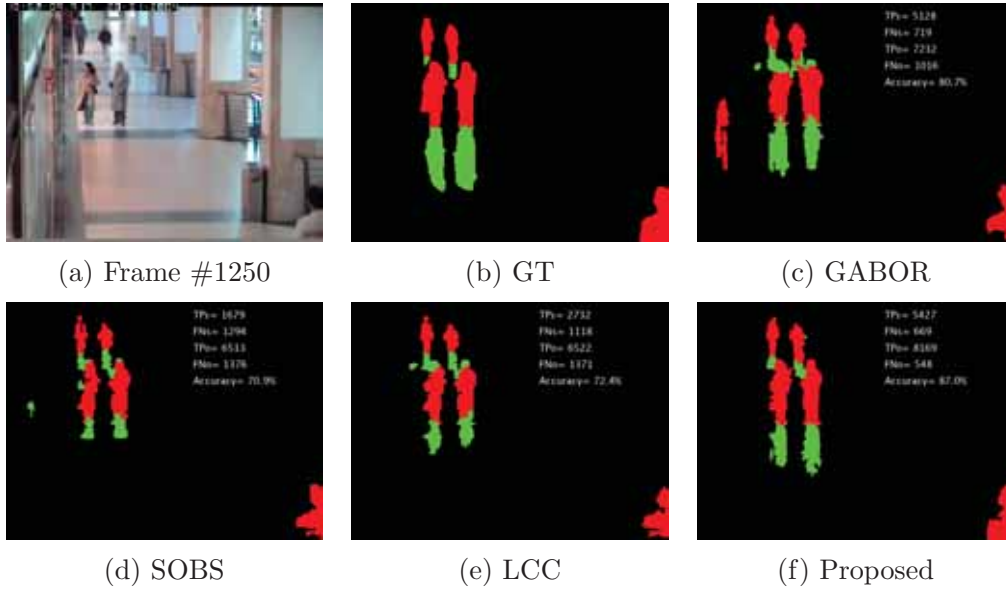


Figure 6.4: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *WBS*.

Table 6.4: Methods quantitative performance for the sequence *WBS*.

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	87.7	94.5	87.7	87.3	80.8	7
SOBS [112]	75.6	86.4	56.7	82.5	71.1	16
LCC [62]	78.8	89.2	70.9	82.7	72.6	12
Proposed	91.8	95.3	89.1	93.7	89.3	9

causes most parts of the foreground object to be reflected back. As a result, the colors of these reflected object parts and parts of the cast shadows are mixed, causing color blending, mainly in the umbra region of the shadow.

For this typical example, SOBS and LCC achieve similar accuracies of 70.9% and 72.4%, respectively. Compared to SOBS and LCC, the accuracy of GABOR is high (80.7% of the object points are correctly detected). With an accuracy of 87.0%, the performance of the proposed framework is much better than that of the other three methods since the proposed method takes advantage of the gradient distribution among neighboring points.

Table 6.4 shows the overall quantitative performance of the methods on 568 frames taken from the sequence *WBS*. The proposed framework achieved higher overall accuracy (89.3% of the foreground objects are detected correctly). With an execution time of 9 frames per second, the run-time efficiency of the proposed method is slightly higher than that of the GABOR method but lower than that of the other two methods.

As discussed earlier, shadow color blending is a problem associated with regions of shadow that are easily misclassified as moving objects. Therefore, the most important evaluation metric in this case is the overall shadow detection rate (η). As shown in Table 6.4, the overall shadow detection rate of SOBS is the lowest ($\eta = 56.7\%$) because SOBS performs pixel-level analysis and strongly depends on the color feature, causing misclassification of most shadow points due to color blending. The shadow detection rate for LCC ($\eta = 70.9\%$) is better than that of SOBS as the method performs region-level analysis and conducts classification on large regions. With $\eta = 87.7\%$ and $\eta = 89.1\%$, respectively, the GABOR and proposed methods effectively address the problem. Taking the execution time into account, the proposed method is a better choice than GABOR.

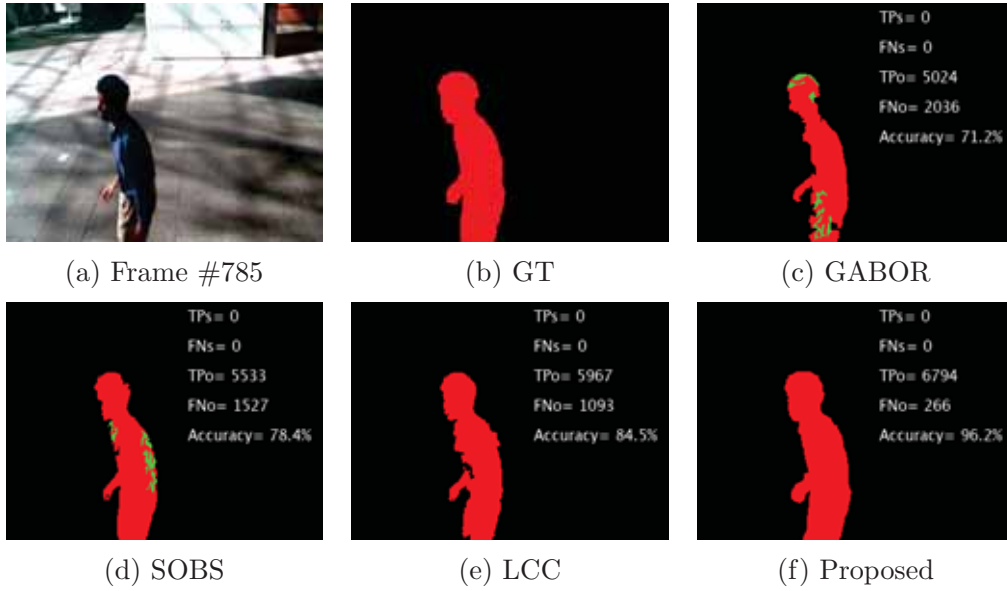


Figure 6.5: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *PPL*.

Table 6.5: Methods quantitative performance for the sequence *PPL*

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	71.2	78.2	73.8	71.5	71.6	7
SOBS [112]	78.3	74.0	76.2	78.4	78.5	16
LCC [62]	84.5	82.3	85.6	84.6	85.1	12
Proposed	87.6	86.9	88.7	89.2	88.4	9

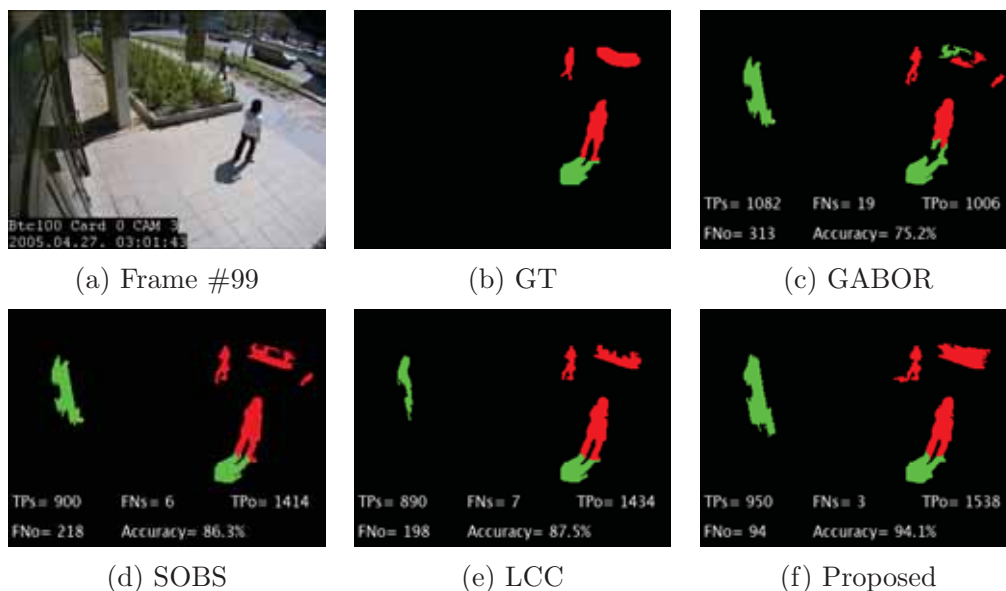


Figure 6.6: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *Seam*.

Table 6.6: Methods quantitative performance for the sequence *Seam*

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	86.6	99.0	98.1	76.3	75.2	8
SOBS [112]	91.2	99.8	99.2	86.9	87.3	18
LCC [62]	91.6	99.3	99.2	87.9	87.7	13
Proposed	95.3	97.8	98.7	90.4	91.2	10

Figures 6.5 and 6.6 show the detection results of the methods for a frame taken from *PPL* and *Seam*, respectively. For these two sequences, the efficiency of the methods is examined under the existence of three problematic situations: foreground-background camouflage, nontextured surfaces, and dark surfaces. These problems influence the foreground object region; therefore, some of these regions may be incorrectly classified as shadows. Meanwhile, the effects of these problem on shadow regions are almost negligible; thus, the shadow detection rate is ignored. For this particular example, no shadow points of the walking person are captured by the camera; therefore, the true positives and false negatives for shadow points are zero.

According to the results shown in Figures 6.5 and 6.6, the LCC method achieves better accuracy than the GABOR method and the SOBS method. Most foreground points ($TP_o = 5967$ of 7060 for *PPL* and $TP_o = 1434$ of 1632 for *Seam*) are correctly detected by LCC. Among the methods, GABOR performs worst, with accuracies of 71.2% for *PPL* and 75.2% for *Seam*, as the method is unable to extract the required texture information from the foreground object regions affected by dark and nontextured surfaces. For example, the dark parts of the walking persons in Figures 6.5 (c) and 6.6 (c) are misclassified as either parts of the background or shadows.

Compared to that of the state-of-the-art methods, the accuracy of the proposed framework is high for the two frames. As shown in Figures 6.5 (f) and 6.6 (f), a total of 96.2% of the object points in the frame taken from the *PPL* sequence and 94.1% of the object points in the frame taken from the *Seam* sequence are correctly classified as foreground objects.

Tables 6.5 and 6.6 provide further evidence of the accuracy of the proposed framework under the three problematic situations mentioned above. A total of 186 frames from the sequence *PPL* and 87 frames from the sequence *Seam* are

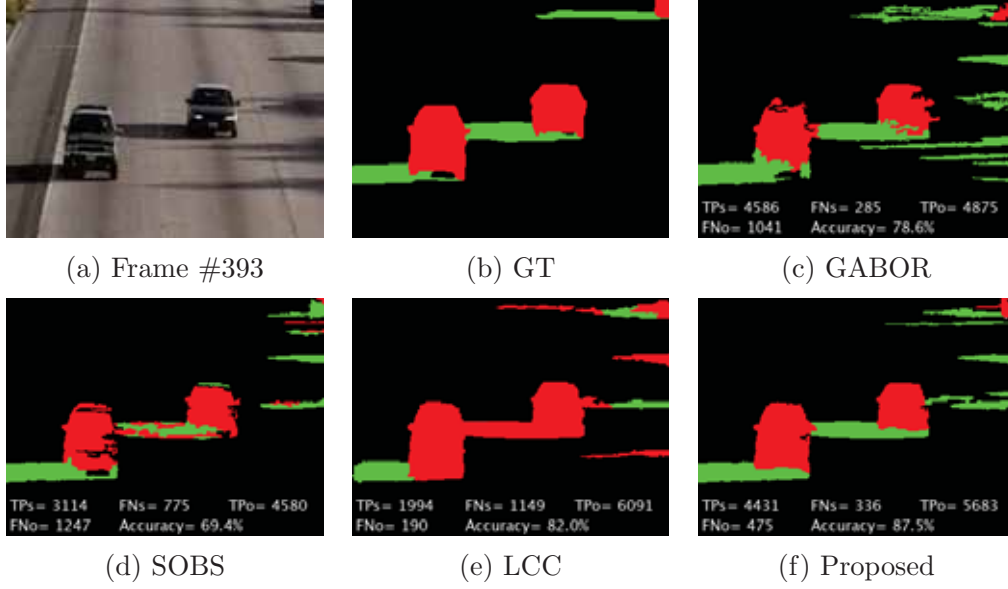


Figure 6.7: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *HwyI*.

Table 6.7: Methods quantitative performance for the sequence *HwyI*

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	87.7	97.0	94.1	82.3	78.6	8
SOBS [112]	79.1	90.8	80.0	78.6	69.3	18
LCC [62]	85.8	87.5	63.5	96.9	87.1	13
Proposed	92.6	96.9	92.8	92.8	87.8	10

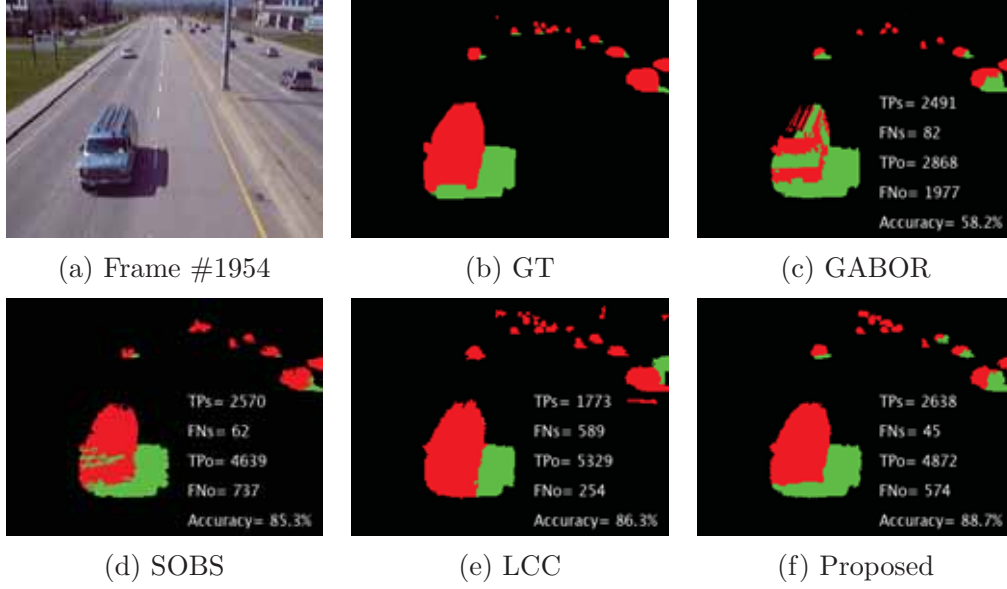


Figure 6.8: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *HwyIII*.

Table 6.8: Methods quantitative performance for the sequence *HwyIII*

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	72.2	98.5	96.8	59.2	58.3	8
SOBS [112]	90.0	99.1	97.3	86.2	84.7	18
LCC [62]	89.4	92.5	81.3	95.4	87.8	13
Proposed	92.4	99.3	98.7	89.4	88.9	10

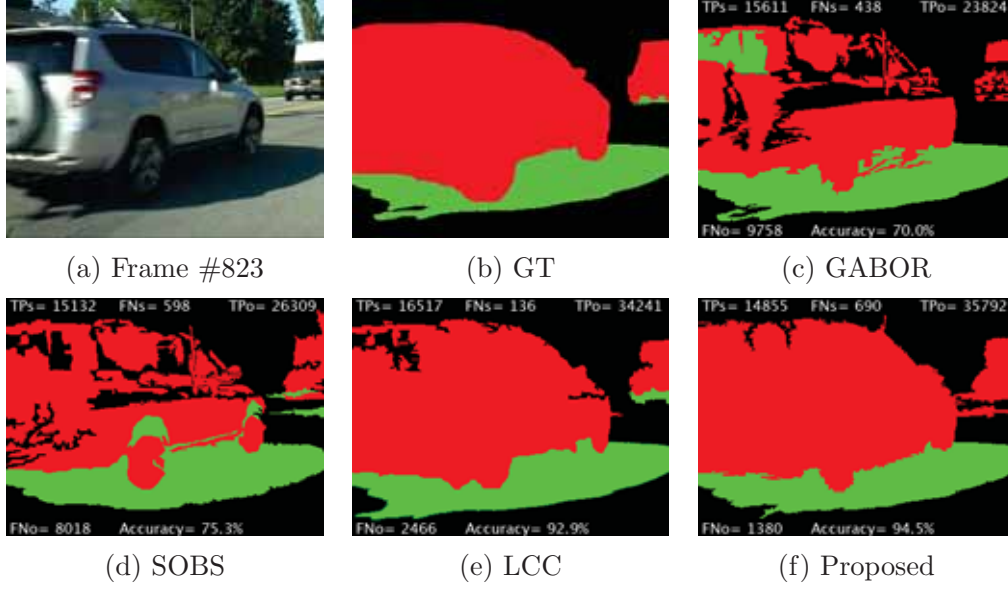


Figure 6.9: Qualitative shadow detection results for GABOR [137], SOBS [112], LCC [62], and the complete proposed shadow detection framework, for a frame taken from the sequence *Bungalows*.

Table 6.9: Methods quantitative performance for the sequence *Bungalows*

Methods	Change Detection Evaluation		Shadow Detection Evaluation			Execution Time (Frames/sec)
	Recall %	Precision %	Shadow detection rate (η %)	Shadow discrimination rate (ξ %)	Accuracy %	
GABOR [137]	79.1	98.9	97.2	70.8	70.1	7
SOBS [112]	82.7	98.6	96.2	76.7	75.3	17
LCC [62]	91.2	96.7	97.1	90.4	88.7	11
Proposed	93.0	97.9	93.7	92.3	90.7	9

used for these quantitative comparisons. Notably, the execution times (frames per second) of the proposed framework are lower than those of the SOBS and LCC methods; however, the overall performance is higher than that of all the state-of-the-art methods.

The performance of the methods is further analyzed for the last three sequences of Table 6.1. Figures 6.7, 6.8, and 6.9 show the detection results of the methods on three frames taken from the traffic sequences *HwyI*, *HwyIII*, and *Bungalows*, respectively. The performance of the methods is assessed under a combination of three problematic situations: chromatic shadows, foreground-background camouflage, and nontextured object surfaces.

For these frames, the accuracy achieved by the LCC method and the proposed method are much higher than those of the other methods. The LCC method correctly detects a total of 82.0% of the foreground objects for *HwyI*, 86.3% of the foreground objects for *HwyIII*, and 92.9% of the foreground objects for *Bungalows*. Therefore, LCC and the proposed framework are the preferred methods to detect moving vehicles in traffic scenes. The reason for the high performance of the LCC method is the ability to detect and eliminate shadows in cases of chromatic shadows.

Despite the existence of these problems in the sequences, the proposed method successfully distinguishes the three classes: foreground objects, shadows, and background. As shown in Figures 6.7 (f), 6.8 (f), and 6.9 (f), most object patches are correctly classified as objects, and almost all shadow patches are successfully classified as shadows. Some misclassification occurs due to strong camouflage between parts of the moving objects and their corresponding backgrounds.

Tables 6.7, 6.8, and 6.9 summarize the overall quantitative performances

of the methods. A total of 245 frames from the sequence *HwyI*, 180 frames from the sequence *HwyIII*, and 468 frames from the sequence *Bungalows* are considered in this comparison. With overall average accuracies of 87.8% for the sequence *HwyI*, 88.9% for the sequence *HwyIII*, and 94.7% for the sequence *Bungalows*, the proposed framework outperforms the state-of-the-art algorithms in the presence of the three main problematic situations mentioned earlier. In addition to the proposed framework, the LCC method achieves high accuracy, as expected.

6.5.3 Quantitative Performance

Table 6.10 shows the pixel-based average performance of the methods in terms of the accuracy with respect to the problematic situations. The execution time, in frames per second, for the methods is also given. A total of 2272 frames from all sequences are used for these quantitative comparisons.

Table 6.10: Overall accuracy performance and execution time for the comprehensive methods.

Methods	GABOR [137]	SOBS [112]	LCC [62]	Proposed framework
Chromatic shadow (%)	69.6	76.9	89.1	88.9
Colour blending (%)	80.7	71.1	72.8	89.3
Camouflage (%)	79.6	77.6	80.4	91.2
Non-textured surfaces (%)	74.2	80.4	79.1	92.5
Dark surfaces (%)	70.7	78.0	73.6	86.4
All problematic situations (%)	74.9	76.8	79.0	89.6
Execution time (frames/sec)	8	18	14	10

The accuracy per problematic situation is calculated using the quantitative results obtained from the associated datasets (the datasets identified as having problematic situations in Table 6.1). For example, to calculate the accu-

racy for the SOBS method under the problem of chromatic shadows, the average accuracy obtained for the sequences *PPL*, *HwyI*, *HwyIII*, and *Bungalows* in Tables 6.5, 6.7, 6.8, and 6.9 is calculated.

As shown in Table 6.10, the accuracies of the LCC method and the proposed framework are much higher than those of the other methods in scenarios with chromatic shadows. For outdoor environments, LCC and the proposed framework are the preferred methods for detecting shadows since the shadows in such cases are usually chromatic. This result is confirmed by the accuracies achieved by the two methods in Table 6.10.

The performance of the methods with reference to color blending is evaluated using the sequence *WBS* only. The accuracies of the SOBS and LCC methods are the lowest (less than 73%). With an accuracy greater than 89%, the proposed framework shows its capability in addressing this problem.

With respect to the problem of foreground-background camouflage, the performance of the GABOR, SOBS, and LCC methods is similar, with the LCC method having a slight advantage (accuracy of approximately 80.4%). Meanwhile, the proposed framework achieves a high accuracy of approximately 91.2%. The average accuracy for cases with camouflage is obtained using all the sequences except for *Campus*.

For nontextured surfaces and dark surfaces, the accuracy of the GABOR method is the lowest among the methods, as expected. The LCC method has slightly higher accuracy than GABOR: a total of 79.1% and 73.6% points are correctly classified under these environmental problems. Meanwhile, the overall performance of the SOBS method is adequate in the presence of these problematic situations (average accuracies of 80.4% and 78.0% are reported for nontextured surfaces and dark surfaces, respectively). Relative to that of SOBS, the

accuracy of the proposed framework is at least 6% higher for both problems, providing further evidence of the efficiency of the proposed framework in addressing these problems.

The quantitative results shown in Table 6.10 provide further evidence of the ability of the proposed framework in the problematic situations. The proposed framework achieves the highest average accuracy (approximately 89.6%), followed by the LCC method (79%), SOBS method (76.8%), and finally the GABOR method (just below 75%).

In terms of average execution time, the proposed framework can produce 10 frames per second, which is slightly faster than the GABOR method (8 frames per second) and slower than the LCC (14 frames per second) and SOBS (18 frames per second) methods. The execution time, the only limitation of the proposed framework, can be improved by decreasing the frame size.

6.6 Evaluation of Features

In this section, the qualitative and quantitative segment-based classification accuracy is analyzed using a number of popular state-of-the-art feature descriptors and the feature descriptors proposed for shadow detection in this thesis.

All sequences provided in Table 6.1 are used to evaluate the performance of each feature descriptor. The features are extracted using ground truth CDMs provided by the referenced link associated with each dataset.

To prepare the segments for feature extraction, the **CDM** binary image is first applied to the current frame and the background image. Superpixel image segmentation [118] is then performed to divide the masked current frame into

nonuniform segments. Finally, the features are extracted as described by the referenced papers.

For a fair comparison, the same datasets and reference dictionary sizes were used for all the feature descriptors. Furthermore, the feature classification for all the feature descriptors is conducted on two levels, segment-level classification and region-level classification, as described in Chapter 5. The proposed multi-independent sparse representation classifier is used for the segment-level classification. For region-level classification, the inter-region dependencies among all the neighboring segments are considered, as described in Section 5.6.

The technique behind each of the popular feature descriptors used for comparison in this experiment is briefly discussed in Section 6.6.1. In Section 6.6.2, the segment-level qualitative performance for all the features is evaluated. Finally, the segment-level quantitative performance with respect to the identified problematic situations is discussed in Section 6.6.3.

6.6.1 State-of-the-art Features

A number of popular feature descriptors are considered for comparison in this section. Most of these feature descriptors are suitable for shadow detection due to their capability to distinguish objects and shadows under varying local and global illumination conditions. The feature descriptors include **LDP** [123], **CLBP** [122], **LTP** [102], **SILTP** [103], and **LGHP** [124]. For the experimental settings, the number of neighboring pixels was set to $P=8$ for all feature descriptors. For **CLBP**, Equation (5.6) in Section 5.4.1 was revised so that each channel from the current frame is divided by the corresponding channel in the background frame. The intensity values for each channel were then normalized and converted to **YIQ** color values. The parameters for **CLBP** were set accord-

ing to [122]. For **LTP** and **SILTP**, the **RGB** color images were first converted to grayscale images; then, the current frame was divided by the background. The neighboring thresholds for the **LTP** and **SILTP** operators are based on images with intensities of $[0, 255]$; therefore, the resultant image was normalized to have intensities in the range of $[0, 255]$. Finally, the thresholds were set to **CLBP**=0, **LTP**=5, **SILTP**=0.1, **LDP**=0, and **LGHP**=0 (for all directions), as reported in their reference papers. The details of each method are given below.

- Local derivative patterns **LDP**:

LDP [123] was proposed to encode directional pattern features using higher-order local derivative variations. Compared to those of the first-order local pattern used in **LBP**, **LDP** is able to extract detailed discriminative features that can be useful for identifying patterns under varying illumination. The **LDP** operator labels the center pixel in a 5×5 patch by comparing two derivative directions at two neighboring pixels and concatenating the results into a 32-bit binary sequence. This feature descriptor is examined since it can extract texture information from smooth surfaces.

- Opponent-color local binary pattern **CLBP**:

CLBP [122] is a combination of color and texture introduced as a good local feature descriptor compared to **LBP**. A set of six feature descriptors is extracted from individual channels and from each pair of color channels using **RGB** color format. To extract features for individual channels, the center pixel in a 3×3 patch is compared to the neighboring pixels in a similar manner to **LBP**. The center pixel taken from one channel is compared to the neighboring pixels from the other channel to extract color and texture information. Thus, the dimension of the **CLBP** feature descriptor is six times that of the monochrome **LBP** histogram. This

feature descriptor is used for comparison since it is based on a combination of color and texture.

- Local ternary patterns **LTP**:

LTP [102] was originally proposed to extract texture information for face detection and recognition. The **LTP** descriptor is extended from **LBP** by simply adding a small offset value for comparison. Given a 3×3 patch, the 8 neighbor pixels are quantized to zero, +1, or -1, according to their gray-level differences compared to the center pixel. Each ternary pattern is then split into its positive and negative **LBP** codes using a coding scheme.

- Scaled invariant local ternary patterns **SILTP**:

LTP provides a robust texture descriptor by introducing a threshold for local intensity comparisons. However, **LTP** is not invariant under a linear intensity transformation. To overcome this problem, **SILTP** [103] was proposed to handle illumination variation, soft shadows in particular. **SILTP** can be extracted in a similar manner to that of **LTP**. The simple difference is that **SILTP** introduces a scale factor to validate the comparing range, which results in the **SILTP** operator being encoded with two bits rather than one.

- Local gradient hexa pattern **LGHP**:

The **LGHP** introduced in [124] is an extension of **LDP**. However, **LGHP** captures more texture details than **LDP** by taking advantage of the relationship among the reference pixel and its neighboring pixels at different distances in different derivative directions.

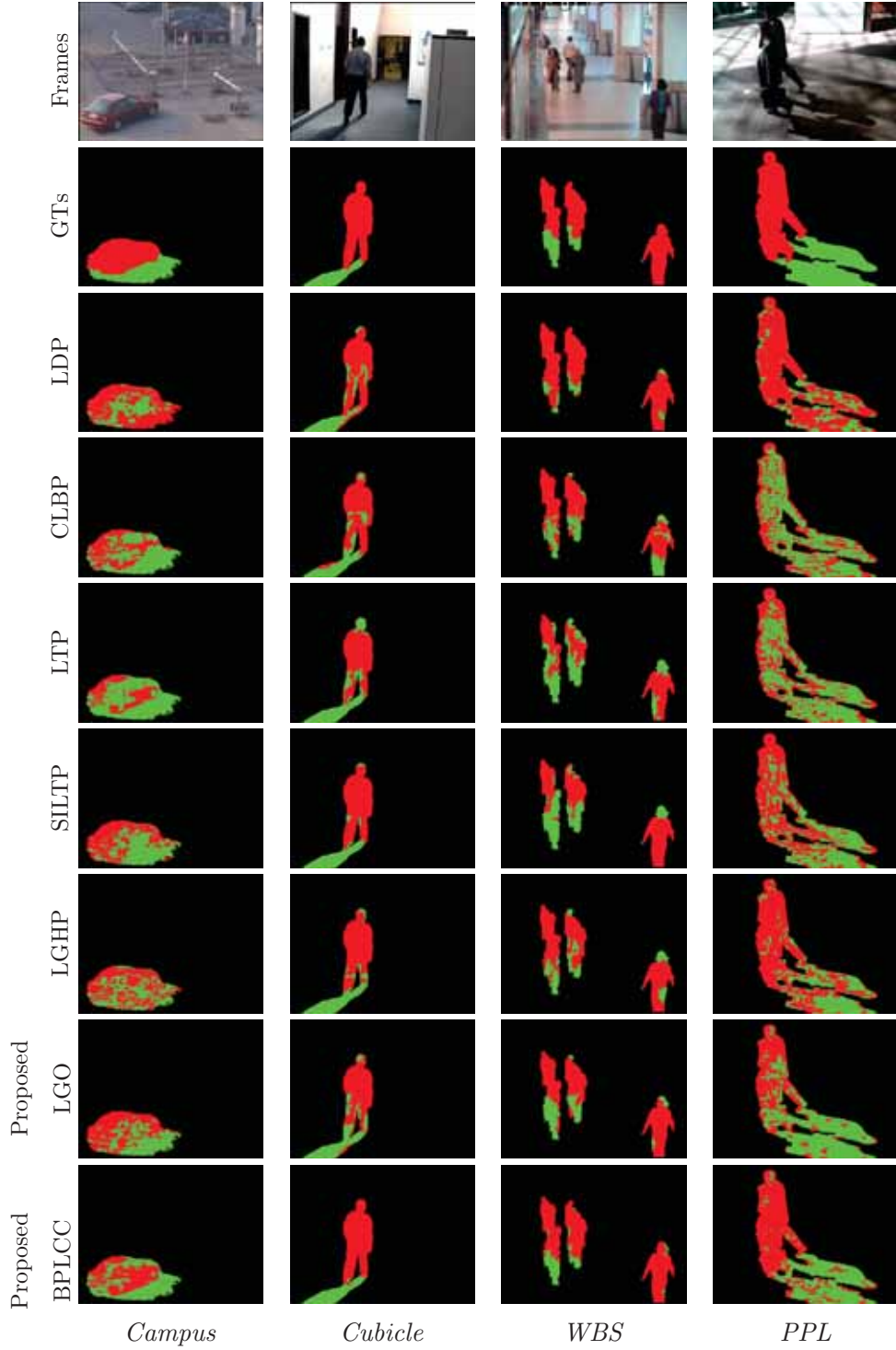


Figure 6.10: Qualitative shadow detection results using different features for a frame taken from *Campus*, *Cubicle*, *WBS*, and *PPL*. Foreground objects are marked in red and shadows are marked in green. Results are best viewed in color.

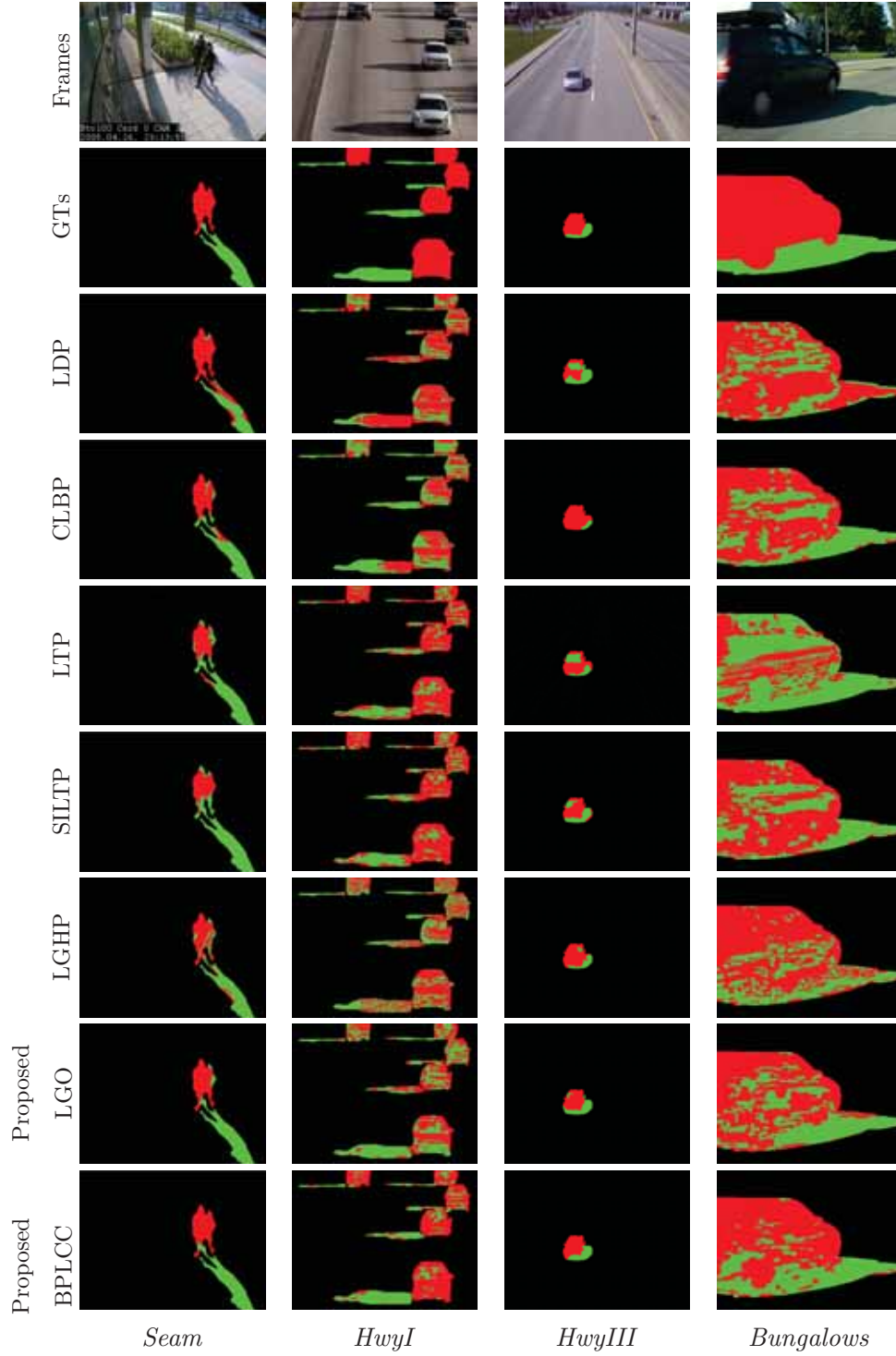


Figure 6.11: Qualitative shadow detection results using different features for a frame taken from *Seam*, *HwyI*, *HwyIII*, and *Bungalows*.

6.6.2 Qualitative Performance

Figures 6.10 and 6.11 show the results of a frame from each of the sequences provided in Table 6.1. These results are obtained by using **LDP** [123], **CLBP** [122], **LTP** [102], **SILTP** [103], **LGHP** [124], **LGO**, and **BPLCC**. Ground truth images are also given to improve the comparison.

The LDP feature descriptor [123] provides satisfactory results in distinguishing object parts from their background in different scenarios. As shown in Figures 6.10 and 6.11 (third row), most of the object parts have been correctly classified as objects. However, **LDP** extracts excessive textural detail, which adversely affects the method when classifying segments under varying illumination. For example, many shadow segments are incorrectly detected as parts of moving objects in Figure 6.10 (third row).

CLBP [122] detects more shadow segments than does **LDP** because the method utilizes a spatial relationship among neighboring pixels within a small patch. Except for those that are located under a strong chromatic shadow, for example, the results for *PPL* in Figure 6.10 (fourth row) and *HwyI* and *Bungalows* in Figure 6.11 (fourth row), most shadow segments are correctly classified as shadow by **CLBP**. The method, however, fails to detect object parts in scenes with strong foreground-background camouflage (for example, the results for *Campus* and *PPL* in Figure 6.10 (fourth row)) or nontextured and dark surfaces (for example, the results for *PPL* and *Bungalows* in Figures 6.10 and 6.11 (fourth row)).

At the same time, **LTP** [102] and its invariant version **SILTP** [103] produce similar results because both rely on only intensity differences among local neighboring pixels based on grayscale images. However, **SILTP** correctly detected more shadow segments as shadows and fewer object segments as objects

than did **LTP**. This difference is clearly evident by comparing the results obtained from the two feature descriptors for *PPL* and *Bungalows* in Figures 6.10 and 6.11 (fifth row) and (sixth row). The reason for this misclassification is that the similarity measure used by **SILTP** has a wider threshold than **LTP**, which causes large intraclass variation that affects the results.

The **LGHP** feature descriptor [124] correctly detects more object segments than do the above feature descriptors since the method captures texture information across different derivative directions at different angular widths. This fact is clearly noted in the results obtained for *PPL* and *Bungalows*. However, **LGHP** extracts unnecessary information, which leads to failure in recognizing background segments under extreme illumination changes. Consequently, many shadow segments are misclassified as objects, as shown in Figures 6.10 and 6.11 (seventh row).

The results from the proposed **LGO** are considerably better than those of all the above feature descriptors since the dominant light direction is estimated and incorporated into the gradient information. Most segments are correctly labeled in Figures 6.10 and 6.11 (eighth row) while maintaining a good balance between the two main components: shadow and object.

Compared to all the above features, the proposed **BPLCC** provides good results for the problematic situations discussed in this thesis. **BPLCC** recognizes most of the shadow segments under chromatic shadows, as shown in the results obtained for *HwyI*, *PPL*, and *Bungalows* in Figures 6.10 and 6.11 (ninth row). In addition, most of the object segments were successfully detected under strong foreground-background camouflage and nontextured and dark surfaces (all results in Figures 6.10 and 6.11 (ninth row)).

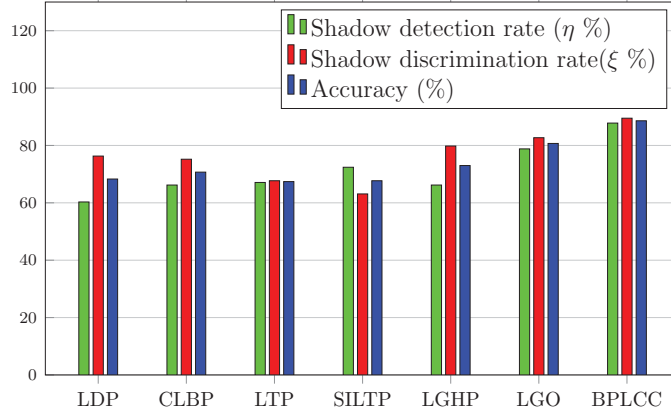


Figure 6.12: The performance of popular features and the proposed features versus shadow detection rate, shadow discrimination rate, and accuracy.

6.6.3 Quantitative Performance

Figure 6.12 shows the overall segment-level quantitative performance in terms of the shadow detection rate, shadow discrimination rate, and accuracy for each of the above feature descriptors. A total of 2574 frames from the last group of datasets are used in this comparison.

With an average shadow detection rate of 87.5%, average shadow discrimination rate of 89.5%, and average accuracy of over 88.6%, the proposed color constancy feature **BPLCC** significantly outperforms all the other feature descriptors, which suggests that shadows and objects are better characterized by incorporating color and texture information from neighboring pixels than by using grayscale local texture and standard gradient descriptors.

6.7 Evaluation of Shadow Detection Methods

In this section, the quantitative and qualitative classification accuracy of the proposed shadow detection method, in terms of the shadow detection rate, shadow discrimination rate, and detection accuracy, is analyzed and compared with that of five well-known moving shadow detection methods: a region-based affinity propagation method (**AP** [138]), a joint-histogram-based method (**JH** [139]), a region-based local color constancy method with adaptive thresholds (**LCC** [62] - this method is used again for comparison due to its accuracy), a feature-based method with statistical learning (**SL** [134]), and another feature-based method with **SVM** [131]). These techniques are discussed in detail in Chapter 2. These methods were selected for comparison since they use different feature descriptors and classifiers.

To obtain the best results, the relative size threshold (λ) used in **LCC** [62] is set to 0.01 for sequences with small shadows (*Cubicle*, *WBS*, and *HwyIII*) and to 0.04 for the remaining sequences (see Table 6.1 for more details on shadow sizes). For all other methods, the major parameters are set according to their reference papers.

The methods are evaluated at the pixel level with respect to each sequence provided in Table 6.1 and each problematic situation discussed in Chapter 3.

For fair comparison, the ground truth **CDMs** are used for all methods, including the proposed shadow detection method. In this way, the task for the methods is to distinguish the two classes: shadows and objects. In the following, the qualitative and quantitative performance and the stability of the methods is discussed.

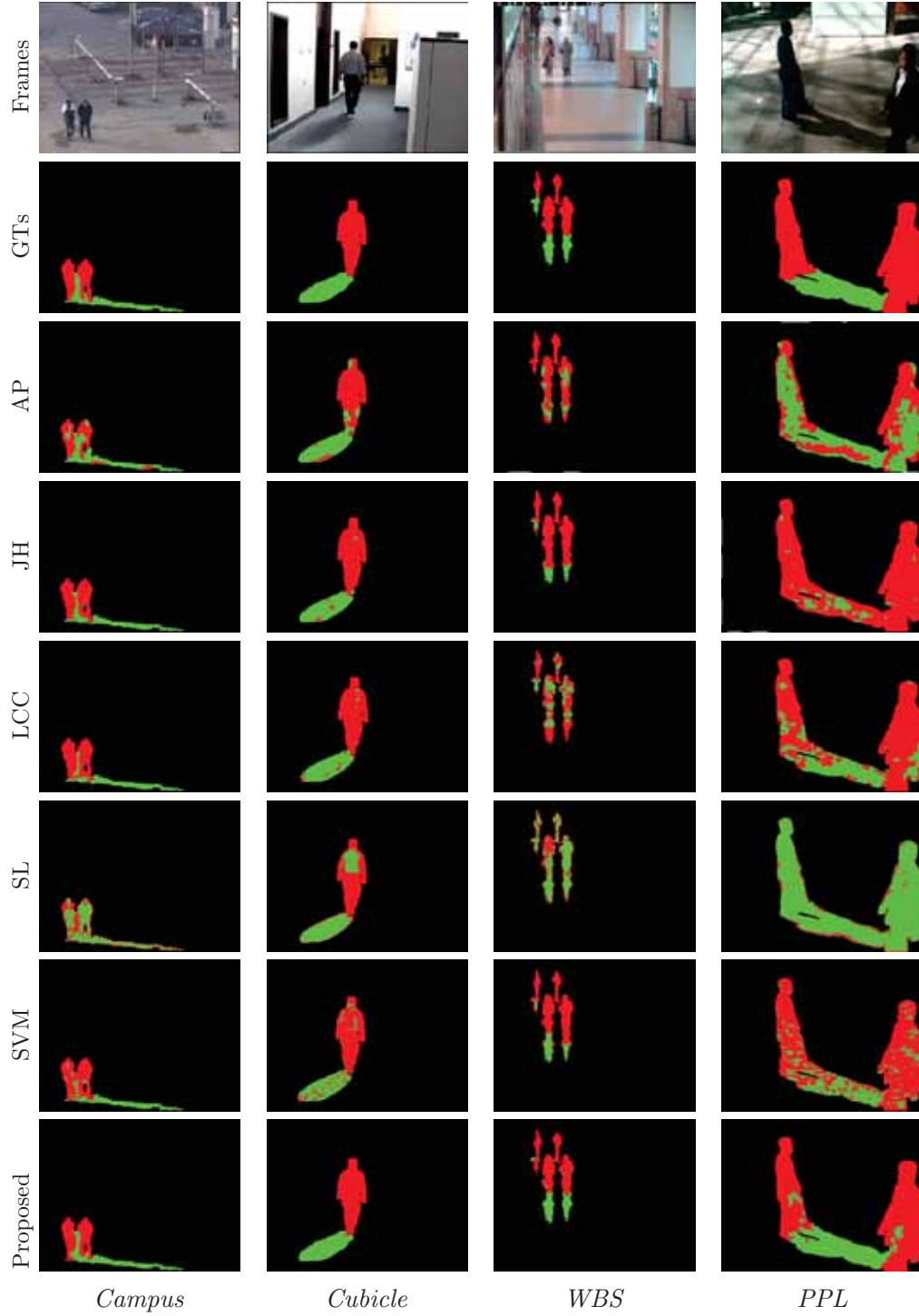


Figure 6.13: Qualitative moving shadow detection results of all compared methods for a frame taken from the first four datasets in Table 6.1.

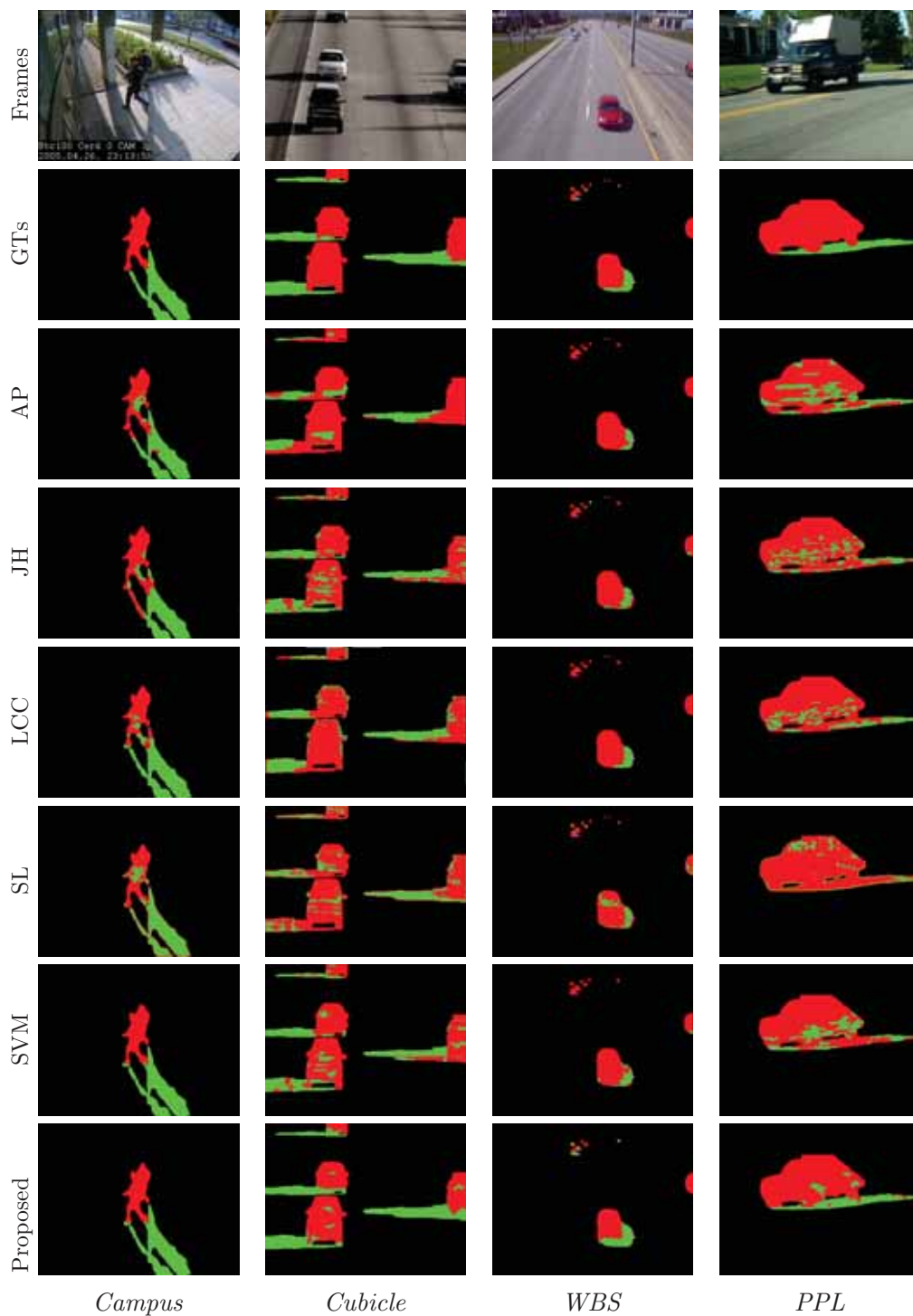


Figure 6.14: Qualitative moving shadow detection results of all compared methods for a frame taken from the last four datasets in Table 6.1.

6.7.1 Qualitative Performance

Figures 6.13 and 6.14 show the shadow detection results for the proposed shadow detection method and the other five methods: a region-based affinity propagation method (**AP** [138]), a joint-histogram-based method (**JH** [139]), a region-based local color constancy method with adaptive thresholds (**LCC** [62]), a feature-based method with statistical learning (**SL** [134]), and another feature-based method with **SVM** ([131]).

A challenging frame from each dataset is shown in the first row of Figures 6.13 and 6.14. The ground truth images are given in the second row. The shadow detection results of the **AP** [138], **JH** [139], **LCC** [62], **SL** [134], **SVM** [131] and proposed shadow detection methods are shown in the third, fourth, fifth, sixth, seventh, and eighth rows, respectively. The results are evaluated based on the problematic situations discussed in Chapter 3.

The performance of the first two methods, the **AP** [138] and **JH** [139] methods, is satisfactory only in the case of camouflage between the foreground and background surfaces (see the results of the two methods for *Cubicle* in Figure 6.13). However, the two methods fail to detect shadow regions affected by chromatic shadows (many false detections occur in the shadow regions in the results of the two methods for *PPL*, *HwyI*, and *Bungalows* in Figures 6.13 and 6.14) and shadow regions affected by color blending (shadow regions misclassified as objects in the results obtained by **AP** for *WBS* in Figure 6.13), and they fail to detect object regions with dark and low-texture properties (many object points were misclassified as shadows in the object regions in Figures 6.13 and 6.14). The main reason for the misclassification by the two methods is that they strongly rely on either texture information (**AP**) or gradient information (**JH**) for classification. In many cases, texture or gradient information is not available,

as discussed in Chapter 3.

Similar to the above two methods, the detection results of the **LCC** method [62] are acceptable for indoor environments, where almost all the shadow points are correctly classified as shadows (see the result obtained for *Cubicle* in Figure 6.13). In addition, **LCC** performs better than **AP** and **JH** in outdoor environments in the presence of chromatic shadows (see the results for all outdoor frames shown in Figures 6.13 and 6.14 (fifth column)) since **LCC** is designed for that purpose. However, the method cannot provide accurate results in other cases, such as shadow color blending (the results of **LCC** for *WBS* in Figure 6.13) and dark and low-textured areas. For example, pixels that belong to the persons in the frame of *PPL* in Figure 6.13 are misclassified as parts of shadows. The reason for this misclassification is that **LCC** strongly relies on gradient constancy measurements, which are obtained from subregions of the luminance ratio image and are used to distinguish shadows from objects. However, object regions with low-reflectance properties or/and smooth surfaces often produce low gradient constancy and are, in turn, misclassified as shadows by **LCC**.

The shadow detection results obtained from the **SL** method [134] and **SVM** method [131] are slightly better than those of all the above methods, except for the result obtained for *Bungalows*, where the size of the shadow is too small to be recognized. Most of the shadow regions in Figures 6.13 and 6.14 are correctly detected by **SL** and **SVM**. However, the **SL** method fails to preserve object points in cases of strong camouflage between the foreground and background regions. This finding is notable in the results obtained for the frames of *Cubicle* and *WBS* in Figure 6.13, in which many object regions that belong to walking persons are misclassified as shadows. In addition, nontextured and dark surfaces result in object misclassification for **SL**, as shown in the results obtained for *PPL* in Figure 6.13. Despite some object misclassifications observed

in the results of *PPL*, *HwyI*, and *Bungalows*, the **SVM** method provides better results than does **SL** when detecting object parts in all cases.

Compared to that of the above five methods, the average performance of the proposed method is outstanding. The proposed method provides accurate results in detecting the actual size of the foreground object and detecting the shadow points under identified problematic situations for shadows. However, a few false detections are observed. For example, some small shadow patches are misclassified as objects in *WBS* in Figure 6.13 due to the presence of extreme color blending in the background. In addition, some small object patches in *PPL*, *HwyI*, and *Bungalows* are misclassified as shadows due to the very low reflectance of these patches. Except for these misclassifications, the overall qualitative results for all the other scenarios are significant, with almost all the patches correctly classified and no false detections.

Figures 6.15 and 6.16 show further evidence of the improvement of the proposed shadow detection method. The two regions, foreground objects and their shadows, are successfully separated in various challenging situations.

6.7.2 Quantitative Performance

The quantitative performance of the shadow detection methods is compared using a total of 1809 frames selected from the third group of datasets. Because different classification levels are used in these methods, the classification accuracy is measured for individual pixels instead of image patches, as shown in Equations (6.3), (6.4), and (6.5).

Figure 6.17 shows the pixel-based classification accuracy in terms of the shadow detection rate and shadow discrimination rate on each dataset. The quantitative results from all the methods are similar for the first two sequences -

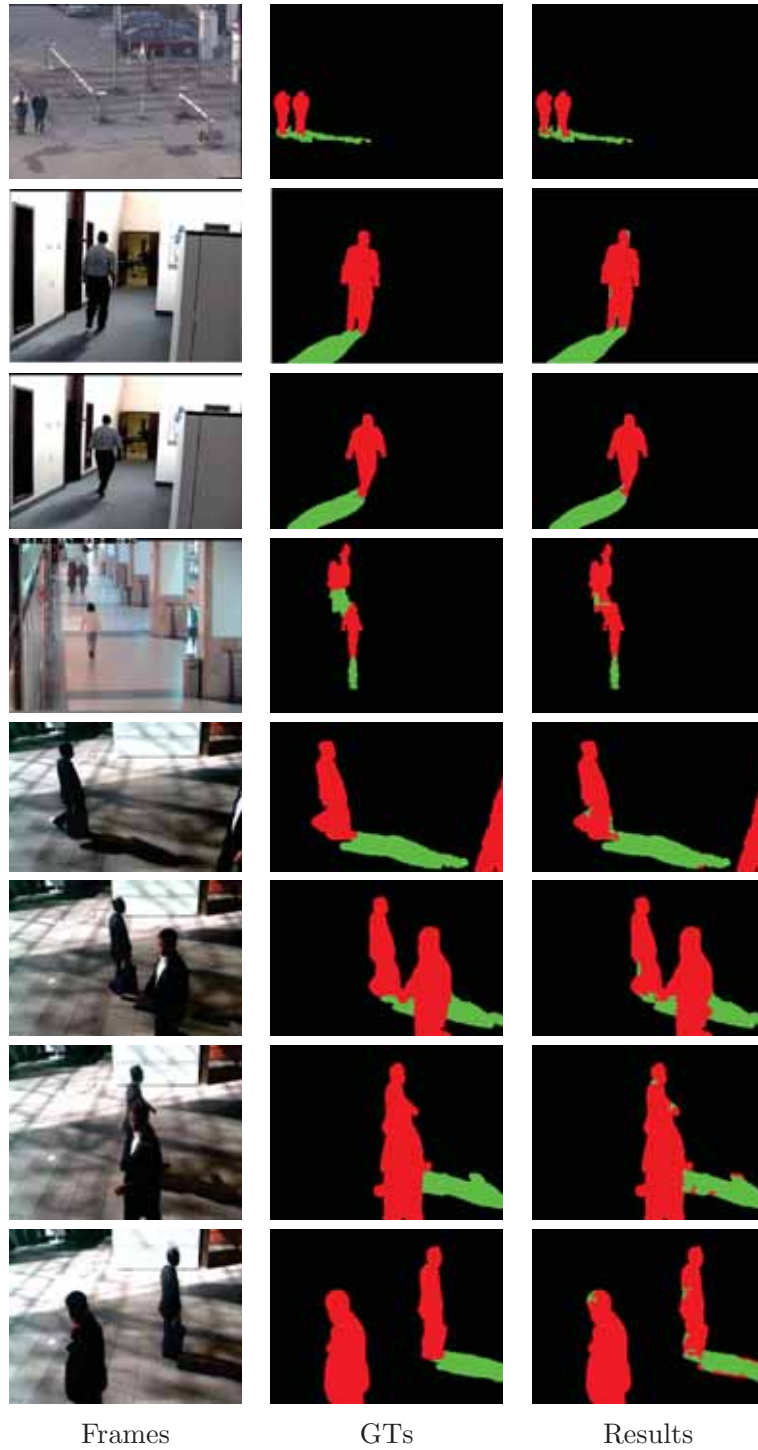


Figure 6.15: Qualitative moving shadow detection results of the proposed method.

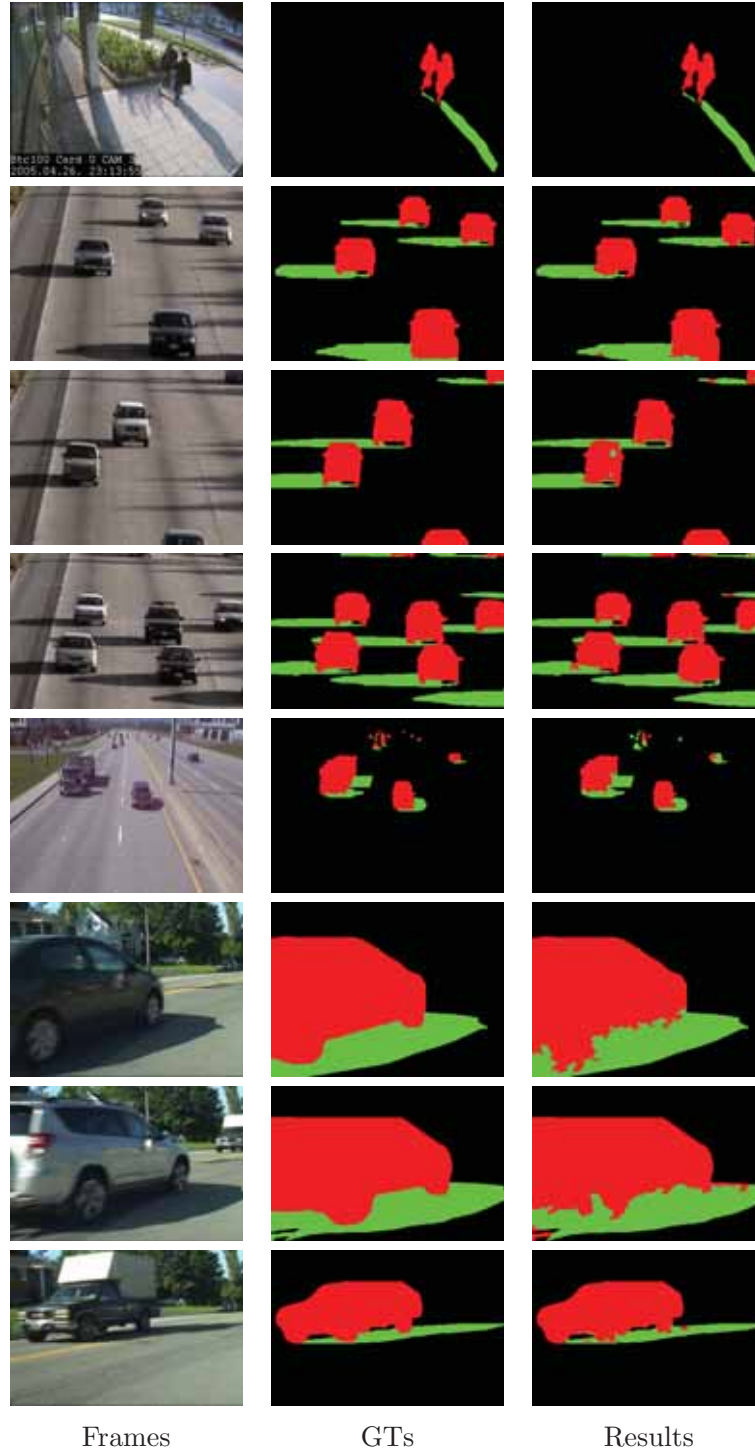


Figure 6.16: Qualitative moving shadow detection results of the proposed method.

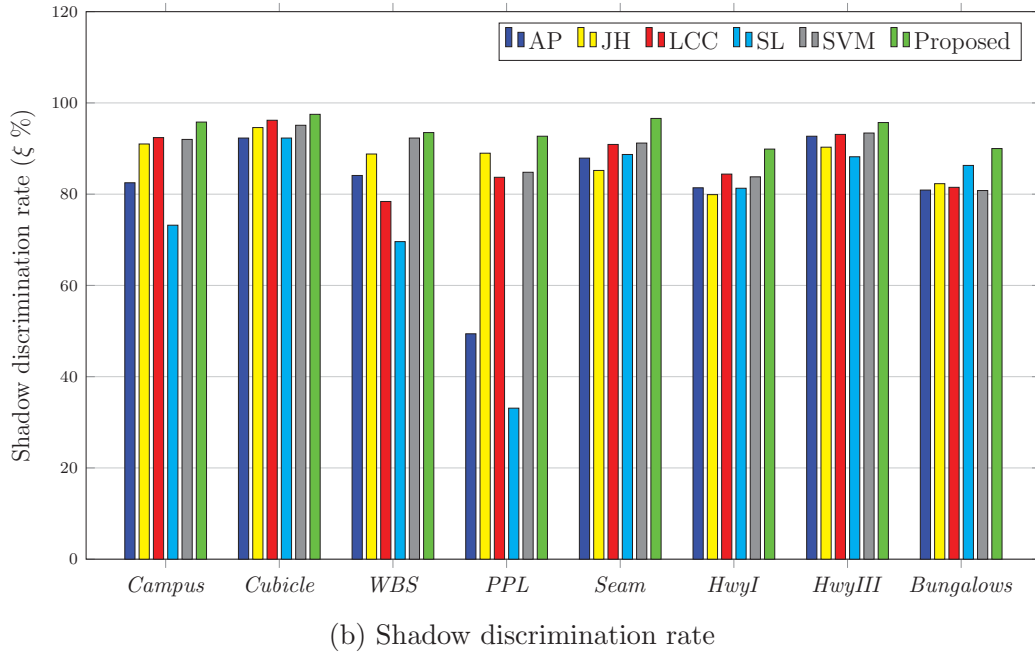
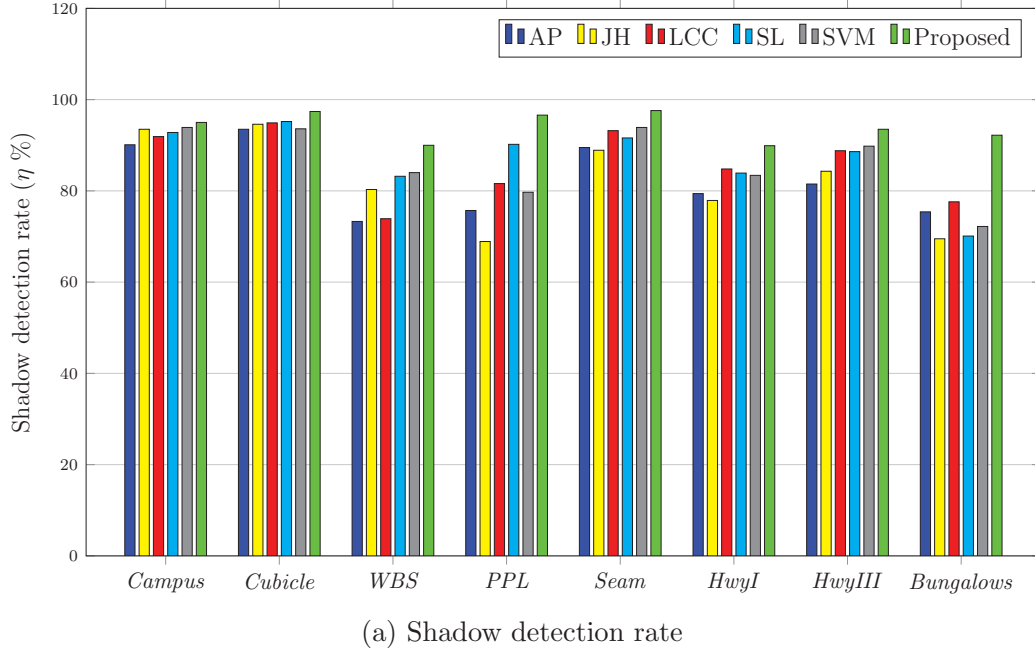


Figure 6.17: The pixel-based classification accuracy, in terms of (a) Shadow detection rate (η %) and (b) Shadow discrimination rate (ξ %), of AP [138], JH [139], LCC [62], SL [134], SVM [131], and the proposed moving shadow detection with respect to all sequences provided in Table 6.1.

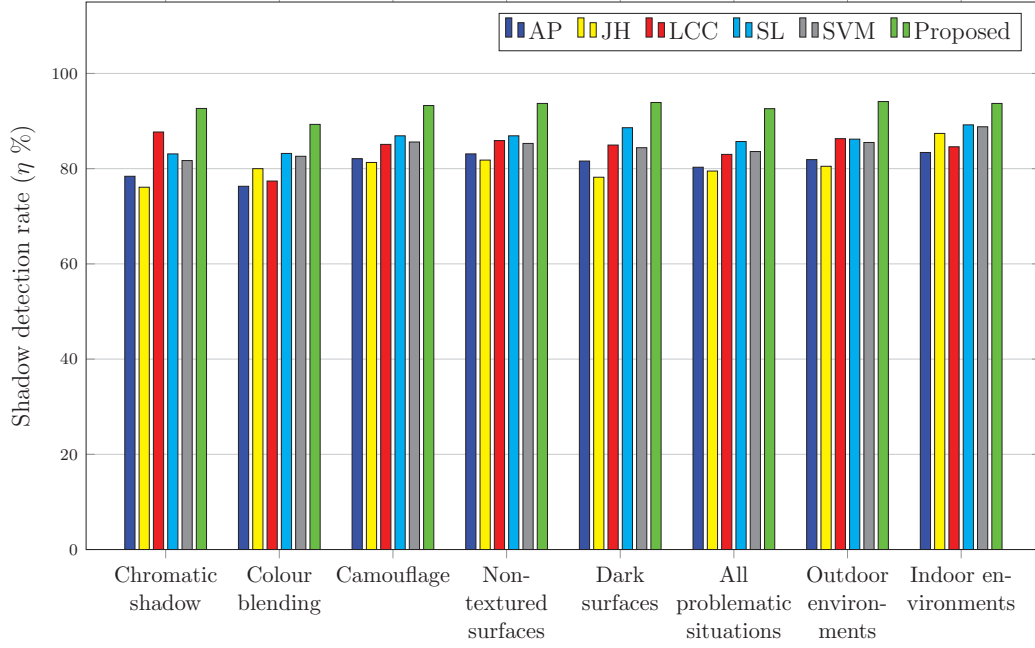
Table 6.11: Average quantitative performance for all shadow detection methods being compared in this experiment.

Methods	AP [138]	JH [139]	LCC [62]	SL [134]	SVM [131]	Proposed
Chromatic shadow	77.2	80.1	87	77.6	82.8	94.9
Colour blending	79.4	83.1	78.3	76.4	86.7	93.1
Camouflage	81.7	83.5	85.6	81.7	86.6	94.6
Non-textured surfaces	81.8	83.8	86.1	81.7	85.6	93.9
Dark surfaces	77.6	80.5	84.2	78.1	84.3	91.4
All problematic situations	79.5	82.2	84.3	79.1	85.1	93.6
Outdoor environments	80.5	83.4	87.0	80.6	86.4	93.2
Indoor environments	85.8	89.5	86.1	85.3	91.2	94.1
Execution time (frames/sec)	21	18	16	14	14	13

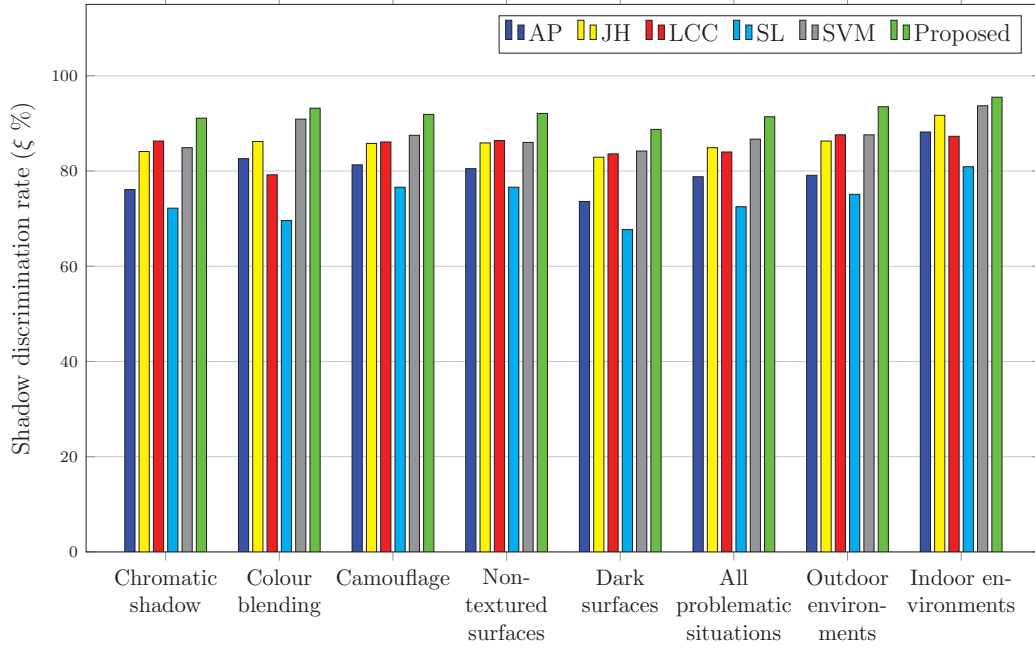
Campus and *Cubicle* - since these sequences present only two problems: foreground-background camouflage and nontextured surfaces. However, the proposed shadow detection method produces superior results for the other datasets, especially for *WBS*, *PPL*, and *Bungalows*.

Figure 6.18 shows a quantitative comparison of the methods for each problematic situation. The two quantitative metrics, i.e., shadow detection rate and shadow discrimination rate, are calculated using the quantitative results obtained from the associated datasets (the datasets identified as having the problematic situations in Table 6.1). The average performance of each method is also calculated for comparison. In addition, the methods are examined against outdoor and indoor environments.

As shown in Figure 6.18 (a), the proposed shadow detection method successfully detected more than 93% of the total shadow points under various conditions. In terms of the shadow discrimination rate, the proposed method obtained a high rate, with over 92% of the total object points correctly classified as objects, as shown in Figure 6.18 (b).



(a) Shadow detection rate



(b) Shadow discrimination rate

Figure 6.18: The pixel-based classification accuracy, in terms of (a) Shadow detection rate (η %) and (b) Shadow discrimination rate (ξ %), of AP [138], JH [139], LCC [62], SL [134], SVM [131], and the proposed moving shadow detection with respect to shadow problematic situations.

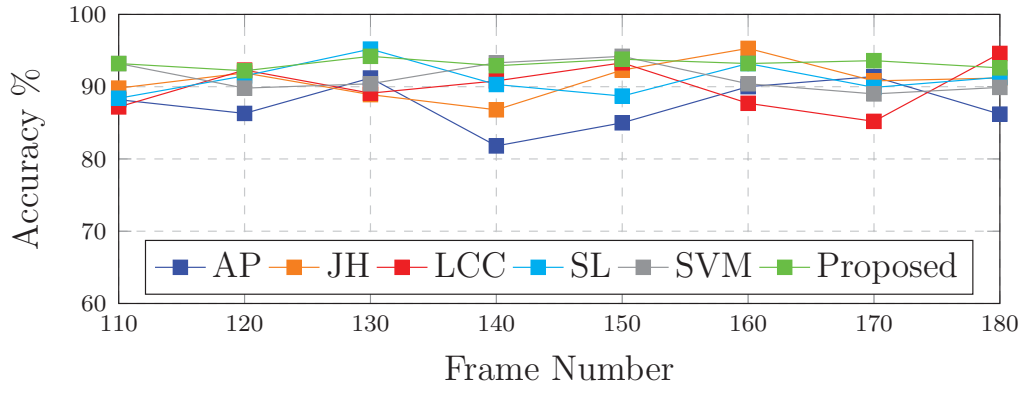
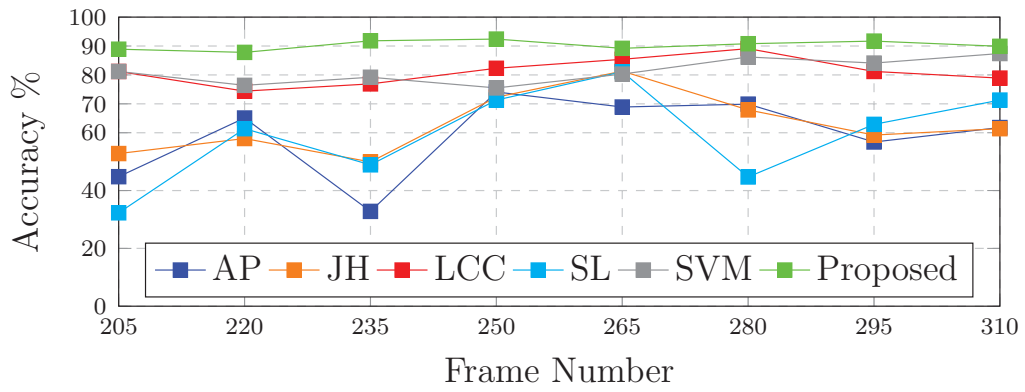
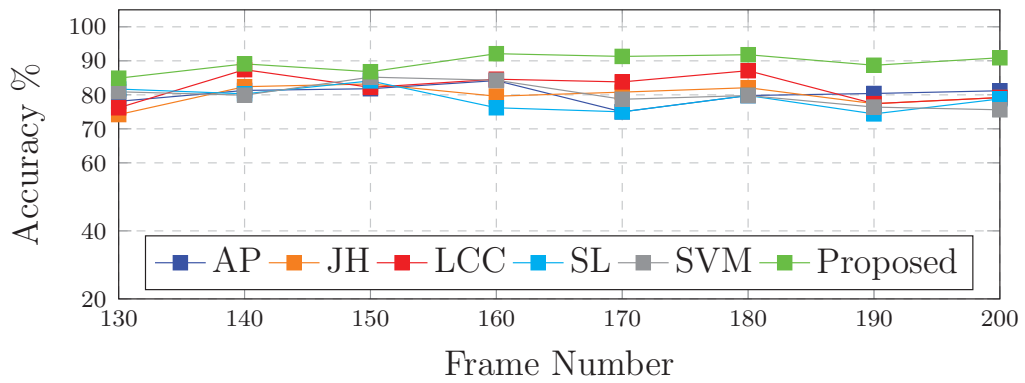
Table 6.11 shows the average accuracy of the methods compared in this experiment. Clearly, as reported in Table 6.11, the proposed joint feature descriptor combined with the **MSR** classifier demonstrates an ability to address all these problems, outperforming the other five methods with an average accuracy of approximately 93.6%.

The computational efficiency of the proposed method is slightly lower than that of all the other methods, including **SVM**. An average of 13 frames per second is achieved using different frame sizes (the frame sizes are reported in Table 6.1), which is considered to be adequate to fulfill the requirements of real-time applications. In addition, it is worthwhile to trade a small amount of computational efficiency for the advantage of accurate detection.

6.7.3 Stability

A constant high detection accuracy over time and maintain stability are often desired. Therefore, the stability of the performance of the shadow detection methods has been examined using Equation (6.6). A highly stable shadow detection method should have difference between the minimum and maximum detection accuracies of close to zero. In practice, a method can be considered to be stable if the difference between the two accuracies is no more than 15%, giving a detection stability above 85%.

Figure 6.19 shows the accuracy over a number of frames taken from three sequences: *Campus*, *PPL*, and *HwyI*. The detection accuracy for eight randomly selected frames from the sequence *Campus* is shown in Figure 6.19 (a). All methods achieve high stability, with detection stabilities of 90.4% for AP, 90.6% for LCC, 91.5% for JH, 93.2% for SL, 94.8% for SVM, and 98.0% for the proposed method.

(a) *Campus*(b) *PPL*(c) *HwyI*Figure 6.19: Stability of shadow detection methods using a number of samples from the sequences; (a) *Campus*, (b) *PPL*, and (c) *HwyI*.

Based on Figure 6.19 (b) (the results for the sequence *PPL*), SL, AP, and JH are the most unstable methods, with detection stabilities of 51.2%, 58%, and 68.6%, respectively. For the SL method, minimum and maximum detection accuracies below 33% and just above 81% are recorded. For the AP method, minimum and maximum detection accuracies of 32% and 74% are obtained. Finally, a minimum accuracy of 49.9% and a maximum accuracy of 81.3% are obtained for JH.

Compared to the above three methods, the LCC and SVM methods achieve high detection stabilities of 85.3% and 88.1%, as shown in Figure 6.19 (b). Meanwhile, the proposed methods can be seen as being more stable (92.8%) given the very low difference between the maximum and minimum detection accuracies when compared to those of the other methods.

For the last sequence, *HwyI*, all the methods are stable since the detection accuracy does not differ significantly between the lowest and highest rates. As shown by the chart in Figure 6.19 (c), the proposed shadow detection method is most stable, with a stability of 92.8%. The second most stable method is JH, with a stability rate of 91.0%. The lowest stability is 88.8%, which is achieved by the LCC method.

6.8 Summary

In this chapter, the results of the proposed change detection method and the proposed moving shadow detection method were presented and discussed. The proposed methods and state-of-the-art methods were compared on well-known shadow detection datasets and the possible problematic situations discussed in Chapter 3. The chapter can be divided into two main parts: experimental setup

and method evaluation.

The experimental setup consists of the used sequences, the evaluation metrics, and the parameter settings. Eight well-known indoor and outdoor moving shadow detection sequences, namely, *Campus*, *Cubicle*, *WBS*, *PPL*, *Seam*, *HwyI*, *HwyIII*, and *Bungalows*, were used to evaluate the methods. The technical details of these sequences, including the associated problematic situations, are summarized in Table 6.1. Six evaluation metrics, i.e., recall, precision, shadow detection rate, shadow discrimination rate, accuracy, and detection stability, were used to evaluate the methods on these sequences. Finally, the major parameters involved in the proposed methods were discussed in detail.

To evaluate the methods, three intensive experiments were conducted on the overall framework proposed in Chapters 4 and 5, the feature descriptors proposed for shadow detection in Chapter 5, and the moving shadow detection proposed in Chapter 5. The performance of the methods and feature descriptors was discussed based on the qualitative and quantitative results. Manually labeled ground truth images were used for all qualitative and quantitative comparisons.

The results showed that the proposed overall framework (the change detection and shadow detection), the proposed feature descriptors (the **BPLCC** and **LGO**), and the proposed shadow detection method performed well under various problematic situations associated with shadow detection.

The first experiment was conducted to evaluate the performance of the proposed comprehensive framework with respect to all sequences and the five main problematic situations. In the experiment, the proposed comprehensive framework performed better than three well-known methods: GABOR, SOBS, and LCC. The proposed framework achieved a high average accuracy of 89.6% under all possible problematic scenarios, an improvement of at least 10% com-

pared to the state-of-the-art comprehensive methods. The proposed framework can produce approximately 10 frames per seconds, which is slower than the SOBS and LCC methods; however, the proposed method can still be used for real-time applications.

In the second experiment, the performance of the proposed feature descriptors (BPLCC and LGO) was evaluated and compared with that of a number of popular state-of-the-art feature descriptors: **LDP**, **CLBP**, **LTP**, **SILTP**, and **LGHP**. The evaluation was performed at the segment-level using the same datasets and reference dictionaries initially selected for this experiment. The results indicate that the proposed BPLCC feature descriptor performed well in terms of shadow detection rate ($\eta = 87.5\%$), shadow discrimination rate ($\xi = 89.5\%$), and average accuracy (88.6%) compared to the proposed LGO feature descriptor ($\eta = 78.8\%$, $\xi = 82.7\%$, and accuracy=80.7%), which was the best of the current state-of-the-art feature descriptors. Therefore, incorporating the color and texture information of the neighboring pixels yields a better feature descriptor that is capable of distinguishing shadows and objects under different environments.

The overall ability of the proposed shadow detection method under the identified problematic situations was evaluated in the final set of experiments. The quantitative and qualitative performance of the proposed shadow detection method in terms of the shadow detection rate, shadow discrimination rate, and detection accuracy was compared with that of five popular moving shadow detection methods: a region-based affinity propagation method **AP**, a joint-histogram-based method **JH**, a region-based local color constancy method with adaptive thresholds **LCC**, a feature-based method with statistical learning **SL**, and another feature-based method with **SVM**. For all these methods, the major parameters were set based on their reference papers.

The AP and SL methods achieved a similar average accuracy of 79% under all problematic situations. Meanwhile, the experimental results show that the JH, LCC, and SVM methods performed better than the AP and SL methods on individual problematic situations. The average accuracy results of these methods were 82%, 84.3%, and 85.1%, respectively. The average accuracy of the proposed shadow detection method under all problematic situations was 93.6%, which is 8.5% higher than the best state-of-the-art method (accuracy of SVM is 85.1%).

In terms of the performance of the methods with respect to the type of environment, the LCC method (accuracy of 87%), SVM method (accuracy of 86.4%), and proposed method (accuracy of 93.2%) are the best choices to detect moving shadows in all outdoor environments. For indoor environments, the best choices are the JH method (accuracy of 89.5%), SVM method (accuracy of 91.2%), and the proposed method (accuracy of 94.1%).

With reference to the execution time, the proposed moving shadow detection method produced approximately 13 frames per second, which is below the execution time of the state-of-the-art methods (AP: 21 frames per second; JH: 18 frames per second; LCC: 16 frames per second; and SL and SVM: 14 frames per second). However, the results of the proposed method are much better than those of all the state-of-the-art methods. As discussed earlier, the computational efficiency of the proposed shadow detection method can be improved by selecting a smaller frame size, which would not affect the overall accuracy of the method since the proposed features are learned locally.

The performance of the shadow detection methods was further evaluated for detection stability. The results obtained for eight samples of the sequences *Campus*, *PPL*, and *HwyI* show that SL (minimum stability of 51.2%), AP (minimum stability of 58%), and JH (minimum stability of 68.6%) are among the most unstable methods in terms of maintaining a constant accuracy. The de-

tection stability rates obtained by LCC and SVM are high, with a minimum stability of 85.3% recorded for LCC and 88.1% for SVM. The proposed method achieved a minimum stability of 92.8%, which is more than 4% higher than that of SVM (the most stable of the state-of-the-art methods).

According to the above discussion, the methods proposed in this thesis, i.e., the change detection method proposed in Chapter 4, the moving shadow detection method proposed in Chapter 5, and the feature descriptor proposed in Chapter 5, have addressed the main aim specified in Chapter 1.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, possible environmental problems in shadow detection systems, including chromatic shadows, shadow color blending, background-foreground camouflage, nontextured surfaces, and dark surfaces, are addressed. A novel comprehensive framework for moving shadow detection is developed to address these problems using a new set of powerful feature descriptors that can be directly extracted from the processing frame without a priori information about the scene.

The proposed framework is divided into two main stages: change detection and moving shadow detection. In the first stage, a new and effective background subtraction method was proposed using a combination of two feature descriptors, color and global intensity consistency GIC, to effectively identify background points under varying global illumination changes and in the presence of image noise.

On the basis of the pixel-level classification process, the background model was formed and updated in a self-organized manner and according to a selective weighted running average. The output of this stage consists of the segmented current frames, the CDMs, and the updated background images segmented into regions.

In the second stage, a novel segment-based moving shadow detection method was proposed to overcome possible problematic situations associated with shadows. A set of two powerful illumination-invariant features - BPLCC and LGO - were used to extract valuable features from image segments, including those collected in the two reference dictionaries. By means of the sparse representation classifier, classification was performed at the segment level by exploring similarities between the new test image feature vector and its best local estimate from each reference dictionary. To achieve better accuracy in the output results, inter-region dependencies among neighboring segments were established to include all dark segments in the labeling process and to enhance the final shadow detection result.

The major advantage of the proposed BPLCC feature descriptor over the traditional state-of-the-art feature descriptors is the ability to distinguish shadow and the object classes from each other in cases with foreground-background camouflage and nontextured surfaces. The proposed LGO feature descriptor takes advantage of the geometric formation of the shadow to design a better gradient descriptor than that of the traditional state-of-the-art feature descriptors.

For classification, a multi-independent sparse representation is used instead of a single representation to overcome the problem of intraclass variation, which often leads to misclassification. The new multi-independent sparse classifier model depends on more than a single class estimate from the dictionary of each class, which, in turn, increases the classification accuracy.

Chapter 7: Conclusion and Future Work

Three set of experiments were conducted to assess the performance of the proposed comprehensive framework, the feature descriptors, and the moving shadow detection method. Eight challenging sequences in indoor and outdoor environments were used in these experiments; each sequence contained at least one of the problematic situations discussed earlier.

In the first set of experiments, the performance of the overall framework was evaluated and compared with that of three popular methods. The experimental results show that the proposed framework can handle various problems associated with image processing, including global illumination changes, image noise, and possible problematic situations related to shadow detection. An overall accuracy of 89.6% was achieved. The accuracy of the proposed approach is at least 10% higher than that of the best state-of-the-art method (LCC), with an accuracy of 79%. In terms of computational efficiency, the proposed method produces fewer frames per second (an average of approximately 10 frames per second) than the other methods; however, the overall performance of the proposed method is much better.

The performance of the proposed feature descriptors - BPLCC and LGO - was evaluated and compared to five well known feature descriptors. The experimental results show that the proposed BPLCC and LGO feature descriptors outperformed the state-of-the-art feature descriptors in terms of shadow detection rate (approximately 87.5% of shadow segments were correctly detected by BPLCC and approximately 78.8% of shadow patches were correctly detected by LGO), shadow discrimination rate (approximately 89.5% of object segments were correctly detected by BPLCC and approximately 82.7% of object segments were correctly detected by LGO), and average accuracy (approximately 88.6% for BPLCC and approximately 80.7% for LGO). These results indicate the improvement of the two proposed feature descriptors compared to the best state-

of-the-art feature descriptors (SILTP with a shadow detection rate of 76.7%; LGHP with a shadow discrimination rate of approximately 80% and an accuracy of approximately 73.4%).

Finally, the performance of the proposed moving shadow detection was evaluated under possible problematic scenarios and compared with five popular moving shadow detection methods. The experimental results illustrate the superior performance of the proposed method in terms of the shadow detection rate, shadow discrimination rate, accuracy, and stability. The improvements helped the proposed method to successfully detect shadow points, resulting in an accuracy of 94.9% for chromatic shadows, 93.1% for shadow color blending, 94.6% for foreground-background camouflage, 93.9% for nontextured surfaces, and 91.4% for dark surfaces, for an average accuracy of 93.6% under all problematic situations, which is 8.5% higher than that of the best state-of-the-art method (SVM, with an average accuracy of 85.1%).

Although the average frame rate (13 frames per second) was lower than that of the current state-of-the-art methods (due to the computational complexity of the multi-independent sparse representation classifier), the results show that the additional time comes with a great reward.

The stability of the methods was finally examined with respect to the accuracy for a number of frames from the sequences. The results show that the proposed method achieved a stability ranging from a minimum of 92.8% to a maximum of 96.8%, indicating that the method is more stable than the state-of-the-art methods.

In conclusion, the aim of this thesis has been achieved by the proposed methods. The proposed methods achieve high performance in terms of the shadow detection rate, shadow discrimination rate, detection accuracy, and sta-

bility. This research has shown that a combination of multi-feature descriptors, specifically designed for shadow detection, yields good detection results under various scenarios with environmental challenges.

7.2 Future Work

The experimental results from the region-based method developed in this thesis show its robustness in different scenarios. The set of the two main feature descriptors, **BPLCC** and **LGO**, has shown the ability to handle the identified problems for shadows. Therefore, the method can be extended and presented in a new framework to eliminate background segments in a given frame affected by global illumination changes. Thus, the two stages of lower-level tasks, namely, object detection and shadow detection, can be combined to detect the actual size of a moving object in less processing time.

The BPLCC feature descriptor has been tested for shadow detection in this research study; however, the descriptor can be used in other computer vision applications, such as face detection and recognition, action recognition, and scene analysis. These applications would provide an additional opportunity to further evaluate this feature descriptor.

The accuracy achieved by the overall shadow detection method was impressive. The only limitation of the proposed method was its speed (frame rate), which was slightly slower than that of the other methods due to using the multi-independent sparse representation classifier that required more time to find the best matches from each reference dictionary. The speed may be improved by reducing the frame size or by reducing the number of samples from each reference dictionary. Alternative solutions would be to use a time-efficient feature

classifier instead of the proposed sparse representation classifier or to learn the proposed feature descriptors using a deep neuronal network to automatically recognize background-scene image patches under these problematic situations.

On the basis of the above discussion, this research could be extended in various directions, including applications of the proposed feature descriptors and improving the speed of the overall method without affecting the average accuracy.

References

- [1] A. Russell and J. J. Zou, “Moving shadow detection based on spatial-temporal constancy,” in *IEEE Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–6, 2013.
- [2] A. Sanin, C. Sanderson, and B. C. Lovell, “Improved shadow removal for robust person tracking in surveillance scenarios,” in *International Conference on Pattern Recognition (ICPR)*, pp. 141–144, 2010.
- [3] H. Asaidi, A. Aarab, and M. Bellouki, “Shadow elimination and vehicles classification approaches in traffic video surveillance context,” *Journal of Visual Languages and Computing*, vol. 25, no. 4, pp. 333–345, 2014.
- [4] S. K. Meher and M. Murty, “Efficient method of moving shadow detection and vehicle classification,” *AEU-International Journal of Electronics and Communications*, vol. 67, no. 8, pp. 665–670, 2013.
- [5] B. Johansson, J. Wiklund, P. Forssn, and G. Granlund, “Combining shadow detection and simulation for estimation of vehicle size and position,” *Pattern Recognition Letters*, vol. 30, no. 8, pp. 751–759, 2009.
- [6] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi, “Understanding transit scenes: A survey on human behavior-recognition algo-

- rithms,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 206–224, 2010.
- [7] T. Gandhi, R. Chang, and M. M. Trivedi, “Video and seismic sensor-based structural health monitoring: Framework, algorithms, and implementation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 169–180, 2007.
- [8] Z. Ruo, P. S. Tsai, J. E. Cryer, and M. Shah, “Shape-from-shading: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.
- [9] J. Norman, Y. Lee, F. Phillips, H. Norman, L. Jennings, and T. McBride, “The perception of 3-D shape from shadows cast onto curved surfaces,” *Acta Psychologica*, vol. 131, no. 1, pp. 1–11, 2009.
- [10] A. O. Ok, “Automated detection of buildings from single VHR multi-spectral images using shadow information and graph cuts,” *Journal of Photogrammetry and Remote Sensing (ISPRS)*, vol. 86, pp. 21–40, 2013.
- [11] I. Sato, Y. Sato, and K. Ikeuchi, “Illumination from shadows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 290–300, 2003.
- [12] L. Yang, T. Gevers, and L. Xueqing, “Estimation of sunlight direction using 3D object models,” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 932–942, 2015.
- [13] L. Wu, X. Cao, and H. Foroosh, “Camera calibration and geo-location estimation from two shadow trajectories,” *Computer Vision and Image Understanding*, vol. 114, no. 8, pp. 915–927, 2010.

- [14] M. Iiyama, H. Koki, K. Kakusho, and M. Minoh, “Usage of needle maps and shadows to overcome depth edges in depth map reconstruction,” in *International Conference on Pattern Recognition (ICPR)*, pp. 1–4, 2008.
- [15] M. D. Levine and J. Bhattacharyya, “Removing shadows,” *Pattern Recognition Letters*, vol. 26, no. 3, pp. 251–265, 2005.
- [16] W. Deng, J. Hu, J. Guo, W. Cai, and D. Feng, “Robust, accurate and efficient face recognition from a single training image: A uniform pursuit approach,” *Pattern Recognition*, vol. 43, no. 5, pp. 1748–1762, 2010.
- [17] X.-H. Cai, Y.-T. Jia, X. Wang, S.-M. Hu, and R. R. Martin, “Rendering soft shadows using multi-layered shadow fins,” *Computer Graphics Forum*, vol. 25, no. 1, pp. 15–28, 2006.
- [18] L. Petrovi, B. Fujito, L. Williams, and A. Finkelstein, “Shadows for cel animation,” in *Proceedings of the 27th annual Conference on Computer Graphics and Interactive Techniques*, pp. 511–516, ACM Press/Addison-Wesley Publishing Co., 2000.
- [19] S.-M. Hu, F.-L. Zhang, M. Wang, R. R. Martin, and J. Wang, “PatchNet: a patch-based image representation for interactive library-driven image editing,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 196–208, 2013.
- [20] S.-P. Lu, S.-H. Zhang, J. Wei, S.-M. Hu, and R. R. Martin, “Timeline editing of objects in video,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1218–1227, 2013.
- [21] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin, “Internet visual media processing: a survey with graphics and vision applications,” *The Visual Computer*, vol. 29, no. 5, pp. 393–405, 2013.

- [22] A. Sanin, C. Sanderson, and B. C. Lovell, “Shadow detection: A survey and comparative evaluation of recent methods,” *Pattern Recognition*, vol. 45, no. 4, pp. 1684–1695, 2012.
- [23] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, “Detecting moving shadows: algorithms and evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918–923, 2003.
- [24] “Datasets with ground truths are available at.” <http://arma.sourceforge.net/shadows/>.
- [25] “Change detection datasets with ground truths are available at.” <http://www.changedetection.net/>.
- [26] “Cvrr-aton datasets with ground truths are available at.” <http://cvrr.ucsd.edu/aton/shadow/>.
- [27] “Fgshbenchmark datasets with ground truths are available at.” <http://web.eee.sztaki.hu/~bcsaba/\\FgShBenchmark.htm>.
- [28] N. Al-Najdawi, H. E. Bez, J. Singhai, and E. A. Edirisinghe, “A survey of cast shadow detection algorithms,” *Pattern Recognition Letters*, pp. 752–764, 2012.
- [29] C. C. Chen and J. Aggarwal, “Human shadow removal with unknown light source,” in *International Conference on Pattern Recognition (ICPR)*, pp. 2407–2410, 2010.
- [30] A. Yoneyama, C. H. Yeh, and C. C. J. Kuo, “Robust vehicle and traffic information extraction for highway surveillance,” *EURASIP Journal on Applied Signal Processing*, pp. 2305–2321, 2005.

- [31] J. W. Hsieh, W. F. Hu, C. J. Chang, and Y. S. Chen, "Shadow elimination for effective moving object detection by Gaussian shadow modelling," *Image and Vision Computing*, vol. 21, pp. 505–516, 2003.
- [32] B. Sheng, L. Dequn, S. Xiaoyan, and W. Qiang, "Human cast shadow elimination method based on orientation information measures," in *IEEE International Conference on Automation and Logistics*, pp. 1567–1571, 2007.
- [33] A. Yoneyama, C. H. Yeh, and C. C. J. Kuo, "Moving cast shadow elimination for robust vehicle extraction based on 2-D joint vehicle/shadow models," in *International Conference on Advanced Video and Signal Based Surveillance*, pp. 229–236, 2003.
- [34] S. Nadimi and B. Bhanu, "Moving shadow detection using a physics-based approach," in *International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 701–704, 2002.
- [35] L. Z. Fang, W. Y. Qiong, and Y. Z. Sheng, "A method to segment moving vehicle cast shadow based on wavelet transform," *Pattern Recognition Letters*, pp. 2182–2188, 2008.
- [36] J. M. Wang, Y. C. Chung, and S. W. Chen, "Shadow detection and removal for traffic images," in *IEEE International Conference on Networking, Sensing and Control*, pp. 649–654, 2004.
- [37] H. Nicolas and J. M. Pinel, "Joint moving cast shadows segmentation and light source detection in video sequences," *Signal Processing: Image Communication*, vol. 21, no. 1, pp. 22–43, 2006.
- [38] A. J. Joshi and N. P. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2055–2063, 2008.

- [39] P. Kumar and K. Sengupta, “A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system,” in *IEEE the 5th International Conference on Intelligent Transportation Systems*, pp. 100–105, 2002.
- [40] Y. Shan, F. Yang, and R. Wang, “Color space selection for moving shadow elimination,” in *4th International Conference on Image and Graphics*, pp. 496–501, 2007.
- [41] M. Subramanyam, K. Nallaperumal, R. Subban, P. Pasupathi, D. Shashikala, S. Kumar, and G. S. Devi, “A study and analysis of colour space selection for insignificant shadow detection,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 2, no. 12, pp. 2476–2480, 2013.
- [42] R. Cucchiara, C. Grana, and M. Piccardi, “Detecting moving objects, ghosts and shadows in video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.
- [43] Y. P. Guan, “Spatio-temporal motion-based foreground segmentation and shadow suppression,” *IET Computer Vision*, vol. 4, no. 1, pp. 50–60, 2010.
- [44] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Cast shadow segmentation using invariant color features,” *Computer Vision and Image Understanding*, vol. 95, no. 2, pp. 238–259, 2004.
- [45] R. Melli, A. Prati, R. Cucchiara, and L. de Cock, “Predictive and probabilistic tracking to detect stopped vehicles,” in *Seventh IEEE Workshops on Application of Computer Vision*, vol. 1, pp. 388–393, 2005.

- [46] A. Cavallaro, E. Salvador, and T. Ebrahimi, “Shadow-aware object-based video processing,” *IEEE Proceedings Vision, Image and Signal Processing*, vol. 152, no. 4, pp. 398–406, 2005.
- [47] L. Kuo-Hua and Y. Mau-Tsuen, “Shadow detection by integrating multiple features,” in *International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 743–746, 2006.
- [48] B. Sun and S. Li, “Moving cast shadow detection of vehicle using combined color models,” in *Chinese Conference on Pattern Recognition*, pp. 1–5, 2010.
- [49] S. Ishida, S. Fukui, Y. Iwahori, M. K. Bhuyan, and R. J. Woodham, “Shadow detection by three shadow models with features robust to illumination changes,” *Procedia Computer Science*, vol. 35, pp. 1219–1228, 2014.
- [50] J. Stander, R. Mech, and J. Ostermann, “Detection of moving cast shadows for object segmentation,” *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 65–76, 1999.
- [51] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, *The Sakbot system for moving object detection and tracking*, pp. 145–157. Springer, 2002.
- [52] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.
- [53] I. Mikic, P. C. Cosman, G. T. Kogut, and M. M. Trivedi, “Moving shadow and object detection in traffic scenes,” in *International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 321–324, 2000.

- [54] D. Toth, I. Stuke, A. Wagner, and T. Aach, "Detection of moving shadows using mean shift clustering and a significance test," in *International Conference on Pattern Recognition (ICPR)*, vol. 4, pp. 260–263, 2004.
- [55] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with HSV color information," in *Proceedings of IEEE on Intelligent Transportation Systems*, pp. 334–339, IEEE, 2001.
- [56] S. Ishida, S. Fukui, Y. Iwahori, M. Bhuyan, and R. J. Woodham, "Shadow model construction with features robust to illumination changes," in *Proceedings of the World Congress on Engineering*, vol. 3, pp. 2195–2200, 2013.
- [57] J. Dai, M. Qi, J. Wang, J. Dai, and J. Kong, "Robust and accurate moving shadow detection based on multiple features fusion," *Optics and Laser Technology*, vol. 54, pp. 232–241, 2013.
- [58] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 99, pp. 1–19, 1999.
- [59] J. Choi, Y. J. Yoo, and J. Y. Choi, "Adaptive shadow estimator for removing shadow of moving object," *Computer Vision and Image Understanding*, vol. 114, no. 9, pp. 1017–1029, 2010.
- [60] K. T. Song and J. C. Tai, "Image-based traffic monitoring with shadow suppression," in *the IEEE*, pp. 413–426, 2007.
- [61] J. Wang, Y. Wang, M. Jiang, X. Yan, and M. Song, "Moving cast shadow detection using online sub-scene shadow modeling and object inner-edges

- analysis,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 978–993, 2014.
- [62] A. Amato, M. G. Mozerov, A. D. Bagdanov, and J. Gonzalez, “Accurate moving cast shadow suppression based on local color constancy detection,” *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2954–2966, 2011.
- [63] C. H. Huang and R. C. Wu, “An online learning method for shadow detection,” in *2010 Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pp. 145–150, 2010.
- [64] A. Leone and C. Distanto, “Shadow detection for moving objects based on texture analysis,” *Pattern Recognition Letters*, vol. 40, pp. 1222–1233, 2007.
- [65] M.-T. Yang, K.-H. Lo, C.-C. Chiang, and W.-K. Tai, “Moving cast shadow detection by exploiting multiple cues,” *IET Image Processing*, vol. 2, no. 2, pp. 95–104, 2008.
- [66] R. Qin, S. Liao, and S. Z. Lei, Z. and Li, “Moving cast shadow removal based on local descriptors,” in *International Conference on Pattern Recognition (ICPR)*, pp. 1377–1380, 2010.
- [67] J. B. Huang and C. S. Chen, “Moving cast shadow detection using physics-based features,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2310–2317, 2009.
- [68] N. Martel-Brisson and A. Zaccarin, “Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

- [69] N. Martel-Brisson and A. Zaccarin, “Learning and removing cast shadows through a multidistribution approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1133–1146, 2007.
- [70] L. Yiyang and D. Adjeroh, “A statistical approach for shadow detection using spatio-temporal contexts,” in *17th IEEE International Conference on Image Processing (ICIP)*, pp. 3457–3460, 2010.
- [71] J. B. Huang and C. S. Chen, “A physical approach to moving cast shadow detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 769–772, 2009.
- [72] M. Khare, R. K. Srivastava, and A. Khare, “Moving shadow detection and removal - a wavelet transform based approach,” *IET Computer Vision*, vol. 8, no. 6, pp. 701–717, 2014.
- [73] C. R. Jung, “Efficient background subtraction and shadow removal for monochromatic video sequences,” *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 571–577, 2009.
- [74] M. Khare, R. K. Srivastava, and A. Khare, “Daubechies complex wavelet based computer vision applications,” *Recent developments in biometrics and video processing techniques*, pp. 138–155, 2013.
- [75] J. Dai, D. Han, and X. Zhao, “Effective moving shadow detection using statistical discriminant model,” *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 24, pp. 5398–5406, 2015.
- [76] E. Bullkich, I. Ilan, Y. Moshe, Y. Hel-Or, and H. Hel-Or, “Moving shadow detection by nonlinear tone-mapping,” in *International Conference on Systems, Signals and Image Processing, Vienna, Austria*, pp. 146–149, 2012.

- [77] I. Huerta, M. B. Holte, T. B. Moeslund, and J. Gonzlez, “Chromatic shadow detection and tracking for moving foreground segmentation,” *Image and Vision Computing*, vol. 41, pp. 42–53, 2015.
- [78] D. Xu, X. Li, Z. Liu, and Y. Yuan, “Cast shadow detection in video segmentation,” *Pattern Recognition Letters*, vol. 26, no. 1, pp. 91–99, 2005.
- [79] Z. Wei, F. Xiang Zhong, X. K. Yang, and Q. M. J. Wu, “Moving cast shadows detection using ratio edge,” *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1202–1214, 2007.
- [80] M. Xiao, C.-Z. Han, and L. Zhang, “Moving shadow detection and removal for traffic sequences,” *International Journal of Automation and Computing*, vol. 4, no. 1, pp. 38–46, 2007.
- [81] C. Shao-Yi, M. Shyh-Yih, and C. Liang-Gee, “Efficient moving object segmentation algorithm using background registration technique,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577–586, 2002.
- [82] N. Martel-Brisson and A. Zaccarin, “Moving cast shadow detection from a Gaussian mixture shadow model,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, vol. 2, pp. 643–648, 2005.
- [83] Y.-M. Wu, X.-Q. Ye, and W.-K. Gu, “A shadow handler in traffic monitoring system,” in *IEEE Conference on Vehicular Technology (VTC)*, vol. 1, pp. 303–307, 2002.
- [84] J. Panicker and M. Wilscy, “Detection of moving cast shadows using edge information,” in *International Conference on Computer and Automation Engineering*, pp. 817–821, 2010.

- [85] S. Huang, B. Liu, and W. Wang, "Moving shadow detection based on SUSAN algorithm," in *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 3, pp. 16–20, 2011.
- [86] E. ShabaniNia and A. R. Naghsh-Nilchi, "Robust watershed segmentation of moving shadows using wavelets," in *8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 381–386, 2013.
- [87] S. M. Smith and J. M. Brady, "SUSAN: A new approach to low level image processing," *International journal of computer vision*, vol. 23, no. 1, pp. 45–78, 1997.
- [88] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *European Conference on Computer Vision*, pp. 343–357, 2002.
- [89] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting objects, shadows and ghosts in video streams by exploiting color and motion information," in *11th International Conference on Image Analysis and Processing*, pp. 360–365, 2001.
- [90] Z. Wei, F. Xiang Zhong, and X. Yi, "Detection of moving cast shadows using image orthogonal transform," in *International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 626–629, 2006.
- [91] H. Celik, A. M. Ortigosa, A. Hanjalic, and E. A. Hendriks, "Autonomous and adaptive learning of shadows for surveillance," in *Ninth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 59–62, IEEE, 2008.
- [92] C. Benedek and T. Szirnyi, *Shadow detection in digital images and videos*. Boca Raton, FL, USA: CRC Press, 2011.

- [93] F. Cogun and A. Cetin, "Moving shadow detection in video using cepstrum," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [94] J. Dai and D. Han, "Region-based moving shadow detection using affinity propagation," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 3, pp. 65–74, 2015.
- [95] P. Kubelka, "New contribution to the optics of intensity light-scattering materials, part I," *Journal of Optical Society of America*, vol. 38, no. 5, pp. 448–457, 1948.
- [96] B.-T. Phong, "Illumination for computer generated images," *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.
- [97] S. K. Singh, C. A. Dhawale, and S. Misra, "Survey of object detection methods in camouflaged image," *IERI Procedia*, vol. 4, pp. 351–357, 2013.
- [98] S. Li, D. Florencio, W. Li, Y. Zhao, and C. Cook, "A fusion framework for camouflaged moving foreground detection in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3918–3930, 2018.
- [99] X. Zhang, C. Zhu, S. Wang, Y. Liu, and M. Ye, "A Bayesian approach to camouflaged moving object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 2001–2013, 2017.
- [100] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [101] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,"

- IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [102] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
 - [103] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, “Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1301–1306, IEEE, 2010.
 - [104] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, “A survey on object detection and tracking methods,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 2970–2978, 2014.
 - [105] D. Zhou and H. Zhang, “Modified GMM background modeling and optical flow for detection of moving objects,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2224–2229, IEEE, 2005.
 - [106] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
 - [107] A. Talukder and L. Matthies, “Real-time detection of moving objects from moving vehicles using dense stereo and optical flow,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 3718–3725, IEEE, 2004.

- [108] A. Russell and J. J. Zou, “Vehicle detection based on color analysis,” in *International Symposium on Communications and Information Technologies (ISCIT)*, pp. 620–625, IEEE, 2012.
- [109] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, and P. Burt, “A system for video surveillance and monitoring,” *VSAM final report*, pp. 1–68, 2000.
- [110] S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, S.-Y. Ma, and L.-G. Chen, “Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques,” *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 732–748, 2004.
- [111] M. Piccardi, “Background subtraction techniques: a review,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, IEEE, 2004.
- [112] L. Maddalena and A. Petrosino, “A self-organizing approach to background subtraction for visual surveillance applications,” *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.
- [113] H. Han, J. Zhu, S. Liao, Z. Lei, and S. Z. Li, “Moving object detection revisited: Speed and robustness,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 910–921, 2015.
- [114] L. Maddalena and A. Petrosino, “Towards benchmarking scene background initialization,” in *International Conference on Image Analysis and Processing*, pp. 469–476, Springer, 2015.
- [115] A. Colombari and A. Fusiello, “Patch-based background initialization in heavily cluttered video,” *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 926–933, 2010.

- [116] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, IEEE, 2005.
- [117] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB.*, vol. 624. Pearson-Prentice-Hall Upper Saddle River, New Jersey, 2004.
- [118] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [119] Y. Hel-Or, H. Hel-Or, and E. David, “Matching by tone mapping: Photometric invariant template matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 317–330, 2014.
- [120] Y. Hel-Or, H. Hel-Or, and E. David, “Fast template matching in non-linear tone-mapped images,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1355–1362, IEEE, 2011.
- [121] R. Brunelli, *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.
- [122] J. Y. Choi, K. N. Plataniotis, and Y. M. Ro, “Using colour local binary pattern features for face recognition,” in *17th IEEE International Conference on Image Processing (ICIP)*, pp. 4541–4544, IEEE, 2010.
- [123] B. Zhang, Y. Gao, S. Zhao, and J. Liu, “Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor,” *IEEE transactions on Image Processing*, vol. 19, no. 2, pp. 533–544, 2010.

- [124] S. Chakraborty, S. Singh, and P. Chakraborty, "Local gradient hexa pattern: A descriptor for face recognition and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.
- [125] Y. Song, W. Cai, Y. Zhou, and D. Feng, "Feature-based image patch approximation for lung tissue classification," *IEEE Transactions on Medical Imaging*, vol. 32, no. 4, pp. 797–808, 2013.
- [126] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.
- [127] J. Wang, Y. Wang, M. Jiang, and X. Yan, "A sub-scene modeling framework for moving cast shadow detection," in *IEEE International Conference on Image Processing*, pp. 1663–1667, 2014.
- [128] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [129] S. Meher and M. Murty, "Efficient method of moving shadow detection and vehicle classification," *International Journal of Electronics and Communications*, vol. 67, no. 8, pp. 665–670, 2013.
- [130] J. Dong, L. Su, and Y. Duan, "Illumination estimation of 3D surface texture based on active basis," in *International Conference on Pattern Recognition*, pp. 870–873, 2010.
- [131] A. J. Joshi and N. P. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2055–2063, 2008.

- [132] C. Benedek and T. Sziranyi, “Bayesian foreground and shadow detection in uncertain frame rate surveillance videos,” *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 608–621, 2008.
- [133] V. Suriya, S. Poullose, and S. Anila, “A shackle process for shadow detection,” *International Journal of computer Applications*, pp. 8–15, 2013.
- [134] J.-B. Huang and C.-S. Chen, “Moving cast shadow detection using physics-based features,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2310–2317, 2009.
- [135] Y. Song, W. Cai, H. Huang, Y. Zhou, D. Feng, Y. Wang, M. J. Fulham, and M. Chen, “Large margin local estimate with applications to medical image classification,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1362–1377, 2015.
- [136] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [137] A. Leone and C. Distanto, “Shadow detection for moving objects based on texture analysis,” *Pattern Recognition*, vol. 40, no. 4, pp. 1222–1233, 2007.
- [138] J. Dai and D. Han, “Region-based moving shadow detection using affinity propagation,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 3, pp. 65–74, 2015.
- [139] Y. Su, A. Li, Y. Cai, G. Feng, and G. Jin, “Moving shadow detection with multifeature joint histogram,” *Journal of Electronic Imaging*, vol. 23, no. 5, p. 053015, 2014.

- [140] Y. Hel-Or and D. Shaked, “A discriminative approach for wavelet denoising,” *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 443–457, 2008.

Appendices

Appendix A: MTM Matching

Let S be a segment containing N pixels taken from a grayscale image, and let i be the index of a pixel in S , i.e., $S(i) : i = \{1, 2, \dots, N\}$. The image segment S can be represented as a column vector V of dimensions $N \times 1$. Let j be an index of a bin $\alpha_j : j = \{1, 2, \dots, k\}$ obtained by dividing the intensity levels L into k bins. Thus, α_j is a vector that contains all the values within a range of $[(j-1) \cdot \frac{L}{k}, j \cdot \frac{L}{k})$.

The total number of bins k can be determined based on the number of intensity levels L available in V . k may have any value within the range of $[1, L]$, i.e., $k = 2^a : a = \{0, 1, 2, \dots, \log_2(L)\}$.

By means of the slice transform **SLT** technique described in [140], k slice binary vectors are produced from the column vector V , each representing the entries of V corresponding to the j th bin. The intensity value of each pixel $i \in V$ is determined according to the following formula:

$$V^j(i) = \begin{cases} 1, & \text{if } V(i) \in \alpha_j \\ 0, & \text{otherwise} \end{cases} \quad (\text{A.1})$$

Appendix A: MTM Matching

An approximated grayscale vector V' to that of the original column vector V can then be obtained from a linear combination of k slice vectors:

$$V' \approx \sum_j^k V^j \cdot \alpha_j \quad (\text{A.2})$$

A larger number of bins results in a better approximation of the original column vector V . In fact, an identical version of V can be achieved by setting all discrete values of L as individual bin values, i.e., $k = L$. Equation (A.2) can then be written as:

$$V' = \sum_j^k V^j \cdot \alpha_j \quad (\text{A.3})$$

where α_j in Equation (A.3) contains all the values within the range of $[(j-1), j)$.

The binary slice vectors are concatenated to form a slice matrix \mathcal{M} with dimensions $\mathcal{R}^{N \times k}$. Each element of the slice matrix \mathcal{M} is represented as m_{ij} : $[i = \{1, 2, \dots, N\}, j = \{1, 2, \dots, k\}]$, and the weights assigned to each bin $\alpha_j : j = \{1, 2, \dots, k\}$. Thus, Equation (A.2) can be represented as:

$$V' = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1k} \\ m_{21} & m_{22} & \dots & m_{2k} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ m_{N1} & m_{N2} & \dots & m_{Nk} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \quad (\text{A.4})$$

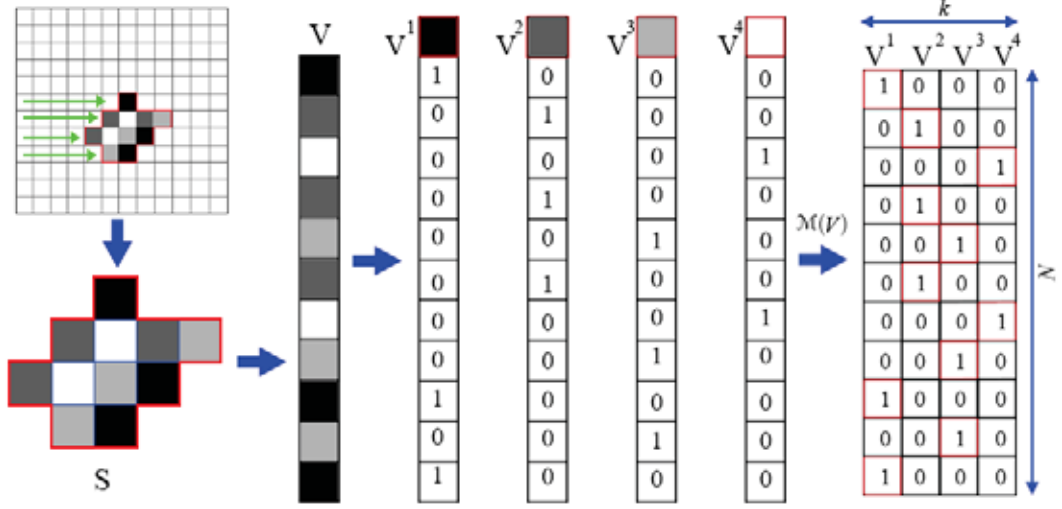


Figure A.1: Formation of the slice transform **SLT** matrix for a small segment with 4 gray values. The column vector V is obtained by scanning the image segment S row-by-row. The column vector is then mapped into a collection of 4 binary slices, each represents one gray-level (one bin). These binary slices are then used to form the **SLT** matrix with a dimension of 11 samples and 4 features (bins).

Using a different weight vector β to that of α results in another image segment (column vector) with a constant tone mapping of V :

$$W' = \mathcal{M} \cdot \beta \quad (\text{A.5})$$

where β has the same length as α . W' is a version of V constructed using the **SLT** matrix of V and a new weight vector β .

Figure A.1 shows the steps for producing binary slices and a **SLT** matrix \mathcal{M} for a small image segment containing 11 pixels and 4 gray-level values. The image segment is first transformed into a $N \times 1$ column vector V . Given 4 gray levels for V , the gray value for each pixel in V is then mapped into a collection of 4 binary column vectors $V^j : j = \{1, 2, 3, 4\}$. These binary column vectors are used to form a slice matrix \mathcal{M} with dimensions $\mathcal{R}^{11 \times 4}$.

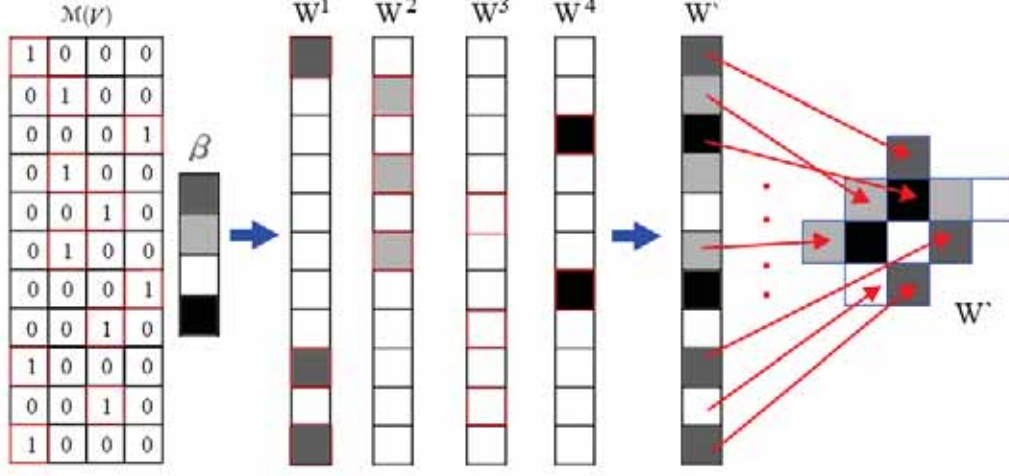


Figure A.2: Construction steps of an image segment W' using the **SLT** matrix of S and a different weight vector β .

Figure A.2 illustrates the steps for constructing a grayscale image segment using Equation (A.5). The image contains 11 pixels with 4 gray levels, which are the weight values of the bins in β .

Given a new image segment organized in a column vector W with the same size and grayscale levels as those of V , the similarity between V and W can be computed using the **MTM** distance measure described in [120]:

$$d(V, W) = \min_{\beta} \frac{\|\mathcal{M}(V)\beta - W\|}{N \text{var}(W)} \quad (\text{A.6})$$

where $W' = \mathcal{M}(V)\beta$, as described in Equation (A.5), and $\text{var}(W)$ is the sample variance of W , which is used along with the sample size N to normalize the distance. $\text{var}(W)$ is computed as follows:

$$\text{var}(W) = \sum_i^N W(i)^2 - \frac{1}{N} \left[\sum_i^N W(i) \right]^2, \quad \forall i \in W \quad (\text{A.7})$$

Appendix A: MTM Matching

Solving the optimization problem presented in Equation (A.6) for all possible values of β yields:

$$d(V, W) = \frac{1}{N \cdot \text{var}(W)} \left[\|W\|^2 - \sum_j^k \frac{1}{|V^j|} \{V^j \cdot W\}^2 \right]. \quad (\text{A.8})$$

where $\|W\|^2$ and $|V^j|$ represent the sum of square intensity values for all pixels $i \in W$ and the sum of ones in a binary slice vector V^j , respectively.

Appendix B: Other Common Datasets



(a) *backdoor*



(b) *busStation*



(c) *hallway*



(d) *highway*



(e) *highwayII*



(f) *laboratory*



(g) *PETS2006*



(h) *intelligent room*

Figure B.1: Other common sequences used to evaluate moving shadow detection methods. The sequences *backdoor*, *busStation*, *highway*, and *PETS2006* can be downloaded from [25] and the sequences *hallway*, *highwayII*, *laboratory*, and *intelligent room* can be downloaded from [26].