

An assessment of the current state of background
model evaluation and a proposal for a fair and
comprehensive evaluation framework and
methodology

By

Sarah Conway

Supervisor: Dr. Kenneth Dawson-Howe

A dissertation submitted to the University of Dublin, in partial fulfilment of the requirements
for the degree of M.A.I. (St.)

Submitted to the University of Dublin, Trinity College, April, 2014

Declaration

I, Sarah Conway, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signature:

Date:

Summary

The ability to model the background of a scene is of great importance in many computer vision applications. This, however, is not a straightforward task as there are many diverse challenges which complicate the process of modelling a scene background. None of the numerous background modelling techniques that have been developed to date are capable of dealing with all challenges with which they may be presented and thus, different techniques are suited to different applications. In order to fully appreciate the capabilities of the various background modelling techniques that exist and to assess their suitability for use in various scenarios it is essential that their performance be comprehensively evaluated.

This dissertation comprises two major components. The first of these is the conduction of a thorough analysis and appraisal of the efforts that have been made in evaluating background modelling techniques¹ to date as reported in the relevant literature. Through this assessment, a number of limitations and weaknesses with how background model evaluations are currently being carried out were identified. Existing evaluations are not, for example, providing comprehensive or objective assessment of background modelling techniques nor are they being conducted in a manner that facilitates the effective comparison of different techniques. In addition, the integrity of the results of these existing evaluations cannot be guaranteed. The poor state of current evaluations is a significant shortcoming of existing background modelling research. Using the findings of this analysis, the second aspect of the dissertation was completed. This involved the determination of the way in which background modelling techniques should be evaluated in order to address the existing limitations, as well as the creation of a standard evaluation framework to allow such evaluations to be carried out.

A significant contribution of this dissertation is the provision of an extensive survey regarding the way in which background modelling techniques are currently being evaluated. This survey provides the research community with a detailed review of the current state of background model evaluation. It allows patterns in how evaluations are being carried out to be seen and allows the weaknesses and limitations that currently exist to easily be identified. The provision of this information may be used to direct future research in the field.

Additionally, the proposed evaluation framework provides a method of comprehensively, objectively and easily evaluating background models in a manner that will ensure that results are accurate, credible and comparable, something which is not currently possible. This will enable the reasonably straightforward compilation of an extensive background model evaluation result reference. Such a resource would be of enormous benefit to the research community in assessing the progress that is being made in background model development and in analysing the weaknesses of the existing body of background modelling techniques in order to direct further development. In addition, it would be of great use to developers wishing to assess their models against those which already exist and would be of benefit in the selection of an appropriate background model for a particular application. As well as this, the ideas of the framework could be applied to the assessment of other types of computer vision algorithms.

¹ The terms “background modelling techniques” and “background models” are used interchangeably.

Acknowledgements

Firstly, I would like to thank my supervisor Dr. Kenneth Dawson-Howe for the guidance and support that he has given me throughout this project.

I would also like to thank my parents, Esther and David for their support and encouragement throughout and for the many hours of proof reading.

Finally, I would like to thank Robert for his encouragement and for his endless patience.

Table of Contents

Declaration	i
Summary	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	ix
List of Tables	xi
Abbreviations	xii
Abstract	xiii
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Project Objectives	1
1.3 Report Structure	2
Chapter 2 Background Subtraction	3
2.1 Basic Background Subtraction Algorithm	3
2.2 Background Modelling	4
2.2.1 Challenges	5
2.2.1.1 Object of Interest	5
2.2.1.2 Updating the Background	5
2.2.1.3 Camouflage	6
2.2.1.4 Illumination Changes.....	6
2.2.1.5 Dynamic Background	6
2.2.1.6 Weather Conditions	6
2.2.1.7 Shadow.....	7
2.2.1.8 Bootstrapping.....	7
2.2.1.9 Camera Characteristics	7
2.2.2 Background Modelling Techniques.....	8
2.2.2.1 Static Frame Difference	9
2.2.2.2 Frame Difference	9
2.2.2.3 Weighted Moving Mean	9
2.2.2.4 Adaptive Background Learning	9

2.2.2.5	Gaussian Mixture Model.....	10
2.2.2.6	KDE.....	10
2.3	Overview of Background Subtraction	11
Chapter 3	Background Model Evaluation	12
3.1	Evaluation Methodology.....	12
3.1.1	Video Datasets	13
3.1.1.1	Existing Datasets	13
3.1.1.2	Dataset Size	15
3.1.1.3	Video Length	16
3.1.1.4	Video Type	16
3.1.1.5	Challenges Considered in Evaluation	17
3.1.1.6	Current State of Dataset Usage in Background Model Evaluation	19
3.1.2	Ground Truth	19
3.1.2.1	Ground Truth Type.....	19
3.1.2.2	Ground Truth Creation.....	23
3.1.2.3	Current State of Ground Truth Usage in Background Model Evaluation	26
3.1.3	Evaluation Metrics	26
3.1.3.1	Accuracy Metrics.....	26
3.1.3.2	Efficiency Metrics.....	29
3.1.3.3	Training Phase Length.....	30
3.1.3.4	Lag.....	30
3.1.3.5	Current State of Evaluation Metric Usage in Background Model Evaluation	30
3.1.4	Comparison Models	32
3.1.5	Background Model Parameters	32
3.1.6	Evaluation Frameworks.....	34
3.1.6.1	ChangeDetection.net	35
3.1.6.2	PETS Metrics.....	35
3.1.6.3	Advantages of Evaluation Frameworks.....	35
3.1.6.4	Limitations of Evaluation Frameworks.....	36
3.1.6.5	Current State of Background Model Evaluation Frameworks	37
3.1.7	Current State of Background Model Evaluation	37
Chapter 4	Proposed Evaluation Framework.....	40
4.1	Creating a Model.....	41
4.2	Submitting a Model.....	42

4.3	Running a Submitted Model	43
4.4	Performance Evaluation.....	43
4.5	Publishing Evaluation Results	45
4.6	Proposed Framework Website	46
4.7	Database	47
4.8	Re-evaluation of Background Models.....	47
4.9	Alternative Framework	47
4.10	Overview of Proposed Evaluation Framework	48
Chapter 5 Proposed Evaluation Methodology		49
5.1	Evaluation Dataset	49
5.1.1	Video Categories	49
5.1.2	Dataset Size.....	51
5.1.3	Dataset Overview.....	51
5.2	Ground Truth	52
5.2.1	Ground Truth Type.....	52
5.2.2	Ground Truth Classifications.....	52
5.2.3	Ground Truth Volume	53
5.2.4	Ground Truth Accuracy	54
5.2.5	Ground Truth Overview	54
5.3	Evaluation Metrics	54
5.3.1	Accuracy Metrics.....	55
5.3.2	Efficiency Metrics.....	56
5.3.3	Training Phase Length.....	57
5.3.4	Lag.....	57
5.3.5	Evaluation Metric Overview.....	57
5.4	Parameter Value Determination.....	58
5.5	Post-Processing.....	58
5.6	Overview of Proposed Evaluation Methodology.....	58
Chapter 6 Implementation		59
6.1	Software Requirements	59
6.1.1	Extensibility	59
6.1.2	Robustness.....	59
6.1.3	Scalability	60
6.1.4	Usability.....	60

6.2	Background Model Evaluation	60
6.2.1	API Creation	60
6.2.2	Background Model Implementation and Adaptation	61
6.2.3	Automatic Background Model Execution	61
6.2.4	Result Evaluation.....	61
6.2.4.1	Accuracy Metrics.....	61
6.2.4.2	Average Frame Processing Time	62
6.2.4.3	Training Phase Length	63
6.2.4.4	Lag Measurement	63
6.3	Evaluation Framework Website	63
6.3.1	Web Server.....	64
6.3.2	Website Design	64
6.3.3	Web Pages.....	64
6.3.3.1	Home Page	65
6.3.3.2	Dataset Page	65
6.3.3.3	Model Submission Page	65
6.3.3.4	Results Page	68
6.3.3.5	Downloads Page.....	71
6.4	Database	72
6.4.1	Database Design.....	72
6.4.2	Database Communications	74
6.4.2.1	C++ Communications	74
6.4.2.2	PHP Communications.....	74
6.5	Implementation Overview	74
Chapter 7	Evaluation.....	76
7.1	Testing of the Proposed Evaluation Framework.....	76
7.1.1	Baseline	77
7.1.2	Dynamic Background	79
7.1.3	Camera Shake	81
7.1.4	Intermittent Object Motion	83
7.1.5	Shadow.....	85
7.1.6	Performance Overview	87
7.2	Advantages of the Proposed Evaluation Framework.....	89
7.2.1	Comprehensive Performance Evaluation	89

7.2.2	Reduction in Developer Workload.....	89
7.2.3	Easily Updated Evaluation Methodology.....	89
7.2.4	Guarantee of Fair and Consistent Results.....	90
7.2.5	Extensive Resource of Background Model Performance Data	90
7.3	Limitations of the Proposed Evaluation Framework	90
7.3.1	Modification of Existing Implementations.....	90
7.3.2	Patented Algorithms	91
7.3.3	Security Concerns.....	91
7.4	Project Successes	91
7.5	Evaluation of Software Characteristics	92
7.5.1	Extensibility	92
7.5.2	Robustness	92
7.5.3	Scalability	93
7.5.4	Usability.....	93
7.6	Overview	93
Chapter 8	Conclusion.....	94
8.1	Future Work.....	94
8.2	Reflection on the Project	95

List of Figures

Figure 2.1.1 - Background subtraction	3
Figure 3.1.1 - Sample frames from the (a) & (b) Wallflower [15] and (c) & (d) PETS 2001 [3] datasets	15
Figure 3.1.2 - Number of videos in evaluation datasets	15
Figure 3.1.3 - Sample frames from synthetic videos used in background model evaluations by (a) & (b) Brutzer et al. [19] and (c) & (d) Vacavant et al. [11].....	16
Figure 3.1.4 - (a) Original frame, (b) Corresponding binary ground truth frame. Foreground pixels are coloured white, background pixels are coloured black (source: [47]).....	20
Figure 3.1.5 - (a) Original frame, (b) Corresponding ground truth frame with a variety of labels. Foreground pixels are coloured white, background pixels are black, unknown pixels are light grey and shadow pixels are dark grey (source: [14])	20
Figure 3.1.6 – (a) Original frame, (b) Corresponding ground truth frame containing each of the five pixel labels. The non-ROI label is used so that the trees waving in the background are ignored and the shadow alone is considered. (source: [14]).....	21
Figure 3.1.7 - (a) Original frame, (b) Corresponding ground truth frame containing labels defined by Brostow et al. [48] (Source: [49])	22
Figure 3.1.8 - Bounding box annotated video frame from the CAVIAR dataset (Source: [1]).....	22
Figure 3.1.9 - (a) Original image, (b) Segmentations of original image, each produced by a different person (Source: [54])	24
Figure 3.1.10 - (a) Original image, (b) Consensus of original image segmentations shown in Figure 3.1.9 (Source: [54]).....	25
Figure 3.1.11 - Aspects of background model performance considered in evaluation	31
Figure 3.1.12 - Evaluation Framework	40
Figure 4.3.1 - Detail of background model evaluation block of Figure 3.1.12	43
Figure 4.4.1- Evaluating background model performance for a single video	45
Figure 4.6.1 - Website structure.....	46
Figure 4.9.1 - Alternative evaluation framework	48
Figure 6.3.1 - Evaluation framework website menu	64
Figure 6.3.2 - Submission of background model details	65
Figure 6.3.3 - Submission of developer details.....	66
Figure 6.3.4 - Submission of background model code files	66
Figure 6.3.5 - Code files selected for upload	67
Figure 6.3.6 - Field required error message	67
Figure 6.3.7 - Invalid email address error message	67
Figure 6.3.8 - Results page feature overview.....	68
Figure 6.3.9 - Challenge category selection tabs	69
Figure 6.3.10 - Sample video frames and corresponding ground truth frames from videos in the dynamic background category.....	69
Figure 6.3.11 - Zoomed sample frame on mouse rollover	69
Figure 6.3.12 - Shadow status selection.....	69
Figure 6.3.13 - Background model evaluation accuracy results for the camera shake category with shadow considered part of the background	70

<i>Figure 6.3.14 - Background model evaluation results for the camera shake category with shadow considered part of the background</i>	70
<i>Figure 6.3.15 - Displayed KDE background model and developer contact details</i>	71
<i>Figure 6.3.16 - Accuracy evaluation results for the KDE background model when face with a number of challenges</i>	71
<i>Figure 6.4.1 - Database entity relationship diagram</i>	72
<i>Figure 6.4.2 - Mapping to relational schema</i>	73
<i>Figure 7.1.1 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the baseline category</i>	78
<i>Figure 7.1.2 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the dynamic background category</i>	80
<i>Figure 7.1.3 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the camera shake category</i>	82
<i>Figure 7.1.4 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the intermittent object motion category</i>	85
<i>Figure 7.1.5 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the shadow category</i>	86

List of Tables

<i>Table 3.1-1 - Scenarios in which background models have been evaluated in previously conducted evaluations</i>	17
<i>Table 3.1-2 - Basic binary classifications</i>	26
<i>Table 4.1-1 - API calls needed to process the dataset of the evaluation framework</i>	41
<i>Table 4.2-1 - Information requested by background model submission form</i>	42
<i>Table 5.1-1 - Challenges to be depicted in standard evaluation dataset</i>	51
<i>Table 5.2-1 - Proposed ground truth labels</i>	53
<i>Table 6.2-1 - Updating true positive, true negative, false positive and false negative counts for a background model and video combination. In this table: TP = true positive, TN = true negative, FP = false positive, FN = false negative, fg = foreground, bg = background, s = shadow, GT = ground truth. The ++ symbol indicates which of the TP, TN, FP or FN counts are to be updated based on the ground truth pixel and result pixel combination.</i>	62
<i>Table 6.4-1 - Overview of database tables and contents</i>	74
<i>Table 7.1-1 - Background model evaluation results for the baseline category</i>	77
<i>Table 7.1-2 - Background model evaluation results for the dynamic background category</i>	79
<i>Table 7.1-3 - Background model evaluation results for the camera shake category</i>	81
<i>Table 7.1-4 - Background model evaluation results for the intermittent object motion category</i>	83
<i>Table 7.1-5 - Background model evaluation results for the shadow category</i>	86
<i>Table 7.1-6 - Background model evaluation results across all categories</i>	87

Abbreviations

CAVIAR	Context Aware Vision using Image-based Active Recognition
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GUI	Graphical User Interface
KDE	Kernel Density Estimation
NPV	Negative Predictive Value
PDF	Probability Density Function
PETS	Performance Evaluation of Tracking and Surveillance
PR Curve	Precision-Recall Curve
PSO	Particle Swarm Optimisation
PWC	Percentage of Wrong Classifications
RDBMS	Relational Database Management System
ROC Curve	Receiver Operator Characteristic Curve
TN	True Negative
TP	True Positive
XML	Extensible Markup Language

Abstract

Background subtraction is a fundamental task in numerous computer vision applications. The primary aspect of this, the modelling and maintenance of a background image for a video sequence, is complicated by a large range of diverse challenges. An abundance of background modelling techniques have been developed but none of these is capable of competently dealing with all of the challenges with which they may be faced. To gain a full understanding of the capabilities of these background modelling techniques it is essential that their performance be comprehensively evaluated in many scenarios. To date, no extensive or comprehensive background model evaluations have been carried out, nor do the facilities for doing so exist. The efforts that have so far been made are inadequate and exhibit many weaknesses and limitations. This project examines these existing efforts and identifies both their deficiencies and the work that is necessary to address these deficiencies. In addition, a solution to the problem of background model evaluation in the form of a standard framework and methodology is proposed. This proposal aims to facilitate the thorough and objective assessment of background model performance in a manner that will allow for an extensive reference of performance data to easily be compiled. Without the adoption of such a solution, the comprehensive and extensive evaluation of background models cannot become a reality. A proof of concept version of the proposed system was developed and, through testing, was found to be capable of such evaluation and thus, a feasible solution to the problem of effective background model evaluation.

Chapter 1 Introduction

1.1 Motivation

Background subtraction is a computer vision technique which aims to extract the foreground of a video stream for further processing. The technique is widely used and is a fundamental task in numerous applications including the detection, tracking and classification of objects, visual surveillance (e.g. crowd monitoring, people counting and action recognition), the detection of security applications (e.g. abandoned luggage, theft and loitering), video annotation and video forensics. A critical aspect of background subtraction is the modelling and maintenance of a background image for a video. The large volume of background modelling algorithms that have so far been developed demonstrates the importance and widespread use of background subtraction.

Despite the widespread availability of background modelling algorithms, no single algorithm exists that is capable of competently handling all of the challenges, e.g. background motion, shadows and illumination changes, with which it will inevitably be faced. To fully understand and appreciate background model capabilities it is essential that the performance of all models is thoroughly evaluated. This is vital in assessing the progress that is being made in background model development and in analysing the weaknesses of the existing body of background modelling algorithms to further development. It is also of importance in aiding in the selection of an appropriate background model for a particular application.

Due to the enormous effort that is required, no large-scale, comprehensive background model performance evaluation has so far been carried out. This is a major shortcoming of background subtraction research and, until extensive and rigorous evaluations can be performed, research in this area will continue to be deficient. This project examines the efforts that have been made in background model evaluation as reported in the relevant literature to date and identifies the weaknesses and the challenges that currently exist. In addition, the manner in which background model evaluations should be carried out is considered and a proposal as to how this may be achieved is presented.

1.2 Project Objectives

This project had a number of major goals. The first of these goals was to carry out a thorough analysis of the state of background model evaluation as reported in literature to date in order to determine the strengths and the limitations of the evaluations which have previously been reported, to identify the main issues that currently exist in background model evaluation and to ascertain what work needs to be done to make the extensive, comprehensive and objective evaluation of background models a reality.

Having determined the current weaknesses in background model evaluation it was aimed that a standard evaluation framework and methodology that would address these current limitations would be created. It was intended that this proposed framework and methodology would facilitate the automatic, rigorous, fair and credible evaluation of background modelling algorithms in a manner that would allow models to be easily compared and ranked against one another.

1.3 Report Structure

Chapter 1 introduces the project and discusses the importance of background subtraction and background modelling and the need for their evaluation. The limitations which currently exist in these evaluations are alluded to and the main objectives of the project are described.

In **chapter 2**, the process of background subtraction and the major challenges that are encountered in performing it are discussed. In addition, the characteristics and operation of some common background modelling techniques are described.

Chapter 3 considers the performance evaluation of background models including the general process and components of evaluation and the strengths and limitations of those background model evaluations which have previously been reported in literature. The main issues which currently exist in background model evaluation are identified and discussed.

Chapter 4 presents a proposal for a background model evaluation framework which addresses many of the issues, identified in chapter 3, which exist with the way in which background model evaluations have previously been carried out. This framework provides many advantages and allows for the capabilities of background models to be fairly and objectively assessed in a manner which allows their performance to be meaningfully compared.

In **chapter 5**, a standard background model evaluation methodology including the video dataset, ground truth and evaluation metrics that should be used, is proposed. This evaluation methodology complements the framework proposed in chapter 4. It addresses many of the issues identified in chapter 3 and allows for background models to be comprehensively, fairly and credibly assessed and to be meaningfully ranked and compared against one another.

Chapter 6 looks at how the proposed evaluation framework and methodology presented in chapters 4 and 5 were implemented and the technologies that were used. The difficulties that were encountered throughout the implementation are discussed as are the ways in which these difficulties were addressed. Also considered are the main software characteristics that the system should exhibit.

In **chapter 7**, the feasibility and the functionality of the proposed evaluation framework are analysed. In doing this, the implemented system is tested and the background model evaluation results that are obtained are analysed. In addition, a review of the advantages and the limitations of the proposed framework as well as of the successes of the project is presented. The framework is also considered in terms of how well it achieves the desired software characteristics that were discussed in chapter 6.

Chapter 8 discusses the conclusions that were reached throughout the completion of this project. The work that was completed is reviewed and assessed and areas of future work are identified. In addition, a reflection on the experience of completing this project is included.

Chapter 2 Background Subtraction

As was mentioned in chapter 1, background subtraction, sometimes referred to as foreground detection, is a computer vision technique which is used to extract the foreground of an image so that it may be used in further processing. Background subtraction is typically performed on images that form part of a video sequence. The separation of foreground and background is a critical part of many computer vision applications including object tracking, intrusion detection, crowd monitoring and people counting. Such applications “rely heavily on the accuracy of foreground object detection” [1] and it is thus important that algorithms exist which allow for highly accurate foreground detection to be carried out. There are, unfortunately, many challenges which severely complicate the process of background subtraction. Discussed in this chapter are some of the major challenges that are encountered in background subtraction as well as the general process by which it is completed and some common techniques of background modelling, a critical aspect of background subtraction.

2.1 Basic Background Subtraction Algorithm

Background subtraction, or the extraction of foreground from the frames of a video stream, is carried out by subtracting the background of the scene depicted in the stream from the video frames so that the foreground pixels can be identified. This is illustrated in Figure 2.1.1. Having extracted the foreground from a video sequence it may then be used in further processing.

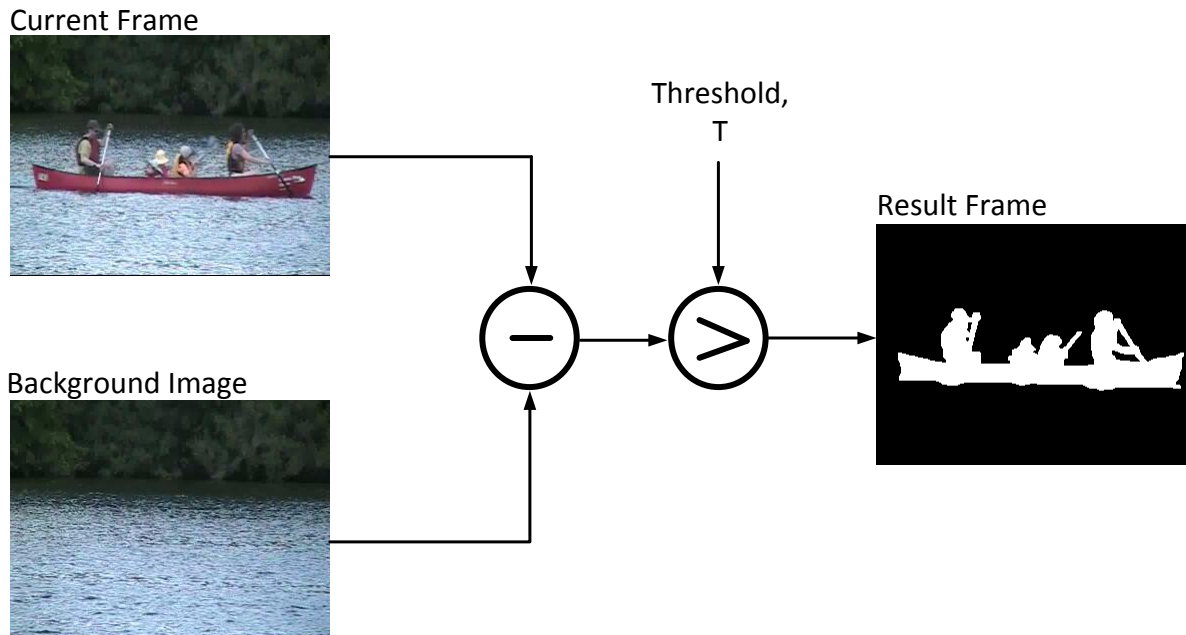


Figure 2.1.1 - Background subtraction

The creation of a background image for a video stream is discussed in section 2.2. Once a background image is obtained, it may simply be subtracted from each video frame to obtain a difference image. In this image, the pixels whose value is the same as the corresponding pixel in the background image will be black while those whose value is different, i.e. the foreground of the

frame, will have a non-black colour. Assuming that both the video frame and the background image are greyscale images, the resulting difference image will also be a greyscale image. In this case, the pixel values of the difference image $d(i, j)$ are simply the absolute value of the difference between the pixel values of the video frame $f_1(i, j)$ and those of the background image $f_2(i, j)$. This is illustrated by the equation below.

$$d(i, j) = |f_1(i, j) - f_2(i, j)|$$

Alternatively, by making use of a threshold, T , as shown in Figure 2.1.1, a binary difference image may instead be obtained. In this case, the video frames and the background image are still assumed to be greyscale images and, if the absolute difference between their pixel values is less than the chosen threshold the corresponding difference image pixel is considered to be background and is typically given a value of 0 (black). Otherwise, the difference image pixel is considered to be foreground and is typically given a value of 255 (white). This is illustrated by the equation below and a sample binary difference image may be seen on the right-hand side of Figure 2.1.1.

$$d(i, j) = \begin{cases} 0 & \text{if } |f_1(i, j) - f_2(i, j)| < \epsilon \\ 1 & \text{otherwise} \end{cases}$$

It is also possible that the background image and the video frames are colour images. This introduces complications in determining how the difference image should be created. It is possible, for example, to just process selected channels. Alternatively, each channel may be processed separately and the resulting difference image composed of the greatest difference that is calculated. Of the three possible difference image types that have been mentioned (greyscale, binary and colour), binary difference images are by far the most common.

In a binary difference image, black pixels typically correspond to background while white pixels correspond to foreground. Ideally, the classifications of background and foreground pixels would be completely accurate with the difference images consisting solely of true positives (foreground pixels correctly identified as foreground pixels) and true negatives (background pixels correctly identified as background pixels). In practice, however, a number of false positives (background pixels incorrectly identified as foreground pixels) and false negatives (foreground pixels incorrectly identified as background pixels) will also be present.

The number of misclassified pixels that are present in a difference image should be minimised but, unfortunately, it is not typically possible to completely avoid them. One of the major reasons for this is that background subtraction is dependent upon the background and the foreground of the video frames being distinct or highly contrasting which can generally not be guaranteed. Additionally, it is important that the value of the threshold be set appropriately. If the threshold value is too low then the number of false positives is likely to be significant while, if it is too high, there is likely to be a large number of false negatives.

2.2 Background Modelling

The modelling and maintenance of a background image for a video stream is a critical aspect of background subtraction. Without the ability to determine the background of a video it is not possible to extract the foreground from it. A large number of techniques for modelling background have been developed. Some of these techniques are described in section 2.2.2. While some

techniques are very straightforward in operation, the accurate modelling of a scene background is not a straightforward task. There are a large number of challenges that exist which can complicate background modelling. Some of the main challenges that are encountered are described in section 2.2.1 while the process of evaluating how well the various background modelling techniques handle these challenges is discussed in future chapters.

2.2.1 Challenges

This section considers some of the major challenges encountered by background modelling techniques in attempting to build and maintain an accurate image of a scene background.

2.2.1.1 Object of Interest

As previously discussed, the aim of background subtraction is to extract foreground from a video sequence by subtracting the scene background from the video frames. To create a background image a background model must be able to identify which parts of the scene belong in the background and which belong in the foreground. This, unfortunately, is not always clear. In general, moving object pixels are regarded as being of interest or in the foreground of a scene. These often comprise objects such as people and vehicles. There is often, however, other movement in the scene which may not be of interest. An example of this is background motion such as a tree moving in the wind (see section 2.2.1.5). Moving shadows, meanwhile, are typically considered to be background motion though in some applications they may be considered to form part of the scene foreground.

Confusion also arises when moving objects stop moving. It is unclear whether an object should immediately become part of the scene background once it has come to a stop, whether it should be integrated into the background after a period of time or whether it should ever become part of the background. If a moving person is the object of interest in a scene, for example, it is likely that, if they come to a stop, they will remain of interest. In addition, a car that is driving will be considered part of the scene foreground. A car that is parked, meanwhile, will be considered part of the background. This implies that a car that comes to a stop in a scene should become part of the background. If the car parks, this is a reasonable assumption but, if it stops at traffic lights and will soon move off again, it is reasonable to assume that it remains in the foreground of the scene.

There is no consensus or correct answer regarding how scenarios such as these should be handled. The answers to questions such as, which objects belong in the foreground of a scene, which belong in the background and when foreground objects should become part of the scene background, are unclear. Exactly how the pixels of video frames should be classified is unknown and thus the exact aims of background subtraction and background modelling are also unknown. This uncertainty in what is to be achieved presents a major difficulty in background modelling and, as will be seen in chapter 3, creates many challenges to the process of background model evaluation.

2.2.1.2 Updating the Background

To maintain a background image with a reasonable degree of accuracy it must be updated as the background of the scene changes. In addition to changes resulting from the scene illumination, weather etc. which will be described later, a background model may encounter such challenges as objects entering the scene and stopping to become part of the background, or background objects beginning to move, i.e. becoming foreground objects and leaving behind a "hole" in the background which will appear as foreground. Similarly, if a door in the scene were to open or close, the door

should remain part of the background, but the movement of the door and the area revealed by moving it, would cause a significant change in the appearance of the background making it incorrectly seem like foreground. It is important that a background image be updated to adapt to changes such as these so that they will not persist in the scene foreground.

2.2.1.3 Camouflage

As was mentioned in section 2.1, it is important in background subtraction that there be a high contrast between the background and the foreground of a scene in order for them to be easily distinguished. High contrast is typically a result of different colours or textures being present in the foreground and background. It is not generally possible to guarantee such high contrast, however. If contrast is low, foreground objects will become camouflaged against the background making it difficult to distinguish the two. This may occur for reasons such as foreground objects being of a similar colour to the background and poor illumination and can result in foreground objects not being detected or many false negative classifications.

2.2.1.4 Illumination Changes

Another challenge that is commonly encountered in background modelling is illumination changes. Illumination changes may be gradual such as the natural variation of light throughout a day or they may be sudden, such as a light being switched off in a room. In the case of a gradual illumination change, a model is required to deal with minor changes in the appearance of the background over a prolonged period of time. In the case of sudden illumination changes, meanwhile, the appearance of the background will change dramatically making much of the frame appear to be foreground. This will result in a large number of false positive classifications. After such a change, it takes some time for a background model to adapt to the new appearance of the background. In addition, as frames darken due to illumination changes, the contrast between the background and the foreground of the scene being examined will decrease. This makes them more difficult to distinguish creating further challenges for the background model. Illumination changes may be encountered in both indoor and outdoor scenes.

2.2.1.5 Dynamic Background

A significant difficulty involved with building and maintaining a background model is the presence of uninteresting background motion or a dynamic background. Such motion can be periodic or irregular and may be a result of trees moving in the wind, flowing water, a waving flag etc. While this motion is valid it is typically not considered to be foreground and should therefore be ignored. Background dynamism may be present in all or part of a frame and is commonly encountered in outdoor videos.

2.2.1.6 Weather Conditions

The weather conditions in a scene can also create considerable and diverse difficulties in background modelling. It was seen in section 2.2.1.5, for example, that wind can contribute to the presence of a dynamic background. In addition, it may cause a camera to shake making for an unsteady video stream. Rain, meanwhile, tends to darken the scene, create noise and can alter large portions of the scene, e.g. if concrete becomes wet its colour will change. Similarly, the presence of snow in a scene will tend to introduce additional false positives as it is mistaken for foreground and will also brighten the scene. Additionally, fog will reduce the contrast of the background and foreground in a scene making them more difficult to distinguish. If the sun becomes occluded by clouds, meanwhile, the

scene will become darker, i.e. the scene illumination will change. These, amongst other weather conditions, have a significant impact on the appearance of a scene background and make it more difficult for a model to maintain a reasonably accurate background image. If a background model is being used in the processing of an outdoor scene, it will inevitably encounter some challenges as a result of the weather.

2.2.1.7 Shadow

The presence of shadows in a scene can be a serious problem for background modelling techniques. Shadows may be cast by both static and moving objects and can greatly affect the appearance of a scene. It was mentioned in section 2.2.1.1, that the importance of shadow and the way in which it should be classified may vary between applications but it is often the case that shadows should be ignored and regarded as part of the background. This is a fairly straightforward task when considering shadows cast by static objects though complications can arise when these shadows move throughout the day as a result of the changing position of the sun in the sky. This movement, however, tends to be gradual so that a background model will be presented with small changes in the scene background over time. Shadows cast by moving objects, meanwhile, present significantly more challenging issues. To a background model, such shadows will appear to be moving objects and may thus be considered foreground. It can be very difficult to distinguish between a moving object and its shadow and, if shadow is to be considered as background, this can cause problems in the further processing of the detected foreground of a scene. For example, a shadow detected as foreground may alter the shape of a true foreground object complicating its detection or, in an application concerned with people counting, a person's shadow may erroneously appear to be a second person (a similar phenomenon may be seen as a result of reflection). The existence of shadows may also reduce contrast in a scene further complicating the creation and maintenance of acceptable background images.

2.2.1.8 Bootstrapping

As will be further discussed in section 2.2.2, some background models require a training period in which they use video frames to build up an initial background image. Ideally, these frames should be free of foreground objects but it sometimes happens that no such frames are available. The training of the model or the creation of the initial background image is therefore polluted by the presence of foreground objects.

2.2.1.9 Camera Characteristics

Challenges may also arise as a result of the characteristics of the camera that is used to record a scene. One such challenge concerns whether the camera is static or in motion. If a camera is static it is always viewing the same area. Parts of the scene may change over time but the position of the camera does not. If the camera is in motion, however, e.g. if it can pan a scene, or if the area within its field of video is otherwise unfixed, e.g. due to camera zoom, the maintenance of a background model becomes significantly more challenging as the pixel values in the video stream will be constantly changing and may thus result in many false positive detections.

While camera motion may be deliberate, such motion is smooth and expected. Additional motion may be introduced, however, as a result of camera unsteadiness. A common cause of this is the shaking of the camera due to wind. An unsteady camera can create significant complications in the modelling of a scene background.

The quality of the video stream is another challenge that background modelling techniques are frequently faced with. It cannot be guaranteed that the frames of the video stream will always be of a high quality. Instead, it is common that background models will be expected to work with poor quality videos the frames of which are of a low resolution, noisy etc. making them more difficult to process.

2.2.2 Background Modelling Techniques

The aim of a background model is to create an accurate depiction of the background of a scene. This is far from a straightforward task with many challenges such as those described in section 2.2.1 complicating the process. The simplest form of background modelling that exists is to use a static background image or an image of the scene in question with no foreground objects present. This technique is quite commonly used but is not robust against changes to the appearance of the scene background. The majority of techniques, however, attempt to update their background image to allow for background changes.

Some background models require a training phase during which frames from the video stream are used to create a background image before background subtraction begins. The length of this phase varies between models. Ideally, the frames that are used would contain no foreground objects so that a background model may be acquired without the complication of foreground pollution. As was mentioned in section 2.2.1.8, however, it often happens that such frames are not available (bootstrapping) meaning that a background model must be created using data which is not solely background data.

Many background models process videos one frame at a time, i.e. they take a frame and produce the result for that frame before any more frames are taken from the video. In these cases the models have no lag and processing is performed online. Other background models, however, exhibit varying amounts of lag, i.e. multiple frames are taken from a video stream before a result frame is returned. For example, if a model takes six frames from a video before returning the result for the first of these frames, the lag of the model is said to be five frames. By using a lag in the process of creating and maintaining a background image, future information can be used in the processing of each frame allowing for a more accurate image to be obtained. If a model takes all frames of a video before returning results for any of these frames, it is said to work offline. The degree of lag that is used dictates the applications in which a background model may be used. If it is necessary that results be obtained in or close to real-time, it is essential that a model with just a small amount or no lag is selected (the exact requirements of an application will determine how much lag is acceptable). In such applications, the use of models which work offline is not acceptable. In some applications such as video forensics, however, it is acceptable for background models to work offline as results are not required right away. Models with any degree of lag are suitable for use such applications. The amount of lag used in maintaining a background image is thus a critical factor in the selection of an appropriate model for a specific application.

Described below is a selection of background modelling techniques. The performance of these techniques is discussed in chapter 7.

2.2.2.1 Static Frame Difference

This background modelling method makes use of a static background image which, as mentioned previously, is the simplest form of modelling possible. The first frame of the video stream is used as the background image and it is therefore important that this frame does not contain any foreground objects. This, however, cannot always be guaranteed. This technique is very straightforward to implement but is not robust to the challenges presented in section 2.2.1 as the background is never updated.

2.2.2.2 Frame Difference

The frame difference background model is also a very straightforward technique. Initially, the first frame of the video stream is used as the background image and, as subsequent frames are processed, the background image is updated to these frames. Thus, to determine the foreground of a frame, the difference between that frame and the previous frame is obtained. This technique allows the background image to be updated to provide some robustness against the challenges described in section 2.2.1. It can be difficult, however, to analyse the result frames of this technique.

2.2.2.3 Weighted Moving Mean

This model is a variation on the running average model which strives to incorporate changes into the scene background. In running average background modelling the background image is calculated as the average of the last m frames of the video stream with each frame having the same influence on the background image. In this weighted variation, the background image is again formed as the average of the last m frames but the pixel values are weighted so that more recent frames will contribute more to the background image. This is illustrated by the equation below where $B_{n+1}(i, j)$ is the value of pixel (i, j) in the updated background image, $f_*(i, j)$ is the value of pixel (i, j) in the foreground images and α , β and γ are the frame weightings.

$$B_{n+1}(i, j) = \frac{\alpha f_n(i, j) + \beta f_{n-1}(i, j) + \dots + \gamma f_{n-m}(i, j)}{m}$$

This technique allows the background image to be updated to incorporate changes in the scene background more quickly than the running average method. It also, however, requires the inefficient storage of the m previous frames so that the oldest can be removed when another is to be added.

2.2.2.4 Adaptive Background Learning

This technique is essentially an approximation to a running average background model. To avoid the inefficiency of storing previous frames, the adaptive background learning technique estimates a running average background image. To update the background image, this model uses the corresponding pixel values from the current background and foreground images. This is illustrated by the equation below where $B_{n+1}(i, j)$ is the value of pixel (i, j) in the updated background image, $f_n(i, j)$ is the value of pixel (i, j) in the current frame, $B_n(i, j)$ is the value of pixel (i, j) in the current background image and α is the learning rate or the rate at which the background image adapts to changes in the scene.

$$B_{n+1}(i, j) = \alpha f_n(i, j) + (1 - \alpha)B_n(i, j)$$

This model allows the background image to be updated to account for changes to the background of a scene such as illumination changes and objects coming to a stop. The value of the learning rate α

will determine how quickly these changes are incorporated into the background. If the learning rate is too large the background image will be updated quickly and moving objects which form the foreground of the scene will be integrated into the background. A low learning rate will reduce this but will also result in the slow adaptation of the background image to valid background changes.

It is possible to overcome the trade-off of slow background updates and foreground being integrated into the background image by only updating the pixels of the background image which correspond to background pixels in the current frame. This is known as selectivity and is a reasonable approach if it can be guaranteed that foreground objects will not stop moving in the scene. If a foreground object does come to a stop it will never be integrated into the background image.

2.2.2.5 Gaussian Mixture Model

The Gaussian mixture model was proposed by Stauffer and Grimson [2] in an attempt to effectively deal with scenes containing uninteresting background motion such as a tree moving in the wind or ripples in water. A number of variations on this model also exist. Each pixel in a scene is modelled using a number of Gaussian distributions based on its previous values. Each pixel is typically represented by three or four distributions which are weighted based on how frequently they have previously occurred. The weighting for the Gaussian distribution m for pixel (i, j) in frame n , is given as $\pi_n(i, j, m)$. These Gaussian distributions for each pixel form the background model of the scene.

For each frame n , every pixel $f_n(i, j)$ is compared to the distributions which model the corresponding pixel in the background model to check if its value is close (based on a threshold) to any of them. If the pixel value is not close to any of the existing distributions a new one is created for the pixel. There is a limit on the number of distributions that can be defined for a pixel and, once this limit is reached, the smallest distribution must be discarded in favour of new ones.

The distributions for each pixel are updated using the new pixel value with the equations below. In these equations, $\pi_n(i, j, m)$ is the weight assigned to distribution m for that pixel, $\mu_n(i, j, m)$ is its average value, $\sigma_n^2(i, j, m)$ is its standard deviation and α is a learning rate.

$$\pi_{n+1}(i, j, m) = \alpha O_n(i, j, m) + (1 - \alpha)\pi_n(i, j, m)$$

$$\mu_{n+1}(i, j, m) = \mu_n(i, j, m) + O_n(i, j, m) \frac{\alpha}{\pi_{n+1}(i, j, m)} (f_n(i, j) - \mu_n(i, j, m))$$

$$\sigma_{n+1}^2(i, j, m) = \sigma_n^2(i, j, m) + O_n(i, j, m) \frac{\alpha}{\pi_{n+1}(i, j, m)} \left((f_n(i, j) - \mu_n(i, j, m))^2 - \sigma_n^2(i, j, m) \right)$$

If a new distribution was created $O_n(i, j, m)$ will be given a value of 1 while if a distribution close to the new pixel value was found, it will be given a value of 0. The weighting of the distribution that is close to a pixel value is used to determine whether that pixel constitutes foreground or background. If the weighting is below a certain threshold it is considered foreground, otherwise it is considered to be in the background.

2.2.2.6 KDE

The KDE (Kernel Density Estimation) background model which was proposed by Elgammal et al. [3], is a non-parametric model. It uses sample pixel intensity values to estimate the probability of a pixel having certain intensity values and aims to “capture very recent information about the image sequence, continuously updating this information to capture fast changes in the scene background”

[3]. Taking x_1, x_2, \dots, x_N to be a recent history sample of intensity values for a pixel, the probability density function (PDF) of that pixel's intensity value x_t at time t may be estimated using the equation below where K_σ is the kernel estimator with a bandwidth of σ .

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^N K_\sigma(x_t - x_i)$$

By assuming K to be a normal distribution and generalising the estimate to use colour features, the PDF estimation may be written as below where x_t is a colour feature with d dimensions.

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2} \frac{(x_{tj} - x_{ij})^2}{\sigma_j^2}}$$

Using this PDF estimate, a pixel is classified as a foreground pixel if $\Pr(x_t) < t$ where t is a global image threshold. Otherwise, it is considered to be a background pixel. It has now been seen how the KDE model identifies foreground pixels given a sample of the N recent intensity values for each pixel. These N values in each sample essentially form the background model of the scene and thus, to create the initial sample, a training phase of N frames is required.

The samples, or the model of the background, must be continuously updated in order for the model to adapt to changes in the scene. The number of values, N , in each sample will affect the success of these updates as will the update approach that is taken. Small values of N will allow the background model to adapt to changes more quickly but will also make it very sensitive to change while larger values of N will provide "a more stable representation of the scene background" [3] but will be slower in adapting to changes.

There are two approaches to updating the background model – selective update and blind update. When using the selective update method, pixel values from each new frame are added to the corresponding set of sample values only if the pixel is classified as a background pixel. When using the blind update method, however, the pixels values from each new frame are added to the appropriate samples regardless of their classification. In each case, the least recent sample value is removed to make space for each new value that is added. While the first of these, the selective update method, can enhance the detection of foreground objects it also results in pixels which are incorrectly identified as foreground being persistently classified in this way. The blind update method does not suffer from this issue but, does result in foreground objects being erroneously integrated into the background image.

2.3 Overview of Background Subtraction

This chapter has provided an overview of the purpose and process of background subtraction and has presented some existing background modelling techniques and some of the challenges that they face. The accurate modelling of a scene background is a complicated process and there exists no technique that is competent in all scenarios. The next chapter will consider the performance evaluation of background modelling techniques and the way in which this has been addressed in the relevant literature to date.

Chapter 3 Background Model Evaluation

In the previous chapter, the process of background subtraction, some of the challenges that are faced in background modelling and some common background modelling techniques were described. Given the large number of background models that are available, the selection of a model that is appropriate for a particular application can be a very difficult and overwhelming task. There is no single background model that performs well in every scenario. Rather, their strengths lie in different areas and a model that performs very well in one set of circumstances may perform extremely poorly in another. Without a good appreciation of a model's capabilities it is impossible to make an informed decision as to which model is most suited to a certain application. To gain such an appreciation, and knowledgeably make these decisions, it is necessary to carry out a thorough evaluation of the capabilities of each background model. By evaluating all models in the same manner, their performance in a range of circumstances can easily be compared. The evaluation of background models in this way would be of great benefit to the research community, to those attempting to select an appropriate model for a task and to developers wishing to assess the performance of their own models. Unfortunately, however, as is discussed throughout the remainder of this chapter, there currently exists no comprehensive evaluation process meaning that there is no way to fully assess background model performance. As a result, a model may be selected simply for being well known regardless of whether it is suited for its intended purpose. The majority of the background evaluations that do exist have been carried out by developers attempting to test their models. These evaluations do not effectively assess model capabilities nor are they carried out in a consistent manner meaning that the outcomes of one developer's evaluations cannot be meaningfully compared to the outcomes of another. Some efforts have been made, namely by Goyette et al. [4] and by Young et al. [5], to address this, by creating systems to evaluate all models in the same manner and taking much evaluation work away from developers. While these do address the problem of incomparable evaluation results, they still suffer from a number of other serious limitations which are discussed later. In the remainder of this chapter, the typical process and components of a background model evaluation are described and the extent to which these have been considered in thirty-three existing evaluations is discussed. The major findings of this analysis are summarised in section 3.1.7. In chapters 4 and 5 a proposal as to how background model evaluations should be performed is presented as is a proposed framework for carrying out evaluations in a fair and consistent manner.

3.1 Evaluation Methodology

In order to evaluate the performance of a background model it must first be executed on a video dataset so that result background/foreground segmentation frames may be obtained for each video. A variety of evaluation metrics are used to assess how well the model has processed the dataset. The most common type of evaluation metrics are those which measure how accurately foreground has been detected but others such as those which consider a model's efficiency also exist. The accuracy of a background model is typically measured by comparing result frames to an actual correct segmentation or corresponding ground truth frame. The various components of a background model evaluation – video dataset, ground truth and evaluation metrics – are discussed in this section.

3.1.1 Video Datasets

To evaluate the performance of a background model, a substantial and varied video dataset which presents a range of challenges is essential. Without such a resource it is not possible to carry out an effective evaluation. A suitable dataset should depict a wide range of challenges including indoor and outdoor scenes, varying weather and lighting conditions and different types of motion. This diversity is important in determining the capabilities and limitations of a model and is therefore essential in permitting informed decisions as to the suitability of a model for a task to be made. In addition, the videos that are included in the dataset should vary in terms of length, resolution and quality amongst other video characteristics so that the ability of models in addressing such variations can be assessed. If a dataset is inadequate, a thorough model assessment is not possible.

3.1.1.1 Existing Datasets

A large number of video datasets, created for use in computer vision research, currently exist including the PETS [6] and CAVIAR [7] datasets. The characteristics of these as well as of a number of other publically available datasets are described below.

- **PETS [6]** – The PETS (Performance Evaluation of Tracking and Surveillance) workshop began in 2000 and a number of datasets which depict a range of scenarios have been developed for it. These datasets are tailored for different challenges such as crowd surveillance (e.g. people counting, density estimation and the tracking of an individual in a crowd of people) [8] and the detection of security events (e.g. loitering, abandoned luggage and theft) [9]. The videos are recorded with different cameras and are of a variety of resolutions and frame rates, which contributes to the diversity of the datasets. Each of the datasets is accompanied by training data.
- **CAVIAR [10]** – The CAVIAR (Context Aware Vision using Image-based Active Recognition) dataset is divided into two subsets. The first of these was filmed at the INRIA Labs in France while the second was filmed in a shopping centre in Portugal. The videos in both sets have a resolution of 384 x 288 pixels, a frame rate of 25 frames per second and are compressed with MPEG2. The INRIA set of twenty-eight videos depicts six scenarios – walking; browsing; resting, slumping or fainting; leaving bags unattended; two people fighting; and people or groups meeting, walking together and splitting up. Each of these is depicted by between three and six videos. The Portugal set depicts twenty-six scenarios each filmed from two different viewpoints. Scenarios include individuals and groups walking, stopping and entering and emerging from shops. The videos of this set are longer than those of the INRIA set with an average of 1500 frames.
- **BMC [11] [12]** – The BMC (Background Models Challenge) dataset was compiled for the 2012 Background Models Challenge. It comprises ten computer generated learning videos and nine real evaluation videos. The learning videos, generated using SiVIC [13], depict several weather conditions (clouds, sun, fog and wind) with varying amounts of noise. Each of these videos is one minute long with the first ten seconds being free of moving objects. The evaluation videos meanwhile, are between one and four hours in length and depict multiple challenges including dynamic backgrounds, shadows and changes in illumination.

- **2012 ChangeDetection.net [14]** – The 2012 ChangeDetection.net dataset was created for use in the 2012 IEEE Change Detection Workshop [15]. It comprises thirty-one videos which depict challenges in real indoor and outdoor scenes across six different categories – dynamic background, camera jitter, intermittent object motion, shadow, baseline and thermal. Each of these categories contains between four and six videos. The videos were recorded using a variety of camera types and vary between approximately 1000 and 8000 frames in length. Frame resolution ranges from 320 x 240 to 720 x 480 and the degree of both noise and compression artefacts varies between videos.
- **2014 ChangeDetection.net [16]** – The 2014 ChangeDetection.net dataset has been developed for use in the 2014 IEEE Change Detection Workshop [17]. It contains fifty-three videos depicting indoor and outdoor challenges across eleven different categories. Six of these categories are the same as those in the 2012 ChangeDetection.net dataset [14] and contain the same videos. The additional five categories are bad weather, low frame rate, night videos, air turbulence and videos recorded with a PTZ (pan-tilt-zoom) camera. Each category again contains between four and six videos. These videos were captured with various types of cameras and have resolutions ranging between 320 x 240 and 720 x 576. The amount of noise and compression artefacts that are present varies between videos.
- **Wallflower [18]** – The Wallflower dataset depicts seven different challenges typical of those encountered in background subtraction – moved object, gradual illumination change, sudden illumination change, dynamic background, camouflage, bootstrapping and foreground aperture. Each challenge is depicted by a single video. The frames of each video have a resolution of 160 x 120 pixels. Aside from the bootstrapping video, all begin with a training period of at least two hundred frames which are free of foreground objects.

Despite the availability of many video datasets such as those described, there exists none that is comprehensive and diverse enough to be considered sufficient for use in a thorough evaluation of background model capabilities. The existing datasets depict only a limited number of scenarios and tend not to exhibit significant variation in terms of length, quality etc. Due to this, no set of videos has been accepted as the standard on which to test background models. The numerous evaluations that have been conducted to date have instead used different combinations of existing, publically available datasets and their own original videos². In the background model evaluations carried out by Brutzer et al. [19] and Benezeth et al. [20], for example, some original videos are used while approximately 20% of the reviewed evaluations used at least part of the Wallflower dataset [18] (see Figure 3.1.1). The PETS 2001 dataset (see Figure 3.1.1) is also popular with just over 15% of the reviewed evaluations making use of its videos. A video from the Wallflower dataset [18] depicting a dynamic background was found to be the most commonly used video. This use of different datasets renders the results produced by the various evaluations incomparable. The remainder of this section considers the characteristics of the evaluation datasets that have previously been used including their size, diversity and the challenges which are depicted. The findings of this examination are used in the development of a proposal for a standard background model evaluation dataset which is presented in chapter 5.

² Some of the background model evaluations that were examined did not report details of the datasets that were used.



Figure 3.1.1 - Sample frames from the (a) & (b) Wallflower [15] and (c) & (d) PETS 2001 [3] datasets

3.1.1.2 Dataset Size

An important characteristic of a video dataset is its size. As is later described in detail, videos presenting a large range of challenges including illumination changes, background objects beginning to move etc. are necessary to perform a thorough background model evaluation. To depict all of the challenges that a background model may face, a significant number of videos are needed. With this in mind, the previous background model evaluations were examined in terms of the number of videos that were used³. The results of this examination are shown in Figure 3.1.2. Of the thirty-three evaluations that were reviewed, twenty-four analysed less than ten videos with several of these [5], [21], [22] using just a single video. Just four of the remaining studies [4], [23], [24], [25] used datasets containing more than thirty videos indicating that dataset size tends not to be seriously considered as an important aspect of evaluation. These observations clearly illustrate that the use of small datasets in background model evaluation is prevalent. This is a major shortcoming of existing evaluations as the use of a small dataset makes it impossible to adequately assess a model's capabilities and thus, diminishes the value of an evaluation

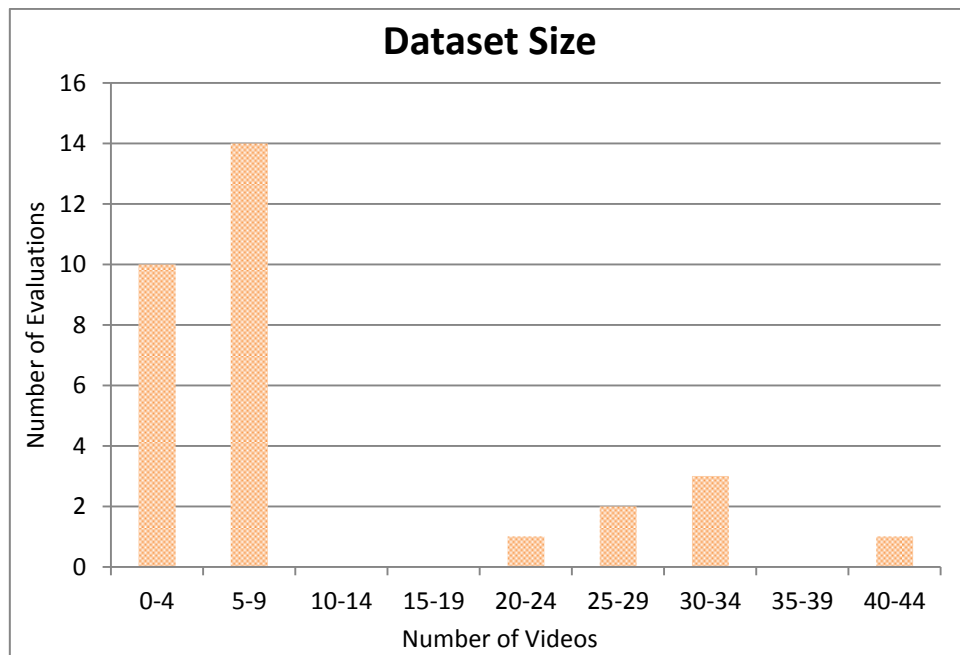


Figure 3.1.2 - Number of videos in evaluation datasets

³ Some of the background model evaluations that were examined did not report details of the datasets that were used.

3.1.1.3 Video Length

An additional dataset characteristic that should be considered is video length. The majority of the videos used in the reviewed evaluations were quite short ranging from less than a minute to approximately five or six minutes. These videos consist, at most, of just a few thousand frames. The only exceptions to this are the evaluation conducted by Brown et al. [26] in which some videos between twenty and thirty minutes in length were used and that reported by Vacavant et al. [11] which makes use of the BMC dataset [12]. As mentioned previously, the BMC dataset contains videos which are between one and four hours in length. While the use of shorter videos is acceptable it is not sufficient to solely use such videos, as has been found to typically be the case. Longer videos must also be included so that the reliability of background models as well as their ability to deal with challenges such as long-term gradual illumination changes can be examined. In order to properly depict and analyse gradual illumination changes, for example, a complete twenty-four hour video showing how illumination varies over an entire day would be needed. Without such videos, a dataset which is intended for use in the evaluation of background models will not be complete as all possible challenges cannot be considered.

3.1.1.4 Video Type

While the majority of the datasets that have been used in the reviewed evaluations comprise real videos, some studies also make use of synthetic and semi-synthetic videos. Brutzer et al. [19], for example, use solely synthetic videos created using “Mental Ray [27], a raytracer provided by Autodesk Maya” [19] (see Figure 3.1.3 (a), (b)). Vacavant et al., meanwhile, [11] make use of both real and synthetic video sequences, some frames of which may be seen in Figure 3.1.3 (c) and (d). In this case, the synthetic videos are created using the SiVIC simulator [13]. Finally, in the study conducted by Benezeth et al. [20], a combination of real, synthetic and semi-synthetic videos are used. The semi-synthetic videos consist of synthetic objects in the foreground moving over real background images.



Figure 3.1.3 - Sample frames from synthetic videos used in background model evaluations by (a) & (b) Brutzer et al. [19] and (c) & (d) Vacavant et al. [11]

The use of synthetic and semi-synthetic videos has great advantages in terms of ground truth creation. It will be seen in section 3.1.2.2 that the creation of ground truth for real videos is typically a very laborious process, the results of which are highly subjective due to differing opinions as to which pixels constitute foreground and which background. In the case of synthetic and semi-synthetic videos, however, this information is readily available and ground truth can thus be automatically generated very quickly and accurately. Despite this major advantage, the use of synthetic and semi-synthetic videos also has some significant drawbacks in that “the synthetic data will probably not faithfully represent the full range of real data” [28]. Even high quality synthetic videos are not truly representative of the real-life scenarios that background models will face in use. The results of evaluations using such data, therefore, are not a realistic portrayal of the capabilities

of the models tested. This trade-off between ease and quality of ground truth generation but unrealistic data and real data which is typical of what models will encounter in use but difficult generation of ground truth, is an important consideration when conducting a background model evaluation. Ideally, testing should be carried out on the same type of data that will be encountered in practice so that performance may effectively be assessed. Thus, evaluators should strive to always use real video data despite the effort involved in creating the associated ground truth. Were a comprehensive and standardised evaluation dataset in place, these issues would be eliminated for those who wish to evaluate their models as both videos and associated ground truth data would be freely available.

3.1.1.5 Challenges Considered in Evaluation

Also examined, was the range of scenarios or challenges that have been considered by the previously reported evaluations. It was mentioned previously that, in order to fully assess a background model, it must be tested in a wide and diverse range of scenarios typical of those that it may face in use. Without addressing such a range of challenges an evaluation cannot be considered complete or adequate. An overview of the challenges that have been considered in the examined background model evaluation studies is presented in Table 3.1-1 below.

Evaluation Scenarios	Studies
Simple moving object	[4] [5] [11] [18] [19] [20] [21] [22] [23] [24] [25] [29] [1] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [26] [42] [43] [44] [45] [46]
Basic Sequence	[19] [23] [4] [25]
Objects becoming still	[5] [22] [24] [40] [46]
People becoming still	[21] [30]
Part of the background begins to move	[21] [18] [36] [37] [26] [46]
Intermittent object motion	[5] [22] [23] [4] [24] [40] [42] [25]
Bootstrapping	[19] [45] [18] [21] [34] [36] [37] [11] [30] [41]
Dynamic background	[45] [23] [1] [11] [30] [31] [4] [19] [20] [18] [21] [34] [35] [46] [36] [38] [39] [41] [26] [25] [44]
Shadows	[23] [11] [30] [4] [19] [34] [46] [37] [40] [26] [25]
Gradual illumination changes	[45] [1] [30] [19] [18] [36] [46] [38]
Sudden illumination changes	[45] [11] [19] [18] [34] [35] [36] [46] [41] [43]
Precipitation	[1] [35] [11] [46] [41] [42] [44]
Door opening/closing	[46]
Noise	[11] [19] [20] [37] [29] [43] [44]
Camouflage/low contrast	[45] [30] [19] [18] [32] [36] [38]
Camera shaking	[23] [4] [37] [41] [26] [25]
Reflections	[42]
Video compression	[19] [44]
Foreground aperture	[45] [18] [36]
Objects with some moving and some static parts	
Thermal sequences	[23] [4] [25]

Table 3.1-1 - Scenarios in which background models have been evaluated in previously conducted evaluations

Though not reported as a challenge considered in the reviewed evaluations, the simple moving object scenario is, inherently, part of each video that is used. Aside from this, all challenges that are listed in Table 3.1-1 are those which have been reported as being examined by evaluators. It should be noted, however, that many video sequences are polluted by challenges other than that being considered. A video used to examine how well a model deals with gradual illumination changes, for example, may also contain some shadow. Although these polluting challenges may be present in the videos that are used, they are not reported here as they are not noted in the results of the evaluations. In the evaluation undertaken by Goyette et al. [4] measures involving ground truth are taken to ignore the effects of polluting challenges. These measures are described in section 3.1.2.

Table 3.1-1 lists many challenges which background models may face in use. Given the significant effect that these can have on model performance and the frequency with which most are encountered, a background model evaluation cannot be considered complete unless these are considered. Without testing a model in these commonly encountered scenarios it is not possible to gain a full appreciation of its capabilities. It may be seen from Table 3.1-1 that some scenarios are given significantly more attention than others. The dynamic background challenge, for example, is considered in almost two thirds of the previously conducted evaluations that were examined. This challenge is prevalent and is therefore critical in fully assessing model performance. While it is good that it is often being considered, its exclusion from some evaluations is a significant shortcoming. Other commonplace challenges such as shadows, bootstrapping and illumination changes are also garnering some attention though, again, these are not always considered, while others are being almost completely disregarded.

This review of the scenarios considered in previous evaluations clearly shows that the coverage of challenges on which background models are being tested is insufficient. No single evaluation examines a range of challenges extensive enough to gain a full appreciation of model capabilities. While the exclusion of unusual challenges such as thermal video sequences can be forgiven, the omission of extremely common challenges cannot. An evaluation which considers just a small number of challenging scenarios is only considering some aspects of model performance and thus severely limits the practicality and applicability of the results that are obtained. A dataset which depicts a large range of challenges is fundamental to accurately determining the strengths of a model as well as where its weaknesses lie. The omission of critical challenges indicates, therefore, that the reported evaluations are not providing a full assessment of background model performance. In addition, different evaluations consider different combinations of challenges meaning that results cannot be compared across evaluations. This significantly reduces the usefulness of evaluation results.

It should also be noted that, although a challenge may be considered in an evaluation it is not necessarily given adequate attention. All instances of a challenge will not be the same and thus it is necessary that videos depicting variations on each challenge are included in the dataset. This will ensure that evaluation results are not too specific to a particular video. In the case of a scene with a dynamic background, for example, there may be variation in both the amount of dynamism that is present and its source. Variation in how challenges manifest themselves is ignored in a large proportion of the reviewed evaluations. In many evaluations, including that carried out by Toyama et al. [18], just a single video is used for each scenario. One of the few evaluations that have given

consideration to this is that reported by Goyette et al. [4]. In this evaluation between four and six videos depict variations on each challenge.

3.1.1.6 Current State of Dataset Usage in Background Model Evaluation

From analysing the background model evaluations that have been reported in the relevant literature to date, it has been found that the datasets that have been used are of an insufficient standard. In most cases, just a small number of videos which tend to be very short are used. In addition, the range of challenges that are considered tends to be poor, with common challenges being disregarded in many evaluations. As well as this, the challenges that are considered are frequently addressed poorly, i.e. with just a single video. The evaluations that were examined have, for the most part, used very different datasets, none of which are suitable for assessing model performance with confidence. A standard, comprehensive dataset, on which all models are tested, is necessary to thoroughly and objectively assess their performance. Until this resource exists, background model evaluations will continue to be inadequate and it will not be possible to hold their results in confidence.

3.1.2 Ground Truth

As was described in the previous chapter, background subtraction results in the production of binary difference images which correspond to the frames of the video being processed. In these images foreground pixels or pixels of interest are typically white while background pixels or those that are not of interest are black. To assess how accurately a model has segmented a video frame into foreground and background, the resulting binary difference image for that frame must be compared against a correct frame segmentation. This correct segmentation is known as ground truth. In this section the various types and characteristics of ground truth are described, the issues regarding its creation are discussed and the ground truth that is used in the background model evaluations that have been reported in literature to date is examined and appraised.

3.1.2.1 Ground Truth Type

Ground truth typically takes one of two distinct forms – pixel-based or bounding boxes. Pixel-based ground truth involves the labelling of each pixel of the video frames. Traditionally, the pixels of ground truth images are classified into one of two categories – background or foreground – and are labelled accordingly with foreground pixels typically being coloured white and background pixels being coloured black. A sample binary ground truth frame labelled in this manner, along with the corresponding original frame, is shown in Figure 3.1.4. A significant number of studies including those carried out by Panahi et al. [38] and Toyama et al. [18] make use of ground truth of this type in their background model evaluations.



Figure 3.1.4 - (a) Original frame, (b) Corresponding binary ground truth frame. Foreground pixels are coloured white, background pixels are coloured black (source: [47])

Sometimes, additional classification categories are defined for ground truth images allowing for more information to be included in them. In the ground truth that was created for the ChangeDetection.net datasets [16], for example, each pixel is labelled as being in one of five categories – static (background), hard shadow, outside region of interest (non-ROI), unknown motion and motion (foreground). Each of these categories has an associated greyscale level which is used to colour the pixels assigned to that category so that they may be distinguished. A frame from the 2012 ChangeDetection.net dataset and its corresponding ground truth frame may be seen in Figure 3.1.5.



Figure 3.1.5 - (a) Original frame, (b) Corresponding ground truth frame with a variety of labels. Foreground pixels are coloured white, background pixels are black, unknown pixels are light grey and shadow pixels are dark grey (source: [14])

This classification of pixels into more than the standard two categories can enhance the quality of background model evaluations by allowing for a better performance assessment to be carried out. An example of how the evaluation process may be improved is the use of the unknown label by Goyette et al. [4]. As will later be discussed in greater detail, it can sometimes be difficult to determine if a pixel should be classified as background or foreground in areas such as foreground object boundaries. Rather than arbitrarily choosing one of these classifications, a pixel whose state is unclear can instead be labelled as unknown to illustrate the uncertainty associated with it. The inclusion of the unknown category means that such pixels can be ignored in evaluation and will thus

not corrupt the calculation of evaluation metrics (see section 3.1.3). In addition, the non-ROI label can be used for two purposes as described by Goyette et al. [4]. The first of these is to ensure that metric calculations are not corrupted by challenges other than that being considered. For example, given a video sequence which contains moving shadows and a partial dynamic background, the pixels in the dynamic part of the background may be given the non-ROI label to indicate that they should be ignored allowing the performance of a model in dealing with just the moving shadows to be examined. This is depicted in Figure 3.1.6. The second purpose of the non-ROI label [4] is to prevent the corruption of the evaluation metrics by initialisation errors. This involves labelling every pixel in the first several hundred frames of each video with the non-ROI label. Additionally, the use of a shadow label to distinguish hard shadows from the background and the foreground can be advantageous as it allows for models to be evaluated based on how well they detect shadows and on how well they can incorporate them into the background. Thus, the use of ground truth categories additional to background and foreground can be beneficial in evaluating the performance of background models in several ways including the avoidance of corruption to evaluation metrics and the enhancement of the ways in which the models may be evaluated.

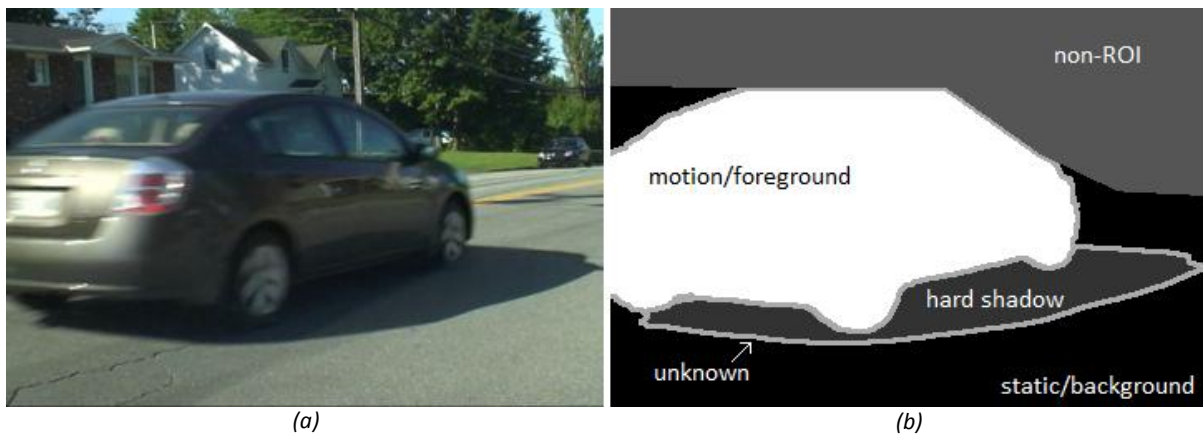


Figure 3.1.6 – (a) Original frame, (b) Corresponding ground truth frame containing each of the five pixel labels. The non-ROI label is used so that the trees waving in the background are ignored and the shadow alone is considered. (source: [14])

Brostow et al. [48] take the classification of pixels in ground truth even further than the five categories used by Goyette et al. [4]. In their efforts to create a ground truth database to “quantitatively evaluate emerging algorithms” [48], thirty-two categories including wall, sky, car and road were defined. An example of a ground truth frame created using these categories may be seen in Figure 3.1.7. This level of classification can be beneficial in evaluating some types of computer vision algorithms. In terms of background model evaluation, however, much of the extra information does not provide additional benefit. This is largely because the segmentation of different background objects is not of interest in background modelling meaning that a large amount of the information would not be relevant. In addition, the useful categories such as unknown pixel status and shadow that were used for the ChangeDetection.net ground truth [4] are not included here. Thus, while the ground truth created by Brostow et al. [48] is very detailed and extremely useful in some applications it is not as well suited for use in background model evaluation as that created by Goyette et al. [4] with its carefully selected categorisations or binary ground truth with a basic background and foreground classification scheme.



(a)

(b)

Figure 3.1.7 - (a) Original frame, (b) Corresponding ground truth frame containing labels defined by Brostow et al. [48] (Source: [49])

Ground truth in the form of bounding boxes is also sometimes used. In this case, bounding boxes are drawn around objects of interest such as people and cars rather than individually labelling each pixel. This type of frame annotation is typically in an XML format and is supplied with some of the available video datasets such as the CAVIAR dataset [7] and the Video Surveillance Online Repository [50]. An example of a frame from the CAVIAR dataset [7] annotated using bounding boxes is shown in Figure 3.1.8. As was the case with pixel-based ground truth, information additional to the basic foreground/background distinction is sometimes included in order to improve the utility of the dataset. In the scene presented in Figure 3.1.8 [7], for example, two types of bounding boxes are used – yellow boxes for individuals and green boxes for groups of people. Head, shoulder, hand and foot positions are also marked as is the direction of the line of sight of each individual.

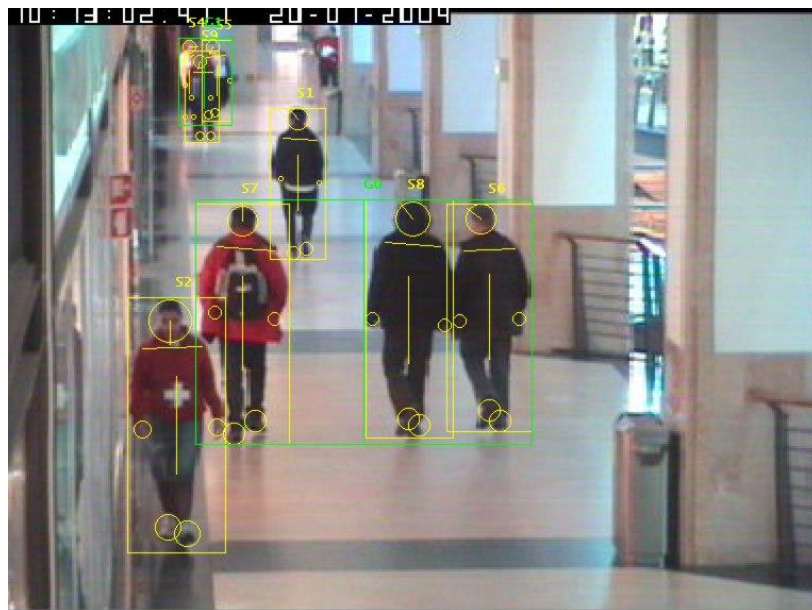


Figure 3.1.8 - Bounding box annotated video frame from the CAVIAR dataset (Source: [1])

Bounding box annotation is of particular use in applications such as object tracking. Given the nature of the result frames produced in background subtraction, however, pixel-based ground truth is of far more use in background model evaluation. The remainder of the discussion regarding ground truth, therefore, will primarily concern the pixel-based variety.

3.1.2.2 Ground Truth Creation

There are, unfortunately, a number of issues concerning the use of ground truth. It was mentioned in section 2.2.1.1, for example, that the distinction of the foreground and background of a scene is not always clear. There is no consensus as to exactly how the frames of a video sequence should be classified with there being much confusion around issues such as if and when foreground objects which come to a stop should be integrated into the background and whether this should vary based on the type of foreground object in question. As ground truth is intended to be a correct segmentation of a scene against which background subtraction results can be compared, this lack of agreement as to how to correctly create it is a major issue. Without knowing which parts of a frame should be foreground and which should be background it is not possible to have complete confidence in the ground truth frames and nor is it therefore possible to have complete confidence in evaluation results obtained using these frames. Evaluations are being carried out to assess the accuracy of result frames without knowing what exactly is to be evaluated. In the ground truth that is currently available and used in the reviewed evaluations, decisions such as how quickly the background should be updated are made by those creating the ground truth and are thus not made consistently. In one set of ground truth frames, for example, a foreground object may be considered background immediately after it comes to a stop while in another it may not be considered background until it has been stopped for a set number of frames. To objectively and fairly evaluate background model performance it is essential that these inconsistencies are eliminated and that all models are tested using the same criteria.

A second issue is the manual nature in which ground truth is typically created. As was discussed previously, the creation of pixel-based ground truth requires the labelling of each pixel based on the category (e.g. background, foreground, shadow) that it falls into. Completing this task manually is very time-consuming meaning that ground truth can be very expensive to obtain and that the amount which is available is severely limited. A number of tools are available which aid in the creation of pixel-based ground truth including InteractLabeler developed by Brostow and Fauqueur [48] as well as those created by Grossmann et al. [51] and Nascimento et al. [22]. Some of these tools [51], [22] make ground truth creation a semi-automated process meaning that the work that is involved can be reduced. Other tools such as ViPER [52] exist which allow for the annotation of video frames using bounding boxes. Despite the availability of such tools the creation of ground truth is still quite a slow process and, as a result, not a great deal is readily available. A possible solution, as presented by Brutzer et al. [19] amongst others, is to make use of synthetic videos. Ground truth is easily and quickly obtainable for such videos but, as discussed in section 3.1.1.4, the use of this type of data means that realism is sacrificed and thus that models are tested in scenarios which are not truly representative of the challenges they will face in practise.

Ideally, every frame of all videos used in a background model evaluation would be accompanied by a corresponding ground truth frame, so that the accuracy of all result frames could be analysed, allowing for an extensive assessment of model performance to be conducted. Due to time and resource constraints, however, this tends to be impractical and thus extensive ground truth is

typically not used in evaluation. One rare example of extensive ground truth availability is the 2012 ChangeDetection.net dataset [14], the videos of which are accompanied by ground truth for every frame. Though this dataset does have weaknesses in terms of diversity, the volume of ground truth data that it provides is commendable. This is uncommon, however, and, due to the large amount of effort that is involved in creating ground truth for every frame, it is typically only available for some video frames of a dataset. The LIMU dataset [53], for example, provides one ground truth frame for every sixteen video frames after an initial training phase. The Wallflower dataset [18], which was seen to be popular in background model evaluations, meanwhile, is accompanied by just a single ground truth frame for each video sequence. This minimal amount of ground truth is not sufficient to effectively assess how well a model has dealt with the challenge depicted in a video. Many other datasets have no accompanying ground truth or are supplied with just bounding box data [6] which, as mentioned, is not suitable for this type of work. Due to this shortage of ground truth availability the amount that is used in background model evaluation is typically small. The evaluation reported by Yuk et al. [34], for example, uses just twenty ground truth frames distributed throughout each video regardless of the number of frames that are in the videos, while the study conducted by Tsai et al. [1] uses between eleven and seventy consecutive ground truth frames close to the end of the videos. In some cases, evaluators attempt to create their own ground truth but they typically do not have the resources to do this sufficiently.

Another issue related to the creation of ground truth is its accuracy. Unless synthetic videos are used, which, as has previously been discussed, is not ideal, it is not possible to automatically create reliable ground truth. Ground truth, therefore, is largely created by humans and, as a result, there is much scope for opinions and misinterpretations to affect the result. If multiple people were to make ground truth for the same video frame, for example, it is quite likely that there would be a significant amount of difference in the outcomes. This is simply a result of differences in human judgement. This is illustrated in Figure 3.1.9 by an example from the Berkeley Segmentation Dataset [54, 55].

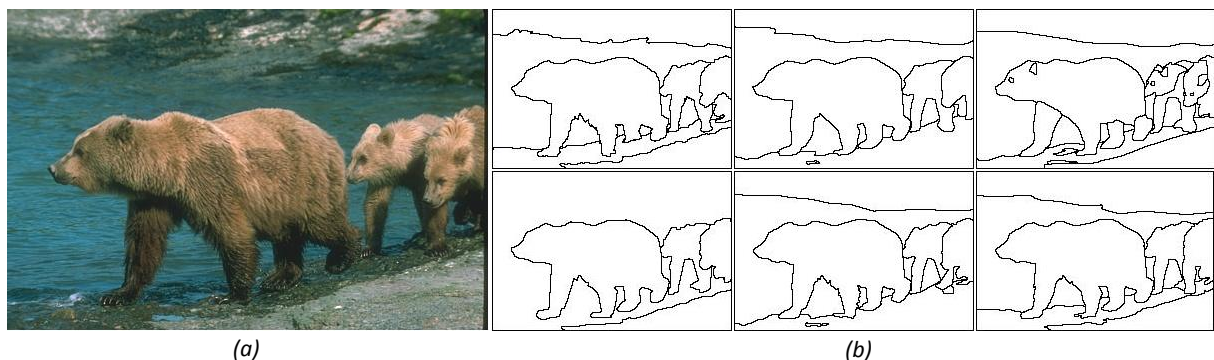


Figure 3.1.9 - (a) Original image, (b) Segmentations of original image, each produced by a different person (Source: [54])

On the left-hand side of Figure 3.1.9 is an original image from the Berkeley Segmentation Dataset [54, 55] while on the right are six different ground truth segmentations of this image performed by six different people. The aim here is to simply mark the boundaries in the image. Although this type of information is different to that which is required in background model evaluation it serves as an excellent example of how all humans will not interpret an image in the same way. It can clearly be seen, for example, that the water line at the top of the image is handled differently in each of the segmentations, as are the front leg and nose of the largest bear. There are also many other

discrepancies in how the human segmentations represent the object boundaries. In addition, some inaccuracies around the object boundaries in the image segmentation presented in Figure 3.1.7 are clear. This lack of consensus as to exactly where an object boundary lies is a major problem in the creation of ground truth. When attempting to distinguish the status of boundary pixels in a frame it is often unclear whether they form part of a foreground object or part of the background, leading to guesses and, inevitably, incorrect classifications being made. This problem is also encountered in areas of motion blur, windows, sparse bushes and trees where foreground and background pixels may be closely mixed together, amongst others. As ground truth is to be regarded as a correct classification, this is not acceptable as, if the ground truth cannot be held in confidence, nor can the evaluations that are carried out with it. It is very difficult, however, to completely avoid misclassifications in ground truth creation.

While misclassifications are inevitable, it is possible to minimise and, in many cases, ignore them. One method of reducing the occurrence of misclassifications is to have multiple people create ground truth for each frame and use the consensus of their classifications as the final accepted ground truth, e.g. if twenty people were to create ground truth for the same frame and fifteen of these considered a certain pixel to be foreground while the remaining five considered it to be background, the pixel would be regarded as foreground. This approach to improving ground truth quality does not appear to have been used in any of the evaluations that have been examined. An example of combining results from different individuals in the context of image segmentation based on object boundaries, however, may be seen in Figure 3.1.10. The six segmentations shown in Figure 6 are combined to form the image on the right hand side of this example. The brighter the pixels in this image the greater the likelihood of them being a boundary pixel. While this approach would greatly improve the quality of ground truth data and would thus allow for a more accurate assessment of background model performance, the amount of time and human effort that would be involved in achieving this is extremely significant and generally impractical.



Figure 3.1.10 - (a) Original image, (b) Consensus of original image segmentations shown in Figure 3.1.9 (Source: [54])

An alternative method of addressing the presence of pixels in video frames whose status is unclear is the use of a ground truth label to distinguish them. By doing this, such pixels may be ignored when background subtraction result frames are compared to the corresponding ground truth. This makes for a more fair evaluation as only those pixels whose status is certain are considered, i.e. it will ensure that a model is not penalised for an error in the ground truth and that the evaluation metrics will not be corrupted by such errors. This approach, as mentioned previously, is used by Goyette et

al. [4] in their ChangeDetection.net datasets. In this case, the “unknown” ground truth label is used for pixels which are difficult to classify. The use of this label may be seen in Figure 3.1.5 and Figure 3.1.6. Evaluations which make use of this dataset, including those reported by Goyette et al. [4] and Nonaka et al. [23], therefore, have the advantage of being more objective and trustworthy than most.

3.1.2.3 Current State of Ground Truth Usage in Background Model Evaluation

From the analysis of the various background model evaluations that have been reported in the relevant literature, it has been found that there is a significant number of issues with the ground truth that is currently in use. Perhaps the most striking limitation is the ignorance as to what exactly is to be evaluated. Without knowing how to correctly classify the pixels of a video frame it is not possible to effectively evaluate how closely the results of background subtraction have matched the correct frame classification. In addition, due to the time consuming nature of ground truth creation the amount that is typically being used is far too small to fully assess a background model’s performance. As well as this, inaccuracies in the ground truth that is created often result in models being penalised for errors that they may not have made. It is essential that decisions regarding how to correctly classify video frames are made and that a standard set of ground truth is created, for a standard dataset, for use in evaluating all background models to ensure fair and consistent results. Rather than putting effort into creating small and inconsistent amounts of ground truth, efforts should be combined to create this standard set. To aid in completing this, a resource similar to LabelMe [56], a crowd sourced image annotation tool produced by the MIT Computer Science and Artificial Intelligence Laboratory could be used.

3.1.3 Evaluation Metrics

A critical part of a background model evaluation is the evaluation metrics that are used to quantitatively assess model performance and allow for the capabilities of the models to be compared. There are several aspects of a model’s performance which may be measured including its accuracy or how well the result frames that are produced match the corresponding ground truth, the length of the training phase, the degree of lag that is present and its efficiency. This section examines these aspects of model performance and the associated metrics. In addition, the use of metrics in the background model evaluations that have been reported in literature to date is discussed.

3.1.3.1 Accuracy Metrics

In order to evaluate the accuracy of a background model the result frames that are produced using it are compared to the corresponding ground truth frames. In making this comparison, some form of numerical measure or metric is needed to assess their similarity and thus, the accuracy of the model. There exist a number of evaluation metrics, typical of binary classification, which are often used in doing this. The most basic of these metrics are shown in Table 3.1-2.

		Actual Value (Ground Truth)	
		True	False
Result Value (BGS result)	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Table 3.1-2 - Basic binary classifications

- A **true positive** (TP) is a correctly identified result, i.e. a pixel that is a foreground pixel in the ground truth is correctly identified as foreground by the background model.
- A **false positive** (FP) is an incorrectly identified result, i.e. a pixel that is a background pixel in the ground truth is incorrectly identified as foreground by the background model.
- A **true negative** (TN) is a correctly rejected result, i.e. a pixel that is a background pixel in the ground truth is correctly identified as background by the background model.
- A **false negative** (FN) is an incorrectly rejected result, i.e. a pixel that is a foreground pixel in the ground truth is incorrectly identified as background by the background model.

The metrics that are typically used for evaluating a binary classification or, in this application, how well a result frame of a background subtraction matches the corresponding ground truth frame, are derived from these four measures. Some of the most commonly used metrics are briefly described below.

- **Accuracy** is the proportion of true results in the entire result population. An accuracy of 100% indicates that the background model results are identical to the ground truth. Thus, a value close to one is desirable. This metric is used in a number of studies on background model evaluation including those carried out by Rosin et al. [29] and by Bashir et al. [24].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Precision**, sometimes called the positive predictive value, is the proportion of true positives in all of the positive results, i.e. it is the proportion of pixels that are correctly identified as foreground in all of the pixels that are identified as foreground. A precision value close to one is desirable. A large number of the previously conducted evaluations [4, 11, 19, 20, 23, 30, 31] that were examined make use of this metric.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**, sometimes called sensitivity or true positive rate, is a measure of how well the model can identify correct results. In other words, it is the proportion of pixels that are correctly identified as foreground in all of the known foreground pixels. A recall value close to one is desirable. Recall is used in a large number of the reviewed background model evaluations [4, 11, 19, 20, 23, 30, 31].

$$Recall = \frac{TP}{TP + FN}$$

- **F-measure**, sometimes called F-score or F1 score, is a harmonic mean of precision and recall. It is a measure of how accurately pixels have been classified as background or foreground. A high F-measure value is desirable with one being the highest possible value. This is a popular metric which is used in several studies including that conducted by Goyette et al. [4].

$$F - measure = 2 \frac{Pr \cdot Re}{Pr + Re}$$

- **Specificity**, sometimes called the true negative rate, is a measure of how well a model can identify negative results. In other words, it is the proportion of pixels that are correctly identified as background in all of the known background pixels. A good specificity value will be close to one. This metric is used in just three of the reviewed studies [4, 23, 24].

$$Specificity = \frac{TN}{TN + FP}$$

- **Negative predictive value (NPV)** is the proportion of pixels that are correctly identified as background in all of the pixels that are identified as background. A value close to one is desirable for this metric. Of the evaluation studies that were reviewed just that carried out by Bashir et al. [24] makes use of this metric.

$$NPV = \frac{TN}{TN + FN}$$

- **False positive rate (FPR)** is the proportion of background pixels that are incorrectly identified as foreground. The false positive rate should, ideally, be close to zero. This metric is used in studies by Nonaka et al. [23] and Goyette et al. [4], amongst others.

$$FPR = \frac{FP}{FP + TN} = 1 - Sp$$

- **False negative rate (FNR)** is the proportion of foreground pixels that are incorrectly identified as background. A value close to zero is desirable for the false negative rate. A number of the reviewed background model evaluation studies including those conducted by Goyette et al. [4] and by Vacavant et al. [11] use this evaluation metric.

$$FNR = \frac{FN}{FN + TP} = 1 - Re$$

- **Percentage of wrong classifications (PWC)** is the percentage of false or incorrect results in the entire result population. A low PWC value is desirable. This metric is used in the evaluations of Nonaka et al. [23] and Goyette et al. [4]

$$PWC = \frac{100(FN + FP)}{TP + FP + TN + FN}$$

- **Similarity**, or the Jaccard coefficient, is a measure of how similar the result frames produced using a background model, are to the corresponding ground truth. A value close to one is desirable for this metric. A small number of the reviewed studies [1, 30] use this metric in their evaluations.

$$Similarity = \frac{TP}{TP + FN + FP}$$

The accuracy metrics described above are just a sample of those which may be used. There exist numerous other options such as total error (FP + FN) [18], normalised probabilistic rand (NPR) index

[57], D-score [11] and weighted quality measure (WQM) [5]. In addition to the simple numerical metrics discussed thus far, PR (Precision-Recall) curves and ROC (Receiver Operator Characteristic) curves are often used. These graphically depict the performance of the algorithm that is being examined in terms of its accuracy. To create a PR curve, as used to present results by Brutzer et al. [19], precision is plotted against recall whereas to create an ROC curve [20], true positive rate is plotted against false positive rate. In PR space, the upper right-hand corner of the plot is indicative of good performance while in ROC space the upper left-hand corner is indicative of this. PR curves can be more informative than ROC curves and “can expose differences between algorithms that are not apparent in ROC space” [58]. It is also worthwhile noting that, in a study undertaken by SanMiguel and Martínez [59], some measures of a background model’s accuracy are presented which are not based on the comparison of result frames to corresponding ground truth frames. Instead, the boundaries of the detected foreground objects are compared “against the colour and motion boundaries of the video sequence” [59]. The use of colour boundaries were found to be more effective than using motion boundaries but neither method matched the success of the standard practice of using ground truth. The use of such accuracy measures is not widespread.

As has been shown here, there are a large number of metrics to choose from when evaluating the accuracy of a background model. As a result, the selection of appropriate metrics can be very challenging. There is no one correct choice as to which metrics should be used and thus, as has been seen, different evaluations have made use of different metrics which renders their results incomparable. The most popular accuracy metrics used in the reviewed evaluations were precision, recall and f-measure. These were used in approximately 30% of these previous evaluations which is indicative of the inconsistency prevalent in metric selection. In addition, many of the background model evaluations that have been examined have made use of multiple metrics. Young and Ferryman [5] note that “an ideal evaluation process would be a single metric that could evaluate any algorithm for a specific task. A single score by which to rank algorithms would render easy comparison of algorithms.” Problems can arise in attempting to achieve this, however. If a metric is used for addressing a large number of algorithms it is possible for it to become too general meaning that subtle differences between algorithms may be overlooked. Conversely, metrics may be specialised to address assumptions associated with them. This can lead to the metrics being applicable to only a small number of algorithms. In both of these scenarios it becomes difficult to meaningfully compare the performance of different algorithms against one another. Care must therefore be taken to ensure that the metrics that are chosen are not “too specialised to one application nor too generalised to a large set of applications” [5]. In studies which use multiple accuracy metrics such as that reported by Goyette et al. [4] the metric values are averaged to produce a single value by which models may be ranked. In doing this, it is important not to use too many metrics as this could result in an excess of information which could again hinder the comparison of the algorithms.

3.1.3.2 Efficiency Metrics

The analysis of background model efficiency is of great importance in the overall evaluation of model performance. Efficiency analysis is essentially the measurement of a model’s resource usage. In order for efficiency to be high, resource usage must be low. The resources that should be monitored in background model performance evaluation are processing speed and memory usage. Models are often required to work in online applications which require video frames to be processed in real time meaning that processing time must be low. In addition, a model may be required to work on a low

memory system such as an embedded system in which case it would be essential for a model to consume as little memory as possible.

A background model's processing speed and memory usage will vary based on the number of frames that are in the videos being processed and the resolution of these frames. To obtain fair and comparable values for these metrics, therefore, they must be measured for each model using the same videos. Background model efficiency is considered in a small number of the reviewed evaluations including those conducted by Piccardi [60] and by Tavakkoli et al. [41].

3.1.3.3 Training Phase Length

It was mentioned in chapter 2, that some background models require a training period during which they use frames from the video stream to create a background image before background subtraction begins. The length of this phase varies between models and can be important in the selection of an appropriate model for a specific application. Thus, the length of the training phase or the number of frames that a model uses to create its initial background image, is another aspect of model performance which should be considered in order to obtain a full appreciation of a model's capabilities. At present, no background model evaluation that has been reported in the relevant literature has considered the length of the models' training phase as a metric in assessing performance.

3.1.3.4 Lag

Chapter 2 also discussed the lag of a background model. Some models have no lag, i.e. they take a frame, process that frame and return the result before looking at the next frame. Others have a small amount of lag, i.e. it is necessary to wait for a small number of frames before results are returned. Others still, have a lag equivalent to the length of the video to be processed, i.e. the model takes all of the video frames to create a background image before returning any results. The lag of a background model is a crucial factor in determining its suitability for use in an application. In an online application, for example, in which results are required in or close to real time, only online models or those with a small degree of lag are acceptable. For an offline application, meanwhile, the degree of lag does not matter. Given that a model's lag dictates the applications in which it can be used, it is plainly a very important aspect of performance to consider and is critical in fully assessing model capabilities. None of the reported background model evaluations consider this as a metric and are therefore not thoroughly evaluating model performance.

3.1.3.5 Current State of Evaluation Metric Usage in Background Model Evaluation

It has been seen that there are several aspects of a background model's performance that may be assessed. Without considering all of these aspects it is not possible to attain a full appreciation of the performance of a model. In reviewing the background model evaluations that have been reported in the relevant literature to date, however, it was found that no evaluation is considering all performance aspects of the models that they are evaluating. This is a major weakness of the existing body of evaluations. Figure 3.1.11 presents an overview of the extent to which the various aspects of background model performance have been given consideration.

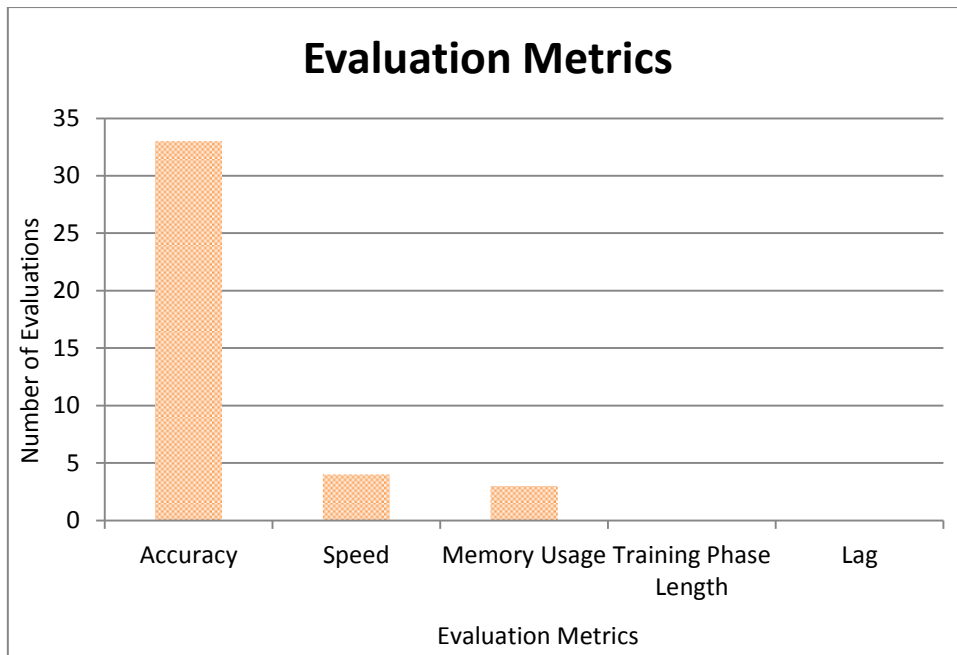


Figure 3.1.11 - Aspects of background model performance considered in evaluation

From the graph in Figure 3.1.11 it is clear to see that all of the thirty-three previously conducted evaluations that were reviewed, assessed model performance in terms of accuracy. While the measurement of model accuracy is a critical part of evaluation, it is just one aspect of their performance. In addition, it must be noted that, although accuracy is being considered in each of the evaluations that were studied, many different combinations of accuracy metrics were used.

Despite its importance to model selection, just a small number of evaluations give consideration to the efficiency of the background models that they are assessing. As model efficiency is an essential part of performance assessment its exclusion from such a large portion of evaluations is a significant limitation of the research that has been carried out to date.

It may also be seen from Figure 3.1.11, that none of the reviewed evaluations considered either the lag of the background models being assessed or the length of their training phases. Depending on the application, knowledge of these can be of critical importance in determining the appropriateness of a model. This is a significant weakness of the way in which the performance of background models is currently being evaluated.

From this analysis it is clear that the background model evaluations that have been reported in literature are not considering all aspects of model performance and are therefore not fully assessing their capabilities and characteristics. The omission of several important aspects of model performance indicates a serious flaw in how previous evaluations have been carried out and in the data that is being used as a basis for model selection. It is essential that all factors which influence a model's performance are analysed as, otherwise, the necessary information will not be available for making an informed and adequate decision as to which model is appropriate for use in a particular application. Additionally, it will not be possible to meaningfully compare the abilities of various background models nor will it be possible to gain a full appreciation as to the strengths and the limitations of a model. For background model evaluations to be recognised as complete and acceptable by the research community this issue needs to be addressed.

3.1.4 Comparison Models

Another important aspect of background model evaluation is the variety of models that are chosen for comparison. It has been seen in the previous sections that there are many significant issues in how comparisons are currently made, i.e. the inadequacy and disparity in the videos, ground truth and evaluation metrics that are used. It should also be noted, however, that evaluations may have weaknesses in terms of the models that have been selected for assessment. A large number of the evaluations that have been reported in literature have been carried out by developers looking to determine and illustrate how well their model performs in comparison to existing models. More than 90% of the reviewed evaluations consider less than ten models for comparison. In the majority of cases, the comparison models tend to be drawn from a small group of older models which are easy to implement or for which existing implementations are readily available. As a result, many newly developed background models are not being compared against the more recent and advanced models that are available.

As background models are, in general, only being compared to a small number of other models, only their performance in the context of these models is being seen. How they compare to the body of available background models in general, however, is unknown. To realistically determine a model's capabilities, it should be compared against all other models, especially the more recent ones which are largely being overlooked. This is an unrealistic expectation of the small scale evaluations that are currently being carried out by developers and it is therefore necessary that an extensive collection of evaluation results be compiled so that developers can simply compare their model performance to it.

3.1.5 Background Model Parameters

As was seen in section 2.2.2, background models have a variety of parameters associated with them including thresholds, learning rates and initial values. The values that are given to these parameters can have a profound impact on the effectiveness of a background model yet the tuning of model parameters is often not given the consideration that is warranted both in use and in evaluation. When working with a single video, a background model may produce completely different results when different sets of parameters are used. In addition, the optimal parameter values for an algorithm in one scenario will not necessarily be optimal in another scenario. For these reasons, to optimise the performance of a background model, it is important that its parameters are selected with care. This, however, can be a very difficult task and can necessitate expert knowledge making the tuning of model parameters a potentially expensive, arduous task whose importance is often overlooked. The ways in which parameter tuning has been addressed in the background model evaluations that have been presented in literature as well as their merits and demerits are discussed here.

In some studies, such as that carried out by Goyette et al. [4], if the models being assessed have been developed by someone other than the authors of the study, the parameter values provided for these models in literature are used. While this is beneficial in that the evaluator does not have to determine parameter values themselves, it also means that the values that are used may not be optimal for the scenarios being considered. This is of particular concern if the authors of the model tuned its parameters using a small dataset as this will likely lead to them being overfit to this dataset rather than being general enough to work well with more diverse ranges of scenarios. There are, of course, circumstances in which the tuning of parameters to a particular scenario is desirable but if

the model creators recommend a set of parameter values many users will accept these as being the most suitable choice and will use them in any scenario. If tuning was not performed over a large number of scenarios the performance of the models will likely be diminished rendering the evaluation an imprecise depiction of their capabilities.

A popular option is to manually tune model parameters using the dataset which is to be used in the evaluation. This approach was taken by approximately half of all of the evaluations that were reviewed. There exist several methods of manually tuning parameters such as ROC analysis as performed by Brutzer et al. [19]. In this study the parameter values suggested by the creators of the models are used as a basis for the parameter search and only a small range around these is searched. This has the potential to introduce some of the issues with assuming the suitability of developer provided values that were mentioned above. Another approach, is the optimisation of a selected metric. In the study conducted by Herrero and Bescós [31], for example, the parameter space is searched for values which optimise the f-measure metric. Again, the developer provided parameter values are used as a basis for this search. In the cases in which a model has multiple parameters, each of the parameters was tuned individually, i.e. the optimum values for one parameter is searched for while the others are held at reasonable values.

In some studies, such as those conducted by Herrero et al. [31] and by Brutzer et al. [19], this manual tuning is performed with the aim of determining the optimal parameter values for each video in the dataset, i.e. different parameters are used for each video, while in others, such as those carried out by Nonaka et al. [23] and Benezeth et al. [20], a single set of optimal parameter values are used across the entire dataset. While tuning parameters to a particular video can be advantageous in that results will be optimised for that video sequence, it is not practical to tune them for every video with which a model may be used. It is likely that users of a model will choose to trust the parameter values used by its evaluators, e.g. if, as part of an evaluation, a certain set of parameters is tuned for use with a particular video which contains a dynamic background, subsequent users of the model may choose to use these same parameters for their own scene with a dynamic background. In doing this, however, as when using developer provided values, there is a risk that the parameters will be overfit to the initial video. Thus, while the model may perform very well on the video for which its parameters were optimised, when it is used with other videos whose characteristics differ, performance may be significantly diminished. In order to reduce these effects of overfitting to a certain video, a number of video sequences should be used in parameter tuning. This was the case in the majority of the evaluations that were considered, i.e. parameters were tuned to optimise results over the entire dataset and these same parameter values were used for all videos in order to ensure fairness in the evaluations.

As an alternative to the manual tuning of parameters, a number of studies have considered models which automatically determine their values. White and Shah [45], for example, make use of a Particle Swarm Optimisation (PSO) algorithm to automate the selection of parameter values which maximise the selected fitness function, f-measure. Parameter values which maximise this fitness function are taken to be optimal. As part of this study, parameters were optimised for each video as well as for the dataset as a whole. Under each of these scenarios, the background model being

tested⁴ was used in processing the entire Wallflower dataset [18] and the results were compared to those reported for this model and dataset by Toyama et al. [18] which were obtained using the same set of manually tuned parameters for all videos. It was found that the automatic tuning of a single parameter set for the entire dataset was far more effective than was the manually tuned set that was used in obtaining the reference results. When using the automatically determined parameters the total error in results was almost halved in comparison to the reference results. This indicates the superiority of automated tuning to the manual alternative. It was also seen that the tuning of model parameters for each video “further reduced the total error by a significant amount” [45] illustrating that “the optimal parameters for one scene are not necessarily best for another” [45]. This is reinforced by the results of the evaluation conducted by Goyette et al. [4] in which the success of one of the most successful background models [61] is attributed, by the authors, “to the use of a dynamic control algorithm for automatically adapting thresholds and other parameter values” [4].

From the findings of the evaluation studies that have been reviewed it is clear that the best option for tuning background model parameters is to automate the process. This allows for far more accurate results to be obtained than is typically achievable with manual tuning. As well as this, the automatic tuning of model parameters makes the use of background models easier and far more accessible. It allows for good results to be achieved more quickly and without the need for expert knowledge in the area. Additionally, it was seen that tuning parameters per scene will provide more accurate results than will tuning them over the entire dataset but that the determination of parameter values at a dataset level is preferable when more well-rounded and general applicability is desired. The tuning of parameters per scene is reasonable when using automatic tuning but, when parameters are being tuned manually it is typically impractical, time-consuming and increases the scope for error. Manual parameter tuning, therefore, should generally be restricted to obtaining a single set of parameters for a dataset. The final option of simply using the parameter values that were recommended by the model developers has the advantage of making the use of models easier as the user does not have to tune parameters themselves but also risks sacrificing result accuracy depending how applicable the recommended parameters are to the scenario in which the model is being used. In conclusion, parameters should be automatically tuned where possible. If this is not an option, a single set of general parameter values should be manually determined over an entire dataset. The way in which parameter tuning is currently handled in background model evaluation is inconsistent and inadequate. Given the significant impact that parameter values can have on the performance of a background model it is important that their determination be given sufficient consideration.

3.1.6 Evaluation Frameworks

A large majority of the background model evaluations that have been carried out have been conducted by developers with the aim of testing how well their model performs in comparison to others. As has been seen, however, these evaluations have a significant number of limitations in terms of the inadequate and disparate nature of their evaluation methodologies. As a result, they have failed to comprehensively assess the capabilities of the numerous background models that

⁴ The Gaussian Mixture Model is the only background model to be tested in this study but it is important to note that the approach taken to automatically tuning parameter values is not limited to this and is applicable to any background subtraction algorithm.

exist and to produce results that can be meaningfully compared with the outcomes of other evaluations. In an attempt to address these issues, a small number of evaluation systems, namely the ChangeDetection.net framework [4] and the PETS Metrics system [5], have been proposed which aim to standardise the evaluation process. This section examines the characteristics of these systems and assesses the benefits that they provide as well as the limitations that are associated with them.

3.1.6.1 ChangeDetection.net

The ChangeDetection.net background model evaluation framework [4] was designed as part of the 2012 IEEE Change Detection Workshop [62]. It aims to provide a comprehensive method of benchmarking background models in terms of their performance. As part of this, a publically available dataset containing thirty-one videos which depict six different challenges was compiled and a detailed set of ground truth images was created to accompany the videos. A set of accuracy evaluation metrics was also selected for use. To evaluate the performance of a background model using this framework participants must download the video dataset from the framework website [62], run their model with the videos and submit the resulting background/foreground segmentation of the video frames to the framework via the website. Upon receiving result frames, the various accuracy metrics are computed and published online and, in addition, the model is ranked against the previously evaluated models.

This system has some merits, primarily in that it attempts to set out a standard by which models may be evaluated in an effort to compile a comprehensive ranking of background model performance data. Unfortunately, however, it suffers from a number of serious limitations which are described in detail in section 3.1.6.4. Due to these limitations, the ChangeDetection.net framework cannot be considered a full or reasonable solution to the problem of background model evaluation. It is certainly a good basis which may be built upon to create a solution but a large amount of work would be required for this to become a reality.

3.1.6.2 PETS Metrics

The PETS Metrics evaluation system [5] was developed to complement the PETS workshop but is no longer online. It considers the more general issue of visual surveillance algorithm evaluation rather than focussing solely on the evaluation of background models but does make use of background models as its proof of concept example. The aim of this system is “to provide an automatic mechanism to compare, in a quantitative manner, a selection of algorithms operating on the same data” [5]. In this case, developers must use a single video to obtain result frames for the background models and submit these results to the system in an XML format via an associated website. Once evaluated, the evaluation results and model rankings could be viewed on this same site. The PETS metrics system suffers from many of the same issues as the ChangeDetection.net framework (see section 3.1.6.4) and, as a result, cannot provide a thorough background model assessment.

3.1.6.3 Advantages of Evaluation Frameworks

The evaluation frameworks that have been described have some advantages over the evaluations that have been carried out by background model developers, primarily the consistent manner in which models are evaluated and a reduction in the amount of work that a model developer must complete. In terms of evaluation consistency, these frameworks ensure that all evaluated background models are being assessed fairly and using the same videos, ground truth and evaluation

metrics so that their capabilities in achieving the same goals may be measured. As a result of this, the performance of background models can be meaningfully compared, a significant improvement over the inconsistent manner in which evaluations have typically been performed. In addition, the frameworks allow for evaluation results to be obtained over time so that a large collection of performance data may be gradually compiled.

A second advantage of the ChangeDetection.net and PETS Metrics evaluation systems is that they reduce the amount of work that must be completed by the developers of background models. By making use of a framework like these developers no longer need to set up an evaluation, gather videos and ground truth, decide on evaluation metrics or evaluate models other than their own for comparison. As has been seen, this process is very time-consuming and requires a great deal of work. The elimination of much of this work is of significant benefit to model developers.

3.1.6.4 Limitations of Evaluation Frameworks

There are also, however, a number of limitations concerning the evaluation frameworks that have been described including the use of insufficient and difficult to update evaluation methodologies and the trust-based nature of the evaluation results that are obtained.

It was seen previously, that the discussed evaluation frameworks ensure that all background models are being assessed using the same videos, ground truth and metrics so as to obtain fair results. While it is encouraging that the issue of evaluation inconsistency is being addressed, it must be noted that the testing that is being carried out is inadequate and does not allow for a comprehensive model assessment to be completed. The PETS Metrics system, for example, makes use of just a single test video which is accompanied by bounding box ground truth information. The ChangeDetection.net framework, meanwhile, makes use of a significantly larger number of test videos which are accompanied by detailed ground truth but the range of challenges that are depicted by these videos is poor with many extremely common challenges such as illumination changes being ignored. In addition, both the PETS Metrics and ChangeDetection.net evaluation systems consider only the accuracy of the background models that are tested. As has previously been discussed, accuracy is only one aspect of a model's performance and, by only measuring this aspect, the evaluation systems are not obtaining a complete portrayal of the capabilities of the models. Thus, while these systems have the advantage of evaluating all models using the same data, the data that is being used is not of a sufficient standard to allow for a full performance evaluation to be carried out.

As well as this, it is very difficult for these evaluation systems to update their datasets or metrics should the need arise. If it becomes apparent that a challenge category has been omitted, for example, the addition of videos depicting this challenge would only be used by models evaluated after the addition and would thus render the evaluation results inconsistent. This issue has already arisen for the ChangeDetection.net framework. In preparation for the 2014 IEEE Change Detection Workshop it was decided that the dataset should be expanded to depict a wider range of challenges. In scenarios such as this, unless all those who have previously submitted results to the system are contacted and asked to produce additional results using the new videos, the background models which have already been evaluated cannot be tested with these new videos causing evaluation results, in this case, those produced using the 2012 ChangeDetection.net dataset, to become outdated. Contacting model developers in this way and requesting that they rerun tests is

impractical and unrealistic. As is illustrated by the significantly smaller number of models evaluated using the updated ChangeDetection.net dataset in comparison to the original, it is unlikely that all models will be re-evaluated using an updated dataset or metrics. This loss of evaluation consistency significantly diminishes the value of the results that are available. The expanded ChangeDetection.net dataset still lacks common challenges such as illumination changes and it is therefore conceivable that, in the future, it may again be updated, further reducing the value of the system.

A third limitation of the evaluation systems that have been discussed is their trust-based nature. As mentioned previously, model developers use the videos provided by the frameworks to obtain background subtraction result frames and submit these frames to the framework for evaluation. By having developers simply submit result frames the framework does not have control over the entire evaluation process and the accuracy of the evaluation results can therefore not be guaranteed. It is impossible to be sure that result frames were obtained correctly and in the same manner for all models and, therefore, whether the evaluation results are truly consistent. It cannot be ensured for example, that there was consistency in how parameter values were obtained for the background models. One developer, for example, may have manually obtained a single set of parameters for the entire dataset while another may have manually optimised parameter values for each video in order to improve results. In addition, it is possible that result frames may have been edited before submission to the framework. The value of the ChangeDetection.net and the PETS Metrics systems, therefore, is reliant upon the honesty and competence of those who submit their segmented frames for evaluation. As a result, it is not possible to have complete confidence in the evaluation results that are obtained.

3.1.6.5 Current State of Background Model Evaluation Frameworks

From this analysis of the attempts which have been made to date to standardise the background model evaluation process it is clear that a significant amount of work remains to be completed before a satisfactory solution is reached. The ChangeDetection.net and PETS systems address the issue of consistency in the datasets, ground truth and metrics that are used in model evaluation but as was seen in section 3.1.6.4, these are not of a sufficient standard to allow a complete evaluation to be carried out. In addition, there exist difficulties in updating the evaluation methodology as illustrated by the issues encountered in the expansion of the ChangeDetection.net dataset. The trust-based nature of the systems also means that their worth is greatly affected by the honesty of those who submit result frames for evaluation and that the integrity of the evaluation results cannot be guaranteed. As a result of these severe limitations, the evaluation systems that have been considered are not capable of comprehensively and objectively assessing background model capabilities. The analysis of these frameworks, however, has been useful in ascertaining how a background model evaluation framework should operate. In the next chapter, a framework is proposed which addresses the issues discussed here and provides a more competent method of assessing the performance of background modelling algorithms.

3.1.7 Current State of Background Model Evaluation

This chapter has reviewed a number of background model evaluations that have been reported in the relevant literature to date in terms of the test videos, ground truth and evaluation metrics that they have used, the models that they have compared and the manner in which their parameter values are determined. From this analysis, it was found that the quality of the existing body of

background model evaluations is quite poor and exhibits a number of significant weaknesses which make it impossible for them to thoroughly and effectively evaluate models, to contribute to the current understanding of background model capabilities or to otherwise be of value to the research community. From the assessment of previous background model evaluations it has been determined that there exist eight major issues which must be addressed in order to improve how evaluation is carried out and to provide the research community with an extensive and accurate reference of the abilities of the numerous background models that exist and continue to be created. The major issues that have been discovered have been discussed in detail throughout this chapter and are summarised below:

1. **No comprehensive video dataset** currently exists for use in evaluating background models. Ideally, a dataset should be diverse and provide a realistic depiction of the challenges that a background model is likely to face in use. Without this, it is not possible to accurately determine how a model will perform in a wide range of possible scenarios.
2. **Common challenges not being considered** – background models may face a wide range of challenges in use and it is important to test how well they will perform when faced with these challenges. The existing evaluations, however, have not evaluated models in an extensive range of scenarios and in many cases have ignored very important and commonly occurring challenges.
3. **Limited ground truth availability** – due to the time-consuming nature of ground truth creation, most background model evaluations make use of just a small amount, which is not sufficient to assess a model's accuracy. In addition, it is difficult to ensure that ground truth is accurate, particularly at the boundaries of foreground objects. Extensive and detailed ground truth should be used to properly assess model performance.
4. **No consensus regarding what is being evaluated** – it is not always clear which parts of a video frame should be considered foreground and which should be considered background. In general, this is often decided by those creating ground truth and these decisions are not consistently made across all evaluations. Without knowing exactly how a video frame should be classified, it is not possible to effectively evaluate how closely the results of background subtraction have matched the correct frame classification.
5. **All aspects of model performance is not being considered** – there are several aspects of a model's performance which may be measured including its accuracy, efficiency and lag. In most of the reviewed evaluations, accuracy is the only aspect of performance that is considered while efficiency is considered in a small number of cases. Without measuring all aspects of a model's performance it is not possible to gain a comprehensive understanding of its capabilities.
6. **Inconsistent evaluation methodologies** – the background model evaluations that have previously been conducted have made use of different combinations of datasets, ground truth and evaluation metrics. If the same data is not used for the evaluation of all models then it is impossible to meaningfully compare their performance.

7. **Parameter values not determined in a consistent manner** – in order for background models to be evaluated fairly and in a manner that allows their performance to be compared it is important that the same approach is taken to setting the parameters of all models. The way in which this is addressed, however, is inconsistent with some evaluations using the same parameters for the entire dataset and others optimising their values for each video. Others automatically determine parameter values.
8. **Limited number of models assessed** – most evaluations assess a very small number of background models and, in general, the same models are continuously being assessed while more recent models are largely not being considered.

These issues highlight the fact that the existing background model evaluations are not properly assessing the capabilities of the models that are being tested. Rather, evaluations are being performed inconsistently and incompletely and are not providing a fair or accurate depiction of how the various models are performing. As a result of this, the outcomes of the various evaluations cannot be trusted as giving a true illustration of the models' abilities and thus, the usefulness of these evaluations is severely limited and unhelpful in the task of selecting an appropriate model for a particular application.

This study of how background models are currently being evaluated indicates the need for a standard method of evaluation to be defined, for all models to be evaluated using this same method and for the evaluation results to be made publically available for all to use. It was seen in section 3.1.6 that there have been a small number of attempts at achieving this but that these efforts created more issues than they were able to solve. For example, standards were set to be used in the evaluation of all models in order to address the issue of evaluation inconsistency but these standards were not of a high quality. In addition, it was found to be difficult to improve these standards without rendering the evaluation results inconsistent. These efforts were also unsuccessful in being able to guarantee the fairness and accuracy of results.

From the analysis that was carried out in this chapter regarding the way in which the problem of background model evaluation has been addressed in the relevant literature to date it is clear to see that the efforts that have been made are of an inadequate standard. As a result, the current body of knowledge regarding the performance of the numerous models that are available is limited and there exists no straightforward method of obtaining definitive and accurate information regarding a model's capabilities. To rectify this, it is essential that the current evaluation methods be improved. A methodology for how background models should be assessed needs to be developed and agreed upon. Based on the findings of the research that has been presented here, a proposal for a background model evaluation framework which allows for the comprehensive, objective and accurate analysis of model performance and which addresses many of the issues that currently exist, has been developed. The details of this proposal are presented in chapters 4 and 5.

Chapter 4 Proposed Evaluation Framework

In chapter 3, the current state of background model evaluation was examined as were the strengths and the limitations of the existing range of evaluations. Based on this research, a background model evaluation framework that will allow for the thorough and impartial assessment of model capabilities was designed. The use of this framework would ensure that the results of all evaluations are comparable and would allow for these results to be compiled into a comprehensive reference of performance data describing the capabilities of different models. Such a reference would be of great benefit to the research community to direct further development and to developers wishing to assess how their model compares to a wide range of existing models in a variety of scenarios. It would also be a useful aid in the selection of a background model for a particular purpose as suitable models could easily be identified.

The proposed background model evaluation framework takes over much of the work involved in assessing the performance of background models. Developers can submit their completed models to the framework via a website, along with information regarding the model and their contact details, to have it fairly assessed against other models using a standard evaluation methodology. It must be ensured that any background model that is submitted is compliant with the requirements of the framework in order for it to be eligible for testing (see section 4.1 for a description of these requirements). The submitted models are executed so as to obtain result frames for each video in the dataset while the associated information is stored for later use. The evaluation metrics - precision, recall, f-measure, average frame processing time, peak memory usage, training phase length and lag - are calculated, stored and published to the framework website to be used for various purposes. Evaluation results are grouped by category, e.g. the performance of models on videos depicting a gradual illumination change or with a dynamic background may be viewed. Models are ranked in terms of their accuracy within each category. An overview of the proposed background model evaluation framework is shown in Figure 3.1.12.

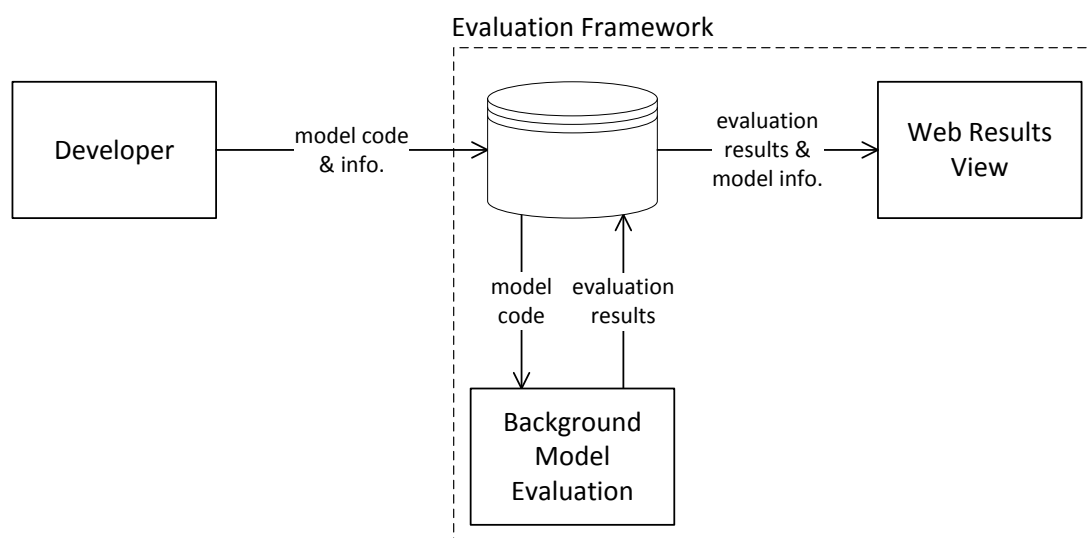


Figure 3.1.12 - Evaluation Framework

The various components of this system including the creation, submission and evaluation of a background model are described in detail in the remainder of this chapter. Also discussed is an alternative architecture that may be used in circumstances where model code cannot be submitted to the framework. The proposed evaluation methodology, i.e. the test videos, ground truth and evaluation metrics that should be used are described in the next chapter while a description of the implementation of the proposed framework is provided in chapter 6. A review of the advantages and the limitations of the proposed framework is presented in chapter 7.

4.1 Creating a Model

While the creation of a background model is carried out by a developer and is therefore not strictly part of the evaluation framework, it is essential that it be addressed in the proposal to ensure that models are developed in such a way that they will be compatible with the framework. To be compatible, a model implementation must be coded in C++ and make use of OpenCV⁵. In addition, all video frames are provided by the framework in the Mat format and the processed frames must be returned in this same format. To allow a model implementation to interface with the evaluation framework a simple C++ API is provided. The API functions are listed in Table 4.1-1. Developers must make use of these functions in their code to obtain the frames for each video and to return the processed frames.

API Function	Return Type	Description
<code>getNumVideos()</code>	<code>int</code>	Returns the number of videos contained in the dataset.
<code>getNumFrames(int video_num)</code>	<code>int</code>	Returns the number of frames in video number <video_num>.
<code>getFrame(int video_num, int frame_num)</code>	<code>cv::Mat</code>	Returns frame <frame_num> from video <video_num>.
<code>putFrame(cv::Mat, int video_num, int frame_num)</code>	<code>void</code>	Used to give processed frame <frame_num> from video <video_num> to evaluation framework.

Table 4.1-1 - API calls needed to process the dataset of the evaluation framework

The API has been made as simplistic as possible so that it will be easy for developers to integrate into their model and will require them to make minimal modifications to their code to make it compatible with the proposed framework.

Using the `getFrames` function, video frames may be requested as they are needed. This facilitates any degree of lag that the developer may decide to use in their model. Some models, for example, may take a frame, process it and immediately return the corresponding result frame while others may take a small number of frames before returning any processed frames to the framework. Such models, as was seen in chapter 2, work online. Others, meanwhile, may work offline and request all frames from a video before returning any results. The facilitation of these different types of

⁵ OpenCV version 2.3.1 has been tested thus far.

processing ensures that, no matter how a model operates, it will be accommodated by the proposed framework.

4.2 Submitting a Model

The submission of a background model for evaluation is completed through a form on the “Submit a Model” page of the framework website which is discussed in section 4.6. Through this form, developers must provide information regarding the model, their own contact information as well as the C++ code files containing the code of their created model. The information that is requested is outlined in Table 4.2-1.

Information	Required/Optional	Comments
<i>Model Information</i>		
Model Name	Required	N/A
Parameters	Required	Parameter name and value should be separated with a space. Different parameters should be separated by a semi-colon.
Website	Optional	N/A
Reference	Optional	Reference should be in the IEEE style.
<i>Developer Contact Information</i>		
First Name	Required	N/A
Last Name	Required	N/A
Email	Required	N/A
Affiliation	Optional	N/A
Model Code	Required	C++ code files should be submitted. Full projects are not acceptable.

Table 4.2-1 - Information requested by background model submission form

All information other than the code files must be submitted through form fields. If a required piece of information is omitted, a message indicating this will be displayed to the user. Similarly, some of the required information must be supplied in a specific format, e.g. email address. If it is not provided in the correct format, this will also be indicated to the user. A file upload service will be available allowing users to select the appropriate code files from their computer. Once the user has entered their information and confirmed submission, the information they have provided is supplied to the evaluation framework.

For each model that is submitted to the framework, the information about that model is inserted into the framework database as is the developer information if it is their first time to submit a model. The C++ code files are stored in an appropriate location and a reference to these files is entered into the database.

The submission of an executable version of the model in place of just the relevant code files was also considered. This would be more straightforward for the framework as it would be able to simply run the submitted executable rather than having to build it as is later discussed. The use of executable files is problematic, however, as executables differ based on the environment in which they are created and will thus only work on specific architectures. It is unreasonable to expect a developer to use a specific architecture to build their executable in order for it to work with the framework. It is

essential that the framework be as easy for developers to use as is possible so that they will be inclined to make use of it. As a result of these projected difficulties, it was determined that the most reasonable approach is to request that developers submit their model code files and to build the corresponding executables locally as part of evaluation.

4.3 Running a Submitted Model

To execute a submitted background model, the relevant code files are inserted into a preconfigured project. This project is automatically built into an executable. If problems are encountered in building the project due to the submitted model being incompatible with the framework, developers will be informed of the issue and will be provided with an explanation as to why their code was not compatible. Once the executable is successfully built, it is run so that the model can process each of the videos in the dataset and provide the framework with result frames that can be used in analysing the performance of the model. These result frames are stored until the framework is ready to evaluate them. This component of the evaluation framework operates in the background model evaluation block of Figure 3.1.12. The detail of this portion of the framework may be seen in Figure 4.3.1.

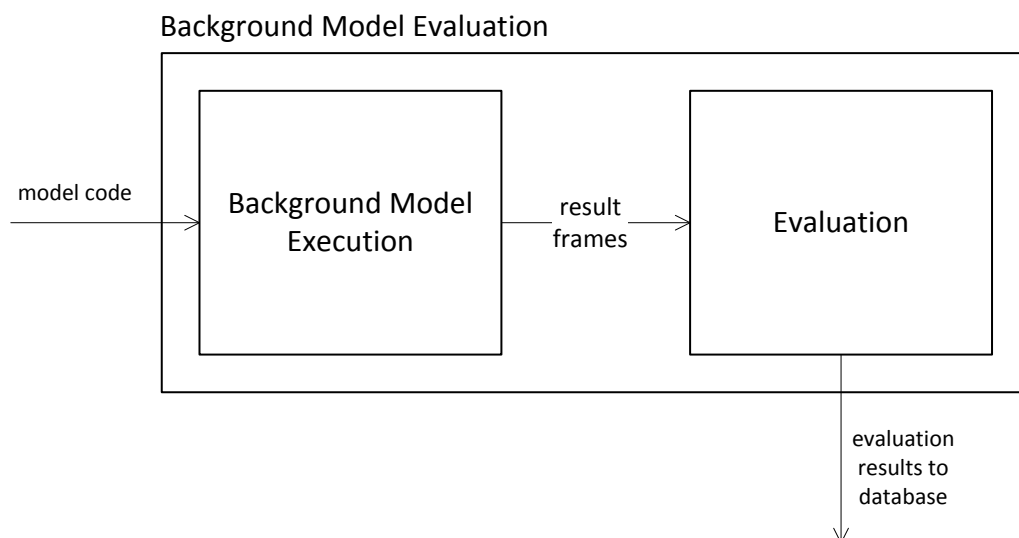


Figure 4.3.1 - Detail of background model evaluation block of Figure 3.1.12

4.4 Performance Evaluation

To assess the performance of submitted background models, several evaluation metrics are used. The metrics that have been selected are described below while a more detailed description of these metrics and a discussion of the reasons for which they were chosen are presented in chapter 5. The process of executing a background model and evaluating its performance on a single video is outlined in Figure 4.4.1.

- **Accuracy Metrics**

As processed video frames are returned from the executing background model for a given video, their accuracy is evaluated by comparing them to the corresponding ground truth

frames. The characteristics of the ground truth that is used are described in detail in chapter 5. For each video in the dataset, two sets of true positive, true negative, false positive and false negative counts are maintained – one for the case in which shadow is regarded as part of the background and the other for the case in which shadow is considered to be foreground. For each comparison of a processed frame and a ground truth frame these counts are updated. Once all frames of a video have been assessed, the accuracy metrics – precision, recall and f-measure – are computed twice using the two sets of total count values for that video and stored in the database for later use. The process of calculating these metrics may be seen in the purple box in the centre of Figure 4.4.1.

- **Average Frame Processing Time**

The amount of time for each video to be completely processed using a background model is measured and used to determine the average time that was required to process each video frame. The execution time over a single video is measured from the time at which the first frame is requested from the evaluation framework to the time at which the final processed frame is returned to the framework. The average time taken for each video frame to be processed is calculated and entered into the database. The green box on the left-hand side of Figure 4.4.1 depicts the calculation of this metric.

- **Peak Memory Usage**

The peak memory usage of the background models is also assessed for each video in the dataset. This information is stored in the database for later use. The calculation of a model's peak memory usage when processing a video is shown in the blue box on the left of Figure 4.4.1.

- **Training Phase Length**

The length of the training phase required by each background model is also determined. This is measured by counting the number of video frames that are requested from the framework before the first result frame is returned by the executing model. The length of the training phase required for each video is recorded in the database for later reference. The measurement of this metric is illustrated in the red box on the right of Figure 4.4.1.

- **Lag**

The lag of the background models is also measured. The amount of lag may vary throughout the processing of a video and thus, the worst case or the maximum observed lag is recorded as the final lag value for each model and video combination. This is stored in the database for later retrieval. The determination of the lag metric is shown in the grey box on the right of Figure 4.4.1.

The calculation of these evaluation metrics provides a thorough assessment of the various aspects of background model performance. The choice of evaluation metrics is discussed in detail in chapter 5.

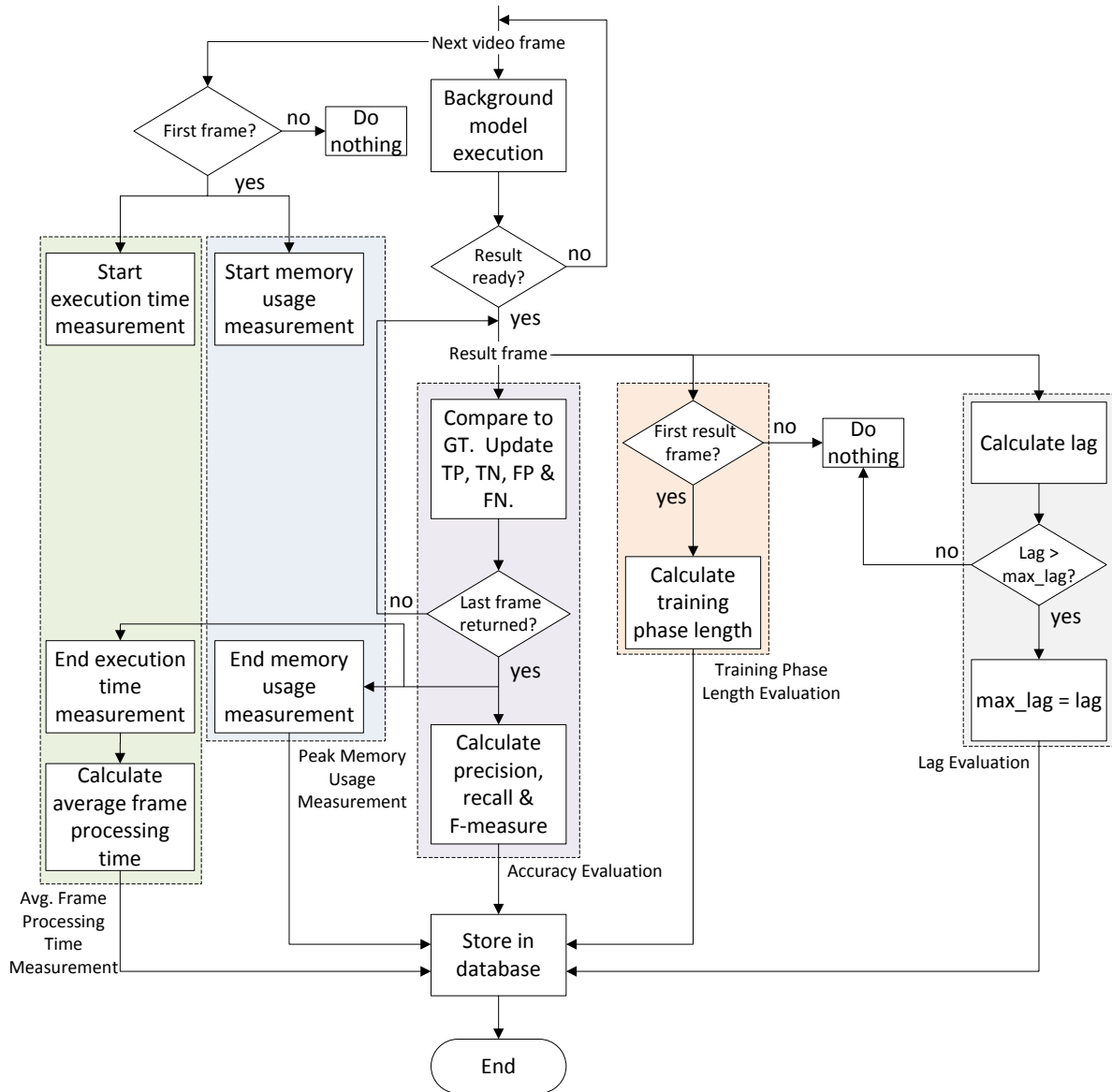


Figure 4.4.1- Evaluating background model performance for a single video

4.5 Publishing Evaluation Results

Once the assessment of a background model's performance is complete, the evaluation results, i.e. the outcomes of calculating the metrics described in section 4.4, are entered into the database and displayed on the results page of the framework website (see section 4.6). This results page displays the evaluation results for all submitted models grouped by category, e.g. performance in the dynamic background category alone may be viewed. The included categories are discussed in chapter 5. Within each category, the average results achieved by each model across all videos in that category are displayed for both shadow considerations.

In addition to this overall review of model performance, average evaluation results in each category for a single model can be viewed. From this, the scenarios in which a background model performs best may easily be seen.

4.6 Proposed Framework Website

The framework website forms the centrepiece of the system. It is what users see and interact with when they wish to submit a model for evaluation, view previous evaluation results, download the dataset etc. The website provides information regarding the framework including a description of the dataset and the metrics that are used and the process by which the evaluation is carried out. The structure of the website is simple as is depicted in Figure 4.6.1. This, combined with a straightforward and intuitive interface design, ensures that users can easily navigate through the site content and quickly find the page that they need. The implementation process and initial appearance of this website are described and shown in chapter 6.

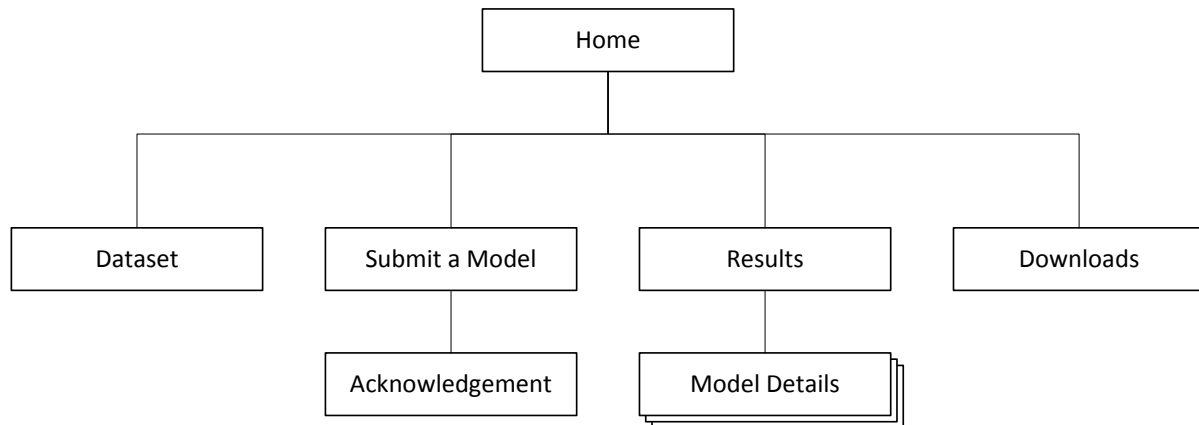


Figure 4.6.1 - Website structure

The content of the various webpages shown in Figure 4.6.1 is briefly described below:

- **Home** – This page briefly describes the concept of background subtraction, the need for background model evaluation, the current state of evaluation and the benefits provided by the proposed framework. It is the first page that users see when they view the website.
- **Dataset** – This page provides a description of the dataset that is used including the challenges depicted and the characteristics of the accompanying ground truth. These are also described in chapter 5.
- **Submit a Model** – This page presents users with a form which they can use to submit a background model that they have developed for evaluation. Users must provide information such as the name of the model, the parameters that are used and their contact details as well as the model code files.
- **Acknowledgement** – This page acknowledges the submission of a background model for evaluation and directs users to the results page to view the results of their model's evaluation once they are available.
- **Results** – This page details the results of evaluating the submitted models for each challenge that is depicted in the dataset as discussed in sections 4.5 and 6.3. The dataset and the scenarios that are depicted are described in detail in chapter 5.

- **Model Details** – This page is reached by selecting a background model on the results page. Information regarding the model and contact details for the developers are provided as are the evaluation results from each dataset category for the model.
- **Downloads** – This page provides the dataset and the evaluation framework API for download. The alternative, local framework that is described in section 4.9 is also provided.

4.7 Database

The database of the evaluation framework contains information regarding the background models that are submitted and the developers of the models. This information is obtained via the submission form on the “Submit a Model” page of the website. The results of the evaluations are also contained within the database. In addition, it contains details of the dataset that is used for the evaluation including the scenarios that are depicted, the video sources and the number of frames in the videos.

4.8 Re-evaluation of Background Models

It was seen in section 3.1.6 that it may sometimes be necessary to update the evaluation dataset or metrics. As the model code is provided to and executed by the framework, such updates are very straightforward to manage and results can be kept consistent. In the event of it being necessary to update the evaluation methodology, the process of model execution and evaluation described in sections 4.3 and 4.4 is simply repeated for each model that has previously been submitted using the new data, e.g. additional test videos, and the newly obtained results are added to those already present in the database. This ease of update is a significant improvement over the previously conducted evaluations which could not achieve this without significant effort and disruption.

4.9 Alternative Framework

There may, unfortunately, be circumstances in which a developer does not wish to submit their background model code to the proposed framework, e.g. if the model is patented. They may, however, still wish to avail of the benefits of having a thorough evaluation carried out so that they can assess how well their model is performing in comparison to others. Such information is essential for directing further development and is expected in any literature that is reported regarding the model. To facilitate this, a downloadable version of the framework will be available which can be used to evaluate models locally. The results of such an evaluation may then be compared to those published on the framework website. In addition, the locally obtained results may be submitted to the framework for publication on the associated website but these will be marked as locally evaluated to indicate that accuracy cannot be guaranteed. In the event of an update to the evaluation methodology these results will be removed from the site as there is no way to rerun the model and obtain updated results. Developers who intend to submit their code may also wish to use this local version in order to test that they have correctly adapted their model to be compatible with the framework. The alternative downloadable version of the framework is available on the downloads page of the website and is depicted in Figure 4.9.1.

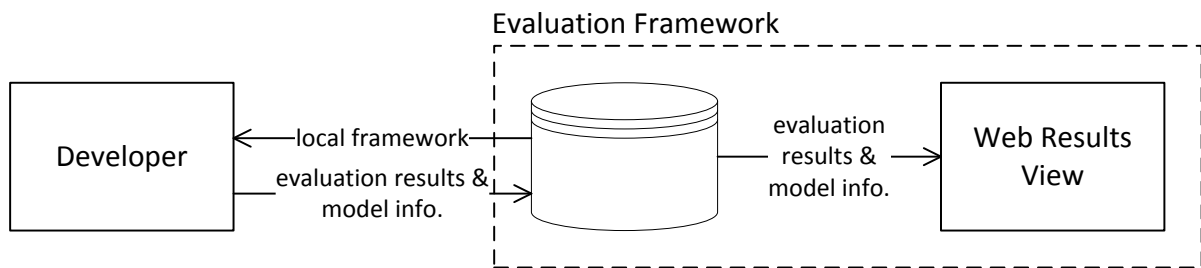


Figure 4.9.1 - Alternative evaluation framework

4.10 Overview of Proposed Evaluation Framework

This chapter has presented a proposal for a background model evaluation framework that addresses many of the issues that exist with the way in which evaluations have so far been carried out (see section 3.1.7). The framework is capable of fairly and objectively assessing background models in a manner which allows them to be meaningfully compared. It has been designed to be very straightforward for developers to interact with and to be scalable. It provides the facility for a large scale background model evaluation to be performed and for an extensive collection of evaluation results to be compiled and made publically available for future use. The proposed framework has a number of advantages over the attempts that have previously been made including the guarantee of fair results, ease of update and a significant reduction in developer workload. These advantages are described in detail in section 7.2. The evaluation methodology that should be used in accompaniment to this framework so that model performance may be comprehensively evaluated is described in chapter 5 while a proof of concept implementation and an assessment of the feasibility of the proposal are provided in chapters 6 and 7 respectively.

Chapter 5 Proposed Evaluation Methodology

In the previous chapter, a proposal for a background model evaluation framework which would allow for the thorough and impartial assessment of background model capabilities was presented. By using this framework it can be ensured that all evaluations are carried out in the same manner and thus that evaluation results are fair and trustworthy. It enables the compilation of a comprehensive reference of background model performance data which would describe the capabilities of various models. As mentioned, a resource like this would be of significant benefit to the research community, to model developers and to those who require assistance in selecting an appropriate model for a particular application. Chapter 4 considered the structure and operation of the proposed framework including an overview of the model submission, execution and evaluation processes, the publication of evaluation results and the framework website. This chapter, meanwhile, considers the evaluation methodology that should be used in assessing the abilities of submitted models including the dataset and ground truth to be used, the evaluation metrics that have been selected as well as the way in which model parameter values should be determined. A detailed description of the proposed evaluation methodology including why various decisions were made is provided here while a precise specification of the proposed methodology is presented in Appendix A.

5.1 Evaluation Dataset

In order to perform a comprehensive background model evaluation, the results of which are reliable and demonstrative of the capabilities of the models that are tested, it is essential that a large and diverse video dataset which depicts a wide range of indoor and outdoor challenges is used. The videos of the dataset should depict real scenes as these will more faithfully represent the scenarios that a background model will encounter in use and should vary from a few minutes to many hours in length in order to fully analyse the models' performance and reliability. For example, a twenty-four hour long video depicting natural illumination changes throughout a day must be included. Without a comprehensive video dataset the true strengths and limitations of a background model cannot be fully or accurately assessed. The remainder of this section considers the size of the required dataset and the challenges that it should depict.

5.1.1 Video Categories

In chapter 3 the challenges that are currently being considered in background model evaluations were examined. From this, it was found that, while some important challenges are being given considerable attention, other equally important and commonly encountered challenges are being overlooked. No previous evaluation has considered a broad enough range of challenges for comprehensive evaluation to be possible. This is a major weakness of the evaluations that have been carried out to date. The proposed evaluation methodology strives to ensure that all of the most pertinent and commonplace challenges are considered so that a full appreciation of the capabilities of the models that are tested may be gained. The challenge categories in which a model should be tested were identified and are listed, along with a brief description, in Table 5.1-1.

Challenge Category	Description
Simple moving object	Foreground objects exhibiting simple motion in a scene. This is inherently part of all videos to be used and provides a basic indication of a model's abilities.
Basic sequence	A variety of challenges typical of those from other categories allowing for a general overview of a model's capabilities to be obtained. This category serves as a general reference.
Objects becoming still	Objects moving in the scene which, at some point, become still.
People becoming still	People moving in the scene who, at some point, become still.
Part of the background begins to move	Objects in the background of a scene begin to move. In addition to a moved object becoming part of the foreground, the area revealed when the object began to move will also appear as foreground.
Intermittent object motion	Objects moving in the scene which stop for a short period of time and then begin to move again, e.g. a car stopping at traffic lights and moving off soon after or, alternatively, stationary objects begin to move and then stop again. This type of scenario may cause ghost artefacts in the background model.
Bootstrapping	No initialisation frames that are free of foreground objects are available meaning that the background model has no opportunity to create a background image using solely background data.
Dynamic background	Uninteresting background motion which may be periodic or irregular. This motion may be introduced to a scene in several ways including a tree waving in the wind, moving water and changing traffic lights.
Shadows	A range of strong and faint shadows of varying sizes present in scenes. Some are static and cast by objects such as buildings while others are cast by moving objects such as trees and people.
Gradual illumination change	The illumination of the scene changes gradually causing slow, widespread changes in the appearance of the background over time. Such changes are often a result of the natural variation of light throughout a day.
Sudden illumination change	The illumination of the scene changes suddenly causing a significant and quick change in the appearance of the scene background. Such a change may be caused by a light being switched on or off.
Precipitation	The presence of precipitation in a scene can greatly change its appearance by introducing noise, darkening the scene and changing the colour of parts of the scene, e.g. concrete becoming wet.
Door opening/closing	A door opening or closing in a scene will cause significant changes. Both the door and the area revealed by its movement will appear as foreground.
Noise	The presence of noise in a video corrupts the actual appearance of the scene. The amount of noise present can vary as a result of the camera quality and other environmental factors.
Camouflage/low contrast	The appearance of foreground objects in the scene is similar to that of the scene background making them more challenging to distinguish.
Camera shaking	Videos are captured with an unsteady camera causing the scene to shake. The amount of shaking that is seen varies between videos.
Reflections	Reflections of various static and moving objects are present in scenes.
Video compression	The compression of videos can introduce compression artefacts which corrupt

	the actual appearance of the scene.
Foreground aperture	Objects of a single colour are in motion in the scene. The uniformity of the objects' colour may cause interior object pixels to appear stationary.
Objects with some moving and some static parts	If only some parts of an object are moving the entire object may not be detected as foreground.

Table 5.1-1 - Challenges to be depicted in standard evaluation dataset

The videos of the dataset are grouped into these categories based on the type of challenges that they depict. The challenge or scenario being considered, however, will not necessarily be the only challenge depicted in a video. Often, a video sequence will contain one dominant challenge accompanied by one or more minor challenges, e.g. a video in which the dominant challenge is a sudden illumination change may also contain a small number of moving shadow pixels. This video would be classified as being solely in the sudden illumination change category despite the presence of shadow pixels. The existence of minor challenges such as these shadow pixels in addition to the dominant one will result in some pollution of the results obtained using this video. It may also happen that there is no single dominant challenge in a video but, instead, that there are multiple major challenges. In this situation, the video would be considered to be part of each of the categories corresponding to these major challenges. As before, however, this will cause result pollution.

While it is intended that the challenges listed here will cover all those that a background model may be faced with, it is possible that, in the future, additional scenarios which need to be considered will present themselves. As was mentioned previously, the existing attempts at developing an evaluation framework [4] [5] do not straightforwardly allow for additional categories to be added or for videos to be added to existing categories without making results inconsistent. This is a major drawback of these existing frameworks which is addressed by the proposed system. The way in which this is addressed is described in section 4.8.

5.1.2 Dataset Size

Given the diversity that is required of it, the dataset that is used in evaluating background models must be quite large. It is necessary that each of the challenge categories listed in Table 5.1-1 be represented by multiple videos so that variations on the challenges may be examined to make for a more robust and reliable evaluation. This is also important in ensuring that the effects of result pollution from secondary challenges in the videos are minimised. As some videos may be considered to be in multiple categories, the exact size of the dataset that should be used is difficult to define. Instead, a minimum category size is defined. Each of the categories listed, as well as any which may be added in the future, must contain a minimum of six videos.

5.1.3 Dataset Overview

In summary, the video dataset that should be used in the evaluation of background models must contain real videos which are diverse in terms of quality and length. At least six videos must depict each of the challenge categories listed in Table 5.1-1 so that the performance of background models' in dealing with these challenges may be reliably examined. The creation of a large, comprehensive and diverse dataset such as this would be of enormous benefit to the research community as it would allow for an extensive and objective background model performance evaluation to be carried out.

5.2 Ground Truth

As described previously, ground truth is the standard against which the accuracy of background subtraction results is assessed. As a result, it is an extremely important aspect of background model evaluation. It is important that the ground truth that is used in an evaluation is reliable as, otherwise, evaluation results cannot be held in confidence and will not truly reflect the capabilities of the background models that are considered. It was seen in chapter 3 that there are several issues associated with the ground truth that has been used in evaluations to date. Described in this section are the characteristics of the ground truth that should be used in order to ensure a comprehensive, fair and reliable evaluation of background models.

5.2.1 Ground Truth Type

The first decision to be made regarding ground truth is the type that is to be used. In chapter 3 the two main types of ground truth were described – pixel-based in which each pixel in a frame is given a label to describe its status (static, moving etc.) and bounding boxes which involves drawing a box around the objects of interest in a scene. Due to the pixel-based nature of the results produced in background subtraction, pixel-based ground truth is more suited for use in this type of evaluation and is thus the form to be used with the evaluation framework that has been proposed. As all videos to be used depict real scenes there is no way to fully automate the creation of ground truth and thus it must be created manually.

5.2.2 Ground Truth Classifications

As discussed in chapters 2 and 3, there is currently no consensus as to what should be considered foreground in a scene and what background. There is therefore much dispute regarding what exactly is to be evaluated when assessing the performance of a background model. There is no single correct answer to this issue and thus, as was seen previously, different assumptions are made in different evaluations. In order to fairly evaluate models in a manner that allows them to be meaningfully compared, a standard classification procedure must be established. This will ensure that the ground truth that is used will be a reasonable and universally accepted representation of the corresponding scene. Described here is a proposal of such a standard.

The ground truth that should be used for a background model evaluation should, for the most part, designate static pixels as part of the scene background and moving pixels as part of the foreground. It is important, however, that pixels are not blindly labelled in this manner as there are a number of exceptions to these classifications which should first be addressed to give consideration to what is actually depicted in the scene. This is essential as, moving objects, though often considered to be, are not always of interest. By acknowledging this, the ground truth will be more useful and a more realistic representation of the scene than if classification had been based on motion alone. The proposed approach to labelling pixels in ground truth frames is outlined below:

- Uninteresting motion such as the waving of trees in the wind should be regarded as part of the background.
- People should always be considered part of a scene foreground regardless of their motion status.

- Other moving objects should be regarded as foreground. If such objects come to a stop, they should remain in the foreground for fifty frames after becoming still at which point they become part of the background.
- Shadows cast by background objects should also be considered as being in the background regardless of whether the shadows are static or dynamic.
- Shadows cast by foreground objects should be separately labelled as shadows to distinguish them from the foreground and the background. As seen previously, it is important in many applications for shadows to be ignored and considered part of the background while in others they are regarded as foreground. In yet other applications, it is necessary that shadows be distinguished from other objects in the scene. By creating a dedicated shadow label the performance of background models in handling shadows in each of these three scenarios can be evaluated.
- Pixels which are difficult to classify with certainty, such as those close to the boundaries of foreground objects (and shadows where applicable) or in areas of motion blur, should also be distinguished from the scene background and foreground as well as from shadows.

Based on these classification guidelines and the ground truth labels that are used by Goyette et al. [4] for the ChangeDetection.net dataset [14], a set of four ground truth labels is proposed. The colours assigned to the ChangeDetection.net labels are retained so that their videos and ground truth may be used as part of the proposed evaluation methodology. The labels that have been defined as well as their associated greyscale colour values are given in Table 5.2-1. The use of these labels is also summarised.

Ground Truth Label	Greyscale Value	Description
Background	0	Used for pixels which can be classified as background with certainty.
Foreground	255	Used for pixels which can be classified as foreground with certainty. Pixels given the foreground label may or may not be moving, e.g. a person who becomes still is no longer moving but should still be classed as foreground.
Shadow	50	Used for pixels which can be classified as shadow with certainty.
Unknown	170	Used for pixels whose status is unclear such as at moving object boundaries and in areas of motion blur. By using this label to indicate uncertainty in pixel status such pixels can be ignored in evaluation ensuring that uncertainty in the ground truth will not corrupt the results of the evaluations.

Table 5.2-1 - Proposed ground truth labels

5.2.3 Ground Truth Volume

It was seen in chapter 3 that, in the background model evaluations that have previously been carried out, varying amounts of ground truth has been used. To ensure that the evaluations are as reliable,

comprehensive and illustrative of the models' performance as is possible, ground truth should be available for every frame of each of the videos in the dataset.

5.2.4 Ground Truth Accuracy

As previously discussed, there is significant scope for error in the creation of ground truth due to the manual nature in which it is made. It is important that this be addressed as, otherwise, background models may be penalised not only for their own misclassifications but also for those introduced by the ground truth creator resulting in imprecise evaluations. While it is not realistic to assume that all misclassifications can be avoided, measures must be put in place to minimise their presence and their effect. As part of the proposed evaluation methodology, two means of reducing these misclassifications are to be used. The first of these is the use of the unknown label which was described above for pixels whose status is unclear. By doing this, the need for estimating the status of ambiguous pixels is avoided. Pixels given the unknown label may simply be ignored during the calculation of evaluation metrics so that they will not impact on the outcomes of the evaluations.

Additionally, to improve ground truth accuracy, multiple different people should create ground truth for every frame in the dataset so that their efforts may be combined to obtain a consensus as to what the final, accepted ground truth should be. As was described in section 3.1.2.2, this can significantly reduce the occurrence of misclassifications. Ground truth should be created for each frame by a minimum of five different people.

5.2.5 Ground Truth Overview

In summary, the ground truth that is to be used with the proposed evaluation framework must be pixel-based. It must be created for each video frame in the dataset and each pixel in the ground truth must be assigned one of four possible labels – background, foreground, shadow or unknown. At all times people are in the foreground regardless of their motion status. Other foreground objects become part of the background once they have stopped for a period of fifty frames. Both the unknown label and the practice of obtaining ground truth through a consensus of at least five interpretations, serve to improve ground truth reliability. The creation of ground truth for a dataset of the scale described here would require an enormous amount of work and time but, without such a resource, it is not possible to comprehensively evaluate background model capabilities and limitations. This work is critical to the success of background model evaluation and the significant inadequacies that currently exist regarding it is one of the major deficiencies of the evaluations that have previously been reported. A potential approach to the creation of the large volume of necessary ground truth is the use of crowdsourcing in the form of a tool similar to the LabelMe annotation tool [56] that was previously mentioned. In doing this, the community could collaborate to produce an extensive and essential resource rather than putting their efforts into creating their own small amounts of inconsistent and often unreliable ground truth. This resource can then be used in the evaluation of all background models so that their performance may be meaningfully compared.

5.3 Evaluation Metrics

It was seen in chapter 3, that evaluation metrics are required to quantitatively assess the performance of background models and to allow their capabilities to be compared. There are a number of aspects of model performance which may be assessed but, from analysing the previously

reported evaluations, it was found that no evaluation to date has given consideration to all of these aspects. Thus these evaluations have not been able to gain a full appreciation of the capabilities of the models that they evaluate. This section presents the evaluation metrics which should be used to ensure that model performance is thoroughly assessed.

5.3.1 Accuracy Metrics

One important aspect of background model performance that must be considered is accuracy or how similar the result frames produced are to the corresponding ground truth frames. It was seen previously that all of the reviewed evaluations consider model accuracy but that it is measured in many different ways. Based on what was learned from the analysis presented in chapter 3, three accuracy metrics – precision, recall and f-measure – have been selected for use in comparing result and ground truth frames. In these comparisons, the areas labelled as unknown in the ground truth will not be considered. The use of these three metrics allow for a comprehensive assessment of background model accuracy to be achieved.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Measure = 2 \frac{Pr \cdot Re}{Pr + Re}$$

It was mentioned previously that the value of the precision metric should be as close to its maximum value of one as possible. As precision is a measure of the proportion of pixels correctly classified as foreground of all pixels given the foreground classification, a high value would indicate a low number of false positive classifications. While this is a desirable outcome, it does not give any consideration to the number of false negative classifications that are present in the result meaning that, if parts of the foreground have been lost, this will not be indicated by the precision metric. Thus, while precision is a good indicator of the quality of a model's foreground classifications, this metric alone is an insufficient measure of how well a model has performed.

As with precision, the recall metric should ideally have a value close to its maximum of one. Recall is the proportion of all actual foreground pixels that are detected and a high value would be indicative of a low number of false negatives, i.e. the recall metric measures the amount of the scene foreground that has been detected. No consideration is given, however, to what portion of the foreground classifications were accurate. Recall is a good indication of how well a model can identify the foreground pixels in a frame but, as it ignores the amount of false positives that are present, excessive noise will not be recognised. Like precision, recall alone is not a sufficient metric for use in assessing the performance of background models.

Although neither precision nor recall alone is a sufficient evaluation metric they complement one another and account for what the other does not consider, i.e. precision considers the presence of false positives but does not acknowledge false negatives while recall considers the opposite. The use of both precision and recall as evaluation metrics provides sufficient information to comprehensively assess model accuracy but does not, however, allow effective ranking of the

models. As explained by White et al. [45], “a measure of quality must be established that can quantify how similar a resulting subtraction frame is to the ground truth for a frame in one scalar value” [45]. A single value is required so that model accuracy can “unambiguously and automatically” [45] be compared and ranked against one another. To take advantage of both the precision and recall metrics, they may be combined into a third metric, f-measure. F-measure is a harmonic mean of recall and precision and is essentially an expression of these two metrics as a single value which may be used for result comparisons. The F-measure metric weights precision and recall equally and allows both to be taken into consideration in determining a background model’s accuracy.

The precision, recall and f-measure metrics must be calculated for each background model for each video in each category in the dataset. To calculate these for a particular model and video combination, each result frame produced by the model for that video should be compared to the corresponding ground truth frame and the numbers of true positives, false positives, true negatives and false negatives when shadow is considered in each background and foreground counted. A running total of each of these quantities should be maintained and, after all frames have been considered, these totals should be used to calculate precision, recall and f-measure for that video. The results of these calculations illustrate the accuracy of the model in processing that video. The precision of the results obtained by a background model for video v in category c , for example, may be calculated as:

$$Pr_{v,c} = \frac{TP_{v,c}}{TP_{v,c} + FP_{v,c}}$$

It was discussed previously that, in order to get a general overview of the accuracy of a model when presented with a challenge, it is necessary to test the model using multiple videos which depict variations on that challenge. To determine the accuracy of a background model on a particular challenge, the precision, recall and f-measure values achieved by that model for each of the videos which depict the challenge should be averaged. The average precision of a background model over all videos in category c , for example, is calculated as:

$$Pr_c = \frac{1}{N_c} \sum_{v=0}^{N_c-1} Pr_{v,c}$$

By using precision, recall and f-measure as the metrics with which to evaluate the accuracy of background models it is ensured that a comprehensive assessment of their abilities may be achieved. If all models are evaluated using these metrics they may be fairly and effectively compared. This will allow the true capabilities of the models to be seen and will aid in further research and in the selection of an appropriate model for a given application.

5.3.2 Efficiency Metrics

Efficiency is also an important aspect of background model performance which should be assessed. A model’s efficiency dictates the applications in which it may be used and thus, an assessment of a model’s performance cannot be complete without giving consideration to it. In measuring a model’s efficiency, the factors to be considered are its processing speed and its memory usage. These can be difficult to measure as they vary based on video characteristics such as length and resolution. It is

therefore necessary that a standard measurement method be defined so that models can be compared in terms of this aspect of their performance.

To measure the speed of a background model in a standard and comparable manner the average time that is required to process a single frame should be calculated. To calculate this value, the total time taken to process a video should be measured and divided by the number of frames in that video. For example, the average frame processing time for a video, by a background model is given as:

$$\text{avg. frame processing time} = \frac{\text{total processing time}}{\text{num. frames}}$$

This measure should be calculated for each background model for each video in the dataset and averaged so that the effects of video characteristics such as frame resolution may be minimised.

When measuring a model's memory usage it is the peak usage that should be considered. This may be measured by sampling the amount of memory being used by the model at discrete intervals in time for each video that is processed and keeping track of the highest observed usage. This should be measured in the processing of each video in the dataset.

5.3.3 Training Phase Length

The length of a model's training phase should also be measured to provide additional insights to its performance. This aspect of a model's performance can influence the applications in which it may be used and thus, it is important that it be considered. To measure this metric, the number of frames used by a model in creating an initial background image must be counted.

5.3.4 Lag

The final evaluation metric that should be considered is the lag of the background models that are being assessed. Lag is essentially a measure of how many frames a model takes between taking frame x and returning a result for frame x . A model's lag is an extremely important aspect of its performance but, at present, it is not being considered in evaluation. As has previously been discussed, the lag of a model dictates the applications that it may be used in and, thus, without this information, it can be very difficult to select an appropriate model for an application. It is essential, therefore, that this metric be computed. With some models, the degree of lag that is present can vary. Due to this, the worst case lag exhibited by each model, when processing each video, should be recorded.

5.3.5 Evaluation Metric Overview

To ensure that all aspects of model performance are assessed a variety of evaluation metrics must be used. Precision and recall should be used to thoroughly assess the accuracy of the background models that are considered and, to enable a straightforward comparison between models, these should be combined to calculate the f-measure metric. To assess efficiency, a model's average frame processing time and peak memory usage should be monitored. In addition, the length of a model's training phase and its worst case lag should be measured. The use of these metrics will allow for a thorough appreciation of model capabilities to be obtained which is essential both in the use of background models and in future research concerning them.

5.4 Parameter Value Determination

Background model developers have two options to choose from when tuning the parameter values of their models. They may opt to manually tune parameters or, alternatively, to have their model auto-tune them. As described in section 3.1.5, the automatic tuning of parameter values allows for far more accurate results to be obtained than is typically achievable with manual tuning. In addition, a single set of parameter values must be used for the entire evaluation as it is not practical or realistic to modify them for every video with which the model is presented. The dataset videos are available for download on the framework website so that they may be used in the tuning process. The associated ground truth, however, may not be used.

5.5 Post-Processing

Developers may choose to make use of post-processing techniques such as opening to remove excessive noise and closing to fill holes in objects, in order to improve the accuracy of the result frames that are returned from the background subtraction process. Any post-processing methods that are included in the developer submitted code as part of their model are acceptable. The evaluation framework will apply no further post-processing and will instead evaluate result frames as they are returned from the submitted models.

5.6 Overview of Proposed Evaluation Methodology

This chapter has presented a proposal for a background model evaluation methodology that addresses many of the limitations of the existing body of evaluations that were discussed in chapter 3. By using this methodology in addition to the framework proposed in chapter 4, the capabilities of background models may be thoroughly and objectively assessed. It was shown previously that such comprehensive and impartial assessment has not been performed to date and that the resources for doing so do not yet exist. The proposals that have been made in this and in the previous chapter provide a solution to this. The implementation of these proposals would be of enormous benefit to the computer vision community as it would finally be provided with a rigorous and credible facility for comprehensive background model performance assessment and ranking. Chapter 6 describes the implementation of a proof of concept version of this system while chapter 7 considers the success and the remaining limitations of the evaluation framework and methodology that have been proposed.

Chapter 6 Implementation

This chapter looks at how the background model evaluation framework and methodology proposed in chapters 4 and 5 was implemented. It should be noted, however, that, due to time constraints, it was not possible to fully implement the proposed framework and thus, a smaller scale, proof of concept was implemented. Among the topics discussed in this chapter are the technologies that were used in implementation, the challenges that were encountered and the ways in which these challenges were addressed. There are three major aspects of the proposed framework to be implemented. First is the evaluation aspect itself which involves running a model and assessing how well it performs. This portion of the framework was implemented in C++ and makes use of the open source computer vision library, OpenCV [63]. The second aspect is the framework website which facilitates the submission of background models, the provision of information regarding the framework and the presentation of evaluation results. This involved the use of a combination of PHP, HTML, CSS and JavaScript. The final aspect of the proposed framework is a database which stores information regarding the submitted background models, developers and the evaluation dataset as well as the evaluation results. The evaluation and the web aspects of the framework essentially communicate through the database. All database queries, insertions and updates are performed using SQL via C++ and PHP. A description of how the various components of the framework were implemented is given below. Also discussed are the general software characteristics that were desired in the implementation.

6.1 Software Requirements

It was desired that the implementation of the proposed evaluation framework exhibit a number of characteristics – extensibility, robustness, scalability and usability. These characteristics are of great importance in the implementation of this framework as well as in software in general and were therefore kept in mind throughout the implementation process. Each of the desired characteristics is briefly described below and a review of how well the finished framework exhibits these characteristics is presented in chapter 7.

6.1.1 Extensibility

Extensibility refers to the ease with which the framework may be extended by way of modifying its existing functionality or adding new functionality. It should be possible to extend the system without significant disruption and without causing harm to the existing functionality. It is important that the background model evaluation framework exhibit this characteristic so that the dataset may be updated and to allow for the possibility that it may, at some point, be desired to add to the framework's functionality.

6.1.2 Robustness

The robustness of a computer system refers to its ability to handle abnormalities such as unexpected input and execution errors. The better a system is able to cope with abnormalities like these the more robust it is considered to be. Robustness of the background model evaluation framework is essential in ensuring that evaluations may be carried out correctly and that any issues which may arise will not severely affect the performance of the framework. It is of particular importance as the

framework is required to work with a large amount of user provided data which has the potential to negatively impact upon its performance.

6.1.3 Scalability

A system's scalability refers to its ability to cope with increasing amounts of work or to be expanded to cope with an increased workload. A system that is capable of this is commonly referred to as a scalable system. The background model evaluation framework must be scalable to ensure that it will be able to handle the evaluation of large numbers of background models.

6.1.4 Usability

The usability of a system refers to how easily it may be used. A system should be simple and intuitive and it should be ensured that there is no significant learning curve that must be overcome in order for it to be used.

6.2 Background Model Evaluation

This section considers the implementation of the evaluation aspect of the proposed framework including the creation of the C++ API that is used to interact with the framework, the implementation of background models and their adaptation to make use of the framework API and the execution and evaluation of submitted background models. Other aspects of the framework such as model submission, database communications and result publication are discussed later in this chapter. As part of the implementation, it was necessary to compile a test dataset with corresponding ground truth. Due to time constraints it was not possible for this to be as comprehensive as is necessary to carry out a complete evaluation of model capabilities. The videos and some ground truth were gathered from a variety of sources. Other ground truth was created for the dataset. The videos and ground truth that were used alongside the proof of concept evaluation framework implementation are discussed further in chapter 7.

6.2.1 API Creation

The main aim in creating the framework API was to provide all functionality that would be necessary to allow a model to interact with the framework in as straightforward a manner as possible. After careful consideration it was determined that the minimal required functionality of the API was to provide video frames to the background models and to facilitate the return of processed frames. In addition, it was deemed necessary to provide a method of determining the number of videos that are contained in the dataset and the number of frames in any video. To provide this functionality, four API functions were created - `getNumVideos`, `getNumFrames`, `getFrame` and `putFrame`. These API functions are described in terms of their purpose, type and parameters in Table 4.1-1.

The API functions to obtain the number of videos in the dataset and the number of frames in a video are reasonably straightforward and operate by simply querying the framework database for the relevant information. This interaction with the framework database is discussed in section 6.4.2. The functions to obtain frames from and return processed frames to the framework are somewhat more complex. The `getFrame` function uses the base path to where the dataset is stored and the desired frame information to construct the path to that frame and fetches it to return to the background model. Similarly, the `putFrame` function constructs the path to the appropriate location for the processed frames to be stored until the framework is ready to look at them and

writes the supplied result frame to this location. The creation of the API functions was completed using C++, OpenCV, SQL and the SQLAPI++ database connector [64] which is discussed in section 6.4.2.

To provide access to these functions for all background models they were packaged as a static library from which they may be referenced. Some difficulties were encountered in creating this library due to the inclusion of the OpenCV library. This was troublesome to overcome due to the limited availability of clear documentation but through persistent efforts the difficulties were resolved by modifying the configuration of the static library creation project.

6.2.2 Background Model Implementation and Adaptation

To test the framework it was necessary to implement some background models and adapt them to interact with it using the associated API. Model implementations were obtained from a publicly available background subtraction library known as BGSLibrary [65] [66]. This library currently contains thirty-four different model implementations. Six of these models were adapted to use the framework API and this was found to be a very straightforward process. Some evaluation results obtained using these models with the evaluation framework are presented in chapter 7.

6.2.3 Automatic Background Model Execution

As was described in chapter 4, the code for all background models that are submitted to the framework is stored and, when a model is to be evaluated, either for the first time or in the event of an update to the evaluation methodology, the relevant code files for that model are inserted into a preconfigured project. A batch file then uses MSBuild [67], a build platform created by Microsoft, to automatically build the application and runs the resulting executable.

This was somewhat troublesome to achieve as, initially, the executable that was produced by building the project did not behave as expected. Result frames were not being written to the appropriate location and thus the framework had no processed video frames to evaluate. This was found to be a logical issue which, once isolated, was easily resolved.

6.2.4 Result Evaluation

It was seen in chapter 5 that there are a number of evaluation metrics that must be calculated in order for background model performance to be thoroughly assessed. This section considers the way in which these metrics are calculated in the proof of concept framework implementation.

6.2.4.1 Accuracy Metrics

The accuracy of a background model is assessed by comparing the result frames that are created using it to the corresponding ground truth frames and measuring their similarity. An integral part of this is the determination of the number of true positives, true negatives, false positives and false negatives that are present in each result frame and maintaining counts of these for each video in the dataset. It was mentioned in chapter 4 that two separate sets of these counts are maintained for each model for each video. One for when shadow is to be considered as part of the background and the other for when it is considered as foreground. This is facilitated by the shadow ground truth classification that was discussed in chapter 5. For each result frame/ground truth comparison, the values of each corresponding pair of pixels determines whether the result pixel is a true positive, true negative, false positive or false negative. The way in which these counts are updated is described in Table 6.2-1.

		Shadow as background				Shadow as foreground			
GT Pixel	Result Pixel	TP	TN	FP	FN	TP	TN	FP	FN
fg	fg	++				++			
fg	bg				++				++
bg	fg			++				++	
bg	bg		++				++		
s	fg			++		++			
s	bg		++						++

Table 6.2-1 - Updating true positive, true negative, false positive and false negative counts for a background model and video combination. In this table: TP = true positive, TN = true negative, FP = false positive, FN = false negative, fg = foreground, bg = background, s = shadow, GT = ground truth. The ++ symbol indicates which of the TP, TN, FP or FN counts are to be updated based on the ground truth pixel and result pixel combination.

Table 6.2-1 provides a clear depiction of how the true positive, true negative, false positive and false negative counts for a model and video combination are maintained. If the corresponding result frame and ground truth pixels are both found to be foreground pixels, for example, both true positive counts must be updated. If a pixel in the result frame is found to be a background pixel while the corresponding ground truth pixel is labelled as shadow, the true negative count for shadow being in the background must be updated as must the false negative count for shadow being in the foreground.

When evaluating the accuracy of a background model this result/ground truth comparison is performed for every pixel in every frame of each video in the dataset. Once all frames of a video have been analysed, the true positive, true negative, false positive and false negative counts for that video are used to calculate the precision, recall and f-measure of the model in processing that video using the relevant formulae for each shadow being considered foreground and background.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Measure = 2 \frac{Pr \cdot Re}{Pr + Re}$$

The results of calculating these metrics for each model/video combination are stored in the framework database (see section 6.4) and the true positive, true negative, false positive and false negative counts are reset, ready for the next video.

6.2.4.2 Average Frame Processing Time

The speed of a background model is assessed based on the average time that it requires to process a single video frame. The time taken for a model to process all frames of a video is measured using the C++ C Time Library [68]. In doing this, the time at which the first video frame is requested is recorded as is the time at which the last processed frame is returned to the framework. The difference between these recorded times is equivalent to the total time taken to process that video. At this point the processing time is in units of clock ticks. To convert it to seconds it must be divided

by the C Time constant `CLOCKS_PER_SECOND`. Dividing the result by the number of frames in the video gives the average frame processing time for that video. For convenience, this time value is converted to milliseconds. The calculation of the average time, in milliseconds, required by a model to process a frame for a video is given as:

$$avg. \text{ frame processing time} = \left[\frac{(end_time - start_time)}{\frac{CLOCKS_PER_SECOND}{num. \text{ frames}}} \right] \times 1000$$

This may be simplified to:

$$avg. \text{ frame processing time} = \frac{1000(end_time - start_time)}{CLOCKS_PER_SECOND \times num. \text{ frames}}$$

This metric is calculated for every video in the dataset for each background model and the result is entered into the database in each case. It may be averaged for each model across all videos to reduce the influence of factors such as frame resolution.

6.2.4.3 Training Phase Length

The length of a background model's training phase is determined by counting the number of video frames that it requests from the evaluation framework before beginning to return processed frames. This is simply determined based on the ID of the first processed frame that is returned, e.g. if frame ten is the first to be returned, it is clear that the model has a training phase of ten frames (frame IDs begin at zero). This metric is stored in the framework database as is described in section 6.4.

6.2.4.4 Lag Measurement

A model's lag or the frame delay between requesting a frame and returning a processed version of it is measured by keeping track of the number of frames that have been requested and the number that have been returned by the model and determining the absolute difference between the two. As lag may vary for a single model a record of the maximum observed lag is maintained. Once the processing of a video by a model is complete, the maximum lag value that has been observed for that video and model is entered into the database.

6.3 Evaluation Framework Website

As was discussed in chapter 4, a website providing information regarding the evaluation process and the facilities to submit background models for evaluation and to view evaluation results accompanies the evaluation framework. The creation of this website involved the use of a web server on which the site could be hosted, the implementation of a number of web pages which have a variety of features using a combination of HTML, CSS, JavaScript and PHP [69] and interaction with the framework database. It was intended that this site be scalable to ensure that it would be able to deal with a significant amount of evaluation information. The remainder of this section describes the completion of these tasks (aside from database communications which are discussed in section 6.4.2) and the challenges that were encountered in completing them.

6.3.1 Web Server

An Apache HTTP server [70] was set up to host the framework website. The Apache Software Foundation initially released the open source web server in early 1995 and by April 1996 had become the world's most widely used web server and remains the most popular today. In 2009 the Apache web server became the first to serve an excess of 100 million websites [71] and by June 2013 it was estimated to be serving almost 55% of active websites [72]. The use of this server in many large scale applications is indicative of its ability to scale in order to handle a growing volume of work. This is an essential characteristic for the framework website to ensure that it will be able to manage the large scale model evaluation for which it has been designed.

The Apache web server may be used on a variety of platforms including Microsoft Windows, Linux and OS X and forms part of the LAMP (Linux, Apache, MySQL, PHP/Perl/Python) open source software stack. During its implementation, the evaluation framework website was hosted locally on the Apache web server that was set up. The setup of the server was straightforward but its configuration to run PHP proved somewhat troublesome due to difficulties encountered in finding clear and helpful documentation.

6.3.2 Website Design

The content requirements of the website were carefully considered as part of the background model evaluation framework proposal. The required site contents and pages are described in section 4.6 while the structure that the website should take was presented in Figure 4.6.1. In the implementation process, therefore, it was necessary to determine the most appropriate manner of displaying the required information. In designing the website to accompany the background model evaluation framework the primary aim was to display all of the necessary information in a way that would ensure that the user experience would be as pleasant and straightforward as possible. It was important that the site be simple and intuitive to allow users to easily and quickly navigate the site content to find the page that they need. If the site is difficult to navigate and the pages are cluttered with content, the user will likely become frustrated which will discourage them from using the site. The website was carefully designed based on these considerations.

6.3.3 Web Pages

The evaluation framework website comprises a number of webpages which contain information regarding the framework and the evaluation methodology. The facility for developers to submit their background models for evaluation is also provided and the evaluation results for all submitted models are displayed. As mentioned, the web pages are created using HTML, CSS, JavaScript and PHP as well as a number of JavaScript libraries. To navigate through the pages of the site, the menu shown in Figure 6.3.1 is used.

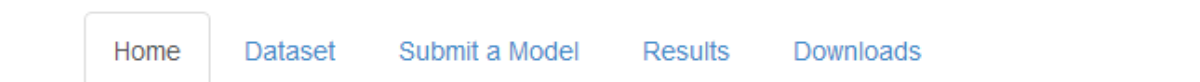


Figure 6.3.1 - Evaluation framework website menu

This menu was created using the JavaScript Bootstrap library [73] for which a small amount of documentation is available. The remainder of this section considers the implementation of the various pages of the site and the challenges that arose during their implementation.

6.3.3.1 Home Page

The home page of the framework website is the page with which users of the site are greeted. It presents an overview of the process of background subtraction, background modelling and some of the challenges that are commonly encountered as well as the need for background model evaluation and thus the purpose of the framework. Aside from the navigation menu tabs (see Figure 6.3.1) which makes use of JavaScript, this page was created using just HTML and CSS. No significant difficulties were encountered in the creation of this page.

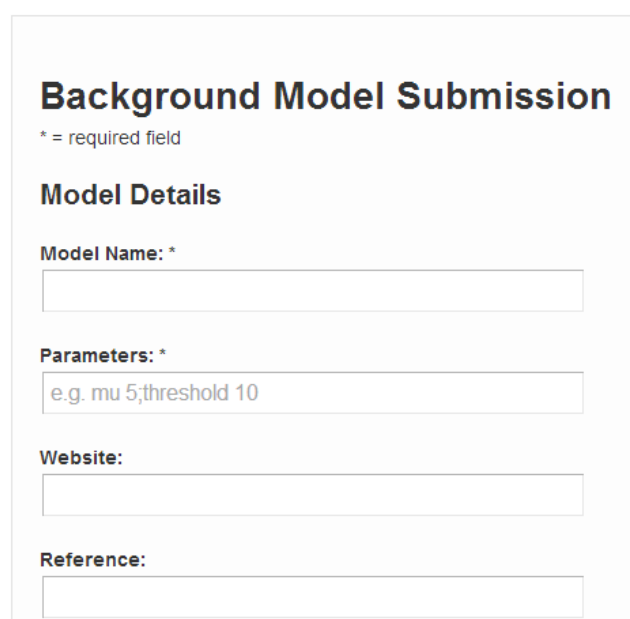
6.3.3.2 Dataset Page

The dataset page details the various challenges that are represented by the dataset that is used in evaluating the submitted background models and provides a sample frame and corresponding ground truth frame from each video of the dataset. Also described is the type of ground truth that is used in the background model evaluations. As was the case with the site home page, the dataset page, except for the navigation menu, was implemented using both HTML and CSS. The creation of this page did not present any major difficulties.

6.3.3.3 Model Submission Page

The model submission page is the page to which developers go when they wish to submit a model for evaluation. Upon navigating to this page, a form facilitating this submission is presented. The various sections of this form are shown in Figure 6.3.2 - Figure 6.3.4 and the process of using it to submit a background model is described below. The form was created using HTML and CSS while PHP was used for the validation of the supplied information, the uploading of code files and the storing of the submitted information in the framework database. Interactions between the evaluation site and the database are discussed in section 6.4.2.

When submitting a background model to the evaluation framework, developers are first asked to provide details about the model that they have created. The requested details, including the model name and the parameters that are used, have previously been discussed in section 4.2. Figure 6.3.2 depicts this part of the model submission form.



Background Model Submission
* = required field

Model Details

Model Name: *

Parameters: *

Website:

Reference:

Figure 6.3.2 - Submission of background model details

Developers are also asked to supply their contact details to the framework. The option exists for these details to be shared only with the framework or for them to be made public so that users can contact developers with any queries they may have regarding their models. The information that is requested – the contact’s name, email address and affiliation – was previously discussed in section 4.2. This part of the model submission form may be seen in Figure 6.3.3.

Contact Details

First Name: *

Last Name: *

Email: *
 Public Private

Affiliation:

Figure 6.3.3 - Submission of developer details

Finally, the background model C++ code files must be provided so that the evaluation may be performed. This may be accomplished by clicking the upload area which is shown in Figure 6.3.4, browsing to and selecting the relevant files to upload or by simply dragging the files to be uploaded to this area. The upload area was created using the open source JavaScript library dropzone.js [74] for which detailed documentation is provided.

Upload Code Files: *

Drag files here or click to browse files...

Submit

Figure 6.3.4 - Submission of background model code files

Once the appropriate code files have been selected (see Figure 6.3.5), these, along with the associated details may be submitted to the framework by simply pressing the submit button.

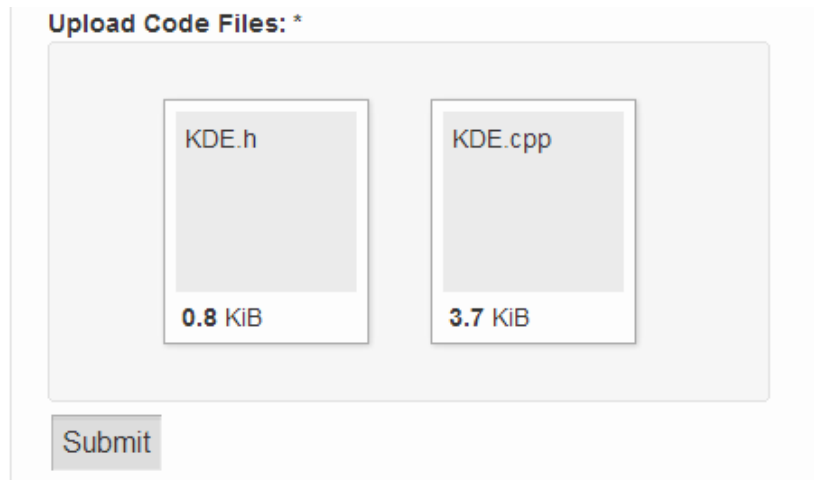


Figure 6.3.5 - Code files selected for upload

The validity of the information that has been provided is checked using PHP. If it is found that the form was filled out correctly the background model submission process is considered complete and the data which has been provided is entered into the relevant tables of the database via PHP as is described in section 6.4.2 while code files are stored in an appropriate location. If this is not the case, however, the submitter is notified of the issues that exist and given the opportunity to rectify them.

It may be seen from Figure 6.3.2 and Figure 6.3.3 that some of the form fields are required while others are optional. If a required field is left blank and a developer attempts to complete submission, the PHP form validation will detect this issue and will not allow the submission to be processed. Instead, the developer will be presented with an error message as shown in Figure 6.3.6.

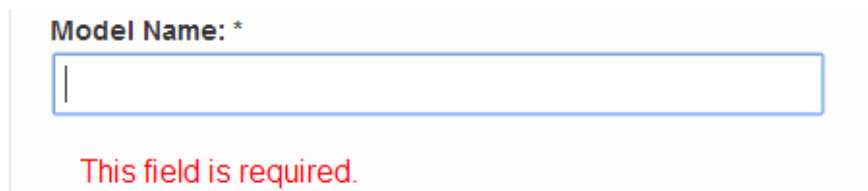


Figure 6.3.6 - Field required error message

In addition, if it is determined in the form validation process that an invalid email address has been provided, the model submission will not be completed and an error message will again be presented to indicate the issue. This may be seen in Figure 6.3.7.

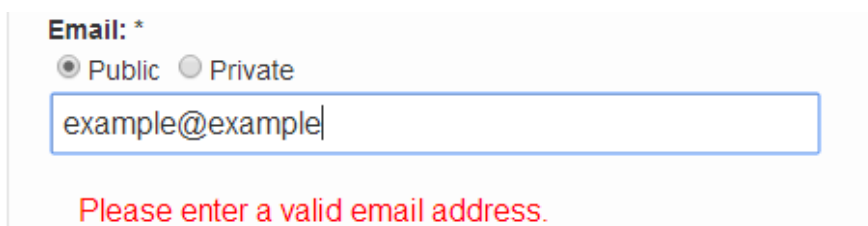


Figure 6.3.7 - Invalid email address error message

Once any such issues have been rectified, the submission process can be completed. Upon the successful submission of a background model, the submission is acknowledged and the submitter is directed to the results page where the evaluation results of their model may be viewed once they are available.

Some difficulties were encountered during the implementation of the model submission page. The most significant of these concerned the submission of code files using the dropzone.js library. Initially, submitted files were not being stored where expected and the reason for this could not be found. Eventually this was determined to be a logical issue and, once isolated, was reasonably straightforward to rectify. Other, minor issues, concerning the appearance of the form were also encountered but these were simply a result of a lack of experience in working with CSS and were resolved through the examination of relevant code samples.

6.3.3.4 Results Page

The results page of the evaluation framework website displays the evaluation results for all background models that have been submitted. Users of the site can come to this page to obtain the evaluation results of their own models, to aid them in selecting an appropriate model for a particular application etc. This page was implemented using a combination of HTML, CSS, JavaScript and PHP. PHP was used to interact with the database in order to obtain evaluation results and model information. This is discussed further in section 6.4.2. The implementation of the remaining aspects of the page which are highlighted in the results page overview shown in Figure 6.3.8 is described here.

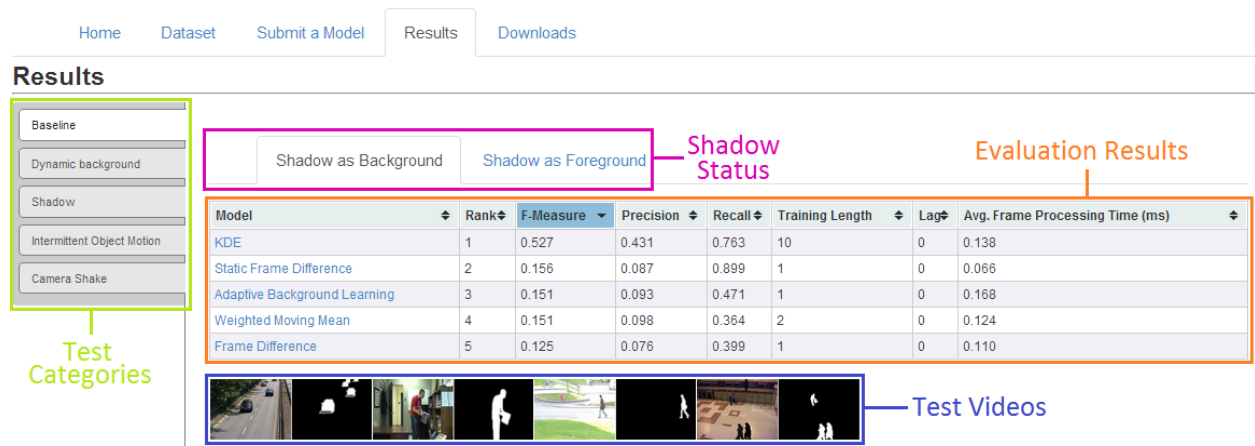


Figure 6.3.8 - Results page feature overview

Upon navigating to the results page, users can select a challenge category from those in which the model was tested in order to see the evaluation results for that particular challenge. This is beneficial when attempting to choose an appropriate background model for a specific application, e.g. if a model is required to work with a scene which has a dynamic background, the performance of the different models when faced with this challenge may be seen by simply selecting the dynamic background category from the challenges tabs. These tabs are highlighted in green in Figure 6.3.8 and may be seen more clearly in Figure 6.3.9.

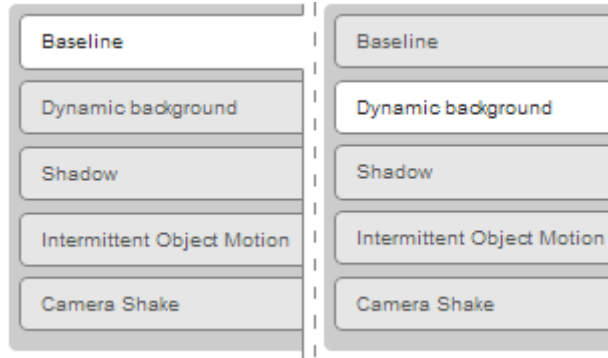


Figure 6.3.9 - Challenge category selection tabs

Having selected a challenge from which to view evaluation results, a sample frame and the corresponding ground truth frame from each video which depicts this challenge are displayed so that users can see how many videos have been used and what variations on the challenge have been considered. The sample video and ground truth frames for the baseline category may be seen in the blue box in Figure 6.3.8 and for the dynamic background category in Figure 6.3.10.



Figure 6.3.10 - Sample video frames and corresponding ground truth frames from videos in the dynamic background category

By hovering over any of these frames with the mouse cursor the image may be viewed at a larger scale as is shown in Figure 6.3.11. This feature was created using CSS.



Figure 6.3.11 - Zoomed sample frame on mouse rollover

Within each challenge category, the user can choose to view evaluation results with shadow being considered as either background or foreground. It was seen earlier that shadow is typically considered to be part of the background but that there are some applications in which it is important for it to be considered foreground. By default, the evaluation results with shadow being regarded as background are displayed but this may be changed using the shadow tabs on the results web page. As with the main site navigation menu, these tabs were created using the JavaScript Bootstrap library [73]. They may be seen in Figure 6.3.12 and in the pink box shown in the results page overview in Figure 6.3.8.



Figure 6.3.12 - Shadow status selection

Once a user has determined what results they wish to see by selecting both a category and their preferred shadow status, the database is queried using PHP (see section 6.4.2) to obtain the relevant evaluation results. For a given category, the average evaluation results for all videos in that category are obtained for each model. This information is displayed in a table created using the tablesorter JavaScript library [75] for which reasonable documentation is provided. A complete example of such a table may be seen in the orange box in Figure 6.3.8. The contents of this table may be sorted based on each column. By default, the background models are listed in descending order of their accuracy, i.e. the most accurate model for the relevant challenge is placed at the top of the table. Shown in Figure 6.3.13 are some evaluation results for a number of background models in the camera shake category when shadow is being considered part of the background. Only accuracy results are shown here due to space limitations.

Model	Rank	F-Measure
KDE	1	0.214
Weighted Moving Mean	2	0.160
Adaptive Background Learning	3	0.131
Frame Difference	4	0.110
Static Frame Difference	5	0.048

Figure 6.3.13 - Background model evaluation accuracy results for the camera shake category with shadow considered part of the background

By simply clicking the header of any table column the rows will be resorted based on the contents of that column. This can be helpful in selecting a background model based on requirements other than accuracy. Figure 6.3.14, for example, shows evaluation results for a number of models sorted in ascending order by the length of their training phases. Again, these results were obtained from running the models on the videos of the camera shake category with shadows being considered part of the background. Just a selection of columns from the table is shown due to space limitations.

Model	Training Length	Lag	Avg. Frame Processing Time (ms)
Static Frame Difference	1	0	0.113
Frame Difference	1	0	0.078
Adaptive Background Learning	1	0	0.165
Weighted Moving Mean	2	0	0.125
KDE	10	0	0.177

Figure 6.3.14 - Background model evaluation results for the camera shake category with shadow considered part of the background

To extend the usefulness of the site, all evaluation results for a specific background model may also be viewed. By clicking on the name of a model in any table, a page will be displayed showing all available information about that model. Figure 6.3.15, for example, shows the model and developer details that are displayed for the KDE background model while Figure 6.3.16 shows the accuracy evaluation results for this model when faced with a number of challenges. These results are again shown in a table created using the tablesorter JavaScript library [75]. The average results over all videos in each challenge category are shown. In addition, results may be viewed with shadow

considered to be either background or foreground as before. By displaying all results for a single model across a large and diverse range of challenges, the strengths and the weaknesses of that model become clear. This is of great benefit to those who are developing and using background models.

Model Name
KDE

Parameters
alpha = 0.3
threshold = 0.0000001

Contact Name
Bob Smith

Contact Email
bob@gmail.com

Affiliation
TCD

Figure 6.3.15 - Displayed KDE background model and developer contact details

Shadow as Background		Shadow as Foreground	
Category	F-Measure	Precision	Recall
Basic Sequence	0.746	0.655	0.895
Shadow	0.663	0.552	0.894
Multimodal Background	0.527	0.431	0.763
Thermal	0.401	0.768	0.301
Intermittent Object Motion	0.388	0.389	0.452
Camera Shaking	0.214	0.127	0.690

Figure 6.3.16 - Accuracy evaluation results for the KDE background model when face with a number of challenges

No significant difficulties were encountered in the creation of the results page but, due to limited experience in working with web technologies, its development took longer than anticipated. The basic implementation of the page was straightforward but the use of CSS to improve its appearance was somewhat challenging as was the use of JavaScript to sort the result table and to develop the tab functionality. In addition, the task of correctly adding data to the result tables in a programmatic manner proved somewhat troublesome. This was the most challenging of all pages to implement due to the large number of features that were required.

6.3.3.5 Downloads Page

The downloads page provides the API static library for use in adapting background models to fulfil the requirements of the framework. In addition to the use of JavaScript to create the navigation menu, HTML and CSS were used to create this page.

6.4 Database

As was mentioned previously, a database is required to store information regarding the dataset that is used, the background models that are submitted to the framework, the model developers and the evaluation results. In the implementation of the evaluation framework, a relational database was used, along with MySQL [76], an open source relational database management system (RDBMS) which is owned by Oracle. Since its release in 1995, MySQL has become the most popular open source RDBMS available. It is used by many large companies and organisations including Wikipedia, Google, WordPress and MyBB and is also suitable for use in smaller scale applications. The proven success of this technology in large-scale applications displays its suitability for continued use as the test dataset and the number of models being evaluated by the framework, grows. The default MySQL interface is a command line but a number of third party GUIs have also been developed. MySQL is a common choice in the development of web applications and forms part of the LAMP (Linux, Apache, MySQL, PHP/Perl/Python) open source software stack. The database is hosted locally for testing purposes. In making the framework available for public use it would be deployed to a remote server.

6.4.1 Database Design

The first step in the design of the database is the determination of the various entities that are required and their associated attributes. The relationships that exist between these entities were also determined. Using this information, an entity-relationship diagram which describes the database in an abstract manner was created. This may be seen in Figure 6.4.1.

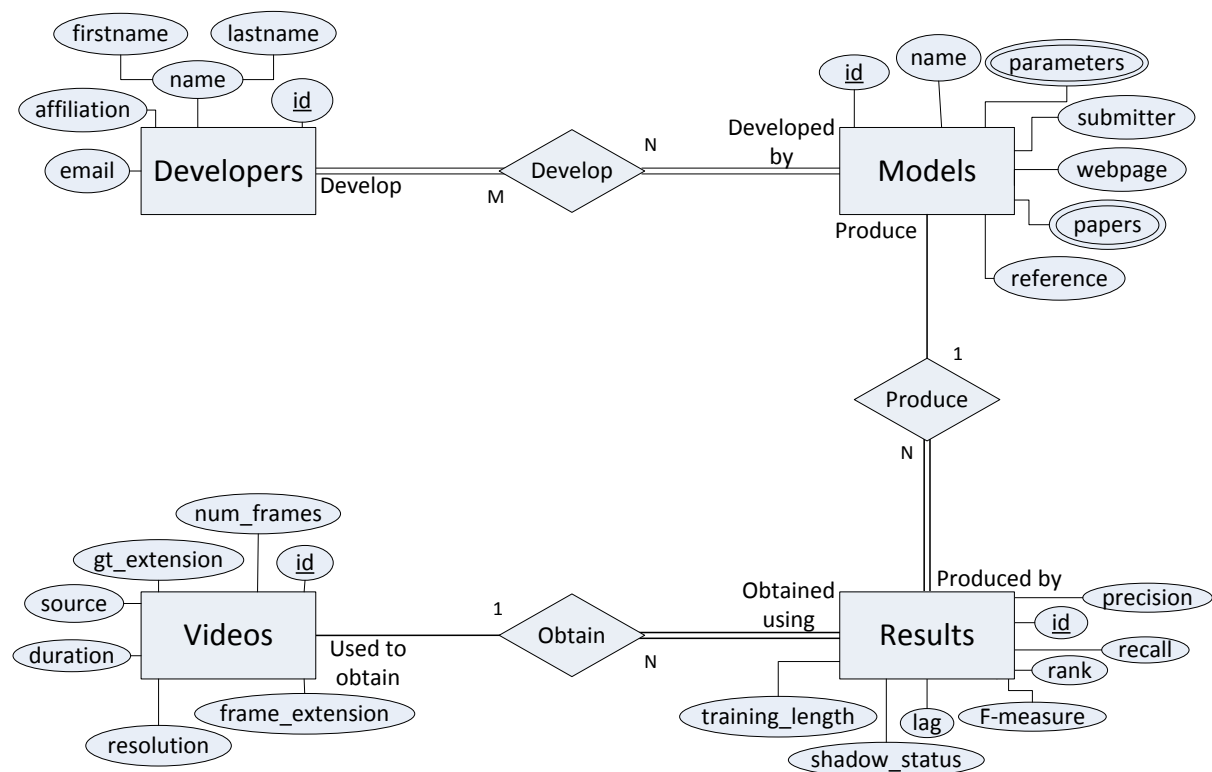


Figure 6.4.1 - Database entity relationship diagram

The next step in the design of the database for the proposed background model evaluation framework was to map the entity relationship diagram to an outline relational schema. This is shown in Figure 6.4.2. The relational schema depicts the tables that are to be contained within the database, the attributes contained in them, the primary key of each table (underlined attribute) which uniquely identifies all entries or tuples and the foreign keys which depict the relationships between the tables (depicted by arrows).

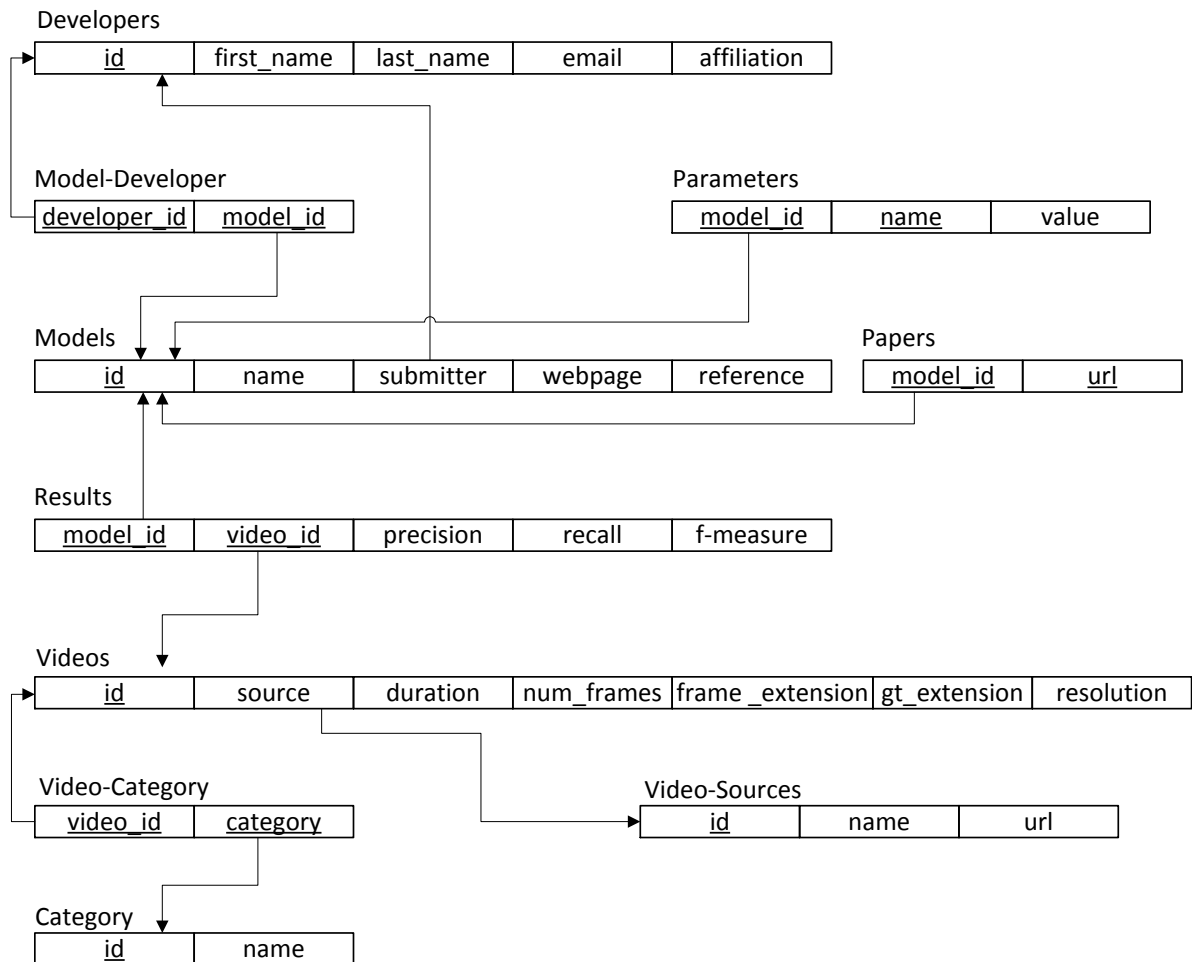


Figure 6.4.2 - Mapping to relational schema

Having created the relational schema for the database the tables could be created. As may be seen from Figure 6.4.2, ten tables are contained within the database. The contents of these tables are briefly described in Table 6.4-1.

Table	Contents
developers	Details of the background model developers including their name, affiliation and contact details.
models	Background model information including name and reference details.
model_developers	Details of which developers have created which models.
parameters	Details of the parameter values used by background models.
papers	Links to papers concerning submitted models.
results	Evaluation results for each model and video combination including precision,

	recall, f-measure, lag and training phase length.
videos	Details of the video dataset used in evaluation including video sources, duration and the number of frames in each video.
video_sources	List of sources of the videos contained in the dataset.
category	List of video categories or the challenges depicted by the videos of the dataset.
video_category	Details of which videos fall into which category.

Table 6.4-1 - Overview of database tables and contents

6.4.2 Database Communications

In order for the evaluation framework to function correctly it was necessary to establish communications between the database and the C++ part of the framework which performs the evaluations so that information regarding the submitted background models and the dataset could be obtained and evaluation results could be stored. In addition, communications between the framework website and the database were also required so that submitted background model and developer information could be inserted and information to be displayed on the site could be obtained. The framework website makes use of PHP to communicate with the database.

6.4.2.1 C++ Communications

To communicate with the database via C++, the C++ connector, SQLAPI++ [64], was used. This connector is capable of interacting with a number of RDBMSs and makes use of their native APIs. It has the ability to connect to a database, query it and manipulate the data contained within it via SQL statements. The connector is well documented and straightforward to use. It was, however, somewhat troublesome to use the query results as desired as they are provided using data types defined by the SQLAPI++ library which were difficult to convert to the required data types.

Some difficulties were encountered in establishing communications with the framework database via C++. It was initially intended that the official MySQL C++ connector [77] be used but this was accompanied by unreliable documentation and proved to be extremely troublesome to configure. Upon researching the issues that were encountered it was found that many others had encountered these same issues but had found no solutions. Rather than spending a large amount of time attempting to resolve the difficulties presented by the official connector, an alternative was sought.

6.4.2.2 PHP Communications

To establish communications with the database via PHP the mysqli extension was used [78]. This is accompanied by detailed documentation and facilitates the straightforward connection to and use of SQL commands to interact with the database. The main issue that was encountered regarding PHP database accesses is that they are slow. Due to this, the result page of the framework website is slow to load. To rectify this, the use of asynchronous database accesses must be investigated.

6.5 Implementation Overview

This chapter has described the way in which a proof of concept version of the background model evaluation framework and methodology proposed in chapters 4 and 5 was implemented, including the technologies that were used and the challenges that were encountered. The main challenges that arose during the implementation were largely a result of being initially unfamiliar with many of the technologies that were used and difficulties in finding required information. While these

challenges were overcome in time they caused the implementation of the various aspects of the framework to be more time-consuming than had initially been anticipated. The next chapter presents some sample evaluation results that were obtained using this implementation and discusses the advantages provided by the evaluation system as well as its limitations.

Chapter 7 Evaluation

In the previous chapters, the current state of background model evaluation and the limitations that exist regarding the manner in which these evaluations are carried out were discussed. An evaluation framework and associated evaluation methodology were proposed to overcome these limitations and to provide a facility to comprehensively and objectively assess the quality of background models. In addition, the implementation of a proof of concept version of the proposed framework was described as were the difficulties that were encountered during the implementation. This chapter considers the feasibility and the functionality of the proposed framework. The implemented version is tested and the background model evaluation results that are obtained are analysed. In addition, a review of the advantages and the limitations of the proposed framework as well as of the successes of the project is presented. The framework is also considered in terms of how well it achieves the desired software characteristics that were discussed in section 6.1.

7.1 Testing of the Proposed Evaluation Framework

To verify that the implemented background model evaluation framework operates as intended and is capable of providing a thorough evaluation of model performance, several model implementations were adapted to work with the framework and the process of model submission, execution and evaluation was carried out for each. This was necessary to assess the feasibility and functionality of the framework. Those models that were selected for testing are those which were described in chapter 2 - static frame difference, frame difference, weighted moving mean, adaptive background learning, KDE and Gaussian mixture model. As previously mentioned, implementations of these models were obtained from the background subtraction library, bgslibrary [65] [66]. Using the API functions described in section 4.1, these were modified to take video frames from the evaluation framework and to return the corresponding processed frames. This adaptation was found to be very straightforward and was completed in just a few minutes. In addition, the submission process operates as expected with all supplied information and model code files being correctly stored.

Once submitted, these models were successfully built and were run using a test video dataset. The videos that were used in testing the proposed evaluation framework were obtained from five of the six categories (baseline, dynamic background, camera shake, intermittent object motion and shadow) of the 2012 ChangeDetection.net dataset [14]. This dataset is not comprehensive enough to provide a complete model evaluation but, due to time constraints, a dataset of a sufficient standard, as outlined in chapter 5, could not be compiled. The complete dataset that was compiled for this testing is larger than that which was actually used as the time and resources were not available to create ground truth for all videos. As the ChangeDetection.net videos are accompanied by detailed ground truth, they were selected for use in the initial framework testing. A sample frame from each video that was used, along with the corresponding ground truth frame, may be seen in Figure 7.1.1 - Figure 7.1.5.

As video frames were processed, the models' performance was assessed as is described in section 4.4 and the results of this performance assessment were successfully entered into the database once available. Once the evaluation was complete, the results were displayed on the framework

website. Some of the evaluation results for the assessed models are presented and discussed here. The ability to review and examine background model evaluation results in this way with minimal effort on the part of the developer beyond the actual model implementation, illustrates the ease of use and successful operation of the proposed evaluation framework. Evaluation results for the case in which shadow is considered to be part of the scene background are provided for each the baseline, dynamic background, camera shake, intermittent object motion and shadow categories and are ranked, in each case, based on their accuracy or f-measure. In the results that are provided, a model’s recall indicates how much of the scene foreground it were successful in determining while its precision is indicative of how much of the scene background was included in the obtained foreground masks. In both cases a value close to one is desirable. In addition to the results of metric calculation, a sample frame from each video is shown along with the segmentations of that frame that were produced using each of the evaluated background models. The examination of these frames provides additional information regarding the performance of the models and is beneficial in understanding why certain results were obtained.

7.1.1 Baseline

It was seen previously that videos in the baseline category contain a variety of challenges typical of the other categories. Evaluation results obtained using videos from this baseline category were therefore considered as an initial overview of model performance. Table 7.1-1 shows the evaluation results that were achieved by the six examined background models on the videos of this category while Figure 7.1.1 depicts a sample frame from each video along with the corresponding ground truth frame and result frames produced using each of the six evaluated background models.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
KDE	0.746	0.655	0.895	10	0	0.151
Static Frame Difference	0.477	0.351	0.935	0	0	0.075
Adaptive Background Learning	0.458	0.634	0.429	0	0	0.121
Gaussian Mixture Model	0.395	0.676	0.328	0	0	0.081
Weighted Moving Mean	0.391	0.719	0.310	2	0	0.069
Frame Difference	0.373	0.621	0.311	1	0	0.053

Table 7.1-1 - Background model evaluation results for the baseline category

From Table 7.1-1 it is clear to see that the accuracy of the KDE model for the baseline category is superior to that of the other evaluated models, i.e. its f-measure value is higher. This is also apparent by considering the KDE result frames that are shown in Figure 7.1.1. From both these frames and the high recall value achieved by the KDE model it is evident that this model is quite adept at finding the foreground of the video frames. Varying performance is seen in terms of its precision or the amount of background that is included in the result as foreground. The high recall and reasonable precision of this model combine to give it quite a good accuracy.

The recall of the static frame difference model or its ability to determine foreground pixels is very high, i.e. it detects almost all pixels of the foreground objects in the scenes. Unfortunately, however, it performs poorly in terms of precision with much of the background often being included in the foreground mask. While it is better even than the KDE model in detecting the scene foreground, its inability to ignore changes in the background severely diminishes its overall accuracy.

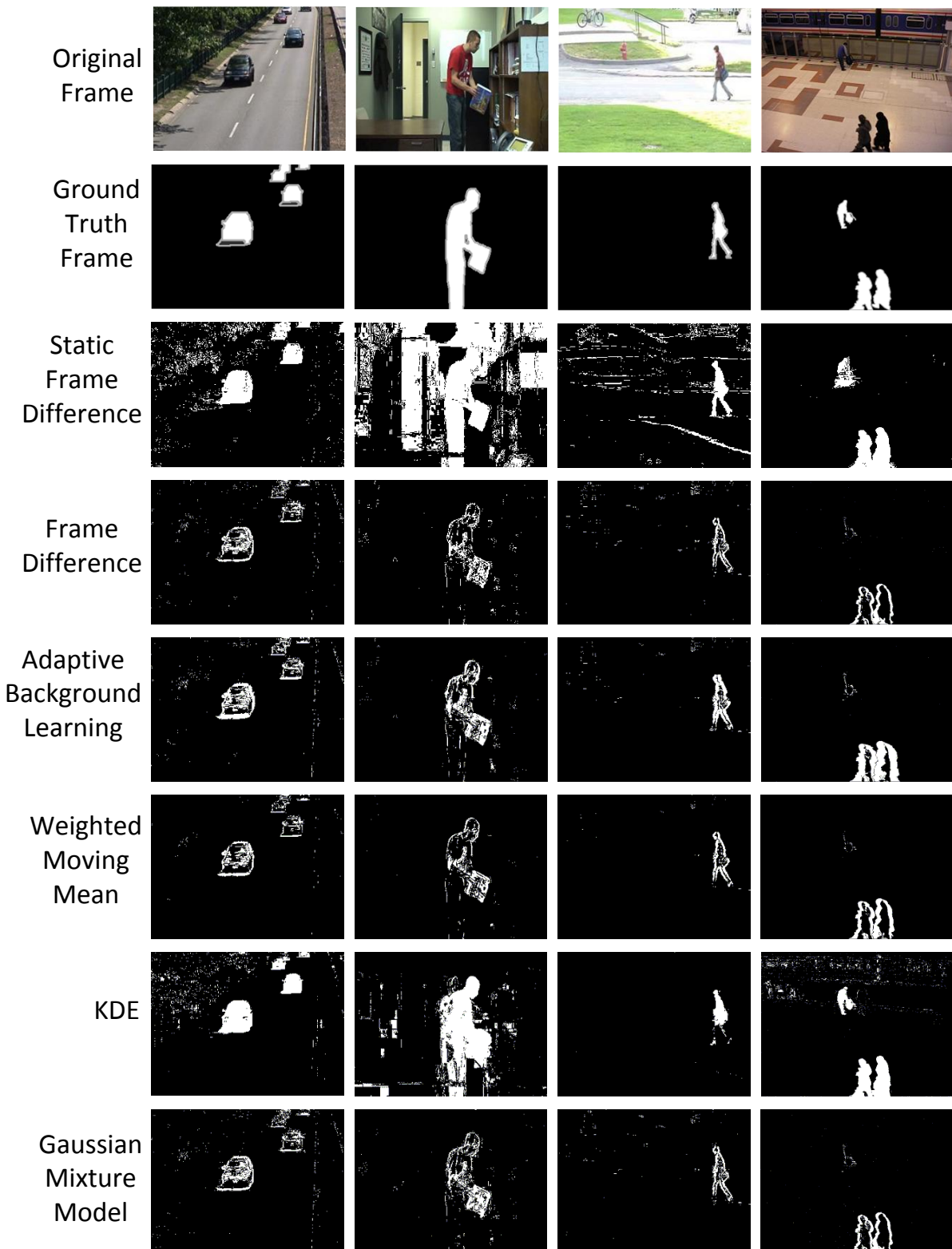


Figure 7.1.1 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the baseline category

Table 7.1-1 shows that the adaptive background learning model, the Gaussian mixture model, the weighted moving mean model and the frame difference model exhibit reasonably similar behaviour

in this baseline category. In each case, particularly that of the weighted moving mean model, precision is reasonably good, i.e. only a small amount of the scene background is mistaken for foreground. This is illustrated by Figure 7.1.1. The recall of these models, however, is significantly lower than that of the KDE and static frame difference models. In each case, much of the interior foreground object pixels are lost which is damaging to their overall accuracies.

From this initial overview of model accuracy the KDE model appears to perform significantly better than all other models examined. While the other models show strength in either their precision or their recall they are let down by poor performance in the other. KDE is the only model to exhibit reasonable performance in both aspects of accuracy. The way in which various challenges affect these initial impressions of model accuracy is examined throughout the remainder of this section.

While the KDE model is superior in this category in terms of its accuracy, it is weakest in terms of the average time that it requires to process a frame. On average, as shown in Table 7.1-1, the KDE model requires 0.151 ms to process each video frame while the remainder require much less.

In addition, the KDE model has a longer training phase than any other that was examined. While this does not appear to have been an issue in this baseline category, it has the potential to prove troublesome in other scenarios and may limit the applications in which the model may be used. Finally, all models that were evaluated in this category exhibited no lag making them suitable for use in both online and offline applications.

7.1.2 Dynamic Background

This category contains videos in which at least part of the background is dynamic. This motion is introduced by moving water and plants moving in the wind. The success of the various background models on the videos in this category is shown by the evaluation results in Table 7.1-2. Figure 7.1.2 shows a sample frame from each video in the dynamic background category as well as the corresponding ground truth frames and the result frames obtained using each of the six assessed background models.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
KDE	0.527	0.431	0.763	10	0	0.138
Static Frame Difference	0.156	0.087	0.899	0	0	0.066
Adaptive Background Learning	0.151	0.093	0.471	0	0	0.168
Weighted Moving Mean	0.151	0.098	0.364	2	0	0.124
Gaussian Mixture Model	0.131	0.080	0.415	0	0	0.331
Frame Difference	0.125	0.076	0.399	1	0	0.110

Table 7.1-2 - Background model evaluation results for the dynamic background category

It is evident from Table 7.1-2 that model accuracy is significantly reduced when faced with a scene containing dynamic background pixels. While the KDE model maintains a moderate overall accuracy or f-measure all other models including, surprisingly, the Gaussian mixture model, performed quite poorly. As may be seen from both Table 7.1-2 and Figure 7.1.2 both the KDE and the static frame difference models again have good recall, i.e. they manage to detect most of the foreground object pixels in the scenes. Unfortunately, however, precision is significantly reduced as the scenes' background motion is mistaken for foreground causing a decrease in overall accuracy.

The KDE model is by far the most capable in handling background motion of all models considered, as is evident from Figure 7.1.2. It performs very well in ignoring background motion that results from moving water whereas all other models examined were very poor in dealing with this. It is not as strong, however, in dealing with background motion that results from moving trees. KDE's moderate abilities in ignoring background motion, i.e. its precision, combined with good recall, make it the best choice of all models examined for a scene in which background motion is present. It does, however, still experience difficulties.

As in the baseline category, the static frame difference model exhibits high recall, i.e. it detects most of the foreground object pixels in the scene. The reason for its good recall is that the model is considering anything that has changed since the first frame of each video to be foreground. Unfortunately, however, this practice means that all background motion in these videos is also considered to be foreground and thus the precision exhibited by the static frame difference model is very low. This is apparent from Figure 7.1.2 from which it can be seen that all areas of background motion have been largely regarded as foreground by this model.

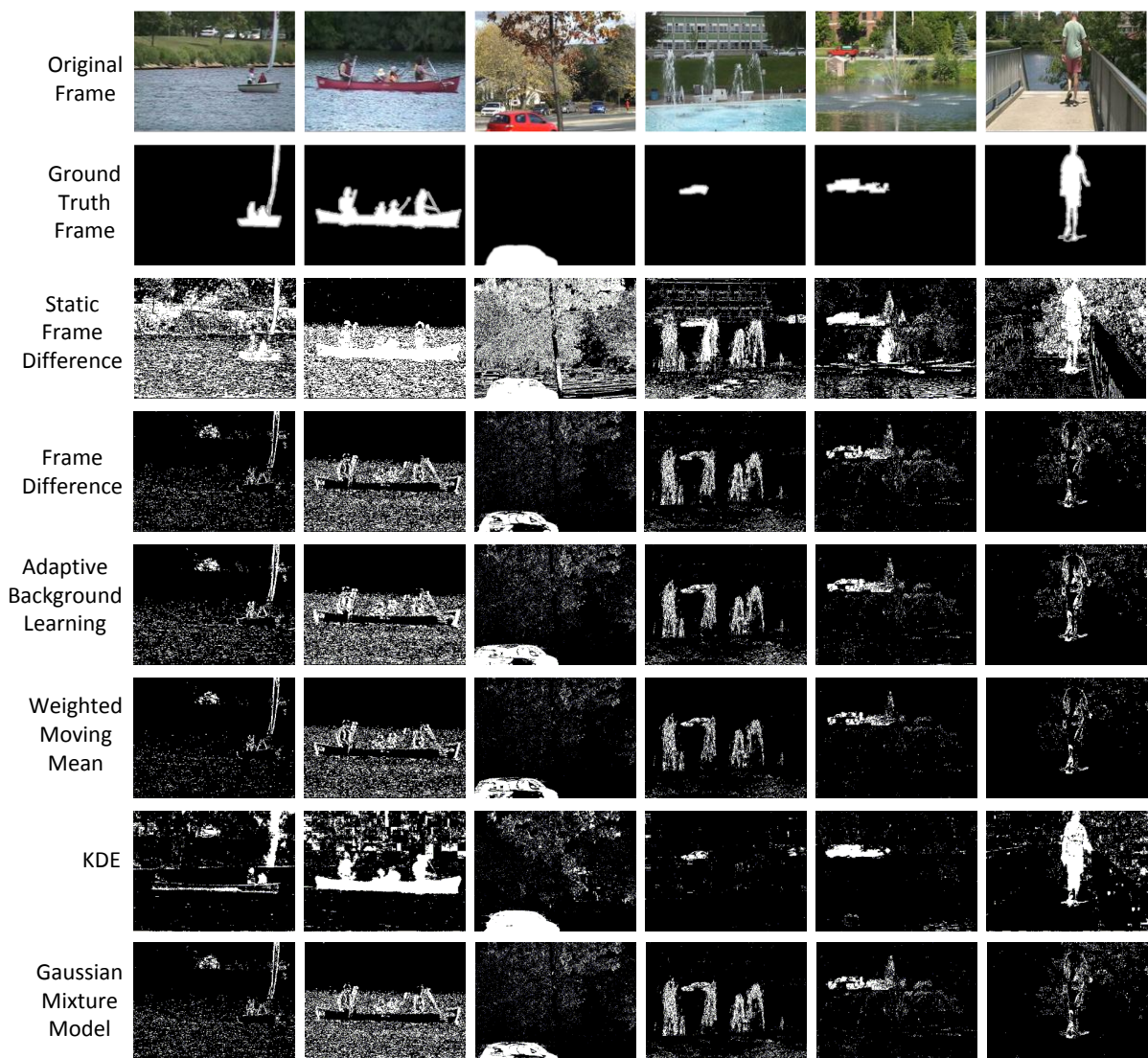


Figure 7.1.2 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the dynamic background category

The recall of the remaining models remains similar to that which was seen in the baseline category but, as with the static frame difference model, their precision has decreased by a considerable amount. Figure 7.1.2 shows that the behaviour of these models is again quite similar. Moving water is largely mistaken for foreground as are, to a lesser extent, moving trees.

From this evaluation it is clear that, of the background models examined, the KDE model is by far the most adept in facing the challenge of a dynamic background. It makes good efforts to ignore all types of background motion with which it is presented and, in the case of moving water, it largely succeeds. The static frame difference model is very strong in detecting foreground objects but has extremely poor precision, i.e. it is very weak in dealing with background motion. The overall accuracy of the remaining models was poor though as may be seen through the examination of Figure 7.1.2 this is not a true illustration of their abilities in ignoring uninteresting background motion. From the visual examination of the result frames produced by each of these models they exhibit similar capabilities in dealing with plant motion as the KDE model but are much weaker when presented with moving water. It is likely, therefore, that if foliage was the source of background motion in all videos in this category the accuracy of the adaptive background learning model, the Gaussian mixture model, the weighted moving mean model and the frame difference model would be significantly improved and comparable to that of the KDE model. This indicates that it may be prudent to introduce additional video categories based on the source of background motion so that a more accurate understanding of the background models' capabilities may be obtained.

The average time that is required to process each video frame has, for several models, increased with respect to the baseline measurements. This is particularly true for the Gaussian mixture model whose average frame processing time has approximately quadrupled as a result of being required to model a larger amount of background dynamism.

The training phase length and lag measurements remain the same for each model as in the baseline category.

7.1.3 Camera Shake

The videos in this category were recorded using an unsteady camera causing the scene to shake. The amount of shaking varies between videos. Evaluation results obtained from the six assessed background models using the videos from this category are shown in Table 7.1-3. Some sample video frames from this category are shown in Figure 7.1.3 along with the corresponding ground truth frames and the result frames produced using the various background models.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
KDE	0.214	0.127	0.690	10	0	0.177
Weighted Moving Mean	0.160	0.121	0.316	2	0	0.125
Adaptive Background Learning	0.131	0.082	0.410	0	0	0.165
Gaussian Mixture Model	0.122	0.079	0.362	0	0	0.372
Frame Difference	0.110	0.072	0.341	1	0	0.078
Static Frame Difference	0.048	0.025	0.708	0	0	0.113

Table 7.1-3 - Background model evaluation results for the camera shake category

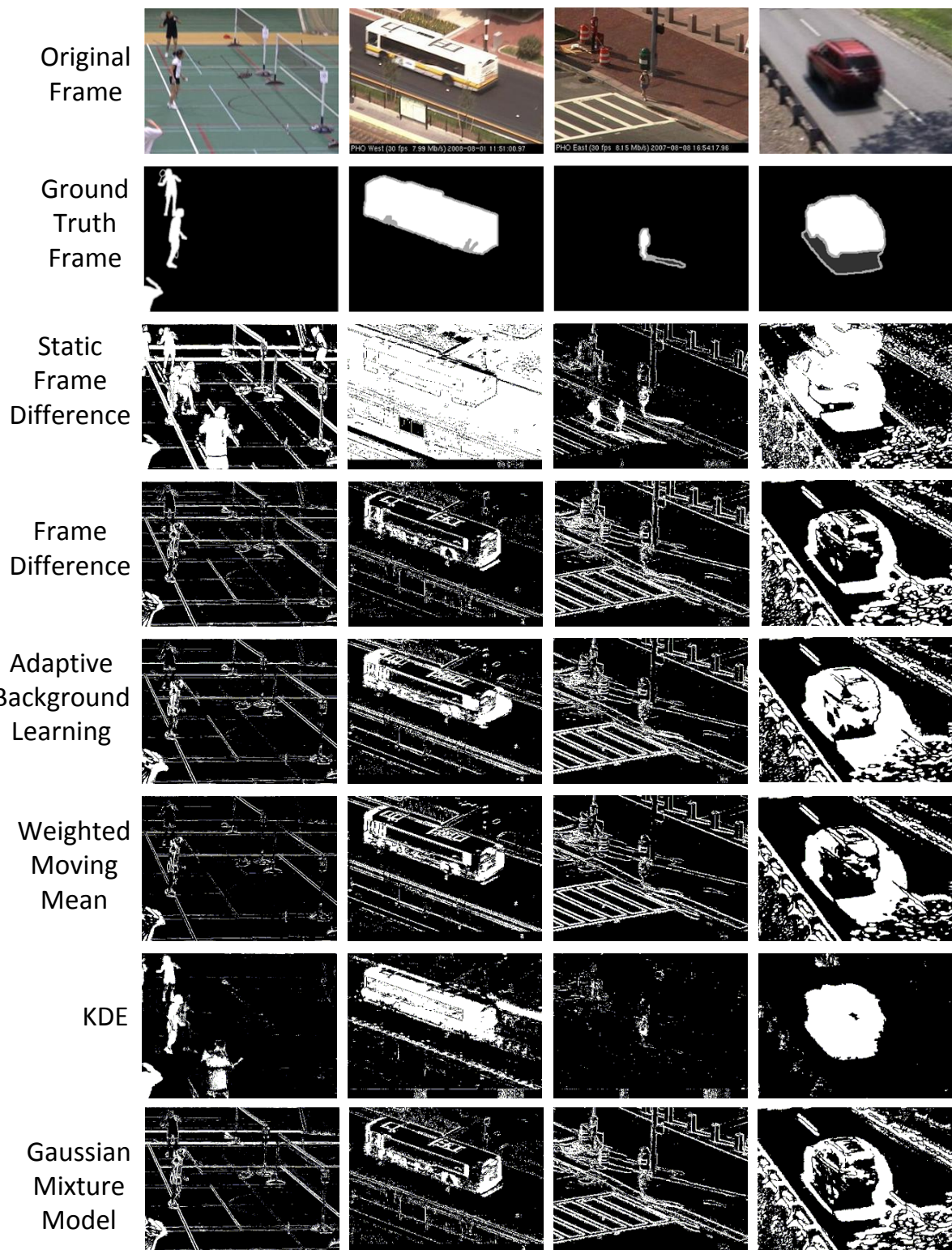


Figure 7.1.3 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the camera shake category

From Table 7.1-3 it is clear to see that the accuracy of the evaluated background models in the camera shake category is extremely poor. The KDE model exhibits the best accuracy but it is not a significant improvement over that of the other models tested. Both the KDE and the static frame difference models achieve moderately good recall values by detecting much of the foreground

objects. The recall for the remaining models is lower, but similar to that which they achieved in the previous categories.

In terms of precision, all of the examined background models perform poorly. The static frame difference model in particular, exhibits an abysmal precision, i.e. it mistakenly detects a large amount of the scene background as foreground. As this model simply considers everything in the scene that is different to the first frame as foreground, the continual shaking of the camera and thus changing of the scene makes it impossible for the background to be completely ignored. The extent to which the camera shaking affects the accuracy of this model is determined by the amount of shaking that is present. The difficulties experienced by this model are shown in Figure 7.1.3.

From the result frames provided in Figure 7.1.3, it can be seen that the shaking of the camera causes all considered models except for the KDE model to mistake the boundaries of all objects in the scene for foreground which greatly reduces their precision. The interiors of background objects are generally correctly classified as background. The KDE model is quite successful in ignoring object boundaries but the results that are obtained are still quite noisy.

A significant weakness that is seen with both the KDE and the static frame difference models is the occurrence of ghosting or the inclusion of objects in the foreground that are no longer present. This can be seen in the result frames for these models in the first and third columns in Figure 7.1.3. Ghosting significantly reduces the precision and thus the accuracy of these models. The KDE model, in particular, would exhibit a far greater accuracy were it not for the presence of ghosts.

In addition, all of the evaluated models have largely classified the shadows that are present in the scenes to be foreground. As the evaluation results being considered are those in which shadow is to be considered as part of the background this also reduces the precision and thus the accuracy of all models.

The average frame processing time for each of the evaluated background models has remained similar to that which was observed previously. Additionally, the length of the models' training periods and their lag remains unchanged.

7.1.4 Intermittent Object Motion

The videos of this category depict scenes in which moving objects come to a stop for a short period of time and then begin to move again or, alternatively, in which stationary objects begin to move and then stop again. The evaluation results for this category are presented in Table 7.1-4 and some sample video, ground truth and result frames are shown in Figure 7.1.4.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
Static Frame Difference	0.411	0.311	0.884	0	0	0.049
KDE	0.388	0.389	0.452	10	0	0.096
Adaptive Background Learning	0.235	0.583	0.156	0	0	0.128
Gaussian Mixture Model	0.198	0.613	0.127	0	0	0.136
Frame Difference	0.184	0.582	0.122	1	0	0.028
Weighted Moving Mean	0.168	0.668	0.100	2	0	0.055

Table 7.1-4 - Background model evaluation results for the intermittent object motion category

The results presented in Table 7.1-4 show that none of the evaluated background models exhibit high accuracy in the intermittent object motion category. The static frame difference model performs best in this category due to its high recall. It managed to determine large quantities of the foreground objects that were present in all of the videos that were used. The precision of this model, however, was, as before, quite low, i.e. it mistakes a large amount of the scene background for foreground which significantly reduces its overall accuracy.

For all except the static frame difference background model, recall is significantly lower than has previously been seen. This is illustrated by Figure 7.1.4 from which it may clearly be seen that these models have missed quite a lot of the scene foreground. A particular weakness exhibited by these results is the inability of the models to keep people in the foreground once they have come to a stop. In the third column of Figure 7.1.4 it may be seen that the static frame difference and the KDE models are the only ones in which the man sitting on the chair is still considered to be of interest. While this man will remain in the foreground mask of the static frame difference model, the KDE model will eventually integrate him into the scene background as has already been the case with the other models. This illustrates the need for multiple ground truth frames to be used in the evaluation of a model's accuracy as the location of these frames can impair results. For example, in the frame from immediately after the man sat down, models such as the adaptive background learning model would detect him and thus, if the evaluation metrics were calculated using just this frame the models' recall and thus overall accuracy would appear to be much better than if just the frame shown below had been used. Thus, it is important that the ground truth frames that are used allow the models' performance to be completely and fairly assessed.

It may also be seen both from Table 7.1-4 and from Figure 7.1.4 that the precision of the adaptive background learning model, the Gaussian mixture model, the frame difference model and the weighted moving mean model is greater than that of either the KDE or static frame difference models. This is particularly apparent in the frame in the sixth column of Figure 7.1.4 in which a car has moved away from its initial location to reveal a new part of the background. This is handled very poorly by the static frame difference model. The KDE model also retains much of the background in the scene's foreground mask though, unlike the static frame difference model, will eventually integrate the changes into the background. This again, is illustrative of the need for many ground truth frames as has been described above.

In the intermittent object motion category, the static frame difference model exhibits the highest accuracy due to its high recall. This, along with the KDE model, retains foreground objects that come to a stop in the foreground better than the others that were evaluated. The KDE model will eventually integrate stopped objects into the background which is beneficial when such objects are not of interest, while the static frame difference model will always keep them in the foreground. This is beneficial when considering people but poor when objects are expected to enter the background. The remaining models show better performance when faced with newly revealed parts of the background. The KDE model will eventually put these areas into the scene background but the static frame difference model never will.

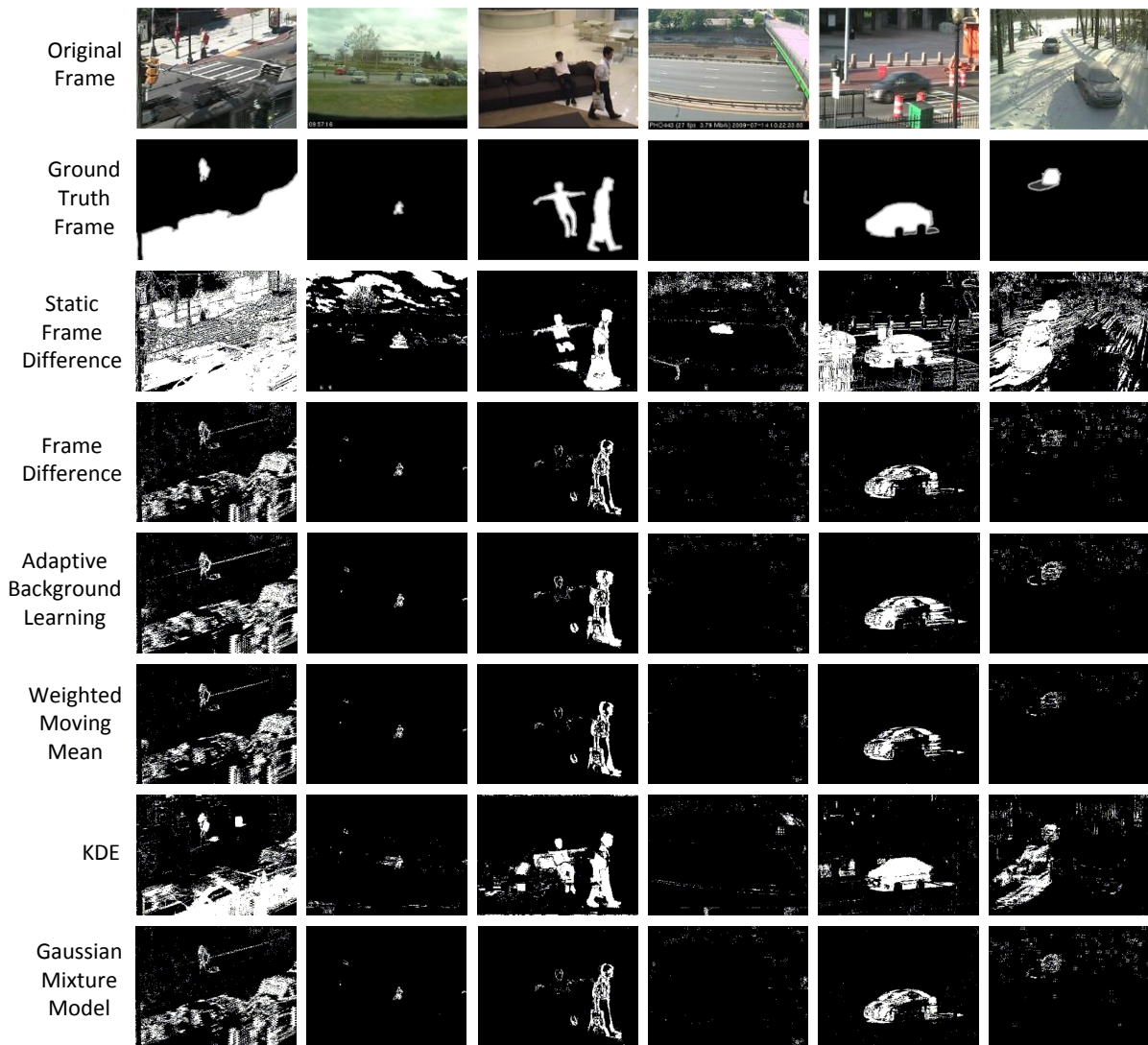


Figure 7.1.4 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the intermittent object motion category

The length of the models' training periods and their lag again remains unchanged but the average frame processing time for each of the evaluated background models has significantly decreased in comparison to the dynamic background and the camera shake categories.

7.1.5 Shadow

The videos in this category contain a range of strong and faint shadows which are cast both by people and by other foreground objects. The evaluation results obtained by the six assessed background models for the videos of the shadow category may be seen in Table 7.1-5 while a number of sample frames and ground truth frames are shown in Figure 7.1.5 along with the result frames produced using each of the models.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
KDE	0.663	0.552	0.894	10	0	0.038
Adaptive Background Learning	0.421	0.635	0.339	0	0	0.050
Static Frame Difference	0.381	0.263	0.917	0	0	0.050
Gaussian Mixture Model	0.364	0.714	0.259	0	0	0.112
Frame Difference	0.343	0.695	0.241	1	0	0.030
Weighted Moving Mean	0.332	0.705	0.234	2	0	0.049

Table 7.1-5 - Background model evaluation results for the shadow category

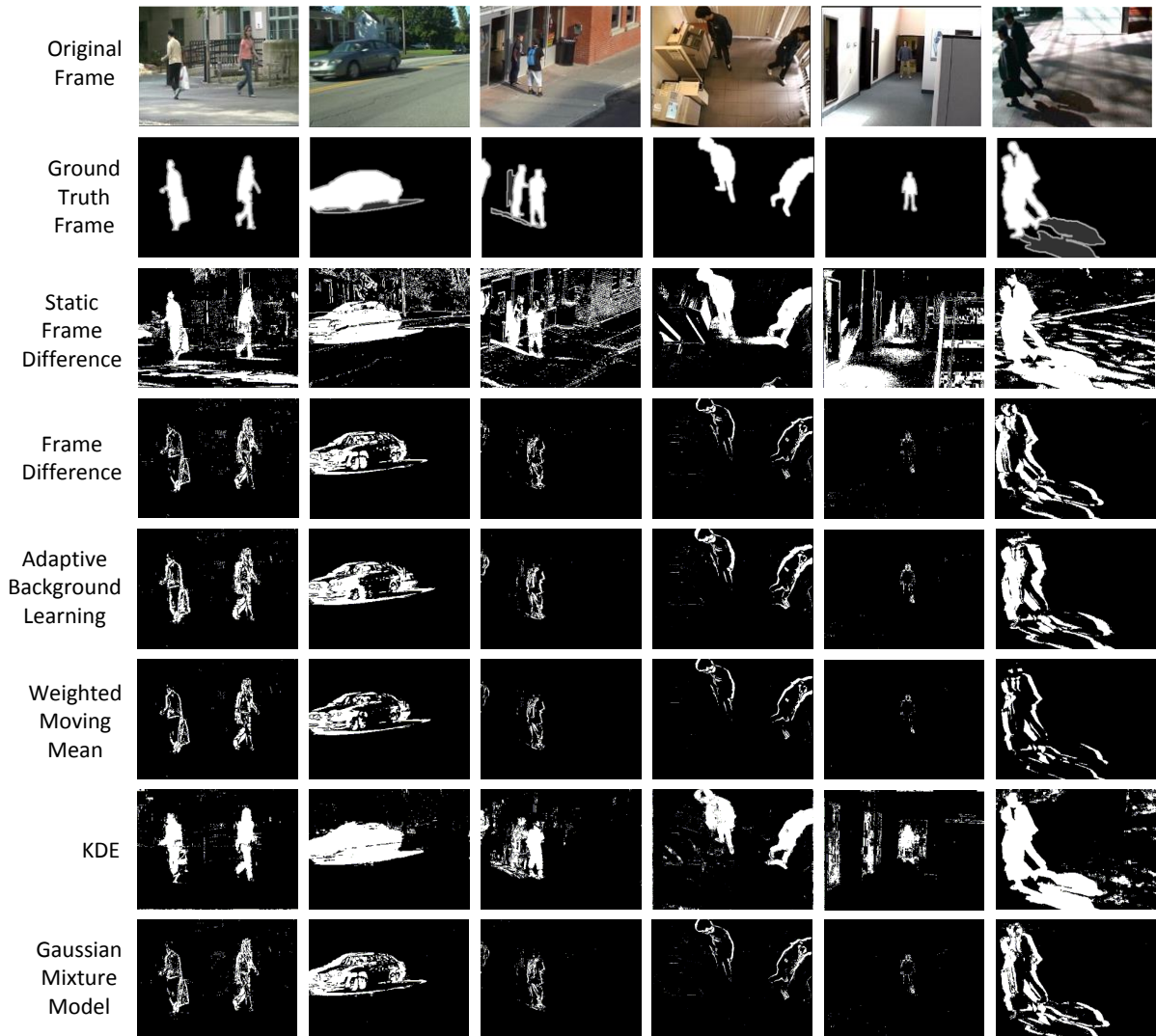


Figure 7.1.5 - Sample video frames and corresponding ground truth frames as well as result frames from the evaluated background models for the shadow category

From the examination of Table 7.1-5, it may be seen that the KDE model is again the most accurate, with an f-measure that is quite a bit higher than that of the other evaluated models. As may be seen in Figure 7.1.5, the KDE and static frame difference models manage to detect most of the scene foreground and thus have very good recall. The other evaluated models achieved significantly lower recall measurements as they mistake much of the foreground for being in the scene background.

These models tend to successfully detect the boundaries of the foreground objects but, in general, miss much of their interiors.

In terms of precision, the static frame difference model performs very poorly. It detects all shadows, even those which are very faint, as foreground and exhibits much other additional noise. The KDE model also regards a large amount of the shadow pixels in the scene as being part of the foreground mask but otherwise has much better precision than the static frame difference model. While the overall accuracy of models such as the Gaussian mixture model, the frame difference model and the weighted moving mean model is low due to their poor recall, they exhibit quite good precision. As may be seen from Figure 7.1.5, these models manage to integrate a large amount of shadow into the background and the result frames obtained using these models show very little other noise.

Thus, while the KDE model exhibits best accuracy in this shadow category due to its ability to recall much of the foreground objects, it actually copes very poorly with the shadows that are present. Those models which handle shadow quite well, however, are penalised for their poor performance in detecting complete foreground objects.

The models' training period and their lag measurements remain unchanged from the previous categories while the average frame processing time of the KDE and adaptive background learning models are significantly reduced when working with the videos of this category.

7.1.6 Performance Overview

An overview of the performance of the evaluated background models across the baseline, dynamic background, camera shake, intermittent object motion and shadow categories is provided in Table 7.1-6. In general, the camera shake and dynamic background categories were found to be most challenging in terms of accuracy, while models tended to exhibit their highest accuracy in the baseline category.

Name	F-Measure	Precision	Recall	Training Phase Length	Lag	Avg. Frame Processing Time (ms)
KDE	0.500	0.471	0.686	10	0	0.070
Static Frame Difference	0.308	0.219	0.876	0	0	0.080
Adaptive Background Learning	0.268	0.457	0.327	0	0	0.128
Gaussian Mixture Model	0.235	0.473	0.278	0	0	0.143
Weighted Moving Mean	0.222	0.512	0.236	2	0	0.072
Frame Difference	0.213	0.471	0.254	1	0	0.041

Table 7.1-6 - Background model evaluation results across all categories

From Table 7.1-6, and from the evaluation results that have been presented and discussed earlier in this chapter, is clear that the KDE model is generally the most accurate of those evaluated, i.e. it typically achieves the highest f-measure value. The frame difference model, meanwhile, is the least accurate of those evaluated. By examining precision and recall metrics and the result frames presented in Figure 7.1.1 to Figure 7.1.5, additional insights to the models' accuracy may be obtained.

Based on the analysis that has been carried out, several observations have been made regarding background model accuracy. The static frame difference model, for example, has, by far the highest

recall of any model evaluated, i.e. it detects the majority of foreground pixels in a scene, but is let down in terms of its overall accuracy by its very poor precision, i.e. it includes many background pixels in its foreground masks. This model performed particularly poorly in the camera shake and dynamic background categories as the scenes in both deviate significantly from their initial appearance. Additionally, in some cases, ghosting is experienced. The static frame difference model tends to detect all shadows as foreground and is excellent at keeping people in the frame foreground. This model is incapable, however of adapting to changes in background appearance as the background image is never updated.

In most scenarios, the KDE background model also exhibits quite good recall but has much better precision than the static frame difference model. It proved to be far superior to all other evaluated models in handling background motion. Its good performance in this category was largely a result of its excellent abilities in dealing with background motion caused by moving water. In addition, this model generally performed far better than the other models in the camera shake category but its accuracy here was hindered as a result of ghosting. The KDE model is slower to update than the models against which it was evaluated (excluding the static frame difference model) and tends to include shadow in its foreground masks.

The remaining models – adaptive background learning, weighted moving mean, Gaussian mixture model and frame difference - exhibit quite similar behaviour in the categories in which they were tested. In general, the recall of these models was found to be poor as many of the interior pixels of foreground objects were missed. Their precision, meanwhile, was mixed. In the camera shake category, for example, precision was very poor with most background object boundaries being considered foreground. Each of these models also performed very poorly when faced with moving water but managed moving trees quite well. All of these models exhibit good precision in the intermittent object motion category and in the shadow category. While the outlines of moving shadows are typically considered as foreground, large amounts of them are integrated into the background. These models also update the background more quickly than the KDE model. Regardless of the precision of these models, their overall accuracy is let down by their consistently poor recall.

The average time required to process each video frame varied between categories. In general, the Gaussian mixture model was found to be the slowest at an average of 0.143 ms per frame while the frame difference model was by far the quickest at an average of 0.041 ms per frame. Evaluation was also begun on a sixth model based on textures which was proposed by Heikkila and Pietikainen [79] but due to the significant processing time that was required by this model it was not possible to complete the evaluation. Based on the videos that were processed using this model, an average frame processing time of 2.048 ms was seen. A model's processing speed may dictate the applications in which it can reasonably be used.

The KDE model required the largest number of frames for training of all the evaluated models. This too can be of interest in selecting an appropriate model for an application. If no frames which are free of foreground are available for training it may be prudent to select a model that does not require much training. In addition, all of the models which were evaluated here exhibited no lag meaning that they may be used in both online and offline applications. As seen previously, lag can also be a very important factor in model selection.

7.2 Advantages of the Proposed Evaluation Framework

The proposed background model evaluation framework has a number of advantages over the way in which developer evaluations are currently being performed (see chapter 3) including the guarantee of comprehensive, objective and comparable evaluation results as well as a significant reduction in the workload of developers. The proposed framework also provides several advantages over the attempts that have previously been made at building a standard evaluation framework, namely those designed by Goyette et al. [4] and by Young et al. [5], such as the ability to easily update the evaluation dataset without sacrificing result consistency. These and the other main advantages of the proposed evaluation framework are discussed below.

7.2.1 Comprehensive Performance Evaluation

As was discussed in chapter 3, a true assessment of a model's capabilities cannot be carried out and thus a background model evaluation cannot be considered complete when using a substandard dataset. A suitable dataset must be large and diverse in terms of the scenarios that are depicted, video quality and video length amongst other characteristics, in order to ensure that it is representative of the challenges with which background models are likely to be faced. Such a dataset has not been created to date nor is it realistically possible to create in the small scale evaluations that developers normally carry out to test their models. As part of the proposed framework, a comprehensive video dataset which depicts all common challenges along with detailed ground truth would be provided and various types of evaluation metrics would be used to allow for models to be thoroughly assessed. The proposed dataset and metrics were described in detail in chapter 5.

7.2.2 Reduction in Developer Workload

By creating a framework to which developers can simply upload their background model code and have it comprehensively evaluated for them in an objective and fair manner that is accepted by the research community, the developers are relieved of having to carry out an evaluation to test their model's capabilities themselves. In addition, their model will be automatically ranked in terms of its performance against an extensive list of other background models. As was seen previously, the completion of a background model evaluation requires the investment of an enormous amount of time and effort in gathering videos and ground truth, implementing other background models for comparison etc. and, as discussed in chapter 3, they are not currently being completed to a sufficient standard. By using the proposed framework developers are no longer required to carry out their own evaluation potentially saving them a great deal of time and money. Developer work is also reduced in comparison to the standard systems previously proposed by Goyette et al. [4] and by Young et al. [5] as all results are obtained by the framework. Thus, the proposed framework allows developers to have their models evaluated more thoroughly than is currently possible in a manner that allows for comparison to numerous other models, with minimal work on their behalf. The use of the proposed framework will also ensure that the results achieved by the model will be accepted by the whole research community as being fair and accurate.

7.2.3 Easily Updated Evaluation Methodology

It was mentioned in chapter 3, that the existing background model evaluations, in particular, the attempts that have been made by Goyette et al. [4] and by Young et al. [5] to create a standard evaluation system, have the major disadvantage of being unable to update the evaluation

methodology that is used, i.e. video dataset, ground truth and evaluation metrics, without causing significant inconsistency in results, rendering them incomparable. To avoid inconsistency, developers would be required to rerun their models using the updated methodology and to submit the new result frames. This, however, is impractical and it is unreasonable to expect all developers who have, at some point, submitted a background model for evaluation, to undertake this work. As model code is submitted to the proposed evaluation framework they may simply be run using the updated methodology without having to contact the developers. This ensures that evaluation results remain complete, consistent and up-to-date.

7.2.4 Guarantee of Fair and Consistent Results

The existing attempts to create a system for background model evaluation, i.e. those of Goyette et al. [4] and Young et al. [5], are completely trust-based and their value is therefore dependent on the honesty of those who submit segmented frames for evaluation. By having developers submit the code for their models and running it as part of the framework it can be guaranteed that all evaluations are carried out in an identical manner ensuring that results are accurate, consistent and comparable. It can also be ensured that the same parameter values are being used for a model for each video of the dataset. In addition, by running all evaluations on a single computer and using a consistent evaluation methodology, models may be fairly assessed based on their memory usage and processing time as well as accuracy.

7.2.5 Extensive Resource of Background Model Performance Data

An additional advantage of the proposed background model evaluation framework is the facilitation of the creation of an extensive reference of evaluation data which describes the performance of a multitude of models. This resource would be beneficial in a number of scenarios including the evaluation of a newly created model, the selection of an appropriate model for a particular application and the analysis of the weaknesses of the existing body of background modelling algorithms to direct further development. The proposed framework is essentially a background model benchmarking facility and greatly simplifies the creation of such a resource.

7.3 Limitations of the Proposed Evaluation Framework

The proposed evaluation framework does, unfortunately, have a small number of limitations, namely the need for developers to adapt their background model implementations to meet the requirements of the framework as discussed in section 4.1, the existence of patented models, the code of which cannot be submitted and security concerns related to the execution of user provided code. These limitations are discussed below.

7.3.1 Modification of Existing Implementations

It was mentioned in chapter 4, that there are a number of requirements that must be met and some adaptations that must be made to background model implementations so that they will be compatible with the evaluation framework. The need for this work, however, may, potentially, discourage developers from submitting their models. For this reason, the requirements to be met and the API that must be used to interface with the proposed framework have been made as simplistic as possible in order to minimise the efforts that are required.

While it may be somewhat of an inconvenience to ensure that a background model implementation is compatible with the evaluation framework, it is significantly less work than would be necessary had the developer been required to carry out the evaluation themselves. In addition, the evaluation carried out by the proposed framework is likely to be far more thorough than is realistically possible in a developer created evaluation and the results will be accepted as being of a high standard and trusted by the research community making it more likely that the model will be used. For these reasons, it would be extremely beneficial for a developer to submit their background model to the proposed framework for evaluation despite the inconvenience involved in ensuring that the model is compatible with it.

7.3.2 Patented Algorithms

It was mentioned previously, that some background models may be patented meaning that developers will not be willing to submit their code for evaluation. Section 4.9 described a solution to this problem in the form of an alternative, downloadable framework that developers may use to locally evaluate patented models. This, however, is a limitation of the framework as the accuracy of evaluation results that are obtained in this way cannot be guaranteed nor can they be easily updated in the event of a change to the evaluation methodology. The volume of results obtained in this way should, ideally, be minimal.

7.3.3 Security Concerns

A third limitation of the proposed evaluation framework is the existence of some significant security concerns. As discussed previously, the framework requires the building and running of user submitted code. This presents potential for the submission of malicious code which may cause significant harm to the framework. It is essential, therefore, that a method of verifying that the submitted code is not harmful be put in place. In addition, the model submission form presents the potential for SQL injection attacks to be launched against the framework. The validation of form input is used to protect against this.

7.4 Project Successes

Overall, the project was very successful and fulfilled the intended objectives. An extensive analysis of the current state of background model evaluation was carried out and the issues which currently exist were identified. Based on this research, an evaluation framework was designed which addresses the current issues. It provides a method of comprehensively evaluating the capabilities of background models and does so in a manner whose integrity can be guaranteed. While a small number of limitations exist, the proposed framework is a significant improvement over the evaluations that have previously been carried out and has several advantages such as the guarantee of fairness, objectivity and accuracy that have not previously been possible to achieve.

The implemented proof of concept version of the evaluation framework functions as intended and demonstrates that the proposed framework is capable of achieving its goals and that it is therefore a feasible solution to the problem of background model evaluation. Due to time constraints, all desired aspects of the proposed framework could not be implemented. The work that remains to be completed is outlined in section 8.1. A lack of experience in working with many of the technologies that were used meant that the implementation of some parts of the framework unfortunately took longer than had been anticipated.

From testing the framework, it was found to be very easy to use. Minimal effort was required to adapt a model to be compatible with the framework and to submit that model to be evaluated. In addition, it can be seen from the evaluation results that were presented earlier in this chapter that the framework provides excellent insights to model performance. By extending the test dataset to the scale described in section 5.1 such insights may be gained about all aspects of model capabilities. The ability to easily obtain such insights renders the proposed framework extremely beneficial to the computer vision community. When examining model accuracy in section 7.1, the sample result frames, in addition to the calculated metric values, were found to be very useful in understanding model performance. The inclusion of some result frames on the framework website may therefore be a useful feature to add in order to provide this benefit to those using the site.

7.5 Evaluation of Software Characteristics

In this section, the proof of concept implementation of the proposed background model evaluation framework is assessed in terms of how well it exhibits the desired software characteristics that were discussed in section 6.1.

7.5.1 Extensibility

The proposed framework is quite extensible. This was clearly demonstrated by the way in which the framework was developed. Initially, a very simple version of the framework was created and, over time, different aspects of its functionality were added. Each added function served to enhance the framework and did not compromise the features that were already in place. In addition, the code that was written was well-commented which would be useful for anyone involved in the further development of the framework. It was also seen that the evaluation methodology can easily be updated without disrupting the framework which further demonstrates its extensibility.

7.5.2 Robustness

During development, the various aspects of the framework were not always sufficiently robust. Regarding the evaluation aspect, a number of faults were introduced which made it prone to crashing. It was ensured, however, that these were corrected and that the framework was stabilised before further development began. At the time of writing, there are no critical bugs in the framework.

As the evaluation framework comprises a number of distinct parts, an issue in one component will not necessarily affect the rest of the system, e.g. if an issue arises in the evaluation aspect of the framework the existing evaluation results can still be displayed on the framework website. This is a strength of the framework as all functionality will not be lost due to an issue in one aspect.

It was mentioned previously, that the robustness of the framework may be threatened due to it being required to work with a large amount of user supplied data. To address this, all data, other than the model code files, that is input via the model submission form is validated to ensure that the correct type of information is being supplied. There remains, however, scope for robustness to be improved in terms of validating that the code that is submitted can be properly handled by the framework. This may include the restriction of the file types that can be supplied.

7.5.3 Scalability

There is potential for the scalability of the evaluation framework to be improved, particularly in terms of the PHP database accesses that are performed by the website. This is of growing importance as the number of background models that are evaluated increases and thus as the number of queries that need to be made increases.

The use of MySQL and an Apache webserver in the framework implementation enhance its scalability. These have been used in many large-scale applications and will therefore be guaranteed to accommodate the growing needs of the evaluation framework.

7.5.4 Usability

The adaptation of background model implementations to work with the evaluation framework was quite straightforward. In addition, the framework website was simply designed and easy to navigate. There are, undoubtedly, however, aspects of the framework in which usability could be improved. The framework's usability could, in the future, be enhanced based on user feedback.

7.6 Overview

This chapter has assessed the design and the proof of concept implementation of the proposed background model evaluation framework in terms of its feasibility, ease of use, operation and the advantages and the limitations that exist. It was found that all objectives of the framework were met but that some aspects such as its security still require some consideration. In terms of the implementation, the framework functions as expected but the functionality of some areas could be extended. The possible extensions are described in section 8.1.

Chapter 8 Conclusion

The success of background subtraction, a widely used computer vision technique, is largely based on the performance of the background modelling technique that is used. In modelling a background, however, there are many challenges that must be contended with which hinder the performance of the modelling techniques. The performance of different background models is affected differently by different challenges and, in order to gain a comprehensive understanding and appreciation of their capabilities, it is essential that they be subject to a rigorous performance evaluation. Unfortunately, however, due to the enormous amount of effort that would be required to properly assess performance, no comprehensive or extensive background model evaluations have been carried out to date.

A number of small-scale background model evaluations have previously been performed but, as was learned from their analysis during this project, they exhibit numerous weaknesses and limitations. These evaluations do not provide comprehensive, objective or comparable assessment and their results, therefore, do not give a realistic depiction of model capabilities. In addition, the results of existing evaluations cannot be guaranteed to be credible or accurate and are thus not of great use to the research community. The poor state of current evaluations is a significant shortcoming of existing background modelling research.

As part of this project, a solution to the problem of background model evaluation was proposed in the form of a standard framework and associated methodology. The proposal was shown to be capable of the rigorous, objective and fair evaluation of background models in a manner that will allow them to be easily compared and ranked against one another and is thus of great benefit to the computer vision community. A system like this is essential in ensuring that extensive and comprehensive background model evaluation can become a reality. In addition, the ideas of the proposed system could be applied to the assessment of other types of computer vision algorithms.

8.1 Future Work

Due to time constraints, a complete implementation of the proposed evaluation framework could not be achieved and there is thus much potential for future work to be completed. The major aspects of this potential work, in no particular order of importance, are outlined below.

- The evaluation dataset should be expanded and extensive ground truth created for all videos to provide a comprehensive evaluation resource. The feasibility of using crowdsourcing to create the necessary ground truth should be considered. This resource may be updated and improved based on feedback from the research community.
- Additional functionality such as the ability to download evaluation results for use in other work should be added to the framework website to improve user experience and to increase the value that it provides. Further improvements to the functionality and the usability of framework website may be made based on user feedback.

- The implemented framework should be extended to consider the peak memory usage of the models that are evaluated.
- Consideration should be given to the security of the framework. It was discussed previously that there is potential for the framework to be attacked by way of malicious code being submitted and executed. To protect the framework it is important that all vulnerabilities be identified and addressed.
- Submitted code files should be validated to ensure that they have been correctly modified to interact with the evaluation framework.
- The compatibility of the proposed framework should be improved. For example, background models that have been coded using languages other than C++ could be facilitated.
- A method of ensuring that the same set of parameter values are used throughout the entire evaluation should be implemented.
- The alternative downloadable evaluation framework that was described in chapter 4 should be implemented so that patented models may be evaluated and to allow the testing of model adaptations before submission. This would increase the value and the appeal of the system.
- The framework should be made available for public use so that it can provide the intended benefits to the research community and to aid in the compilation of a comprehensive performance reference through the submission of models by the community.

8.2 Reflection on the Project

Prior to completing this project I had an interest in the area of computer vision but had only a high-level understanding of background modelling and the challenges that are faced and very little knowledge of the process and state of evaluation in this area. In completing the project, I have attained much knowledge and awareness of these matters and hope that this may be of use in projects that may be undertaken in the future.

In addition, the completion of this project allowed me to improve my technical skills. In particular, my knowledge of web development and of working with databases has improved as have my general programming abilities. During the implementation phase of the project it was necessary to learn about and use various new technologies which will be very advantageous in future work.

Through the completion of a project of this scale, the importance of thorough planning before attempting to implement a solution to ensure that work may be completed efficiently and to a high standard, was reinforced.

Supplements

A copy of this document is included on the accompanying CD. Also included is the project source code, a copy of the project presentation and some screenshots of the website that was created.

Bibliography

- [1] W. Tsai, L. C. M. Sheu, and H. Liao, "A robust background modeling and foreground object detection using color component analysis," presented at the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, 2012.
- [2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, p. 252 Vol. 2.
- [3] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," presented at the 6th European Conference on Computer Vision, Dublin, 2000.
- [4] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "changedetection.net: A new change detection benchmark dataset," presented at the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, 2012.
- [5] D. P. Young and J. M. Ferryman, "PETS metrics: on-line performance evaluation service," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005, pp. 317-324.
- [6] (29th January). *Performance Evaluation of Tracking and Surveillance (PETS) datasets, 2000-2009*. Available: <http://ftp.pets.rdg.ac.uk/>
- [7] (2011, 29th January). *CAVIAR Test Case Scenarios*. Available: <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>
- [8] (2009, 21st February 2014). *PETS 2009 Benchmark Data*. Available: <http://www.pets2009.net/>
- [9] (2007, 21st January). *PETS 2007 Benchmark Data*. Available: <http://pets2007.net/>
- [10] (2002 - 2005, 21st February 2014). *CAVIAR: Context Aware Vision using Image-based Active Recognition*. Available: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>
- [11] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequière, "A Benchmark Dataset for Outdoor Foreground/Background Extraction," presented at the Computer Vision - ACCV 2012 Workshops, 2012.
- [12] Background Models Challenge. (2012, 29th January). *Videos*. Available: <http://bmc.univ-bpclermont.fr/?q=node/6>
- [13] D. Gruyer, C. Royere, N. Du Lac, G. Michel, and J. Blosseville, "SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems," in *ITS World Congress, London, UK*, 2006.
- [14] ChangeDetection.net. (2012, 29th January). *Details of the dataset 2012*. Available: <http://www.changedetection.net/>
- [15] P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "IEEE Workshop on Change Detection in conjunction with CVPR 2012," ed, 2012.
- [16] ChangeDetection.net. (2014, 29th January). *Details of the dataset 2014*. Available: <http://www.changedetection.net/>
- [17] (2014, 10th March 2014). *IEEE Change Detection Workshop in Conjunction with CVPR 2014*. Available: <http://wordpress-jodoin.dmi.usherb.ca/cdw2014/>
- [18] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," presented at the Seventh International Conference on Computer Vision, Kerkyra, Greece, 1999.
- [19] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of Background Subtraction Techniques for Video Surveillance," presented at the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
- [20] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," presented at the 19th International Conference on Pattern Recognition, 2008. ICPR 2008., Tampa, Florida, 2008.

- [21] M. Harville, "A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models," in *Computer Vision — ECCV 2002*. vol. 2352, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., ed: Springer Berlin Heidelberg, 2002, pp. 543-560.
- [22] J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *Multimedia, IEEE Transactions on*, vol. 8, pp. 761-774, 2006.
- [23] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi, "Evaluation report of integrated background modeling based on spatio-temporal features," presented at the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, 2012.
- [24] F. Bashir and F. Porikli, "Performance evaluation of object detection and tracking systems," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2006.
- [25] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 21-26.
- [26] L. M. Brown, A. W. Senior, Y.-I. Tian, J. Connell, A. Hampapur, C.-f. Shu, *et al.*, "Performance evaluation of surveillance systems under varying conditions," in *Proceedings of IEEE PETS Workshop*, 2005.
- [27] (2014, 22nd February). *Mental Ray Standalone*. Available: <http://www.autodesk.com/products/mental-ray-standalone/overview>
- [28] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent patents on computer science*, vol. 1, pp. 32-54, 2008.
- [29] P. L. Rosin and E. Ioannidis, "Evaluation of global image thresholding for change detection," *Pattern Recognition Letters*, vol. 24, pp. 2345-2356, 10// 2003.
- [30] L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," *Image Processing, IEEE Transactions on*, vol. 17, pp. 1168-1177, 2008.
- [31] S. Herrero and J. Bescós, "Background Subtraction Techniques: Systematic Evaluation and Comparative Analysis," *Advanced Concepts for Intelligent Visions Systems*, vol. 5807, pp. 33-42, 2009.
- [32] L. Goldmann, T. Adamek, P. Vajda, M. Karaman, R. Mörzinger, E. Galmar, *et al.*, "Towards Fully Automatic Image Segmentation Evaluation," in *Advanced Concepts for Intelligent Vision Systems*. vol. 5259, J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 566-577.
- [33] D. S. Lee, J. J. Hull, and B. Erol, "A Bayesian framework for Gaussian mixture background modeling," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 2003, pp. III-973-6 vol.2.
- [34] J. S. C. Yuk and K. Y. K. Wong, "An efficient pattern-less background modeling based on scale invariant local states," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, 2011, pp. 285-290.
- [35] A. Tavakkoli, M. Nicolescu, and G. Bebis, "Automatic Robust Background Modeling Using Multivariate Non-parametric Kernel Density Estimation for Visual Surveillance," in *Advances in Visual Computing*. vol. 3804, G. Bebis, R. Boyle, D. Koracin, and B. Parvin, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 363-370.
- [36] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of gaussians for foreground detection-a survey," *Recent Patents on Computer Science*, vol. 1, pp. 219-237, 2008.
- [37] A. Prati, R. Cucchiara, I. Mikic, and M. M. Trivedi, "Analysis and detection of shadows in video streams: a comparative evaluation," in *Computer Vision and Pattern Recognition*,

2001. *CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. II-571-II-576 vol.2.
- [38] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari, "Evaluation of Background Subtraction Methods," in *Digital Image Computing: Techniques and Applications (DICTA), 2008*, 2008, pp. 357-364.
- [39] R. Pless, J. Larson, S. Siebers, and B. Westover, "Evaluation of local models of dynamic backgrounds," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003.
- [40] T. Tanaka, A. Shimada, D. Arita, and R.-i. Taniguchi, "Non-parametric Background and Shadow Modeling for Object Detection," in *Computer Vision – ACCV 2007*. vol. 4843, Y. Yagi, S. Kang, I. Kweon, and H. Zha, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 159-168.
- [41] A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu, "Non-parametric statistical background modeling for efficient foreground region detection," *Machine Vision and Applications*, vol. 20, pp. 395-409, 2009/10/01 2009.
- [42] S.-c. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," 2007.
- [43] A. Lanza, F. Tombari, and L. Stefano, "Second-Order Polynomial Models for Background Subtraction," in *Computer Vision – ACCV 2010 Workshops*. vol. 6468, R. Koch and F. Huang, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 1-11.
- [44] O. Barnich and M. Van Droogenbroeck, "ViBE: A powerful random technique to estimate the background in video sequences," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009, pp. 945-948.
- [45] B. White and M. Shah, "Automatically tuning background subtraction parameters using particle swarm optimization," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 1826-1829.
- [46] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," presented at the Proceedings of the eleventh ACM international conference on Multimedia, Berkeley, CA, USA, 2003.
- [47] Y. A. Sheikh. (2005, 22nd February 2014). *Background Modeling*. Available: http://www.cs.cmu.edu/~yaser/new_backgroundsubtraction.htm
- [48] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, pp. 88-97, 1/15/ 2009. (27th February 2014). *Motion-based Segmentation and Recognition Dataset*. Available: <http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>
- [49] The Imagelab Laboratory of University of Modena and Reggio Emilia. (29th January). *VISOR, Video Surveillance Online Repository*. Available: <http://imagelab.ing.unimore.it/visor/index.asp>
- [50] E. Grossmann, A. Kale, and C. Jaynes, "Towards interactive generation of" groundtruth" in background subtraction from partially labeled examples," in *Proc. ICCV VS-PETS workshop*, 2005.
- [51] (30th January). *ViPER: The Video Performance Evaluation Resource*. Available: <http://vipertools.sourceforge.net/>
- [52] D. o. A. I. T. Laoratory for Image and Media Understanding, Kyushu. (2008-2012, 29th January). *Dataset: Detection of Moving Objects*. Available: <http://limu.ait.kyushu-u.ac.jp/dataset/en/>
- [53] P. Arbelaez, C. Fowlkes, and D. Martin. (2007, 27th February 2014). *The Berkeley Segmentation Dataset and Benchmark*. Available: <https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>
- [54] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 898-916, 2011.

- [56] MIT Computer Science and Artificial Intelligence Laboratory. (2005, 1st March 2014). *LabelMe*. Available: <http://labelme2.csail.mit.edu/Release3.0/index.php>
- [57] C. Pantofaru and M. Hebert, "A Comparison of Image Segmentation Algorithms," Robotics Institute, Carnegie Mellon University, 2005.
- [58] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," presented at the Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, 2006.
- [59] J. C. SanMiguel, Marti, x, and J. M. nez, "On the Evaluation of Background Subtraction Algorithms without Ground-Truth," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, 2010, pp. 180-187.
- [60] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 2004, pp. 3099-3104 vol.4.
- [61] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The Pixel-Based Adaptive Segmenter," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 38-43.
- [62] (2012, 9th February 2014). *IEEE Change Detection Workshop in Conjunction with CVPR 2012*. Available: <http://www.changedetection.net/>
- [63] OpenCV. (2014, 15th March 2014). *OpenCV*. Available: <http://opencv.org/>
- [64] SQLAPI++. (2013, 15th March 2014). *SQLAPI++ Library*. Available: <http://www.sqlapi.com/>
- [65] A. Sobral, "BGSLibrary: An OpenCV C++ Background Subtraction Library," presented at the IX Workshop de Visão Computacional (WVC'2013), 2013.
- [66] A. Sobral. (2013, 20th March 2014). *bgslibrary: A Background Subtraction Library*.
- [67] Microsoft. (2008, 15th March 2014). *MSBuild*. Available: [http://msdn.microsoft.com/en-us/library/wea2sca5\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/wea2sca5(v=vs.90).aspx)
- [68] cplusplus.com. (2014, 5th March 2014). *<ctime> (time.h)*. Available: <http://www.cplusplus.com/reference/ctime/>
- [69] The PHP Group. (2014, 20th March 2014). *PHP*. Available: <http://ie1.php.net/>
- [70] The Apache Software Foundation. (2014, 20th March 2014). *Apache HTTP Server Project*. Available: <http://httpd.apache.org/>
- [71] Netcraft. (2009). *February 2009 Web Server Survey*. Available: http://news.netcraft.com/archives/2009/02/18/february_2009_web_server_survey.html
- [72] Netcraft. (2013, 20th March 2014). *June 2013 Web Server Survey*. Available: <http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>
- [73] M. Otto, J. Thornton, C. Rebert, and J. Thilo. (20th March 2014). *Togglable tabs*. Available: <http://getbootstrap.com/javascript/#tabs>
- [74] M. Meno. (2012, 20th March 2014). *dropzone.js*. Available: <http://www.dropzonejs.com/>
- [75] C. Bach. (20th March 2014). *tablesorter. Flexible client-side table sorting*. Available: <http://tablesorter.com/docs/>
- [76] MySQL. (4th February 2014). *About MySQL*. Available: <http://www.mysql.com/about/>
- [77] MySQL. *Download Connector/C++*. Available: <http://dev.mysql.com/downloads/connector/cpp/>
- [78] PHP. (15th March 2014). *MySQL Improved Extension*.
- [79] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 657-662, 2006.

Appendix A – Summary of Proposed Evaluation Methodology

Presented here is an overview of the characteristics of the methodology that should be used in background model evaluation.

Dataset:

- Real scenes should be depicted in all videos.
- Videos should vary from a few minutes to several hours in length. At least one 24 hour video depicting the natural changes in illumination over the course of a day should be included.
- Videos should exhibit significant variation in quality (e.g. stability and the amount of noise that is present) and resolution.
- Each of the challenges listed in Table 5.1-1 should be depicted in at least six videos.

Ground Truth:

- Pixel-based ground truth should be used.
- Ground truth frames should be available for every video frame.
- All ground truth frames should be created manually by at least five people and a consensus of their efforts should be considered as the final, accepted ground truth frames.
- Uninteresting background motion and shadows cast by background objects should be considered background.
- People and other moving objects should be considered foreground.
- Shadows cast by foreground objects should be labelled as shadow to distinguish them from the foreground and the background.
- Pixels which are difficult to classify, e.g. at foreground object boundaries, should be labelled to indicate this. These pixels should not be included in metric calculations.
- Each pixel in every ground truth frame should be given one of four labels – background, foreground, shadow or unknown

Evaluation Metrics:

- Accuracy metrics
 - Precision
 - Recall
 - F-measure
- Efficiency metrics
 - Average frame processing time
 - Peak memory usage
- Length of training phase
- Maximum lag
- All evaluation metrics should be calculated for each model for each video.