Fast and Robust Global Motion Estimation in Video Object Segmentation

Bin Qi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Electrical and Computer Engineering) at
Concordia University
Montréal, Québec, Canada

March 2005

Your file Votre référence
ISBN: 0-494-04388-1
Our file Notre référence
ISBN: 0-494-04388-1

ABSTRACT

Fast and Robust Global Motion Estimation in Video Object Segmentation

Bin Qi

To meet the growing requirements of different video applications such as video surveillance or coding, many video processing techniques have been developed to analyze and represent video sequences. Video object segmentation is an object-based video processing technique which aims to detect semantically meaningful components, i.e., objects, in a video sequence. In case the video sequence contains global (camera) motion, global motion estimation is required to compensate the camera motion before segmentation.

This thesis studies methods to automatically segment moving objects in the presence of camera motion without user interaction. It proposes a fast and robust global motion estimation method oriented to video object segmentation. In addition, it integrates this method into a modular scheme to segment objects in the presence of camera motion. This video object segmentation scheme consists of three main steps: global motion detection, global motion estimation and compensation, and object segmentation. The object segmentation is based on: change (motion) detection, temporal adaptation, and edge adaptation. Some improvements are proposed in each part of the object segmentation.

The proposed methods aim at four goals: automatically adapt to camera motion, robust (insensitive) to noise and artifacts, temporally stable segmented objects and low computational cost. The proposed methods are reliable which was confirmed by experimenting on more than 10 indoor and outdoor video shots both with and without camera motion. Simulation results show that the proposed GME method achieves more satisfactory results than the reference methods. For object segmentation, encouraging results are also achieved.

Acknowledgments

I would like to express my deepest appreciation to my supervisor Dr. Aishy Amer, for her wise suggestions, continuous support, and great patience during my research as well as the active helps to improve the text of this thesis. I would also like to thank all the professors with whom I have interacted during my studies at Concordia University.

My special thanks to members of the Vidpro group, Dr. Carlos A. Vazquez, Kenneth Ryan, Firas Achkar, Francois Lapalme, and Mohamed El-Hilali, for useful advices and comments.

My deepest gratitude to my parents, my lovely daughter, and all my relatives, especially my brother Jin Qi and my cousin Shuwei Huang. Without their love and support I would never have reached this level. My special thanks go also to Zhige Chen, Hui Song, Ying Yu, Zhen Yang, Yiexin Wang, Meiyu Qin, Yuyong He, David Claveau, and Joel Worrell.

This work was supported, in part, by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Table of Contents

Chapter 1

Introduction

The demand of visual information has increased tremendously in everyday life and in professional area. For example, Statistics Canada show that Canadians spend an average of 21.6 hours per week watching television in 2002 [2]. Video games and internet movie have become increasingly popular among younger generations. Video conferencing takes place around the different parts of the world. Besides, there will be a growing need for this information in the future, e.g., video retrieval will allow us to efficiently search for various types of video documents of interest to ourselves. To meet the requirements of these different video applications, various video processing techniques have developed to analyze and represent video. Recently, content-based video processing is becoming increasingly important.

This thesis proposes a fast and robust method for content-based video processing. This method aims at estimating camera motion to facilitate object segmentation. Furthermore, this method is integrated into a modular object segmentation scheme. This thesis proposes some improvements within this scheme. Several standard video sequences, both with and without camera motion are used to test this scheme.

1.1 Background

A video is a sequence of two dimensional (2D) images projected from a dynamic three dimensional (3D) scene onto the image plane of a video camera [39]. A video sequence usually contains thousands of shots. A shot is a (finite) sequence of images recorded contiguously (usually without viewpoint change) and represents a continuous, in time and space, action or event driven by moving objects. In the remainder of this thesis, the term *video* refers to a video shot. The objective of video processing is to represent the video with a reduced amount of data that contains important information. The question is: which information is important?

To answer the above question, one has to take the consideration of the characteristics of human visual system (HVS). For example, HVS attracts more to moving objects and their features than to still objects and the background. More precisely, HVS focuses on high-level features (e.g., object, event) more than low-level features (e.g., shape, texture) [39]. Thus important information is objects and motion. The development of video processing field also proves this assumption [3]: In MPEG-1 and MPEG-2, coding is low-level pixel or block based. In MPEG-4, the content-based concept is imported. In video surveillance, object-based video processing aims to meet the requirement of detecting objects and their behavior automatically.

Video object segmentation (VOS) is a challenging problem because of the complex content in natural video scene, e.g., objects are rarely composed of only one homogeneous characteristic, either in motion, color or luminance. For instance, a human body can wear clothes of different colors and walk with different moving parts. Furthermore, noises such as shadow or luminance change also affect the correct segmentation. Another scenario to be considered is that the video sequence contains a moving camera where the main issue is how to separate moving objects from the changing background. In addition, many applications have real-time requirement (e.g., videophone), an efficient and reliable method for VOS is very desirable.

VOS becomes a very active research field in video processing, not only because of its importance, but also because of its difficulty.

1.2 Motivation

The ideal goal of VOS is to identify the semantically meaningful components of a video frame and group the pixels belonging to such components [1]. The MPEG-4 standard [3] introduces the new content-based concept and framework but leaves algorithms for the segmentation of video into separate video objects as an open topic. Many different VOS methods have been presented in the literature [15]- [27]. Some of them use computational intensive techniques to achieve accurate results. Without real-time consideration, a content-based video representation approach could lose its applicability. Many of the efficient methods can not handle complicated situations (e.g., moving camera).

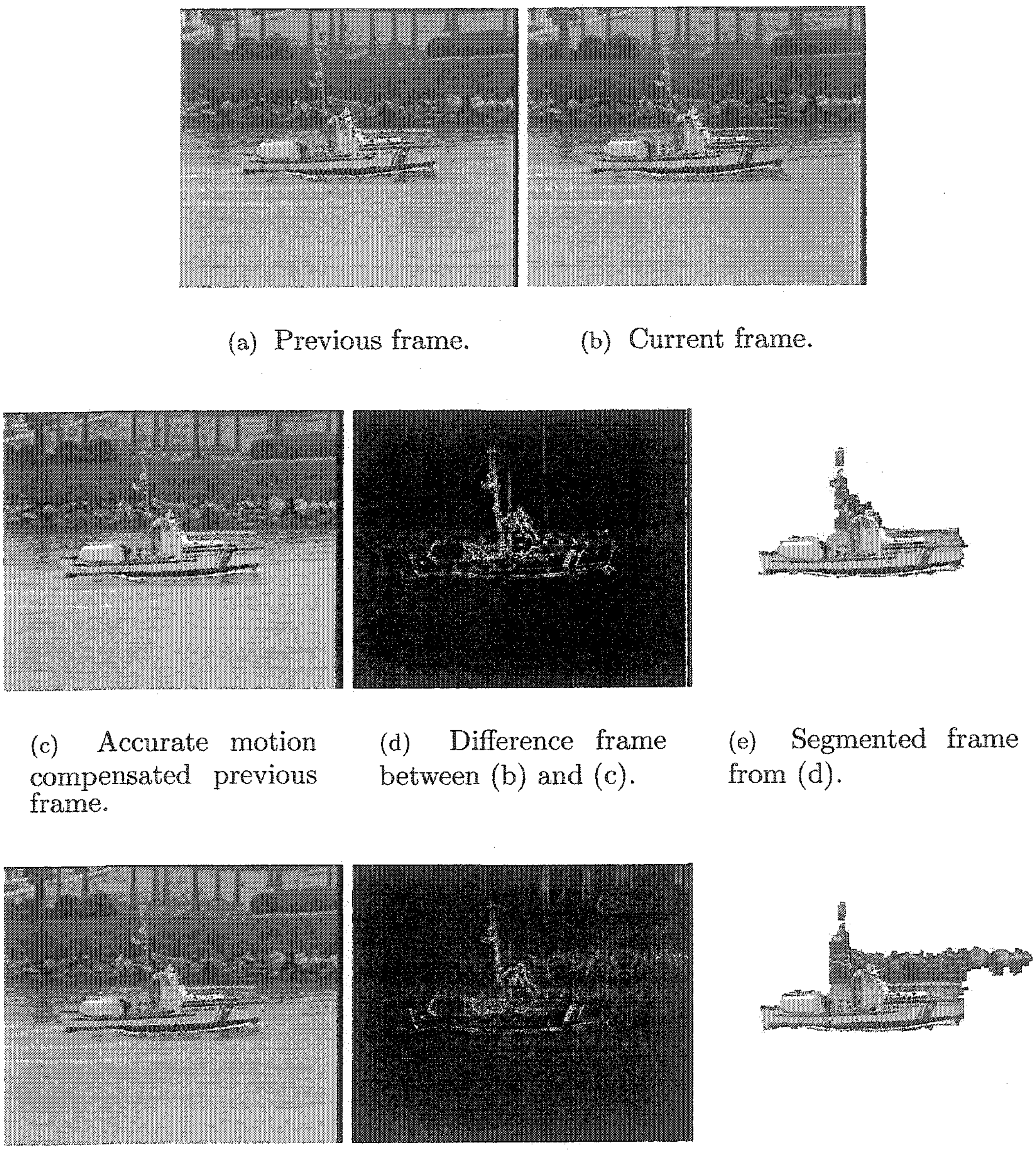
Since motion plays an important role in video, VOS is closely related to another problem, motion estimation. In general, motion can be classified as global motion and local motion. The term *global motion* (GM) is used in this thesis to describe the apparent 2D motion introduced by camera motion. It can be parameterized by a motion model. The process to estimate these parameters is known as global motion estimation (GME). GME is usually followed by other tools, such as global motion compensation (GMC).

GME has many applications, such as sprite generation, video coding, scene construction and VOS. Depending on different applications, the objective of GME is also different. The objective of GME for video coding is to remove the GM redundancy resulting in high coding efficiency. This technique can be found in MPEG-4 verification model (VM) [3].

In that model, block-based motion estimation and compensation are employed to explore the temporal redundancies of the video content. Each macro-block is selected to use GMC or local motion compensation (LMC) depending on the sum of absolute difference (SAD) associated with GME or local motion estimation (LME). Usually, the one with less SAD is chosen. In applications of VOS, the objective of GME followed by GMC is to estimate and compensate GM, and then extract the objects based on the motion compensated previous frame and the current frame. Thus, the results of GME in VOS must be accurate. Fig. 1.1 shows some examples of motion compensation and segmentation results. To demonstrate the accuracy of GME, the absolute difference frame between the current frame and the motion compensated previous frame is included in Fig. 1.1. (Note that the difference frames are brightened four times for visual attention.) Comparing Fig. 1.1 (d) and (g); (e) and (h), we can see that an accurate GME result can be used to successfully separate the background and the objects, giving a satisfactory object mask; while an inaccurate result still contains some background information and the objects are difficult to detect based on this result. Furthermore, in video coding, even if GMC fails, LMC can be used to maintain the high coding quality. While LMC is avoided in VOS because we aim at compensating the background motion and retaining the objects.

After a survey in the literature, we found that most GME methods focus on video coding. In this case, the computed motion need not resemble the *true* motion of frame points as long as some minimum bit rate is achieved (for a given video quality) [40]. Meanwhile, most VOS methods either assume that there is no GM or directly adopt a coding-oriented GME method.

Computational complexity is always a challenge in GME. More accuracy usually means



(a) Previous frame.

(b) Current frame.

(c) Accurate motion compensated previous frame.

(d) Difference frame between (b) and (c).

(e) Segmented frame from (d).

(f) Inaccurate motion compensated previous frame.

(g) Difference frame between (b) and (f).

(h) Segmented frame from (g).

Figure 1.1: Examples of motion compensation and segmentation results. Comparing (d) and (g); (e) and (h), it can be seen that accurate GME can successfully compensate the background differences, resulting satisfactory object mask; while inaccurate GME still remains some background residuals, and the segmented object is not accurate.

extra computational burden. Some GME techniques have to sacrifice certain quality to gain the speed which might not be suitable for VOS.

1.3 Overview of the Proposed Methods

The objective of this thesis is to study methods which can automatically segment moving objects in the presence of GM without user interaction. This thesis proposes a GME method oriented to VOS. In addition, it integrates this GME method into a VOS scheme. Some improvements are proposed in this scheme (see section 1.4). Simulations of the VOS scheme should result in temporally stable moving objects for various ranges of video sequences containing different contexts.

The proposed methods in the VOS scheme are oriented to the following requirements:

1. automatically adapt to GM.

2. robust (insensitive) to noise, artifacts and clutter.

3. stable segmented objects over time.

4. low computational cost.

To achieve these requirements, the VOS scheme consists of three parts: global motion detection, global motion estimation and compensation, and object segmentation (See Fig. 1.2).

- **Global motion detection** GM detection aims at detecting the existence of GM (usually caused by a moving camera) between the current frame I_n at time instant n and the previous frame I_{n-1}. A fast and noise robust GM detection technique is presented.

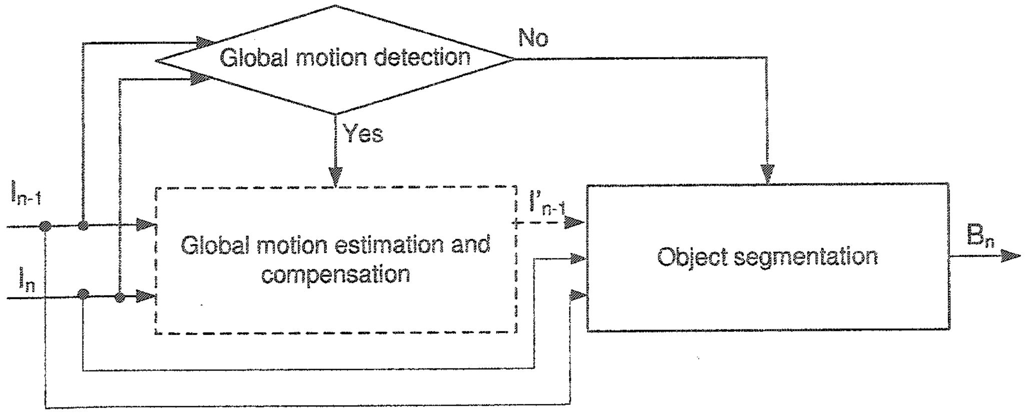


Figure 1.2: Block diagram of the proposed method. I_n is the current frame at time instant n, I_{n-1} is the previous frame at time instant $n-1$, I'_{n-1} is the GM compensated previous frame, and B_n is a binary frame of the segmented objects.

- **Global motion estimation and compensation** If GM is detected, a fast and robust GME and GMC method is applied to obtain the compensated frame I'_{n-1}. This GME method is based on hierarchical differential approach[1]. A combination of 3-step search and motion vector (MV) prediction is proposed for initial estimate. Two robust estimators are also proposed: to estimate GM in the first frame and to reject outliers using objects information.

- **Object segmentation** The VOS scheme is based on change detection either between I_n and I_{n-1} (if there is no GM) or between I_n and I'_{n-1} (if there is GM). A noise robust binarization method [28] is adopted for thresholding the detected changes. An improved thresholding technique is proposed to handle sequences with heavily cluttered background. A new temporal adaptation technique is used to stabilize the segmented results over time. Some post processing tools are used to remove artifacts and clutter and to complete the objects. Finally, edge detection and warping is added to get precise object boundaries.

[1] See section 2.5 for the details of this approach.

During each part, computational cost is of a concern to achieve the efficiency of the whole method. Furthermore, noise and artifacts adaptation is achieved at each processing level.

1.4 Contributions Overview

The following list states which parts of this thesis are original to the knowledge of the author at the time the proposed methods of this thesis have beeen developed:

- a fast and noise robust GM detection method which detects GM without estimating the GM parameters.
- a fast and robust GME method oriented to VOS using
    1. a combination of 3-step search and MV prediction for initial estimate,
    2. residual information from the previous frame for robust estimation, and
    3. a new robust estimator considering the neighborhood.
- improved segmentation methods as follows:
    1. an improved morphological double thresholding technique to remove background clutters,
    2. a new temporal adaptation technique to obtain stable results, and
    3. a new edge warping technique to improve the accuracy at object boundaries.

In addition, various reference methods of GME and VOS were studied, implemented, and their performance analyzed to compare with the proposed methods. An objective performance measure [38][2] is also co-implemented to objectively measure the performance of both

[2] See appendix A for the details of this measurement.

the reference and the proposed methods.

1.5 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 discusses related work to GME and proposes a GM estimation method and a GM detection method. Chapter 3 introduces related work to VOS and presents the VOS scheme used and the improvements in this scheme. Chapter 4 contains the simulation results and discussions. Chapter 5 concludes this thesis. Appendix A explains an objective performance measure for VOS [38]. Appendix B summarizes the abbreviations used in this thesis. Appendix C contains a publication related to Chapter 2 of this thesis. List of Figures and Tables are included at the end of the thesis. Note that related work to GM detection, GME, and VOS are given in the related chapters: section 2.3.1, section 2.1, and section 3.1, respectively.

Chapter 2

Global Motion Estimation for Object Segmentation

This chapter proposes a fast and robust GME method which is oriented to VOS. This method combines some basic GME principles for video coding and adds several improvements for VOS. Furthermore, this method is adaptive to different kinds of camera motion and to different size (CIF/SIF/PAL) of video sequences.

This chapter is organized as follows. Section 2.1 reviews general approaches in GME. Section 2.2 introduces several motion models for GME. Section 2.3 proposes a new GM detection technique. Section 2.4 explains one reference GME method [8]. Section 2.5 explains the other reference GME method [9]. Section 2.6 presents the proposed GME method. Section 2.7 introduces bilinear GMC technique. Section 2.8 summarizes the chapter.

2.1 Related Work

GME is one of the most widely used methods in video processing. Many approaches have been developed. Broadly, GME can be classified into three categories: phase correlation approach [4, 5], background matching approach [6, 7, 8], hierarchical differential approach [9, 10, 11].

Phase correlation approach assumes that there is a global translation between consecutive frames at the block level. Frames are first transformed from spatial domain to frequency domain using Fourier transform, then take the advantage of Fourier shift theorem; the translation part between two frames can be identified. The advantages of using phase correlation are its fractional-pel accuracy and its insensitivity to illumination changes compared to displaced frame difference (DFD) based method [39]. The limitation of this approach is that the translational model is not always suitable for general video sequences. A complement can be found in [5] where phase correlation is used as coarse estimation followed by a refinement in spatial domain.

Background matching approach is proposed in [6, 7, 8]. This approach is based on block match algorithm (BMA)[1], but generalized to the whole background. In [7], motion vectors for each block are firstly found using BMA. Then four global motion parameters are estimated based on these vectors. If the Euclidean distance between two matching points derived using BMA and GME respectively is beyond a threshold, then this block is considered as foreground block and will be eliminated. After that, feature points are selected from the remaining blocks. Finally, eight GM parameters are refined using the selected feature points. In [8], a confidence measure is assigned to refine the motion parameters obtained from the same BMA algorithm. The confidence measure assignment combines cornerness measure (using Plessy corner detector) and distinctness measure (using SUSAN edge detector) as a weighting function for each block. The author states that it can successfully assign low weight to outliers' blocks. Background matching approach is easy to implement, but it may lose some detail information since it is block based. Another disadvantage for this approach is that BMA is a time consuming task.

Hierarchical differential approach [9, 10, 11] is an efficient and effective tool for GME

[1] See section 2.4 for the details of BMA.

which is suggested in MPEG-4 frame work [3]. Hierarchical implementation can handle large search range and speed up computations. In hierarchical differential approach, a frame pyramid is built using spatial pre-filtering (e.g. Gaussian) and sub-sampling. The computation starts at the top level with an initial estimation. Then, iterative GM parameter optimization (e.g., gradient descent method) is performed to refine the estimation until a convergence criterion is met. The result is projected onto the lower level of the pyramid and the parameter optimization is repeated. This loop is continued until the bottom of the pyramid is reached[2]. Since GME is a computational intensive task, many efforts focus on accelerating the computational speed. [10, 11] are modified faster versions of [9]. In [10], history GM information is used as predictors instead of traditional N-step search in initial motion estimate (see section 2.5.2). In [11], several improvements are proposed such as motion edge selection, residual-block based outliers' rejection and adaptive weight function.

Hierarchical differential approach has many advantages, such as its large search range and fast convergence. This thesis proposes a fast and robust GME method based on hierarchical differential approach. The proposed method consists of three steps: frame pyramid construction, initial motion estimate, iterative motion parameter optimization using gradient descent method. Contributions of the proposed GME method are: 1) fast initial estimate using a combination of 3-step search [12] and motion vector prediction [10], 2) robust estimation[3] using residual information from the previous frame, 3) a new robust estimator considering the neighborhood to eliminate outliers. Compared to the reference methods [8] and [9], the proposed method is more accurate and more efficient. Another distinctive difference between the proposed method and coding-oriented GME methods is

[2] Section 2.5 will explain the details of this method.

[3] See section 2.5.4 for the definition of robust estimation.

that it is oriented to the requirement of video object segmentation.

2.2 Motion Models and Estimation Criteria

2.2.1 Motion models for GME

The purpose of a motion model is to describe the motion between consecutive frames of a real video sequence. Using a parametric model, we are able to estimate the parameters of the parametric model and reconstruct the frame that is an approximation of the real world [39]. In GME, one single model applies to the whole frame. There are different parametric models used to estimate GM. Depending on selected model, we can represent the real world with more or less detail and precision.

Usually, an 8-parameter perspective model (Eq. 2.1) is sufficient for GME [9].

$$\begin{aligned} x_i' &= \frac{(a_0 + a_1 x_i + a_2 y_i)}{(a_6 x_i + a_7 x_i + 1)} \\ y_i' &= \frac{(a_3 + a_4 x_i + a_5 y_i)}{(a_6 x_i + a_7 x_i + 1)} \end{aligned} \tag{2.1}$$

where (x_i, y_i) denotes the i^{th} pixel in the current frame, (x_i', y_i') denotes the corresponding pixel in the previous frame, and $\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ are the GM parameters. Furthermore, several simplified models can be derived from Eq. 2.1 [9]:

- 6-parameter affine model

$$\begin{aligned} x_i' &= a_0 + a_1 x_i + a_2 y_i \\ y_i' &= a_3 + a_4 x_i + a_5 y_i \end{aligned} \tag{2.2}$$

- 4-parameter translation-zoom-rotation model

$$\begin{aligned} x_i' &= a_0 + a_1 x_i + a_2 y_i \\ y_i' &= a_3 - a_2 x_i + a_1 y_i \end{aligned} \tag{2.3}$$

- 3-parameter translation-zoom model

$$\begin{aligned} x_i' &= a_0 + a_1 x_i \\ y_i' &= a_2 + a_1 y_i \end{aligned} \tag{2.4}$$

- 2-parameter translation model

$$\begin{aligned} x_i' &= a_0 + x_i \\ y_i' &= a_1 + y_i \end{aligned} \tag{2.5}$$

In this thesis, 6-parameter affine model (Eq. 2.2) is selected since the projected 2D motion of most camera motions can be described by this model [39].

2.2.2 Estimation criteria

The model discussed needs to be incorporated into an estimation criterion. Most of the criteria arise from the *constant-intensity* assumption. Since motion is estimated (and observed by the human eye) based on the variations of intensity and/or color, we can reasonably assume that the intensity remains constant along a motion trajectory [40]. Upon this assumption, a difference error e_i between the intensity value of the current frame I_n at time instant n and the motion compensated previous frame I'_{n-1} is defined in Eq. 2.6. And the estimation criterion is defined to minimize the estimation error, which is either the sum of square differences (SSD) (Eq. 2.7) or the sum of absolute differences (SAD) (Eq. 2.8). The summation in Eq. 2.7 and Eq. 2.8 is carried out over the number of pixels N in the frame.

$$e_i = I'_{n-1}(x_i', y_i') - I_n(x_i, y_i) \tag{2.6}$$

$$SSD = \sum_{i=1}^{N} e_i^2 \tag{2.7}$$

$$SAD = \sum_{i=1}^{N} |e_i| \tag{2.8}$$

2.3 Global Motion Detection

2.3.1 Related work

GM should only be estimated and compensated if a GM has been detected. To detect GM, the method in [26] estimates 8-perspective GM parameters (Eq. 2.1). If the absolute value of one of the estimated motion parameters a_0, $(a_1 - 1)$, a_2, a_3, a_4, $(a_5 - 1)$, a_6, and a_7 is greater than 2.5, then GM is detected. This detector is not very reliable since it will fail if GME fails. Furthermore, it is not efficient since GME is a time consuming task.

2.3.2 Proposed GM detection technique

In this section, a fast and noise-robust GM detection technique is presented. This technique can detect GM without estimating the GM parameters.

First, the binarization method [28] including change detection and thresholding is applied between the current frame and the previous frame. Then the obtained binary frame is divided into nine equal blocks (see Fig. 2.1). In each block, the number of white pixels is counted. If this number is greater than 15% of the total pixels in this block, this block is considered as a moving block b_m. If there is GM during consecutive frames, the moving blocks are distributed throughout the whole frame; otherwise the moving blocks are concentrated at the location where objects are moving. After observation of real video sequences, we found that objects are most likely to stay in the center of the frame, sometimes they are moving into one side of the frame. Thus, different weights $w_{b_i}, i \in 1, 2, \cdots, 9$ is assigned to each block according to its position (see Fig. 2.1). The highest weight is assigned to the blocks where the GM is most likely to occur, while the lowest weight is assigned to the center block where objects motion is most likely to occur. Finally, the sum of weighted moving blocks b_s is calculated for each frame using the following equation:

$$b_s = \sum_{i=1}^{9} w_{b_i} \times b_i, \qquad b_i = \begin{cases} 0 & : \; N_{w_b} \leq 15\% \times N_b \\ 1 & : \; N_{w_b} > 15\% \times N_b \end{cases} \tag{2.9}$$

where N_b is the total number of pixels in each block, N_{w_b} is the number of white pixels in each block. Simulations show that if we set up a threshold $t_d > 10$ for b_s, GM can be successfully detected.

4	2	4
2	1	2
4	2	4

Figure 2.1: Kernel for GM detection with different weights.

In case the sequence is noisy, the noise might interfere with the results. For example, a noisy sequence with a still background may cause the binary frame to contain moving blocks in background areas, misleading that the background is moving. Since the binarization method [28] uses a spatial average filter to adapt to noise, the following solution is proposed to handle GM detection in case of noisy sequences. First, the noise estimation method [13] is used to estimate the noise standard deviation or the peak signal-to-noise ratio ($PSNR_n$) of a frame. Then the window size W_f of the spatial average filter is adjusted as follows:

$$W_f = k_s / PSNR_n \tag{2.10}$$

where k_s is set 120 through experiments in this thesis. Finally, W_f is rounded to an odd integer.

2.3.3 Results

Table 2.1 shows the GM detection results of noisy *Hall* test sequences (no GM) using different W_f of the spatial average filter according to Eq. 2.10. The average value of b_s for the whole sequence as well as the percentage of the false detection are presented in Table 2.1. The percentage of the false detection is the percentage of GM falsely detected frames for the sequence without GM and vice versa. It can be seen that if we fix the window size of the spatial average filter as 3×3 (the regular size in [28]), the false detection is higher in noisy sequence with $PSNR_n = 25dB$. However, after we adjust the window size of spatial average filter according to Eq. 2.10, the percentage of false detection is reduced to 0.

$PSNR_n$	W_f	*average* b_s	*false detection rate*
45dB	3×3	0.08	0%
30dB	3×3	0.08	0%
25dB	3×3	2.6	0.05%
	5×5	0.07	0%

Table 2.1: Results of GM detection for noisy *Hall* sequences.

Table 2.2 gives sample results using the proposed GM detection technique. It can be seen that the proposed method has a good performance to detect GM. To compare our GM detector, we have applied the GM detector in [26]. The detector in [26] fails to detect little GM. For example, for *Coastguard* test sequence, the proposed method has 0% false detection compared to 83% in [26]. The average computational time for the proposed GM detection method is 0.09 sec/frame for the CIF/SIF sequences and 0.22 sec/frame for the PAL sequences. It is about 8 times faster than [26].

sequence name	$PSNR_n$	W_f	*average* b_s	*false detection rate*	*with/no GM*
Miss_america	45dB	3×3	0.68	0%	no GM
Survey	42dB	3×3	3.73	0.04%	no GM
3cars	44dB	3×3	4.97	0%	no GM
Hall	45dB	3×3	0.08	0%	no GM
	30dB	3×3	0.08	0%	
	25dB	5×5	0.07	0%	
Stefan	50dB	3×3	18.92	0.02%	with GM
Coastguard	37dB	3×3	16.88	0%	with GM
Marble	45dB	3×3	20.90	0%	with GM
Tennis	45dB	3×3	14.16	0.09%	with GM
Flowergarden	52dB	3×3	20.05	0%	with GM
	30dB	3×3	19.28	0%	
	25dB	5×5	17.51	0%	

Table 2.2: Sample results of the proposed GM detection.

2.3.4 Discussion

To make the proposed GM detection technique more generic, we can apply it every k frames for online process. For offline process, detect the mean value of k frames is more reliable.

2.4 A Background Matching GME method - Reference Method 1

This section summarizes a GME method using background matching approach [8] that consists of four steps (see Fig. 2.2):

1. Estimate block-based motion vectors using block match algorithm (BMA).
2. Estimate initial GM parameters.
3. Assign confidence measure to each vector.
4. Iteratively refine GM parameters by removing error vectors.

BMA is a motion estimation algorithm that works as follows: First, both the current and the previous frame are divided into non-overlapping small regions, called blocks. Then

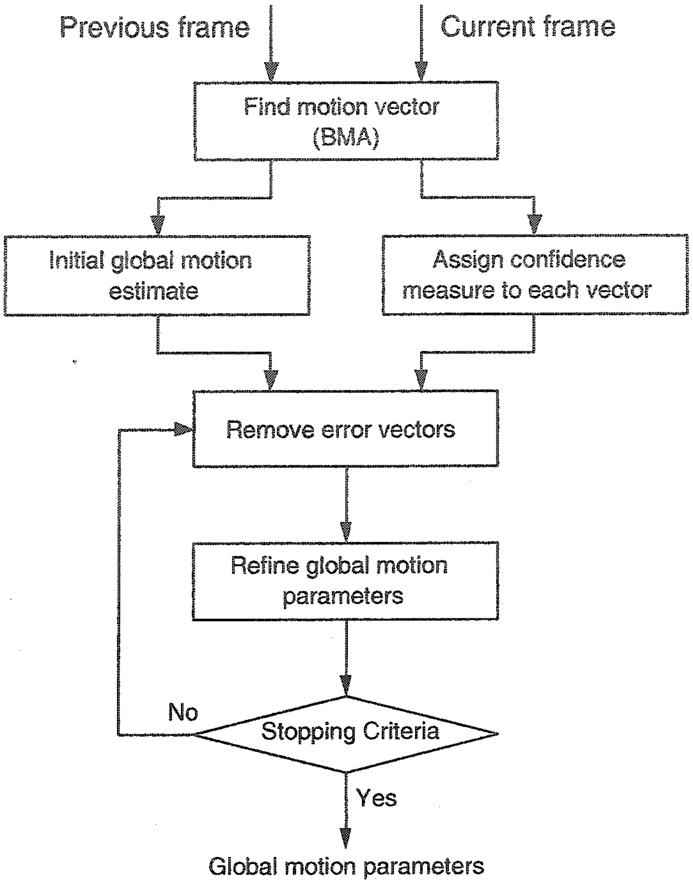


Figure 2.2: Block diagram of the reference GME method [8].

a motion vector (MV) is found for each block in the current frame which is the displacement vector between the spatial positions of this block and its matching block in the previous frame. The matching criterion is the minimization of the sum of absolute difference SAD_b as in Eq. 2.11 between these two blocks [39].

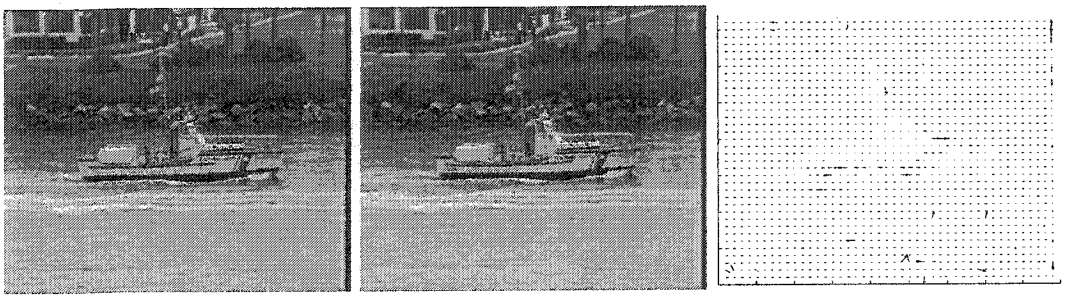
$$SAD_b = \sum_{i=1}^{N_b} |e_i|, \quad e_i = I_{n-1}(x_i', y_i') - I_n(x_i, y_i) \tag{2.11}$$

where N_b is the total number of pixels in each block.

In [8], the block size is 8×8 pixels ($N_b = 64$ in Eq. 2.11), and the total number of block is $N_m = (Rows \times Cols)/64$. Fig. 2.3 shows an example result of MV field using BMA. After the MVs for each block are obtained, initial GME is executed using a 6-parameter affine model as follows:

$$\begin{aligned} v'_x(x_m, y_m) &= a_0 + a_1 x_m + a_2 y_m \\ v'_y(x_m, y_m) &= a_3 + a_4 x_m + a_5 y_m \end{aligned} \qquad (2.12)$$

where (x_m, y_m) denotes the position of m^{th} block. $v'_x(x_m, y_m), v'_y(x_m, y_m)$ denote the estimated horizontal and vertical MVs respectively. $\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5)$ are the GM parameters.



(a) Previous frame. (b) Current frame. (c) MV field.

Figure 2.3: An example result of MV field using BMA.

From Fig. 2.3(c) we can see that the MV field contains local and error vectors which will bias the GME results. To remove those error vectors, a confidence measure algorithm is assigned to each vector. This algorithm consists of two detectors: Plessy corner detector [31] for cornerness value C_m and SUSAN edge detector [32] for distinctness value L_m of each block. Then the confidence measure is defined as follows:

$$w_m = \frac{(C_m + \lambda L_m)}{(C_{max} + \lambda L_{max})} \qquad (2.13)$$

where

$$C_{max} = \max(C_m), \quad L_{max} = \max(L_m) \quad m = 1, \ldots, N_m \qquad (2.14)$$

and λ is a coefficient to normalize C_m and L_m to the same level.

Let $v_x(x_m, y_m), v_y(x_m, y_m)$ denote the BMA result of the horizontal and vertical MVs for m^{th} block, respectively. The GM parameters are refined iteratively by minimizing the following estimation criterion:

$$E(\mathbf{a}) = \sum_{m=1}^{N_m} \{w_m([v_x(x_m, y_m) - v_x'(x_m, y_m)]^2 + [v_y(x_m, y_m) - v_y'(x_m, y_m)]^2)\} \qquad (2.15)$$

2.5 A Hierarchical Differential GME Method - Reference Method 2

In this section, a GME method using hierarchical differential approach [9] is explained. It consists of the following three steps (see Fig. 2.4):

1. Construct the low-pass frame pyramid using down-sampled Gaussian filter.

2. Estimate the initial motion parameters at the top level of the pyramid using 3-step search matching.

3. Execute gradient descent method from the top to bottom level of the pyramid to optimize the estimate result.

2.5.1 Hierarchical representation to build the frame pyramid

Hierarchical representation, or multi-resolution representation, is a widely used strategy in image processing. Using this representation, the original frame is rebuilt like a pyramid (see Fig. 2.5). The bottom level is the original frame. Then the resolution is reduced by half, both horizontally and vertically, between successive levels. Before reducing, a low-pass filter (e.g., Gaussian filter) is employed here to reduce the noise and smooth the output frame. After the frame pyramid has been built, the estimation starts at the top (coarsest)

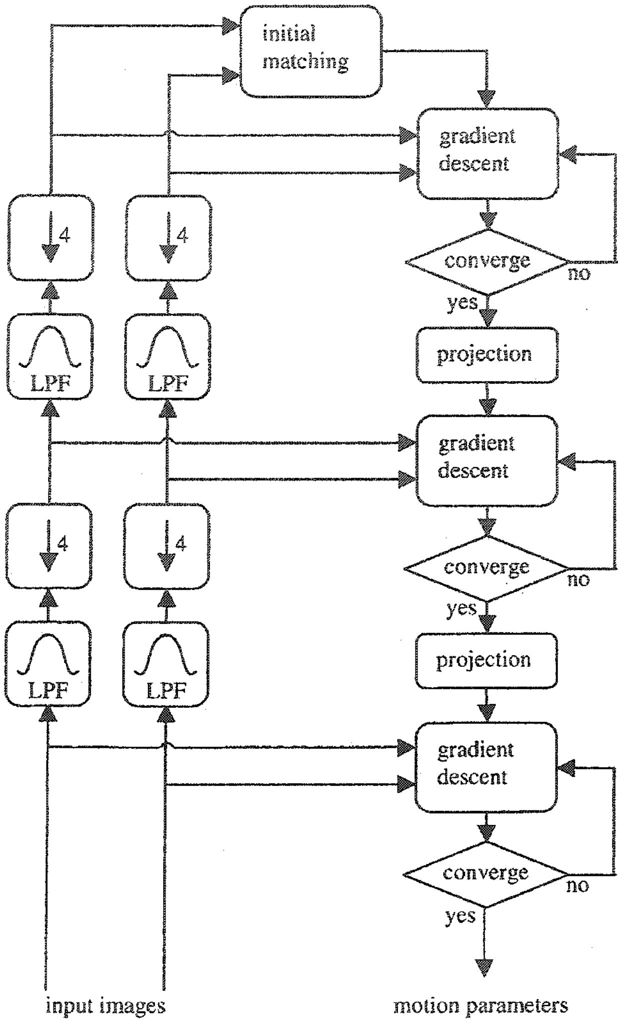


Figure 2.4: Block diagram of the reference method [9].

level and progress to the next finer level until it reaches the bottom level. The result from the previous coarser level will be projected to the current level as an initial solution.

The advantages of using the hierarchical approach are twofold [39]. First, detail information at a finer resolution may interfere with the estimation; therefore, the result obtained at the coarsest level is more likely to be close to the true solution. By projecting of this result to the next finer level and repeating this till the finest level, the final result is also more likely to be close to the true solution. Second, if we define a search range R for searching

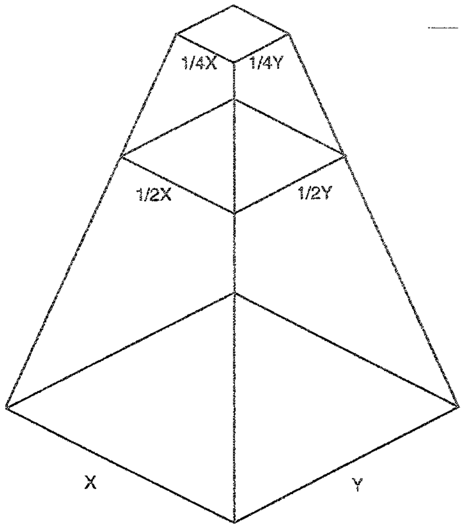


Figure 2.5: Illustration of the structure of an frame pyramid.

the corresponding pixels at the finest level, the search range scales down to $R/2^{L-1}$ at the coarsest level with an L-level pyramid. Furthermore, since the projection of the result from the previous coarser level provides a good start, the searching iterations can be reduced at the current level. Therefore, the total number of operations is smaller than that required by directly searching at the finest level, and the computation is speeded up.

2.5.2 Initial motion estimate

After the down-sampled frame pyramid has been built, an initial motion estimate is executed at the top level of the pyramid. At this initial stage, the result need not be accurate but must assure the convergence of the subsequent gradient descent algorithm. So [9] assumes that there is only translational camera motion, and the 2-parameter translation motion model (Eq. 2.5) is applied at the top level of the pyramid. A 3-step search [12] is adopted to obtain this initial estimate.

Fig. 2.6 shows how this search works. At the first step, the search range is four and

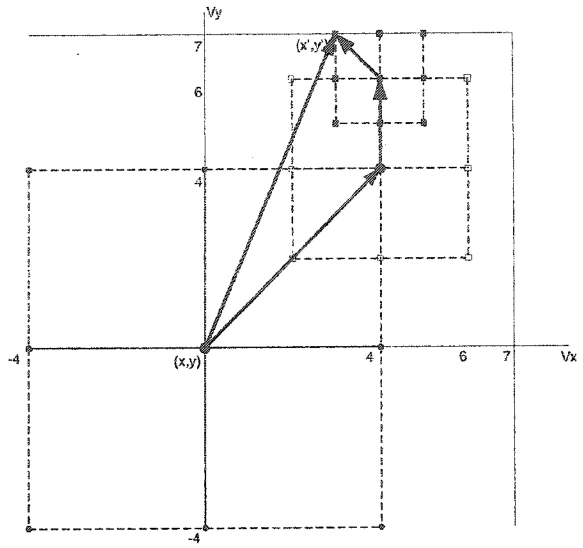


Figure 2.6: An example of 3-step search method [12]. The search ranges are four, two, one at each step, and the matching vectors are [4,4], [0,2], [-1,1] at each step. The final matching vector is [3,7].

the most matching vector is found in this range. Then, the search range is down to two at the second step. Finally, the search range is down to one pixel at the last step and the final matching MV is found. Only 25 trial vectors are used here but can cover a maximum displacement of ±7 pixels at the top pyramid level corresponding to ±28 pixels at the bottom level. In most cases, this range is large enough to cover the camera motion between two consecutive frames. So it is a practical technique.

2.5.3 Gradient descent method

After initial motion parameters are estimated, the next step is to adapt the motion parameters $\mathbf{a} = (a_0 \dots a_k)$ by minimizing SSD. Define

$$E(\mathbf{a}) = SSD = \sum_{i=1}^{N} e_i^2 \tag{2.16}$$

Next, take some particular point P as the origin of the coordinate system with coordinates $\mathbf{a}$, then the function $E(\mathbf{a})$ in Eq. 2.16 can be approximated by its Taylor series [44]:

$$\begin{aligned} E(\mathbf{a}) &= E(\mathbf{P}) + \sum_k \frac{\partial E}{\partial a_k} a_k + \frac{1}{2} \sum_{k,l} \frac{\partial^2 E}{\partial a_k \partial a_l} a_k a_l + \cdots \\ &\approx E(\mathbf{P}) - \mathbf{d} \cdot \mathbf{a} + \frac{1}{2} \mathbf{a} \cdot \mathbf{D} \cdot \mathbf{a} \end{aligned} \tag{2.17}$$

where $\mathbf{d}$ is an N vector whose components are the partial derivative of $E(\mathbf{a})$. $\mathbf{D}$ is an $N \times N$ Hessian matrix whose components are the second partial derivative matrix of $E(\mathbf{a})$.

$$\mathbf{d} = -\nabla E(\mathbf{a})|_{\mathbf{P}}, \qquad \mathbf{D} = \frac{\partial^2 E}{\partial a_k \partial a_l}|_{\mathbf{P}} \tag{2.18}$$

From Eq. 2.17, the gradient of E can be easily calculated as:

$$\nabla E(\mathbf{a}) = \frac{E(\mathbf{a}) - E(\mathbf{P})}{\mathbf{a}} = \mathbf{D} \cdot \mathbf{a} - \mathbf{d} \tag{2.19}$$

Using Newton's method to search for the zero of the gradient of the function near the current point, we set $\nabla E = 0$ in Eq. 2.19 to determine the updated point [44].

Because the 6-parameter affine motion model in Eq. 2.2 depends nonlinearly on $\mathbf{a}$, the minimization must proceed iteratively until meeting the stopping criterion. This procedure is achieved by using the following gradient descent method iteratively.

$$\mathbf{a}^{t+1} = \mathbf{a}^t + \mathbf{D}^{-1} \cdot \mathbf{d} = \mathbf{a}^t + \delta \mathbf{a} \tag{2.20}$$

where $\mathbf{a}^{t+1}$ and $\mathbf{a}^t$ denote the motion parameters at t and $t+1$ iterations respectively, $\delta \mathbf{a}$ is the update term of $\mathbf{a}$.

The gradient of $E(\mathbf{a})$ in Eq. 2.16 with respect to the parameters $\mathbf{a}$, which will be zero at the $E(\mathbf{a})$ minimum, has components

$$\nabla E(\mathbf{a}) = \frac{\partial E}{\partial a_k} = 2 \sum_{i=1}^{N} e_i \frac{\partial e_i}{\partial a_k} \quad k = 0, 1, \ldots, 5 \tag{2.21}$$

Taking an additional partial derivative gives

$$\frac{\partial^2 E}{\partial a_k \partial a_l} = 2\sum_{i=1}^{N} \frac{\partial e_i}{\partial a_k}\frac{\partial e_i}{\partial a_l} - e_i \frac{\partial^2 e_i}{\partial a_l \partial a_k} \approx 2\sum_{i=1}^{N} \frac{\partial e_i}{\partial a_k}\frac{\partial e_i}{\partial a_l} \tag{2.22}$$

It is conventional to remove the factors of 2 by defining

$$\beta_k \equiv -\frac{1}{2}\frac{\partial E}{\partial a_k} = -\sum_{i=1}^{N} e_i \frac{\partial e_i}{\partial a_k} \tag{2.23}$$

$$\alpha_{kl} \equiv \frac{1}{2}\frac{\partial^2 E}{\partial a_k \partial a_l} \cong \sum_{i=1}^{N} \frac{\partial e_i}{\partial a_k}\frac{\partial e_i}{\partial a_l} \tag{2.24}$$

Making $[\alpha] = \frac{1}{2}\mathbf{D}$ in Eq. 2.20, the updated term δa can be rewritten as the set of linear equations.

$$\sum_{i=1}^{M} \alpha_{kl}\delta a_l = \beta_k \tag{2.25}$$

Using singular value decomposition (SVD) [44], the increments δa_l can be solved. Then it is added to the current set of parameters to get the next approximation (see equation 2.20).

The stopping criterion is defined as meeting one of the following two conditions, depending on which one comes earlier:

- The iteration reaches its maximum N_{max}.
- The update term is smaller than a preset threshold. There are two thresholds used here: threshold ε_1 for the update of the translation parameters a_0 and a_3, and threshold ε_2 for the update of the remaining parameters.

In [9], $N_{max} = 32$, $\varepsilon_1 = 0.1$, and $\varepsilon_2 = 0.001$. The gradient descent starts at top level of the frame pyramid, and then continues in a top-down approach. At each level, the iteration will stop if a stopping criterion is met. The projection of the motion parameters from the current level onto the next one is performed by multiplying the translation parameters a_0

and a_3 by two, and remaining the others unchanged. The final motion parameters are obtained after the procedure stops at the bottom level.

2.5.4 Robust estimation

One conflict in GME is that there is only one GM model applied to the whole frame, but not all the pixels in that frame experience the same GM. Therefore, those pixels which have local motion will cause big SSD and bias the estimate of GM parameters. **Robust** estimation aims at solving this problem. The term **robust** has various definitions, but in general, refers to a statistical estimator, it means "insensitive to small departures from the idealized assumptions for which the estimator is optimized" [45]. The world *small* here is interpreted as fractionally large departures for a small number of data points, leading to the notion of outlier points. The basic idea of robust estimation in GME is to identify the pixels that are not undergoing the GM as outliers, and the remaining pixels as inliers [39]. Then the outliers will be eliminated from the next iteration, and only the inliers will be used for the rest of the estimation. In [9], a modified robust estimator, so-called *truncated quadratic* function is used. First, all $\{|e_i|, 1 \leq i \leq N\}$ (N is the total number of the pixels in the frame) are sorted in descending order. Then the threshold e_p is defined so as to exclude the top p% ($5 < p < 15$ suggested by [9]) of the sorted $|e_i|$s. A pixel i is classified as an inlier if $|e_i| \leq e_p$. So the Eq. 2.7 is modified as:

$$SSD = \sum_{i=1}^{N} \rho(e_i), \quad \rho(e_i) = \begin{cases} e_i^2 & : \quad |e_i| \leq e_p \\ 0 & : \quad |e_i| > e_p \end{cases} \tag{2.26}$$

The truncated quadratic function is used only within the gradient descent part of the algorithm; it is not applied in the initial estimate. The threshold e_p is initialized as a reasonably big number (no $|e_i|$s excluded) and is updated after the first iteration at each level.

2.6 Proposed GME Method

Although the reference GME method 2 [9] performs well in several video sequences , some disadvantages and weaknesses are as follows:

- 3-step search has two limitations in initial motion estimate: first, it can not predict the MV correctly if the *true* MV is out of the search range. Second, this pixel-based searching and matching algorithm is still time consuming.
- The robust estimator which excludes the $p\%$ of the largest $|e_i|$s (section 2.5.4) sometimes mis-classifies the outlier pixels as inliers or vice versa.
- Temporal information is not sufficiently used between consecutive GME results.
- High computational complexity for real-time video applications.

In this section, a fast and robust GME method is proposed to improve those disadvantages. This method uses hierarchical differential approach based on reference method 2 [9]. The block diagram of the proposed method is presented in Fig. 2.7. It consists of three steps: frame pyramid construction, initial motion estimate, iterative motion parameter optimization using gradient descent method. Improvements are addressed as follows:

1. Using MV predictor [10] combined with 3-step search [12] for fast initial motion estimate (see section 2.6.1).
2. Using residual information from the previous frames (except for the first frame) to classify the outliers for robust estimation (see section 2.6.2).

3. Since there is no residual information available for the first frame, to improve the accuracy of the GME for the first frame, outlier elimination is considered in neighborhood instead of at each individual pixel (see section 2.6.3).

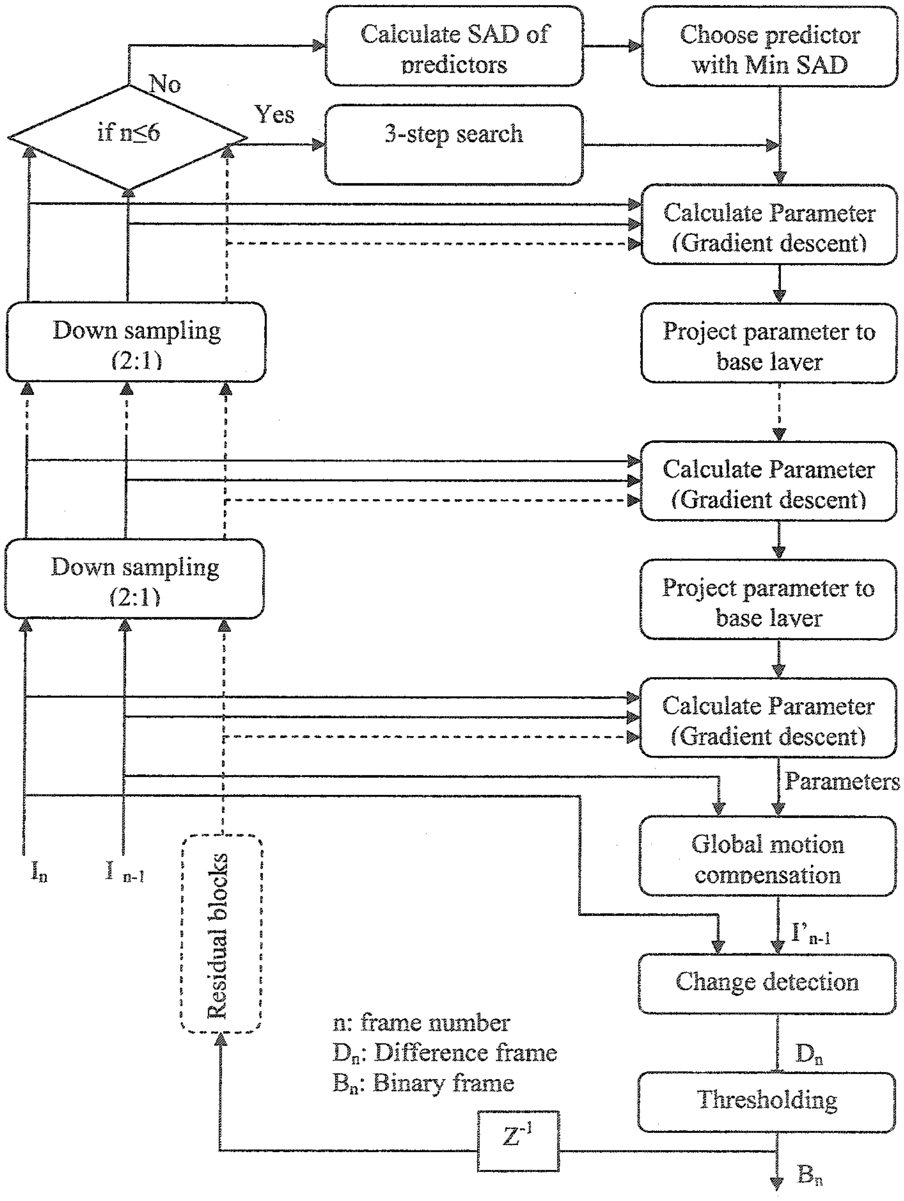

Calculate SAD of predictors
Choose predictor with Min SAD
No
if n≤6
Yes
3-step search
Calculate Parameter (Gradient descent)
Down sampling (2:1)
Project parameter to base layer
Calculate Parameter (Gradient descent)
Down sampling (2:1)
Project parameter to base layer
Calculate Parameter (Gradient descent)
Parameters
Global motion compensation
I_n
I_{n-1}
Residual blocks
I'_{n-1}
Change detection
n: frame number
D_n: Difference frame
B_n: Binary frame
D_n
Thresholding
Z^{-1}
B_n

Figure 2.7: Block diagram of the proposed GME method.

2.6.1 A fast initial motion estimate combining 3-step search and MV prediction

Due to the limitations of 3-step search, we select a more efficient method called MV prediction [10] as a substitution.

Originally, there are six candidates used as motion predictors in [10]: zero MV, past MV, acceleration MV, long-term average MV, historical minimum MV, and historical maximum MV. Besides, these predictors can be combined to generate new predictors. So there are 36 predictors for 6-parameter affine motion model. Our simulation results show that the first four MVs are sufficient and are chosen in the proposed method to reduce the computational complexity. The definitions of each of them are stated below:

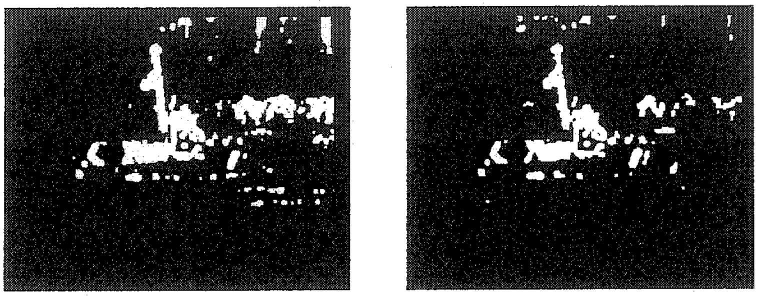
$$\begin{array}{l} \text{zero MV} : \vec{v}_{zero} = 0 \\ \text{past MV} : \vec{v}_{past} = \vec{v}_{n-1} \\ \text{acceleration MV} : \vec{v}_{acceleration} = 2\vec{v}_{n-1} - \vec{v}_{n-2} \\ \text{long-term average MV} : \vec{v}_{average} = \frac{1}{5}\sum_{i=1}^{5}\vec{v}_{n-i} \end{array} \qquad (2.27)$$

To further reduce the computational complexity, the selection of the predictor is executed in two steps [10]. First, assuming there is only translational motion (see Eq. 2.5), the MV with the minimum *SAD* from all candidate MVs is selected and fixed. Then, the rest of the components of candidate MVs are selected again using the same approach. In this way, the complexity is equivalent to computing the *SAD* of 8 predictors instead of 24.

We need six frames to obtain all those MVs in Eq. 2.27. So the 3-step search [12] is still kept as the initial motion estimate for the first *six* frames. From the *seventh* frame on, we use the MV prediction method [10]. Note that this combination is different from [10], where it did not address this problem.

There are several reasons why we chose MV prediction instead of 3-step search. First of all, camera motion is usually continuous over time, thus it is possible to predict MV using history information. Second, six components of the motion parameters can be predicted instead of two, which is more accurate when camera motion is more complex. Third, the MV prediction is applied both in initial estimate and the gradient descent method, which makes it more robust. Finally, it is faster.

Fig. 2.8 shows an example of the binarization compared results between the reference method [9] and the one using MV prediction. From Fig. 2.8, it can be seen that the binarized result is improved.



(a) Reference method 2 [9].

(b) Using MV prediction.

Figure 2.8: Binarization compared results between the reference method 2 [9] and the one using MV prediction. Object pixels are set white, and background pixels are set black.

2.6.2 Using residual (object) information for robust estimation

Since GME is a pre-process for VOS, binary residual frame B_n will be obtained after GME and GMC (see section 3.4). Assuming that the background motion is successfully compensated after GMC, the residual information between the current frame I_n and the motion compensated previous frame I'_{n-1} should contain the objects and the newly appeared background. This information is derived by applying a binarization method [28] which consists of change detection to obtain the difference frame D_n between I_n and I'_{n-1} and thresholding of D_n to obtain B_n (see section 3.4).Then B_n is used to eliminate outliers when estimating motion parameters of the next frame. To prevent misclassification, the pixels in B_n are further grouped into blocks before it is used for GME as follows:

1. The binary frame B_n is segmented into small blocks. The size of the block w_b depends on the frame size. We set 8x8 for CIF/SIF frame, and 12x12 for PAL frame in all simulations. The number of pixels belonging to the objects (marked as white in B_n) in each block is calculated. The top $h\%$ of the blocks with most white pixels are selected as candidate outlier blocks. $20 < h < 40$ and is set 30 in this thesis by experimenting.

2. If a candidate block is a boundary block, it is labeled as an outlier block.

3. If a block is not at boundary and has more than three candidate blocks in its 8-neighborhood, it is labeled as an outlier block.

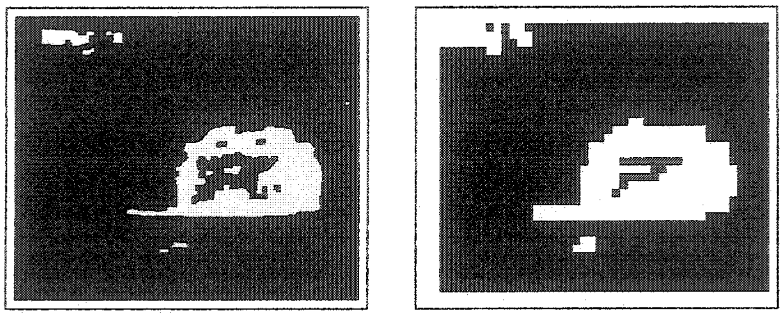
4. If after the previous steps, a candidate block has at least one outlier block in its 8-neighborhood, it is labeled as an outlier block.

The reason we preserve the boundary blocks is that these blocks are most likely newly appeared background and can not find the coordinate pixels in the previous frame, thus it should not participate in the GME. Then Eq. 2.7 is modified to:

$$SSD = \sum_{i=1}^{N} \rho(e_i), \quad \rho(e_i) = \begin{cases} e_i^2 & : \ B_{n-1}(x_i, y_i) = 0 \\ 0 & : \ B_{n-1}(x_i, y_i) = 1 \end{cases} \tag{2.28}$$

where $B_{n-1}(x_i, y_i)$ is the i^{th} pixel in B_{n-1}.

Fig. 2.9 shows the results of creating blocks for the *Ferrari* test sequence (frame 22). In Fig. 2.9, white blocks are the object blocks as well as newly appeared background blocks which will be excluded for GME. It can be seen that outliers are successfully extracted.



(a) Binary frame. (b) Making into blocks.

Figure 2.9: Making into blocks using binary frame.

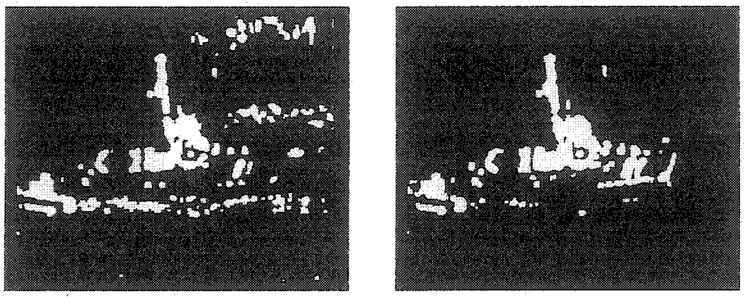
To not propagate estimate errors if the GME of the previous frame fails (e.g., the total number of the outlier blocks changes drastically), the residual information from the last successful GMC is used instead as follows:

$$\begin{array}{l} O_d = |P_n - P_{n-1}|/P_{n-1} \\ If \quad (O_d > t_O) \\ \quad B_n = B_{n-1}; \quad \mathbf{a_n} = \mathbf{a_{n-1}}; \\ \quad P_n = 0.3P_n + 0.7P_{n-1} \end{array} \tag{2.29}$$

with O_d, the outlier difference, P_n (P_{n-1}), the number of the outlier blocks in B_n (B_{n-1}), $\mathbf{a_n}$ ($\mathbf{a_{n-1}}$), the GM parameters of I_n (I_{n-1}), and $0.4 < t_O < 1$.

There are two advantages of using residual information for robust estimate. First, if the GMC of the previous frame does not fail, the residual information correctly contains pixels which are not undergoing GM. Using this information to eliminate outliers is closer to the truth than a statistical estimator. Second, since this residual information is used for VOS, there is no extra computational cost involved. Even if applying this method to video coding or other applications, it is still efficient since the binarization method we applied is fast.

Fig. 2.10 shows an example of the binarization compared results between the reference method [9] and the one using residual information for robust estimation. From Fig. 2.10, it can be seen that the binarized result is improved.



(a) Using robust estimator in reference method 2 [9].

(b) Using residual information for robust estimation.

Figure 2.10: Binarization compared results between the robust estimator in reference method 2 [9] and the one using residual information.

2.6.3 Robust estimation for the first GM compensated frame

The estimation method in Section 2.6.2 can not be applied for the first GM compensated frame I_1' since there is no previous residual information available. Robust estimate in I_1' is, however, of significant importance for algorithm convergence. We improved the scheme in Eq. 2.26 as follows: instead of considering the pixels in the difference frame individually, neighboring pixels are also considered when eliminating the outliers. A pixel i is classified as an inlier only if: a) $|e_i| \leq e_p$, and b) it has m_i neighbors ($m_i > 6$) in its 8-neighborhood $W_8(i)$ with $|e_j| \leq e_p$, $j \in W_8(i)$. Therefore, the Eq. 2.26 is further modified to:

$$SSD = \sum_{i=1}^{N} \rho(e_i), \quad \rho(e_i) = \begin{cases} e_i^2 & : \ |e_i| \leq e_p \wedge m_i > 6 \\ 0 & : \ \text{otherwise} \end{cases} \tag{2.30}$$

Fig. 2.11 shows the compared binarization results between [9] and the proposed method using Eq. 2.30. As can be seen, better binarization results can be achieved.

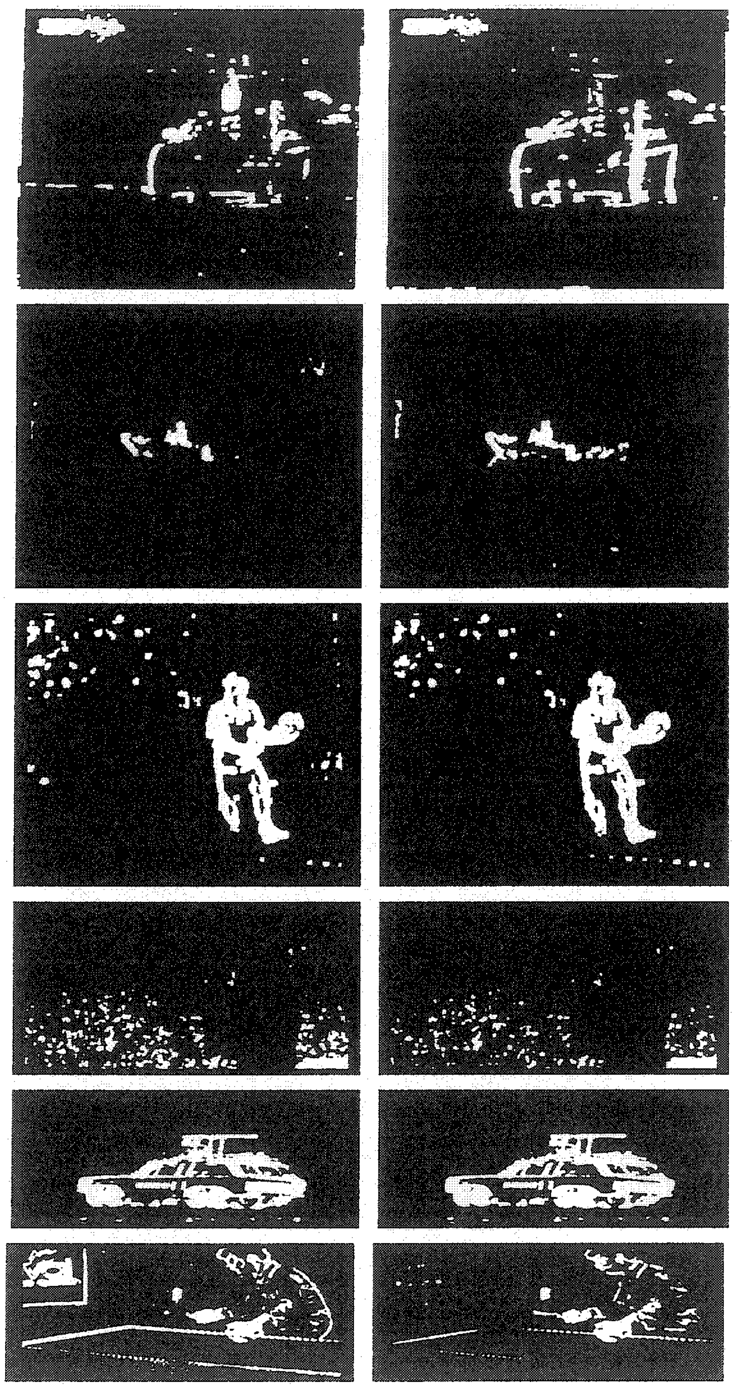
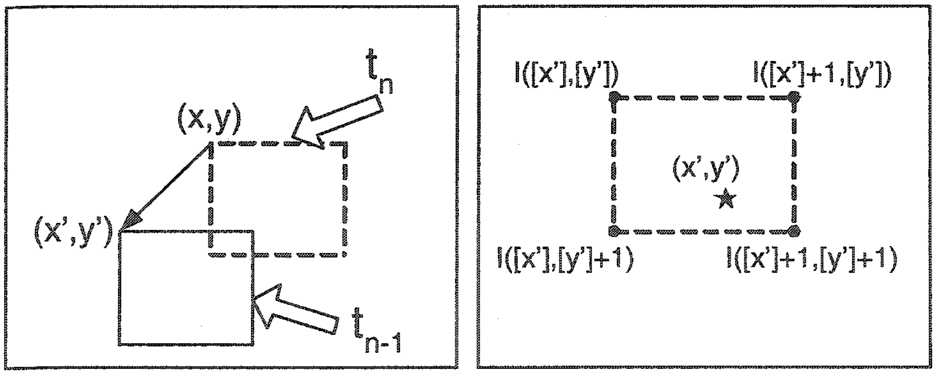


Figure 2.11: Binarization compared results of *Ferrari, Coastguard, Stefan, Marble, Car,* and *Tennis* test sequences for the first frame using robust estimator with (right column) and without (left column) considering the neighborhood pixels.

2.7 Global Motion Compensation

After the GM parameters are obtained, the predicted frame I'_{n-1} can be obtained from the previous frame I_{n-1} using these parameters. This prediction technique is called global motion compensation (GMC). As shown in Fig. 2.12, the predicted value $I'(x, y)$ at the location (x, y) in the I'_{n-1} is copied from the pixel value at (x', y') in the I_{n-1}, where (x', y') is calculated using GM parameters using the assumed motion model. Since the displacements x', y' are real numbers, an interpolation technique is applied to predict the intensity value at (x', y'). Bilinear interpolation is the most widely used interpolation technique [40], and is also adopted in both the reference methods [9, 8] and the proposed method.



(a) Calculate the displacement of (x,y) using GM parameters.

(b) Four nearest neighbors considered in bilinear interpolation.

Figure 2.12: Global motion compensation using bilinear interpolation.

Let $[\cdot]$ denotes the value floor to the nearest integer (see Fig. 2.12),

$$\begin{aligned} I'(x, y) &= \alpha\beta I([x'], [y']) + (1-\alpha)\beta I([x']+1, [y']) \\ &\quad + \alpha(1-\beta) I([x'], [y']+1) \\ &\quad + (1-\alpha)(1-\beta) I([x']+1, [y']+1) \end{aligned} \tag{2.31}$$

where

$$\alpha = [x'] + 1 - x', \quad \beta = [y'] + 1 - y' \tag{2.32}$$

According to the Eq. 2.31, each estimated pixel in the output frame is a weighted combination of its four nearest neighbors in the input frame.

2.8 Summary

GME is an important step in video processing. Efficient and robust GME method is still a challenge. This chapter has proposed a fast and robust GME method. This method is designed for VOS since it aims at successfully compensating the background and extracting the objects. However, it also can be generalized to other applications. The proposed GME method consists of three steps: frame pyramid construction, initial motion estimation, GM parameter optimization using gradient descent method. Contributions of the proposed method are: 1) using a combination of 3-step search and MV prediction for initial motion estimate, 2) using residual information from previous frame for robust estimation, 3) a new robust estimator considering the neighborhood. A fast and noise robust GM detection technique is also proposed in this chapter.

Chapter 3

Object Segmentation Based on Global-Motion Compensation

The objective of this chapter is to integrate the proposed GME method into a VOS scheme and to improve selected modules of this scheme to achieve good segmentation. This scheme is based on change detection in GM compensated frame differences from sequences with GM.

This chapter is organized as follows. Section 3.1 reviews related works. Section 3.3 gives an overview of the VOS scheme used. Section 3.4 explains a noise-robust change detection method [28] and proposes an improved morphological double thresholding method. Section 3.5 introduces a new temporal adaptation technique. Section 3.6 explains the structure of the post processing. Section 3.7 introduces edge adaptation and proposes a new edge warping technique. Section 3.8 summarizes the chapter.

3.1 Related Work

Since VOS is still an open problem, many techniques are being developed. Depending on whether a user controls the segmentation process or not, these techniques can be classified into two categories: supervised and unsupervised approaches. In supervised segmentation [15, 16, 17], also called semi-automatic segmentation, user assistance is needed to define the interested objects in at least some key frames. Supervised segmentation should identify both moving and still objects. Unsupervised segmentation [18]- [28], also called automatic segmentation, can extract objects without user assistance. At present, automatic VOS is mainly limited to identify moving objects. This is sufficient in most cases though, since in many applications we are more interested in moving objects [28].

According to the primary segmentation criteria, segmentation algorithms again can be broadly classified into two approaches: one is based on spatial-temporal homogeneity [16]-[21] and the other is based on change detection [22]- [28]. The major steps of spatial-temporal-homogeneity-based approach can be summarized as follows. First, the original frame is partitioned into homogeneous regions based on some spatial features. Several techniques are available for this purpose, for example, recursive shortest spanning tree (RSST) [27], K-medians [20], and watershed [19, 21]. Watershed algorithm attracts more attention recently because of its simple implementation and precise results compared to the other algorithms. After that, either user assistance [16, 17] or motion estimation [18, 19] is processed to obtain the object regions. In motion estimation, the MV of each region is estimated and regions with similar MVs are merged together. Motion segmentation is not reliable due to aperture and occlusion problems. Some erroneous MVs could have a negative effect on the segmentation results [21]. Therefore, a model-matching technique [17, 21] is further introduced after the object regions are obtained at the initial stage (e.g., the first frame). Those object regions are labeled as an initial object model. Then in the following frames, the objects are tracked based on the best match with the object model from the previous frame, and the model is also updated.

In change detection based approach [22]- [28], first, the moving objects are detected from frame difference of either two successive frames [22, 23, 24] or the current frame and the background frame [25, 28]. Then, a boundary-fine-tuning process based on spatial or temporal information is applied to obtain accurate results. Luminance edge information is the most popular feature used to correct object boundaries [22, 23, 24]. Several drawbacks exist in traditional change detection based approach. First, noise in the background region (e.g., shadow, artifact) can cause detection errors. In [25, 27], shadow detection and elimination techniques are introduced to reduce the shadow effect. In [28], a memory-based change detection along with an artifact-adaptive thresholding method is proposed to adapt to noisy sequences. Second, the result of the frame difference is not consistent if the speed of the object changes significantly in the sequences, causing unstable segmentation results. In [25, 28], the frame differences are calculated between the current and background frames, resulting in more stable objects. But this technique needs the background frame available throughout the whole sequence.

Spatial-temporal homogeneity based approach has promising results at object boundaries. But it sometimes needs user assistance, which might not be suitable for certain kinds of applications. Model-matching assumes that the shape of the objects does not change dramatically from frame to frame. It has difficulty in dealing with a large nonrigid object movement [24]. Furthermore, any objects that appear after the first frame can not be detected since there is no matching model for them [21]. Change detection based segmentation enables automatic detection of any moving objects (including the new appearance and the one with a large nonrigid movement) without user assistance. Because motion information plays the most important role to distinguish moving objects from the background, this approach should be more efficient than the spatial-temporal homogeneity based approach. Algorithms in spatial-temporal homogeneity based approach spend large computational power in processing the background when partitioning the whole frame into spatial homogeneous regions in the first step.

3.2 An Analysis Model-based Segmentation Scheme (Reference Scheme)

This section introduces a generic VOS system [27] which is based on [26]. This system, called COST 211 Analysis Model (AM), is presented by the European-Algorithmic Group 211 which is a forum and research network on video analysis (see Fig. 3.1).

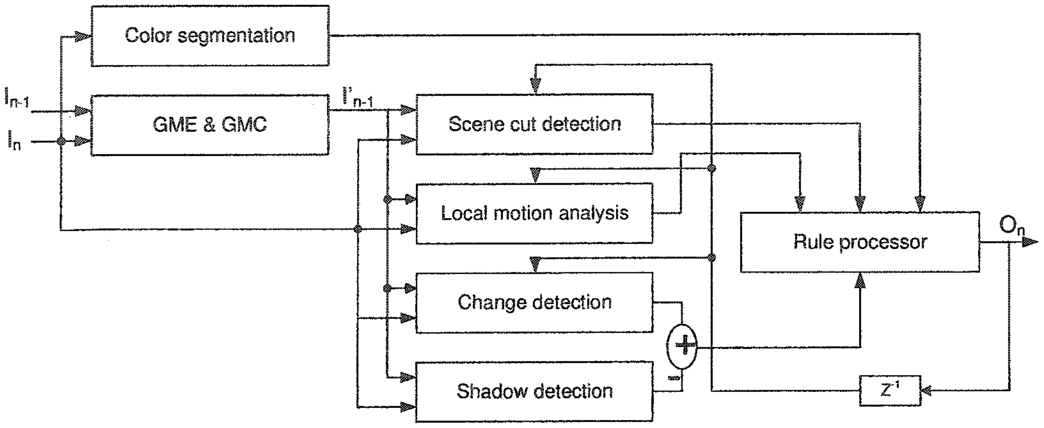


Figure 3.1: Block diagram of the reference VOS method.

- **GME & GMC** This module estimates GM between two consecutive frames using an affine motion model. GMC is achieved by bilinear interpolation.
- **Scene cut detection** This module detects a scene cut if the background mean square error (MSE) between two consecutive frames is greater than a threshold. Then all parameters are reset to their initial values.
- **Change detection** The algorithm for this change detection module is subdivided into three steps [26]: computation of the initial change detection mask (CDM), relaxation of the initial CDM for spatial homogeneity, and temporal coherency of the object shapes.

- **Shadow detection** This module generates a binary mask of moving cast shadows using the method presented in [35]. Then, pixels which are detected as moving cast shadows are deleted from the change detection mask.

- **Local motion analysis** This module estimates a dense displacement vector field by the hierarchical block matching algorithm.

- **Color segmentation** In this module, the current frame is first simplified using a non-linear diffusion filter[36]. This filter generates a simplified image with sharp boundaries. Then color segmentation is used on the simplified image to get a desired number of regions.

- **Rule processor** This module processes some rules to merge the results of the above modules. Each region in color segmentation is proceeded using these rules to decide if it belongs to foreground or background in the final object mask.

3.3 Overview of the VOS Scheme used

The VOS scheme used consists of the following steps (see Fig. 3.2 and Fig. 3.3):

- **GME & GMC** The GME & GMC method proposed in chapter 2 is applied to the current frame I_n and previous frame I_{n-1}.

- **change detection (binarization)** A noise-robust change detection method [28] including thresholding is adopted. This method reduces object noise in the binarized frame B_{nc}. To further remove background clutters, an improved morphological double thresholding is applied.

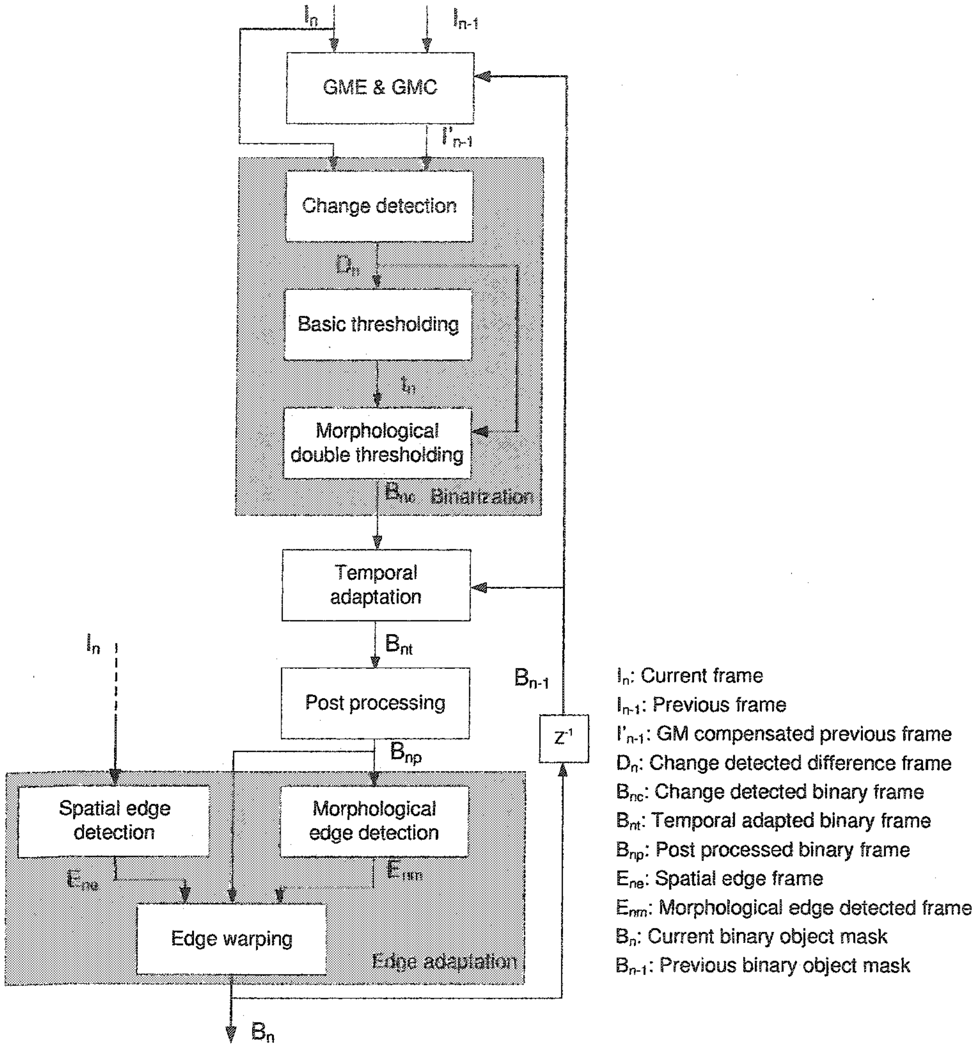

I_n
I_{n-1}
GME & GMC
I'_{n-1}
Change detection
D_n
Basic thresholding
t_n
Morphological double thresholding
B_{nc}
Binarization
Temporal adaptation
B_{nt}
Post processing
B_{np}
B_{n-1}
z^{-1}
I_n
Spatial edge detection
Morphological edge detection
E_{ne}
E_{nm}
Edge warping
Edge adaptation
B_n
I_n: Current frame
I_{n-1}: Previous frame
I'_{n-1}: GM compensated previous frame
D_n: Change detected difference frame
B_{nc}: Change detected binary frame
B_{nt}: Temporal adapted binary frame
B_{np}: Post processed binary frame
E_{ne}: Spatial edge frame
E_{nm}: Morphological edge detected frame
B_n: Current binary object mask
B_{n-1}: Previous binary object mask

Figure 3.2: Block diagram of the VOS scheme used (cf. Fig: 3.3).

- **temporal adaptation** A new temporal adaptation technique is used to stabilize the segmented results.

- **post processing** This module includes small region removal, morphological closing, hole filling and gap filling to construct the objects.

- **edge adaptation** An edge adaptation technique which consists of edge detection and edge warping is applied to improve the accuracy at object boundaries.

Fig. 3.3 shows example results of each step in the VOS scheme used.

Contributions of the VOS scheme used are: 1) an improved morphological double thresholding technique to remove the background residual clutters and to enhance the presence of the objects, 2) a new temporal adaptation technique to stabilize the segmentation results, and 3) a new edge warping technique to improve the accuracy at object boundaries.

3.4 Change Detection

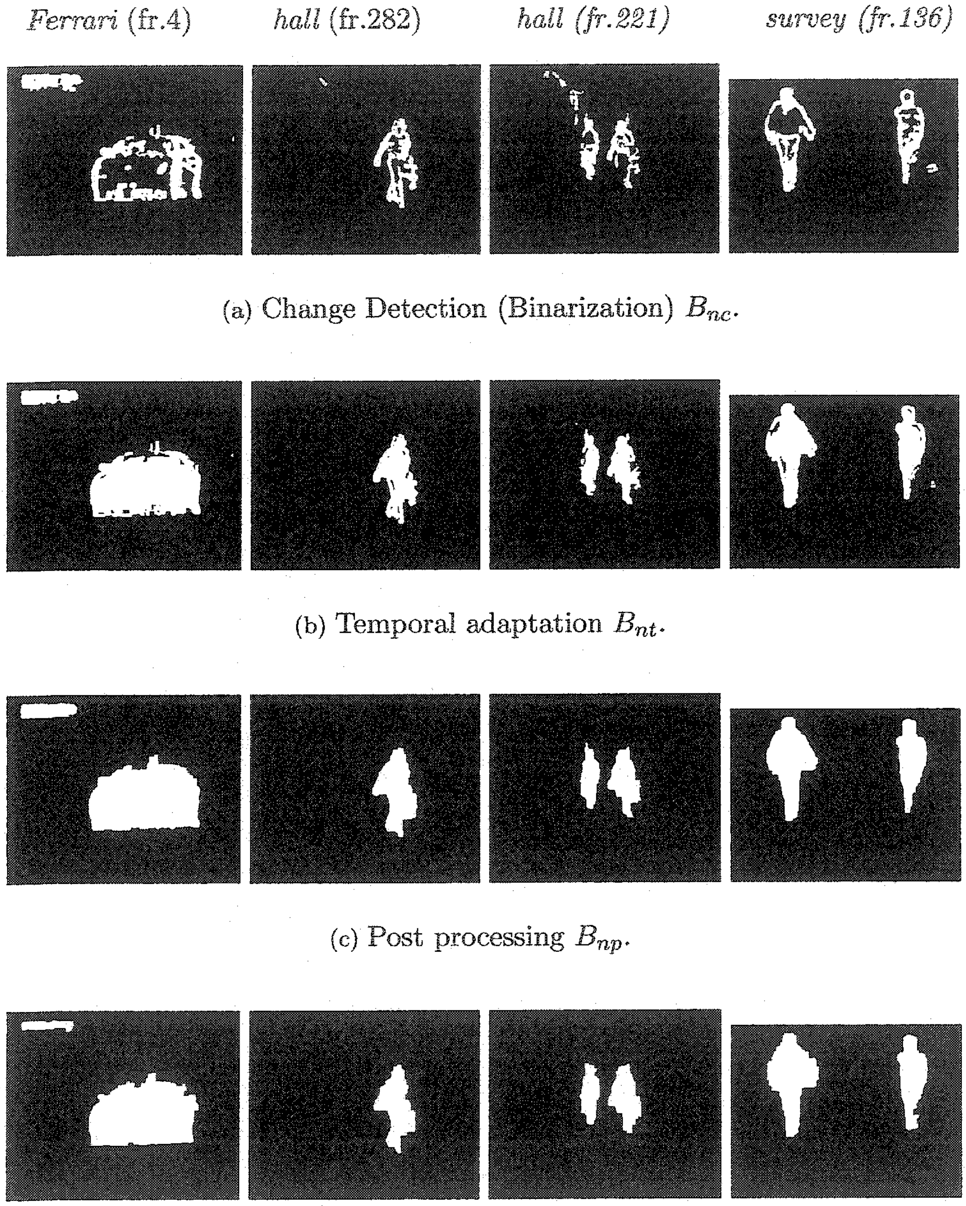
Generally, change detection is followed by thresholding in VOS. Change detection aims at finding which pixels of a frame have changed and group them into objects. Thresholding further separates objects from a background, or discriminating objects from other objects that have distinct gray levels [28]. In this thesis, we construct a basic scheme of change detection and thresholding [28] and morphological double thresholding for heavily cluttered background.

3.4.1 Basic scheme of change detection [28]

Fig. 3.4 shows the block diagram of the basic change detection method [28]. In the block diagram, I_n indicates the current frame, I'_{n-1} indicates the previous frame (or the motion compensated previous frame in case of GM). D_n and t_n indicate the change detected difference frame and the derived threshold respectively. Let

$$D_n(x_i, y_i) = CD(I_n(x_i, y_i) - I'_{n-1}(x_i, y_i)) \tag{3.1}$$

where (x_i, y_i) denotes the position of the i^{th} pixel, $I_n(x_i, y_i)$, $I'_{n-1}(x_i, y_i)$, and $D_n(x_i, y_i)$ are the intensity values of the i^{th} pixel in I_n, I'_{n-1}, and D_n respectively. CD is the change detection operator includes a spatial average filter and a spatial MAX filter (see Fig. 3.4).



(a) Change Detection (Binarization) B_{nc}.

(b) Temporal adaptation B_{nt}.

(c) Post processing B_{np}.

(d) Edge adaptation B_n.

Figure 3.3: Example results of each step in the VOS scheme used (cf. Fig. 3.2).

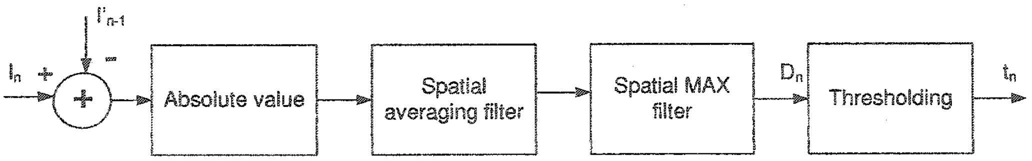


Figure 3.4: Block diagram of the binarization method [28].

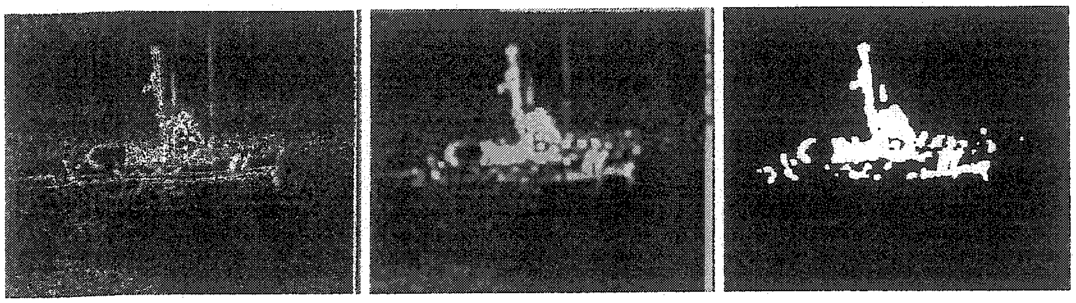
Assume either no GM or GM is compensated, and the global illumination remains more or less constant between the frames. Then the pixel locations where D_n differs from zero indicate regions "changed" as a result of local object motion. The binary frame B_{nc} is then defined as:

$$B_{nc}(x_i, y_i) = \begin{cases} 1: & D_n(x_i, y_i) > t_h \\ 0: & otherwise \end{cases} \tag{3.2}$$

where $B_{nc}(x_i, y_i)$ denotes the value of i^{th} pixel in B_{nc}, and is called a segmentation label field, which is equal to "1" for changed regions and "0" otherwise. t_h is a threshold estimated based on [28]. This non-parametric thresholding method uses both global (block-based) and local (block-histogram-based) decision criteria. It is also adapted to the estimated noise.

Since, the change detection method in [28] was tested mainly in video sequences without camera motion, minor changes are made as follows to cooperate with the situation where camera motion is compensated: Because of possible errors in GME and GMC, the difference frame D_n includes more artifacts than the one without camera motion. Therefore, stronger spatial filter is necessary, instead of using 3 × 3 window for average filter in [28], a 5 × 5 window is applied in this thesis. Note that the window size of MAX filter remains 3 × 3.

Fig. 3.5 shows an example result of this change detection method. We can see that the object is successfully separated from the background.



(a) Simple difference. (b) Change detection. (c) Thresholding.

Figure 3.5: An example result of change detection [28] in case of GM.

3.4.2 Morphological double thresholding

In some applications, the background is heavily cluttered due to the sensor noise, illumination change, and residual of GME and GMC (in case of GM). To further remove these background residual clutters and to enhance the presence of the objects, an improved morphological double thresholding is proposed which is based on [29]. The method [29] consists of three steps:

1. Apply thresholding to the change detected frame D_n using threshold t_{mask} to obtain the mask binary frame B_{mask} (Fig. 3.6 (a)):

$$B_{mask}(x_i, y_i) = \begin{cases} 1: & D_n(x_i, y_i) > t_{mask} \\ 0: & otherwise \end{cases} \tag{3.3}$$

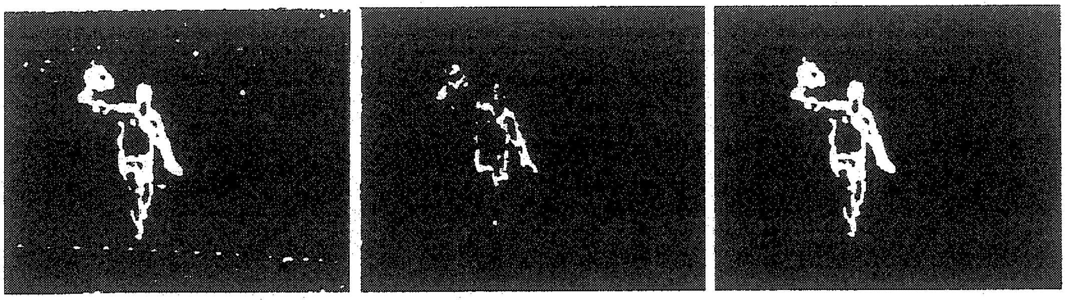
2. Apply thresholding to the same change detected frame D_n using threshold t_{marker} ($t_{marker} > t_{mask}$) to obtain the marker binary frame B_{marker} (Fig. 3.6 (b)):

$$B_{marker}(x_i, y_i) = \begin{cases} 1: & D_n(x_i, y_i) > t_{marker} \\ 0: & otherwise \end{cases} \tag{3.4}$$

3. Apply morphological reconstruction [30] between B_{mask} and B_{marker} to obtain the reconstructed binary frame B_{nc} (Fig. 3.6 (c)). The reconstruction extracts the union

of the connected components of B_{mask} which contains at least one pixel of B_{marker}:

$$B_{nc} = \bigcup_{B_{marker} \cap B_{mask} \neq 0} B_{mask} \tag{3.5}$$



(a) Mask frame B_{mask}.

(b) Marker frame B_{marker}.

(c) Morphological reconstructed frame B_{nc}.

Figure 3.6: Illustration of double thresholding using morphological reconstruction [30].

The proposed improvements to the method [29] are as follows. In [29], the threshold t_{mask} is a fixed value and $t_{marker} = t_{mask} + T_{rc}$, where T_{rc} is a constant. The morphological reconstruction is applied between the mask frame and the marker frame and the resulting number of detections d (white regions) which includes both true targets and false alarms is counted. If $d > 4 \times T0$ (T0 is the number of objects expected), t_{marker} is increased by T_{rc} again and the morphological reconstruction is applied between the mask frame and the updated marker frame. This process is repeated until $d \leq 4 \times T0$ or t_{marker} can not be increased any more.

This process has several disadvantages. First, the expected number of objects $T0$ is unknown before the objects are successfully segmented. Second, using the resulting number of detections d as a stopping criterion is not reliable. Third, using a fixed value as t_{mask} is not generic. Finally, it is not efficient since iterative morphological reconstruction is computed repetitively.

In this thesis, an improved morphological double thresholding method is proposed. Compared to [29], the improvements are addressed as follows:

- $t_{mask} = \alpha \times t_h$. The thresholding function in [28] is applied on D_n to derive the threshold t_h (see Eq. 3.2). $0.4 \leq \alpha \leq 0.8$ and is set 0.65 in this thesis.
- $t_{marker} = \beta \times t_h$. $1.2 \leq \beta \leq 1.6$ and is set 1.4 in this thesis.
- The morphological reconstruction is executed only once between the mask frame and the marker frame using t_{mask} and t_{marker} derived above.

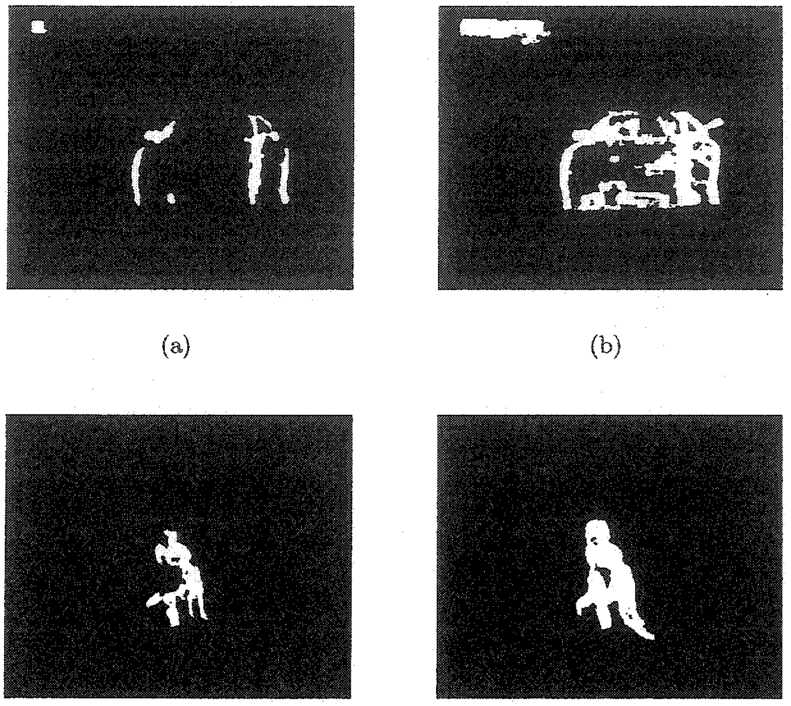
Fig. 3.7(a) and (c) show two example results of [29] where the objects are not complete. Results of the proposed improvements are in Fig. 3.7(b) and (d).

Fig. 3.8 shows some comparison results between the basic thresholding [28] and the proposed morphological double thresholding. Fig. 3.8 (a) has residual background clutters from GME and GMC while Fig. 3.8 (c) contains background clutters due to the illuminant change. As can be seen in Fig. 3.8 (b) and (d), after applying double thresholding, some clutters are removed. However, the objects are preserved. Note that morphological double thresholding needs extra computational cost for morphological reconstruction (about 0.03 sec/frame for CIF format sequences).

3.5 Temporal Adaptation

The results of the binary frame B_{nc} is not stable for some frames due to the following reasons:

- When the frames have complex contents (e.g., fast and complicated camera motion), the limitation of GME and GMC causes B_{nc} to contain background information and



(a) (b)

(c) (d)

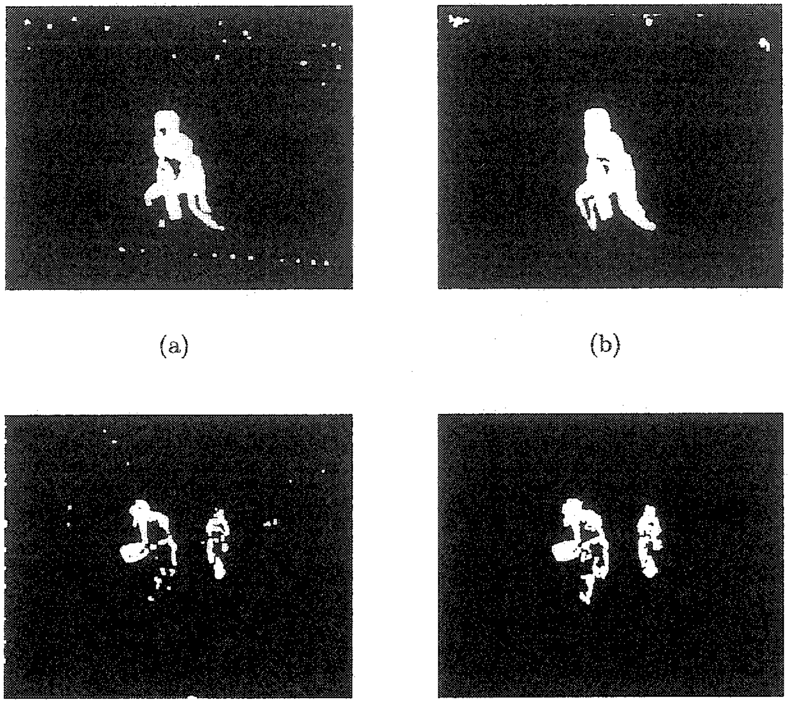
Double thresholding [29]. Improved version.

Figure 3.7: Comparison of results of double thresholding between [29] and the proposed improvements.

give erroneous objects.

- When the objects move slightly during two successive frames, the limitation of change detection causes B_{nc} to contain only partial object information.

To compensate the above weaknesses and get more stable results, temporal information is introduced to adapt B_{nc}. Several temporal adaptation techniques can be found in the literature. For example, in [26] and [27], a memory buffer MEM is built for each pixel. The current binary frame is updated by adding pixels which were also labeled as changed in *one* of the last L frames, where L denotes the depth of the buffer adapted with the motion vectors and the size of the objects (see Eq. 3.6). This temporal adaptation is



(a) (b)

(c) (d)

Basic thresholding [28]. Double thresholding.

Figure 3.8: Comparison of results of basic thresholding [28] and improved double thresholding.

theoretically reasonable. However, after experimenting, we found that the memory depth L is very sensitive. It often enlarges the objects by misadding the previous object pixels to the current object mask. Also, it may misclassify the noisy pixels as objects if *one* of the last L frames contains noisy pixels.

$$MEM(x_i, y_i) = \begin{cases} L & : \; B_{nc}(x_i, y_i) = 1 \\ \max(0, MEM(x_i, y_i) - 1) & : \; B_{nc}(x_i, y_i) = 0 \end{cases} \tag{3.6}$$

We propose a new temporal adaptation technique that is based on the object-tracking principle to stabilize the binary results. This adaptation is based on the observation that the objects can be tracked throughout the consecutive frames while noises and artifacts are randomly appear and can not be tracked. The proposed temporal adaptation consists of three steps.

First, a buffer MEM for each pixel is built and initialized as 0. MEM is updated in each frame as follows:

$$MEM(x_i,y_i)=\begin{cases} MEM(x_i,y_i)+1 & : \quad B_{nc}(x_i,y_i)=1 \\ \max(0,MEM(x_i,y_i)-1) & : \quad B_{nc}(x_i,y_i)=0 \end{cases} \tag{3.7}$$

In Eq. 3.7, the value of each pixel for MEM is incremented by one if the pixel is set white in the binary frame B_{nc}, and is decreased by one when it is set black in B_{nc} until the value in MEM decreases to zero. If a pixel is set white in B_n for consecutive frames, the coordinate value in MEM will increase to a certain large value. Therefore, using this buffer MEM, we can detect the newly appeared object or the recently moving object. Note that this buffer MEM is different from [26, 27] (see Eq. 3.7 and Eq. 3.6).

Second, the previous binary object frame B_{n-1} is virtually morphological dilated and eroded m times respectively, resulting in dilated frame $B_{d_{n-1}}$ and eroded frame $B_{e_{n-1}}$.

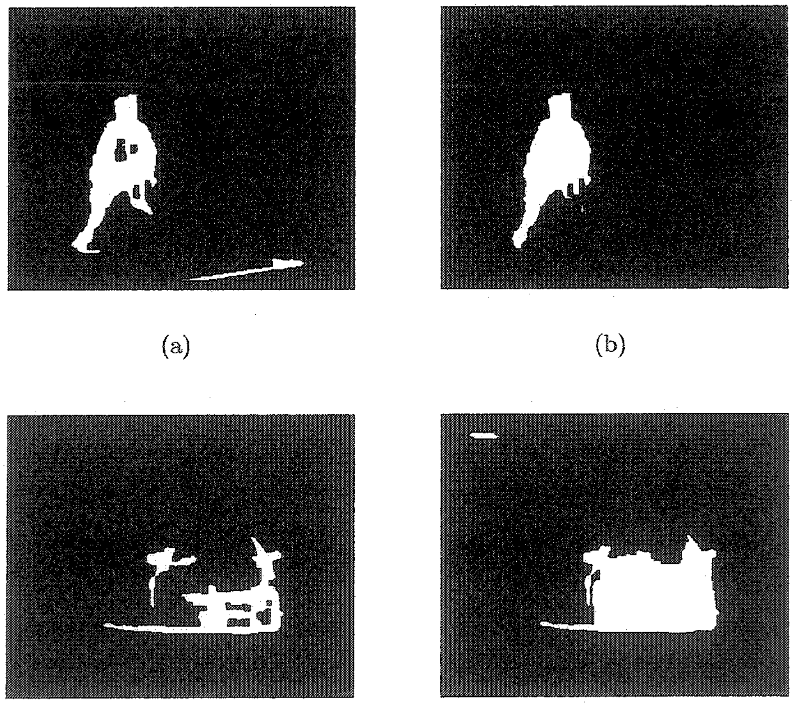
Third, the temporally adapted frame B_{nt} is obtained by Eq. 3.8 which states that: 1) the region within the eroded frame $B_{e_{n-1}}$ is assumed to be object pixels; 2) if the pixel is set white in B_{nc} within $B_{d_{n-1}}$, it will be set white in the adapted frame B_{nt}; 3) if the value in MEM is greater than q frames, the corresponding pixel will also be set as an object pixel. $1 < q < 10$, where $q = 2$ in this thesis which is a compromise between false detection and delay in object detection; and 4) if the pixel meets none of the above conditions, it will set black in B_{nt}.

$$B_{nt}(x_i,y_i)=\begin{cases} 1: & B_{e_{n-1}}(x_i,y_i)=1 \\ 1: & B_{d_{n-1}}(x_i,y_i)=1\cap B_{nc}(x_i,y_i)=1 \\ 1: & MEM(x_i,y_i)>q \\ 0: & otherwise \end{cases} \tag{3.8}$$

where $B_{d_{n-1}}(x_i,y_i)$ and $B_{e_{n-1}}(x_i,y_i)$denotes the value of i^{th} pixel in the dilated and eroded frame of B_{n-1}, respectively.

Theoretically, the number m of dilation and erosion should be adapted to the motion of the object. Since motion estimation is a time consuming task, for the reason of efficiency, m is approximated to *five* in CIF/SIF format and *nine* in PAL format. Simulation results show that unless the objects move very fast, these numbers are suitable.

Fig. 3.9 shows two examples of temporal adaptation. Fig. 3.9 (a) contains background noise while Fig. 3.9 (c) contains part of the object. From Fig. 3.9 (b) and (d) it can be seen that they are well adapted. Fig. 3.10 shows two comparison results between [26] and the proposed temporal adaptation. It can be seen that [26] contains more noisy pixels than the proposed technique.

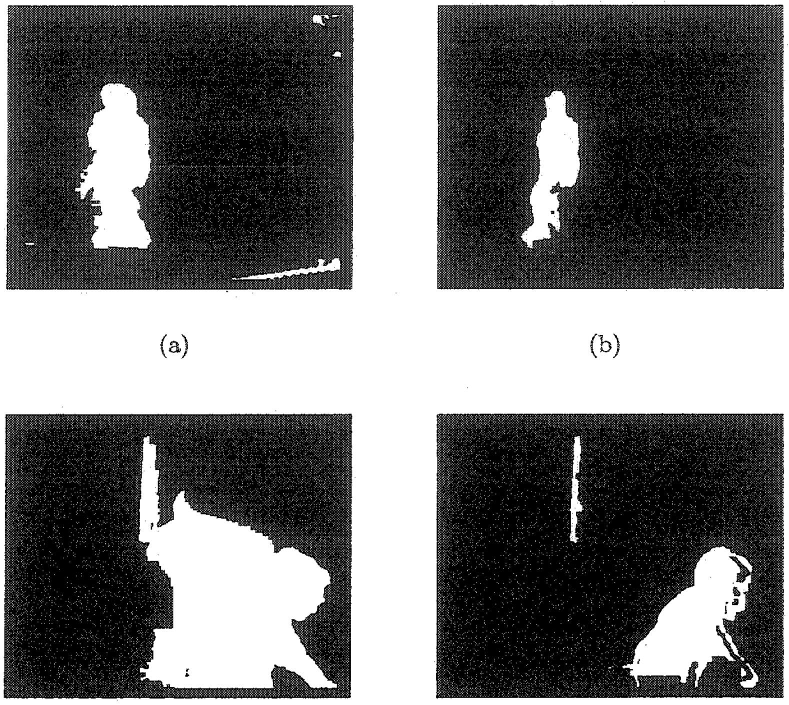


(a) (b)

(c) (d)

Before temporal adaptation. After temporal adaptation.

Figure 3.9: Comparison of results before and after temporal adaptation.



(a) (b)

(c) (d)

Temporal adaptation [26]. Proposed temporal adaptation.

Figure 3.10: Comparison of results between [26] and the proposed temporal adaptation.

3.6 Post Processing

Binary image post processing tools [42] can be used for post processing to remove artifacts or clutter and to complete the object regions in B_{nt}. Artifacts or clutter can be removed by small region removal. Morphological closing is the most popular operator to construct the objects. However, it has limitation when the objects are over segmented. In this case, other processing tools can be used such as gap filling. The disadvantage of gap filling is that it can connect objects when they are close. After experimenting several post processing tools, we choose the following method (see Fig. 3.11):

- **Region counting and small region removal**

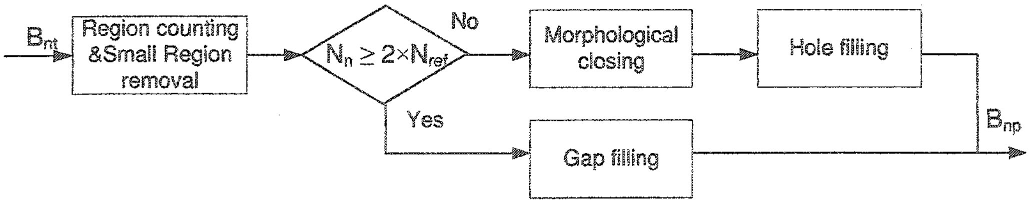


Figure 3.11: Post processing of B_{nt}.

- Count the region number N_n in temporal adapted frame B_{nt} with the size S_r greater than a threshold t_r $(30 < t_r < 150)$.
- Remove from B_{nt} the regions with size $S_r \leq t_r$.
- Store the number of the regions in the first segmented frame as N_{ref} and update N_{ref} every l frames (l is set to the frame rate in this thesis).

- **Gap filling** If $N_n \geq 2 \times N_{ref}$, count the number g of black pixels (the gap) between two consecutive white pixels in both horizontal and vertical directions of B_{nt}. Fill the gap with white pixels if g is less than a threshold t_g. $8 \leq t_g \leq 32$ where t_g should ideally depends on the object and frame size.

- **Morphological closing** If $N_n < 2 \times N_{ref}$, apply morphological closing m times with window size 3×3. m depends on the frame size. It is set up as *five* for CIF/SIF format and *nine* for PAL format.

- **Hole filling** [42] Fill the holes inside the objects. Holes are identified first by inverting the frame (replacing white with black and vice versa), then by labeling the regions that do not touch any border of the frame as the original holes. After the pixels of the holes are added back to the original frame, the holes are successfully filled.

3.7 Edge Adaptation

After the previous steps, the resulting frame B_{np} is coarse at object boundaries, especially at non-rigid parts where object boundaries are complicated. To get more accurate segmentation results, the edge adaptation technique is often used for this purpose. In [26] and [27], edge adaptation consists of the following steps: First (edge detection), Sobel edge detector is used to obtain the luminance edges of the current frame. In second (edge selection), the luminance edges are selected within two limits: the outer limit is the contour of the twice dilated object mask, the inner limit is set to *six* pixel. Third (edge warping), within these limits, every border pixel of the object mask is warped to the nearest luminance edge, if a luminance edge is found within an adaptation radius R around that border pixel. This edge adaptation technique has two disadvantages: First, using the adaptation radius R may have problems finding the correct luminance edges. Besides, the computational cost will increase significantly with the increase of the radius R. Second, using object mask for edge selection is not reliable if the object mask is far away from the correct object boundaries. Furthermore, it is a implicit step since the adaptation is processed within a search range R.

In this section, we propose an edge adaptation technique to adapt B_{np} to spatial luminance edge of the current frame I_n. This edge adaptation technique consists of two steps: edge detection and edge warping (see Fig. 3.2).

3.7.1 Edge detection

Spatial edge detection

There are many spatial edge detection methods developed, including gradient-based methods (i.e., Kirsch, Sobel, Canny), and Laplacian-based methods (i.e., Laplacian of Gaussian, Difference of Gaussian) [42]. Canny edge detector [33] is considered to have the best performance to date, however, it is not clear how to set the thresholds properly. Besides, it is computationally intensive. After doing some experiments and comparisons, Sobel operator [42] is selected in this thesis for edge detection due to its efficiency. We found that the results are not subjectively degraded compared to the one using Canny edge detector (see Fig. 3.12(a) and Fig. 3.12(b)).

In the Sobel operator, first, the horizontal gradient frame I_x and vertical gradient frame I_y of I_n are obtained by convolution of I_n with the masks S_x and S_y as follows:

$$I_x = I_n \otimes S_x, \qquad I_y = I_n \otimes S_y \tag{3.9}$$

where

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ 1 & 0 & 1 \end{bmatrix}, \qquad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.10}$$

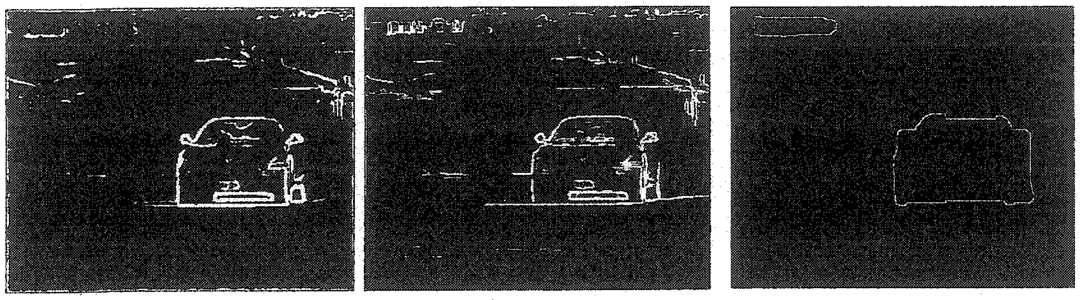
Then spatial edge frame E_{ne} (see Fig. 3.12(b)) is obtained using Eq. 3.11.

$$E_n(x_i, y_i) = \sqrt{I_x^2(x_i, y_i) + I_y^2(x_i, y_i)} = \begin{cases} 1: & E_n(x_i, y_i) > t_e \\ 0: & otherwise \end{cases} \tag{3.11}$$

where $E_{ne}(x_i, y_i)$ is the value of i^{th} pixel at position (x_i, y_i) in E_{ne}, and t_e is a threshold for binarization obtained by applying the thresholding method [28] to the current frame I_n.

Morphological edge detection

Let E_{ne} denote the spatial edge frame. E_{nm} denote the morphological edge frame of the object mask B_{np}. The edge frame E_{nm} of B_{np} is detected using the morphological binary detection in [14]. An example result of this binary detection is shown in Fig. 3.12(c).



(a) Spatial edge frame using Canny operator.

(b) Spatial edge frame using Sobel operator.

(c) Morphological edge frame E_{nm} [14].

Figure 3.12: An example result of edge detection.

3.7.2 Edge warping

In image processing, warping is originally used to relocate the points in aerial photographs, where the image is geometrically distorted due to the plane position [42]. We use this concept to adjust the object boundaries where every border pixel ($E_{nm}(x_i, y_i) = 1$) of the object mask B_{np} is warped to the nearest edge pixel in the spatial object edge frame E_{no}, if there exists such a spatial object edge pixel within a search range. Instead of using the radius R as a search range in [26], we propose a different approach that is more reliable and more efficient. The warping process consists of the following steps:

- Let p_i be an edge pixel in E_{nm}. Find the direction θ_d of p_i out of the *eight* directions as in Fig. 3.13.
- Define a search range, R_d, centered at p_i as follows:
    - If $\theta_d \in \{\theta_1, \theta_2, \theta_3, \theta_4\}$, R_d is perpendicular to θ_d.
    - If $\theta_d \in \{\theta_5, \theta_6, \theta_7, \theta_8\}$, R_d lies in both horizontal and vertical directions.
- Let p_j be an edge pixel in E_{ns}. Find whether there is a $p_j \in R_d$ and $p_j \neq p_i$. If there

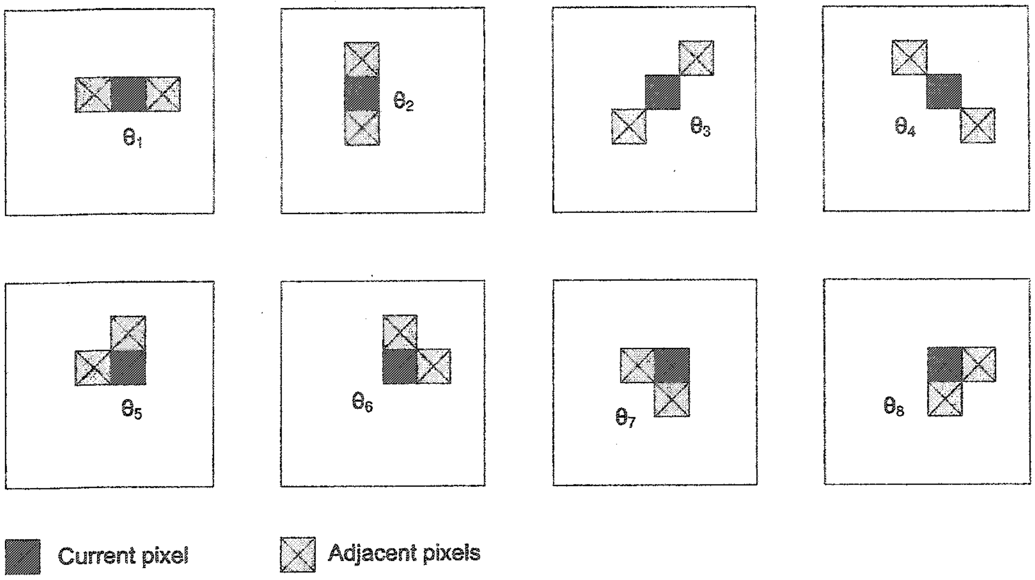


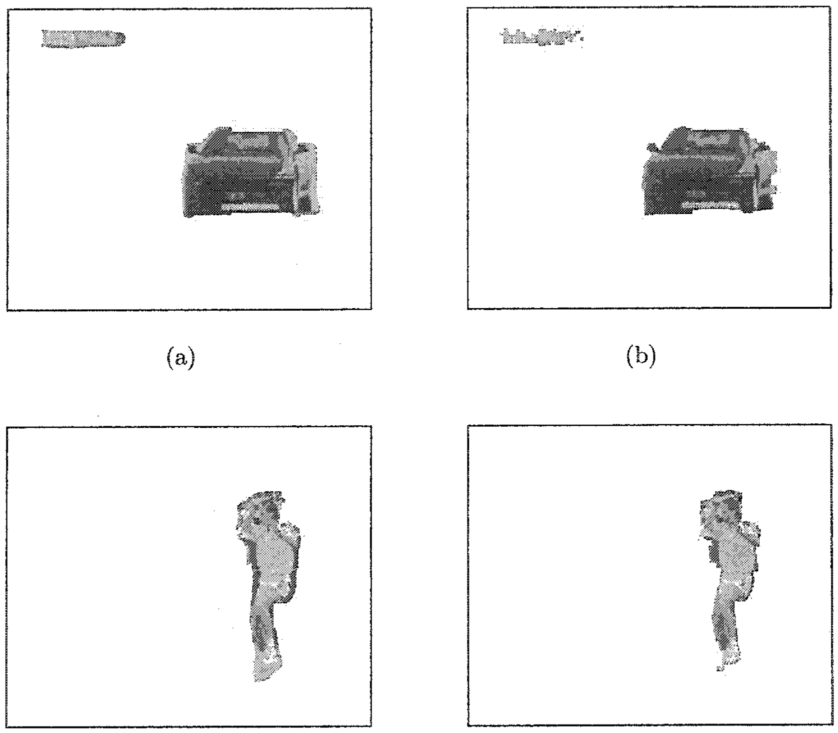
Figure 3.13: Different edge directions for edge warping.

exists more than one p_j within R_d, chose the one with the Euclidean distance closest to p_i along the search range.

- Warp p_i to p_j as follows:
  - If p_j is white in B_{np}, then p_j is assumed to be inside an object and all pixels at the line segment $[p_i, p_j)$ connecting p_i and p_j in B_{np} are set black (where p_i is set black and p_j white).
  - If p_j is black in B_{np}, then p_j is assumed to be outside an object and all pixels at the line segment $[p_i, p_j]$ connecting p_i and p_j in B_{np} are set white (where p_i and p_j are both set white).

The size of the search range R_d depends on the frame size. In this thesis, it is set to 10 pixels for CIF (SIF) format and 20 pixels for PAL format.

The proposed technique has three advantages compared to [26]. First, using the direction of the border pixels to define the search range can enhance the possibility of finding the correct luminance edge. Second, since the search range is changed from a circle to one (or two) line, the number of searches is reduced. Thus it is faster than [26]. Third, it uses no edge selection and thus it avoids missing luminance edges. Fig. 3.14 shows two examples of edge adaptation. When the luminance edge is detected, the boundary of the object is accurately adapted.



(a) (b)

(c) (d)

Before edge adaptation. After edge adaptation.

Figure 3.14: Comparison of results before and after edge adaptation. See the boundary of the left side mirror of the car in (a) (b), and the boundary of the human body in (c) (d).

3.8 Summary

VOS is one of the most challenging topics in digital video processing. A large variety of methods have been developed. This chapter has integrated the proposed GME method into a VOS scheme and improved selected modules of this scheme to achieve good segmentation. The scheme used consists of the following steps: GME & GMC, change detection, temporal adaptation, post processing and edge adaptation. Improvements of this scheme are: 1) an improved morphological double thresholding technique to remove the background residual clutters and to enhance the presence of the objects, 2) a new temporal adaptation technique to stabilize the segmentation results, 3) a new edge warping technique to improve the accuracy at object boundaries.

Chapter 4

Results and Discussions

This chapter presents and discusses the results of the proposed GME method (Chapter 2) and the VOS scheme used (Chapter 3). To evaluate the performance of the method and the scheme, several standard test sequences are tested with or without GM. Simulation results are compared to the reference methods subjectively and objectively.

This chapter is organized as follows. Section 4.1 lists the test video sequences. Section 4.2 presents the results of the proposed GME method. Section 4.3 presents the results of the VOS scheme used. Section 4.4 summarizes the chapter.

4.1 Test Sequences

Simulations are carried out using several standard test sequences (video shots) with or without camera motion as follows:

4.1.1 Test sequences with GM

- *Coastguard* (Fig. 4.1(a)) This 352x288 CIF sequence has 300 frames. It contains two ships driving in opposite directions with mainly translational camera motion. Note that the moving ships also cause the water motion which may interfere with the GME and the VOS.

- *Ferrari* (Fig. 4.1(b)) This CIF sequence has 70 frames. It contains a car moving fast with complex camera motion. Furthermore, there is dust behind the car which complicates the GME.

- *Stefan* (Fig. 4.1(c)) This CIF sequence has 300 frames. It contains a tennis player and inconsistent but fast camera motion, which is difficult to predict. It also has many local motions in the audience region.

- *Marble* (Fig. 4.1(d)) This 512x256 test sequence has 30 frames. It contains a polyhedral scene with a slightly moving marbled block and moving camera.

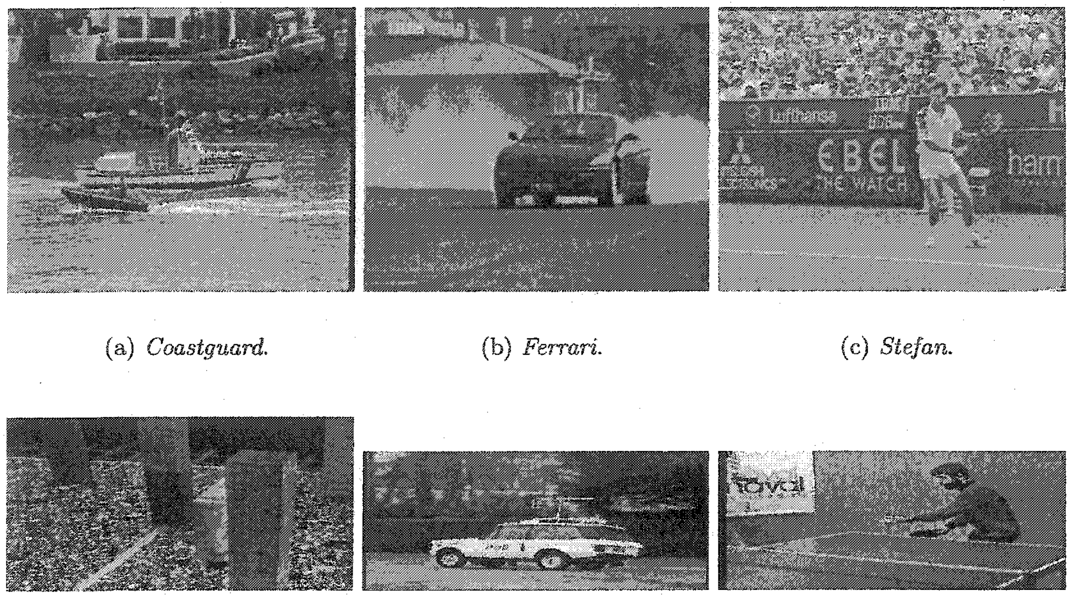
- *BBCcar* (Fig. 4.1(e)) This 720x576 interlaced PAL test sequence has 60 frames. It contains a fast moving jeep with a pan and slight rotate camera motion. There are moving tree leaves at the up-right corner of this sequence.

- *Tennis* (Fig. 4.1(f)) This interlaced PAL test sequence has 38 frames with camera zoom in a player playing table tennis[1].

4.1.2 Test sequences without GM

- *Survey* (Fig. 4.2(a)) This 320x240 SIF sequence has 1000 frames. It is an outdoor sequence with a still background. It contains several pedestrians walking on the street, entering and leaving the scene. Because this sequence is originally recorded with an analog NTSC-based camera and reconstructed to SIF format, interlace artifacts are presented in the constructed frames.

[1] The whole sequence contains three shots. We only select the one with GM.



(a) *Coastguard.* (b) *Ferrari.* (c) *Stefan.*

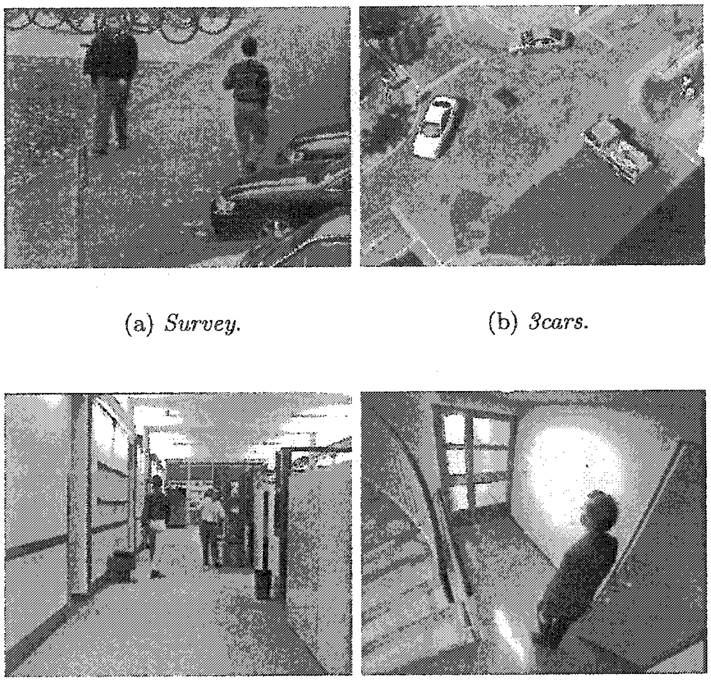
(d) *Marble.* (e) *BBCcar.* (f) *Tennis.*

Figure 4.1: Test sequences with GM.

- *3cars* (Fig. 4.2(b)) This SIF sequence has 66 frames. It is an outdoor sequence with a still background. It contains a traffic intersection with several vehicles and pedestrians moving.

- *Hall* (Fig. 4.2(c)) This CIF sequence has 300 frames. It it an indoor scene with shadows and illuminant changes and a still background. Two persons enter the scene and one leaves afterward.

- *Stair* (Fig. 4.2(d)) This CIF sequence has 1475 frames. It it an indoor noisy scene with shadows, and illuminant changes and a still background. The scene contains two doors and parts of the stairs. The first person enters the back door, goes to the front door and exits. Then he returns and exits from the back door. The second person comes down the stairs, goes to the back door, and to the front door and exits. Then



(a) *Survey.* (b) *3cars.*

(c) *Hall.* (d) *Stair.*

Figure 4.2: Test sequences without GM.

he returns and goes up the stairs.

4.2 Results of the Proposed GME

Simulations of the GME are carried out using the standard test sequences *Coastguard, Ferrari, Stefan, Marble, BBCcar,* and *Tennis.* The proposed method is compared to the reference method 1 in [8] and the reference method 2 in [9].

4.2.1 Subjective GME results

Fig. 4.3 to Fig. 4.14 show the comparison results of the proposed GME method, the reference methods 1 [8] and 2 [9]. In these figures, the absulute difference frames between the current frame I_n and the motion compensated previous frame I'_{n-1} (see Eq. 4.1) and the binarized object frames (see chapter 3) are presented to show the affect of using different GME methods on VOS. For the purpose of visual attention, the difference frames are brightened *four* times. As can be seen, the proposed GME method achieves better subjective results.

$$D_n(x_i, y_i) = |I'_{n-1}(x'_i, y'_i) - I_n(x_i, y_i)| \tag{4.1}$$

To further test the performance of the proposed method, we conducted an initial subjective test where we asked two experts and two non-experts to evaluate the segmentation output. We showed them three frames in sequential order: original frame, proposed segmented frame, reference segmented frame [9]. This was repeated for all the frames in the test sequences. The results of this test are presented in Table 4.1. It shows that the average improvement of the proposed method is 39% compared to the reference method 2 [9].

sequence name	*better*	*similar*	*worse*	*improved*
Coastguard	198(frame)	93(frame)	9(frame)	189(frame)
3000(frame)	66%	31%	3%	63%
Ferrari	53(frame)	16(frame)	2(frame)	51(frame)
70(frame)	76%	23%	3%	73%
Stefan	105(frame)	192(frame)	3(frame)	102(frame)
300(frame)	35%	64%	1%	34%
Racecar	27(frame)	28(frame)	12(frame)	15(frame)
67(frame)	40%	42%	18%	22%
BBCcar	2(frame)	58(frame)	0(frame)	2(frame)
60(frame)	3%	97%	0%	3%

Table 4.1: Statistical subjective comparison results between the proposed method and the reference method. Here, *"better"*, *"similar"*, *"worse"* means the results of the proposed method are better, similar or worse than the reference method 2 [9]; *"improved"* means the overall improvements of the proposed method compared to the reference method 2 [9].

4.2.2 Objective GME results

To evaluate the GME objectively, we use three criterion: the mean absolute error (MAE), the consistency of the segment, and the computational complexity.

MAE between the current frame I_n and GM compensated previous frame I'_{n-1} (Eq. 4.2) is measured and compared to objectively compare the proposed GME method and the reference GME method 1 [8] and 2 [9], .

$$MAE = \frac{1}{N} SAD = \frac{1}{N} \sum_{i=1}^{N} |e_i|, \quad e_i = I'_{n-1}(x'_i, y'_i) - I_n(x_i, y_i) \tag{4.2}$$

where N is the number of the pixels for the whole frame.

Fig. 4.15 and Fig. 4.16 show the objective comparison of the MAE results for each test sequences. As can be seen, the proposed GME method has lower MAE than both the reference methods [9, 8].

Since the size of the objects is temporally consistent in a video shot, the percentage of the segmented object pixels should change gradually throughout the sequence.Fig. 4.17 shows the comparison percentage of object pixels through each test sequence. In Fig. 4.17 and Fig. 4.18, the proposed method shows more stable result than both the reference methods [9, 8] overall, which confirms the observation of the subjective results in section 4.2.1.

Since GME is a time-consuming task, efficiency is another important criterion in evaluation of the different methods. The average computational time per frame for each sequence using different methods are compared in Fig. 4.19. It shows that the proposed method is about 1.5 time faster than the reference method 2 [9] and 2.5 time faster than the reference method 1 [8] using the C program language under a Sparc-Sun-Solaris 2.8 1010MHz system.

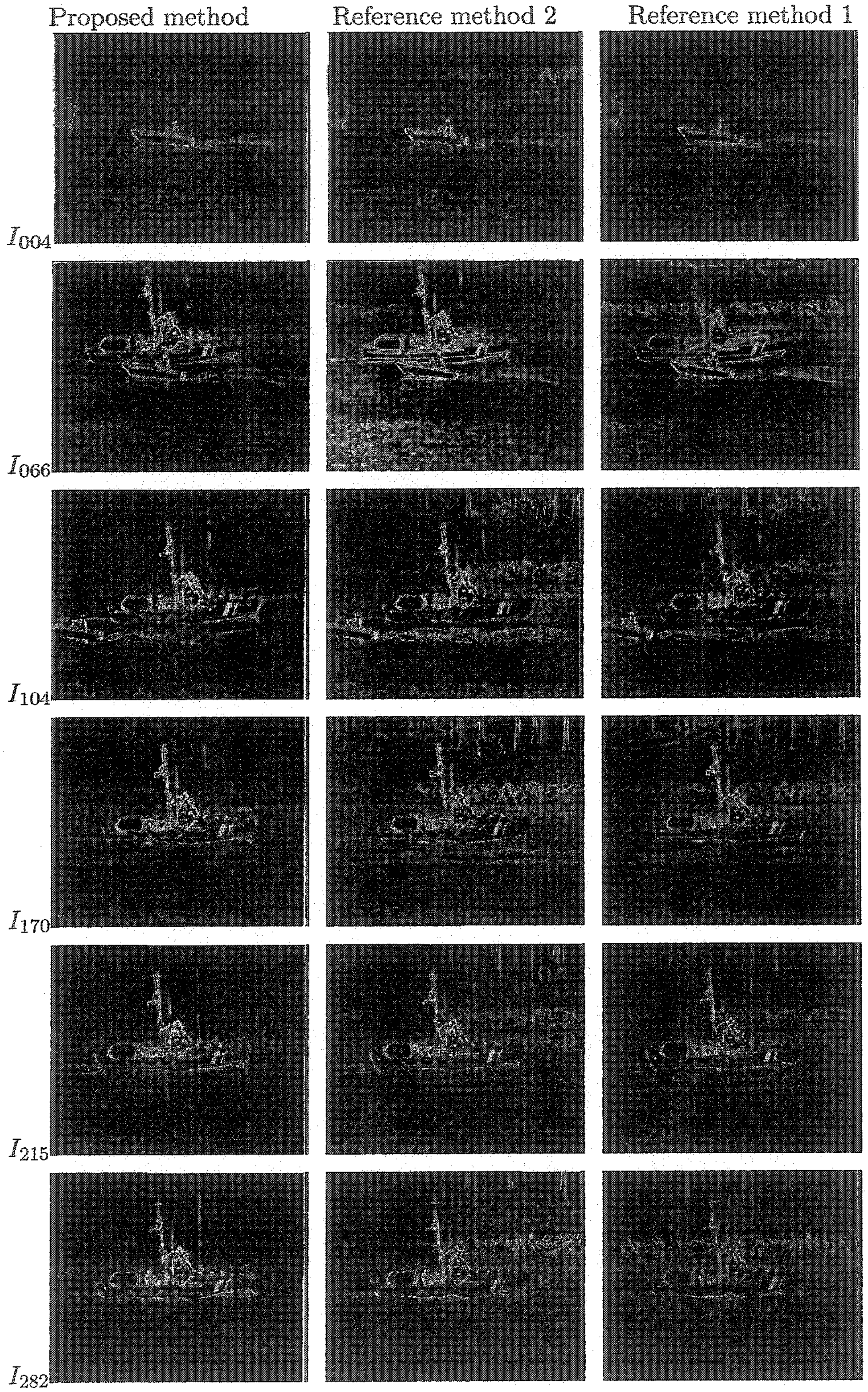


Figure 4.3: GME compared results of the difference frames of *Coastguard* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

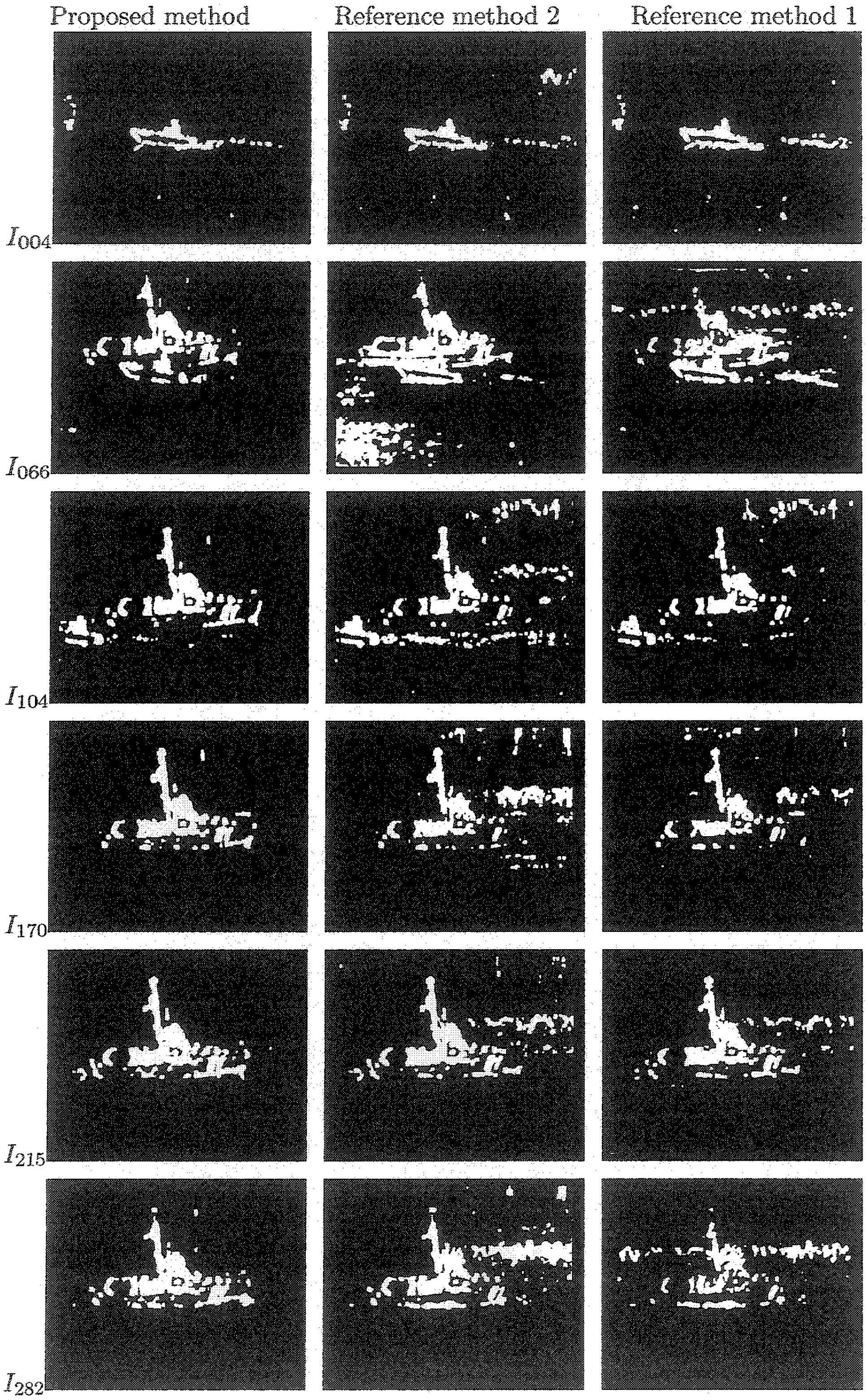


Figure 4.4: GME compared results of the binarized object frames of Fig. 4.3. The waves of the water interfere with GME of the reference methods, while the results of the proposed method remain stable.

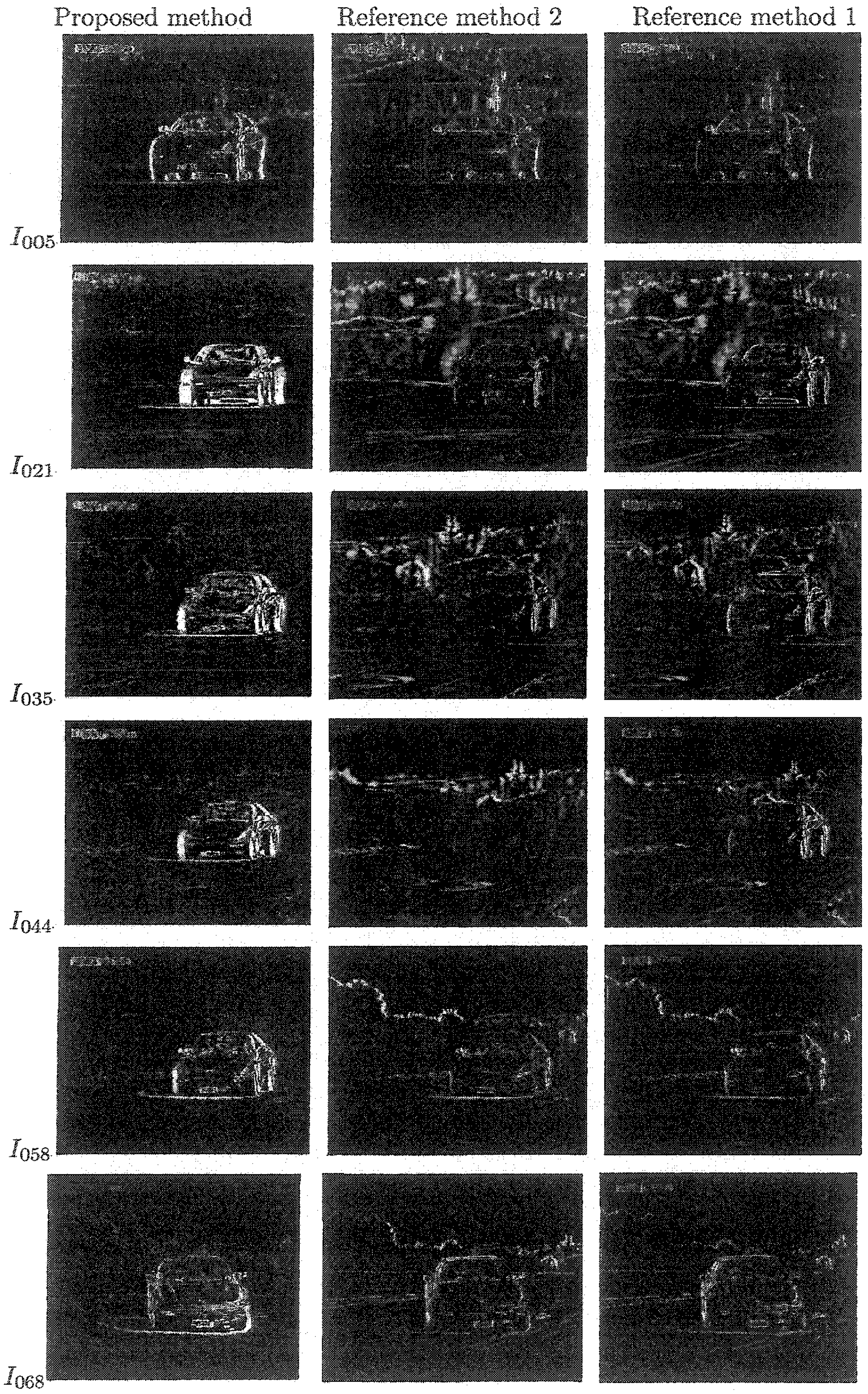


Figure 4.5: GME compared results of the difference frames of *Ferrari* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

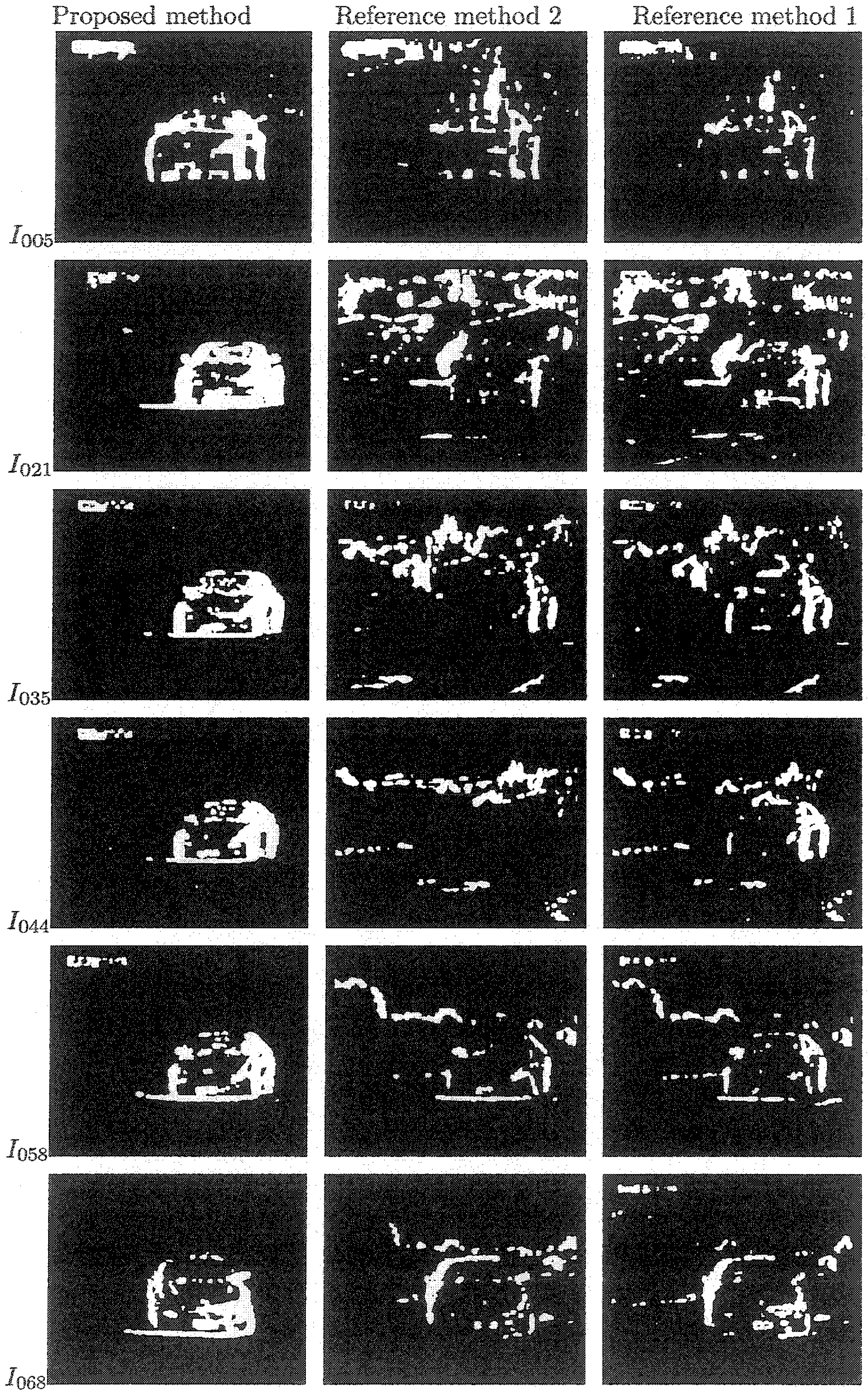


Figure 4.6: GME compared results of the binarized object frames of Fig. 4.5. In this sequence, the object motion is dominant in some frames. This makes the reference method fail to identify the outliers, while the proposed method does not have this problem.

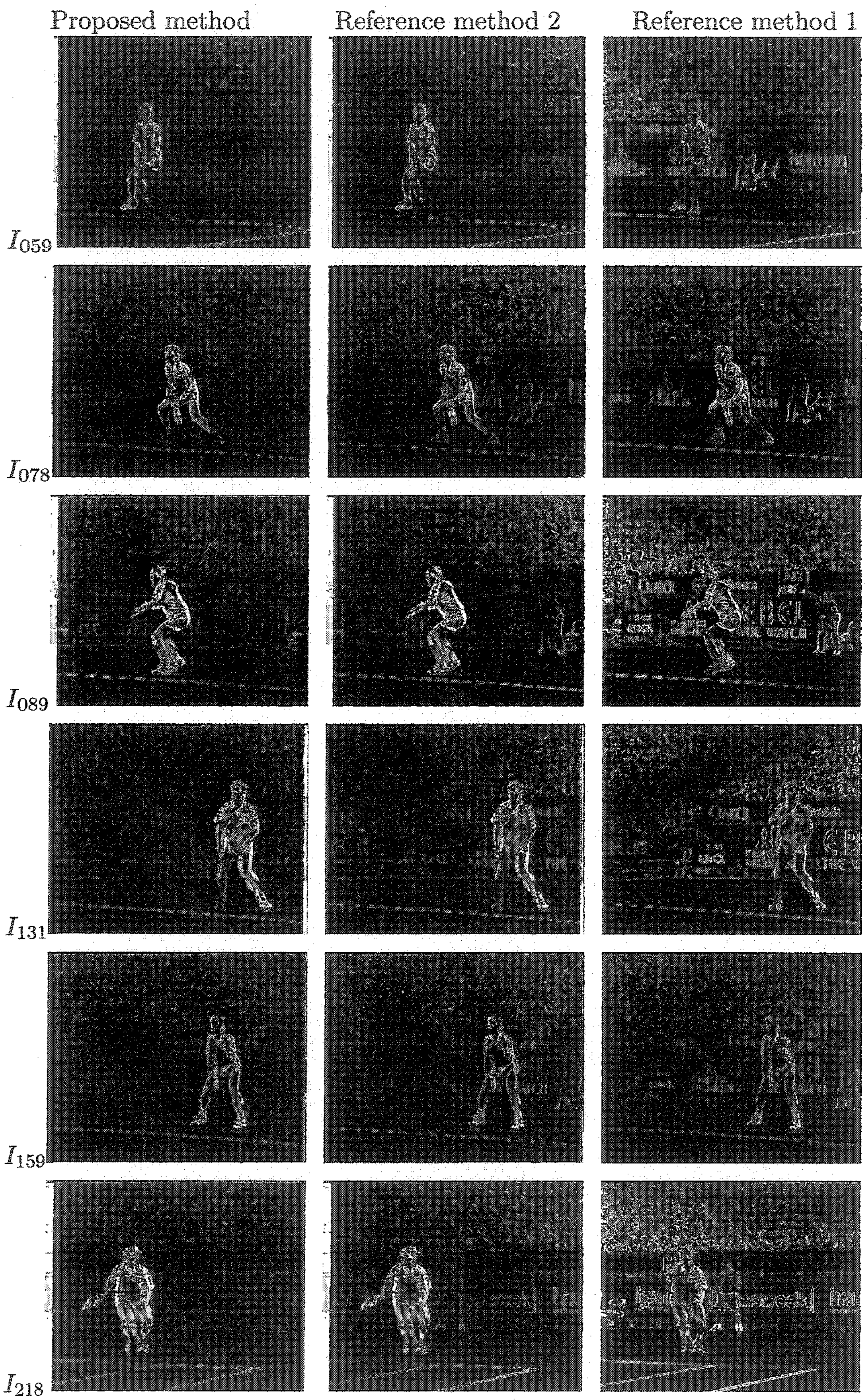


Figure 4.7: GME compared results of the difference frames of *Stefan* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

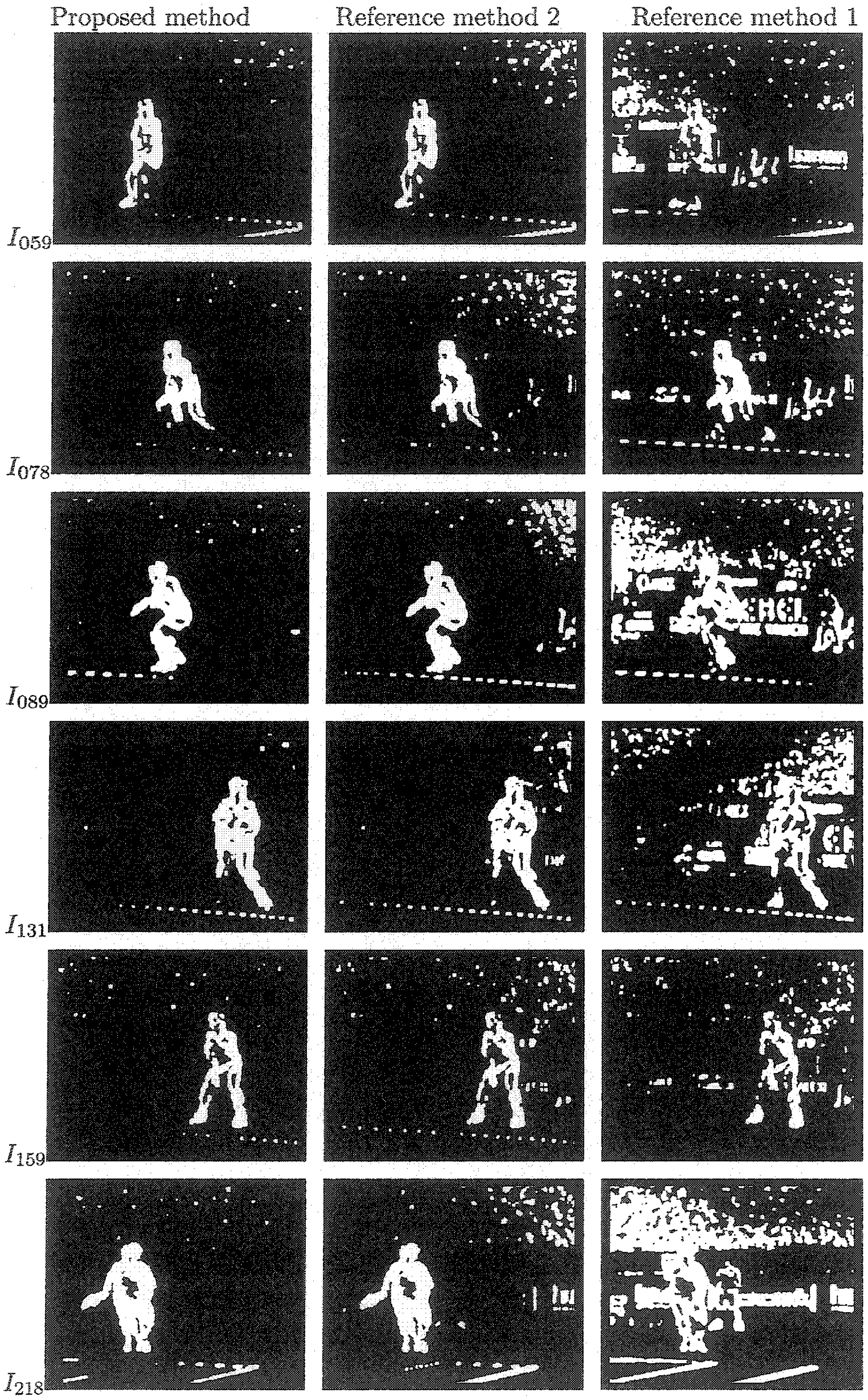


Figure 4.8: GME compared results of the binarized object frames of Fig. 4.7. This difficult sequence contains lots of local motions in the audience area (see the top part of the frames) which interfere with the GME. But the proposed method still performs better than the reference methods.

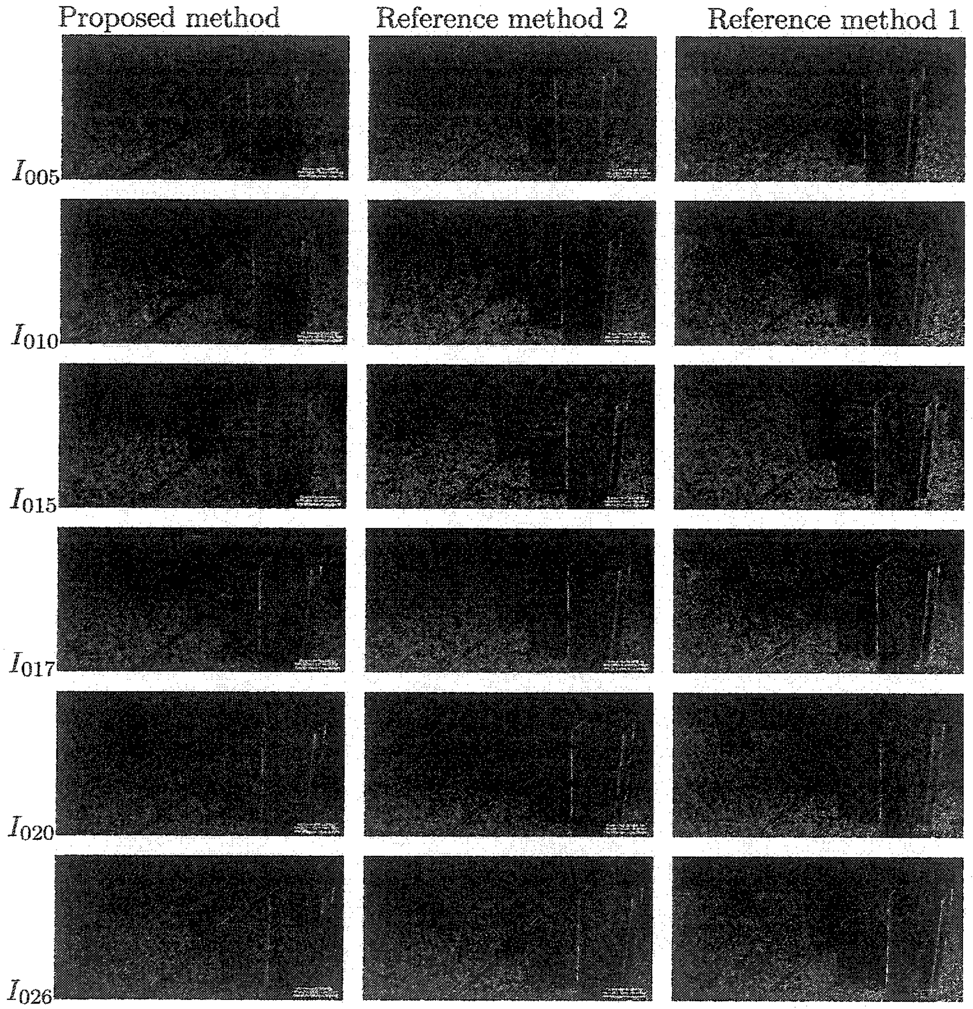


Figure 4.9: GME compared results of the difference frames of *Marble* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

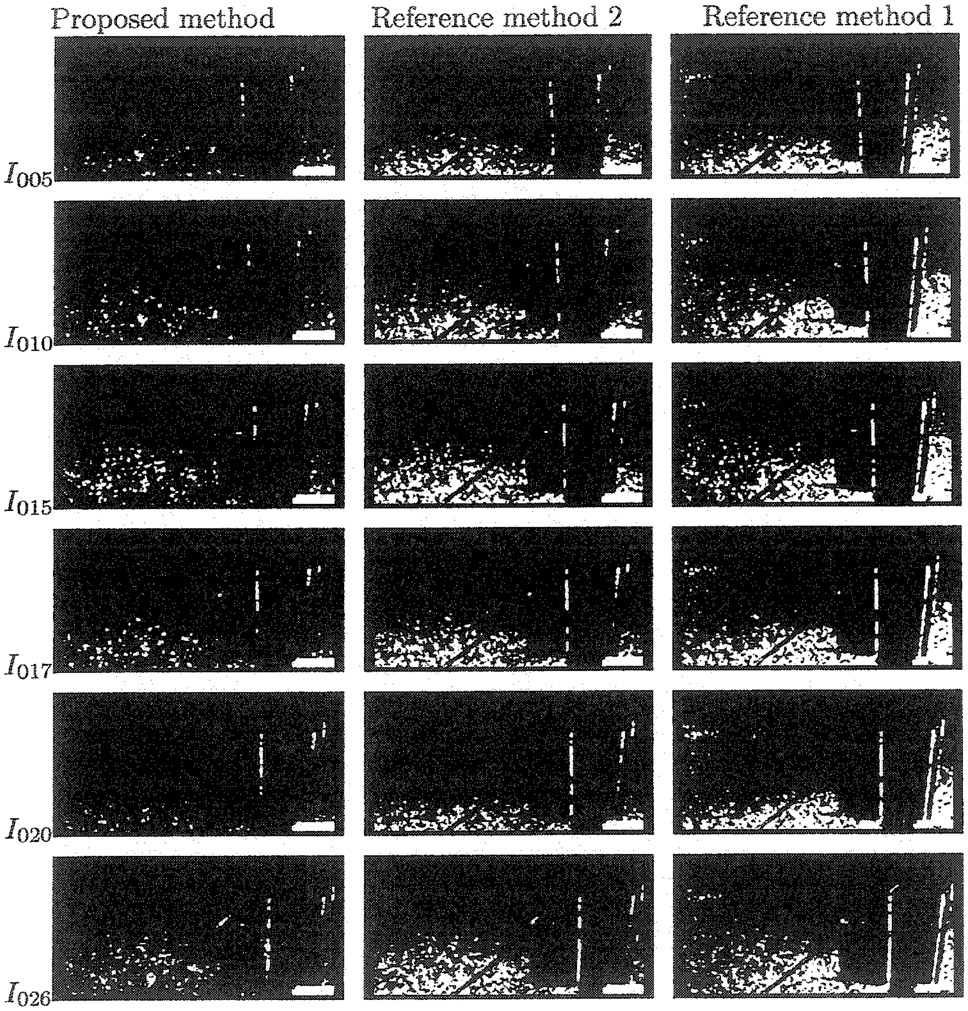


Figure 4.10: GME compared results of the binarized object frames of Fig. 4.9. This sequence contains high texture in the background which requires accurate GME and GMC. The proposed method performs better than the reference methods in compensating the GM.

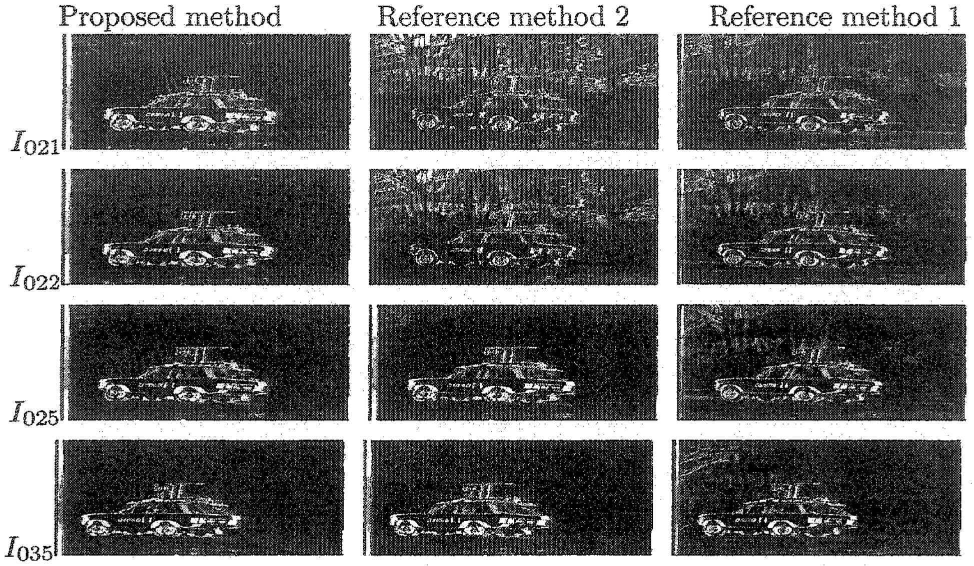


Figure 4.11: GME compared results of the difference frames of *BBCcar* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

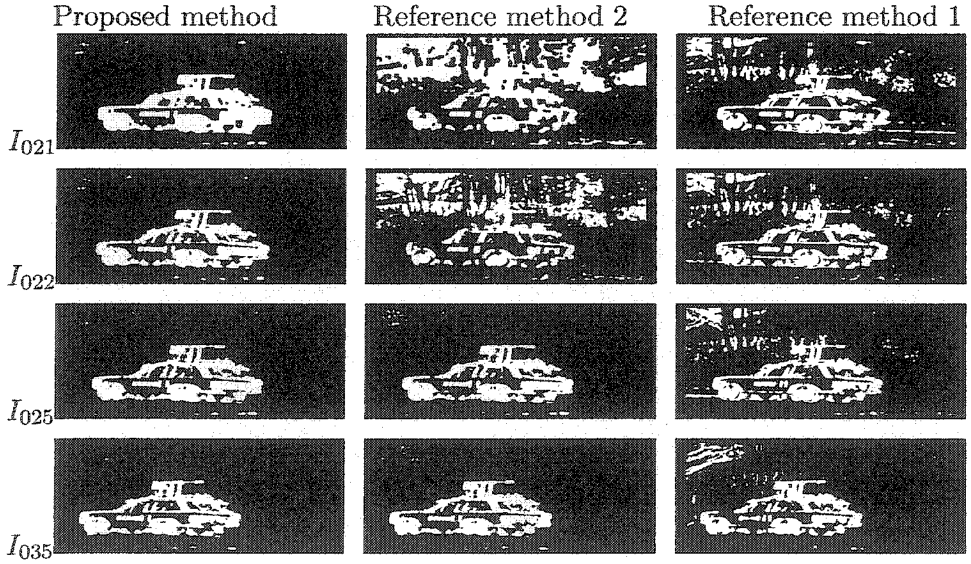


Figure 4.12: GME compared results of the binarized object frames of Fig. 4.11. The proposed method shows better results to compensate the shake of the tree leaves (see the top-left corner of the frame).

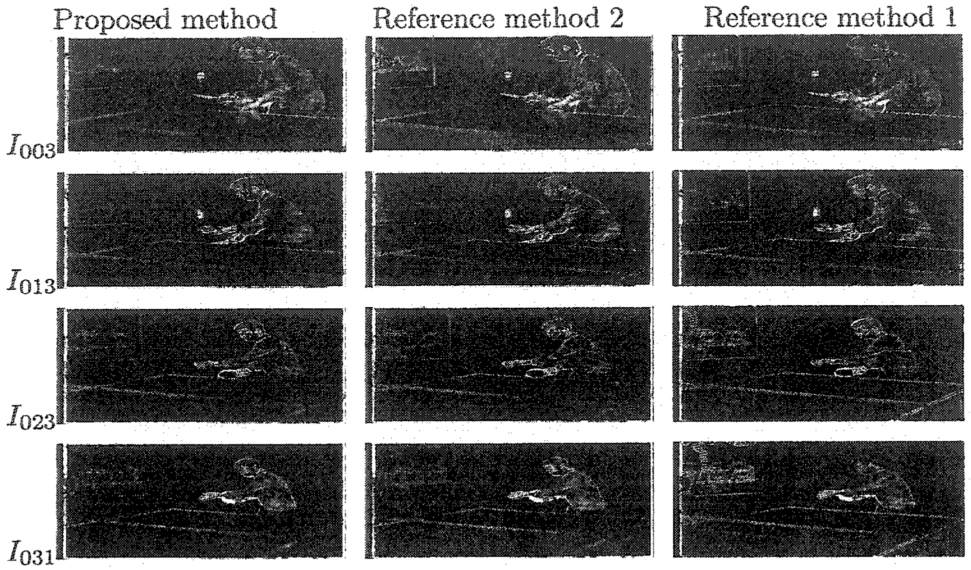


Figure 4.13: GME compared results of the difference frames of *Tennis* test sequence among the proposed method, the reference method 2 [9] and the reference method 1 [8].

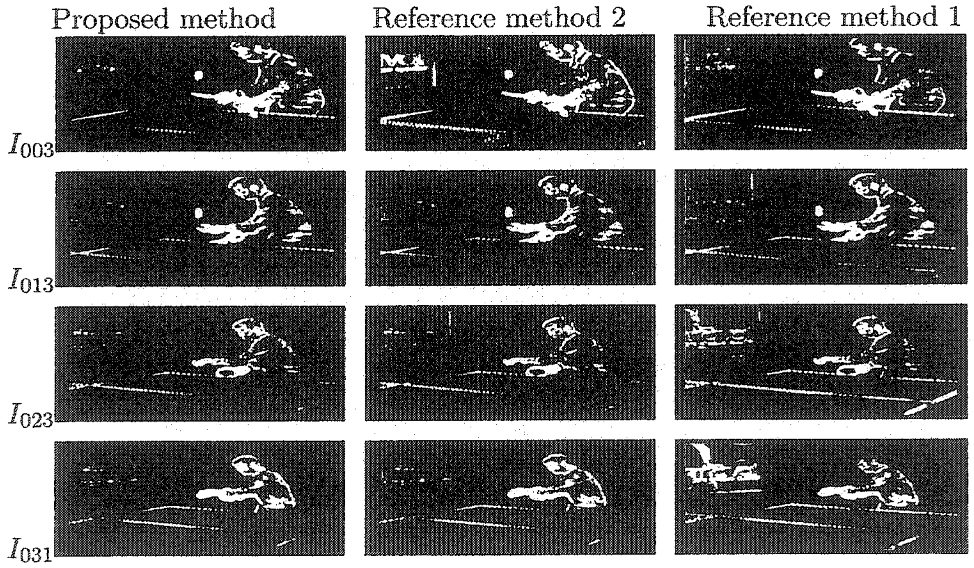


Figure 4.14: GME compared results of the binarized object frames of Fig. 4.13. This sequence contains camera zooming in to the object which is difficult for GME. The proposed method has the best performance compared to the reference methods.

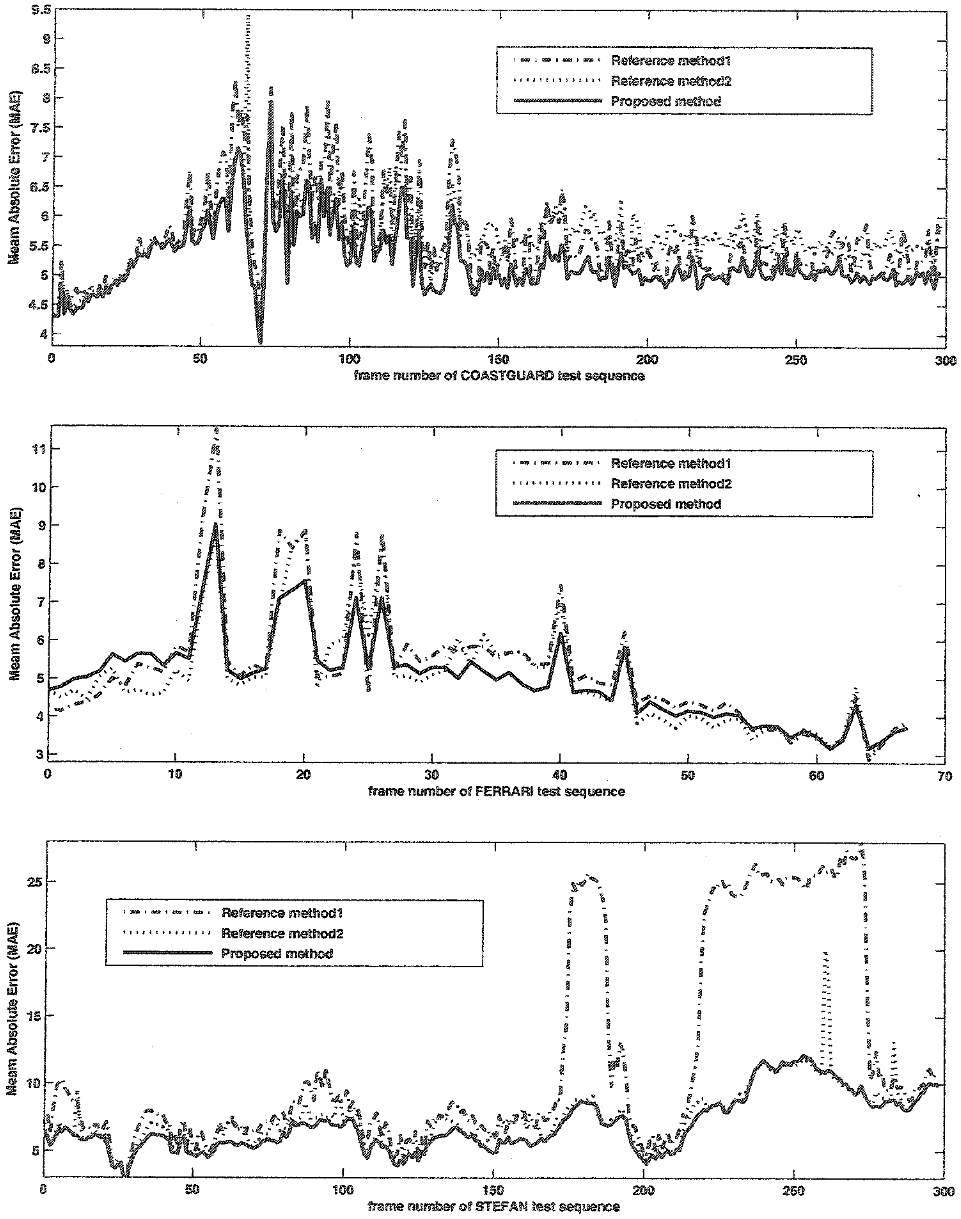


Figure 4.15: Objective GME compared results of MAE among the proposed method, the reference method 2 [9] and the reference method 1 [8]. In *Coastguard* sequence, there is a sudden tilt of camera motion during frame 70 causing relatively large MAE, while the proposed method shows stable MAE. In *Ferrari*, the proposed method has better subjective results compared to both the reference methods. However, they show similiar MAE in this figure. A discussion about this can be found in section 4.2.3. In *Stefan*, the camera moves very fast after frame 180. The reference methods have difficulties in estimating GME accurately in this situation, resulting in large MAE; while the proposed method remains stable MAE.

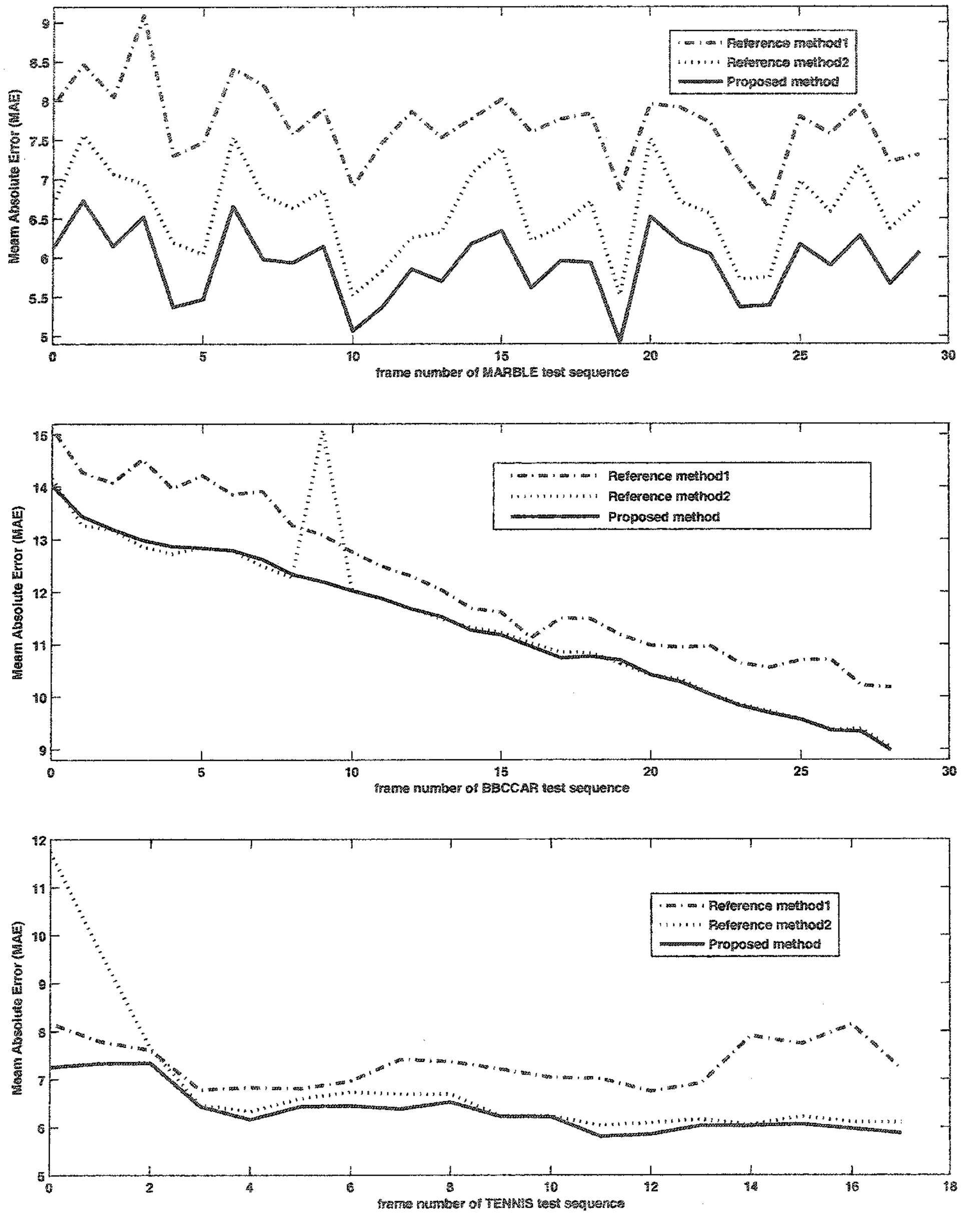


Figure 4.16: Objective GME compared results of MAE among the proposed method, the reference method 2 [9] and the reference method 1 [8] (continued). In *marble* sequence, the proposed method has the smallest MAE throughout the whole sequence. In *BBCcar*, the proposed method has almost the same MAE as the reference method 2. However, in frame 9, the reference method 2 fails in GME, while the proposed method succeeds. In *Tennis*, the proposed method has significently lower MAE for the first frame compared to both the reference methods because of the proposed robust estimator applied for the first frame.

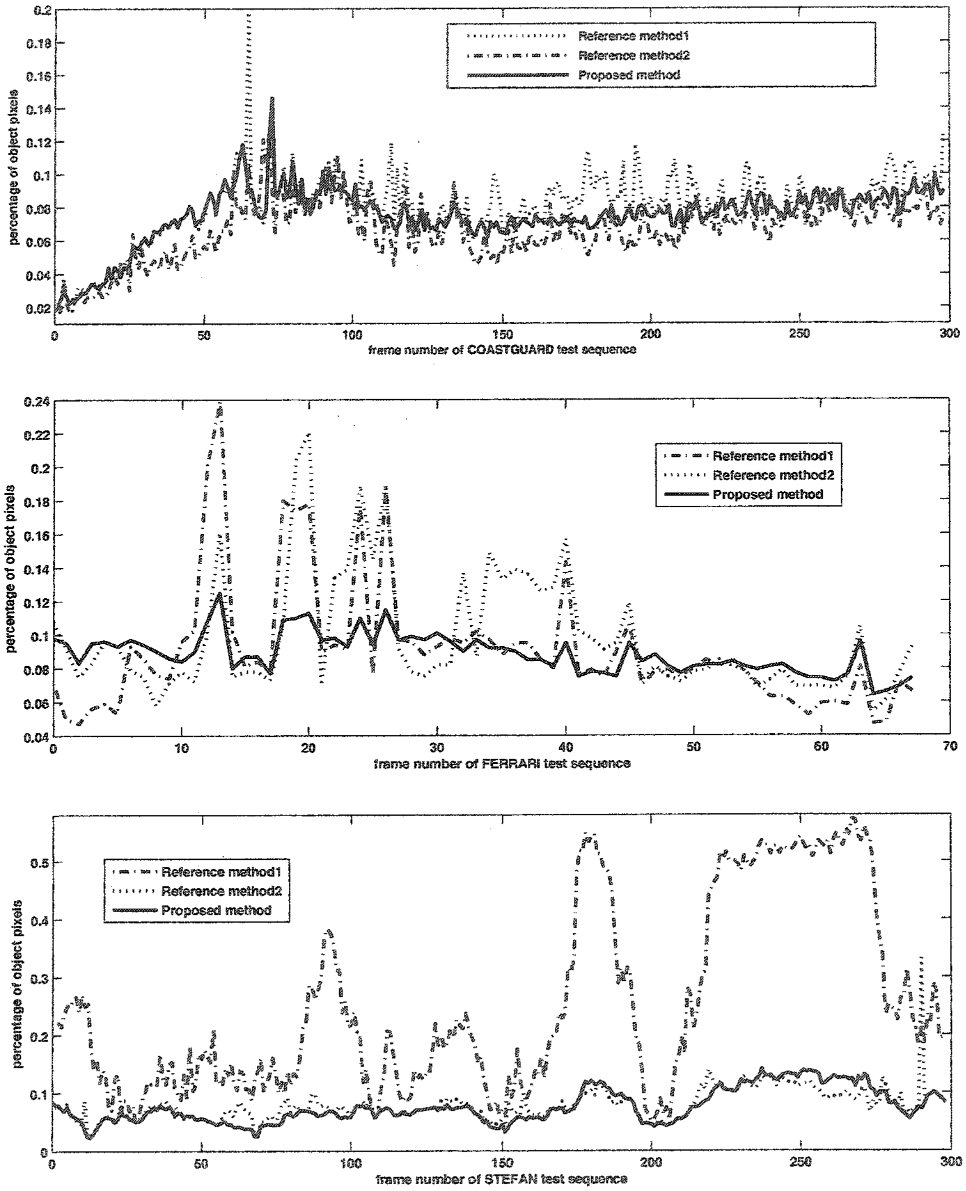


Figure 4.17: Compared results of the percentage of object pixels among the proposed GME method, the reference method 2 [9] and the reference method 1 [8]. Compared to both the reference methods, the proposed method achieves more stable results for each test sequences.

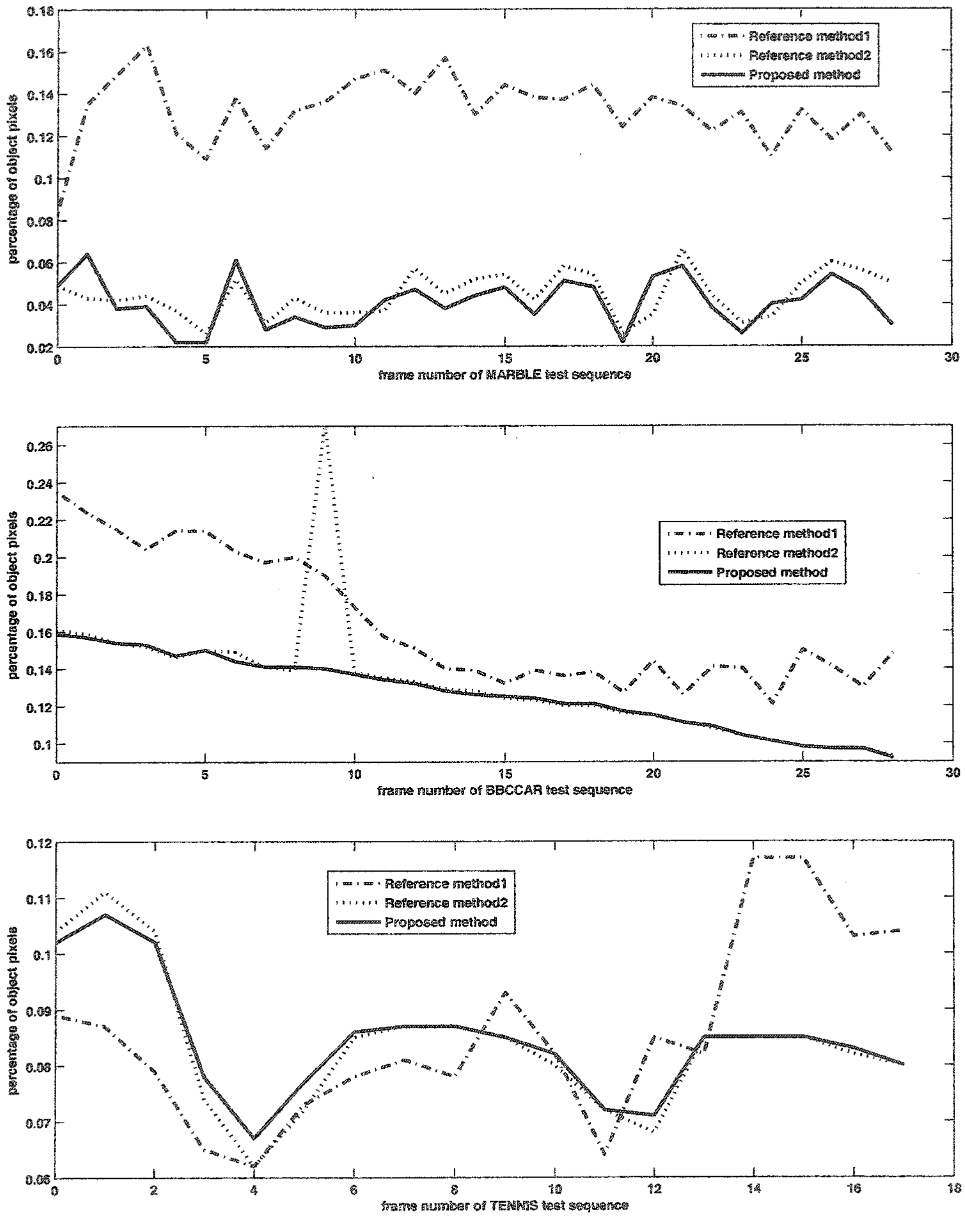
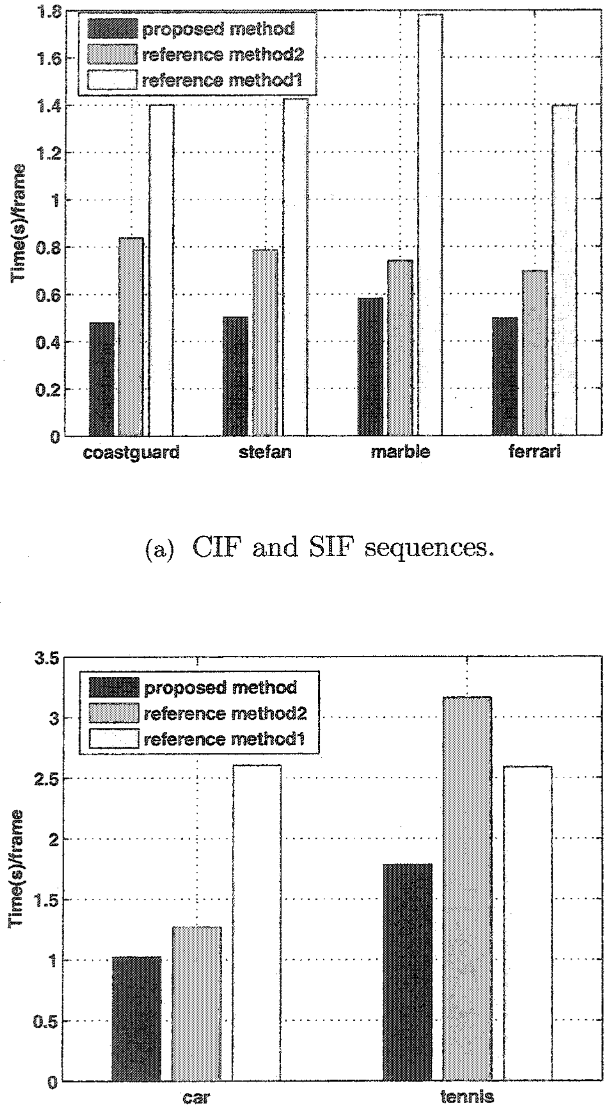


Figure 4.18: Compared results of the percentage of object pixels among the proposed GME method, the reference method 2 [9] and the reference method 1 [8] (continued). Compared to both the reference methods, the proposed method achieves more stable results for each test sequences.



(a) CIF and SIF sequences.

(b) PAL sequences.

Figure 4.19: Compared GME computational time for the proposed GME method, the reference method 2 [9] and the reference method 1 [8]. The proposed method is more efficient than both the reference methods.

4.2.3 Discussions

This section has presented the simulation results of the proposed GME method. Both subjective and objective results show that the proposed method is more robust, faster, and more suitable for object segmentation than the reference methods. Several discussions are addressed as follows:

- The evaluation of using MAE is not always suitable for the VOS oriented GME method because we want background motion to be compensated but leave the objects as residuals. Since the objects are often in the center of the frame, the prediction error can be calculated at the four corners of the frame (see Fig. 4.20). In this thesis, the corners are chosen to be the rectangles and the size of each rectangle is $0.25Rows \times 0.25Cols$. Fig. 4.21 shows the comparison corner MAE (CMAE) results of *Ferrari* test sequence. Compare Fig. 4.21 and Fig. 4.15, it can be seen that the proposed method has similar MAE but less CMAE than the reference methods.

- For interlaced test sequences, we tested several deinterlacing methods, but the results are not satisfactory and significantly affect GME. Thus, at present, we skip every other field when processing an interlaced test sequence.

- The proposed GME method is oriented to object segmentation. However, it has the potential extension for the other applications (e.g., video coding). Further experiments can be executed for its evaluation on this issue.

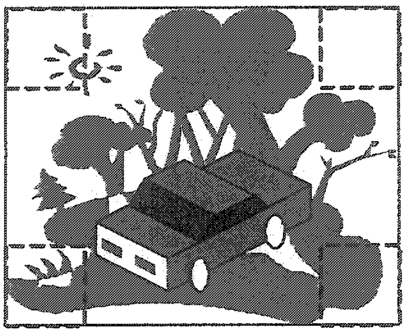


Figure 4.20: Calculate the prediction error at the four corners of the frame.

4.3 Results of the VOS scheme used

Simulation results of the VOS scheme used are carried out using standard test sequences *3cars, Survey, Hall, Stair, Coastguard, Ferrari, Stefan, BBCcar,* and *Tennis.* The first four are sequences without GM. The rest of the sequences include GM.

4.3.1 Subjective results

Fig. 4.22 to Fig. 4.27 show the subjective VOS results of randomly selected frames from test sequence *3cars, Survey, Coastguard, Ferrari, BBCcar,* and *Tennis.* We can see that the results are satisfactory in most cases. Fig. 4.28 to Fig. 4.30 show the comparison VOS results of randomly selected frames between the scheme used and the reference scheme [27]. It can be seen that the VOS scheme used contains less noises and artifacts than the reference scheme.

4.3.2 Objective results

We measure the performance of the VOS scheme used objectively using the objective measure presented in [38] where the inputs are the original frames and the binary object frames.

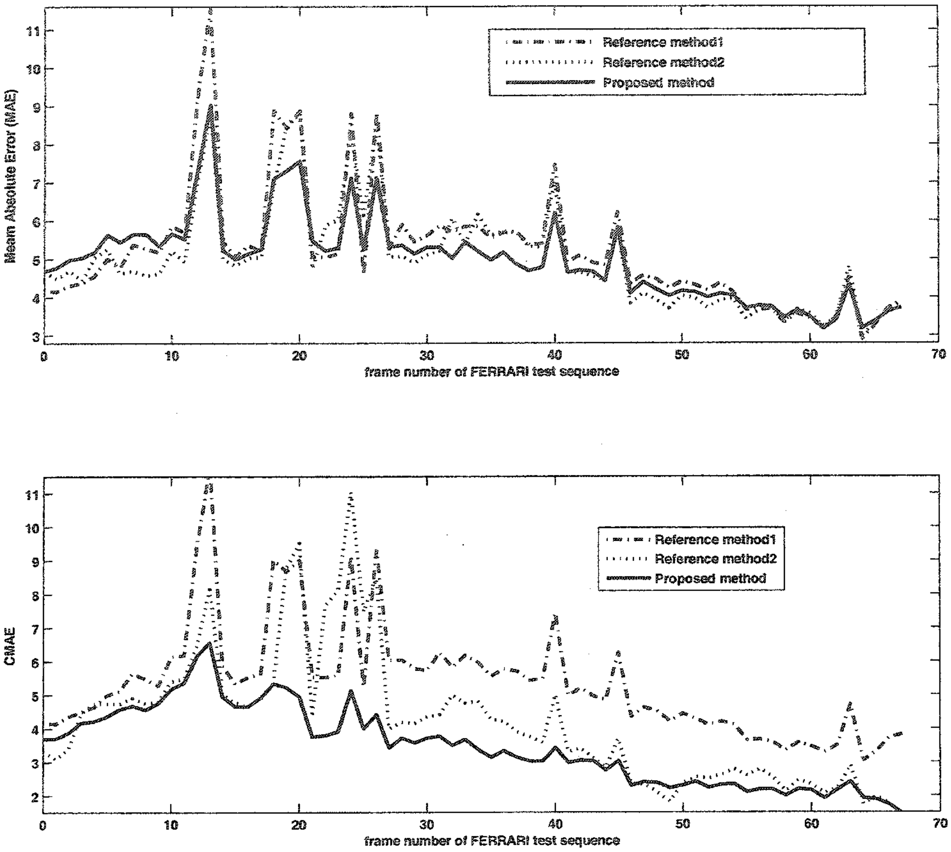


Figure 4.21: GME compared results of MAE and CMAE.

As stated in [38], there are three statistics calculated independently for performance measurement: spatial color contrast along object boundary, temporal color histogram difference, and motion difference along object boundary. These features can also be combined to give an overall score (normalized to one). The lower the score, the better the performance of the VOS[2].

Fig. 4.31 shows the comparison VOS measurements for *Hall* test sequence among the scheme used, the reference scheme [27], and the ground truth (i.e., manually segmented

[2] See appendix A for the details of this measure.

results). The scheme used has a lower score than the reference scheme and closer performance to the ground truth . Fig. 4.32 shows the comparison VOS measurements for *Stair* test sequence between the scheme used and the reference scheme [27]. The scheme used also here shows a lower score than the reference scheme. Fig. 4.33 shows objectively VOS measurements for test sequences with GM. It can be seen that the overall scores for all the test sequences are temporally stable.

Table 4.2 shows the computational time for each test sequences. The average VOS computational time per frame is 0.32 second for CIF/SIF test sequences and 0.78 second for PAL test sequences under multitasking Sparc-Sun-Solaris 2.8 1010MHz. The average computational time per frame of the whole process is 0.78 second for CIF/SIF test sequences and 2.18 second for PAL test sequences with GM.

sequence name	*GME computational time/frame*	*VOS computational time/frame*
Coastguard (CIF)	0.48s	0.27s
Ferrari (CIF)	0.50s	0.29s
Stefan (CIF)	0.50s	0.30s
Hall (CIF)	No GM	0.48s
3cars (SIF)	No GM	0.28s
Survey (SIF)	No GM	0.30s
BBCcar (PAL)	1.0s	0.81s
Tennis (PAL)	1.8s	0.75s

Table 4.2: Computational time for each test sequence.

4.3.3 Discussions

This section has presented the simulation results of the VOS scheme used. Both subjective and objective results show that the scheme used has satisfactory results for test sequences with and without GM. Several discussions are addressed as follows:

- The VOS scheme used has satisfied performance for noisy background. However, it can not extract the complete objects when the motion of the objects can hardly be detected (see frame 162 and 202 in Fig. 4.28). Better solutions may use object tracking and matching.

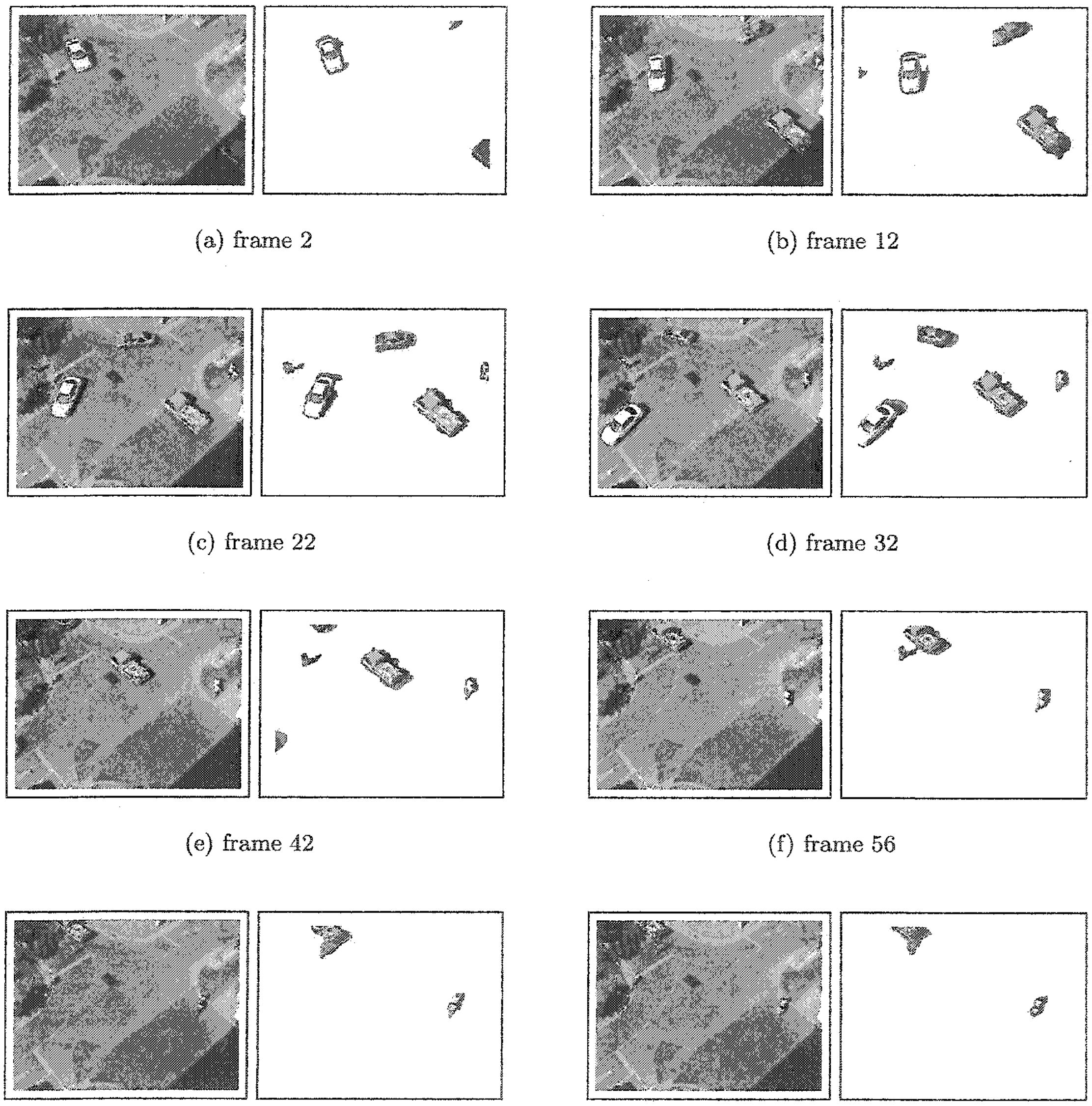
- The two thresholds for double thresholding can be better adapted through more tests.

- Post processing used in this thesis may connect close objects due to the gap filling method. Thus an adaptive gap filling is needed.

- If the sequence contains continuous complicated camera motion and can not be finely compensated, it will effect the segmentation results (see Fig 4.27 where the tennis table is not finely compensated).

4.4 Summary

This chapter has presented the simulation results of the proposed GME method and the VOS scheme used. Both subjective and objective results are presented and compared to the reference methods. Several discussions are also addressed. It shows that the proposed GME method is more robust, faster, and more suitable for VOS than the reference methods [9, 8]. The VOS scheme used is efficient and noise-adaptive that achieves satisfactory results for several standard test sequences with and without GM.



(a) frame 2

(b) frame 12

(c) frame 22

(d) frame 32

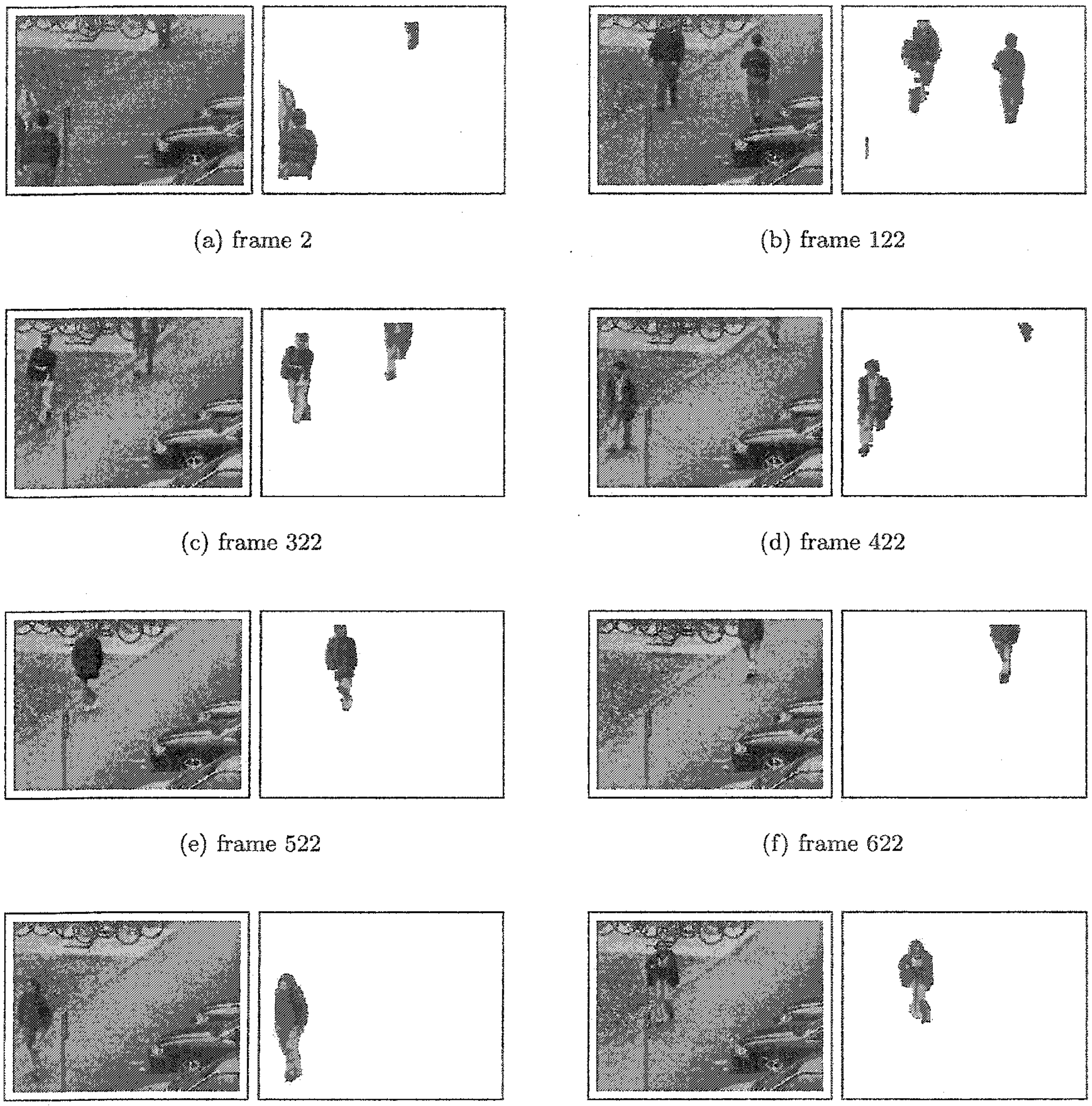
(e) frame 42

(f) frame 56

(g) frame 62

(h) frame 65

Figure 4.22: Randomly selected segmentation results of *3cars* test sequence (no GM). The multi-objects are successfully segmented.



(a) frame 2

(b) frame 122

(c) frame 322

(d) frame 422

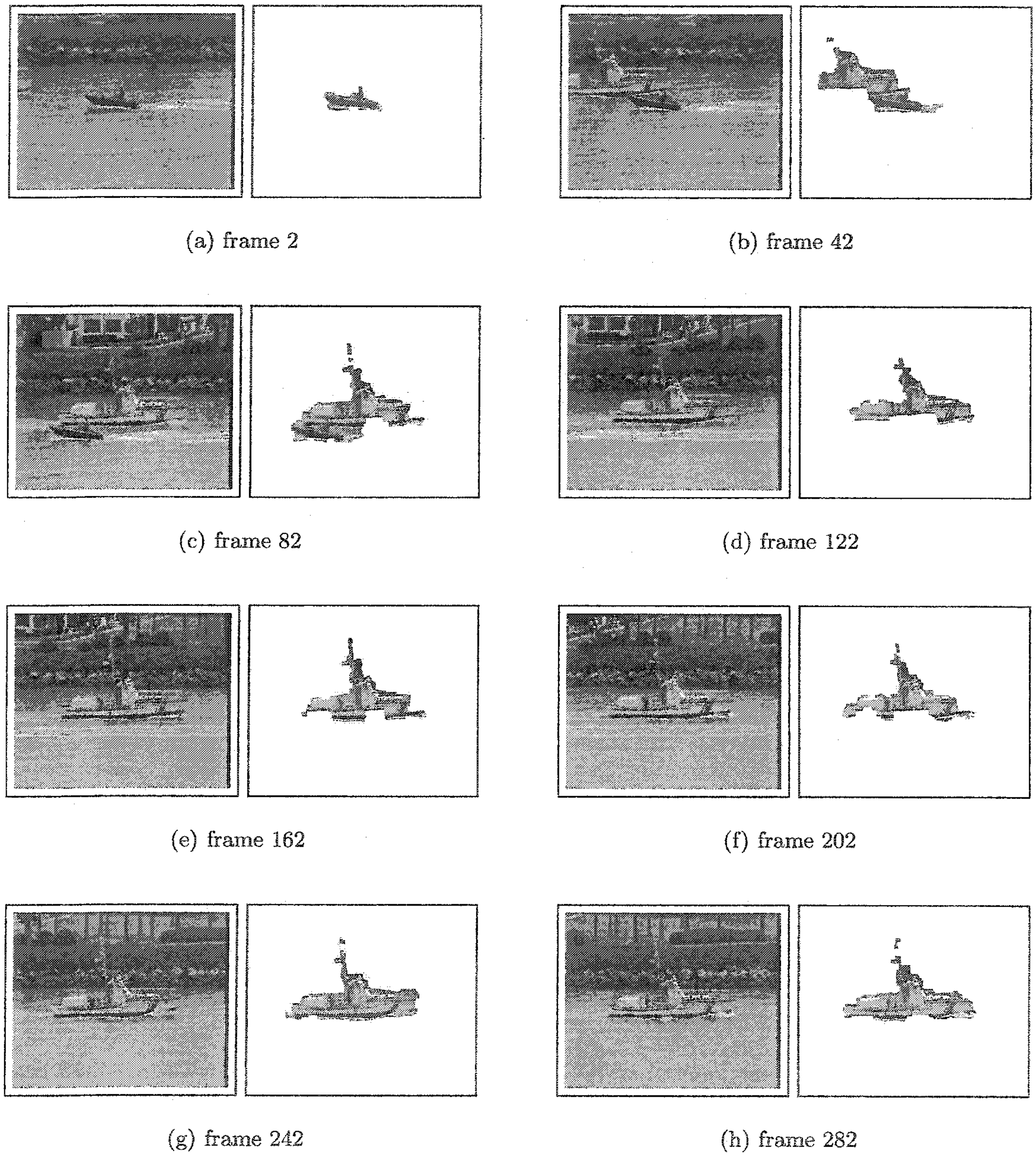
(e) frame 522

(f) frame 622

(g) frame 722

(h) frame 922

Figure 4.23: Randomly selected segmentation results of *Survey* test sequence (no GM). The objects are successfully detected. However, in some frames, the objects are not complete (see frame 122) due to the difficulty of detecting slightly moving objects.



(a) frame 2

(b) frame 42

(c) frame 82

(d) frame 122

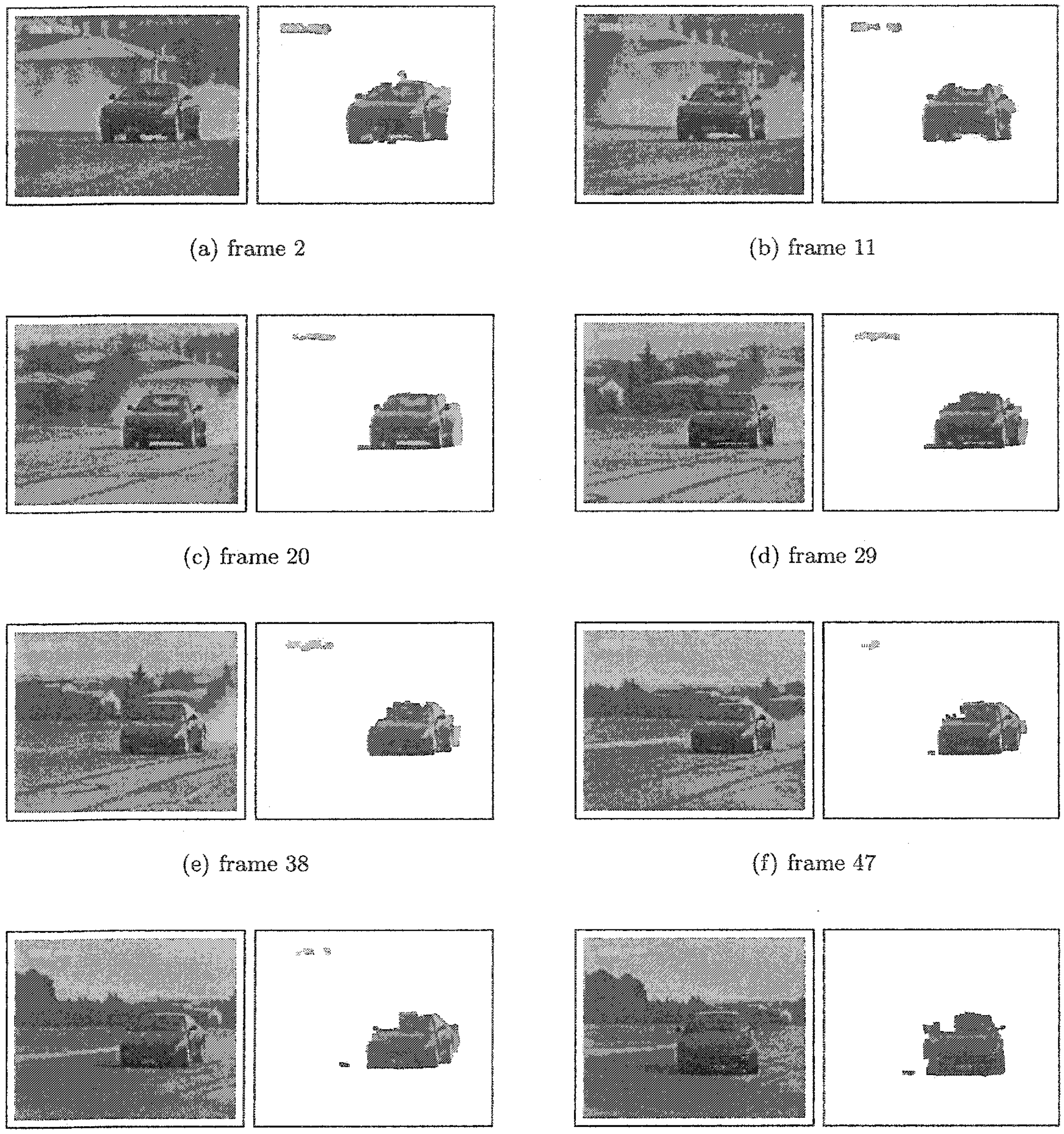
(e) frame 162

(f) frame 202

(g) frame 242

(h) frame 282

Figure 4.24: Randomly selected segmentation results of *Coastguard* test sequence (with GM). The overall segmentation results are satisfied for this sequence.



(a) frame 2

(b) frame 11

(c) frame 20

(d) frame 29

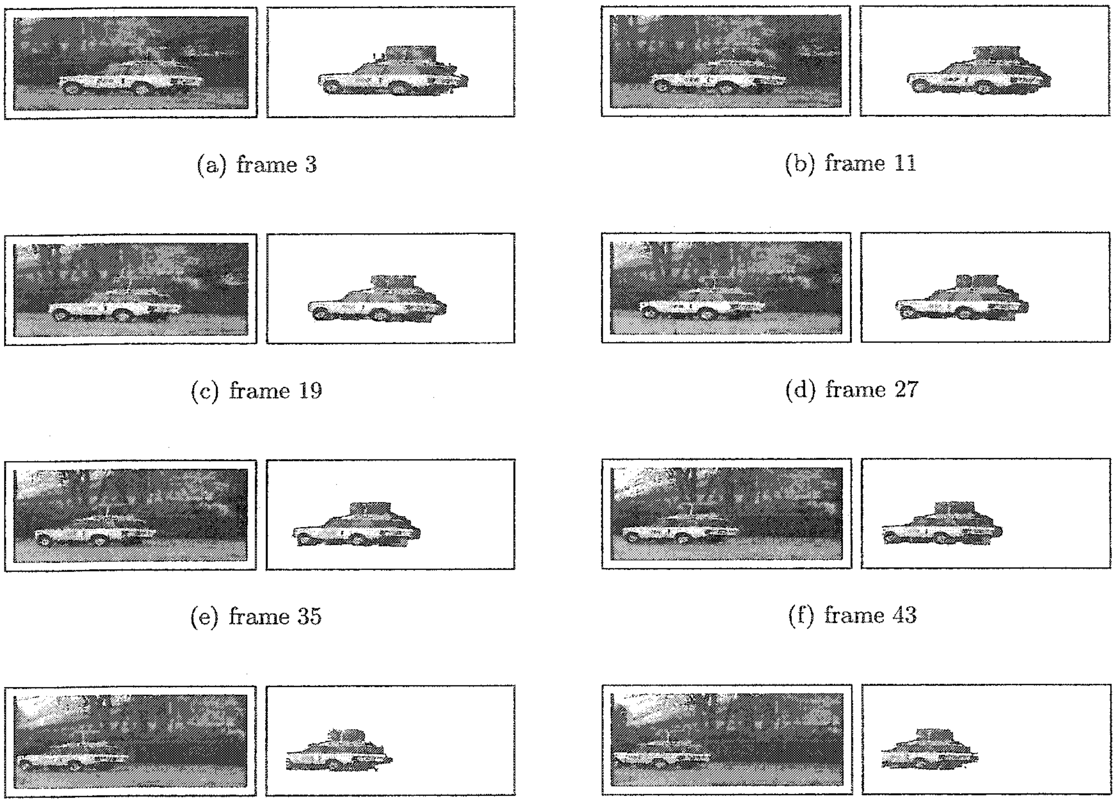
(e) frame 38

(f) frame 47

(g) frame 56

(h) frame 65

Figure 4.25: Randomly selected segmentation results of *Ferrari* test sequence (with GM). The results are satisfied except that there are some occlusions in the last few frames. This is because part of the vehicles can hardly be detected in both change detection and edge detection.



(a) frame 3

(b) frame 11

(c) frame 19

(d) frame 27

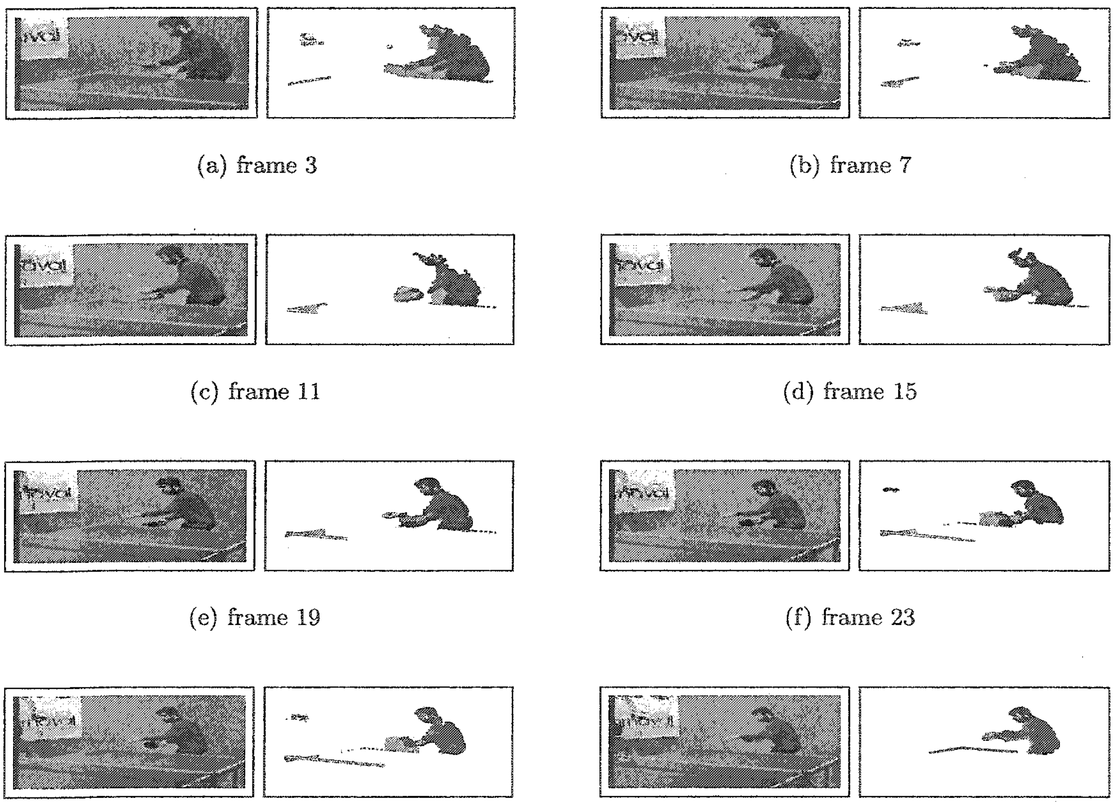
(e) frame 35

(f) frame 43

(g) frame 51

(h) frame 59

Figure 4.26: Randomly selected segmentation results of *BBCcar* test sequence (with GM). The results are satisfied and stable for this sequence.



(a) frame 3

(b) frame 7

(c) frame 11

(d) frame 15

(e) frame 19

(f) frame 23

(g) frame 27

(h) frame 31

Figure 4.27: Randomly selected segmentation results of *Tennis* test sequence (with GM). This sequence also contains residual from GME.

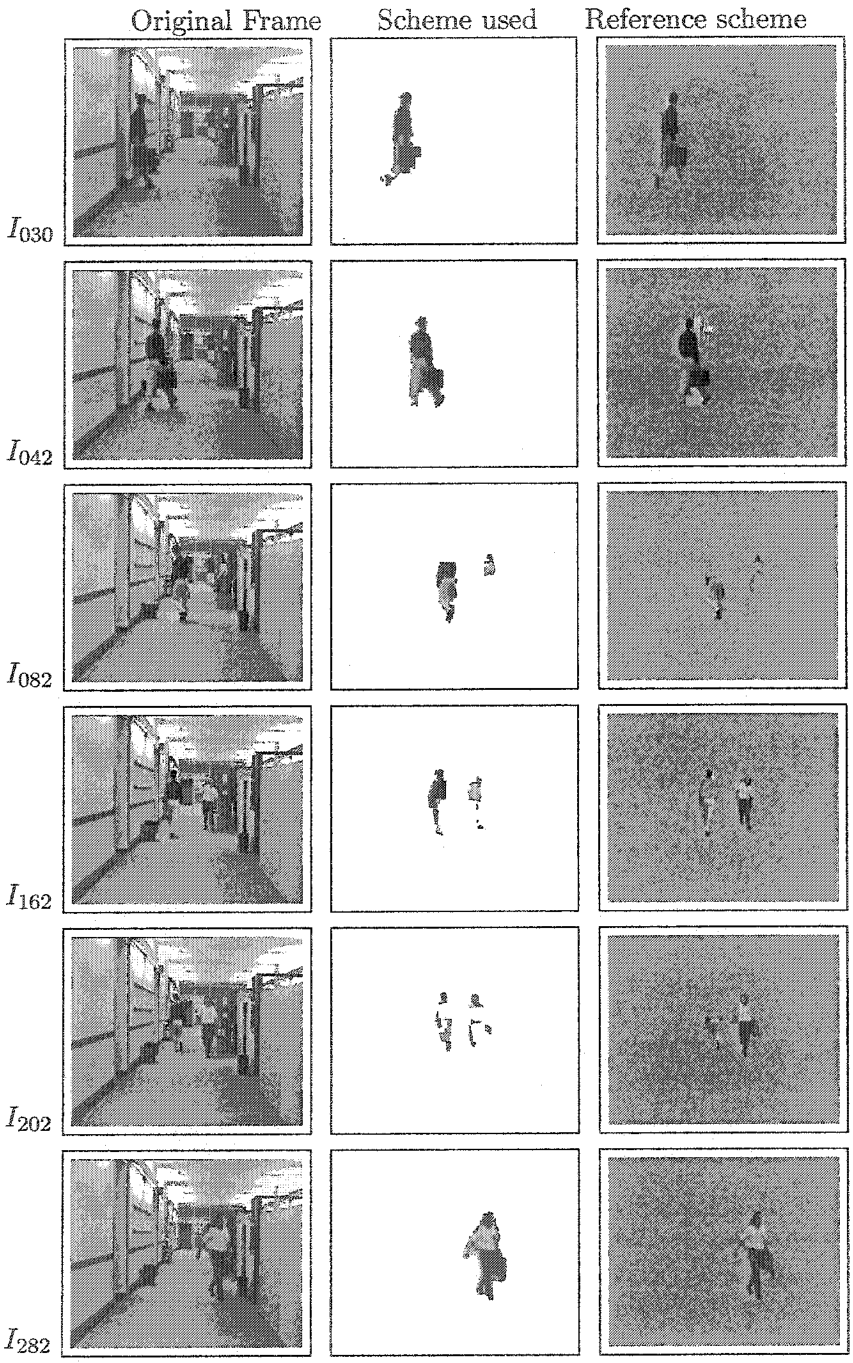


Figure 4.28: Randomly selected VOS compared results of *Hall* test sequence (no GM)between the scheme used and the reference scheme [27]. The scheme used has compatible results to the reference method. However, both methods encounter problems to detect slightly moving objects.

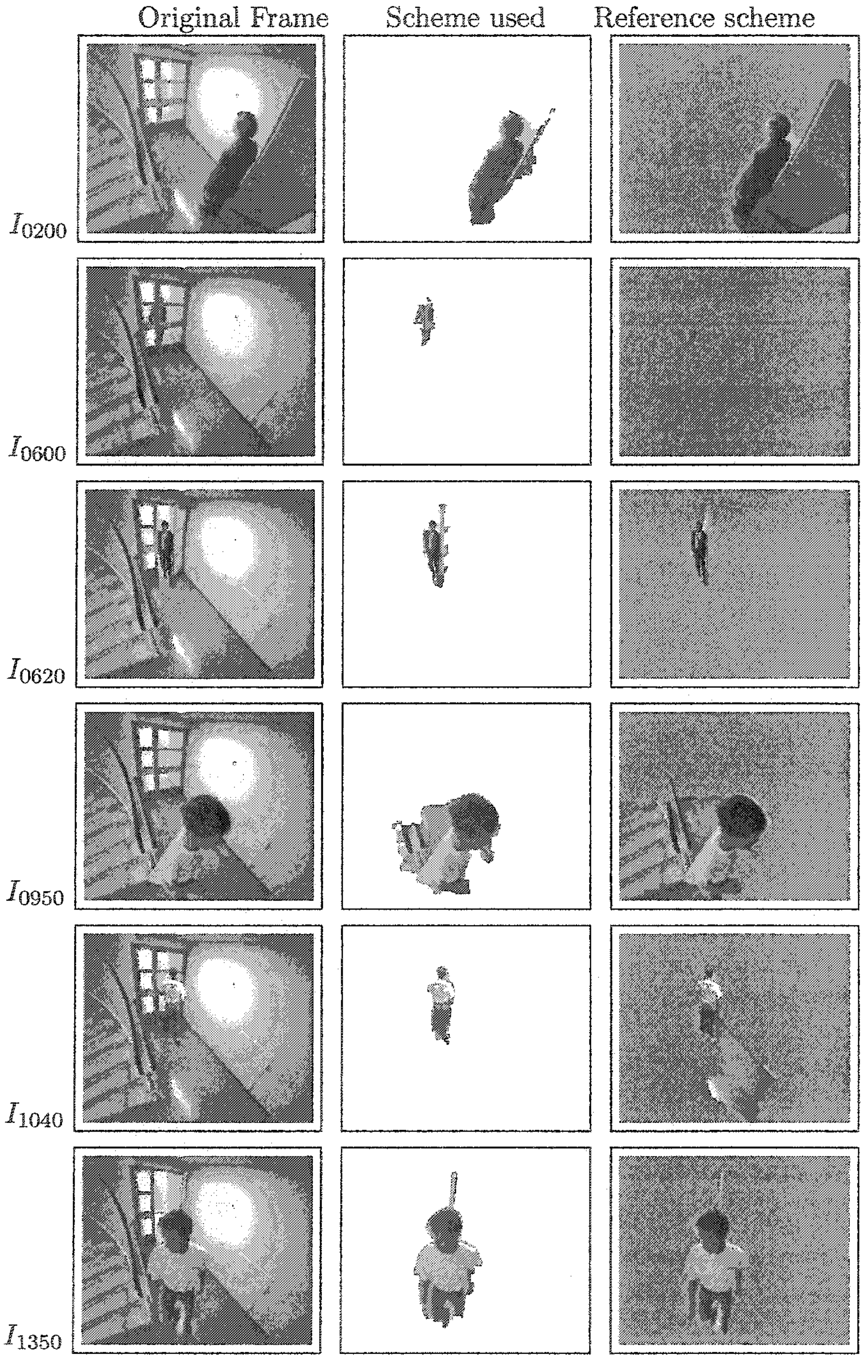


Figure 4.29: Randomly selected VOS compared results of *Stair* test sequence (no GM) between the scheme used and the reference scheme [27]. The scheme used has less artifacts than the reference method for this sequence (see Frame 200, 950, and 1040). Also the scheme used has better performance to preserve small objects than the reference method (see Frame 600).

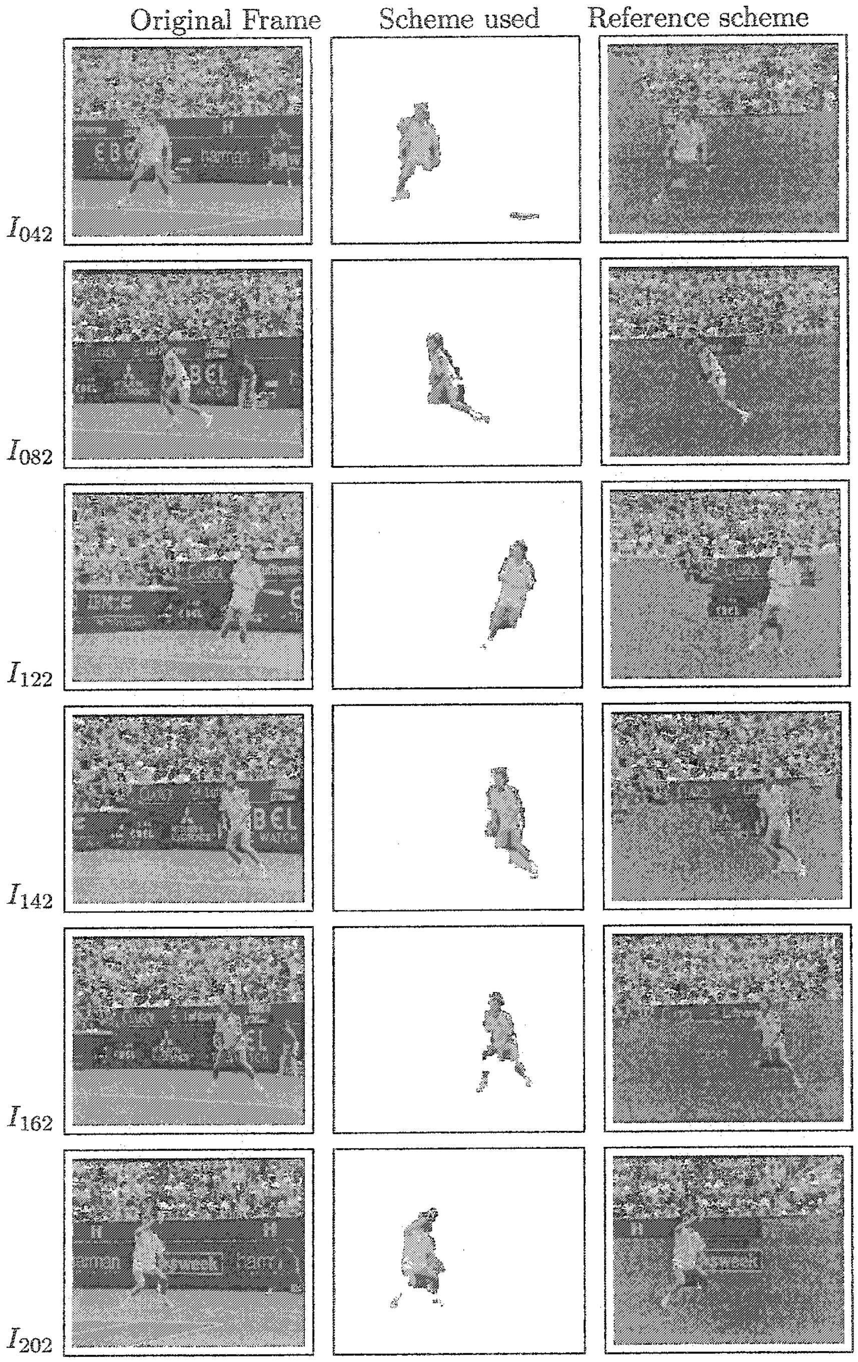


Figure 4.30: Randomly selected VOS compared results of *Stefan* test sequence (with GM) between the scheme used and the reference scheme [27]. The segmentation results of the scheme used are better than the reference method because of the accurate GME and GMC as well as the noise-robust change detection.

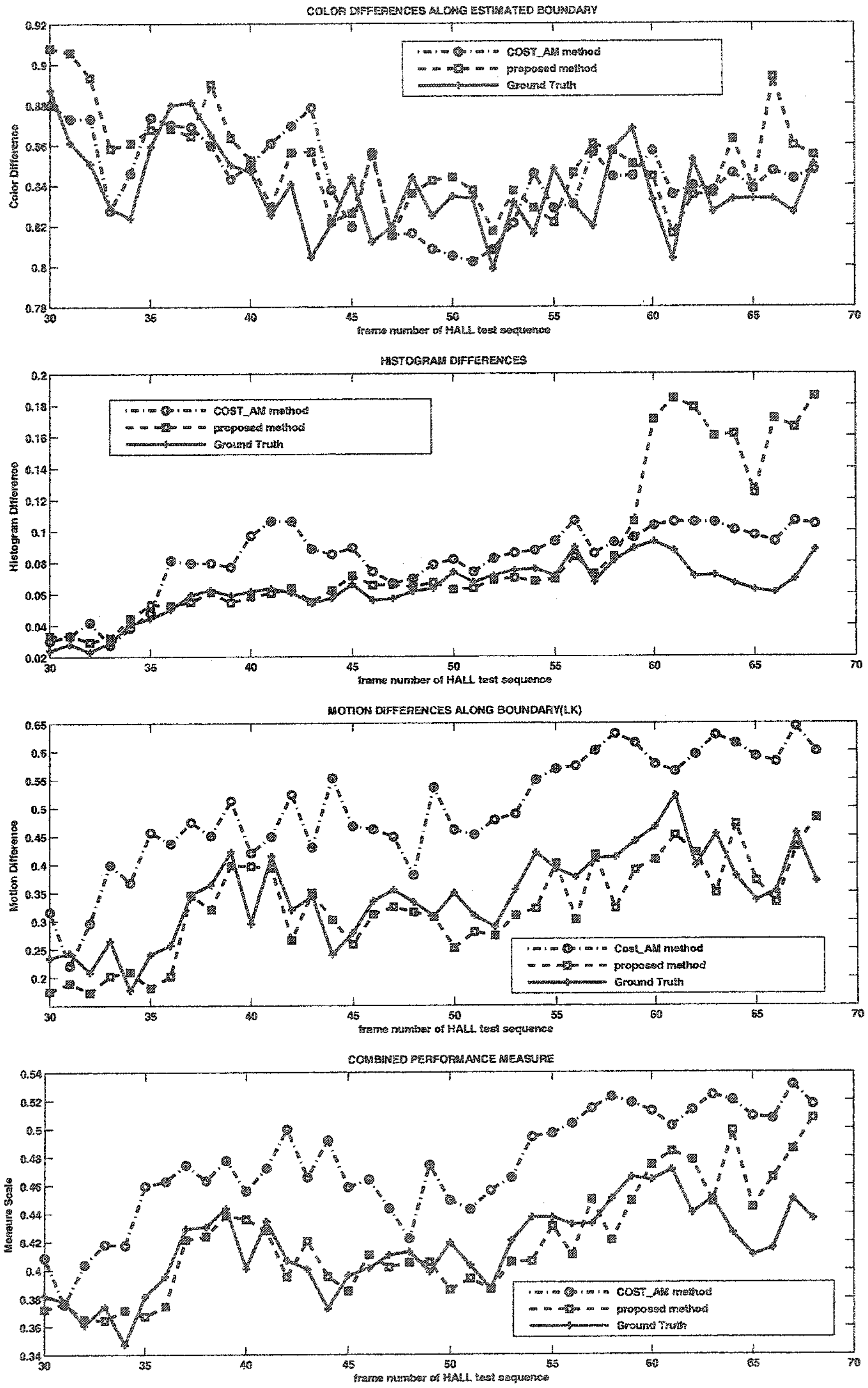


Figure 4.31: Objective performance measure of *Hall* test sequence using the reference scheme [27], the scheme used, and the ground truth. The scheme used has a lower score than the reference scheme which indicates that the scheme used has better performance than the reference scheme. Note that the scheme used also has closer performance than the reference scheme to the ground truth.

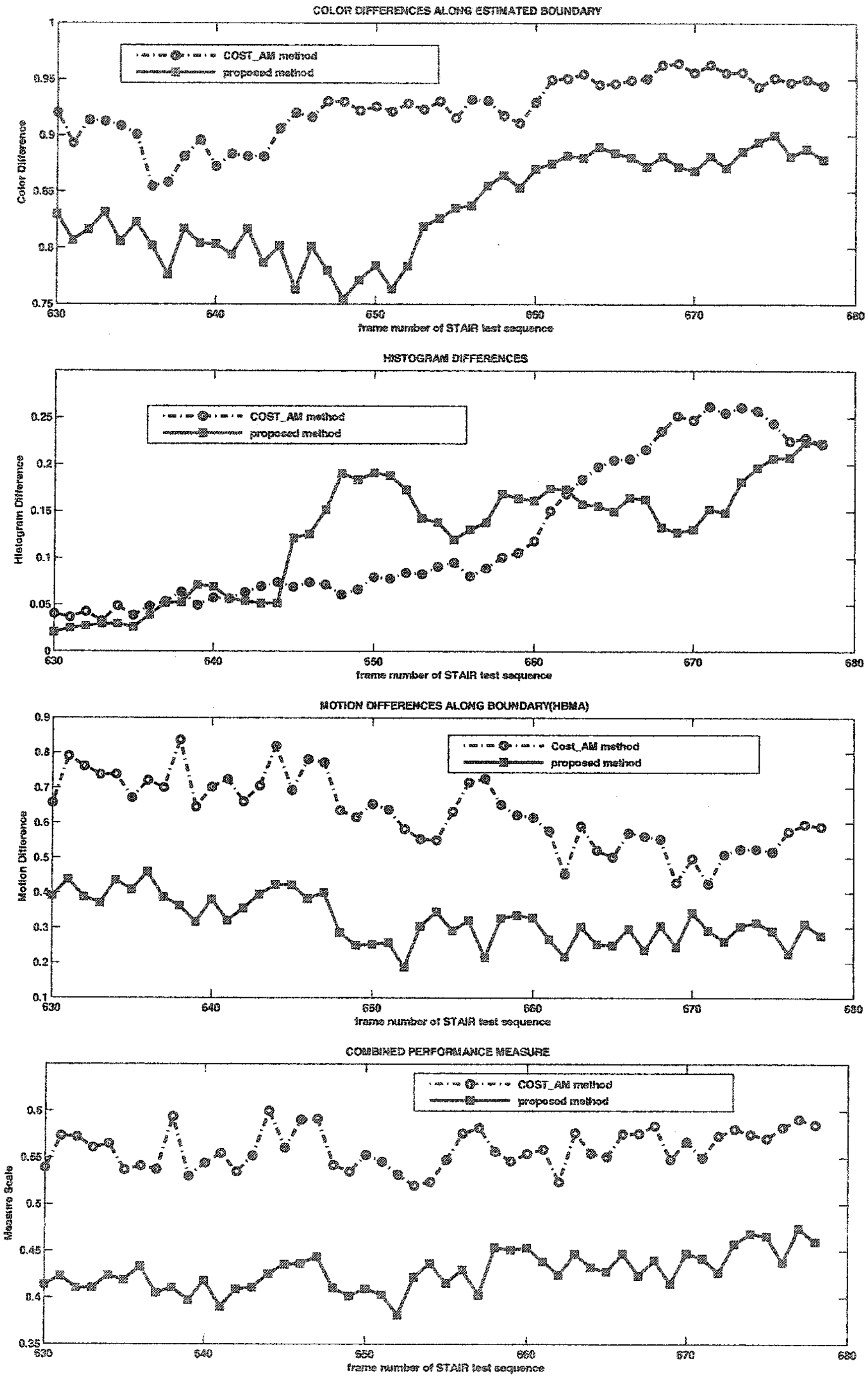


Figure 4.32: Objective performance measure of *Stair* test sequence using the reference scheme [27] and the scheme used. The scheme used has a lower overall score than the reference scheme, which means that the scheme used has a better performance than the reference scheme.

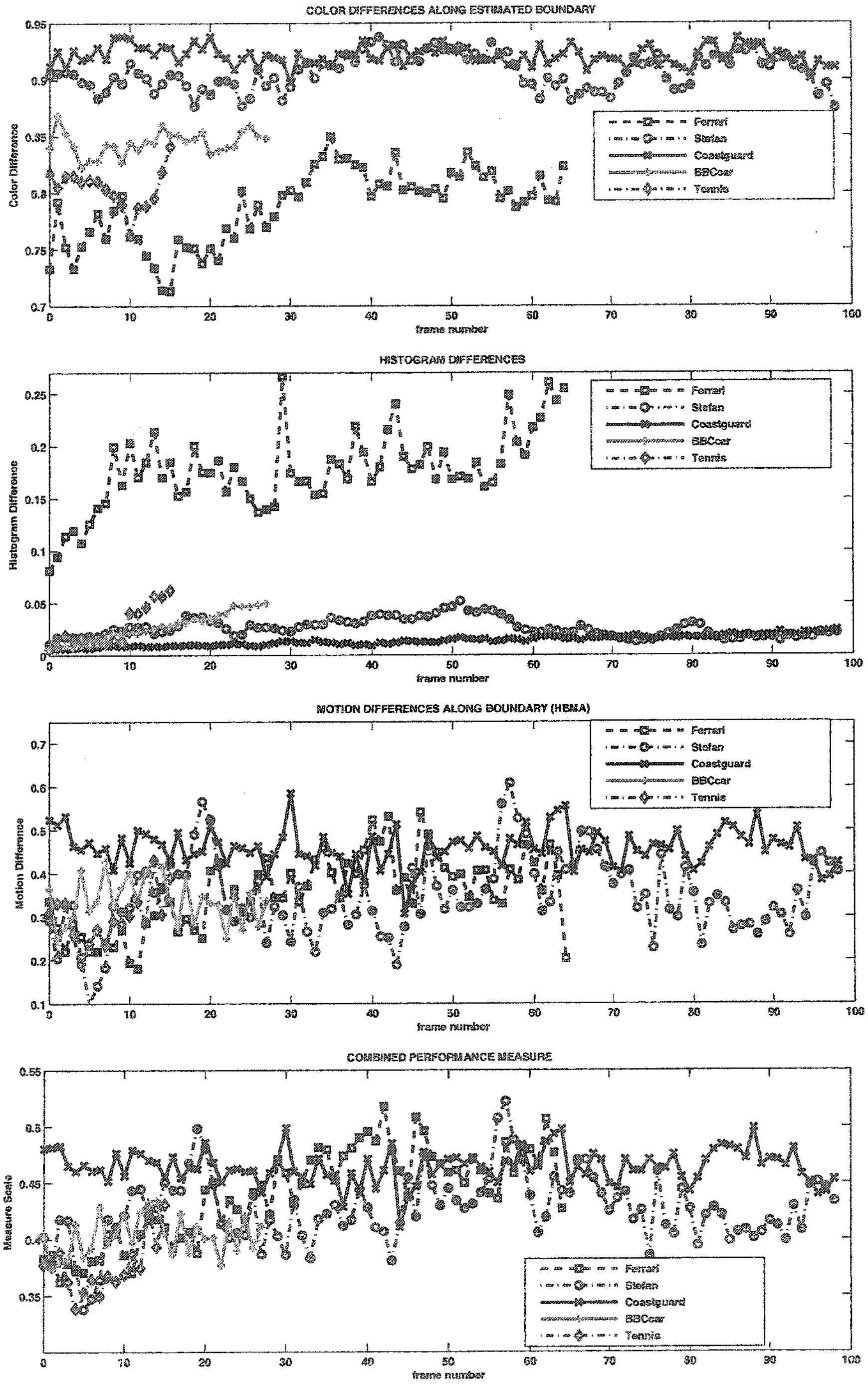


Figure 4.33: Objective performance measure of test sequences with GM. The overall scores for all the test sequences are within a scale and temproally stable, which indicate that the scheme used achieves satisfactory performances for these test sequences.

Chapter 5

Conclusion

5.1 Summary

Most object segmentation methods either assume that there is no GM or directly adopt a coding-oriented GME method to estimate the GM. However, the objective of GME in video coding is different from the one in object segmentation. This thesis has proposed a fast and robust GME method oriented to object segmentation. Furthermore, this GME method has been integrated into a modular object segmentation scheme. This thesis has proposed some improvements (see section 5.2) within this scheme.

The proposed methods aim at four goals: 1) automatically adapt to GM, 2) robust (insensitive) to noise and clutter, 3) stable segmented objects over time, and 4) low computational cost. This thesis has approached these goals through the detection, estimation and compensation GM, through the adaptation to noise and reduction of clutter, through the combination of temporal and spatial information, and through modularity of the proposed methods.

The proposed methods (GM detection, GME, and VOS) provide a response of 0.32 second per frame for CIF/SIF video sequences without GM, and 1.0 second per frame with GM on a multitasking Sparc-Sun-Solaris 2.8 1010MHz without specialized hardware. The reliability of the proposed methods have been demonstrated by experimenting on more than 10 indoor and outdoor video shots containing a total of 6371 frames including sequences with and without GM. Both subjective and objective simulation results show that the proposed GME method is more efficient and can achieve better results than the reference methods. For object segmentation, encouraging results were also achieved.

5.2 List of Contributions

Contributions of this thesis are:

- a fast and noise-robust GM detection technique which can detect GM without estimating the GM parameters. The proposed GM detection method first applies a change detection method [28] between the current and the previous frame. Then the binary frame is divided into *nine* equal blocks and the sum of the weighted number of blocks with the white pixels beyond a threshold is calculated. To adapt to the noisy sequences, the window size of the spatial average filter for change detection is adjusted with the PSNR of the sequence.

- a fast and robust GME method oriented to VOS. This method combines basic GME principles for video coding and adds several improvements for VOS. Contributions in this part include:

  1. a combination of 3-step search and MV prediction for initial motion estimate. The traditional 3-step search for initial motion estimate is applied for the first *six* frames of the processed video sequences. For the rest of the frames, a fast MV prediction technique is applied instead.

  2. using residual (object) information from the previous frame for robust estimation. This residual information is used to eliminate outliers when estimating GM

parameters of the next frame. Note that since this residual information is reused for VOS, there is no computational cost involved.

3. a new robust estimator considering the neighborhood for the first compensated frame to improve the accuracy of the GME result since there is no residual information available at that time.

- improvements to a VOS scheme based on change detection using GM compensated frame difference. The improvements in this scheme are:

    1. an improved morphological double thresholding technique to remove the background residual clutters and to enhance the presence of the objects. This technique obtains two binary frames using two thresholds obtained from a noise-robust thresholding method [28]. Then morphological reconstruction is applied to simplify the frame.

    2. a new temporal adaptation technique to obtain more stable object results. First, a buffer is built and updated in each frame to detect the newly appeared objects. Then the adaptation ranges are set up as the dilated and eroded previous object frame. The adaptation is processed using the buffer information as well as the adaptation ranges.

    3. a new edge adaptation technique to improve the accuracy at object boundaries. This task consists of two steps: edge detection, and edge warping. First, spatial luminance edge of the current frame as well as the binary edge of the object mask is detected. Then every binary edge pixel is warped to the nearest luminance edge pixel within an adaptive search range.

5.3 Conclusion

The conclusion of this thesis is drawn in this section and is divided into three parts: global motion detection, global motion estimation and video object segmentation:

Global motion detection

In the proposed scheme, GM detection is a condition to decide whether or not GME is necessary for the processed sequence. Applying GME conditionally using the proposed fast GM detection method can speed up the computational time significantly. It also avoids possibly estimating errors imported by GME in case no GM is available.

Global motion estimation

- An accurate GME result is a critical step for a successful object segmentation. The proposed GME method is the most important contribution in this thesis since it achieves satisfactory results that outperform the reference methods. A paper based on this method was submitted and is accepted for publication by a prestigious IEEE conference on video and image processing (see appendix B). The proposed GME method contains three contributions. The first one lies in the initial estimation to speed up computational time and to improve the results. However, it is the two proposed robust estimators that contribute the most to achieve the major improvements. This is because these two robust estimators can identify the object pixels and reject these pixels as outliers for GME.

- The proposed GME method is designed for VOS since it aims at successfully compensating the background and extracting the objects using the remaining differences between the current and the compensated frames. While in video coding, the computed motion need not resemble the *true* motion of frame points as long as some minimum bit rate is achieved.

- The objects may not be detected even if the background is successfully compensated if the motion of the objects is very small (see the results of *Marble* test sequence in Fig. 4.9 and Fig. 4.10). Unless other information is added, this change detection based scheme can not handle these situations.

Video object segmentation

- The VOS scheme used consists of several other modules except GME and GMC. Some of these modules are proposed as part of this thesis; others are adopted as is to construct the whole scheme. Each of the modules used is analyzed as follows:
  - The change detection method [28] used performs well in reducing the noises and artifacts because of the spatial averaging filter and the MAX filter applied to the difference frames. The proposed double thresholding technique has visual improvements compared to the basic thresholding, especially when the background is heavily cluttered.
  - The proposed temporal adaptation technique improves the results by two strategies: first, it uses a buffer to identify the newly appeared objects from the clutter. Second, it attempts to track the objects by applying morphological dilation and erosion to the previous objects. The benefit of using this module is to stabilize the segmentation results. Since it adapts well for the test sequences, it can be considered as the most significant improvement in the VOS scheme.

  - The methods used for post processing of the binary frame are not adaptive as they may connect district objects. Further resolution in adaptive methods is needed.
  - The purpose of adding an edge adaptation module is to improve the accuracy at object boundaries. More accurate edge detection in the gray-level frame is needed (e.g., using other edge detector than Sobel).

- The VOS scheme used is modular. Some parameters can be adjusted according to the characteristics of the processed video sequences.

- The VOS scheme is not designed for a specific application. However, due to its efficiency and robustness, it can be applied in real-time video applications such as video surveillance, video editing and video coding.

5.4 Possible Extensions

There are a number of issues to be considered to enhance the performance of the proposed methods:

- Change detection methods have a common difficulty in detecting complete objects when the objects slightly move. Adaptive solution lies in combining other features, such as spatial information or object matching.

- Post processing (gap filling and morphological closing) techniques used in this thesis may connect close objects. Applying them adaptively using extra conditions may solve this problem. For example, a conditional morphological dilation and erosion technique is presented in [13].

- Many edge detectors are proposed in the literature with different pros and cons. Other edge detectors can be tested and applied instead of Sobel edge detector. For example, Roberts edge detector may have better performance in eliminating shadows.

- With an accurate de-interlacing method for interlaced sequences, the proposed method can be conducted using consecutive fields instead of every other field.

- Since GME and VOS interact in this thesis, better integration between these two methods would help achieve better overall results.

Bibliography

[1] D. Zhang, G. Lu, "Segmentation of Moving Objects in Image Sequence: A Review" *Circuits, System and Signal Processing,* Vol.20(2), pp.143-183, 2001.

[2] "Television Viewing" *The Daily, Statistics Canada,* Nov.21, 2003.

[3] "MPEG-4 Video Verification Model Version 18.0" ISO/IEC *JTC1/SC29/WG11 MPEG2001/N3908,* Pisa, January, 2001.

[4] L.Hill, T.Vlachos, "On the estimation of global motion using phase correlation for broadcast applications" *IPA '99, Proceedings of IEE Internation Conference on Image Processing and Its Applications,* Vol.2, pp.721-725, July 1999.

[5] S.Kumar, M.Biswas, T.Q Nguyen, "Global Motion Estimation in Frequency and Spatial Domain" *ICASSP '04, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.3, May 2004.

[6] F. Moscheni, F. Dufaux, M. Kunt, "A New Two-stage Global/local Motion Estimation Based on a Background/foreground Segmentation" *ICASSP '95, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.4, pp.2261-2264, 1995.

[7] C. Hsu, Y. Tsan, "Mosaics of Video Sequences with Moving Objects" *ICIP '01, Proceedings of IEEE International Conference on Image Processing,* Vol.2, October 2001.

[8] J.S.Lee, K.Y.Rhee, S.D.Kim, "Moving Target Tracking Algorithm Based on the Confidence Measure of Motion Vector" *ICIP '01, Proceedings of IEEE International Conference on Image Processing,* Vol.1, pp.369-372, October 2001.

[9] F.Dufaux, J.Konrad, "Efficient, Robust and Fast Global Motion Estimation for Video Coding" *IEEE Transactions on Image Processing,* Vol.9, No.3, March 2000.

[10] W.C.Chan, O.C.Au, M.F.Fu "Improved Global Motion Estimation Using Prediction and Early Termination" *ICIP '02, Proceedings of IEEE International Conference on Image Processing,* Vol.2, Sept.2002.

[11] Y.He, B.Feng, S.Yang, Y.Zhong "Fast Global Motion Estimation for Global Motion Compensation Coding" *ISCAS '01, Preceeding of IEEE International Symposium on Circuits and systems,* Vol.2, May 2001.

[12] T.Koga, K.Iinuma, A.Hirano, Y.Iijima, and T.Ishiguro "Motion Compensated Interframe Coding of Video Conferencing" *Proc. Nat. Telecommun. Conf.,* pp. G5.3.1-G5.3.5, Dec.1981.

[13] A. Amer and Eric Dubois "Fast and Reliable Structure-Oriented Video Noise Estimation" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 15, No. 1, January 2005, pp. 113-118.

[14] A. Amer "Binary Morphological operations for effective low-cost boundary detection" *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI),* vol. 17, No. 2, pp. 201-213, March 2003.

[15] N.Li, S.Li, W.Y.Liu, C.Chen, "A Novel Framework for Semi-automatic Video Object Segmentation" *ISCAS '02, Preceeding of IEEE International Symposium on Circuits and systems,* Vol.3, pp.811-814, May 2002.

[16] R.Castagno, T.Ebrahimi, M.Kunt, "Video Segmentation Based on Multiple Features for Interactive Multimedia Applications" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.8, No.5, pp.562-571, Sep.1998.

[17] H.F.Chen, F.H.Qi, S.Zhang, "Supervised Video Object Segmentation Using A Small Number of Interactions" *ICASSP '03, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.3, April 2003.

[18] J.Guo, J.Kim, C.-C.J.Kuo, "Fast Video Object Segmentation Using Affine Motion and Gradient-based Color Clustering" *1998 IEEE Second Workshop on Multimedia Signal Processing,* pp.486-491, Dec.1998.

[19] W.Zeng, W.Gao, "Accurate Moving Object Segmentation by A Hierarchical Region Labeling Approach" *ICASSP '04, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.3, May 2004.

[20] W.Wei, K.N.Ngan, N.Habili, "Multiple Feature Clustering Algorithm for Automatic Video Object Segmentation" *ICASSP '04, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.3, May 2004.

[21] H.Xu, A.A.Younis, M.R.Kabuka, "Automatic Moving Object Extraction for Content-Based Applications" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.14, No.6, pp.796-812, June 2004.

[22] E. P. Ong, B. J. Tye, W. S. Lin, M. Etoh, "An Efficient Video Object Segmentation Scheme" *ICASSP '02, IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol.4, pp.3361-3364, May 2002.

[23] T.Meier, K.N.Ngan, "Video Segmentation for Content-Based Coding" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.9, No.8, pp.1190-1203, Dec.1999.

[24] C.Kim, J.-N.Hwang, "Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.12, No.2, pp.122-129, Feb.2002.

[25] S.-Y.Chien, S.-Y.Ma, L.-G.Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registraction Technique" *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.12, No.7, pp.577-586, July 2002.

[26] M. Roland, W. Michael, "A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera", *Signal Processing,* Vol.66, pp.203-217, 1998.

[27] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora, "Image Sequence Analysis for Emerging Interactive Multimedia Services - The European COST 211 Framework", *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 8, No. 7, November 1998, pp. 802-813.

[28] A. Amer, " Memory-based spatio-temporal real-time object segmentation", *Proc. SPIE Int. Symposium on Electronic Imaging, Conf. on Real-Time Imaging (RTI),* Santa Clara, USA, vol. 5012, Jan. 2003.

[29] U. Braga-Neto, M. Choudhary, J. Goutsias, "Automatic Target Detection and Tracking in Forward-looking Infrared Image Sequences Using Morphological Connected Operators" *Journal of Electronic Imaging,* 13 (4), pp.802-813, October 2004.

[30] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms" *IEEE Tran. Image Processing,* Vol.2, No.2, April 1993.

[31] C.G.Harris, M.Stephens, "A Combined Corner and Edge Detector" *Proc. of Alvey Vision Conference,* 1988.

[32] S.M.Smith, J.M.Brady, "SUSAN-A New Approach to Low Level Image Processing" *International Journal of Computer Vision,* Vol.23 (1), pp.45-78, 1997.

[33] J. Canny, "A computational approach to edge detection" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.8(6), pp.679-698, 1986.

[34] T. Aach, A. Kaup, R. Mester, "Statistical model-based change detection in moving video" *Signal Processing,* 31 (2), pp. 165-180, March, 1993.

[35] J. Stauder, R. Mech, J. Ostermann, "Detection of moving cast shadows of moving objects in video sequences" *IEEE Transactions on Multimedia,* vol. 1, no. 1, pp65-76, March 1999.

[36] P. Perona, J. Malik, "Scale-space and edge detection using anisotropic diffusion" *IEEE transactions on Pattern Analysis and Machine Intelligence,* Vol.12, No.7, July 1990.

[37] O.J.Morris, M. de J. Lee, A.G. Constantinides, "Graph theory for image analysis: an approach based on the shortest spanning tree" *IEE Preceedings,* Vol. 133, Pt. F, No.2, April 1986.

[38] C. Erdem, B. Sankur, and A. Tekalp "Performance Measures for Video Object Segmentation and Tracking" *IEEE Transactions on Image Processing,* Vol.13, No.7, July 2004.

[39] Y.Wang, J.Ostermann, Y.Q.Zhang, "Video Processing and Communications" Prentice Hall, 2001.

[40] Al Bovik, "Handbook of Image and Video Processing" Academic, 2000.

[41] A.M.Tekalp, "Digital Video Processing" Prentice Hall, 1995.

[42] John C. Russ, "The image processing handbook" 4th edition, CRC Press, 2002.

[43] E.R.Davies, "Machine Vision: Theory, Algorithms, Practicalities" 2nd editioin, Academic, 1997.

[44] W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery, "Numerical Recipes in C" Cambridge University Press, 1992.

[45] P.J.Huber, "Robust Statistics" New York: Wiley, 1981.

Appendix A

Objective Performance Measure for VOS [38]

The performances of an object segmentation algorithm are mainly evaluated subjectively. Few objective measures exist and no standard objective measure is available. In the last years, objective measures to VOS are an active filed of research in video processing. To evaluate the object segmentation using the proposed GME method, the objective measure in [38] was used. In this section, details of this method are given. As stated in [38], there are three statistical features calculated independently for performance measure: spatial color contrast along object boundary, temporal color histogram difference, and motion difference along object boundary.

- **Spatial color contrast along object boundary**
  Assuming that object boundaries are coincided with color boundaries, if the objects are successfully segmented, there should be a color difference between the pixels that along the segmented object boundaries. To measure this difference, first, a set of probe pixels are established by drawing normal lines of length L astride the segmented object boundaries at equal intervals (see Fig. A). Then the color probes are defined as $M \times M$ regions centered at the two ends P_o^i and P_I^i of each normal line (see Fig. A). The measure of color difference is calculated as follows:

$$0 \leq d_{color} = 1 - \frac{1}{K_t} \sum_{i=1}^{k_t} \delta_{color}(i) \leq 1 \tag{A.1}$$

$$\delta_{color}(i) = \frac{||C_o^i - C_I^i||}{\sqrt{3 \times 255^2}} \tag{A.2}$$

  where K_t is the total number of normal lines, C_o^i and C_I^i denote the average color calculated in the $M \times M$ region of each pair of the probe pixels P_o^i and P_I^i. $L = M = 3$ in [38].
  The worst score of d_{color} is 1 and it decreases as the color difference along object boundaries increases.

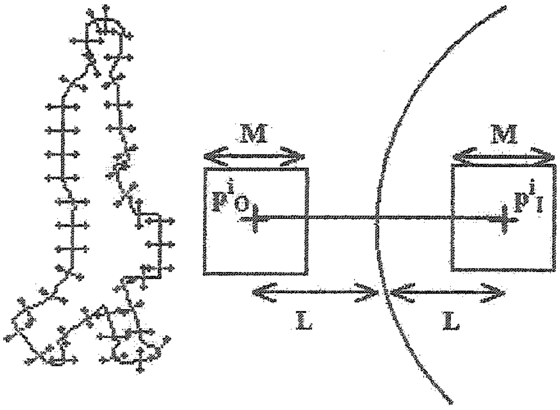


Figure A.1: Probe pixels along the object boundaries [38].

- **Temporal color histogram difference**
  Assuming that the color histogram of the objects is temporally stationary, and this histogram is

different from the color histogram of the background, the difference between this histogram H_t at time t and H_{ref} at a reference time should represent the stabilization of the segmentation. A χ^2 metric is used to calculate this difference as follows:

$$0 \leq d_{hist} = \frac{\sum_{j=1}^{B} \frac{[r_1 H_t(j) - r_2 H_{ref}(j)]^2}{H_t(j) + H_{ref}(j)}}{N_{H_t} + N_{H_{ref}}} \leq 1 \tag{A.3}$$

$$r_1 = \sqrt{\frac{N_{H_{ref}}}{N_{H_t}}}, \qquad r_2 = \frac{1}{r_1} \tag{A.4}$$

$$N_{H_t} = \sum_{j=1}^{B} H_t(j), \qquad N_{H_{ref}} = \sum_{j=1}^{B} H_{ref}(j) \tag{A.5}$$

where B indicates the number of bins in the histogram.

The best score of d_{hist} is 0 when there is no histogram difference between the segmented objects at time t and the one at a reference time, and it increases as the histogram difference increases.

- **Motion difference along object boundary**
  Assuming that the object boundaries are coincided with motion boundaries, if the objects are successfully segmented, there should be a motion difference between the pixels along the segmented object boundaries. To measure this motion difference, the same geometry of the probes used for measuring the color difference is adopted (see Fig. A). However, the parameters are adjusted to $L = 5$ and $M = 3$ in [38]. The measure of motion difference is calculated as follows:

$$0 \leq d_{motion} = 1 - \frac{\sum_{i=1}^{K_t} \delta_{motion}(i)}{\sum_{i=1}^{K_t} w_i} \leq 1 \tag{A.6}$$

$$\delta_{motion}(i) = d(v_o^i, v_I^i) \cdot w_i \tag{A.7}$$

$$0 \leq w_i = R(v_o^i) \cdot R(v_I^i) \leq 1 \tag{A.8}$$

where v_o^i, v_I^i denote the average motion vectors calculated in the $M \times M$ region of each pair of the probe pixels P_o^i and P_I^i. $d(v_o^i, v_I^i)$ is the distance between these two average motion vectors, which is defined as:

$$0 \leq d(v_o^i, v_I^i) = 1 - exp(-\frac{||v_o^i - v_I^i||}{\sigma^2}) \leq 1 \tag{A.9}$$

and σ is set 1 in [38]. $R(.)$ denotes the reliability of the motion vector v^i, and is defined as:

$$R(v^i) = exp(-\frac{||v^i - b^i||^2}{2\sigma_m^2}) \cdot exp(-\frac{||c(p^i) - c(p^i + v^i)||^2}{2\sigma_c^2}) \tag{A.10}$$

where b^i denotes the backward motion vector at location $p^i + v^i$ in frame $t+1$; $c(.)$ denotes the color intensity, and the parameters σ_m and σ_c are chosen as 5 and 10 in [38].

The worst score of d_{motion} is 1 and it decreases as the motion difference along object boundaries increases.

- **Combined measurement** Furthermore, these features can also be combined to give an overall score as in Eq. A.11. The parameters μ_1, μ_2, and μ_3 can set adaptively as far as the sum $\mu_1 + \mu_2 + \mu_3$ is restricted to be one. The lower the score d, the better the performance of the VOS.

$$d = \mu_1 d_{color} + \mu_2 d_{hist} + \mu_3 d_{motion} \tag{A.11}$$

Appendix B

Publication

Based on the proposed GME method presented in Chapter 2 the following paper was submitted and accepted for publication at a prestigious IEEE conference: Bin Qi and Aishy Amer, "Robust and Fast Global Motion Estimation Oriented to Video Object Segmentation", IEEE International Conference on Image Processing (ICIP), Genoa, Italy, 11-14 September 2005.

Appendix C

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
HVS	Human Visual System
VOP	Video Object Plane
VOS	Video Object Segmentation
GM	Global Motion
GME	Global Motion Estimation
GMC	Global Motion Compensation
VM	Verification Model
AM	Analysis Model
LME	Local Motion Estimation
LMC	Local Motion Compensation
SAD	Sum Absolute Difference
SSD	Sum Square Difference
MV	Motion Vector
BMA	Block Matching Algorithm
SVD	Singular Value Decomposition
PSNR	Peak Signal to Noise Ratio
MSE	Mean Square Error
MAE	Mean Absolute Error
CMAPE	Corner Mean Absolute Error
CDM	Change Detection Mask
RSST	Recursive Shortest Spanning Tree
CIF	Common Intermediate Format
QCIF	Quarter Common Intermediate Format
SIF	Source Input Format
PAL	Phase Alternate Line
MPEG	Moving Picture Experts Group
DFD	Displaced Frame Difference
CD	Change Detection

List of Figures

List of Tables