الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Nº Réf :……

Centre Universitaire

Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie **Département de Mathématiques et Informatique**

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité : Sciences et Technologies de l'Information et de la Communication (STIC)

Thème
Video segmentation and its applications

Préparé par : Koko Sarra
Seddiki Hassina

Devant le jury :

Dib. Abderrahim	**(MAA)**	**C.U.Abd Elhafid Boussouf**	**Président**
Yassaadi. Sabrina	**(MAA)**	**C.U.Abd Elhafid Boussouf**	**Examinateur**
Boulmerka. Aissa	**(MCB)**	**C.U.Abd Elhafid Boussouf**	**Rapporteur**

Année Universitaire : 2017/2018

Acknowledgements

Praise be to Allah for helping us, enlighten the way to complete our work and study.
Our thanks to our very dear parents, brothers, sisters, colleagues and friends who have encouraged us, supported throughout our journey. A special thanks to our supervisor Dr A. Boulmerka for his presence, his help and especially for his valuable advice that helped us to fulfill our Thesis.
We would like to express our sincere thanks to everyone at the science and technology institute, especially the teachers who have taught us during all our years of study.
Finally, we thank all those who contributed directly or indirectly to the completion of this work.

Abstract

We often hear the old adage "a picture is worth a thousand words", where if you want to convey complex semantic information you can use just an image and it will fulfill the purpose, let alone for the video that contains more meaningful information than its predecessor.

where the interest of this work focuses on video segmentation and all about it which is a low-level computer vision problem that can be the primary step of a wide range of higher-level tasks such as object tracking, activity recognition...etc. where the main goal of video segmentation is separate moving object from the background with an acceptable and meaningful way depending on several methods and features.

This dissertation contains three chapters, the first chapter are purely theoretical, including a presentation of video Segmentation With its characteristics, the second chapter introduces an improvement of a recent segmentation approach that uses the concept of objects proposals, the final chapter

includes an evaluation of our proposed method compared to other methods that used in video segmentation to prove its effectiveness.

key words: *superpixels, optical flow, proposals, gradient...etc.*

Résumé

Nous entendons souvent le vieil adage "une image vaut mille mots", ou si vous voulez transmettre une information sémantique complexe, vous pouvez utiliser juste une image et elle remplira le but, sans parler de la vidéo qui contient des informations plus significatives que son prédécesseur.

Où l'intérêt de ce travail se concentre sur la segmentation vidéo et tout ce qui est un problème de vision informatique de bas niveau qui peut être l'étape primaire d'un large éventail de tâches de niveau supérieur telles que le suivi des objets, la reconnaissance des activités.

L'objectif principal de la segmentation vidéo est de séparer les objets en mouvement de l'arrière-plan d'une manière acceptable et significative en fonction de plusieurs méthodes et fonctionnalités.

Cette thèse contient trois chapitres, le premier chapitre est purement théorique, y compris une présentation de la vidéo Segmentation Avec ses caractéristiques, le deuxième chapitre introduit une amélioration d'une approche de segmentation

récente qui utilise le concept de propositions d'objets, le dernier chapitre fournit une comparaison entre un ensemble de méthodes utilisées dans la segmentation vidéo et notre méthode afin de prouver leur efficacité.

Mots-clés : *superpixels, flux optique, proposals, gradient ... etc.*

ملخص

غالبا ما نسمع القول المأثور: " ان الصورة تساوي ألف كلمة" حيث إذا كنت تريد نقل معلومات دلالية معقدة يمكنك استخدام مجرد صورة وستحقق الهدف ناهيك عن الفيديو الذي يحتوي على معلومات أكثر فائدة من سابقتها.

حيث يتركز اهتمام هذا العمل على تجزئة الفيديو وكل ما يتعلق به فهو يعتبر مشكلة منخفضة المستوي في مجال الرؤية بالكمبيوتر والتي يمكن ان تكون الخطوة الأساسية لمجموعة كبيرة من المهام ذات المستوى الأعلى مثل تتبع الكائنات التعرف على النشاط ... الخ.

الهدف الرئيسي من تقسيم الفيديو هو فصل الكائن المتحرك عن الخلفية بطريقة مقبولة وذات مغزى اعتمادا على العديد من الطرق والميزات.

تحتوي هذه الأطروحة على ثلاثة فصول، الفصل الأول نظري بحت حيث يعرف بتقسيم الفيديو و خصائصه ، الفصل الثاني يقدم تحسينًا في طريقة تجزئة حديثة تستخدم مفهوم مقترح الكائنات ، اما الفصل الأخير فيعرض مقارنة بين مجموعة من الطرق المستخدمة في تجزئة الفيديو وطريقتنا من أجل إثبات فعاليتها.

Contents

List of Figures

List of Tables

1

Introduction

The human brain can achieve the remarkable feat of processing an image seen for just 13 milliseconds where a simple person can simply and successfully identified things in the environment around it in an interesting time. In addition to its simplicity, the *human vision system* is a complex yet powerful process and requires finite elements and organization. Besides, any imbalance in only one element will affect the whole visual system.

Among the researchers who highlighted the computer vision field, there are computer scientists, where they tried to embody this finite precision system and make the machine simulate the human being.

Nowadays, there is an advanced in research in computer vision fields, where it's exist in various fields such as medical, industry, surveillance, entertainment and until we find it in games and advertising. So computer vision has taken a great place in human life and has facilitated many things that were difficult in the past. The main benefits of such applications in the computer vision field are gaining time which means gaining money. To this end, one of the crucial tasks is segmentation which is the operation of separate things and objects in an image and/or a video with an acceptable and meaningful way.

There is a fast outbreak of segmentation algorithms over the last years, which proves that segmentation algorithms have fortified its position in the world of digital vision where this indicates to the great benefit that it gave to humans.

For video segmentation, there are two main classes:

supervised approaches that where you have input variable (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

unsupervised approaches where you only have input data (X for example an image)and no corresponding output variables, and those classes are used to discover patterns in a video that leads to actionable insights. Note that in our dissertation we are interested only in unsupervised approaches which is the most interesting one.

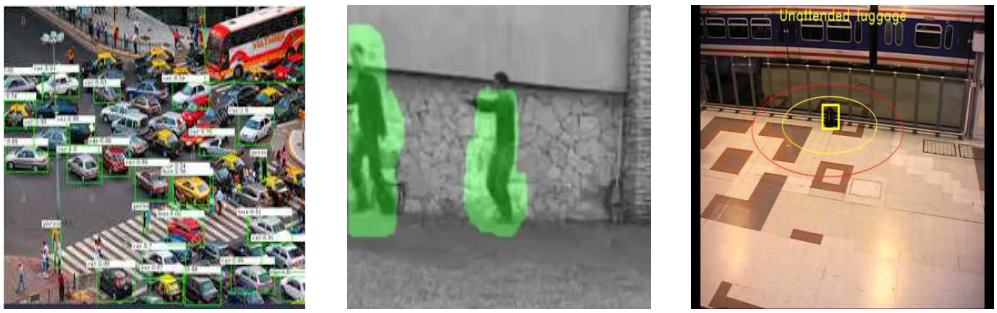
1.1 Applications of video segmentation

The applications of video segmentation in the field of automatic vision processing are interesting, where there are several. among them, we find the following:

1.1.1 Object Based Surveillance Analysis

Surveillance allows identifying any abnormal activity in a given environment thereby enhancing public safety and reducing crime. If there are more number of smart cameras used in surveillance process, then there is a risk that person who monitors may not be able to analyse the videos effectively [19]. .

So for that, many algorithms are proposed to solve this problem and make this task easy and automatic based on the concept of video segmentation to detect cars in roads, detect unattended bags, uncover suspicious activity and abnormal behavior [19].



(a) cars detection (b) abnormal behavior (c) unattended luggages

FIGURE 1.1: Object-based surveillance analysis applications [42], [41], [45].

1.1.2 Tourism

Depending on the tourist spot, system presents the cultural or heritage story. where system displays the user movement along with narration. Actually, using segmentation the virtual tourist guide can be associated along with the narration as is shown in Figure.1.2. This is one of the interesting applications adding real world experience through mobile devices [19].

FIGURE 1.2: An illustration of a tourism application of video segmentation [36].

1.1.3 Interactive computer games

Computer games are a popular consumer electronics item, the game players find it captivating to interact with games and they may find it even more engaging to interact through natural, unencumbered hand or body motions [32]. The structure of games provides a context which can allow dramatic, appropriate responses from simple visual measurements (Figure. 1.3). The vision system may track the position of the visual center of mass of the player to detect his different movements to make a balance between the player and the game [32].

FIGURE 1.3: Interactive computer games [46], [47].

1.2 Outline of the dissertation

In this dissertation, we followed the main structure composed of a general introduction, three chapters, and a conclusion and perspective work. The first chapter represents state of the art about video segmentation and most concepts to be used

later. The second chapter represents the proposed approach which is based on spatiotemporal object proposals. The third chapter represents the experimentations and the evaluation of the proposed approach along with some compared state-of-the-art methods. Finally, the present dissertation finished with a conclusion that summaries all the chapters of the dissertation and presents our perspectives and future interests.

2

State of the art and related works

2.1 Introduction

In this Chapter, we provide an overview of video segmentation. Firstly, we present the definition and challenges of video segmentation as well as some features which are effective in the process of segmentation either in images or in videos. Then, we describe some refinement methods that can be used to improve segmentation results. Next, we present some successful algorithms in video segmentation. After that, we mention some evaluation metrics that are used in the third chapter of this

dissertation to evaluate segmentation methods and applications of video segmentation. Finally, we provide three of the most used datasets in order to evaluate video segmentation algorithms.

2.2 Video segmentation

2.2.1 Definitions

An image can be a reflection of a perceptible reality. From a mathematical point of view, an image is a matrix of numbers that represent semantic information. We can define an image as a function $z = f(x, y)$. Which, at each point of the plane (x, y), associates a value z and that is illustrated on Figure. 2.1.

FIGURE 2.1: Mathematical representation of an image[48].

Image segmentation is the process of partitioning an image into non-overlapped, consistent regions that are uniform with respect to some characteristics like intensity, color, tone or texture, to name a few.

A video is composed of a sequence of images. and as it's shown on Figure. 2.2.Different from still image segmentation, video segmentation should take into account the temporal information [19].

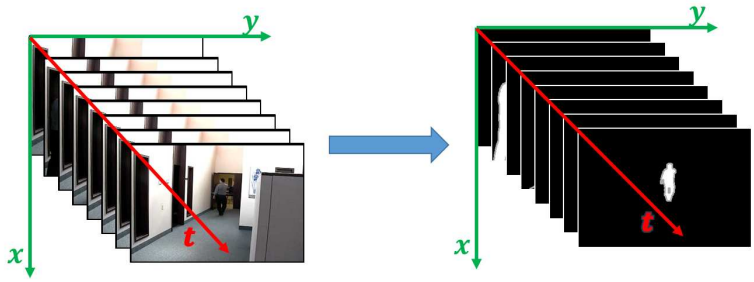


FIGURE 2.2: Representation of time factor in video segmentation.

Video segmentation is a way of dividing a movie into meaningful segments for visual information extraction to use them in different applications. The Figure. 2.3 represents an example of video segmentation process.

FIGURE 2.3: Illustration of video segmentation results[33].

2.2.2 Video segmentation challenges

Several challenges may affect and obstruct video segmentation. In the following, we mention a non-exhaustive list of these challenges:

- **Cast shadows:**

  tend to be classified as parts of the foreground. Indeed, they may generate patterns of movement.

- **Dynamic backgrounds:**

can be caused by moving background , such as swaying tree leafs, water flowing, etc.

- **Noisy videos:**

  noise can be generated by several sources such as sensor noise, low-quality cameras or compression artifacts. Indeed, noisy videos tend to produce numerous false detections.

- **Camouflage effects:**

  occur when a moving object or some of its parts are made of colors similar to the background. They may cause false negatives for foreground detection.

- **Camera jitter:**

  can be caused by camera instability (e.g., by wind or vibrations).

2.3 Segmentation features

The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Selecting the right features (such as color, optical-flow ...etc) plays a critical role in image and video segmentation.

2.3.1 Color

An image is created by sampling the incoming light. The colors of the incoming light depending on the color of the light source illuminating the scene and the material that the object is made of [18]. Going back some years, many cameras (and displays, e.g., TV-monitors) only handled gray-scale images. As the technology matured, it became possible to capture color images, and today most cameras

capture color images [18].

2.3.1.1 Color spaces

Color is one of the fundamental properties of objects that has been used as important cue in several object segmentation and there are many different color spaces each with its own properties and advantages, in this section, we choose two of the most used ones, namely **RGB** and **CIE Lab** [34].

RGB: Color is considered as one of the fundamental properties of objects, that has been used as an important cue in several object segmentation works where each pixel color in the image is described by the primary colors (red, green and blue) [18] where each pixels is represented as formula 2.1 and this is illustrated by the Figure. 2.4.

$$Colorpixel = [Red, Green, Blue] = [R, G, B] \tag{2.1}$$

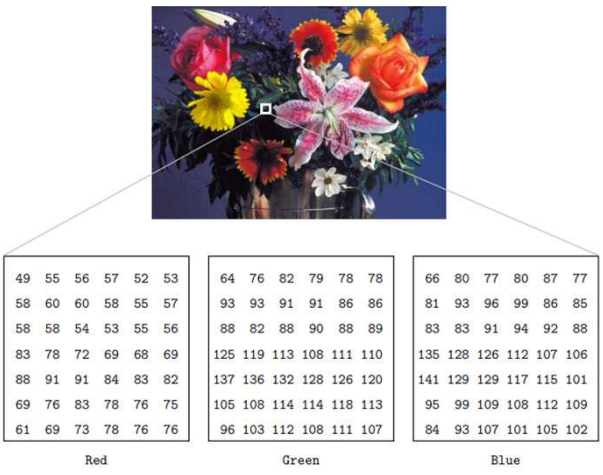


FIGURE 2.4: Example of an image and its RGB color space values [18].

CIE Lab: this color space is defined by the International Commission on Illumination (CIE), it expresses color as three numerical values L* represents the lightness of color going from 0 (dark) to 100 (white), while the a* and b* channels are the two chromatic components. The first of these two (a*) represents the colors position between red/magenta (+a) and green (-a). Similarly, b* indicates its position between yellow (+b) and blue (-b). In practice, their range goes from -128 to 127 with 256 levels [34].

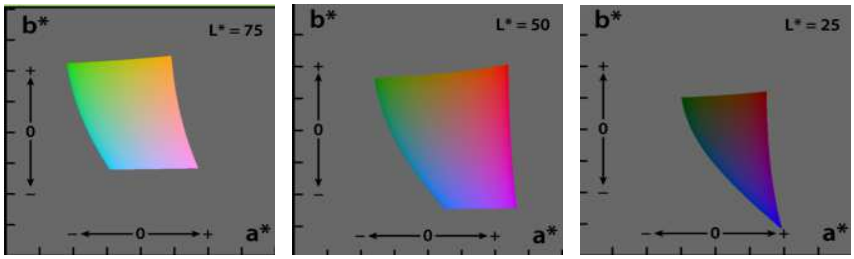


FIGURE 2.5: Illustration of the CIE LAB color space [44].

2.3.2 Edges

Edges are useful in many applications since they define the contour of an object(see Figure.2.6(d)). It is therefore of great importance to have a clear definition of where an object starts and ends [18]. An edge in an image is defined as a position where a significant change between regions of different color, intensity, or texture.

To enable edge detection, we use the concept of the gradient; we can define the gradient as the difference between the previous and next value. For each point in the image, we have two gradients: one in the x-direction and one in the y-direction. The resulting gradient is defined as a vector $G(g_x, g_y)$(in Eq.2.2), where g_x is the gradient in the x-direction and g_y is the gradient in the y-direction.

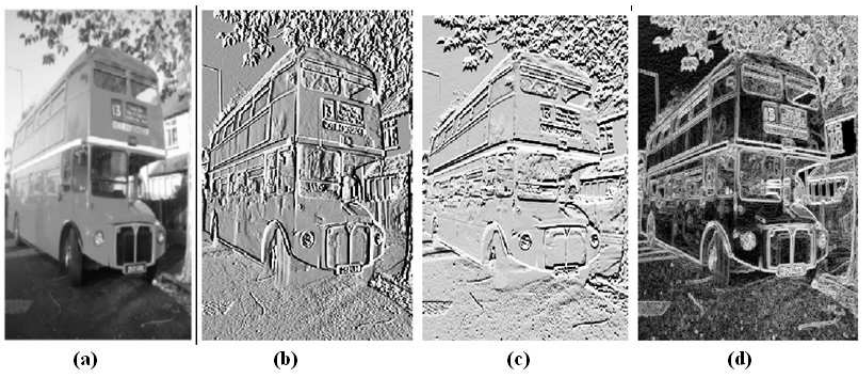


FIGURE 2.6: Illustration of image gradient[43]. (a) gray-level image, (b) the first-order partial derivatives in the x-direction and (c) y-direction, and (d) the length of the gradient.

$$g_x(x,y) = f(x+1,y) - f(x-1,y) \tag{2.2}$$

$$g_y(x,y) = f(x,y+1) - f(x,y-1) \tag{2.3}$$

The approach used to extract the edges is to make thresholding for the amplitude of the gradient, knowing that the amplitude represents the length of the gradient vector and calculated as:

$$Magnitude(x,y) = |g_x(x,y)| + |g_y(x,y)| \tag{2.4}$$

The orientation of the gradient vector is another feature that can be extracted from the gradient information. The orientation of the gradient represents the angle of change in intensity between pixels shown by the following formula:

$$Orientation(x, y) = arctan\frac{g_x(x, y)}{g_y(x, y)} \tag{2.5}$$

As shown in the images of Figure. 2.6, the x-derivative seems to emphasise vertical edges while the y-derivative seems to emphasise horizontal edges. Unlike to the gradient that it contains information about both derivatives and therefore emphasizes edges in all directions.

2.3.3 Optical-flow

Motion detection is a hot study field of computer vision. Its purpose is to extract moving object area in image sequences. Extracting moving object effectively and exactly is the foundation of tracking and sorting of moving target in computer vision. Up to the present, there are several proposals; one of them is optical-flow.

Optical-flow is the displacement field for each of the pixels in an image sequence. For every pixel, a velocity vector $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ is found which says:

- How quickly a pixel is moving across the image.
- The direction of its movement.

Usually, video segmentation algorithms begin by computing optical-flow between pairs of subsequent frames $(t, t + 1)$ using the state-of-the-art algorithms [4, 26]. Figure. 2.7) shows an illustration of the optical-flow feature. In this Figure, the optical-flow in Figure 2.7)-(c) is obtained by the application of the method in [4] on the two input images given in Figures 2.7)-(a) and 2.7)-(b), respectively. Note that the implementation by [26] supports large displacements between frames and has a computationally very efficient GPU implementation.

(a) frame I_t (b) frame I_{t+1} (c) optical-flow

FIGURE 2.7: Illustration of the optical-flow feature. (a)-(b) input frames. (c) the resulting optical flow using the method in [4].

2.3.4 Superpixels

A superpixel is a set of pixels in image that have common properties(the same color for instance), which can be used to replace the rigid structure of the pixel grid and the major advantage of superpixels is to reduce the input and the quantity of data which are used in algorithms for example and that minimize the runtime which is an important and sensitive point [1]. There are a set of methods and algorithms that are used to segment images of pixels into images of superpixels, one of them is the SLIC method. Figure 2.8 illustrates the results of superpixels oversegmentation by the application of the SLIC method.

FIGURE 2.8: Images over-segmented using SLIC approach into superpixels of size 64, 256, and 1024 pixels [1].

2.3.5 Object proposals

Object-level features namely *object proposals* are used to bridge the gap between low-level features and object-level object detection [10]. For an input image, object proposal methods generate a set of object candidates which are likely to include the object of interest. Thus, those object candidates can cover the entire objects in the image with excellent accuracy [8].

Let $I = \{I_1, I_2, ...\}$ be the set of input frames. For the t-th frame I_t, static region-ranking method generate object proposal segmentations $P_t = \{p_t^1, p_t^2, ...\}$ (see Figure. 2.9 for an illustrative example). The proposals are generated via method in [8] where the goal of this method is to propose candidates for any object in an image based on estimated boundaries, geometry, color, texture and part of learning, each stage of this process must encourage diversity among the proposals while minimizing the number of candidates to consider.

Each frame would have hundreds of object candidates. Different from super-pixel regions, these proposal segments are more "object-like" as they have more distinct occlusion boundaries and their appearances are in obvious contrast with nearby pixels [10].

2.4 Refinement methods

When dealing with video segmentation, there exists a variety of methods that are used to improve the quality of the segmentation and give efficiency to the algorithms. In this section, we give the principal ones such as the GMM models (Gaussian mixture models), the MRF (Markov Random Field), to cite a few.

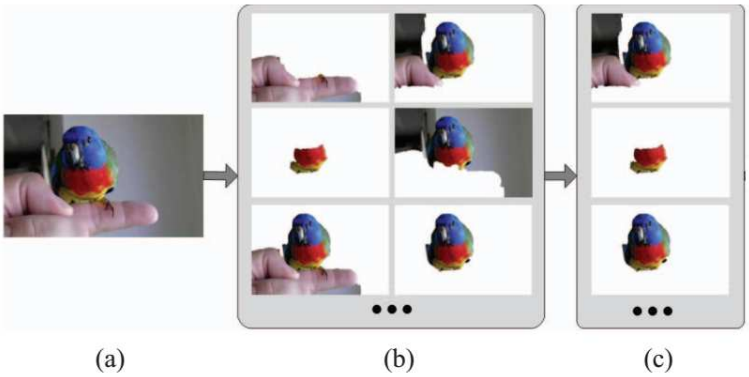


FIGURE 2.9: Illustration of an image and their object proposals generated using the method in [8]. (a) Input frame. (b) Object proposals from the frame in (a). (c) A set of proposal candidates selected via a ranking strategy.

2.4.1 Markov random field model (MRF)

Markov random field models are undirected probabilistic graphical models which are a wide-spread model in computer vision. The unifying ideas in using MRFs for vision are the following [2]:

- Images are dissected into an assembly of nodes that may correspond to pixels or superpixels.
- Hidden variables associated with the nodes are introduced into a model designed to "explain" the values (colors) of all the pixels.
- A joint probabilistic model is built over the pixel values and the hidden variables(for example a graph).

The motivation for constructing such a graph is to connect the hidden variables associated with the nodes. For example, for the task of segmenting an image into foreground and background. Each node x (pixel or superpixel) has an associated random variable y that may take the value of 0 or 1, corresponding to the foreground or background [2]. The Figure.2.10 demonstrates a graph that has been

constructed using MRF model.

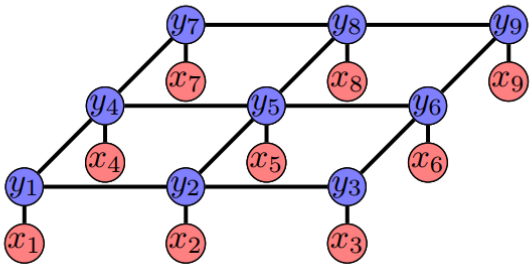


FIGURE 2.10: Representation of the graph constructed by the Markov random field (MRF) model [2].

2.4.2 Gaussian mixture model (GMM)

Gaussian functions are suitable for describing many processes in mathematics, science, and engineering, making them very useful in the fields of image and video processing. For example, an image can be simply modeled with the Gaussian distribution according to the central limit theorem from the probability theory [11] and calculated in Eq. (2.6).

$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp \frac{-(x-\mu)^2}{2\sigma^2} \tag{2.6}$$

where

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.7}$$

and

$$\sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{2.8}$$

with n is the whole number of the pixels in the image, μ is the mean and σ is the variance.

Gaussian mixture model (GMM) is a probabilistic model for representing normally distributed subpopulations within an overall population. Techniques based on GMM are applied to many different tasks such as are speech recognition, image segmentation, to cite a few see [23]. In the case of image segmentation, GMM is used to represent the image units (as in Eq.(2.9), For example, in foreground image segmentation, colors of objects can be represented with a GMM model [23].

$$p(x) = \sum \alpha_i g(x|\mu_i, \sigma_i) \tag{2.9}$$

where each α_i is the mixting probability and each μ_i and σ_i are parameters that defining the i-th component of the GMM model. As being probabilities, the α_i must satisfy $\alpha_i > 0, i = 1...K$ and $\sum_{i=1}^{K} \alpha_i = 1$. For example, Figure 2.11 presents a Gaussian mixture model with two components.

2.4.3 Graph cut methods (GC)

An undirected graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ is defined as a set of nodes (vertices) $\mathbf{V}$ and a set of undirected edges $\mathbf{E}$ that connect these nodes. A cut on a graph is a partition of $\mathbf{V}$ into two subsets $\mathbf{A}$ and $\mathbf{B}$ where

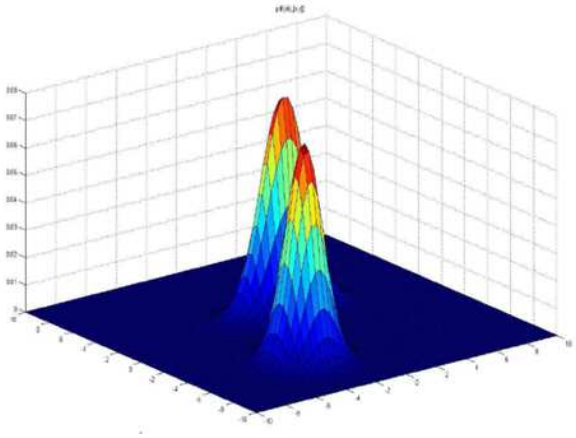


FIGURE 2.11: A Gaussian mixture model with two components [23].

$$\mathbf{A} \cup \mathbf{B} = \mathbf{V}, \mathbf{A} \cap \mathbf{B} = \emptyset \tag{2.10}$$

Perhaps the simplest and best-known graph cut method is the **min-cut formulation**. The min-cut of a graph is the cut that partitions $\mathbf{G}$ into disjoints segments such that the sum of the weights associated with edges between the different segments are minimized. That is the partition that minimizes $C_{min}(\mathbf{A}, \mathbf{B})$ as the following:

$$C_{min}(\mathbf{A}, \mathbf{B}) = \sum_{u \in \mathbf{A}, v \in \mathbf{B}} \mathcal{W}_{uv} \tag{2.11}$$

Basically, each pixel in the image is viewed as a node in a graph and links ($\mathbf{E}$) are formed between nodes with weights corresponding(as is shown in Figure.2.12), each pixel has links with all its neighbors and one with background node and another one with foreground node [3]. The link weight between pixel i and pixel

j will be denoted $\mathcal{W}_{ij}^{I}$and the terminal weights between pixel i and the foreground node (s) and background node (t) as $\mathcal{W}_i^s$ and $\mathcal{W}_i^t$ as is shown in Figure.2.11.

The schema in Figure. 2.12 represents a construction of an undirected graph from an image and the cut that separate the nodes belong to the object from the nodes that belong to the background [3].

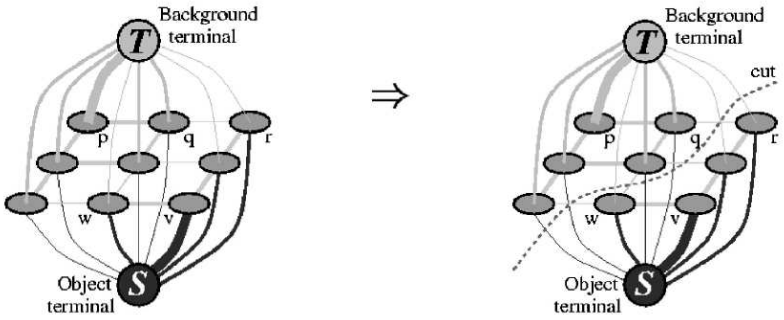


FIGURE 2.12: Image representation as a graph with the cut results [3].

2.5 Unsupervised video segmentation

In video object segmentation, the task is to separate out foreground objects from the background across all frames. To this end, video segmentation approaches can be classified into two main categories, namely, *unsupervised approaches* vs. *supervised approaches*.

In this work, we are interested in the *unsupervised approaches*, where the main assumption is that there is no human involvement on the video. The goal of unsupervised approaches is to model the underlying structure, distribution or features of the data to learn more about the data.

In addition to the *appearance* information which also drives image segmentation, video data provide a rich and complementary source of information in form

of *object motion*. It is natural to expect that both *appearance* and *motion* should play a key role in successfully segmenting objects in videos [15].

2.5.1 Appearance models

A number of *appearance priors* (spatial information) have been proposed for object video segmentation, and the most widely used ones are *contrast saliency cue* and *background prior score*. In this section, we present some appearance models.

2.5.1.1 Contrast saliency cue

Many works [7, 10, 31] use the region contrast against its surrounding scales as a saliency cue, which is computed as the summation of its appearance differences from all other regions and weighted by their spatial distances. In this way, the contrast saliency cue for superpixel r_t^n in frame I_t can be written as

$$F_{cnt}(R_t^n) = \sum_{m=1}^{|R_t|} \phi(R_t^n, R_t^m) \|c_t^n - c_t^m\|_2 \tag{2.12}$$

where c_t^n and c_t^m are colors of regions R_t^n and R_t^m, respectively.

$\phi(R_t^n, R_t^m) = exp(D(R_t^n, R_t^m)/\sigma^2)$ controls the spatial influence between two regions R_t^n and R_t^m. $D(R_t^n, R_t^m)$ is a square of Euclidean distance between region centers of R_t^n and R_t^m.

2.5.1.2 Background prior

The background prior called *boundary connectivity* [31] is a measure to quantify how heavily a region R_t^n is connected to the boundaries of frame I_t. It is defined as

$$BndCon(R_t^n) = \frac{|p|p \in R_t^n, p \in Bnd(I_t)|}{\sqrt{|\{p|p \in R_t^n\}|}} \tag{2.13}$$

where $Bnd(I_t)$ is the set of image boundary patches and p is an image patch. It has an intuitive geometrical interpretation; it is the ratio of a region's perimeter on the boundary to the region's overall perimeter, or the square root of its area.

The background prior $w_{bg}(R_t^n)$ is mapped from the boundary connectivity value of the region r_t^n. It is close to 1 when boundary connectivity is large, and close to 0 when it is small, it is defined by 2.15

$$w_{bg}(R_t^n) = 1 - \exp\left(-\frac{F_{bg}(R_t^n)}{2\sigma_{bg}^2}\right) \tag{2.14}$$

where σ is a parameter such that $\sigma_{bg} \in [0.5, 2.5]$.

2.5.1.3 Background weighted contrast

The contrast saliency cue in Eq. (2.12) can be extended by introducing the background prior $w_{bg}(R_t^n)$ as a new weighting term. The enhanced contrast, called *background weighted contrast*, is defined as:

$$F_{wcnt}(R_t^n) = \sum_{m=1}^{|R_t|} \phi(R_t^n, R_t^m) \|c_t^n - c_t^m\|_2 w_{bg}(R_t^n) \tag{2.15}$$

2.5.1.4 Objectness score

Objectness score is a measure that quantifies how likely it is for a region to be a part of the foreground, otherwise, the objectness score of a region tells us how

likely it is to contain an object.

2.5.2 Motion models

When dealing with video sequences, *motion* (temporal information) provides a powerful feature for unsupervised video segmentation in addition to appearance. That is because segment objects that have a different motion pattern (i.e., the objects that move differently than their surroundings) often attract more attention [10, 20]. In this section, we give some used spatiotemporal features based on motion.

2.5.2.1 Motion boundaries

Motion boundaries (i.e., image points where the optical-flow field changes abruptly) reveal the location of occlusion boundaries, which very often correspond to physical object boundaries.

Let $\vec{f_p}$ be the optical-flow vector at pixel p. The simplest way to estimate motion boundaries is by computing the magnitude of the gradient of the optical-flow field(as in Eq.2.16):

$$b_p^m = 1 - exp(-\lambda^m \|\nabla \vec{f_p}\|) \tag{2.16}$$

where $b_p^m \in [0, 1]$ is the strength of the motion boundary at pixel p; λ^m is a parameter controlling the steepness of the function.

2.5.2.2 Difference in direction of motion

While the motion boundaries (b_p^m) correctly detects boundaries at rapidly moving pixels, where b_p^m is close to 1, it is unreliable for pixels with intermediate b_p^m

values around 0.5, which could be explained either as boundaries or errors due to inaccuracies in the optical-flow. Based on the difference in direction between the motion of pixel p and its neighbors $\mathcal{N}$, a second estimator is proposed by [20] to disambiguate between those two cases:

$$b_p^\theta = 1 - exp(-\lambda^\theta \max_{q \in \mathcal{N}}(\delta\theta^2_{\theta_{p,q}})) \tag{2.17}$$

where $\delta\theta^2_{\theta_{p,q}}$ denotes the angle between $\vec{f_p}$ and $\vec{f_q}$. The idea is that if n is moving in a different direction than all its neighbors, it is likely to be a motion boundary.

2.5.2.3 Motion contrast-based saliency

Given the optical-flow vector $o_t = (u, v)$ between two consecutive frames I_t and I_{t+1}, the motion distribution of region r_t^n is encoded by two descriptors: a normalized histogram of the flow magnitude $o^{grad} = grad(\sqrt{u^2 + v^2})$, and the distribution of flow orientation $o^{ori} = arctan(v/u)$. Based on the histogram $Hist_{flow}$ of motion feature $flow = \{o^{grad}, o^{ori}\}$, we compute the motion contrast score M_{bg} for region r_n^t with Eq.**??**:

$$M_{bg}(r_n^t) = 1 - exp\Big(- \chi^2\big(Hist_{flow}(r_n^t), Hist_{flow}(BP)\big)\Big) \tag{2.18}$$

where BP represents the background proposals.

2.5.2.4 Motion Gradient Summation

The motion gradient score $M_{grd}(r_n^t)$ (as in Eq2.19 is computed by making use of the motion gradient summation technique [10, 30]. This score is defined as the

average Frobenius norm of optical-flow gradient in the boundary of object region r.

$$M_{grd}(r_n^t) = \|o_t\|_F = \sqrt{\sum_{i=x,y}\sum_{j=x,y}|(u_i, v_j)|^2} = \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2} \tag{2.19}$$

where $o_t = (u, v)$ is the optical-flow of consecutive frames I_t and I_{t+1}, u_x, and u_y are optical-flow gradients in the x direction and v_x and v_y are those in the y direction.

2.5.2.5 Object Region Consistency

Interframe score for each Region r_n^t in frame I_t is defined based on the salient regions of the previous frame [10]. Specifically, each object region r_n^{t-1} for frame I_{t-1} can be warped to frame I_t according to the forward optical-flow. By estimating the overlap between region r_n^t in frame I_t and the warped object regions, the temporal consistency score M_{cnt} can be given by the following formula:

$$M_{cnt}(r_n^t) = \frac{\hat{r}_n^{t-1} \bigcap r_n^t}{Area(r_n^t)} \tag{2.20}$$

where $\hat{r}_n^{t-1}$ denotes the warped regions of the region r_n^t from frame I_{t-1} to frame I_t according to optical-flow o_t.

2.5.3 Some recent illustrative approaches

To illustrate the overall process of unsupervised video segmentation, we present briefly three recent methods from the state-of-the-art.

2.5.3.1 Fast video segmentation

The goal of the method in [21] is to separate moving objects from the background; this approach has two main steps which are:

A) **Efficient initial foreground estimation :**

In this stage, there is a primary estimation for the object boundaries based on motion boundaries estimation using optical-flow to determine which pixels are inside the object and which ones are outside in video sequence. The result of this step is the *inside-outside maps*.

B) **Foreground-background labelling refinement :**

In the previous stage, there is an approximation estimation of the object which can be false or not correct due to wrong optical-flow estimation. For this reason that this stage has come to refine the spatial accuracy of the inside-outside maps and to segment the whole object in all frames.

Each superpixel $s_t^i \in S_t$ can take a label $l_i^t \in \{0, 1\}$. A labelling $\mathcal{L} = \{l_i^t\}_{t,i}$ of all superpixels in all frames represents a segmentation of the video. To evaluate a labeling, an energy function is defined as:

$$E(\mathcal{L}) = \sum_{i,t} A_i^t(l_i^t) + \alpha_1 \sum_{i,t} L_i^t(l_i^t) + \alpha_2 \sum_{(i,j,t)\in\mathcal{E}_s} V_{i,j}^t(l_i^t, l_j^t) + \alpha_3 \sum_{(i,j,t)\in\mathcal{E}_t} W_{i,j}^t(l_i^t, l_j^{t+1}) \tag{2.21}$$

where A_t is a unary potential evaluating how likely a superpixel is to be foreground or background according to the appearance model of frame I_t. The second unary potential L_t is based on a location prior model encouraging foreground labelings in areas where independent motion has been observed.

Both the appearance model and the location prior parameters are derived from the inside-outside maps. The pairwise potentials V and W encourage spatial and temporal smoothness, respectively. The scalars α weight the various terms.

The output segmentation is the labeling that minimizes :

$$\mathcal{L}^* = \underset{\mathcal{L}}{\operatorname{argmin}}\, E(\mathcal{L}) \tag{2.22}$$

As E is a binary pairwise energy function with submodular pairwise potentials, we minimize it exactly with graph-cuts.

2.5.3.2 Consistent video saliency

The main idea behind the approach in [28] is to estimate salient regions included in frames of a video sequence. To this end, it oversegments each frame into a set of superpixels and uses different spatial and motion features such as the color gradient, and optical-flow. In addition, energy optimization is used in order to perform global refinement. This novel spatiotemporal saliency detection method can estimate the background and the foreground (object), even in the case where the scene of the input video is complex. More precisely, this method has three main steps: *saliency estimation*, *saliency cues refinement* and *spatiotemporal saliency optimization*.

A) **Saliency estimation:**

The purpose of this stage is to produce a primary estimation of the object.

Figure. 2.13 summarizes all phases of saliency estimation steps where (a)

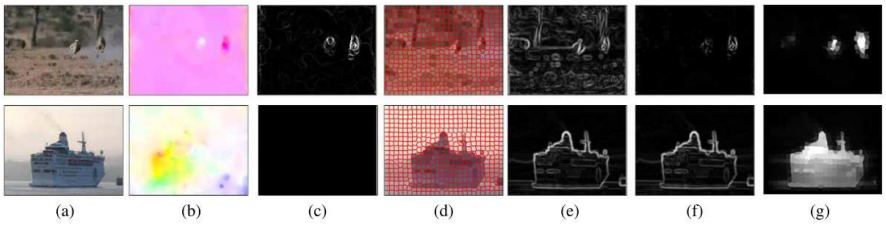


FIGURE 2.13: Illustration of a saliency estimation method steps [28].

Represents two frames from different videos, first they computes the optical-flow field v for each frame of video sequence (Figure. 2.13.(b)) and also computes the optical-flow gradient magnitude M^o of v (Figure. 2.13.(c)). Then they abstract each frame into superpixels (Fig. 2.13.(d)) and after that they Compute the color gradient magnitude M^c of abstraction that previously abstracted (Fig. 2.13.(e)). Thereafter they Combine the color gradient magnitude and the optical-flow gradient magnitude into spatiotemporal gradient field M as in Figure. 2.13.(f) and finally saliency detection results is computed using the gradient flow field (see Figure. 2.13.(g)).

B) **Saliency cues refinement:**

In this step, there is a detection of the salient regions by considering the local spatiotemporal consistency for each frame. It is based on local Saliency cue and global saliency cue.

- **Local Saliency cue:**

  The region contrast against its surrounding scales is used as a saliency cue, which is computed as the summation of its color differences from other regions and weighted by their spatial distances.

- **Global Saliency cue:**

  The global saliency measure of a superpixel is defined as the length of

its shortest distance to the virtual backgrounds. The distance between any two superpixels $R_{t,n}$ and $R_{t,m}$ considers the color distance and the gradient flow field distance.

C) **Spatiotemporal saliency optimization:**

The saliency of superpixel q is $S_{k_q}(x_q)$ computed by the last step. An *energy function* is proposed to encourage the spatiotemporal consistency of the whole video saliency map. The final saliency of each superpixel is defined as s_q, which is further optimized through the proposed spatiotemporal saliency energy function as follows:

$$\mathcal{F} = \mathcal{F}_{unary} + \mathcal{F}_{smooth} \tag{2.23}$$

$$= \alpha \sum_{q} \left(s_q - S_{k_q}(x_q)\right)^2 + \sum_{q,q' \in \mathcal{N}} w_{q,q'} \left(s_q - s_{q'}\right)^2 \tag{2.24}$$

where the set $\mathcal{N}$ contains all the spatially adjacent superpixels within one frame and the temporally adjacent superpixels in a neighborhood: if $\|x_q - x_{q'}\| \leq 800$ and $|k_q - k_{q'}| = 1$, superpixels q and q' are temporally adjacent. The parameter α is the positive coefficient for balancing the relative influence between $\mathcal{F}_{unary}$ and $\mathcal{F}_{smooth}$.

2.5.3.3 Saliency geodesic video segmentation

In this method, there is an abstraction of each frame into superpixels, after that for each superpixel two kinds of edges are extracted, namely spatial edge and motion boundary edges. Then these two types of features are combined to produce the spatiotemporal edge probability map. After that, the distance geodesic is used to generate foreground probability map (initial object) [27].

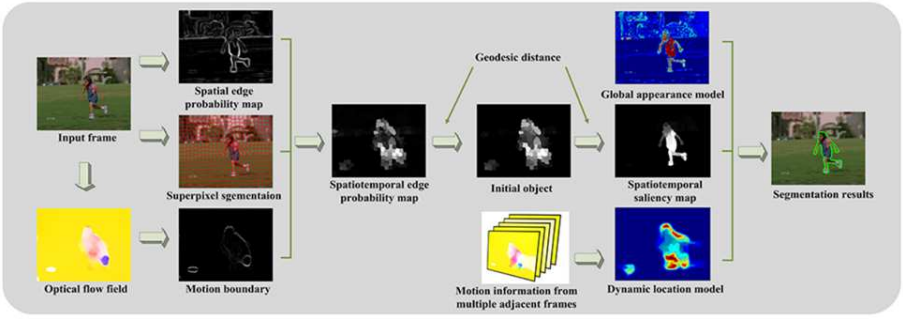


FIGURE 2.14: General review of saliency-aware geodesic video object segmentation [27].

Similarly to other segmentation works [21], an energy function is defined for labeling L of all the pixels is in Eq. (2.21).

2.6 Datasets

There are several datasets of different sizes and varying resolution that are used in video processing and segmentation. In this work we choose three of them **SegTrackV2** [16], **FBMS** [4] and **DAVIS** [22].

2.6.1 SegTrackV2 dataset

SegTrackV2 is a small dataset composed of 14 densely annotated videos of humans and animals contains((see Figure.2.15). It is designed to be challenging with respect to background-foreground color similarity, fast motion, and complex shape deformation [16].

Although several approaches have extensively used it, its content does not sufficiently span the variety of challenges encountered in realistic video object segmentation applications. Furthermore, the image quality is not any more representative of modern consumer devices, and due to the limited number of available

FIGURE 2.15: Sample sequences from the SegtrackV2 dataset [16].

video sequences, progress on this dataset plateaued.

2.6.2 FBMS dataset

The Freiburg-Berkeley Motion Segmentation (FBMS) dataset that contains 59 video sequences with 720 frames is a popular dataset for motion segmentation, i.e. clustering regions with similar motion. Despite being recently adopted by works focusing on video object segmentation [4].

The dataset does not fulfill several important requirements. Most of the videos have low spatial resolution, segmentation is only provided on a sparse subset of the frames, and the content is not sufficiently diverse to provide a balanced distribution of challenging situations such as fast motion and occlusions.

FIGURE 2.16: Sample sequences from the FBMS dataset [4].

2.6.3 DAVIS dataset

The Densely Annotated Video Segmentation (DAVIS) [22] is a novel dataset that specifically designed for the task of video object segmentation. This dataset has a sufficiently large amount of data to ensure content diversity and to provide a uniformly distributed set of challenges. The quality of the data also plays a crucial role, as it should be representative of the current state of technology. To this end, DAVIS comprises a total of 50 sequences, 3455 annotated frames, all captured at Full HD 1080p spatial resolution [22] (see Figure.2.17).

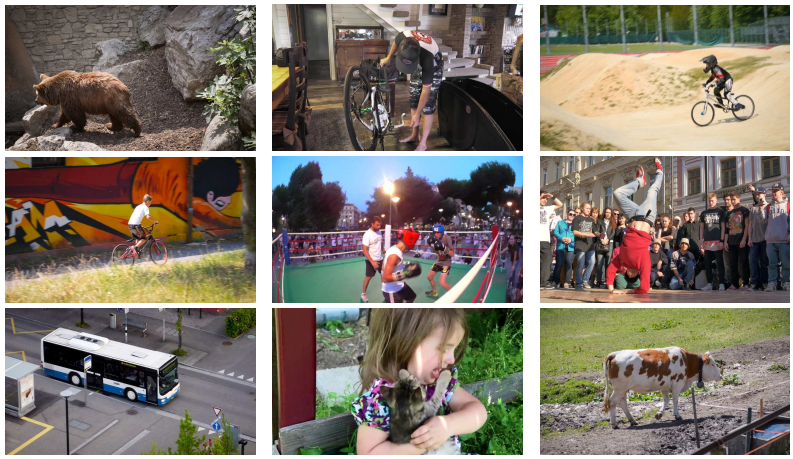


FIGURE 2.17: Sample sequences from the DAVIS dataset [22].

2.7 Evaluation Metrics

We report the precision versus recall curves (PR curves), $F-score$ curves, and mean absolute errors (MAEs) for evaluation.

2.7.1 $F-score$

The precision value represents the ratio of correctly assigned salient pixels to all the pixels in the detected regions, while the recall rate is the percentage of detected regions among the true positive samples. The curves are averaged over each video sequence. The $F-measure$ considers both precision and recall and can be computed as

$$F_\beta = \frac{(1+\beta^2)\cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{2.25}$$

We set $\beta^2 = 0.3$ throughout our experiments.

2.7.2 Mean absolute error (MAE)

The mean absolute error (MAE) is defined between a saliency map S and the binary GT as

$$MAE = \frac{1}{|I|} \sum_{x} |S(x)GT(x)| \tag{2.26}$$

where $|I|$ represents the number of pixels and x stands for all image pixels.

2.8 Conclusion

In this chapter, a general survey about video segmentation is presented. Also, some useful methods and features for improving video segmentation have been presented. In addition, a brief review of some state-of-the-art unsupervised methods for video segmentation has been given. In the following chapter, we extend our study by proposing a new video segmentation approach based on combining and voting spatiotemporal object proposals.

3

Video segmentation using spatiotemporal object proposals

3.1 Introduction

In this chapter, we proposed a new video segmentation approach by combining a spatiotemporal proposal generating technique with a proposal-based video segmentation approach. For the best understanding of the proposed approach, we present a detailed description of the components included in the proposed overall algorithm.

3.2 Object proposals-based saliency detection

The main idea of the proposed approach is the use of the *object proposals* which are regions that represent segments of objects (see Figure. 3.1 for an illustration of object proposals). These regions (proposals) are then used to identify salient object regions by using some saliency cues. The graphic in Figure. 3.1 summarizes the main steps of the proposed object proposals approach.

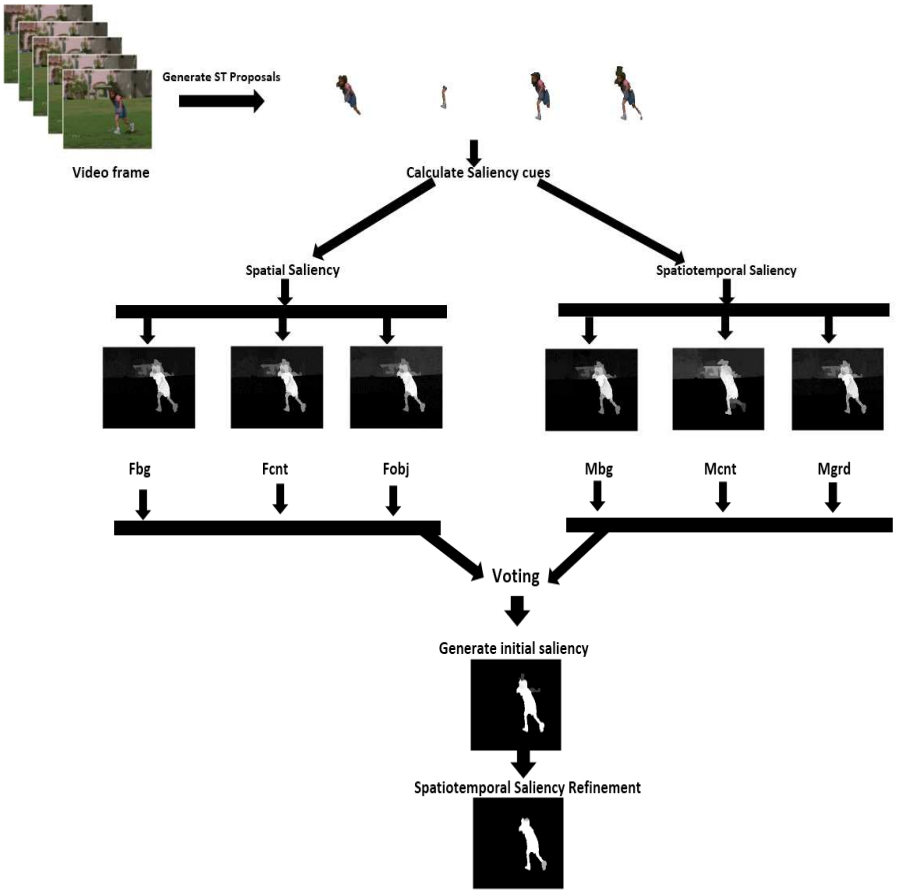


FIGURE 3.1: Video saliency detection using object proposals process.

3.2.1 Spatiotemporal generation of object proposals

For each input frame from the input video $\mathbf{I} = \{I_1, I_2, \ldots\}$, *the static spatial region-ranking method* is used to generate hundreds or thousands of proposals via [8]. We note this set of proposals $\mathbf{P_{sp}}$. As shown by the Algorithm 1, this method can generate good initial candidates based on *spatial cues* such as occlusion boundaries, geometric context, and texture. However, using only spatial cues can lead to non-temporally consistent proposals. To overcome this limitation, we propose to use a new spatiotemporal approach to filter out the proposals which are not temporally consistent. In other words, remove out object proposals located out of the region of interest (ROI) detected by a given spatiotemporal approach. Note that, sophisticated spatiotemporal approaches such as [27] or [28] can be used to detect accurate location priors. In our case, we use the [27] one.

Let $\mathbf{P_{sp}}$ be the set of candidate spatial proposals generated by Algorithm 1 as described in the previous paragraph. And let $\mathbf{P_{st}}$ be the set of resulting filtered spatiotemporal proposals, where $\mathbf{P_{st}} = \{p_t^1, p_t^2, \cdots\}$. Algorithm 2 summarizes the steps of the proposed procedure used to filter out "bad" spatiotemporal proposals.

In fact, the proposed technique filters out proposals in $\mathbf{P_{st}}$ based on the saliency map generated by the method in [27]. Firstly, the values of the input saliency map are rescaled in the range $[0, 1]$. Let $\mathbf{M}$ be the resulting saliency map. Then, the resulting saliency map $\mathbf{M}$ is processed using the following equation $1-\exp(-f*\mathbf{M})$. After that, the saliency map $\mathbf{M}$ is binarized using the Otsu's algorithm and let $\mathbf{FG}$ be the resulting binary mask. Finally, we take the set of proposals inside the binary mask $\mathbf{FG}$ with an overlapping ratio of ξ. The rest of proposals are removed out. The new set of proposals is noted $\mathbf{P_{st}}$.

Once the spatiotemporal proposals are generated, we use the approach in [10] to generate the final *video saliency map*. To this end, two kinds of saliency cues are fused. Namely, the *spatial saliency cues* and *motion saliency cues*, respectively.

Algorithm 1: Generate spatial object proposals [8].

Data: The input frame from the video sequence I_t
Result: The set of spatial proposals $\mathbf{P_{sp}}$
Precomputation;
- *Occlusion Boundaries*;
- *Geometric Context (non-planar vertical surface)*;
- *Hierarchical Segmentation*;
- *Probability of BG region classifier*;

Train Classifiers;
- *Homogeneous Region Classifier*;
- *Region Affinity Classifier*;
- *Layout Classifier*;
- *Ranking Model*;

Region Proposal;
- *Select seeds*;
- *For each image I, seed* $S \in S_I$ *and parameters* (γ, β) ;
  * *Compute superpixel affinity map*;
  * *Propose Region*;
- *Split regions with disconnected components and add to set*;
- *Remove redondant regions with* $\geq 90\%$ *overlap*;

Region ranking;
- *For each proposal* $p \in P$*, compute appearance features* x_p;
- *For each image I find (approximate) highest scoring ranking with greedy inference*;

3.2.2 Ranking of spatiotemporal object proposals

Based on the objectness score from [8] that refers how likely a proposal is to contain an object, top 200 candidates are selected for each frame. After that, a ranking score $\mathrm{R}(p_t^n)$ is computed for each proposal p_t^n in frame I_t, which is defined as Eq. (3.1) [10]:

$$R(p_t^n) = R^F(p_t^n) + R^M(p_t^n) \tag{3.1}$$

where R^F and R^M represent *spatial* and *motion* saliency scores, respectively

Algorithm 2: Proposed procedure to detect spatiotemporal proposals.

Data: Spatial candidates $\mathbf{P_{sp}}$, Saliency map $\mathbf{M}$, Parameters (list of factors $\mathcal{F}$, list of levels $\mathcal{L}$, overlap threshold ξ)
Result: The set of filtered spatiotemporal proposals $\mathbf{P_{st}}$
Rescale the saliency map $\boldsymbol{M}$ *in the range* $[0, 1]$;
for *each factor* f *in the factors list* $\mathcal{F}$ **do**
  $\mathbf{M} \leftarrow 1 - \exp(-f * \mathbf{M})$;
  $\boldsymbol{th} \leftarrow$ grey thresholding of $\mathbf{M}$ using Otsu's thechnique;
  for *each level* l *in the levels list* $\mathcal{L}$ **do**
    $\boldsymbol{th_{new}} \leftarrow \boldsymbol{th} + l$;
    $\mathbf{FG} \leftarrow$ binarise$(\mathbf{M}, \boldsymbol{th_{new}})$;
    for *each proposal* $\boldsymbol{p}$ *in the spatial candidates* $\boldsymbol{P_{sp}}$ **do**
      $\boldsymbol{M_p} \leftarrow$ the binary mask of the proposal $\mathbf{p}$;
      $\boldsymbol{\omega} \leftarrow$ the overlap between $\boldsymbol{M_p}$ and $\mathbf{FG}$;
      if $\boldsymbol{\omega} > \xi$ **then**
        Add the proposal p to the list of spatiotemporal proposals $\mathbf{P_{st}}$;
      end
    end
  end
end

[10].

3.2.3 Spatial saliency analysis

In addition to the abjectness score $\mathbf{F_{obj}}$, we use two other saliency priors for saliency detection, namely, background prior score $\mathbf{F_{bg}}$ and center-surround contrast prior score $\mathbf{F_{cnt}}$ in order to calculate spatial score R^F formulated as Eq.(3.2) [10]:

$$R^F(p_t^n) = F_{obj}(p_t^n) + F_{bg}(p_t^n) + F_{cnt}(p_t^n) \tag{3.2}$$

Note that the three previous(F_{obj}, F_{bg}, F_{cnt}) terms must be normalized to $[0, 1]$,

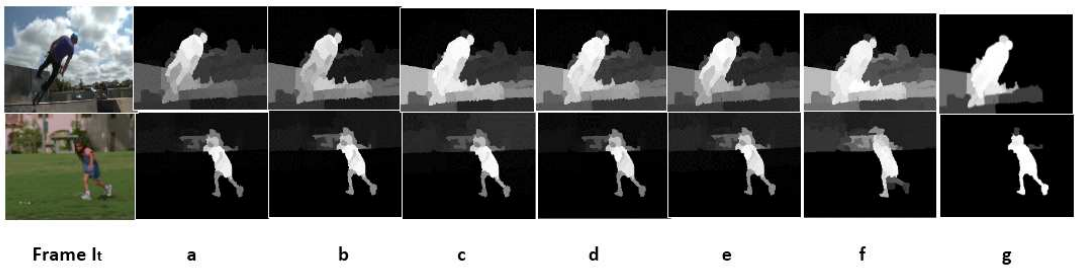


FIGURE 3.2: Saliency results generated with each prior score. (a) Proposal background prior, which is generated by accumulating all the proposals with their background scores as in Eq. (3.3). (b) center-surround contrast prior Eq. (3.4). (c) The proposal object prior has been obtained during the process of extracting object segmentation candidates (d) Motion contrast as in Eq. (3.7). (e) Gradient summation in Eq. (3.8). (f) Object proposal consistency by Eq. (3.9). (g) Initial saliency results by using the combined scores of all priors as in Eq. (3.11)

(see Figure. 3.2 (a), (b), and (c) for an illustration of spatial saliency priors)[10].

3.2.3.1 Background prior score

The background regions are the most probable regions that could be connected with the image boundaries. Zhu and al [31] proposed a type of region-level background prior called boundary connectivity. Boundary connectivity is defined as the percentage of intersection between object proposal p_t^n and the boundary $Bnd(I_t)$ of frame I_t to the square root of its area Area(p_t^n).

The background prior score $\mathbf{F_{bg}}$ for proposal p_t^n is defined as Eq. (3.3) [10]:

$$F_{bg}(p_t^n) = exp(-\frac{p_t^n \cap Bnd(I_t)}{\sqrt{Area(p_t^n)}}) \tag{3.3}$$

Whenever that the proposal occupies a large part of the boundaries, it will be

assigned by a higher background prior score which indicates that it is less likely to be a salient object proposal [10].

3.2.3.2 Proposal center-surround contrast score

This saliency cue is used to measure the contrast between a proposal and its surroundings, to get the ratio of how this proposal is homogeneous with their neighbors. For that, first for proposal p_t^n and its neighboring regions a CIELab color space histogram is computed for each one. The contrast score F_{cnt} for proposal p_t^n is computed as Eq. (3.4) [10]:

$$F_{cnt}(p_t^n) = 1 - exp(-\chi^2(Hist_c(p_t^n), Hist_c(Dil(p_t^n)))) \tag{3.4}$$

where $Dil(p_t^n)$ denotes the dilated region of proposal p_t^n, and $\chi^2(Hist_c(p_t^n), Hist_c(Dil(p_t^n)))$ is the chi-squared distance between color histograms [10].

3.2.4 Spatiotemporal saliency analysis

In addition to appearance cues, the motion cues are an important factor to be taken into account for video saliency detection. To this end, motion contrast-based saliency method is designed. Firstly, the optical-flow is computed using the *large displacement optical-flow method* [5]. After that, smoothing processing is applied on the initial optical-flow maps over the temporal domain to obtain more robust motion information. Finally, a motion saliency score is formulated as Eq. (3.5)[10]:

$$R^M(p_t^n) = M_{bg}(p_t^n) + M_{grd}(p_t^n) + M_{cnt}(p_t^n) \tag{3.5}$$

where M_{bg} is a motion contrast score, M_{grd} is an optical-flow gradient based score and M_{cnt} represents a motion consistency score (see Figures 3.2 (d), (e), and (f) for an illustration of spatiotemporal saliency priors)[10].

3.2.4.1 Smoothing optical-flow

Generally, the object is not moving in all the sequence of video which means it may stop moving abruptly, that will cause discontinuities and inaccuracies in the optical-flow. To this end, a Gaussian filter G is used to preserve temporal continuity and obtain a more robust optical-flow estimation and get a smoothing version of the optical-flow information[10].

The smoothed optical-flow o_t is define as Eq 3.6:

$$o_t = \frac{\sum_{i=-l}^{l} G(i;0,1) * o_{t+i}}{\sum_{i=-l}^{l} o_{t+i}} \tag{3.6}$$

where l is the number of adjacent frames used in smoothing the optical-flow.

3.2.4.2 Motion contrast-based saliency

In video sequences, the optical-flow of foreground objects and background are generally distinguishable. Consequently, a motion background contrast score $M_{bg}(p_t^n)$ is designed to reflect this theory. From the background prior score F_{bg} computed via Eq. (3.3), we get the background proposals where their F_{bg} is less than e^1 [10].

For each two consecutive frames I_t and I_{t+1}, we get the smoothed optical-flow $o_t = (u, v)$ from Eq. (3.6), then we calculate the flow magnitude $o^{grad} = grad(\sqrt{u^2 + v^2})$ and the distribution of flow orientation $o^{ori} = arctan(v/u)$ to make flow histograms of p_t^n and BP, where BP is the background of frame I_t that contains all proposals that represent background (calculated via Eq. (3.3)) [10].

$$M_{bg}(p_t^n) = 1 - exp(-\chi^2(Hist_{flow}(p_t^n), Hist_{flow}(BP))) \tag{3.7}$$

Whenever that the motion contrast score of proposal p_t^n is small that indicates it is less likely to be a salient object proposal [10].

3.2.4.3 Motion gradient summation

The rationale behind motion contrast is that the motion pattern of an object is distinct from that of the background. This assumption can also be exploited via the gradient of the optical-flow. Indeed, distinct motion patterns cause velocity and orientation discontinuities. That is, the optical-flow gradient will be large around the salient object boundary. Therefore, a motion gradient score $M_{grd}(p_t^n)$ is computed by making use of the motion gradient summation technique such in [39]. This score is defined as the average Frobenius norm of the optical-flow gradient in the boundary of object proposal p_t^n [10].

$$M_{grd}(p_t^n) = \|o_t\|_F = \sqrt{\sum_{i=x,y}\sum_{j=x,y} |(\mu_i, \upsilon_j)|^2} = \sqrt{\mu_x^2 + \mu_y^2 + \upsilon_x^2 + \upsilon_x^2} \tag{3.8}$$

where $o_t = (\mu, \upsilon)$ is the smoothed optical-flow of consecutive frames I_t and I_{t+1} , μ_x, and μ_y are optical-flow gradients in the $x-$direction and υ_x and υ_y are those in the $y-$ direction [10].

According to the definition of the motion gradient score, the higher value a pixel is, the higher the possibility it associates with moving salient object boundary. Actually, due to the approximation of optical-flow computation, the gradient of optical-flow cannot correspond to magnitude values in the boundaries of a moving

object exactly. Therefore, we compare the average optical-flow gradient magnitude at the proposal boundary and in a dilated version of this boundary (10 pixels)[10].

3.2.4.4 Object proposal consistency

It is clear that salient object regions are consistent over time. Therefore, proposals corresponding to salient objects should also remain temporally consistent in adjacent frames. An interframe score is defined for each proposal p_t^n in frame I_t, based on the salient proposals of the previous frame. Specifically, each object proposal p_{t1}^n for frame I_{t1} can be warped to frame I_t according to the forward optical-flow. The overlap between proposal p_t^n in frame I_t and the warped object proposals is then estimated. This yields the temporal consistency score M_{cnt} defined by Eq. (3.9)

$$M_{cnt}(p_t^n) = \frac{\hat{P}_{t-1}^n \cap P_t^n}{Area(P_t^n)} \tag{3.9}$$

where $\hat{P}_{t-1}^n$ denotes the warped regions of the proposal P_t^n from frame $t1$ to frame t according to optical-flow o_t. Based on this function, fractional proposals corresponding to the background should be filtered out, while object proposals should remain consistent over time [10].

3.2.5 Voting for Saliency

After the calculation of the previous saliency scores, a ranking score $R(p_t^n)$ is computed for each proposal p_t^n using Eq. (3.1) where for each frame I_t, the 20% of proposals are taken and a subset P_t^s of proposals P_t that have the highest ranking scores are constructed, and m as the number of salient object proposals in P_t^s [10].

for each salient proposal $p_t^i \in \mathbf{P_t^S}$, a binary mask M_t^i is generated where:

$$M_t^i(x) = \begin{cases} 1 & \text{if pixel x in frame } I_t \text{ belongs to proposal } p_t^i \\ 0 & \text{otherwise} \end{cases}$$

then for each pixel x, the saliency value is computed by accumulating the binary masks of the selected proposals $\mathbf{P_t^S}$, which is computed as:

$$O_t(x) = \frac{1}{m}\sum_i M_t^i(x) \tag{3.10}$$

Then this value is normalized to get an initial saliency estimation as (see Figure. 3.2) Eq. (3.11)[10]:

$$S_t^{Ini}(x) = 1 - exp(-\frac{O_t(x)}{\sigma^2}) \tag{3.11}$$

where σ is a constant parameter set to $\sigma = 0.3$,

3.2.6 Spatiotemporal saliency refinement

Generally, the initial saliency can be considered as an acceptable solution. However, some ambiguities appear at the boundary of the objects, along with temporal non-consistency. Consequently, a *saliency refinement* process is introduced to improve the initial saliency estimation [10].

3.2.6.1 Object boundary refinement

The idea behind this step is to refine the saliency map to improve the initial estimation. Firstly, the SLIC method [1] is applied for each frame I_t to get superpixels $R_t = \{r_t^1, r_t^2, ...\}$ (about 500 superpixels per frame). Then we obtain

the set $S_t^{Ini}(r_t^i)$ of superpixels r_t^i where its value is the averaged saliency value of its pixels. After that, there is a classification of superpixels $\mathbf{R_t}$ according to two thresholds τ^{high} and τ^{low} into three distinct parts: 1) foreground (salient) regions $\mathbf{F_t}$; 2) background (nonsalient) regions $\mathbf{B_t}$; and 3) uncertain regions $\mathbf{U_t}$ [10].

$$\mathbf{F}_t = \left\{ r_t^f | S_t^{Ini}(r_t^f > \tau^{high}, \forall r_t^f \in \mathbf{R}_t) \right\} \tag{3.12}$$

$$\mathbf{B}_t = \left\{ r_t^b | S_t^{Ini}(r_t^b < \tau^{low}, \forall r_t^b \in \mathbf{R}_t) \right\} \tag{3.13}$$

$$\mathbf{U}_t = \mathbf{R}_t - \mathbf{F}_t - \mathbf{B}_t \tag{3.14}$$

Where the two thresholds τ^{high} and τ^{low} are set to 0.8 and 0.2, respectively [10].

Moreover, a graph-based approach is followed to refine the saliency value of the uncertain regions, where nodes are superpixels and links between any two adjacent superpixels (r_t^i, r_t^j) are weighted by the Euclidean distance between features (the average CIELab color space and the mean optical-flow magnitude).

In addition, the geodesic distance is computed between uncertain superpixels $r_t^u \in U_t$ and background superpixels $r_t^b \in B_t$ and foreground superpixels $r_t^f \in F_t$. Note that, the weight is set as zero between any two background superpixels and any two foreground superpixels. The saliency value of each uncertain superpixel $r_t^u \in \mathbf{U_t}$ is then given by Eq. (3.15) [10]

$$S_t^{Ref}(r_t^u) = 1 - exp\big(\underset{r_t^f \in F_t}{max}\ d^{geo}(r_t^u, r_t^f) \times \underset{r_t^b \in B_t}{min}\ d^{geo}(r_t^u, r_t^b)\big) \tag{3.15}$$

3.2.6.2 Temporal saliency consistency

Although the refined saliency estimation S^{Ref} gives satisfactory results, it can be further improved, especially regarding the temporal consistency. So, a propagation process is introduced to propagate the per-frame saliency maps over time. For the first frame I_1, the location prior is initialized with the refined proposal saliency map S_1^{Ref}. For the following frames, the saliency value of superpixel r^j is computed as Eq. (3.2.6.2)[10]:

$$S_{t+1}^{Fin}(r_{t+1}^j) = \frac{\sum_i \phi(r_t^i, r_{t+1}^j).\psi(r_t^i)}{\sum_i \phi(r_t^i} S_t^{Ref}(r_t^i)$$

$$\text{with } \psi(r_t^i) = exp(-o^{grad}(r_t^i)) \tag{3.16}$$

Where $\phi(r_t^i, r_{t+1}^j)$ indicates the overlap between superpixel r_i^t warped by optical-flow and superpixel r_t^{j+1}, and o^{grad} is the same normalized histogram of flow magnitude to compute the motion contrast score [10].

3.3 Conclusion

In this Chapter of our thesis, we highlight our proposed video segmentation approach which is based on a new spatiotemporal proposal generation technique combined with a recently proposed approach for video saliency detection using object proposals. In the next Chapter, we present the qualitative and quantitative results that we have obtained during the experimental analysis [10].

4

Experimentations

4.1 Introduction

In this Chapter, we provide a comparison between the proposed method and a set of video segmentation approaches. We use the implementations provided by the authors of these methods to extract the characteristics of each approach compared to the proposed one. In this experiment, all the tests were performed on a Windows platform and under the same computer configuration Intel(R) Core(TM) i3-4005U CPU @ 1.70 GHz with 4.00 Go, on Matlab as a development tool.

4.2 Evaluation

We use a set of models that used for the segmentation whether the spatial or the spatiotemporal and we evaluate them on two datasets Segtrack V2 and Davis which are with 14, and 50 video sequences, respectively and each dataset contain manually annotated pixelwise ground-truth for every frame. Our main purpose, in this Chapter, is this section is to make an experimental comparison between their quality and their performance and to give the readers a comprehensive view of those methods compared with the proposed method.

For our experiments, we apply some evaluation measures to determinate how are close model predictions to human annotations (groundtruth) and to measure the proportion of the accurate segmentation from the inaccurate one for all the models that we choose and the proposed method.

And to achieve that, we utilize three metrics which are **"F-measure"**, **"the MAE measure"** and **precision-recall (PR) measure**.

Concerning F-measure, we need for each method to binarize the obtained saliency maps to compute Precision and Recall in order to utilize them in the calculation of the F-measure in each video for every threshold, after that, we take the average of the F-measure over videos in a dataset. Either for the MAE, for each method, we use the obtained saliency maps and the groundtruth to calculate the error percentage for each video then we take the average. Finally, the best method is that which gives a value of F-measure close to 1 and a percentage of error MAE close to 0.

4.2.1 Comparison on SegtrackV2 dataset

The SegTrackV2 dataset was originally introduced to evaluate tracking algorithms. For this dataset, we choose a set of segmentation methods, which are : Saliency

Filters Contrast Based Filtering for Salient Region Detection **SF** [9], Saliency detection: A spectral residual approach **SR** [12], Saliency Detection via Absorbing Markov Chain **MC** [13], Saliency estimation using a non-parametric low-level vision model **SIM** [14], Static and space-time visual saliency detection by self-resemblance **SeR** [24], Visual saliency detection by spatially weighted dissimilarity **SWD**[6], Consistent Video Saliency Using Local Gradient Flow Optimization and Global Refinement. **CVS** [28], Saliency aware geodesic video object segmentation **SAGV** [27], and Video Saliency Detection Using Object Proposals method (VSDOP) [10]. We compare these approaches with the proposed method to show the effectiveness of the proposed approach.

Table 4.1 presents a simple description of aforementioned methods:

	Method	Year	Features				Imlementation
			Color Space	Edges	Region	Motion	
Spatial	**SR**	2007	Rgb	No	pixel	No	Matlab
	SeR	2009	Rgb	yes	Pixel	No	Matlab
	SIM	2011	Rgb	No	Pixel	No	Matlab
	SWD	2011	Rgb	No	patches	No	Matlab and C/C++
	SF	2012	Rgb	No	pixel and region	No	Matlab
	MC	2013	Rgb	No		No	Matlab
	SP	2014	Rgb and Lab	No	pixel and superpixel	Yes	Matlab and C/C++
Spatiotemporel	**CVS**	2015	RGB and LAB	yes	pixel and superpixel	yes	Matlab and C/C++
	SAGV	2015	Rgb and Lab	yes	pixel and superpixel	yes	Matlab
	SGSP	2015	Rgb and Lab	yes	pixel and superpixel	yes	Matlab and C/C++
	VSDOP	2017	Rgb and Lab	yes	superpixel an object proposals	yes	Matlab and C/C++

Table 4.1: Description of used spatial and spatiotemporal methods.

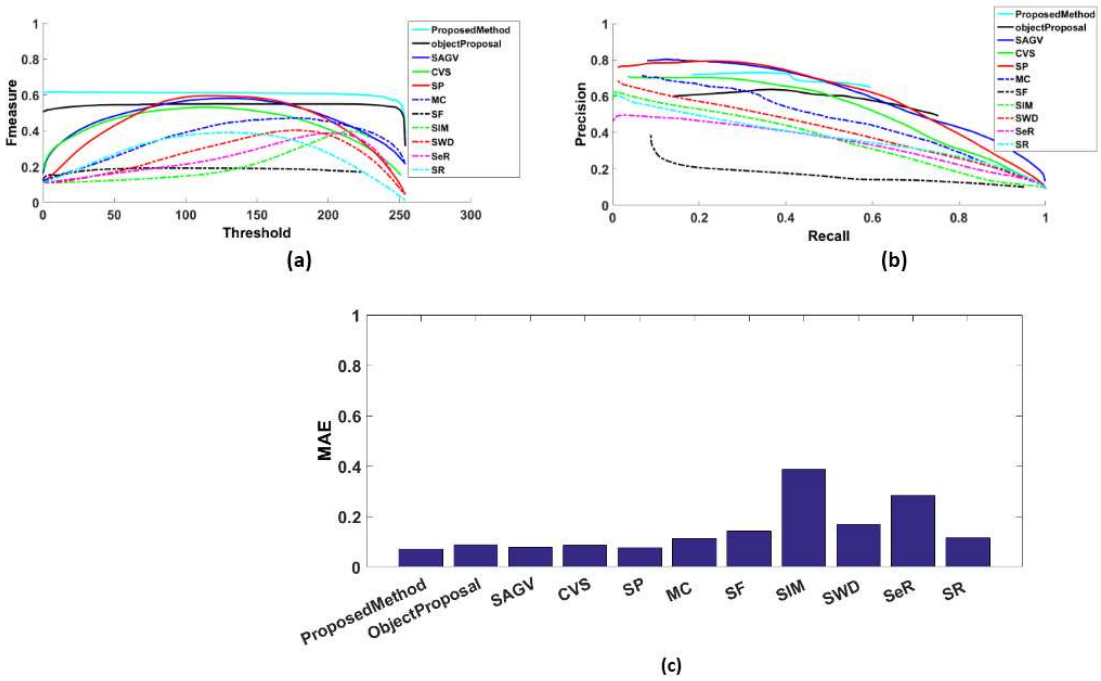


FIGURE 4.1: Illustration of segmentation methods results on SegTrackv2 dataset. (a) F-measure. (b) Recall and Precision. (c) MAE.

As we can see in the Figure. 4.1 and Table 4.2, there is a difference in methods results, where we can observe that there is two main groups of those models. We can observe also that the first group of methods has better results in terms of all the three metrics.

In fact, the first group contains CVS, SAGV, SP, along with the proposed method. Firstly, one characteristic of those models is that they have a high F-measure value, in most of cases superior to 0.5. Secondly, its main absolute error is small compared with the second group that contains the rest of methods and their highest F-measure are inferior to 0.5.

These results can be justified by the fact that the first group are *spatiotemporal models* which take into account motion cues (optical flow) in addition to the appearance factor (superpixels, color...) where that enhances the process of the

Method	Proposed method	Object proposal	SAGV	CVS	SP	MC	SF	SIM	SWD	SeR	SR
bird_of_paradise	**0.927**	**0.845**	**0.607**	**0.534**	**0.959**	**0.878**	**0.883**	**0.224**	**0.545**	**0.310**	**0.259**
birdfall	**0.044**	**0.036**	**0.252**	**0.057**	**0.778**	**0.041**	**0.017**	**0.122**	**0.054**	**0.029**	**0.031**
bmx	**0.837**	**0.824**	**0.699**	**0.518**	**0.773**	**0.758**	**0.390**	**0.444**	**0.526**	**0.505**	**0.592**
cheetah	**0.538**	**0.158**	**0.727**	**0.754**	**0.606**	**0.473**	**0.262**	**0.687**	**0.568**	**0.601**	**0.704**
drift	**0.823**	**0.836**	**0.743**	**0.578**	**0.562**	**0.382**	**0.132**	**0.498**	**0.451**	**0.515**	**0.460**
frog	**0.764**	**0.777**	**0.703**	**0.726**	**0.711**	**0.408**	**0.290**	**0.480**	**0.466**	**0.558**	**0.437**
girl	**0.867**	**0.868**	**0.861**	**0.891**	**0.793**	**0.792**	**0.293**	**0.739**	**0.676**	**0.692**	**0.428**
hummingbird	**0.504**	**0.448**	**0.565**	**0.462**	**0.540**	**0.282**	**0.180**	**0.347**	**0.417**	**0.271**	**0.539**
monkey	**0.800**	**0.804**	**0.891**	**0.827**	**0.664**	**0.689**	**0.507**	**0.431**	**0.506**	**0.260**	**0.602**
monkeydog	**0.285**	**0.188**	**0.412**	**0.475**	**0.248**	**0.220**	**0.051**	**0.073**	**0.307**	**0.200**	**0.299**
parachute	**0.886**	**0.583**	**0.933**	**0.659**	**0.950**	**0.814**	**0.040**	**0.785**	**0.626**	**0.886**	**0.318**
penguin	**0.136**	**0.612**	**0.586**	**0.493**	**0.518**	**0.709**	**0.631**	**0.479**	**0.593**	**0.494**	**0.478**
soldier	**0.770**	**0.683**	**0.619**	**0.504**	**0.647**	**0.516**	**0.074**	**0.651**	**0.171**	**0.394**	**0.453**
worm	**0.820**	**0.835**	**0.831**	**0.876**	**0.687**	**0.784**	**0.036**	**0.579**	**0.544**	**0.530**	**0.412**
F-measure	**0.615**	**0.550**	**0.581**	**0.531**	**0.595**	**0.471**	**0.191**	**0.396**	**0.402**	**0.386**	**0.389**

TABLE 4.2: The best result of F-measure for each method per video.

segmentation and makes it more precise in determining the object, contrary to the second group which contains only the *spatial methods* where only the appearance model is used to infer the segmentation mask.

Also as shown on Figure. 4.1, we can observe that the PR curve(4.1(b)) and F-measure curve(4.1(a)) given by the proposed method is clearly above baselines and have a lower MAE value compared the other methods which indicate that their prediction values are close to the groundtruth and it has a higher accuracy in video segmentation.

More precisely, we can observe that the proposed method which is a combination of spatiotemporal generation of object proposals and video segmentation using object proposals surpasses the original method [10]. In fact, the original method of [10] gives an F-mesure of 0.55, however the proposed method results gieves an F-measure=0.61. This is due to the fact that the original method is based only on spatial object proposals which leads to inaccurate segmentations,

especially when there is a camouflage between object and background. So, the spatial features is insufficient to give correct object proposals. Consequently, we include the spatiotemporal cues on the object proposals generation step which gives a more accurate location of the object.

To get a qualitative comparison, we take five frames for each chosen videos (bird of paradise, girl, monkeydog, parachute). Note that every video have a challenge. This allows to evaluate the robustness of the proposed method compared with the rest of methods.

Figures. 4.2, 4.3, 4.4 and 4.5 show qualitative results given by the proposed approach along with the compared methods. We notice that the result of the spatiotemporal methods is more accurate than the spatial ones excepting the MC and SF where their result is close to the spatiotemporal in "bird of paradise" video because the object was very salient (the background and the foreground colors are easy to discriminate).

As we can observe from the qualitative results, the results of the proposed method are very close to the groundtruth annotations which gives an accurate detection of the objects.

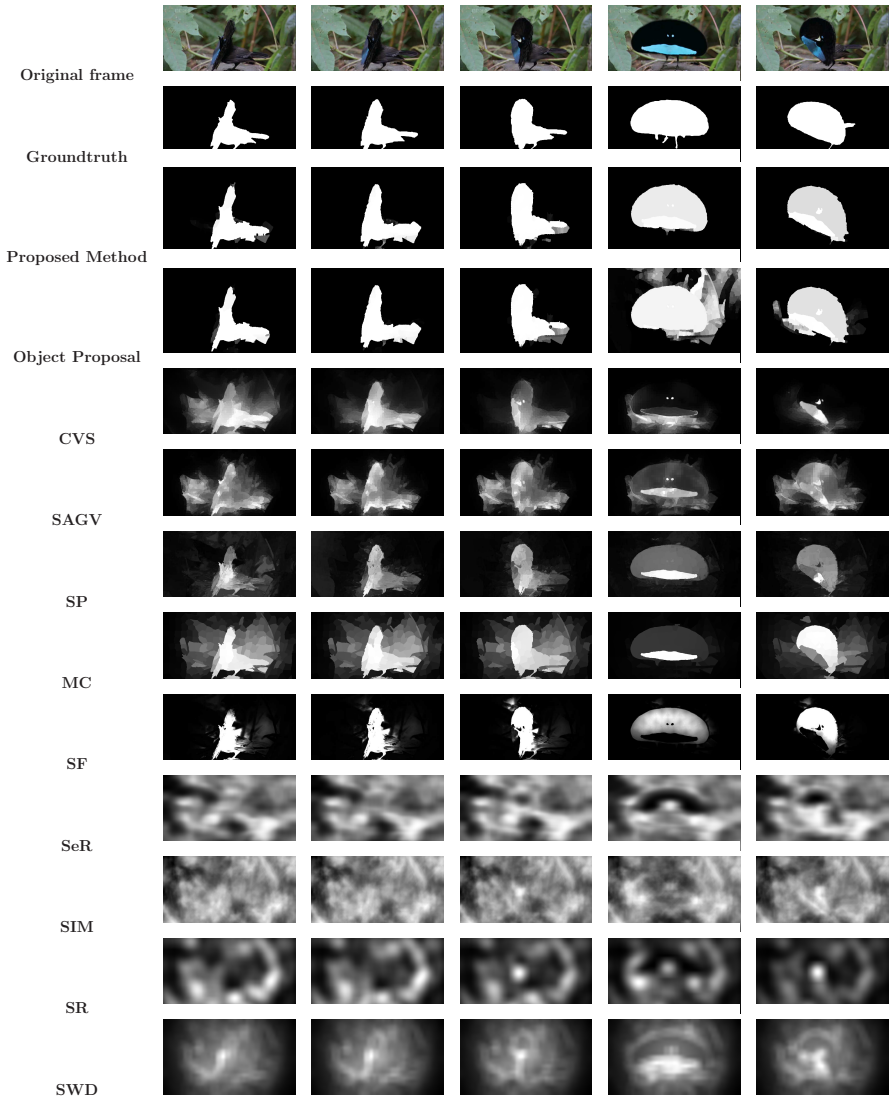


FIGURE 4.2: Visual comparison of methods with GT on SegtrackV2 for bird of paradise video.

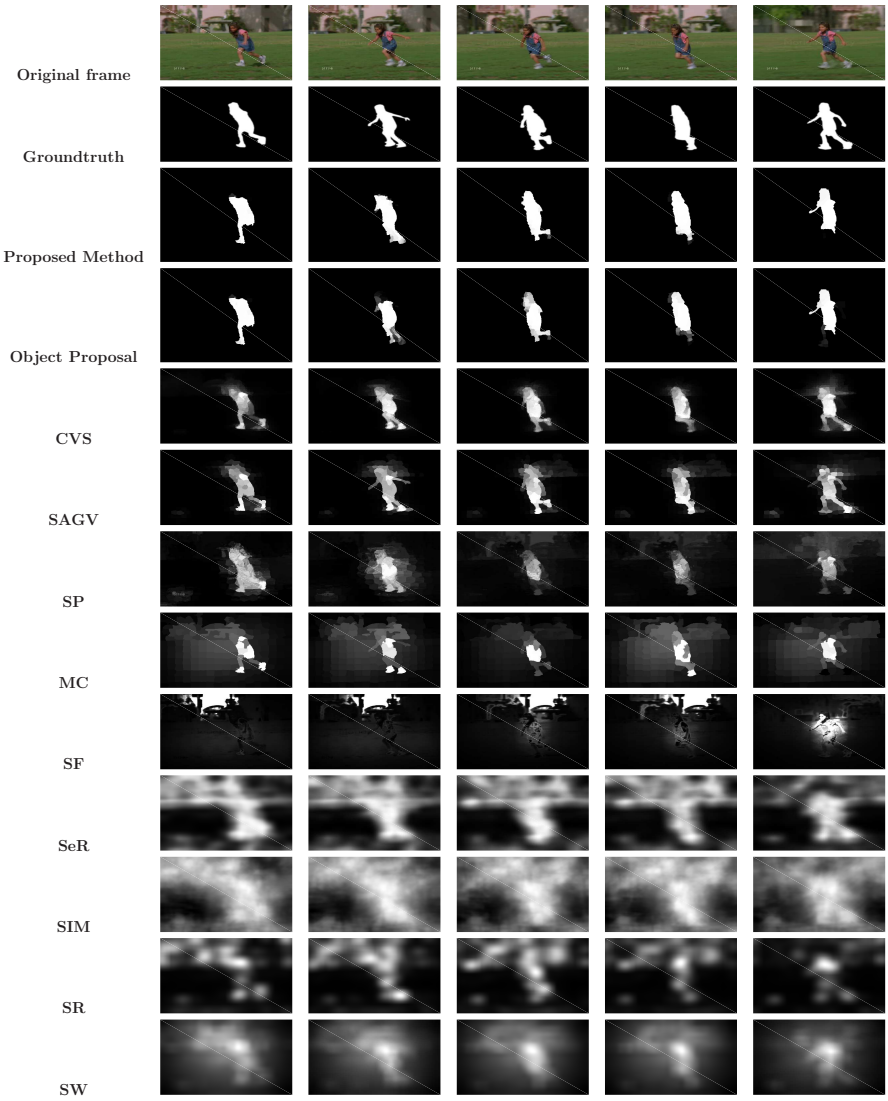


FIGURE 4.3: Visual comparison of methods with GT on SegtrackV2 for girl video.

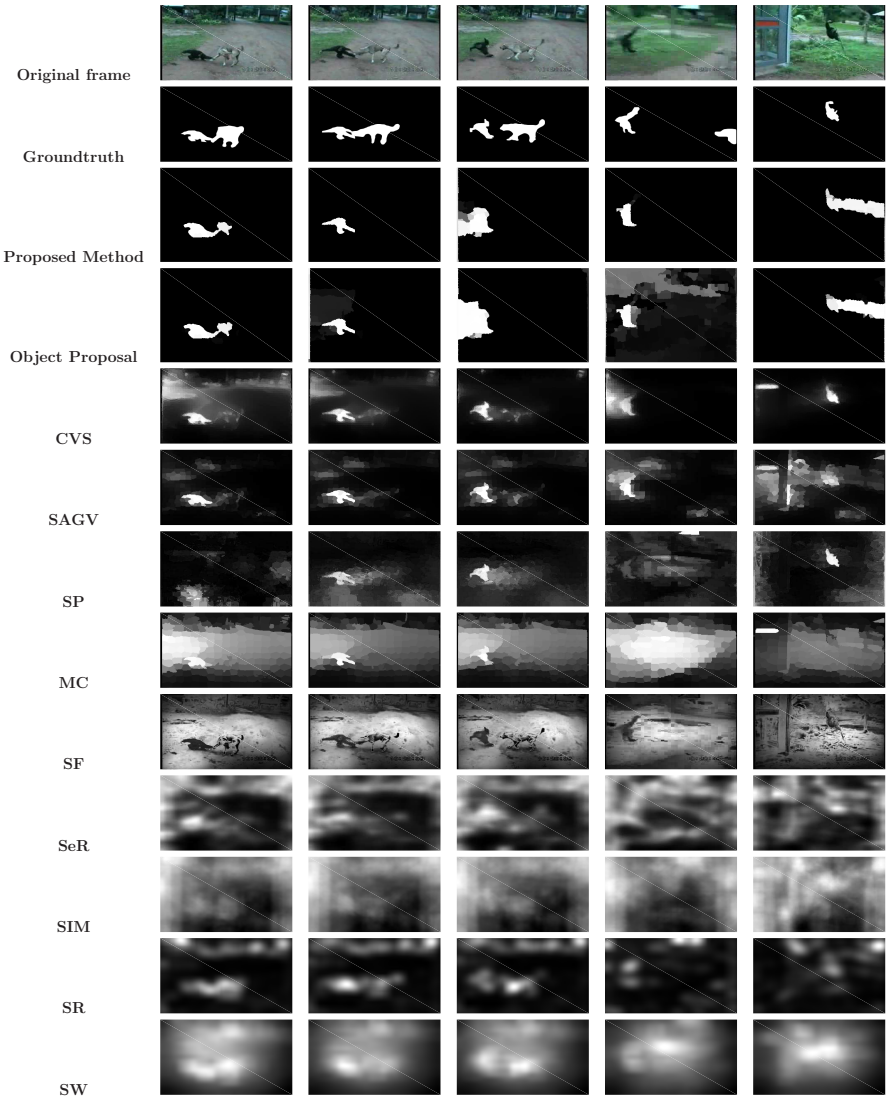


FIGURE 4.4: Visual comparison of methods with GT on SegtrackV2 for monkeydog video

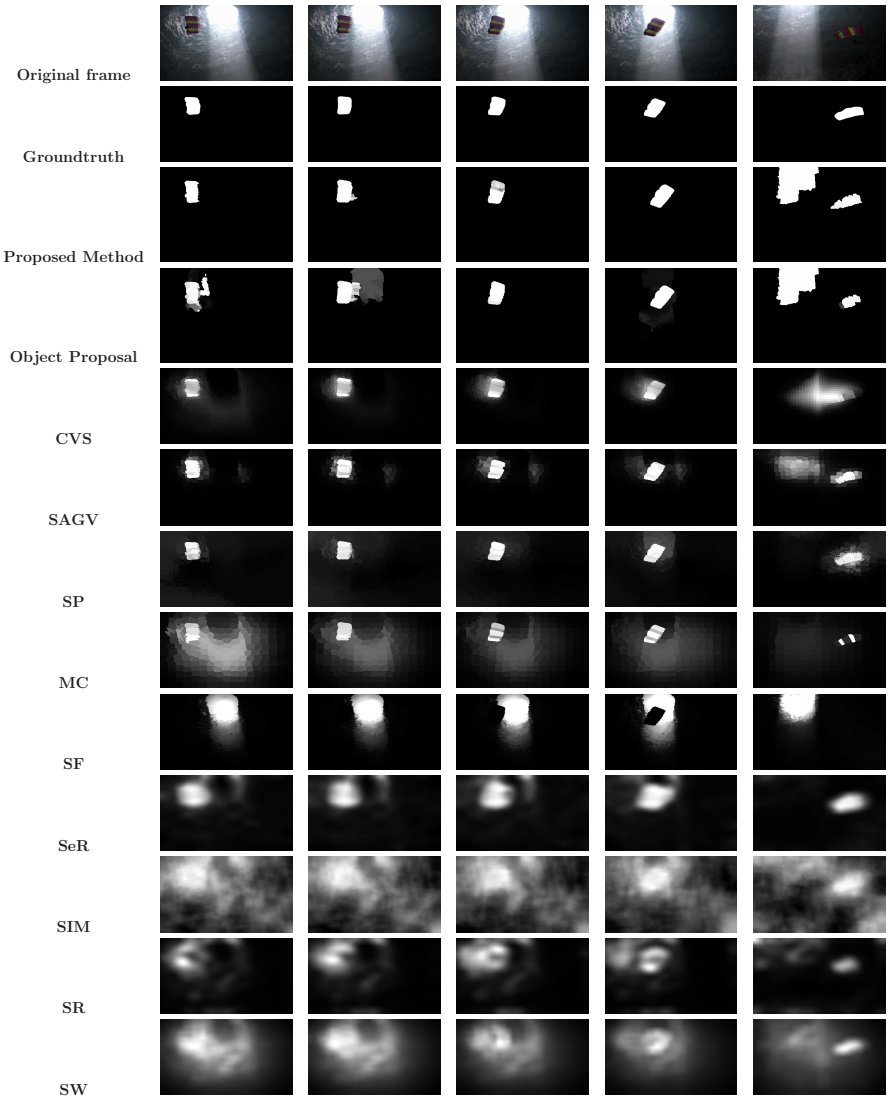


FIGURE 4.5: Visual comparison of methods with GT on SegtrackV2 for parachute video

4.2.1.1 Comparison on Computational time

When we compute the three metrics on methods and found a method better than the others, this doesn't really indicate that it is the optimum one. There are other factors to consider, among them we find the computational time which is a sensitive factor to be observed.

So we calculate the computational time for each method for a typical 327×259 and 360×640 frame from Segtrackv2. The obtained results are shown in the Table 4.3. As a general observation, the computational time of the spatiotemporal methods is bigger than the spatial ones and that is because the first ones use in addition to the spatial features, the spatiotemporal features such as the optical flow.

Method	**MC**	**SF**	**SIM**	**SWD**	**SeR**	**SR**	**CVS**	**SAGV**	**Proposed method**
Computational time(s) per frame 327x259	1.45	0.76	1.26	0.23	2.12	0.16	35.51	14.62	166.27
Computational time(s) per frame 360x640	0.42	1.93	4.76	0.43	1.55	0.03	20.02	20.15	601.78

Table 4.3: Computational time of methods on Segtrackv2 dataset

4.2.2 Comparison on Davis dataset

Regarding this dataset, due to our limited project time and because the Davis dataset is so big, we evaluate it in just spatial methods (the same methods that we use in segtrack) which are : **SF** [9], **SR** [12], **MC** [13], **SIM** [14], **SeR** [24],

SWD[6], and for the spatiotemporal ones, we take precomputed results which are

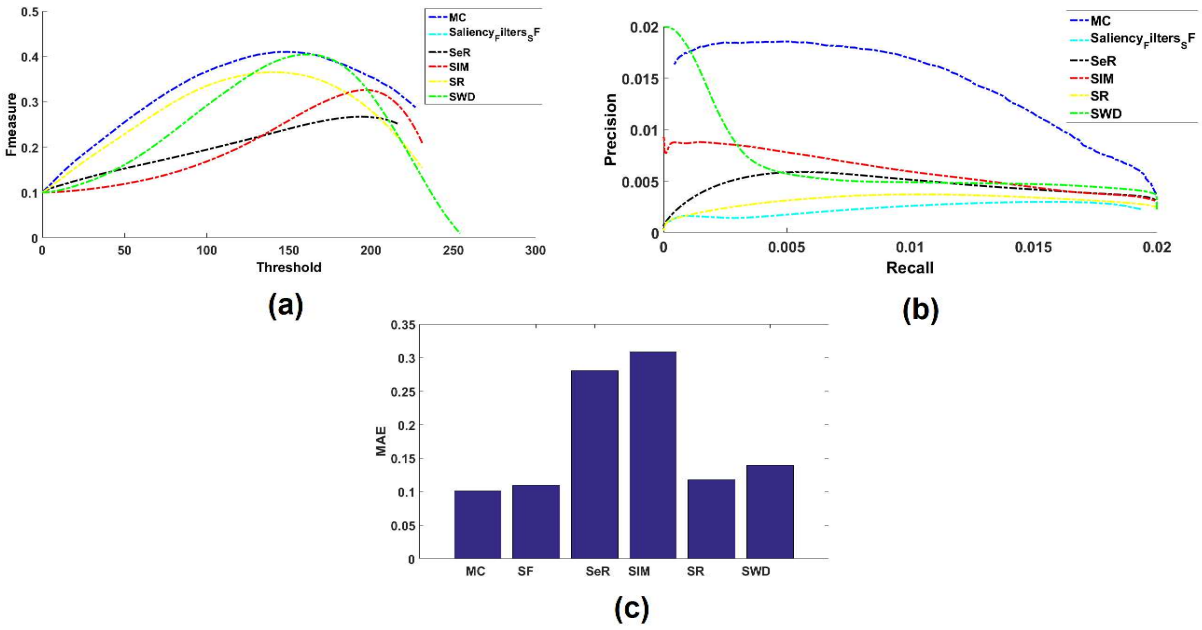


FIGURE 4.6: Illustration of segmentation methods results on Davis dataset given by spatial methods. (a) F-measure. (b) Recall and Precision. (c) MAE.

Method	MC	SF	SIM	SWD	SeR	SR
bear	0.7312	0.1842	0.3528	0.3009	0.3008	0.2190
blackswan	0.2883	0.3348	0.1416	0.3220	0.1471	0.1794
bmx-bumps	0.2055	0.0182	0.3243	0.1561	0.3103	0.4861

bmx-trees	**0.2692**	**0.0231**	**0.3716**	**0.3329**	**0.0673**	**0.3310**
boat	**0.1890**	**0.1380**	**0.4362**	**0.4262**	**0.4692**	**0.3048**
breakdance	**0.6071**	**0.3526**	**0.1915**	**0.3892**	**0.2987**	**0.2144**
breakdance-flare	**0.5946**	**0.4557**	**0.3178**	**0.6513**	**0.2637**	**0.4077**
bus	**0.8339**	**0.5742**	**0.4516**	**0.5860**	**0.2744**	**0.6177**
camel	**0.6917**	**0.1795**	**0.2503**	**0.4902**	**0.1857**	**0.3609**
car-roundabout	**0.6994**	**0.4821**	**0.2710**	**0.4942**	**0.2071**	**0.2508**
car-shadow	**0.5362**	**0.2267**	**0.5459**	**0.7189**	**0.3160**	**0.6364**
car-turn	**0.2706**	**0.1021**	**0.4215**	**0.2537**	**0.2880**	**0.7253**
cows	**0.6326**	**0.2099**	**0.4815**	**0.6825**	**0.3114**	**0.5130**
dance-jump	**0.3417**	**0.4303**	**0.1694**	**0.3850**	**0.1997**	**0.1295**
dance-twirl	**0.2807**	**0.1073**	**0.2230**	**0.4083**	**0.2030**	**0.2050**
dog	**0.5744**	**0.8037**	**0.4196**	**0.4869**	**0.3630**	**0.4523**

dog-agility	**0.2252**	**0.1849**	**0.1471**	**0.2109**	**0.1463**	**0.1658**
drift-chicane	**0.0493**	**0.0565**	**0.5655**	**0.5127**	**0.2236**	**0.1906**
drift-straight	**0.4555**	**0.1643**	**0.2746**	**0.2261**	**0.3597**	**0.2102**
drift-turn	**0.4225**	**0.3054**	**0.5209**	**0.4747**	**0.6104**	**0.4933**
elephant	**0.5627**	**0.2984**	**0.1694**	**0.4543**	**0.1698**	**0.1684**
flamingo	**0.5341**	**0.3123**	**0.2359**	**0.4853**	**0.2041**	**0.1746**
goat	**0.4429**	**0.1078**	**0.0954**	**0.3108**	**0.0861**	**0.1798**
hike	**0.5109**	**0.0690**	**0.2485**	**0.5490**	**0.1476**	**0.4717**
hockey	**0.6058**	**0.0608**	**0.3726**	**0.4734**	**0.1712**	**0.4339**
horsejump-high	**0.5915**	**0.3436**	**0.3654**	**0.5183**	**0.2143**	**0.5009**
horsejump-low	**0.3578**	**0.1694**	**0.2390**	**0.3934**	**0.2007**	**0.5894**
kite-surf	**0.4736**	**0.0175**	**0.7135**	**0.4176**	**0.5765**	**0.5805**
kite-walk	**0.8422**	**0.6399**	**0.7116**	**0.7282**	**0.6975**	**0.5605**

libby	**0.1876**	**0.0607**	**0.1651**	**0.3035**	**0.1233**	**0.2197**
lucia	**0.6650**	**0.0999**	**0.4536**	**0.5703**	**0.4203**	**0.5032**
mallard-fly	**0.4532**	**0.0406**	**0.2081**	**0.1329**	**0.4114**	**0.2253**
mallard-water	**0.4866**	**0.0568**	**0.2028**	**0.4574**	**0.0925**	**0.1986**
motocross-bumps	**0.3611**	**0.3395**	**0.6676**	**0.3566**	**0.6791**	**0.6531**
motocross-jump	**0.5010**	**0.3993**	**0.5218**	**0.5869**	**0.5904**	**0.5751**
motorbike	**0.1597**	**0.0676**	**0.5399**	**0.2934**	**0.4497**	**0.6127**
paragliding	**0.6448**	**0.6881**	**0.6385**	**0.6160**	**0.5000**	**0.8165**
paragliding-launch	**0.8094**	**0.5140**	**0.6510**	**0.7395**	**0.5690**	**0.6784**
parkour	**0.4669**	**0.3845**	**0.3172**	**0.3551**	**0.1441**	**0.3492**
rhino	**0.6894**	**0.4458**	**0.2011**	**0.4945**	**0.2001**	**0.2035**
rollerblade	**0.1772**	**0.0400**	**0.2236**	**0.1429**	**0.2363**	**0.6286**
scooter-black	**0.3437**	**0.3269**	**0.1703**	**0.3702**	**0.1428**	**0.2096**

scooter-gray	**0.2570**	**0.1009**	**0.3005**	**0.2891**	**0.2261**	**0.5661**
soapbox	**0.4948**	**0.2137**	**0.5248**	**0.5225**	**0.4053**	**0.3914**
soccerball	**0.6711**	**0.2972**	**0.3629**	**0.1039**	**0.1497**	**0.1967**
stroller	**0.4892**	**0.3628**	**0.4292**	**0.5752**	**0.2905**	**0.4007**
surf	**0.5928**	**0.5757**	**0.6539**	**0.5656**	**0.8423**	**0.7200**
swing	**0.7281**	**0.4380**	**0.3960**	**0.6785**	**0.2540**	**0.6015**
tennis	**0.5322**	**0.0314**	**0.6394**	**0.5085**	**0.5482**	**0.5971**
train	**0.5727**	**0.5407**	**0.5293**	**0.4636**	**0.5557**	**0.3916**
F-measure	**0.4102**	**0.2086**	**0.3262**	**0.4044**	**0.2671**	**0.3653**

TABLE 4.4: The best results of F-measure for spatial methods on Davis dataset.

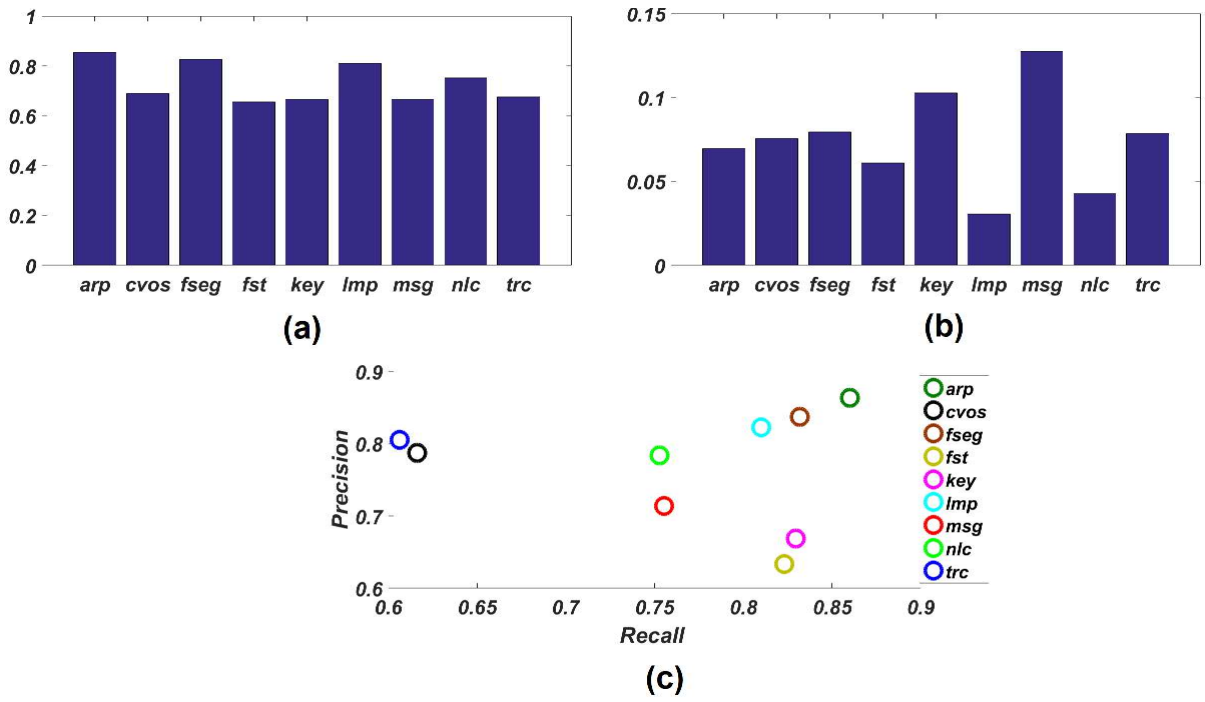


FIGURE 4.7: Illustration of segmentation methods results on Davis dataset given by spatiotemporal methods. (a) F-measure. (b) MAE. (c) Recall and Precision.

Method	arp	cvos	fseg	fst	key	lmp	msg	nlc	trc
bear	0.95	0.94	0.95	0.95	0.92	0.89	0.94	0.94	0.96
blackswan	0.95	0.48	0.90	0.89	0.88	0.69	0.59	0.91	0.64
bmx-bumps	0.58	0.60	0.16	0.22	0.24	0.62	0.53	0.76	0.65
bmx-trees	0.62	0.53	0.65	0.19	0.22	0.69	0.50	0.53	0.47

boat	**0.51**	**0.07**	**0.76**	**0.43**	**0.08**	**0.62**	**0.18**	**0.01**	**0.16**
breakdance	**0.87**	**0.49**	**0.61**	**0.54**	**0.68**	**0.66**	**0.55**	**0.84**	**0.35**
breakdance-flare	**0.93**	**0.67**	**0.84**	**0.71**	**0.68**	**0.90**	**0.43**	**0.93**	**0.57**
bus	**0.94**	**0.88**	**0.92**	**0.92**	**0.84**	**0.92**	**0.96**	**0.82**	**0.90**
camel	**0.96**	**0.91**	**0.93**	**0.66**	**0.82**	**0.92**	**0.91**	**0.88**	**0.92**
car-roundabout	**0.85**	**0.91**	**0.95**	**0.87**	**0.67**	**0.89**	**0.58**	**0.65**	**0.80**
car-shadow	**0.87**	**0.82**	**0.95**	**0.79**	**0.66**	**0.88**	**0.91**	**0.81**	**0.74**
car-turn	**0.90**	**0.86**	**0.95**	**0.89**	**0.88**	**0.86**	**0.49**	**0.88**	**0.77**
cows	**0.97**	**0.79**	**0.94**	**0.88**	**0.70**	**0.93**	**0.92**	**0.96**	**0.95**
dance-jump	**0.87**	**0.71**	**0.79**	**0.68**	**0.84**	**0.70**	**0.08**	**0.82**	**0.67**

dance-twirl	**0.89**	**0.70**	**0.89**	**0.55**	**0.42**	**0.83**	**0.76**	**0.66**	**0.71**
dog	**0.88**	**0.62**	**0.94**	**0.85**	**0.86**	**0.92**	**0.73**	**0.92**	**0.93**
dog-agility	**0.38**	**0.32**	**0.86**	**0.44**	**0.17**	**0.58**	**0.14**	**0.73**	**0.38**
drift-chicane	**0.89**	**0.45**	**0.62**	**0.80**	**0.20**	**0.76**	**0.90**	**0.38**	**0.81**
drift-straight	**0.69**	**0.62**	**0.84**	**0.81**	**0.22**	**0.77**	**0.72**	**0.62**	**0.59**
drift-turn	**0.81**	**0.81**	**0.93**	**0.73**	**0.31**	**0.74**	**0.66**	**0.39**	**0.51**
elephant	**0.90**	**0.75**	**0.94**	**0.88**	**0.74**	**0.90**	**0.89**	**0.58**	**0.92**
flamingo	**0.89**	**0.84**	**0.86**	**0.94**	**0.89**	**0.80**	**0.94**	**0.63**	**0.90**
goat	**0.87**	**0.10**	**0.91**	**0.62**	**0.76**	**0.89**	**0.88**	**0.02**	**0.92**
hike	**0.96**	**0.94**	**0.91**	**0.94**	**0.96**	**0.94**	**0.85**	**0.96**	**0.91**
hockey	**0.89**	**0.89**	**0.85**	**0.54**	**0.60**	**0.92**	**0.77**	**0.87**	**0.76**

horsejump-high	**0.92**	**0.92**	**0.82**	**0.65**	**0.39**	**0.90**	**0.82**	**0.90**	**0.73**
horsejump-low	**0.89**	**0.87**	**0.85**	**0.63**	**0.76**	**0.86**	**0.83**	**0.71**	**0.84**
kite-surf	**0.83**	**0.64**	**0.45**	**0.25**	**0.71**	**0.60**	**0.50**	**0.54**	**0.62**
kite-walk	**0.86**	**0.63**	**0.78**	**0.73**	**0.53**	**0.90**	**0.74**	**0.91**	**0.07**
libby	**0.87**	**0.57**	**0.74**	**0.83**	**0.82**	**0.86**	**0.09**	**0.84**	**0.33**
lucia	**0.95**	**0.94**	**0.89**	**0.75**	**0.94**	**0.94**	**0.75**	**0.93**	**0.88**
mallard-fly	**0.61**	**0.60**	**0.81**	**0.60**	**0.78**	**0.60**	**0.04**	**0.81**	**0.53**
mallard-water	**0.60**	**0.28**	**0.89**	**0.11**	**0.86**	**0.42**	**0.06**	**0.88**	**0.13**
motocross-bumps	**0.88**	**0.64**	**0.88**	**0.49**	**0.87**	**0.82**	**0.51**	**0.66**	**0.55**
motocross-jump	**0.89**	**0.45**	**0.84**	**0.69**	**0.31**	**0.77**	**0.54**	**0.44**	**0.45**

motorbike	**0.80**	**0.56**	**0.49**	**0.64**	**0.58**	**0.84**	**0.79**	**0.77**	**0.77**
paragliding	**0.96**	**0.96**	**0.80**	**0.69**	**0.96**	**0.95**	**0.98**	**0.96**	**0.95**
paragliding-launch	**0.87**	**0.86**	**0.82**	**0.69**	**0.70**	**0.87**	**0.80**	**0.85**	**0.82**
parkour	**0.94**	**0.47**	**0.88**	**0.61**	**0.36**	**0.81**	**0.64**	**0.94**	**0.70**
rhino	**0.96**	**0.79**	**0.94**	**0.86**	**0.77**	**0.93**	**0.96**	**0.85**	**0.95**
rollerblade	**0.92**	**0.71**	**0.83**	**0.39**	**0.63**	**0.81**	**0.88**	**0.89**	**0.81**
scooter-black	**0.83**	**0.82**	**0.84**	**0.58**	**0.61**	**0.80**	**0.50**	**0.36**	**0.52**
scooter-gray	**0.83**	**0.68**	**0.85**	**0.44**	**0.53**	**0.83**	**0.63**	**0.68**	**0.63**
soapbox	**0.89**	**0.88**	**0.85**	**0.50**	**0.88**	**0.85**	**0.55**	**0.83**	**0.57**
soccerball	**0.95**	**0.52**	**0.90**	**0.94**	**0.96**	**0.77**	**0.71**	**0.92**	**0.69**
stroller	**0.92**	**0.79**	**0.82**	**0.64**	**0.86**	**0.73**	**0.78**	**0.92**	**0.84**

surf	**0.98**	**0.64**	**0.96**	**0.59**	**0.93**	**0.70**	**0.76**	**0.86**	**0.32**
swing	**0.93**	**0.84**	**0.87**	**0.43**	**0.88**	**0.91**	**0.86**	**0.92**	**0.74**
tennis	**0.91**	**0.78**	**0.83**	**0.44**	**0.77**	**0.88**	**0.84**	**0.92**	**0.52**
train	**0.95**	**0.94**	**0.89**	**0.89**	**0.64**	**0.91**	**0.92**	**0.85**	**0.93**
F-measure	**0.855**	**0.690**	**0.826**	**0.655**	**0.665**	**0.810**	**0.666**	**0.752**	**0.675**

TABLE 4.5: The obtained F-measure results for spatiotemporal methods per video.

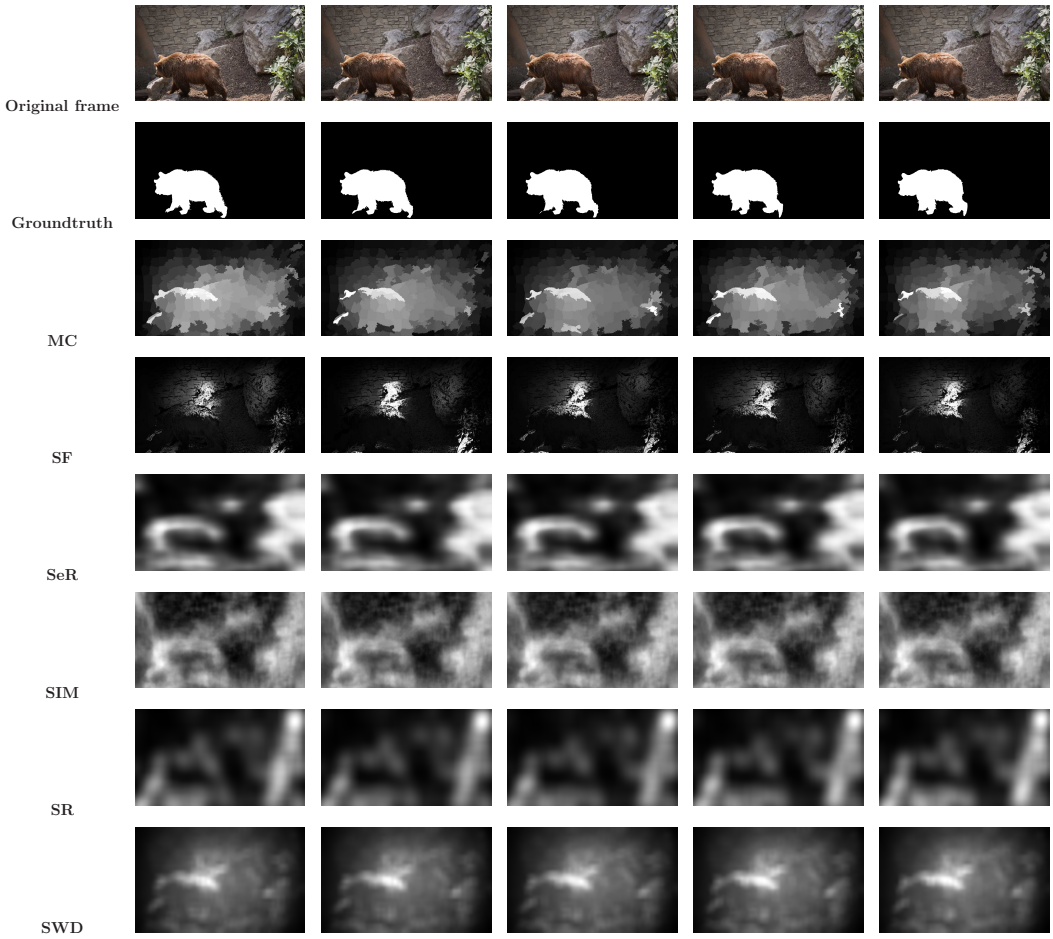


FIGURE 4.8: Visual comparison of methods with GT on Davis (bear video).

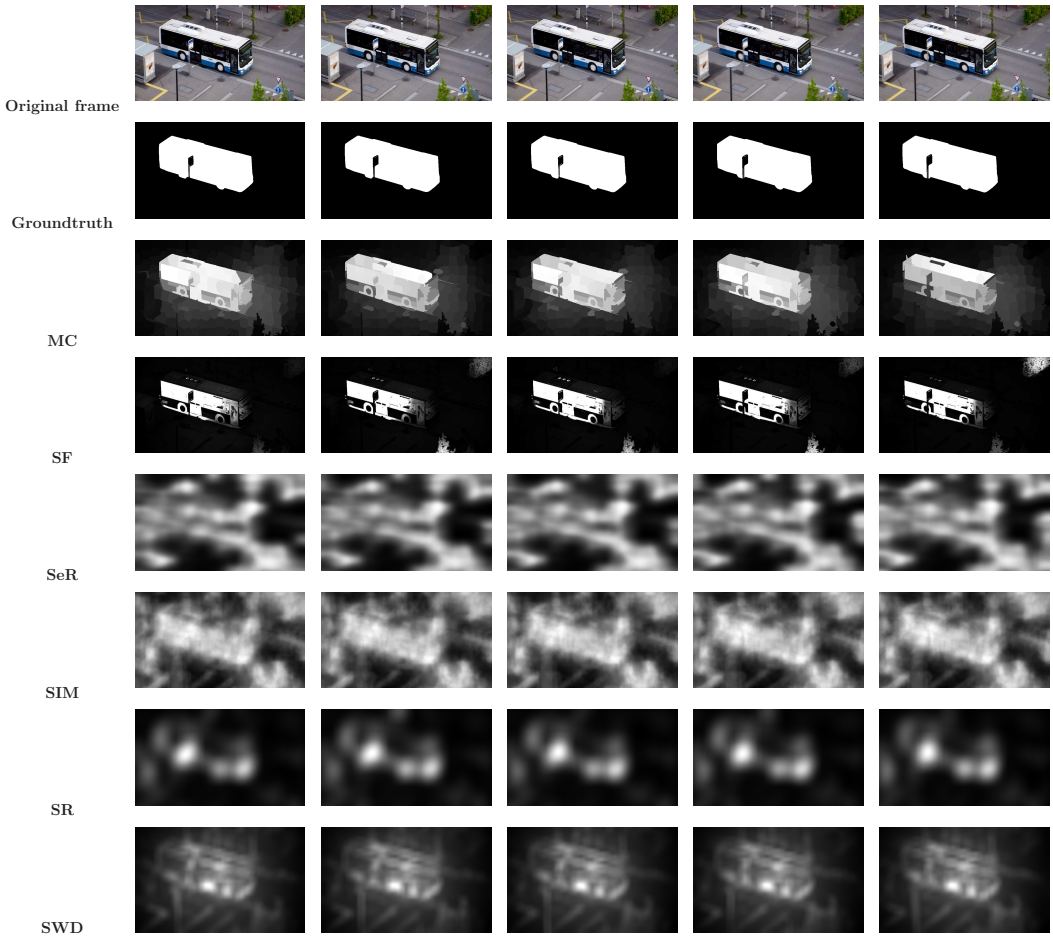


FIGURE 4.9: Visual comparison of methods with GT on Davis(Bus video)

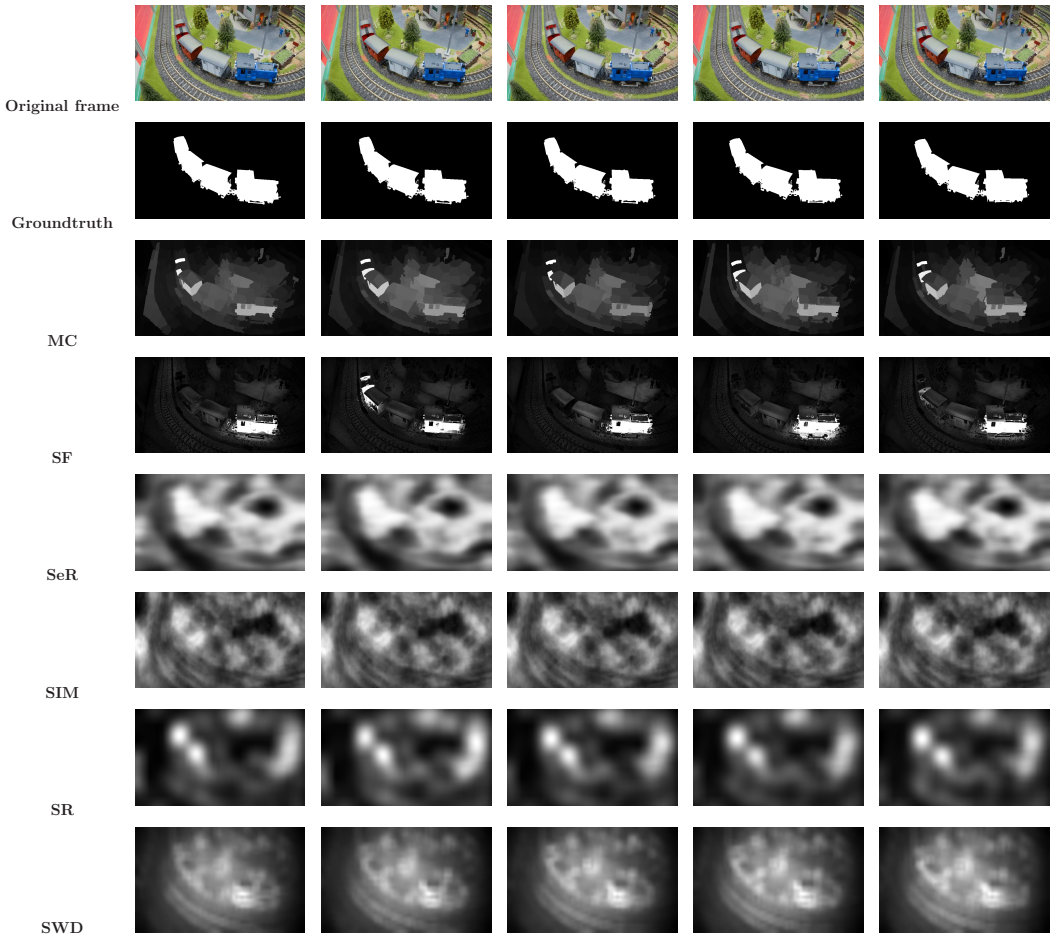


FIGURE 4.10: Visual comparison of methods with GT on Davis(train video)

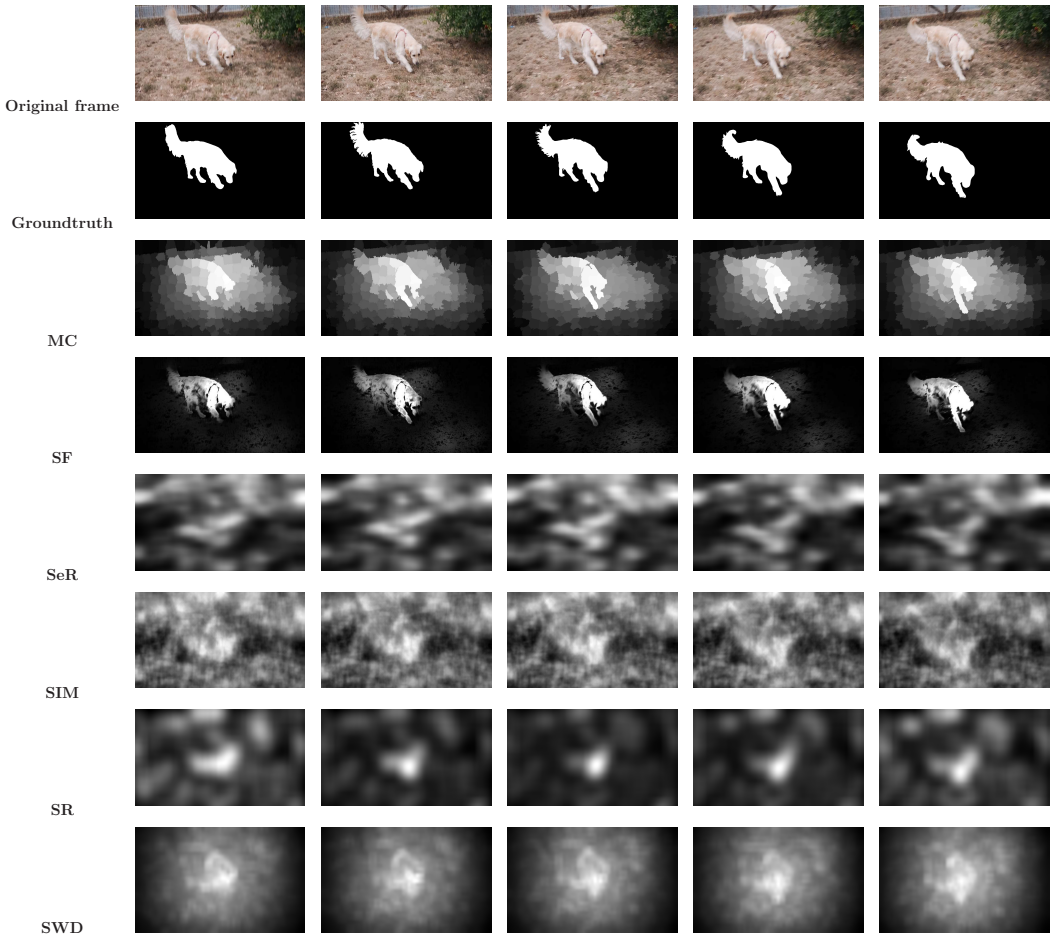


FIGURE 4.11: Visual comparison of methods with GT on Davis(dog video)

As we can see in the previous Figures. 4.8, 4.9, 4.11 and 4.10, we can observe that there is a difference between methods and as we said in the previous section (segtrack dataset) and because those methods use only spatial cues, the results are not so good and lack precision. This is due to the absence of motion cues which is robust to estimate the objects in motion especially in complex scenes.

4.2.2.1 Comparison on computational time

We calculate the computational time for each method for a typical 480×854 frame. The results are shown in the Table 4.6.

Method	**MC**	**SF**	**SIM**	**SWD**	**SeR**	**SR**
Computational time per frame	6.82 s	1.19 s	4.94 s	1.08 s	1.71 s	0.48 s

TABLE 4.6: The observed computational time of the compared methods on the Davis dataset.

4.3 Conclusion

In this chapter, we conducted an experimental study on the proposed approach along with some video segmentation methods. We used three metrics to measure accurately the performance of video segmentation methods and make a comparison between them to extract the characteristics and the performance of each one. We have observed that using spatiotemporal generation of object proposals can improve the video segmentation results.

5

Conclusions and perspectives

Video segmentation is a hot topic in computer vision that has been studied for many years and opened the way for researchers to compete among themselves to develop effective and practical algorithms and methods in this area, which is why we have chosen to be the subject of our thesis.

In this thesis, Firstly, we studied several concepts about video segmentation such as features (superpixels, optical flow ...etc), refinement methods and other related concepts to master the subject. After that, we choose a recent method in video segmentation to support these theoretical concepts by something practical and give more examples to extend thesis content, and the reason that makes us

select it, is that this method is based on object proposals where they give us prior information about the object because they are more "object-like" which are expected to cover all objects in an image. Another adventage of using object proposals in segmentation methods is to reduce the search space of objects in an image/video. Although the process to generate these proposal segments is very expensive where a single image required about 2-7 muinutes, the reason that make us also select this method is that it uses various saliency cues whether spatial or temporal and that helps to give a well initial estimation.

Although this method is robust to detect objects, but it have a weakness point which is using only spatial information to generate object proposals. However, in some video using only spatial information is not effective to detect moving object. This is what led us to think about a way to solve this problem as we proposed a method that generate spatiotemporal object proposals.

To evaluate the method proposed approach, we have used several evaluation metrics to illustre the effectiveness of each approach and give a quantitative and qualitative comparison.

At this stage, we can say that we have realized most of the work requested, nevertheless, there are still prospects to enrich this project among them the run-time and testing the proposed method with all the possible values of the used parameters.

References

[1] R. Achanta, A. Shaji, and K. Smith. SLIC Superpixels Compared to State of the art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274-2282, 2012.

[2] A. Blake, P. Kohli and C. Rother. *Markov Random Fields for Vision and Image Processing*. Springer Science, 2011.

[3] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. *IEEE ICCV*, 2001.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *ECCV*, 282-295, 2010.

[5] T. Brox and J. Malik, Large displacement optical flow: Descriptor matching in variational motion estimation, IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 3, pp. 500513, Mar. 2011.

[6] L.Duan, Ch.Wu, J.Miao. Visual saliency detection by spatially weighted dissimilarity. *IEEE CVPR*, 2011.

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. IEEE TPAMI, 37(3), 569-582, 2015.

[8] I. Endres and D. Hoiem. Category-Independent Object Proposals with Diverse Ranking. *IEEE TPAMI*, 36(2), 222-234, 2014.

[9] F.Perazzi, Ph.Krhenbl, Y.Pritch, A.Hornung. Saliency Filters: Contrast Based Filtering for Salient Region Detection. *IEEE CVPR*, Providence, Rhode Island, USA, June 16-21, 2012.

[10] F. Guo, W. Wang, and Jianbing Shen. Video Saliency Detection Using Object Proposals. *IEEE Transactions on Cybernetics*, 2017.

[11] H. Guo. A Simple Algorithm for Fitting a Gaussian Function. *IEEE Signal Processing Magazine*, 28(5):134-137, 2011.

[12] X.Hou, L.Zhang.Saliency detection: A spectral residual approach. *IEEE CVPR*, 2007.

[13] B.Jiang, L.Zhang, H.Lu, Ch.Yang, M.Yang.Saliency Detection via Absorbing Markov Chain. *IEEE ICCV*, 2013.

[14] N.Murray, Maria Vanrell, Xavier Otazu and C. Alejandro Prraga. Saliency Estimation Using a Non-Parametric Low-Level Vision Model. *CVPR 2011.*

[15] S.D. Jain, B. Xiong, K. Grauman. Fusionseg Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *IEEE CVPR*, 2017.

[16] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure ground segments. *IEEE ICCV*, 2192-2199, 2013.

[17] Z. Liu, J. Li, L. Ye. Saliency Detection for Unconstrained Videos Using Superpixel-level Graph and Spatiotemporal Propagation. *IEEE TCSVT*, 2527 - 2542, 2016.

[18] T. B. Moeslund. *Introduction to Video and Image Processing, Building Real Systems and Applications*. Springer-Verlag London, 2012.

[19] K. N. Ngan and H. Li. *Video Segmentation and Its Applications*. Springer Science, 2011.

[20] A. Papazoglou, V.Ferrari. Object segmentation by long term analysis of point trajectories. *IEEE ICCV*, 1777-1784, 2013.

[21] A. Papazoglou, V. Ferrari. Fast object segmentation in unconstrained video. *IEEE ICCV*, 2013.

[22] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. *IEEE CVPR*, 2016.

[23] M. A. Rodrguez. *Implementation of Gaussian Mexture Models in .Net technology for automatic speech recognition*. PhD Thesis. The University of Science and Technology AGH in Krakow, 2011.

[24] H.Seo, P.Milanfar. Static and space-time visual saliency detection by self-resemblance. *J.ofVis.*, 2009.

[25] S.-g. Wei, L. Yang, Z. Chen and Z.-f. Liu. Motion Detection Based on Optical Flow and Self-adaptive Threshold Segmentation. *Procedia Engineering*, 15:3471-3476, 2011.

[26] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. *ECCV*, 2010.

[27] W. Wang, J. Shen, and F. Porikli. Saliency aware geodesic video object segmentation. *IEEE CVPR*, 3395-3402, 2015.

[28] W. Wang, J. Shen, and L. Shao. Consistent Video Saliency Using Local Gradient Flow Optimization and Global Refinement. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 24(11):4185-4196, 2015.

[29] A. Yilmaz. *Object Tracking and Activity Recognition in Video Acquired Using Mobile Cameras*. PhD Thesis. The University of Central Florida, Orlando, Florida, 2004.

[30] D. Zhang, O. Javed and M. Shah. Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions. *CVPR*, 628-635, 2013.

[31] W. Zhu and S. Liang and Y. Wei and J. Sun. Saliency Optimization from Robust Background Detection. *CVPR*, 2814-2821, 2014.

[32] W. T. Freeman, K. Tanaka, J. Ohta and K. Kyuma. Computer Vision for Computer Games. Intl Conf. on Automatic Face and Gesture Recognition, 100-105, 1996.

[33] N. Marki, F. Perazzi and al. Bilateral Space Video Segmentation. *CVPR*, 743-751, 2016.

[34] X. Wang, R. Hansch, and al. Comparison of Different Color Spaces for Image Segmentation using Graph-cut. *VISAPP*, 301-308, 2014.

[35] S. Wei, L. Yang, and al. Motion Detection Based on Optical Flow and Self-adaptive Threshold Segmentation. *Elsevier*, 3471-3476, 2011.

[36] S. Ardeshir, K.Collins-Sibley, and M. Shah. Geo-semantic segmentation. *CVPR*, 2015.

[37] *https* : *//fr.dreamstime.com/image − libre − de − droits − jouer − les − jeux − interactifs − avec − le − kinect − xbox − image*35169296 (retrieved as on 21/06/2018)

[38] *https* : *//davischallenge.org/davis*2016/*soa_compare.html* (retrieved as on 30/05/2018)

[39] *http* : *//obligement.free.fr/articles/traitement_images_*3.*php* (retrieved as on 30/05/2018)

[40] *https* : *//docs.opencast.org/r/*4.*x/admin/modules/videosegmentation/* (retrieved as on 11/04/2018)

[41] *http* : *//vision.seecs.edu.pk/abnormal − event − detection/* (retrieved as on 22/06/2018)

[42] *https* : *//software.intel.com/en − us/articles/object − detection − on − drone − videos − using − caffe − framework* (retrieved as on 22/06/2018)

[43] *http* : *//www.uio.no/studier/emner/matnat/math/MAT* − *INF*1100/*h*08/*kompendiet/images.pdf* (retrieved as on 22/06/2018)

[44] Jacob Rus, *TheCIEL*∗*a*∗*b*∗*colorspace*, *onlyshowingcolorswhichcanberepresentedwithinthega*
Labcolorspace.svg, 2007.

[45] Kevin Smith, Pedro Quelhas, and Daniel Gatica-Perez. Detecting Abandoned Luggage Items in a Public Space. 2006.

[46] *https* : *//www.rtbf.be/tendance/techno/detail*$_c$*lap − de − fin − pour − le − kinect − de − microsoft?id* = 9802014 (retrieved as on 24/06/2018)

[47] *http* : *//www.gadgetguy.com.au/kinect−to−connect−kids−with−cookie− monster − and − cubs/* (retrieved as on 24/06/2018)

[48] A Iske, J Levesley. *Multilevel scattered data approximation by adaptive domain decomposition*. Springer, 2005.