Unsupervised Video Object Segmentation using Conditional Random Fields

By
Asma Hamza Bhatti
NUST201464104MSEECS61314F

Supervisor
Dr. Asad Anwar Butt
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MS CS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(April 2016)

Approval

It is certified that the contents and form of the thesis entitled "**Unsupervised Video Object Segmentation using Conditional Random Fields**" submitted by **Asma Hamza Bhatti** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Asad Anwar Butt**
Signature: ________________

Date: ________________

Committee Member 1: **Dr. Anis ur Rahman**

Signature: ________________
Date: ________________

Committee Member 2: **Dr. Omar Arif**

Signature: ________________
Date: ________________

Committee Member 3: **Dr. Muhammad Moazam Fraz**

Signature: ________________
Date: ________________

Dedicated to my husband

Mehran Bhatti

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Asma Hamza Bhatti**
Signature: ________________

Acknowledgment

This thesis would not have been possible without the continuous support of my supervisor Dr. Asad Anwar Butt and co-supervisor Dr. Anis ur Rahman. I would also like to thank my committee members, Dr. Omar Arif and Dr. Muhammad Moazam Fraz, for their guidance and support.

I want to especially thank my family and friends for their constant moral and spiritual support.

List of Abbreviations and Acronyms

CCTV	Closed Circuit Television
CPMC	Constrained Parametric Min Cuts
CRF	Conditional Random Field
DAG	Directed Acyclic Graph
DP	Dynamic Programming
EM	Energy Minimization
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HVS	Human Visual System
LSC	Linear Spectral Clustering
MAP	Maximum A-Posteriori
MMR	Maximum Marginal Relevance
MRF	Markov Random Field
Prec.	Precision
Rec.	Recall
SLIC	Simple Linear Iterative Clustering
SSC	Salient Segment Chain
VOE	Video Object Extraction

List of Figures

List of Tables

Abstract

Object Segmentation is an open area of research in the field of Computer Vision. Image and video segmentation are two separate segmentation problems that aim to identify the objects of interest in images and videos respectively. While Image segmentation methods use visual cues for separating foreground objects from background, video segmentation solutions, in addition to visual cues, can use motion and other temporal cues as well. Video segmentation has uses in a wide variety of applications including scene analysis and object tracking. Human Visual System (HVS) has the ability to identify object(s) of interest from a scene. An individual's focus and attention is directed to the object(s) of interest by the visual system as soon as visual contact is made with the scene. The rest of the details go in the background and the person does not pay attention to the unnecessary details. There has been a tremendous amount of research carried out by biologists to explain the ways in which the HVS works. Additionally, a lot of efforts have been made by researchers to provide a solution that mimics the behavior of the HVS. Video Object Segmentation is a challenging problem that aims to automate the properties of the HVS in order to identify objects present within a video. The properties of the input videos can vary depending on the surrounding conditions, camera quality and object appearance and size. The effectiveness of the segmentation algorithm can depend on the properties of the video sequences. The videos can contain single or multiple objects present in few or all of the frames. These objects can be occluded, deformed and can have interactions with each other. Additionally, the videos can also contain motion blur, camera movement, slow motion and appearance change. In this work, the aim is to provide a generic segmentation solution that produces an effective result for all types of input videos. The proposed solution will be evaluated on publicly available datasets and compared against commonly used state-of-the-art solutions.

Table of Contents

Chapter 1

Introduction

1.1 Video Object Segmentation

Computer Vision is an area of Computer Science that aims to produce solutions which can help in the interpretation of real-world data, e.g. images and videos etc. The problems associated with Computer Vision include segmentation, object tracking, object recognition, image analysis and scene analysis etc. The solutions of these problems usually make use of a combination of machine learning and image processing algorithms. The automation of the identification, recognition, analysis and tracking of real-world data can save a lot of time and human effort otherwise required for the manual completion of such tasks. The introduction of high performance hardware has raised the demands for the automation of such tasks. As a result, a lot of research is being carried out in different areas of Computer Vision.

Object Segmentation is an open area of research in the field of Computer Vision that involves the identification and retrieval of objects of interests in a given dataset. Each pixel of a given image is classified as background or foreground based on the features that it possesses. A foreground pixel is usually labeled as one (1) while a background pixel is labeled as zero (0).

Image and video segmentation are two separate segmentation problems that aim to identify the objects of interest in images and videos respectively. A number of different features can be used for the labeling of pixels. These include color, saliency score, optical flow and spatial co-ordinates etc. Different techniques make use of different features in different ways for the extraction of objects. Image segmentation methods (Bai and Wang, 2014; Achanta et al., 2008; Li and Chen, 2015; Li et al., 2012) usually use visual and spatial cues for the extraction of interesting objects from a given image. On the other hand, video segmentation techniques (Fukuchi et al., 2009; Li et al., 2013b; Banica et al., 2013; Wang et al., 2015; Shi, 2012), in addition to visual and spatial cues, also make use of temporal cues for the correct identification of objects.

1.2 Motivation

Video segmentation has uses in a wide variety of applications. The segmentation results can be used as an input for saliency detection (Xu et al., 2013) , multiple object tracking (Milan et al., 2015), object recognition, motion analysis and scene analysis. Additionally, video segmentation has applications in the field of surveillance (Luque et al., 2008). The segmentation algorithms can be incorporated into Closed-Circuit Television (CCTV) cameras to provide automated segmentation of unusual targets in order to improve the security systems of public places.

The Human Visual System (HVS) has the ability to identify object(s) of interest from a scene. An individual's focus and attention is directed to the object(s) of interest by the visual system as soon as visual contact is made with the scene. The rest of the details go in the background and the person does not pay attention to the unnecessary details. There has been a tremendous amount of research carried out by biologists to explain the ways in which the HVS works. Additionally, a lot of efforts have been made by researchers to provide a solution that mimics the behavior of the HVS. Such solutions make use of eye movement patterns and attention models to imitate the visual system of humans (Fukuchi et al., 2009).

Video Object Segmentation is a challenging problem that aims to automate the properties of the HVS in order to identify objects present within a video. The properties of the input videos can vary depending on the surrounding conditions, camera quality and object appearance and size. The effectiveness of the segmentation algorithm can depend on the properties of the video sequences. The videos can contain single or multiple objects present in few or all of the frames. These objects can be occluded, deformed and can have interactions with each other. Additionally, the videos can also contain motion blur, camera movement, slow motion and appearance changes.

Various unsupervised video object segmentation techniques have been proposed in the past. Each solution has presented a different algorithm for the identification of object(s) of interest in a given video sequence. However, all the existing solutions have problems that limit the effectiveness of the results. One problem is the error caused due to similarities between the features of neighboring foreground and background pixels. If neighboring foreground and background pixels have similar color, motion and saliency scores, there is a high possibility that both will be labeled as foreground or background causing the mis-segmentation rate (or error segmentation rate) to rise. Additionally, camera movement and motion blur can reduce the accuracy of the results. Also, there are solutions with high tolerance for errors but they perform segmentation of single objects only. Hence, there is a need to propose a solution that overcomes the aforementioned problems and lowers the mis-segmentation rate.

1.3 Proposed Methodology

The focus of the research is to provide an unsupervised video object segmentation method that effectively identifies all of the objects in a given video sequence. The different methodologies that can be employed for solving the segmentation problem are explained. Additionally, the existing solutions already presented for solving the given problem are discussed outlining the benefits and drawbacks of each method. A solution is proposed that improves on the results of the existing techniques. The solution is evaluated on publicly available video datasets using average pixel error rate, precision, recall and F1-score as the evaluation measures. The results of the proposed solution are compared with the state-of-the-art methods to provide a quantitative difference between the proposed and existing solutions.

1.4 Contributions

The thesis makes the following contributions:

- The extension of superpixel segmentation technique originally proposed for images to perform frame by frame spatio-temporal oversegmentation of videos.

- A foreground separation model that uses multiple features to produce segmentation maps by introducing thresholds defined based on differences between foreground and background segments.

- The construction of a Conditional Random Field (CRF) and introduction of a potential function to show the relationships between different pixels in a given frame and solving it to get the final segmentation results.

1.5 Organization of Thesis

The rest of the thesis is organized as follows: Chapter 2 presents a literature review of the existing techniques for video object segmentation. In Chapter 3, the proposed solution is discussed in detail. Chapter 4 provides experimental results of the solution on publicly available datasets and comparison with state-of-the-art methods. Finally, Chapter 5 concludes the thesis and gives possible future directions.

Chapter 2

Literature Review

This chapter discusses some of the existing solutions for performing unsupervised video object segmentation which involves the use of different types of graphs. The methods construct graphs to show the relationships between different parts of a video. Individual pixels or a set of closely related pixels can be used as nodes of the graph while edges between the nodes are responsible for showing the weighted similarity between them. Some of the commonly used graph based methods are discussed. The algorithms proposed by different researchers are discussed along with the experimental results of each technique. The benefits and drawbacks of each solution are also stated.

2.1 Markov Random Field

Markov Random Field is a type of undirected graph used to model joint probability distributions. The nodes of the graph can be represented through pixels/superpixels while edges between nodes show the dependencies between specific pixels/superpixels. In Markov Random Fields, each node has a separate unary potential for each possible output. Hence, in case of binary segmentation, each node has two unary potentials usually obtained by calculating negative log likelihood of each label. Similarly, each edge represents a pairwise potential between two nodes and is calculated using features of both the nodes. In Markov Random Fields, the unary potential of a given node is only dependent on the node and the respective label itself. Similarly, the pairwise potential of two nodes is only dependent on those two observations. The graph can be solved using energy minimization techniques to obtain the final segmentation results.

The solution proposed by (Fukuchi et al., 2009) uses saliency and Markov Random Fields to solve the segmentation problem. Figure 2.1 shows the framework of the proposed solution. It introduces a saliency based human visual attention model for obtaining the visual attention density for each frame of the input sequence. A final eye focusing density map is obtained by combining density from the saliency map and eye movement patterns of individuals acquired manually. A Markov Random Field (MRF) is built where the density values from the visual attention model are used to estimate the priors of foreground/background as well as the feature likelihoods. This technique uses priors of previous frames while calculating the priors of the next frames. The priors for each frame are updated using Kalman filter as the combination of previous segmentation results and the original priors for the current frame. Salient regions are obtained from the Markov Random Field by maximum a posteriori (MAP) estimation using graph cuts solved by an Energy Minimization (EM) algorithm. The solution is evaluated over ten different sequences but the quantitative results are not made available. Nevertheless, the method provides satisfactory results for simple input sequences but fails when the similarity between color and saliency values of the neighboring foreground and background pixels increases.

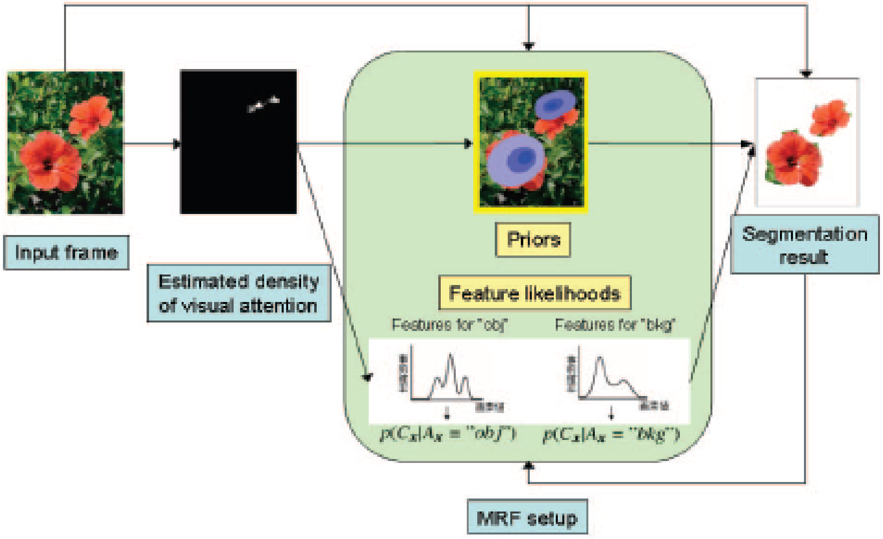

Figure 2.1: Figure taken from (Fukuchi et al., 2009) outlining the proposed framework

(Lee et al., 2011) propose a solution that uses Markov Random Field construction in the last step of implementation. Fig 2.2 shows the overview of the proposed technique. Initially, the algorithm proposed by (Endres and

Hoiem, 2014) is used to obtain a set of possible objects for each frame of the video. The objects are then scored according to their relative movement and difference in shape and color to the rest of the frame to determine the likelihood of the object proposal belonging to the foreground. Then, set of objects with similar region scores across frames are clustered together to form key-segment hypotheses representing a single object across frames. The key-segment hypotheses are used to construct a series of space time Markov Random fields across frames showing the relationship between pixels in terms of shape, color and motion used in an energy function. The energy function is minimized using graph cuts to get the final segmentation. The method is evaluated on a single dataset using segmentation error as the evaluation measure. It fails to produce satisfactory results for videos with motion blur, camera movement and foreground/background color similarity. Additionally, the technique yields high error rates for videos having multiple objects.

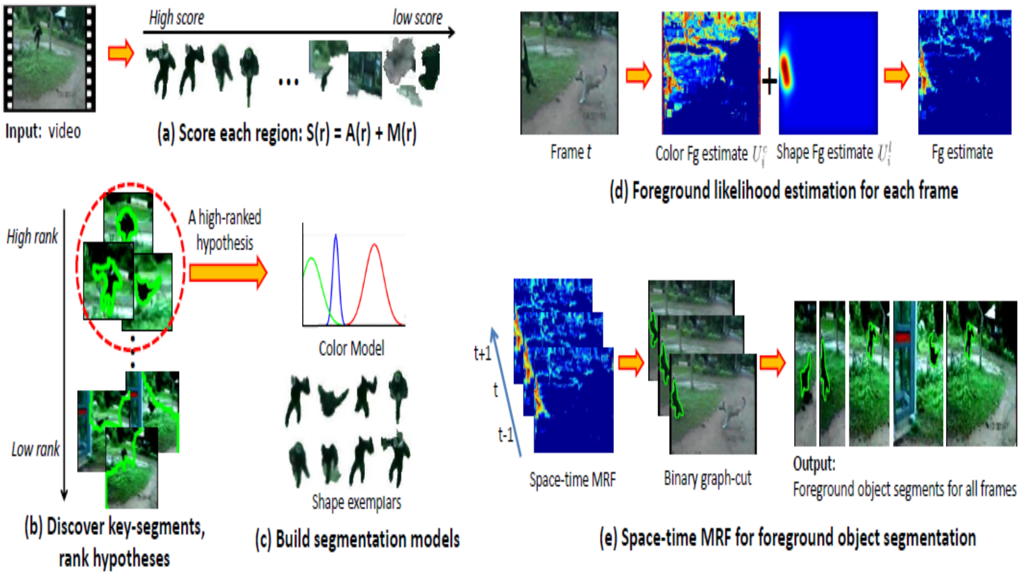
Input: video
High score
low score
(a) Score each region: S(r) = A(r) + M(r)
Frame t
Color Fg estimate U_i^c
Shape Fg estimate U_i^l
Fg estimate
(d) Foreground likelihood estimation for each frame
High rank
A high-ranked hypothesis
Color Model
Low rank
Shape exemplars
(b) Discover key-segments, rank hypotheses
(c) Build segmentation models
t+1
t
t-1
Space-time MRF
Binary graph-cut
Output:
Foreground object segments for all frames
(e) Space-time MRF for foreground object segmentation

Figure 2.2: Figure taken from (Lee et al., 2011) showing the proposed framework

2.2 Conditional Random Field

Conditional Random Field is a variant of Markov Random Field considered more suitable for binary classification. Unlike Markov Random Fields, the unary and pairwise potential of nodes in a Conditional Random Field can be made as a function of all the possible observations. This allows each random variable to be conditioned over a set of global observations. The Conditional Random Field can be solved using energy minimization techniques.

(Li et al., 2013b) propose a solution that constructs a Conditional Random Field to perform segmentation of video sequences. Figure 2.3 shows the model of the proposed solution. The method generalizes well for video sequences captured by moving cameras. It separates the foreground from background without treating either of them as outliers. Each frame of the video is segmented into superpixels using Turbopixels (Levinshtein et al., 2009) and a saliency score is calculated for each superpixel. The motion saliency score for each pixel is calculated using optical flow and the motion saliency image is converted into a binary image to obtain the shape information from the motion saliency map. The shape calculation process is performed for multiple resolutions making the process scale invariant. The color cues for the foreground and background parts are obtained by thresholding the shape likelihood in order to separate the foreground and background regions. Gaussian Mixture Model (GMM) tuned using Expectation Maximization (EM) algorithm is used to find the RGB distribution for each model. The color models, visual saliency score and shape likelihood are integrated into a Visual Object Extraction (VOE) model. Conditional Random Field (CRF) for the VOE model is constructed and solved using graph based energy minimization techniques. To preserve spatio-temporal consistency, a pairwise term is introduced in the CRF that uses the information of the neighboring pixels to maintain consistency. The method is evaluated on eight different video sequences and compared with state-of-the-art techniques using average mis-segmentation rate as the evaluation metric. The proposed technique is able to outperform most of the existing techniques. However, a similarity in color, shape, saliency and motion of neighboring foreground and background pixels increases the mis-segmentation rate of the method.

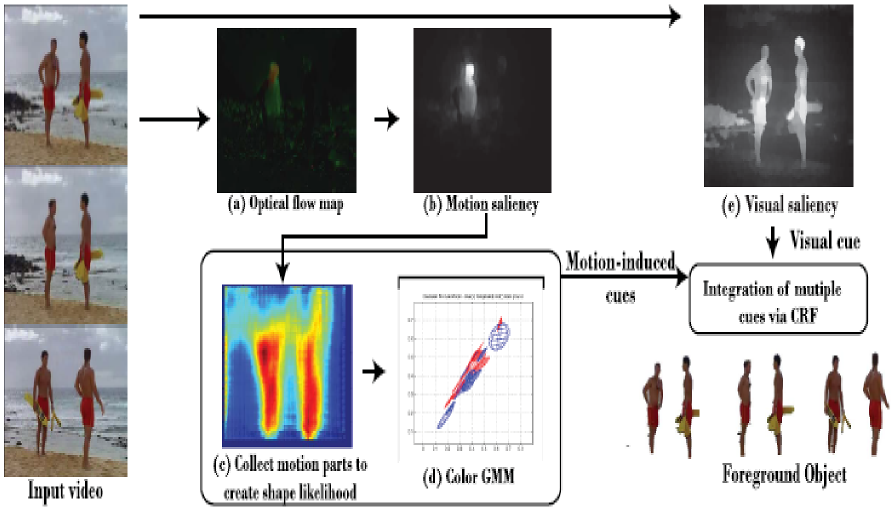

Figure 2.3: Figure taken from (W. T. Li, 2013) outlining the proposed framework

2.3 Directed Acyclic Graph

Directed Acyclic Graph (DAG) is a directed graph that does not consist of any cycles. It can be used to solve the video segmentation problem. The nodes of the graph can be represented by pixels/superpixels and edges can be used to show the association between different pixels/superpixels. It can be solved using Dynamic Programming (DP) to obtain the highest/lowest weighted path to get the segmentation solution.

(Zhang et al., 2013) present a layered Directed Acyclic Graph (DAG) based framework using spatial and temporal features to extract primary objects from a given video sequence. Figure 2.4 illustrates the proposed framework. The process involves obtaining a set of object proposals and constructing a DAG across all frames to show the association between different frame objects. A layered structure is formed such that each frame is represented by two layers in the graph. The graph consists of unary and binary edges. The unary edges show the appearance and motion information of each object proposal while the binary edges measure region, color, location, size and shape similarity between two object proposals. Dynamic programming is used to find the highest weighted path of the graph. The initial object segmentation results are refined using Gaussian Mixture Model (GMM) and Markov Random Field (MRF) based optimization to get the per pixel segmentation results. The technique is evaluated on two datasets using average pixel error as the evaluation measure. The solution does not segment videos with multiple objects and fails to produce satisfactory results for videos with foreground/background color and motion similarity.

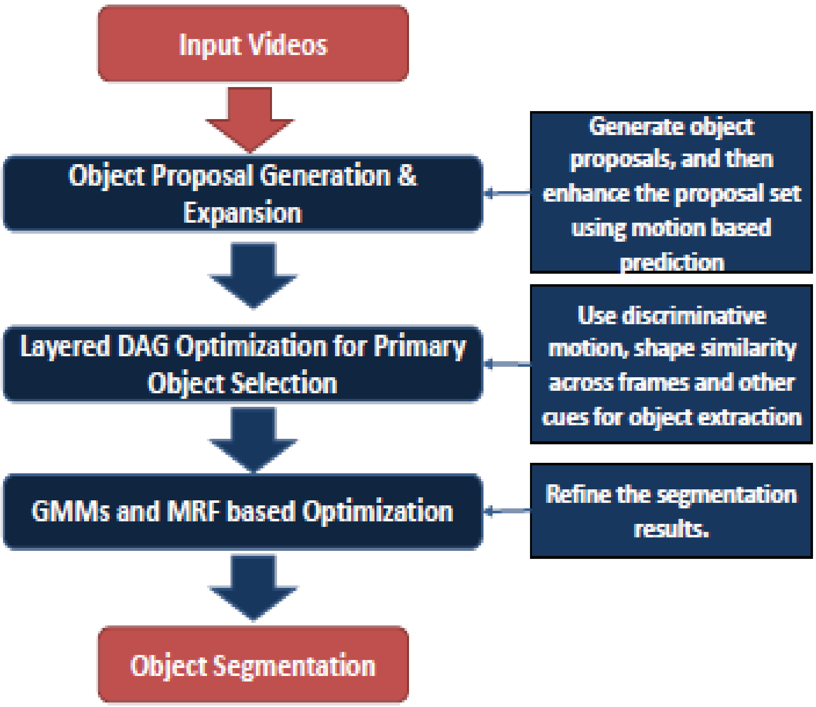
Input Videos
Object Proposal Generation & Expansion
Generate object proposals, and then enhance the proposal set using motion based prediction
Layered DAG Optimization for Primary Object Selection
Use discriminative motion, shape similarity across frames and other cues for object extraction
GMMs and MRF based Optimization
Refine the segmentation results.
Object Segmentation

Figure 2.4: Diagram taken from (Zhang et al., 2013) outlining the proposed framework

2.4 Clustering

Clustering is an unsupervised machine learning technique commonly used to group similar data items together. The pixels/superpixels can be used as the data points that need to be grouped according to specific features. The similarity between different pixels/superpixels is exploited using a distance function and similar items are grouped together to segment the data into two or more groups. Figure 2.5 provides an example showing clustering of 2-dimensional data into four clusters (RJ, 2015). Clustering can be performed using a variety of different methods (Han, 2005),

- Partitioning Methods

- Hierarchical Methods
- Density Based Methods
- Grid Based Methods

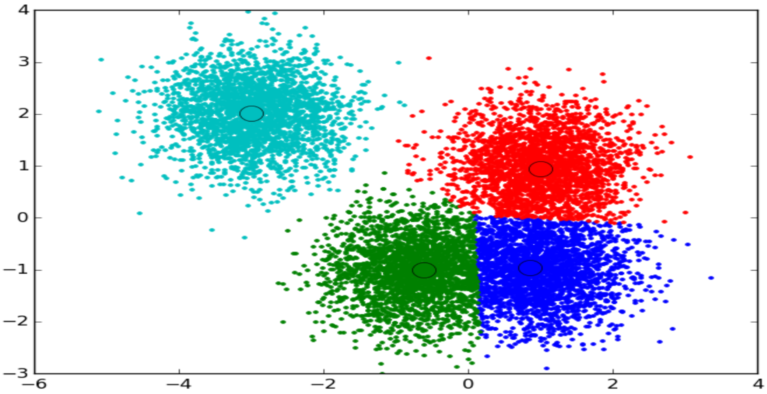

Figure 2.5: Example of 2-dimensional Clustering

(Galasso et al., 2013) proposes a video segmentation solution that uses Spectral Clustering to produce the final results. The method introduces the use of motion aware superpixel segmentations to form groupings of pixels similar in motion and appearance. The superpixels are used in multiple between and within frame combinations that use appearance and motion as features to show the association between superpixels over all the frames. Three different types of superpixel combinations are used:

- between-frame
- within-frame
- within and between frame.

The between-frame combination consists of two affinity matrices showing short-term-temporal affinity and long-term-temporal affinity where short-term affinity is measured between a small set of frames while long-term is calculated between a larger number of frames. The within-frame affinity consists of across-boundary-appearance affinity and across-boundary-motion affinity. Similarly, within and between frame affinities consist of spatio-temporal-appearance and spatio-temporal-motion affinity matrices. Spectral

Clustering is performed over the superpixels to get the final segmentation results. The solution is evaluated on a single dataset using average error as the evaluation measure. It fails to produce good results for videos with camera movement and blurred motion. Additionally, it is unable to segment multiple object videos effectively.

2.5 Others

Some of the existing solutions construct special graphs for performing segmentation. (Banica et al., 2013) propose a technique that extends Constrained Parametric Min Cuts method (CPMC) (Carreira and Sminchisescu, 2012) to generate motion and salient specific segment pool for each frame. A large coarse pool of Salient Segment Chains (SSCs) is constructed that correspond to paths in a trellis arranged and connected such that each node at each frame is connected to all the nodes of the immediate previous and next frames. A coarse SSC is built by initializing it with a segment and greedily adding other segments based on the robust-mean of their Euclidean distance from all the other segments. The chain is stopped when the distance falls below a specified threshold. The SSCs are ranked using Maximum Marginal Relevance (MMR) measure using per frame average segment overlap as the redundancy measure and the top 150 SSCs are used for further processing. A set of refined SSCs are obtained by labeling the pixels based on color using Gaussian Mixture Models, location using Euclidean distance transform, and foreground and boundary priors. Each refined SSC corresponds to one object in the video. A video partition and a potential function are defined such that a video partition does not have any overlapping SSCs and it cannot be extended using the SSCs from the pool. The potential function uses the properties of the segments within an SSC and the affinities between different SSCs to produce a segmentation solution using an energy minimization technique. The algorithm is evaluated on three different datasets and average per frame pixel error is used as the evaluation measure. The proposed solution outperforms the state-of-the-art methods but fails to produce satisfactory results for videos with blurred motion and camera movement.

(Wang et al., 2015) propose a saliency aware graph based solution that uses geodesic distance to solve the segmentation problem. Figure 2.6 shows the proposed framework. The method over-segments the input frames to obtain superpixels using SLIC (Achanta et al., 2008). Spatial static edges and motion boundary edges are obtained for all the superpixels. The spatio-temporal edge probability map is constructed by combining the spatial and motion edge information. Object probability of each superpixel is computed by using geodesic distance (i.e. the shortest path between two superpixels in one frame) in an intra-frame graph. The intra-frame graph consists of superpixels as nodes and edges forming the connection between the nodes. The weight on the edges is calculated using the spatio-temporal boundary probability of both nodes. The probability of a superpixel belonging to the foreground is calculated using shortest geodesic distance of the superpixel to the image boundaries. An initial set of background and foreground labeling are obtained using a self-adaptive threshold. Then, an inter-frame graph is constructed for each pair of subsequent frames to produce spatio-temporal saliency maps by computing the geodesic distance between the background regions of the two frames. A global appearance model for foreground and background is obtained using the saliency maps. Additionally, the motion information of few subsequent frames is used to build the dynamic motion model for each frame. The saliency maps, global appearance and dynamic location models are used to obtain the final segmentation by defining an energy function that consists of saliency, location and appearance in the unary terms and spatial and temporal information in the pairwise terms. The energy function is solved using graph-cuts to get the final segmentation results. The technique is evaluated on two different datasets using average per frame pixel error rate for evaluation. The results show that the method outperforms the state-of-the-art solutions. However, it only performs single object segmentation and hence, fails to produce a low error rate for videos consisting of multiple objects.

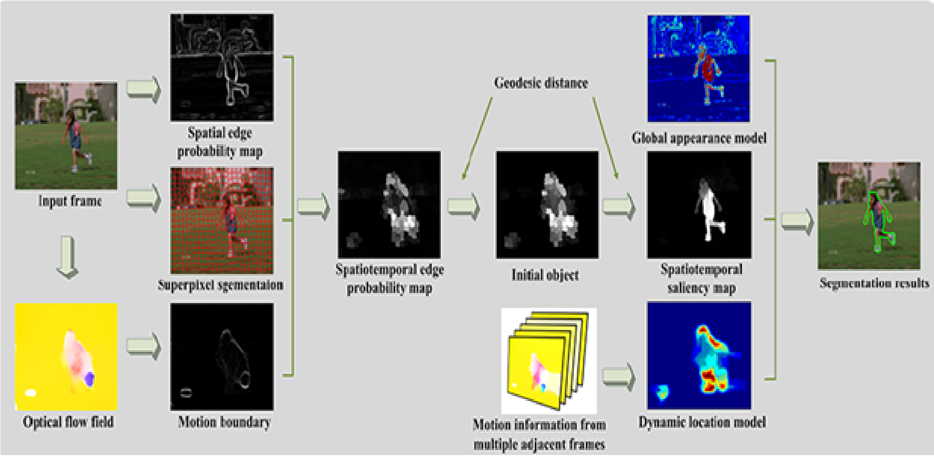

Figure 2.6: Figure taken from the (Wang et al., 2015) showing the proposed framework

Chapter 3

Proposed Methodology

This chapter proposes a solution for video object segmentation. The method consists of four steps:

- Feature Extraction
- Superpixel Segmentation and Merging
- Foreground Separation Model
- Conditional Random Field Construction and Solving.

Figure 3.1 provides an overview of the proposed framework.

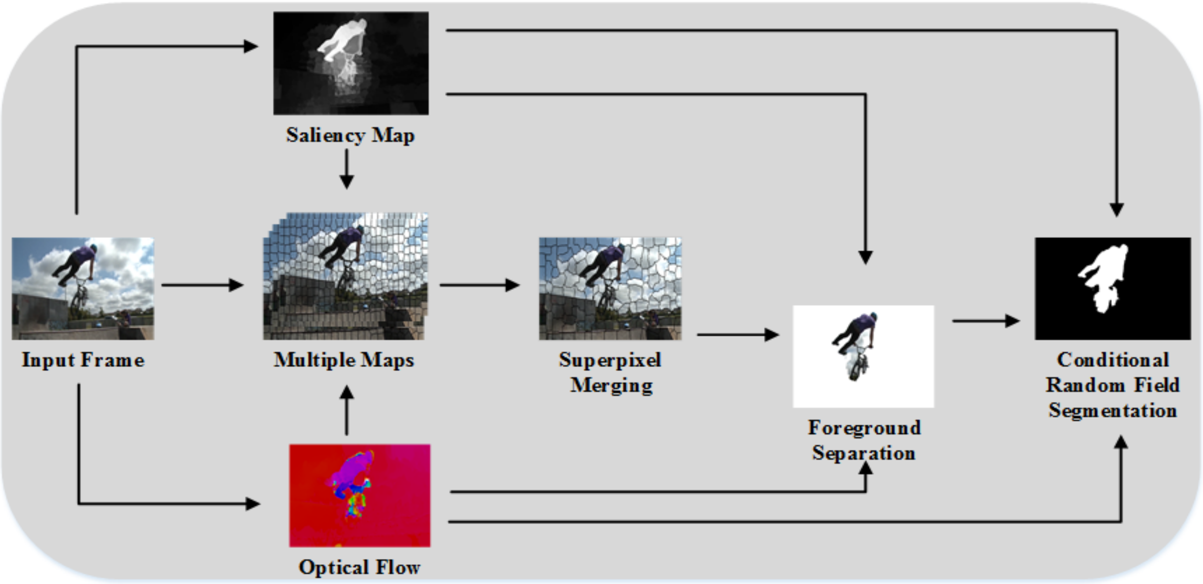

Figure 3.1: Overview of Proposed Framework

3.1 Feature Extraction

A number of different features are extracted from the input videos. The CIELAB color space representation and spatial co-ordinate values of each frame of the input video are obtained.

3.1.1 Optical Flow

Dense optical flow of each frame is calculated using the method given by (Chang et al., 2013). The method calculates the backward and forward optical flow of all the pixels of a frame by calculating the apparent motion of each pixel. The backward optical flow is the motion of pixels in frame t compared to the pixels in frame $t-1$ while forward optical flow is the motion of pixels in frame t compared to the pixels in frame $t+1$. The mean of the forward and backward motion of each pixel is calculated and used by the solution.

The method also uses the CIELAB color representation of the optical flow as mentioned in (Chang et al., 2013). The method works by constructing a color wheel for the possible optical flow values and assigns a specific color to each motion value. An example input frame and its corresponding optical flow color representation are showed in figure 3.2

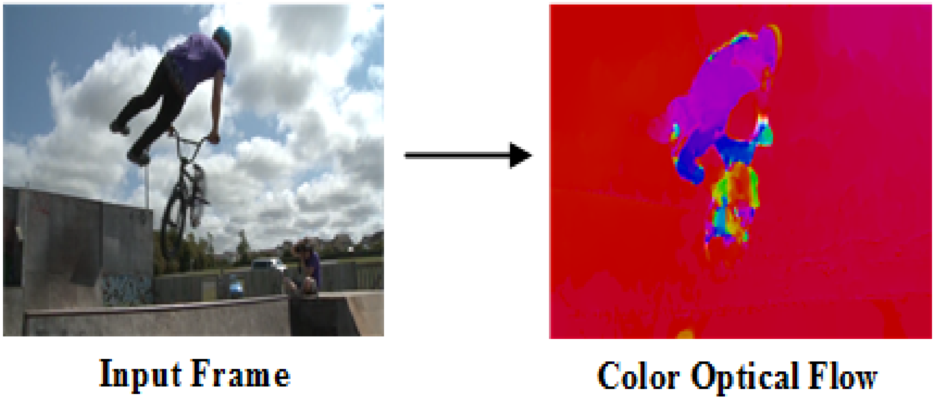

Figure 3.2: Example image with optical flow color representation

3.1.2 Spatio-temporal Saliency

The spatio-temporal saliency score for each input frame is obtained using the method proposed by (Liu et al., 2014). The method uses motion and color as local features at the superpixel level and global features at the frame level.

The temporal and spatial saliency is measured at the superpixel level and a pixel-level saliency method is derived to obtain temporal and spatial saliency maps at the pixel-level. An adaptive fusion method is used to compute the final spatiotemporal saliency map. Figure 3.3 shows an input image and its corresponding spatio-temporal saliency map.

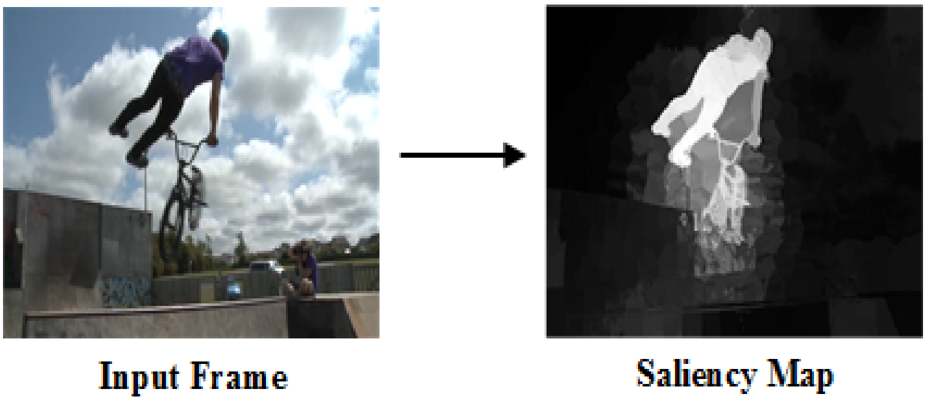

Figure 3.3: Example image with spatio-temporal saliency map

3.2 Superpixel Segmentation and Merging

The technique proposed by (Li and Chen, 2015) is extended to obtain spatio-temporal superpixel segmentations for the input frames. The original technique was proposed for dataset consisting of images and produced superpixels that were uniform in nature and required low computation time. The method represented each image pixel using a 5-dimensional feature vector $p = (l, \alpha, \beta, x, y)$ where l, α, β were CIELAB color components while x, y were the spatial coordinates of each pixel in the image. Originally, the 5-dimensional feature vector was mapped onto a 10-dimensional feature vector $\phi(p)$ to generate superpixels using Linear Spectral Clustering (LSC) using,

$$l_1(i,j) = \cos\frac{\pi}{2}.l(i,j) \tag{3.1}$$

$$l_2(i,j) = \sin\frac{\pi}{2}.l(i,j) \tag{3.2}$$

Where l_1, l_2 were the two mappings of l component of CIELAB color value of a pixel present at location (i, j). The mappings of the other features were calculated in the same manner.

While the original algorithm provides satisfactory results for images with variation in foreground and background color, the performance deteriorates when there is similarity between adjacent foreground and background pixels of the input image. This happens because all pixels with similar colors located close to one other are grouped into a single superpixel.

In order to produce superpixels with separate foreground and background pixels irrespective of their spatial closeness and color similarity, the feature vector is extended to include optical flow and saliency values resulting in an 8-dimensional spatio-temporal feature vector $p = (l, \alpha, \beta, x, y, u, v, s)$ where u, v is the optical flow and s is the saliency score of the respective pixel. The extended 8-dimensional feature vector is mapped onto a 16-dimensional feature vector and superpixels are generated for each frame of the video using Linear Spectral Clustering.

Instead of using a single oversegmentation for each frame, multiple different superpixel segmentations for each frame are obtained with varying number of superpixels using different values for color, saliency and flow constants. For identifying the exact number of segmentations to use, two segmentations are started with and the number is increased until the superpixels are constructed such that the process is not too time consuming and foreground/background separation is maximized. A total of five superpixel segmentations are taken as the superpixel segmentation and merging using five combinations takes average computation time and produces a good level of separation between foreground and background. The use of different constant values helps generate five non-identical segmentation maps-each grouping pixels based on emphasis given to different features. Table 3.1 shows the values of constants and superpixels used for each segmentation.

S. No	Pixels per Superpixel	Color	Saliency	Motion
1	450	100	0	0
2	400	100	20	50
3	350	100	20	30
4	300	100	20	40
5	250	100	10	50

Table 3.1: The parameters used for representing the input frame in different superpixel segmentations.

The five oversegmentations are merged into a single superpixel map that accurately segments the input frame into a segmentation having 500 pixels per superpixel. For the merging process, the technique proposed by (Li et al., 2012) is used. The method constructs a bipartite graph showing relationships between pixels and superpixels within and between the multiple superpixel segmentations. The graph is solved using spectral clustering to produce a single oversegmentation effectively merging different superpixel segmentations into one. Figure 3.4 shows the result of the superpixel segmentation process over an image.

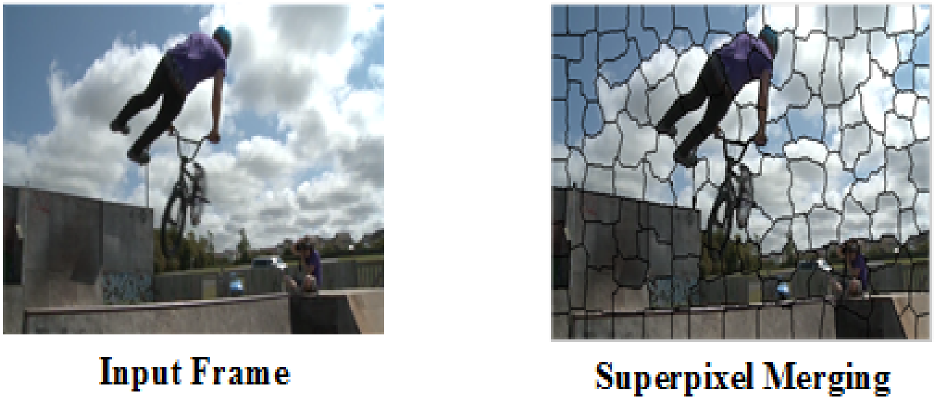

Figure 3.4: Example image with superpixel segmentation map

3.3 Foreground Separation Model

This step presents a foreground separation model that uses the superpixel segmentation from the previous step, and spatio-temporal saliency score, CIELAB color and CIELAB optical flow color representation as features to obtain an initial segmentation map for each frame of the input video. This model consists of three different thresholding steps:

- Saliency thresholding
- Optical Flow thresholding
- Color thresholding

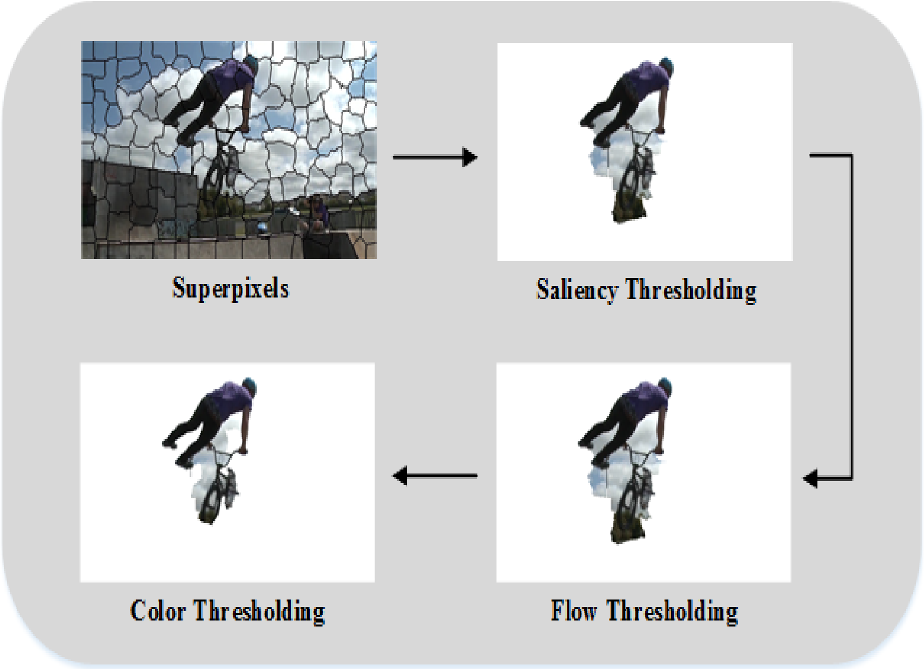

Figure 3.5: Overview of foreground separation model

Figure 3.5 shows the results produced after different steps of the foreground separation model.

3.3.1 Saliency Thresholding

In order to obtain an initial set of background segments having low saliency scores, a saliency threshold,s, is introduced. All the pixels in the input frame are labeled using,

$$label_s = \begin{cases} 0, & \text{if } saliency(i,j) < s \\ 1, & \text{otherwise} \end{cases} \tag{3.3}$$

Where $saliency(i, j)$ is the spatio-temporal saliency score of a given pixel in an input frame.

Then, the number of pixels labeled as foreground and background are counted for each segment. In order to assign same label to all the pixels in a given segment, the entire segment is labeled with the label of the majority of pixels in the frame to get the results of saliency thresholding.

3.3.2 Optical Flow Thresholding

As a result of saliency thresholding, the majority of background segments having low spatio-temporal saliency scores are classified to be belonging to the background. However, there will exist some background segments which are classified as the foreground due to their high spatio-temporal saliency score. Such segments have a high saliency score due to their spatial closeness with foreground segments. In order to relabel such background segments currently labeled as foreground, a flow threshold f is defined.

For each foreground segment, the segments adjacent to it in the $N4$ neighborhood are found. For each adjacent background segment, the Euclidean distance between the average CIELAB optical flow color values of the two segments is calculated. All the foreground segments are relabeled,

$$label_f = \begin{cases} 0, & \text{if } dist(i,j) < f \\ 1, & \text{otherwise} \end{cases} \tag{3.4}$$

Where $dist(i,j)$ is the Euclidean distance in average optical flow of two segments, i and j, such that one is currently labeled as foreground while the other is background. The relabeling process is repeated until there is no change in the labels of the segments.

3.3.3 Color Thresholding

This step consists of two parts:

- Superpixel level color thresholding
- Pixel level color thresholding

After flow thresholding, there is a possibility of having segments belonging to the background but currently labeled as foreground due to their close color similarity with the foreground segments.To relabel such segments, a superpixel level color threshold, c_s, is defined,

$$c_s = \frac{1}{t} \sum_{i,j \in B_s} \sqrt{(l_i - l_j)^2 + (\alpha_i - \alpha_j)^2 + (\beta_i - \beta_j)^2} \tag{3.5}$$

Where l, α, β are the CIELAB color values of two neighboring segments i, j belonging to B_s the set of segments labeled as background, and t is the total count of such segment pairs.

Again, the background segments adjacent to currently labeled foreground segments are found and the Euclidean distance between the average CIELAB color values of the two segments is calculated. The foreground segments are relabeled,

$$label_c = \begin{cases} 0, & \text{if } dist(i,j) < c_s - \epsilon \\ 1, & \text{otherwise} \end{cases} \tag{3.6}$$

Where $dist(i,j)$ is the Euclidean distance in average CIELAB color of two segments, i and j, such that one is currently labeled as foreground while the other is background and ϵ is a constant. The relabeling process is repeated until there is no change in the labels of the segments.

Finally, for background pixels falsely labeled as foreground another threshold c_p is defined,

$$c_p = \frac{1}{t} \sum_{m,n \in B_p} \sqrt{(l_m - l_n)^2 + (\alpha_m - \alpha_n)^2 + (\beta_m - \beta_n)^2} \tag{3.7}$$

Where l, α, β are the CIELAB color values of two neighboring pixels m, n belonging to B_p the set of pixels labeled as background, and t is the total count of such pixel pairs.

All the foreground pixels are relabeled using,

$$label_p = \begin{cases} 0, & \text{if } dist(m,n) < c_p \\ 1, & \text{otherwise} \end{cases} \tag{3.8}$$

Where $dist(m,n)$ is the Euclidean distance in CIELAB color of two pixels, m and n, such that one is currently labeled as foreground while the other is background. The relabeling process is repeated until significant change in the labeling stops.

3.3.4 Post-processing

Finally, the number of foreground and background pixels for each segment are counted and each segment is relabeled as the majority label. Then, simple morphological operations are applied to fill holes and remove unnecessary segments existing in isolation. The foreground separation model provides an initial labeling for superpixels helping to identify possible object regions. The result of this model is used in the next step while performing final segmentation.

3.4 Conditional Random Field Construction and Solving

This is the final step of the proposed solution. In this, a Conditional Random Field is constructed and solved to get the final segmentation results. This step consists of three parts:

- Preprocessing
- Potential Function Definition
- Segmentation using Conditional Random Field

3.4.1 Preprocessing

The foreground separation provides an initial labeling for possible foreground and background pixels. This labeling can be used to calculate the likelihood of a pixel belonging to a particular class. Gaussian Mixture Model (GMM) implemented by (Li et al., 2013b) is used to model the likelihood for each pixel in a given frame. Gaussian Mixture Models (GMM) are parametric probability distributions that use weighted sum of multiple Gaussian densities for representing conditional probabilities. The parameters for the distribution are learned using an Expectation Maximization (EM) algorithm.

The color and optical flow color representation of the input frame are used separately to calculate the individual GMM likelihoods for foreground and background of both features using the labeling provided by foreground separation model. This information is used in the unary potential calculation of each pixel.

The labeling produced by foreground separation model for each frame is converted into RGB color space by assigning different color values to both labels. This color representation is used in the pairwise potential calculation to impose a high cost on the incorrect labeling of neighboring pixels.

3.4.2 Potential Function Definition

Here, a potential function is defined that is used to label the unary and pairwise energies of nodes in the next step. The potential function uses spatio-temporal saliency, GMM color, GMM optical flow color and RGB foreground separation labeling for energy calculation.

3.4.2.1 Unary Potential

The unary potential is used to label the nodes with the likelihood of each possible label. Since there are two possible labels, each node will have two unary potentials. The unary potential uses spatio-temporal saliency, GMM color and GMM optical flow color values of each pixel to calculate the energy. The unary energy of each pixel for foreground is calculated using,

$$u_f(i,j) = \lambda_f GF_f(i,j) + \lambda_c GF_c(i,j) + \lambda_s(-\log saliency(i,j)) \tag{3.9}$$

Where $GF_f(i,j), GF_c(i,j)$ are the GMM color foreground likelihood and GMM optical flow color foreground likelihood for a pixel at location (i,j) of the frame. $saliency(i,j)$ is the spatio-temporal saliency score of the pixel while $\lambda_f, \lambda_c, \lambda_s$ are tuning constants.

Similary, the unary potential of each pixel for background is calculated using,

$$u_b(i,j) = \lambda_f GB_f(i,j) + \lambda_c GB_c(i,j) \tag{3.10}$$

Where $GB_f(i,j), GB_c(i,j)$ are the GMM color background likelihood and GMM optical flow color background likelihood for a pixel at location (i,j) of the frame.

It should be noted that saliency scores have not been used while calculating background unary potential. This is because if a pixel belongs to the background, its saliency score should ideally be close to zero. Furthermore, the saliency value of foreground pixels should be significantly high. Hence, the saliency score is only used in the calculation of foreground unary potential.

3.4.2.2 Pairwise Potential

The pairwise potential is used to label the edges connecting two nodes in the Conditional Random Field. It is defined to impose spatial consistency among neighboring pixels. The method of calculating pairwise term is inspired from (Li et al., 2013b) and is found using,

$$p(a,b) = L_a - L_b(\lambda_1 + \lambda_2(\exp\left(\frac{-||r_a - r_b||}{\beta}\right))) \tag{3.11}$$

Where L_a, L_b are the labels for pixels a, b (connected in the 8-neighborhood) predicted by the foreground separation model and r_a, r_b are the RGB foreground separation labeling value of each pixel. λ_1, λ_2 are tuning parameters and β is the average difference in color of the RGB representation of the two labels.

3.4.3 Segmentation using Conditional Random Field

This step involves constructing a Conditional Random Field and solving it to obtain the final segmentation results for the input sequence. Each frame of the video is taken separately and the pixels in it are used to represent the nodes of the CRF. The pixels are labeled using their unary background and foreground potential. An edge is maintained between every pixel and each of its $N8$ neighbors. The weight of the edge is determined by calculating the pairwise energy maintaining spatial consistency throughout the graph.

The overall energy of the graph is represented as the summation of the unary and pairwise potentials for each pixel. The aim is to find the labeling that minimizes the overall energy of the graph. To solve the CRF, max-flow/min-cut algorithm implemented by (Li et al., 2013b) is used. Figure 3.6 shows an example segmentation produced by the CRF construction and solving step.

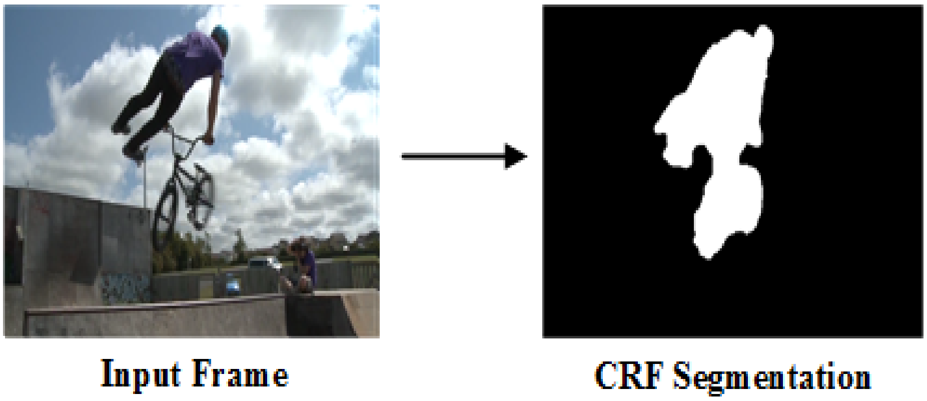

Figure 3.6: Example image with final segmentation result

Chapter 4

Implementation and Results

This chapter includes the implementation details of the proposed solution. Additionally, the experimental results of the proposed technique and comparison with state-of-the-art methods is also covered in detail.

4.1 Implementation Details

The solution is implemented and evaluated on Matlab R2014 on a 64 bit machine with Intel i7 processor and Windows 10 operating system. The implemented solution takes approximately 2.5 minutes to segment a single frame of size 259 x 327.

The constants mentioned in the previous chapter are learned by taking a set of frames from each input sequence of SegTrack v2 dataset and iteratively evaluating the results over different values to find the ones that produce the lowest error. For the foreground separation model, each constant is identified separately by using the output of separate threshlolding steps and choosing the value that minimizes the overall error for that particular step. In pairwise term of CRF construction and Solving, (Li et al., 2013b) suggest the ratio of the two parameters to be 1:5. The parameter learning process is started with this ratio and different combinations of both the constants in the range of 1-10 are tried. The parameter values that give the lowest error are selected. For the unary term of the Conditional Random Field, different combinations of the three parameters for three possible values (i.e. 0.00, 0.50 and 1.00) are evaluated and the error for each set is calculated. This is used to understand the effect of each parameter on the overall results. The parameter values producing the lowest error are taken and the value for each parameter is refined by keeping the other two parameters constant and only changing that particular parameter between 0.00-1.00 using a window of 0.10. The parameter values producing the lowest error are selected. In the next step, the parameter values are refined again using values in the range of 0.01 of the new values. Then, the constants producing the overall lowest error are selected. The selected constant values are mentioned in Table 4.1

S. No	Methodology Step	Name	Value
1	Foreground Separation Model	s	75.00
2	Foreground Separation Model	f	4.00
3	Foreground Separation Model	ϵ	10.00
4	CRF Construction and Solving	λ_f	0.70
5	CRF Construction and Solving	λ_c	0.13
6	CRF Construction and Solving	λ_s	1.00
7	CRF Construction and Solving	λ_1	6.00
8	CRF Construction and Solving	λ_2	10.00

Table 4.1: The parameters used in different steps of the solution.

4.2 Datasets

The proposed solution is evaluated on two different video object segmentation datasets. The use of variety of videos from different datasets helps explore the extent to which the proposed and existing solutions conform to generic sequences.

4.2.1 SegTrack v2 Dataset

The SegTrack v2 dataset (Li et al., 2013a) consists of 14 video sequences. The properties of all the video sequences of the dataset are mentioned in Table 4.2.

Sequences	Frames	Objects	Motion	Properties	Camera
birdofparadise	97	1	Smooth	Appearance change	Static
birdfall2	29	1	Smooth	Simple	Static
bmx	35	2	Blur	Object occlusion, deformation and interaction	Static
cheetah	28	2	Blur	Object occlusion, deformation and interaction	Moving
frog	278	1	Slow	Simple	Static
drift	73	2	Smooth	Object occlusion and interaction	Moving
girl	20	1	Smooth	Object deformation	Moving
hummingbird	28	2	Blur	Object deformation, occlusion and interaction	Static
monkey	30	1	Smooth	Object deformation	Moving
monkeydog	70	2	Blur	Object deformation	Moving
parachute	50	1	Smooth	Simple	Moving
penguin	41	6	Smooth	Object occlusion	Static
soldier	31	1	Smooth	Object deformation	Moving
worm	242	1	Blur	Simple	Static

Table 4.2: The properties of different sequences of SegTrack v2 Dataset.

4.2.2 Videos used by (Fukuchi et al., 2009)

A total of 10 videos were used by (Fukuchi et al., 2009) for evaluation in their video segmentation paper. Out of those, only 9 have been used in the experimentation. The properties of each video are mentioned in Table 4.3.

Sequences	Frames	Objects	Motion	Properties	Camera
AN119T	84	1	Smooth	Simple	Static
BR128T	102	1	Smooth	Appearance change	Static
BR130T	65	1	Blur	Appearance change	Static
DO01_013	73	3	Smooth	Appearance change	Moving
DO01_014	85	1	Smooth	Simple	Moving
DO01_030	85	1	Smooth	Object deformation	Static
DO01_055	47	1	Blur	Object deformation	Moving
DO02_001	64	1	Blur	Simple	Moving
VWC102T	91	1	Slow,Blur	Object deformation	Moving

Table 4.3: The properties of different sequences used by (Fukuchi et al., 2009).

4.3 Evaluation Measures

For evaluation, average pixel error per frame (Tsai et al., 2012) , precision, recall and F1-measure are used. The average per-frame pixel error rate for a sequence is the average number of pixels per frame misclassified when evaluated against the ground-truth segmentation. Precision is a measure of exactness for a given sequence and is calculated using,

$$precision = \frac{t_p}{t_p + f_p} \tag{4.1}$$

Where t_p, f_p are true positives and false positives respectively.

Recall is a measure of completeness and is calculated using,

$$recall = \frac{t_p}{t_p + f_n} \tag{4.2}$$

Where t_p, f_n are true positives and false negatives respectively.

F1-score is the harmonic mean of precision and recall calculated using,

$$F_1 = \frac{2.precision.recall}{precision + recall} \tag{4.3}$$

In terms of accuracy, a lower average pixel error rate, and higher precision, recall and F1-score are preferred.

4.4 State-of-the-art methods

The proposed solution is compared with three state-of-the-art methods. The first method was proposed by (Lee et al., 2011). The solution uses spatio-temporal features to score different regions of an image. The regions are then clustered and ranked to identify key segments that are later labeled as foreground and background. The second method was proposed in by (Galasso et al., 2013) which uses multiple superpixel segmentations to represent each frame in the video. The superpixel maps are solved using spectral clustering to get the final segmenation results. The third method was proposed by (Zhang et al., 2013) which presents a layered Directed Acyclic Graph (DAG) based framework using spatial and temporal features to extract primary objects from a given video sequence. The initial segmentation results are refined by Gaussian Mixture Model (GMM) and Markov Random Field (MRF) based optimization.

4.5 Other Comparison method

The proposed solution is also compared with an alternative method which was proposed during the course of the research. This alternative method uses the same first three steps as the proposed solution but uses inter-frame Spectral Clustering in the last step of implementation. The method will be referred to as the Spectral Clustering (SC) solution.

4.6 Time Complexity

Table 4.4 includes the system specifications of the implementations of the proposed and state-of-the-art solutions along with running time complexity of each over a 10 frame video sequence of size 259 x 327.

Method	Specifications	Running Time
Ours	Matlab R2014, 64 bit machine with Intel i7 processor and Windows 10 operating system	25 minutes
Spectral Clustering	Matlab R2014, 64 bit machine with Intel i7 processor and Windows 10 operating system	30 minutes
(Lee et al.)	Matlab R2012, 64 bit machine with 4 processors and Virtual Machine Ubuntu operating system	40 minutes
(Zhang et al.)	Matlab R2014, 64 bit machine with Intel i7 processor and Windows 10 operating system	18 minutes
(Galasso et al.)	Matlab R2012, 64 bit machine with 4 processors and Virtual Machine Ubuntu operating system	30 minutes

Table 4.4: The system specifications and running time complexity of proposed and state-of-the-art methods.

4.7 Results

Table 4.5 shows the average pixel error rate for Segtrack v2 dataset while Table 4.6 shows the same for dataset used by (Fukuchi et al., 2009). The lowest error for each sequence is shown in bold.

	Ours	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
bird of paradise	**2753**	11572	35264	13783
birdfall	626	1091	**150**	18533
bmx	**10213**	12071	152527	17133
cheetah	2424	22977	**1914**	4532
drift	**15031**	21864	150664	29643
frog	**4385**	8860	7217	19955
girl	4957	2054	**1495**	4079
hummingbird	17696	18288	**13601**	21404
monkey	7039	**3205**	3430	18697
monkeydog	7967	9081	**1676**	2947
parachute	1251	**204**	3695	407
penguin	36797	**24772**	30205	37092
soldier	9503	57586	24764	**5592**
worm	**3140**	46600	3509	5930
Average	**8841**	17159	30722	14266

Table 4.5: The average per-frame pixel error rate of different segmentation techniques over SegTrack v2 dataset.

	Ours	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
AN119T	1855	**1679**	18461	5348
BR128T	**12364**	19624	14732	24033
BR130T	**1643**	1817	7702	5214
DO01_013	**10924**	18159	23835	26348
DO01_014	3540	**1232**	2050	19356
DO01_030	**15884**	26378	29220	61826
DO01_055	**2594**	73893	11323	5538
DO02_001	1731	**804**	1738	16573
VWC102T	**8143**	13171	18489	14708
Average	**6520**	17417	14172	19883

Table 4.6: The average per-frame pixel error rate of different segmentation techniques over dataset used by (Fukuchi et al.)

Table 4.7 shows the precision, recall and Table 4.8 shows the F1-measures of the proposed solution and state-of-the-art methods on different video sequences of the SegTrack v2 dataset. The highest precision, recall and F1-measure for each sequence is shown in bold.

	Ours		(Lee et al.)		(Zhang et al.)		(Galasso et al.)	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
bird of paradise	0.97	**0.96**	0.86	0.85	**0.99**	0.12	0.97	0.68
birdfall	0.43	**0.80**	0.00	0.00	**0.89**	0.78	0.01	0.38
bmx	0.68	0.76	**0.98**	0.30	0.10	**0.99**	0.53	0.05
cheetah	0.63	**0.46**	0.03	0.23	**0.94**	0.40	0.00	0.00
drift	**0.81**	0.47	0.53	0.53	0.13	**1.00**	0.00	0.00
frog	**0.86**	0.47	0.00	0.00	0.73	**0.91**	0.05	0.10
girl	0.67	0.77	**0.97**	0.77	0.96	**0.80**	0.81	0.61
hummingbird	0.78	0.33	**0.98**	0.22	0.88	**0.48**	0.93	0.09
monkey	0.46	**0.92**	**0.67**	0.91	0.65	0.91	0.07	0.22
monkeydog	0.10	0.24	0.01	0.03	**0.97**	**0.40**	0.00	0.00
parachute	0.84	0.82	**0.99**	**0.96**	**0.99**	0.84	0.97	0.92
penguin	0.30	0.13	0.56	1.00	**0.79**	0.06	0.44	**0.67**
soldier	0.39	**0.90**	0.09	0.95	0.17	0.84	**0.68**	0.19
worm	**0.53**	0.84	0.06	**0.85**	0.06	0.52	0.00	0.00
Average	0.60	0.63	0.48	0.54	**0.66**	**0.65**	0.39	0.28

Table 4.7: The precision and recall of different segmentation techniques over SegTrack v2 dataset.

	Ours	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
bird of paradise	**0.97**	0.85	0.22	0.80
birdfall	0.56	0.00	**0.83**	0.02
bmx	**0.72**	0.46	0.18	0.10
cheetah	0.53	0.06	**0.56**	0.00
drift	**0.59**	0.53	0.24	0.00
frog	0.61	0.00	**0.81**	0.07
girl	0.72	0.86	**0.88**	0.70
hummingbird	0.47	0.36	**0.62**	0.17
monkey	0.61	**0.77**	0.76	0.11
monkeydog	0.14	0.02	**0.57**	0.00
parachute	0.83	**0.97**	0.91	0.94
penguin	0.18	**0.72**	0.11	0.53
soldier	**0.54**	0.17	0.29	0.30
worm	**0.65**	0.11	0.11	0.00
Average	**0.58**	0.42	0.51	0.27

Table 4.8: The F1-measure of different segmentation techniques over SegTrack v2 dataset.

Table 4.9 shows the precision, recall and Table 4.10 shows the F1-measures of the proposed solution and state-of-the-art methods on different video sequences of the dataset used by (Fukuchi et al., 2009) in their research. The highest precision, recall and F1-measure for each sequence is shown in bold.

	Ours		(Lee et al.)		(Zhang et al.)		(Galasso et al.)	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
AN119T	**0.98**	0.92	0.92	**0.99**	0.92	0.90	0.86	0.85
BR128T	**0.78**	0.24	0.00	0.00	0.74	**0.78**	0.12	0.10
BR130T	0.93	0.86	0.87	**0.92**	0.64	0.03	**0.94**	0.40
DO01_013	**0.98**	**0.61**	**0.98**	0.34	0.65	0.05	0.88	0.04
DO01_014	0.88	0.77	**0.95**	0.94	0.86	**0.97**	0.27	0.46
DO01_030	**0.98**	0.47	0.53	**1.00**	0.86	0.00	0.32	**1.00**
DO01_055	**0.76**	0.69	0.05	**0.86**	0.28	0.82	0.45	0.58
DO02_001	**0.96**	0.88	**0.96**	0.97	0.90	0.96	0.41	**0.99**
VWC102T	**0.56**	0.47	0.04	0.02	0.31	**0.84**	0.19	0.19
Average	**0.87**	0.66	0.59	**0.67**	0.68	0.59	0.49	0.51

Table 4.9: The precision and recall of different segmentation techniques over dataset used by (Fukuchi et al.).

	Ours	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
AN119T	0.95	**0.96**	0.91	0.85
BR128T	0.36	0.00	**0.76**	0.11
BR130T	**0.90**	0.89	0.06	0.56
DO01_013	**0.75**	0.51	0.09	0.08
DO01_014	0.82	**0.94**	0.91	0.34
DO01_030	0.63	**0.69**	0.00	0.49
DO01_055	**0.72**	0.10	0.41	0.50
DO02_001	0.92	**0.96**	0.93	0.58
VWC102T	**0.51**	0.03	0.45	0.19
Average	**0.73**	0.56	0.50	0.41

Table 4.10: The F1-measures of different segmentation techniques over dataset used by (Fukuchi et al.).

The proposed solution is compared with the Spectral Clustering solution using only the first 50 frames of videos having more than 50 frames. This is due to the high memory and time requirement of the Spectral Clustering solution when constructing and solving the inter-frame graph. Table 4.11 shows the average pixel error per frame and Table 4.12 shows the precision, recall and F1-measures of the proposed solution, Spectral Clustering solution and state-of-the-art methods on the SegTrack v2 dataset. The lowest error and highest precision, recall and F1-measure are shown in bold.

	Ours	SC	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
bird of paradise	**1312**	4845	13041	35429	11832
birdfall	626	980	1091	**150**	18533
bmx	**10213**	10339	12071	152527	17133
cheetah	2423	2626	22977	**1914**	4532
drift	**10879**	19614	15503	147480	42130
frog	**336**	2854	3387	1767	15216
girl	4957	4802	2054	**1495**	4079
hummingbird	17696	18237	18288	**13601**	21404
monkey	7039	7230	**3205**	3430	18697
monkeydog	7030	8065	9337	**2241**	3671
parachute	1251	1104	**202**	219	382
penguin	36797	36786	**24751**	30325	37022
soldier	9503	6660	57586	24764	**5592**
worm	**644**	2839	1470	4147	17745
Average	**7908**	9070	13212	29963	15569

Table 4.11: The average per-frame pixel error rate of proposed technique, Spectral Clustering solution and state-of-the-art methods over SegTrack v2 dataset.

	Ours	SC	(Lee et al.)	(Zhang et al.)	(Galasso et al.)
Precision	0.61	0.61	**0.63**	0.62	0.43
Recall	**0.66**	0.60	0.57	**0.66**	0.31
F-measure	**0.60**	0.57	0.52	0.54	0.28

Table 4.12: The precision, recall and F-measure of different segmentation techniques over the complete SegTrack v2 dataset.

Figure 4.1 and Figure 4.2 show some visual segmentation results for all techniques on SegTrack v2 dataset and (Fukuchi et al., 2009) sequences respectively.

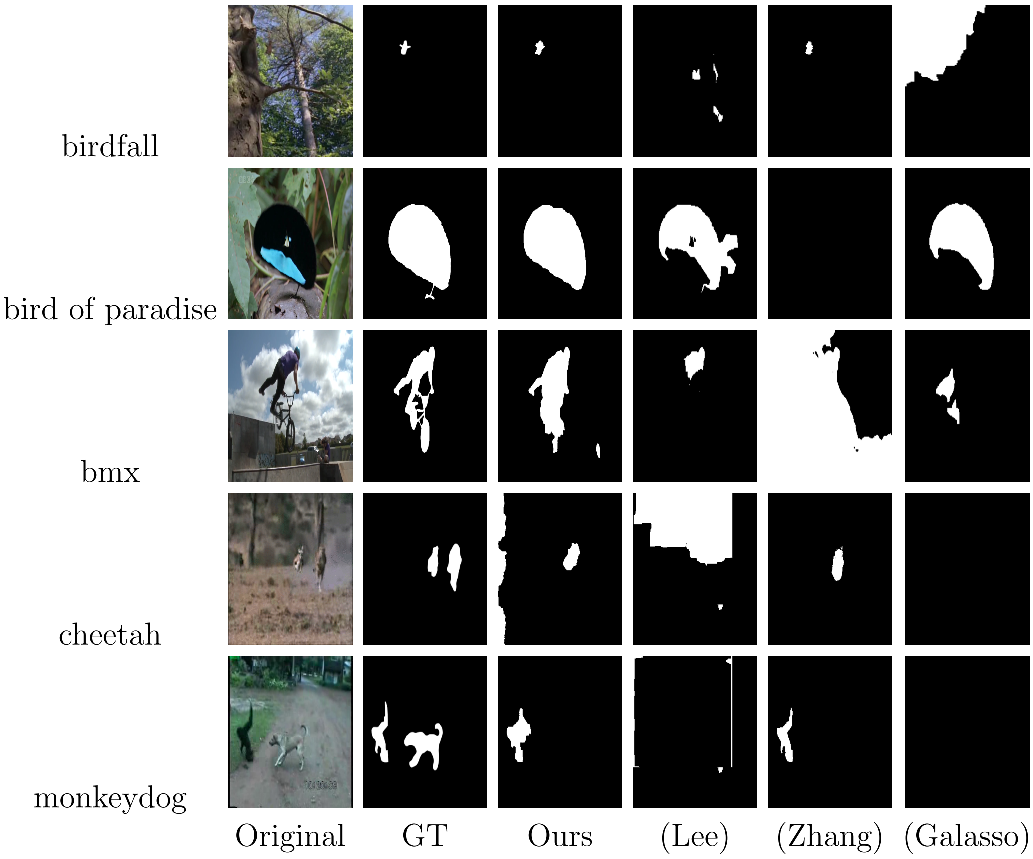

Figure 4.1: Comparison of our segmentation results with other methods against the ground-truth on SegTrack v2 dataset.

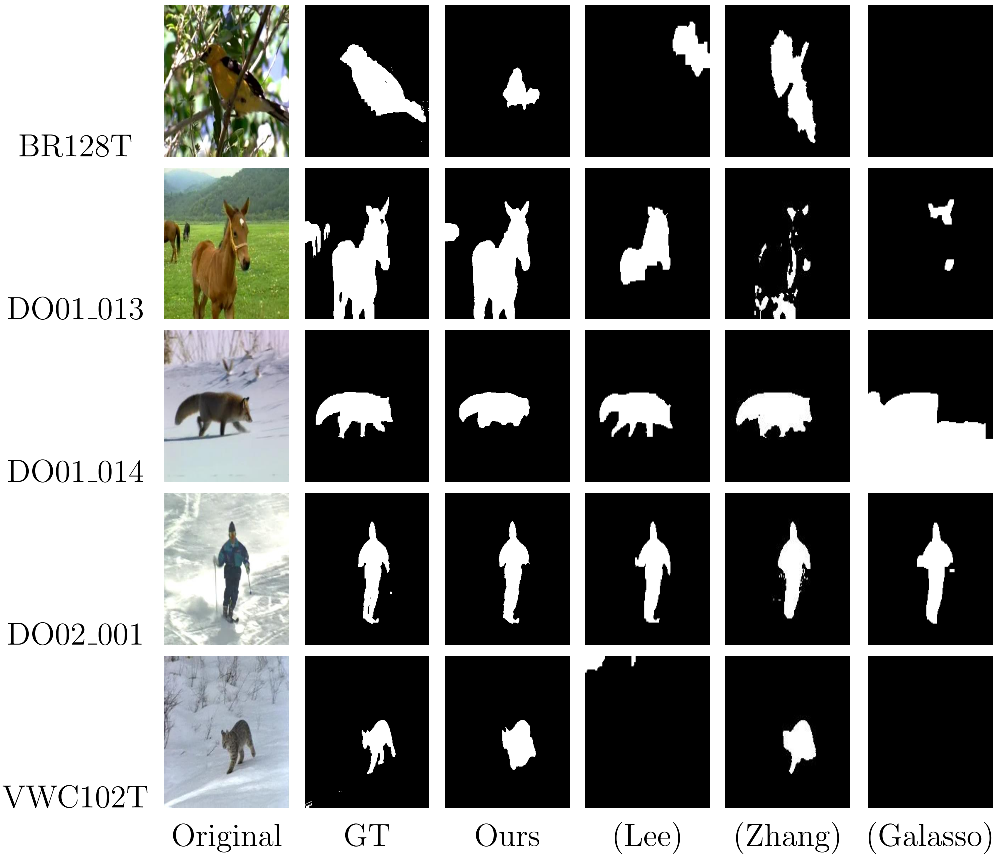

Figure 4.2: Comparison of our segmentation results with other methods against the ground-truth on (Fukuchi et al.) dataset.

4.8 Discussion

Tables 4.5 and 4.6 show that the proposed technique produces a lower error rate compared to the other techniques. The proposed methodology not only outperforms the existing techniques but maintains a margin of 5,425 over the SegTrack v2 dataset and 7,652 over the dataset provided by (Fukuchi et al., 2009) when compared with the second best performing technique. The average pixel error rate over both datasets proves the superiority of the proposed technique over the existing solutions.

Tables 4.7, 4.8, 4.9 and 4.10 show that the proposed solution produces the highest F1-measure for both datasets. Although, the methodology has a lower average precision and recall on the SegTrack v2 dataset and lower recall over the other dataset, the difference between the highest quantities is not too high. Additionally, as the F1-score is higher for both datasets, it can be concluded that the performance of the proposed solution is better since

F1-score is the harmonic mean of precision and recall and is considered a better measure for evaluation.

Tables 4.11 and 4.12 show the comparison of our solution with another method proposed during the research process. The results show that our method produces a lower average error and higher precision, recall and F1-score than both the previously proposed and existing state-of-the-art methods.

The proposed solution outperforms the existing solutions on both the datasets and maintains decent results for all input videos. The results can be analyzed to conclude that the presented methodology is generic in nature as it is able to produce satisfactory results for both datasets. It should be noted that while the technique by (Galasso et al., 2013) and (Zhang et al., 2013) produce the lowest average per-frame pixel error rate for one dataset, they provide the highest error for the other dataset. This drastic change in performance reflects the non-generic nature of the existing solutions. Additionally, the visual segmentation results in Figures 4.1 and 4.2 clearly show that the proposed solution is able to generate satisfactory results for all types of videos consisting of both single and multiple objects. However, although the presented technique produces decent results for both datasets, the results can still be improved to further lower the overall segmentation error. The proposed solution generates a higher error for videos with camera movement and multiple moving objects which can be improved to produce better results.

The other techniques produced high error rates due to their inability to separate foreground and background parts having a high color similarity particularly for sequences captured with a moving camera. The proposed methodology produces superpixels that decently separate the foreground and background pixels on the basis of color, optical flow, saliency and spatial coordinates. Additionally, the foreground separation model introduces thresholds that iteratively separate the foreground segments from the background for individual frames. The final step of the algorithm uses the result of foreground separation model along with other features to construct a CRF that on solving produces suitable segmentation results.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The thesis discussed the problem of video object segmentation as an area of research in Computer Vision. The methodologies that can be employed for performing video object segmentation were covered in detail. Additionally, the existing solutions for solving the problem of video segmentation were discussed along with the benefits and draw back of each technique.

The thesis proposed a solution for solving the given problem and explained each step of the process in detail. The presented solution used graph based methods to solve the segmentation problem. A Linear Spectral clustering based implementation was presented that used color, motion, spatial co-ordinates and saliency to oversegment every frame of the input video into five different superpixel segmentations. The multiple segmentations were merged by constructing a bipartite graph and solving it using Spectral Clustering. Furthermore, a foreground separation model was proposed that used color, optical flow and spatio-temporal saliency to provide an initial segmentation map for each frame. Finally, a CRF was constructed and solved using energy minimization to obtain the final segmentation results.

The presented solution was evaluated on two datasets using average pixel error rate, precision, recall and F1-score as the evaluation measures. The solution was also compared with state-of-the-art methods and both numerical and visual results showed that it outperformed all the techniques on both datasets.

5.2 Future Work

Although, the solution proposed in the thesis produces a lower error than the existing techniques, the technique can still be improved in the future to produce better results. The limitation of the solution to extract multiple objects effectively can be improved by introducing temporal cues that effectively identify the movement of separate objects in different parts of the frame and store their respective information separately. The spatio-temporal saliency implementation used to provide saliency scores to each pixel of the frame can be improved to produce better results by taking camera movement into consideration. Hence, more research can be carried out in the area of video object segmentation to improve the overall results for the problem and produce a lower error than the one produced by the proposed solution.

Bibliography

Achanta, R., Estrada, F., Wils, P., and Süsstrunk, S. (2008). Salient region detection and segmentation. In *ICVS*, pages 66–75.

Bai, X. and Wang, W. (2014). Saliency-svm: An automatic approach for image segmentation. *Neurocomputing*, 136:243 – 255.

Banica, D., Agape, A., Ion, A., and Sminchisescu, C. (2013). Video object segmentation by salient segment chain composition. In *IEEE-ICCVW*, pages 283–290.

Carreira, J. and Sminchisescu, C. (2012). Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE-TPAMI*, 34(7):1312–1328.

Chang, J., Donglai, W., and Fisher, J. (2013). A video representation using temporal superpixels. In *IEEE-CVPR*, pages 2051–2058.

Endres, I. and Hoiem, D. (2014). Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):222–234.

Fukuchi, K., Miyazato, K., Kimura, A., Takagi, S., and Yamato, J. (2009). Saliency-based video segmentation with graph cuts and sequentially updated priors. In *IEEE-ICME*, pages 638–641.

Galasso, F., Cipolla, R., and Schiele, B. (2013). Video segmentation with superpixels. In *ACCV*, ACCV'12, pages 760–774, Berlin, Heidelberg. Springer-Verlag.

Han, J. (2005). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Lee, Y., Kim, J., and Grauman, K. (2011). Key-segments for video object segmentation. In *IEEE-ICCV*, ICCV '11, pages 1995–2002, Washington, DC, USA. IEEE Computer Society.

Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2290–2297.

Li, F., Kim, T., Humayun, A., Tsai, D., and Rehg, J. (2013a). Video segmentation by tracking many figure-ground segments. In *IEEE-ICCV*, pages 2192–2199.

Li, W., Chang, H., Lien, K., Chang, H., and Wang, Y. (2013b). Exploring visual and motion saliency for automatic video object extraction. *IEEE TIP*, 22(7):2600–2610.

Li, Z. and Chen, J. (2015). Superpixel segmentation using linear spectral clustering. In *IEEE-CVPR*, pages 1356–1363.

Li, Z., Wu, X.-M., and Chang, S.-F. (2012). Segmentation using superpixels: A bipartite graph partitioning approach. In *IEEE-CVPR*, pages 789–796.

Liu, Z., Zhang, X., Luo, S., and Le Meur, O. (2014). Superpixel-based spatiotemporal saliency detection. *IEEE TCSVT*, 24(9):1522–1540.

Luque, R. M., Domínguez, E., Palomo, E. J., and Muñoz, J. (2008). A neural network approach for video object segmentation in traffic surveillance. In *Image Analysis and Recognition, 5th International Conference, ICIAR 2008, Póvoa de Varzim, Portugal, June 25-27, 2008. Proceedings*, pages 151–158.

Milan, A., Leal-Taixe, L., Schindler, K., and Reid, I. (2015). Joint tracking and segmentation of multiple targets. In *IEEE-CVPR*, pages 5397–5406.

RJ, D. (2015). The mean shift clustering algorithm. [Online; accessed 2-March-2016].

Shi, J. (2012). Video segmentation by tracing discontinuities in a trajectory embedding. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1846–1853, Washington, DC, USA. IEEE Computer Society.

Tsai, D., Flagg, M., Nakazawa, A., and Rehg, J. (2012). Motion coherent tracking using multi-label mrf optimization. *IJCV*, 100:190–202.

Wang, W., Shen, J., and Porikli, F. (2015). Saliency-aware geodesic video object segmentation. In *IEEE-CVPR*, pages 3395–3402.

Xu, L., Li, H., Zeng, L., and Ngan, K. N. (2013). Saliency detection using joint spatial-color constraint and multi-scale segmentation. *J. Vis. Comun. Image Represent.*, 24(4):465–476.

Zhang, D., Javed, O., and Shah, M. (2013). Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *IEEE-CVPR*, pages 628–635.