A Fire Fly Optimization Based Video Object Co-Segmentation

Ashly Ann Varghese[1], Jemily Elsa Rajan[2]

[1]Dept.of CSE, Caarmel Engineering College, M G University, Kottayam, Kerala, India
Email: ashlyannvarghese91@gmail.com
[2]Assistant Professor in CSE, Caarmel Engineering College, M G University, Kottayam, Kerala, India
Email: jemily86@gmail.com

Abstract- *Video object co-segmentation is a method of segmenting and discovering primary object from multiple videos without human annotation. In this paper, we are presenting a firefly based video object co-segmentation method. Here we are using SIFT flow method for object segmentation and firefly algorithm for object optimization. SIFT flow descriptor discovers the corresponding object using inter-frame motion flow from optical flow. But it will not capture the optimal motion of inter-frame. So we are using Firefly approach in addition to SIFT flow. Firefly approach captures optimal inter-frame motion based on the velocity updation and position of the particle for better results. SIFT flow produces segmented primary objects over the entire video data set. Finally we combine this segmented object to get high quality original videos. The results that we obtained from conducting the experiment show that the Firefly based co-segmentation method achieves high accuracy compared with the existing co-segmentation method.*

Keywords*— *Video object co-segmentation, energy optimization, object refinement, spatio-temporal scale-invariant feature transform (SIFT) flow.

I. INTRODUCTION

Video data is one of the fastest growing resources of publicly available data on the web. Taking such resources for learning and for many other purposes in an easy way is a big opportunity – but equally a big challenge. For leveraging such data sources, algorithm is needed to deal with the unstructured nature of such videos which is beyond today's state-of-the-art. The natural extension of image co-segmentation is the video co-segmentation. Image co-segmentation was introduced by Rother. Later, image based object co-segmentation was proposed, which introduced the use of object proposals for segmenting similar object from image pairs. The idea of co-segmentation was extended to handle multiple object classes and to work in internet collections where all images did not share the same object. Co-segmentation is the task of segmenting "something similar or related" in a given set of images.

Efficient and automatic extraction of objects of interest from multiple videos is very important and also it is a very challenging problem. The appearance or motions of these interested objects maybe exhibit drastically different. In some videos, the foreground appearance or motions are very much different, while possibly low contrast with the background. These challenges cause great difficulties on existing video segmentation techniques. Moreover these methods have some assumption that the background is dramatically distinct from the appearance of object, which is against the situation as we mentioned before. Additionally, these methods lead to unsatisfactory performance.

The extraction of main common object from a set of related videos is known as the video co-segmentation. It utilizes visual properties across multiple videos to infer the interested object with the absence of priori information about foregrounds or videos. There are a few methods designed for this problem till now [5], [6], [9]. These methods have some assumptions on the appearance or motion patterns of foreground. For example, method [5] assumes that the foreground objects from different videos have similar appearance model and similar motion patterns which is distinct from the background and method [6] indicate that the coherent motion of regions and similar appearance are able to conduct the segmentation. Moreover, one general limitation of these approaches [5], [6] is that the set of videos is assumed to be similar or related for foregrounds and backgrounds. The method [9] treats the task of video co-segmentation as a multi-class labelling problem, but its classification results heavily rely on the chroma and motion features. In total, these existing video co-segmentation approaches [5], [6], [9] have two main limitations.

- Both methods abuse motion and appearance based cues and ignore the fact that there are considerable videos with the common object low contrast with the background.
- In both methods, the process of inferring common objects does not completely explore the correspondence of objects from different set of videos, which is essential for the task of video co-segmentation.

These approaches simply assume that the objects are similar in motion patterns or appearance, which is not suitable for the frame that includes objects with large variations in appearance or motion. Besides, there are videos that consist of some frames not containing the common object of the whole video sequence. In some instance, the foreground object switch between video shots or moves out of camera. However, this fact is ignored by most previous work in both video object co-segmentation and segmentation methods. Most of the existing methods assume that the foreground object exists in every frame, and hence they are unable to perform well for this issue.

This work presents a co-segmentation framework for detecting and segmenting out common objects from multiple, contextually related videos without imposing above constraints. In this work, we explore the underlying properties of video objects in three cues: intra-frame saliency, inter-frame consistency and across-video correspondence. Based on these properties, we introduce a firefly approach, which captures the optimal inter-frame motion based on the velocity updation and position of the particle. In this optimization process, we use a spatio-temporal SIFT flow and conventional SIFT flow. Spatio-temporal SIFT flow captures inter-frame motion, which integrates optical flow and conventional SIFT flow captures across-videos correspondence information. This new spatio-temporal SIFT flow produces reliable estimations of common foregrounds over the entire video data set. Finally we combine this segmented object to get high quality original videos.

II. PROPOSED SYSTEM

Our main aim is to jointly segment multiple videos containing a common object in an unsupervised manner. Our algorithm has four main stages: object discovery among multiple videos, object refinement between video pairs, object segmentation on each video sequence and compute optimal flow by using Firefly.

2.1 Object Discovery

For object discovery from multiple videos, our method explores the video dataset structure and associates the global information with the intra-frame information. To discover the common object, we considering three main properties of targeted object: a) intra-frame saliency b) inter-frame consistency and c) across-video similarity. Intra-frame saliency means the foreground pixels should be relatively dissimilar to other pixels within a frame. Inter-frame consistency means the foreground pixels should be more consistent within a video. And across-video similarity means the pixels of foreground should be more similar to other pixels between different videos. We propose a new spatio-temporal SIFT flow algorithm that integrate saliency, SIFT flow and optical flow to explore the comparisons between different videos. To infer the common objects without the assumption that the object must exist in each frame, we designed an object discovery energy function.

Pixel saliency reflects how salient the pixel is, namely, the degree of its dissimilarity within the image. There are many methods in computer vision that concentrate on this topic. We use [8] yet any other saliency methods such as [7] can be incorporated. Let $\mathbf{V} = \{V_1, V_2, ..., V_N\}$ be a set of N input videos. $\mathbf{F}n = \{F_n^1, F_n^2, ..., F_n^i, ...\}$ is a set of frames belong to video V_n. We compute a normalized saliency map M_n^i for frame F_n^i. Based on intra-frame saliency property, the larger value of $M_n^i(\mathbf{x})$, the more likely that the pixel $\mathbf{x} = (x, y)$ belongs to object. Then we build a saliency term $\mathcal{A}_n^i(\mathbf{x})$ to define the cost of labeling pixel $\mathbf{x}$ for foreground ($l_n^i(\mathbf{x}) = 1$) or background ($l_n^i(\mathbf{x}) = 0$) [1].

SIFT flow [3], [4] can be used to build a dense correspondence map across different scenes and object appearances. SIFT flow is shown to accommodate variations. We combine optical flow and local saliency into a superior spatio-temporal SIFT flow to build dense correspondences between pixels in different videos. Through spatio temporal SIFT flow, reliable correspondences $w_{nn'}^{ii'} = (u_{nn'}^{ii'}, v_{nn'}^{ii'})$ between the pixels of frame F_n^i and $F_{n'}^{i'}$ from different videos are established. In other words, pixel $\mathbf{x}$ of frame F_n^i is associated with the pixel $\mathbf{x} + w_{nn'}^{ii'}(\mathbf{x})$ of frame $F_{n'}^{i'}$ .These correspondences indicate whether pixels belong to the common object (even when it may be very salient within its frame).

We establish correspondences between a part of the pixels with high saliency values of one frame and the pixels from the frame of the other video. We select the pixels $\boldsymbol{R}_n^i = \{\mathbf{x} \mid M^i{}_n(\mathbf{x}) > \tau\}$ to explore their correspondences. In experiments, we fixed $\tau = 0.4$. This strategy improves matching accuracy by reducing the disturbance of those un-salient pixels which are very close to background, and it enables our method to remove some salient pixels that do not belong to common object.

Let s_n^i and $s_{n'}^{i'}$ be two SIFT fields of frame F_n^i and $F_{n'}^{i'}$ respectively that we want to match. The terms s_n^{i+1} and $s_{n'}^{i'+1}$ refer to the SIFT fields of frame F_n^{i+1} and $F_{n'}^{i'+1}$ respectively. F_n^{i+1} is the consecutive frame for F_n^i , and $\mathbf{N}_s$ is the spatial 8-neighborhoods of a pixel. Given the set of salient pixels $\boldsymbol{R}_n^i$, the energy function for spatio-temporal SIFT flow [1] is defined as follows:

$$E = E_S + \alpha 1 E_{OS} + \alpha 2 E_{Disp} + \alpha 3 E_{Smooth} + \alpha 4 E_{Sal} \quad (1)$$

where the energy function contains the SIFT based data term (E_S), the optical flow compensated SIFT based data term (E_{OS}), displacement term(E_{Disp}),the smoothness term (E_{Smooth}) and the saliency term (E_{Sal})

It is possible to sample only a few representative frames or sample at a low frame-rate from video to perform the object discovery process instead of using all the frames. We select frame $\mathbf{f}n = \{f_n^1, f_n^2, ..., f_n^k, ...\}$ every other five or ten frames from video V_n to perform object discovery process. For the k-th frame $f_1^k, f_2^k,, ..., f_N^k,$ of every video, we compute their spatio-temporal SIFT flow to capture their correspondence. Next, we calculate the distance of the point **x** of frame f_n^k from its corresponding points of other frames $\aleph(f_n^k) = \{f_1^k, ..., f_{n-1}^k, f_{n+1}^k, ... f_N^k\}$ in SIFT feature [1]:

$$S_n^k(\mathrm{x}) = \frac{1}{|N-1|}\sum_{f_{n'}^k \in \aleph(f_n^k)} || s_n^k(\mathrm{x}) - s_{n'}^k\left(\mathrm{x} + w_{nn'}^{kk}(\mathrm{x})\right) ||_1 \quad (2)$$

We normalize this term with values in [0, 1], where the smaller values indicate greater chance belonging to common object since the smaller distances to corresponding points. Similar to the saliency term, we build a matching term $\mathcal{M}_n^k$ (**x**) to define the cost of labeling pixel **x** for foreground ($l_n^k(\mathrm{x})= 1$) or background ($l_n^k(\mathrm{x})= 0$) [1].

For frame f_n^k , we use the above saliency and matching terms to build an object discovery energy function [1] as:

$$\mathcal{E}_n^k(\mathrm{x}) = \epsilon_1 \mathcal{A}_n^k(\mathbf{x}) + \epsilon_2 \mathcal{M}_n^k(\mathbf{x}) + \mathcal{V}_n^k(\mathbf{x}) \quad (3)$$

where $\mathcal{V}_n^k$ (**x**) for frame f_n^k is the smooth term [1] and $C_n^k(\mathrm{x})$ indicates the color value of pixel **x** in f_n^k , spatial pixel neighborhood $\mathbf{N}_s$ consists of eight spatially neighboring pixels within one frame. This object discovery energy can be efficiently solved by traditional graph cut algorithm [10] and we are able to roughly estimate the common object over the video dataset. The scalars ϵ weight the various terms.

There are many videos that include frames that do not contain the common object. Current video co-segmentation approaches disregard this challenge and assume common object appears in every frame. Our method effectively handles this difficulty. One intuition is that the frames that do not contain the common object are not consistent with the frames that contain the object. Therefore, we further leverage the inter-frame consistency property. Based on (3), we get object-like areas and background areas for each frame. Suppose frame f_n^{k-1}contains the common foreground while f_n^k does not. Their estimated object-like area should be different. We employ Gaussian mixture models (GMM) to characterize the common object appearance. For frame f_n^{k-1} , the GMMs for object-like area and background are defined as $\{GMM_{f_n^{k-1}}^f, GMM_{f_n^{k-1}}^b\}$, respectively. We use an object consistence term $\mathcal{C}_n^k$ (**x**) to measure the consistency of estimated objects in video according to the appearance model of object [1]. Then we add this object consistence term into our object discovery energy function:

$$\mathcal{E}_n^k(\mathrm{x}) = \epsilon_1 \mathcal{A}_n^k(\mathbf{x}) + \epsilon_2 \mathcal{M}_n^k(\mathbf{x}) + \epsilon_3 \mathcal{C}_n^k(\mathbf{x}) + \mathcal{V}_n^k(\mathbf{x}) \quad (4)$$

We set parameter $\epsilon_1 = \epsilon_2 = \epsilon_3 = 50$ for all the test videos in our experiments. Since five or ten frames between frame f_n^{k-1} and f_n^k, the estimated GMM for frame f_n^{k-1} is helpful for identifying whether the frame f_n^k contains the common object.

We use $\mathbf{T}_n^k$ to denote the object-like area in frame f_n^k, and the number of pixels belonging to the object-like area $\mathbf{T}_n^k$ is expressed as $|\mathbf{T}_n^k|$. We consider whether frame f_n^k, contains the common object in case the ratio

$$k_n^k = |\mathbf{T}_n^k| / |\mathbf{T}_n^{k-1}| \quad (5)$$

is relatively large ($k_n^k > 0.2$) and conclude that the foreground object of frame F_n^k is not changed. If this ratio is small, we assume the objects between frame F_n^{k-1} and F_n^k are not consistent. In this case, frame F_n^k is considered to not contain the common object and we set $\mathbf{T}_n^k = \emptyset$. The GMM of the frame f_n^k is set as:

$$GMM_{f_n^k}^f = GMM_{f_n^{k-1}}^f$$

$$GMM_{f_n^k}^b = GMM_{f_n^{k-1}}^b$$

In this way, the GMM for common object is kept consistent across the whole video sequence by ignoring the 'noise' frames. The frames that are detected to not contain the objects in object discovery step, will be not taken into consideration in next object refinement process.

2.2 Object Refinement

We obtain a coarse estimation for the common object in the dataset from the previous step. Based on this, we seek to obtain a more accurate estimation for foreground object in every video. Our intuition is to remove the pixels that are similar to background based on the estimation result. Nevertheless, this also requires determining what foreground would look like. To filter out background pixels we divide the object-like area into sub-regions based on their variations. We utilize spatio-temporal SIFT flow for this purpose.

First, a pair of videos (V_n, $V_{n'}$) is randomly selected from dataset. Their spatio-temporal SIFT flow between frames f_n^k and $f_{n'}^k$ is constructed. Lack of continuities of spatio-temporal SIFT flow field shows the variation of object structure (but not color variation) yet robust to object details. This property of spatio-temporal SIFT flow field is very important. Through the computation of the lack of continuities of spatio-temporal SIFT flow field, depending on the structure variation, we divide the object-like area into a few regions. This enables us to evaluate every part of the object-like area whether belongs to foreground using GMMs.

For each region **t** of object-like area $\mathbf{T}_n^k$, we build the GMM_t^b for background $\check{\mathbf{T}}_n^k$ and the GMM_t^f for the remaining region (object) $\mathbf{T}_n^k \setminus \mathbf{t}$. The likelihood $\rho_n^k(\mathbf{x}_t)$ of pixels $\mathbf{x}_t \in \mathbf{t}$ for foreground is estimated using $\{GMM_t^f, GMM_t^b\}$. We compare the texture of region **t** with the background and object-like area using the local binary pattern (LBP) features, which is used for describing the local spatial structure of an image. To model the texture of foreground and background in frame f_n^k, two normalized histograms (H_t^f and H_t^b) are calculated in LBP domain. For region **t**, the pixels belonging to the object-like area $\mathbf{T}_n^k \setminus \mathbf{t}$ are used for formulating the LBP histogram H_t^f while the pixels belonging to the background area $\check{\mathbf{T}}_n^k$ are sampled for forming H_t^b. Thus the probability $l_n^k(\mathbf{x}_t)$ of pixels $\mathbf{x}_t \in \mathbf{t}$ for foreground is estimated through these two LBP histograms as follows:

$$l_n^k(\mathbf{x}_t) = \frac{H_t^f[\mathbf{x}_t]}{H_t^f[\mathbf{x}_t] + H_t^b[\mathbf{x}_t]} \tag{6}$$

where $H_t^f[\mathbf{x}_t]$ (with value in [0, 1]) indicates the value of histogram H_t^f at pixel $\mathbf{x}_t$.

We combine $\rho_n^k(\mathbf{x}_t)$ and $l_n^k(\mathbf{x}_t)$ and we call it as $o_n^k(\mathbf{x}_t)$, where the term $o_n^k(\mathbf{x}_t)$ denotes the probability of the pixel $\mathbf{x}_t$ for foreground according to both appearance and texture models. If $o_n^k(\mathbf{x}_t) < 0.5$, pixel $\mathbf{x}_t$ will be classified into background. There is no need to consider all of regions $\mathbf{t} \in \mathbf{T}_n^k$. If the area of region **t** is too large or too small, we will ignore these regions. These constraints will take fewer regions into account and enhance the efficiency of our object refinement. In our experiments, the region with $\frac{|\mathbf{t}|}{|\mathbf{T}_n^k|} > 0.5$ or $\frac{|\mathbf{t}|}{|\mathbf{T}_n^k|} < 0.05$ will be directly classified into foreground. After frame f_n^k has been refined, we update $\{GMM_{f_n^k}^f, GMM_{f_n^k}^b\}$ to provide guidance for the following object segmentation process. This object refinement process is executed across video pairs and more correct estimation for foreground object is achieved.

2.3 Object Segmentation

To get the segmentation results of each pixel, a graph-cut based method is employed once the correct estimations for foreground of each video are obtained. Recall our definition of $\mathbf{f}_n = \{f_n^1, f_n^2, ..., f_n^k, ...\}$ is that we select frame $\mathbf{f}_n$ every other five or ten frames from video V_n. After the object refinement process, we get more correct estimation for common object and update the appearance model of the object and background $\{GMM_{f_n^k}^f, GMM_{f_n^k}^b\}$ for frame f_n^k, which can be used to conduct the segmentation in next five or ten frames of f_n^k. For frame F_n^i, we obtain the likelihood of pixel **x** for foreground as $p_n^i(\mathbf{x})$ using our appearance models estimated by its temporally nearest frame of $\mathbf{f}_n$.

For video V_n, we update the labelling $\{l^i_n\}_i$ for all pixels to obtain the final segmentation results through an object segmentation function. This object segmentation function $F_n(\mathbf{x})$ based on spatio-temporal graph by connecting frames temporally can be defined as follows:

$$F_n(\mathrm{x}) = \sum_i \{ \sum_x \mathcal{U}_n^i(\mathrm{x}) + \gamma 1 \sum_{x,y \epsilon Ns} \mathcal{V}_n^i(\mathrm{x,y}) + \gamma 2 \sum_{x,y \epsilon Nt} \mathcal{W}_n^i(\mathrm{x,y}) \} \quad (7)$$

where the set $\mathbf{N}_s$ contains all the 8-neighbors within one frame and the set $\mathbf{N}_t$ contains the backward nine neighbors in pairs of adjacent frames. The parameters γ are the positive coefficient for balancing the relative influence between various terms. The unary term $\mathcal{U}_n^i$ [1] defines the cost of labeling pixel $\mathbf{x}$ with foreground and background according to our appearance model. The pairwise terms $\mathcal{V}_n^i$ and $\mathcal{W}_n^i$ encourage spatial and temporal smoothness, respectively [1]. These two terms favor assigning the same label to neighboring pixels that have similar color. We use binary graph cuts [10] to obtain the optimal solution for (7), and thus get the final segmentation results. The final labelling $\{l_n^i\}_i$ for all pixels in all frames represents a segmentation of the video V_n.

2.4 Compute Optimal Flow by Using Firefly

Firefly captures the optimal inter-frame motion, which is based on the position and velocity updation of the flies. In this optimization process, we use a spatio-temporal SIFT flow and conventional SIFT flow. Spatio-temporal SIFT flow integrates optical flow, which captures inter-frame motion and conventional SIFT flow captures across-videos correspondence information.

Here inter-frame motion is estimated using Firefly optimization [2]. Firefly optimization is a computation method that optimizes a problem by frequently trying to increase a candidate solution with regard to a given measure of quality. This uses a number of flies that set up a swarm moving everywhere in an N dimensional search space looking for the best solution. Every flies takes track of its coordinates in the solution space, which are related with the best solution that is achieved to this point by that flies is called as personal best position (pbest) and the other best value achieved until now by any flies in the neighbourhood of that flies is called as global best position (gbest).

All flies move towards the optimal point with a velocity. Initially all of the flies velocity is assumed to be zero. To improve the searching more effective, initially the object model is projected into a high-dimensional feature space, and then firefly-based algorithm is used to search over the high dimensional space and congregate to some global features of the object. This firefly based algorithm considers even the inter frame motion estimation to speed up the searching procedure. The proposed algorithm can be used to estimate the inter-frame motion at each pixel in a video sequence. This new spatio-temporal SIFT flow generates reliable estimations of common foregrounds over the entire video data set.

III. EXPERIMENTAL RESULTS

The purpose of this work is to co-segment the common objects automatically from related videos with large appearance variations or foreground/ background motion patterns, even when some frames do not contain the common object. There has been very little comparative work to address these problems. We have tested our method on video groups with similar foreground from our dataset. As shown in Fig 1, our co-segmentation results clearly show that our method handles such situation. In order to show the effectiveness of our algorithm on segmenting out the common object from diverse categories, we further test on video groups with large variations on foreground class. We finally evaluate our method on Car2 video groups; these video groups have some frames not containing the common object. We also present quantitative comparisons with previous method. We collect video groups that have same or similar object, and also a pixel-level segmentation ground-truth for each video is available. Our method is evaluated on these video groups and compared to previous cosegmentation method [1].

FIG 1. VIDEO CO-SEGMENTATION RESULTS ON VIDEO GROUPS WITH SIMILAR FOREGROUND OBJECT

Our method utilizes a deeper understanding of the properties of foreground object, including intra-frame saliency, inter-frame consistency and across-video similarity. These properties are further integrated into our optimisation process: spatio-temporal SIFT flow and object discovery energy function, which enables our method to produce more accurate results and outperform the state-of-the-art approaches. Firefly optimization is used which captures the optimal inter-frame motion based on the position and velocity updation of the flies. It is very common that some frames do not contain the object of interest in real video data, such as object moving out of camera or the shot switching effect as shown in Fig 2. But there are very few methods notice this fact, most of co-segmentation methods assume every frame contains the interesting object, which cannot handle these issues well. The proposed method tries to tackle these problems and is evaluated on the newly collected video group: **Car2.** In **Car2** some frames do not contain the foreground object. The frames without common object are naturally indicated by returning an empty labeling through our method.

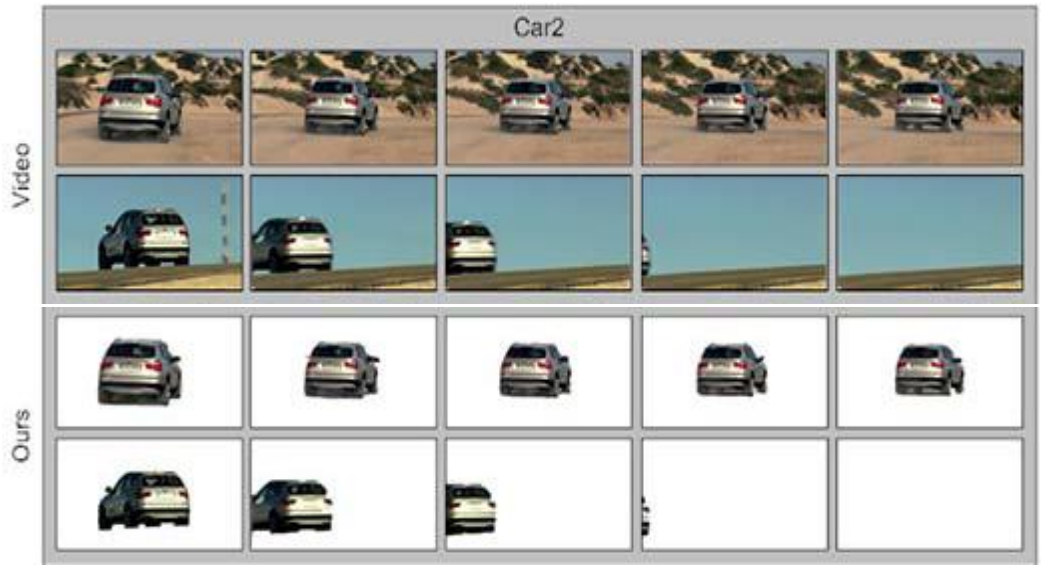

FIG 2. VIDEO CO-SEGMENTATION RESULTS ON VIDEOS WITH SOME FRAMES NOT CONTATINING THE COMMON OBJECT.

Accuracy of the system is calculated with the values of the True Negative, True Positive, False Positive, False negative for actual class and predicted class outcomes. In this research it is the ratio of exact images segmented.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

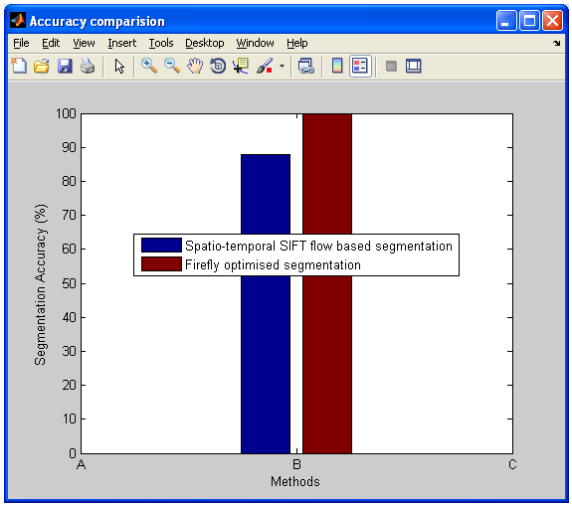

FIG 3 SPATIO-TEMPORAL SIFT FLOW BASED SEGMENTATION VS FIREFLY OPTIMIZED SEGMENTATION

Fig 3 shows the comparison of Spatio-temporal SIFT flow based segmentation and Firefly optimized segmentation in terms of accuracy. In the x axis, the two methods are plotted and in the y axis, accuracy in percentage is plotted. From the graph, it is clear that Firefly optimized segmentation has more accuracy while Spatio-temporal SIFT flow based segmentation has less accuracy which means that the proposed approach has better performance in terms of accuracy.

IV. CONCLUSION

The proposed video co-segmentation method discovers the common object over an entire video dataset and segments out the objects from the complex backgrounds. In this system, we use a spatio-temporal SIFT flow to segment the video object. The next step is optimization of the SIFT output for better results, for that we are using Firefly optimization algorithm, Firefly optimization captures the optimal inter-frame motion based on the position and velocity updation of the flies. Finally the segmented objects are combined to form a high quality video. The results of experiment show that the proposed system achieves high accuracy compared with the existing system. Here, spatio-temporal SIFT flow algorithm that integrates saliency, SIFT flow and optical flow to explore the correspondences between different videos. In future various optimization algorithms like ant colony, pso are used for measuring optical flow.

REFERENCES

[1] Wenguan Wang, Jianbing Shen, Xuelong Li,and Fatih Porikli "Robust Video Object Cosegmentation," *IEEE Trans.* On Image Processing, Vol. 24, No. 10, October 2015

[2] Hema Banati and Monika Baj," Fire Fly Based Feature Selection Approach", in IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011I.S.

[3] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[4] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in Internet images," in *Proc. IEEE CVPR*, Jun. 2013, pp. 1939–1946.

[5] J. C. Rubio, J. Serrat, and A. López, "Video Co-segmentation," in *Proc. ACCV*, 2012, pp. 13–24.

[6] D.-J. Chen, H.-T. Chen, and L.-W. Chang, "Video object cosegmentation," in *Proc. ACM Multimedia*, 2012, pp. 805–808.

[7] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[8] B. Jiang, L. Zhang, H. Lu, M.-H. Yang, and C. Yang, "Saliency detection via absorbing Markov chain," in *Proc. IEEE ICCV*, Dec. 2013, pp. 1665–1672.

[9] W.-C. Chiu and M. Fritz, "Multi-class video Co-segmentation with a generative multi-video model," in *Proc. IEEE CVPR*, Jun. 2013, pp. 321–328.

[10] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001