Real-time Referred Video Object Segmentation (RVOS)-based Video Summarization using Multimodal Transformers

Shashwat Srivastava (2021EEB1210)
Ranjeet Singh (2021EEB1203)

Department of Electrical Engineering
Indian Institute of Technology Ropar

Under the guidance of Dr. Santosh Kumar Vipparthi

May 12, 2025

Abstract

Video summarization aims to produce short, abstracted summaries of lengthy videos, capturing the most salient information. Approaches to date have been to rely on low-level features or unsupervised clustering. The current work explores an alternative direction by applying Referring Video Object Segmentation (RVOS) to query-based video summarization. We utilize the Multimodal Tracking Transformer (MTTR), a state-of-the-art recent deep neural architecture for RVOS, as the foundation. The aim is to provide a system that initially segments and detects objects mentioned in a natural language query using MTTR, and subsequently produces a summary based on segmented frames. We employ a two-stage pipeline: RVOS with an MTTR model pre-trained on MS-LSI, and two alternative summarization approaches – uniform temporal sampling and a feature-based analysis method. The system efficiently processes input videos, performs text-queried segmentation, and produces annotated clips and analytical results, demonstrating the potential of RVOS for semantic video summarization.

1 Introduction

The increase in video content demands efficient ways to scan and understand lengthy videos. Video summarization is used to reduce this burden by creating short versions with the most important content. What constitutes "essential" content is subjective and task-dependent. Most traditional summarization methods are based on visual activity, motion, or scene change and do not take into account semantic understanding or user-oriented interest.

Referring Video Object Segmentation (RVOS) is a promising method. RVOS aims to segment a target object instance in a video based on a natural language description (query). This facilitates high-fidelity, semantic video content understanding in terms of user interest. Models like the Multimodal Tracking Transformer (MTTR) [1] have posted outstanding performance in segmenting and tracking text-referred objects with precision.

This paper studies the application of RVOS within the video summarization pipeline. The overall aim is to implement an "RVOS-Summarizer" that takes advantage of the semantic anchoring supported by MTTR. For a long video and text query (e.g., "the person kicking the ball"), the system initially applies MTTR to slice the specified object over frames. After segmentation, a summarization module processes segmented frames to form a short video clip. Query-based approach makes it possible to generate highly related summaries pertaining to specific objects or actions of concern. We use two summarization approaches based on the

output of MTTR: a uniform sampling baseline approach and a more complex feature analysis technique.

2 Work Done

The project involved several key implementation stages:

1. **MTTR Model Setup:** The groundwork was set for the Multimodal Tracking Transformer (MTTR). This involved creating dependencies and loading a pre-trained checkpoint of the MTTR model for the RVOS task. Its performance was compared with standard benchmarks, and results were explored in the following sections.

2. **Video Processing Pipeline:** An automated pipeline was constructed utilizing Python libraries like yt_dlp and MoviePy to download YouTube videos, cut them to defined lengths, and make them ready for additional processing by the MTTR model.

3. **RVOS Inference:** The pre-trained MTTR model was employed to run inference over the segmented video clips. The model returns segmentation masks for the target object for each frame (or frames) based on a text query. These masks were then overlaid on single original frames to render an annotated video as the RVOS result.

4. **Summarization Module Implementation:** Two distinct summarization approaches were implemented downstream from the MTTR module, operating on the annotated video output:

   - **Uniform Sampling Summarizer:** A simple approach that samples frames at a constant interval based on a user-specified summary percentage. This is a baseline temporal summarization method.
   - **Feature-based Analyzer:** A sophisticated module was implemented to carry out a detailed analysis of the video content post-RVOS. Utilizing libraries such as OpenCV and audio processing libraries, it computes different metrics frame by frame:
     - Optical Flow Magnitude (Motion analysis)
     - Scene Change Scores (Detecting cuts/shifts)
     - Mask Coverage (Ratio of the frame covered by the object delineated)
     - Frame Shannon Entropy (Complexity/Information content)
     - Audio Onset Strength (Detection of audio events)

     This module generates plots for all metrics, providing information about the video dynamics and the behavior of the referred object.

5. **Integration and Output Generation:** All components were integrated into end-to-end scripts that can accept a video URL, query, and parameters, and output the annotated full clip, the uniformly sampled summary clip, and a collection of analytical plots.

3 Methodology

Our proposed RVOS-Summarizer follows a two-stage methodology. First, RVOS is performed using MTTR, and second, a summarization module processes the output.

3.1 Referring Video Object Segmentation (RVOS) with MTTR

We employ the Multimodal Tracking Transformer (MTTR) [1] for the RVOS task. MTTR models RVOS as a sequence prediction problem, leveraging the power of Transformers to process visual and textual information jointly.

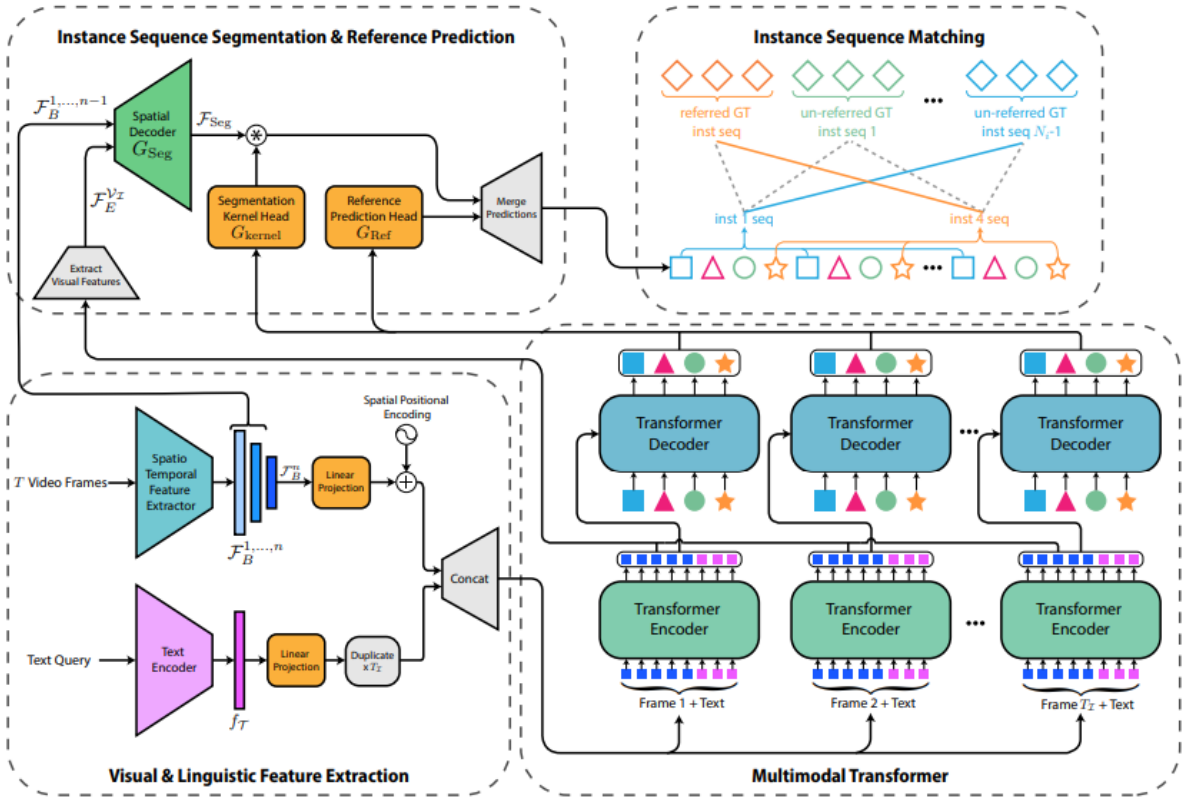


Figure 1: Detailed architecture of the Multimodal Tracking Transformer (MTTR) [1]. It uses spatio-temporal and text encoders, followed by a multimodal Transformer to predict instance sequences.

As shown in Figure 1, the MTTR system takes as input a series of video frames and a text query. It uses separate encoders to extract visual (spatio-temporal) and textual features and fuses them afterwards using a multimodal Transformer encoder-decoder architecture. Its strongest point is that it is able to explain object, text, and temporal sequence relationships. The system produces several sequences of instance predictions (segmentation masks over time) for the entities in the video. A specific prediction head determines which sequence is for the text query. For our project, we use a pre-trained MTTR model, with the weights frozen during summarization.

3.2 Video Summarization Module

Once MTTR generates the segmentation masks for the referred object (producing an annotated video), the summarization module takes over. We implemented two strategies:

3.2.1 Uniform Sampling Summarizer

This is a simple baseline approach. After MTTR produces the annotated video (original frames + overlaid masks), this module once more opens the annotated clip. Based on a user-specified '–summary_percent' (e.g., 50%), it calculates a sampling interval N. It then iterates over the

frames and writes each Nth frame to a new output file, thereby producing a temporally reduced version of the annotated clip.

3.2.2 Feature-based Analyzer

This approach attempts to generate a better summary by means of content feature analysis. The current framework focuses on the computation and representation of such features rather than selecting segments directly for a summary. The overall process is outlined in Algorithm 1.

Algorithm 1 Feature-based Analysis

Require: Annotated Video V_{ann} (Frames $F_1..F_N$ with masks $M_1..M_N$), Audio Track A
Ensure: Feature vectors $Feat_{motion}, Feat_{scene}, Feat_{mask}, Feat_{entropy}, Feat_{audio}$

```
Initialize empty feature vectors
Extract audio waveform from A
Calculate audio onset strength Feat_audio from waveform
for i = 1 to N do
    Calculate mask coverage c_i = Area(M_i)/Area(F_i)
    Append c_i to Feat_mask
    Calculate Shannon entropy e_i of F_i
    Append e_i to Feat_entropy
    if i > 1 then
        Calculate optical flow flow_{i-1→i} between F_{i-1} and F_i
        Calculate mean flow magnitude m_i = Mean(|flow_{i-1→i}|)
        Append m_i to Feat_motion
        Calculate histogram distance d_i = Bhattacharyya(Hist(F_{i-1}), Hist(F_i))
        Append d_i to Feat_scene
    end if
end for
Generate plots from feature vectors
```

This analysis provides rich information that could potentially be used to select keyframes (e.g., frames with high motion, large mask coverage, or coinciding with audio events) in a future implementation.

3.3 Overall System Flow

Figure 2 illustrates the complete workflow of the system, showing the two paths for summarization/analysis after the initial RVOS stage.

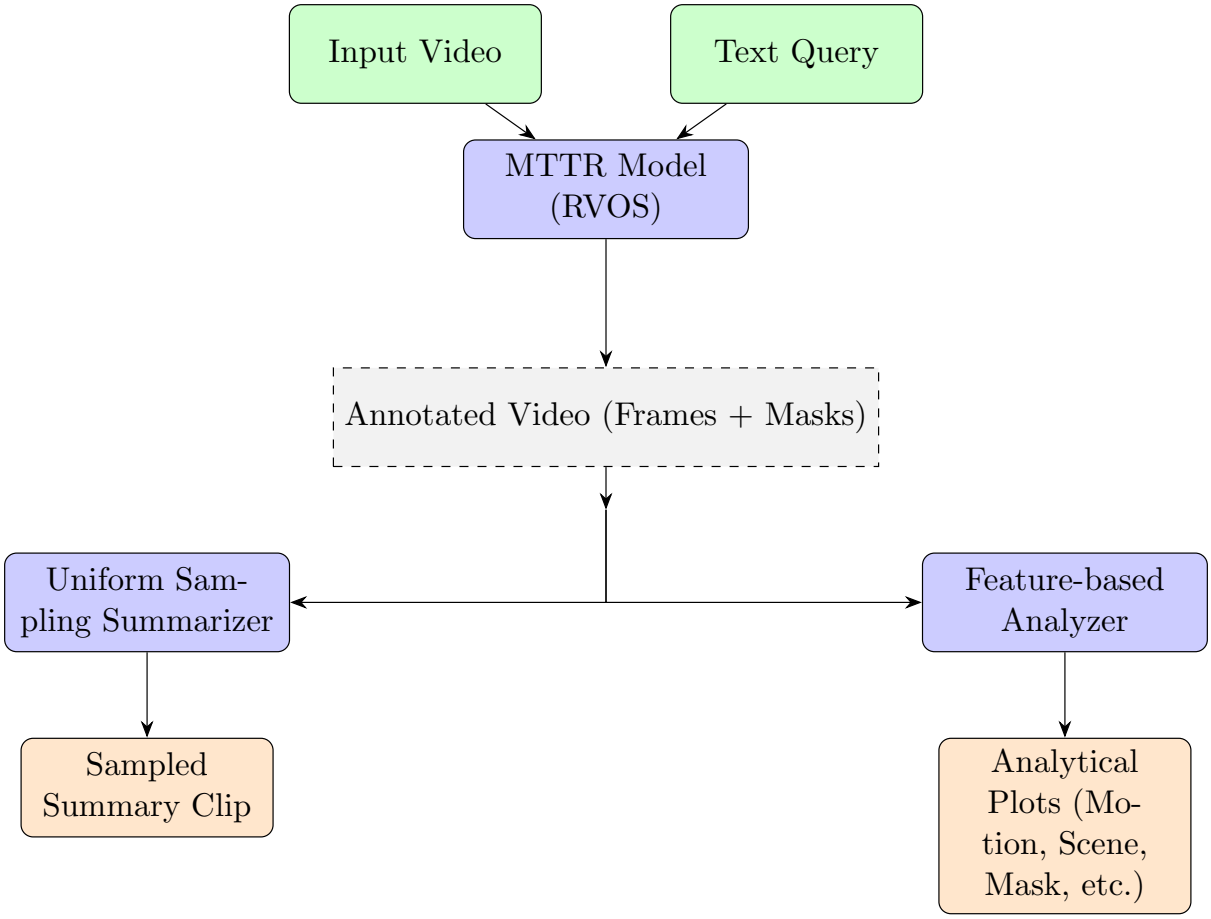


Figure 2: Overall block diagram of the RVOS-Summarizer system. Video and text query are processed by MTTR for segmentation. The output feeds into either a Uniform Sampling summarizer or a Feature-based Analyzer.

4 Observations

Key observations from the implementation and testing include:

- **MTTR Effectiveness:** The pre-trained MTTR model proved to be highly effective in delineating the object described by the text query, irrespective of complex backgrounds or objects that appear similar (See Figure 3). Its capability to follow the referred object through various frames is essential in giving consistent input to the process of summarization.
- **Query Specificity:** The quality and relevance of the summary heavily depend on the specificity and clarity of the input text query. Vague queries might lead to incorrect segmentation or tracking failures.
- **Uniform Sampling Limitations:** Although uniform sampling is easy and efficient, it has the potential to overlook short but significant events concerning the object being queried if such events occur between instances of sampling. This method achieves temporal compression but lacks semantic prioritization.
- **Feature Analysis Insights:** The feature plot-like representations that the Feature-based Analyzer computes (see Figure 4) are very informative. In particular, optical flow increases are tied to object-in-question actions and mask coverage is a measure of object

importance. Large scene change scores clearly identify cuts. Such features have potential to guide a more advanced keyframe selection algorithm.

- **Computational Cost:** The RVOS phase with MTTR is expensive computationally. Optical flow computation, along with feature analysis, adds a lot of overhead. Real-time deployment would require additional model distillation or optimization.

5 Limitations of Existing Works

Traditional video summarization techniques often suffer from:

- **Semantic Focus Deficit:** Techniques that rely only on low-level features (motion, color) cannot favor content by meaning or user interest.
- **Generic Summaries:** Unsupervised techniques produce generic summaries that may or may not be specific to certain informational requirements of specific users.
- **Complexity in RVOS Pipelines:** While there is certainly some existing RVOS literature, it is generally in the form of sophisticated multi-stage pipelines with distinct detection, tracking, and language grounding modules, rather than the more end-to-end form of MTTR. It is difficult to combine these sophisticated pipelines with summarization.

Our approach, leveraging MTTR, addresses the semantic focus limitation by directly incorporating language queries. However, our current implementation also has limitations:

- The Feature-based Analyzer presently only produces plots; it does not employ them to pick summary frames (as demonstrated in Algorithm 1). Adding the selection logic is future work.
- Upstream RVOS model (MTTR) performance is most critical. Segmentation or tracking errors will inevitably detract from the summary.
- The computational cost, as already suggested.

6 Results

6.1 RVOS Performance (MTTR)

The initial MTTR model was validated using the A2D-Sentences benchmark. The performance of the different MTTR configurations, compared to other state-of-the-art approaches (as reported in the initial paper [1]), is shown in Table 1. MTTR operates significantly better, proving its effectiveness in the basic RVOS task.

Table 1: MTTR performance on A2D-Sentences [2], extracted from [1].

	Precision @ K					**IoU**		
Method	50%	60%	70%	80%	90%	Overall	Mean	mAP
MTTR (w = 8, ours)	72.1	68.4	60.7	45.6	16.4	70.2	61.8	44.7
MTTR (w = 10, ours)	75.4	71.2	63.8	48.5	16.9	72.0	64.0	46.1

6.2 Qualitative Summarization Results

Figure 3 displays a sample input frame and the resulting output frame after processing by our system. MTTR ran RVOS on the text query "a guy dancing." The segmented object is easily distinguished with a colored mask in the output frame.

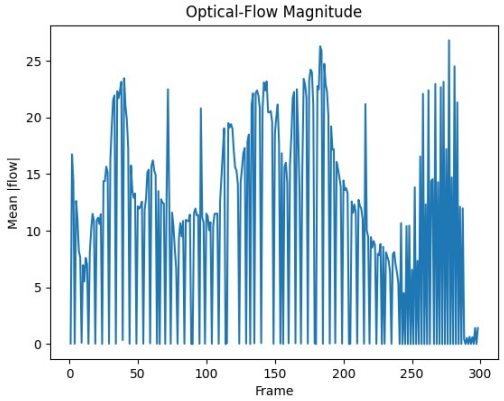
(a) Input video frame

(b) Output frame with RVOS mask

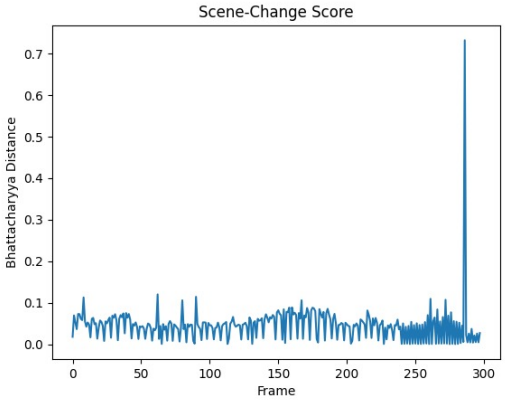
Figure 3: Example input and output frame from the RVOS-Summarizer pipeline. The output shows the object referred to by the text query "a guy dancing" segmented by MTTR.

6.3 Feature-based Analysis Plots

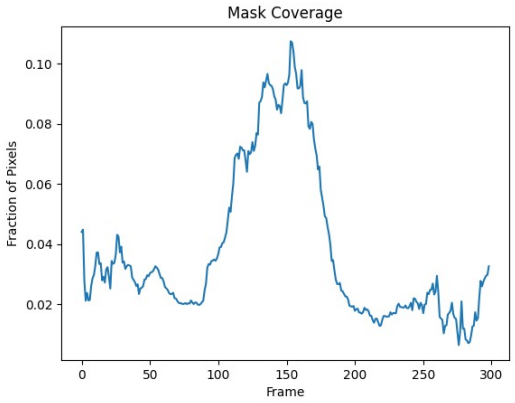
The Feature-based Analyzer generates a suite of plots visualizing video characteristics over time, as calculated by Algorithm 1. Figure 4 presents examples of these plots for a sample processed video clip. These plots provide quantitative data supporting the observations made earlier regarding video dynamics and object behavior.



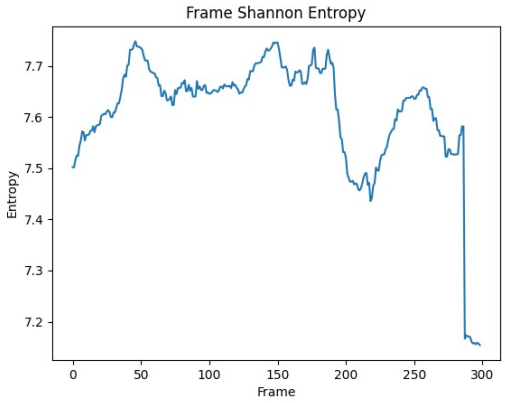
(a) Mean Optical Flow Magnitude per Frame



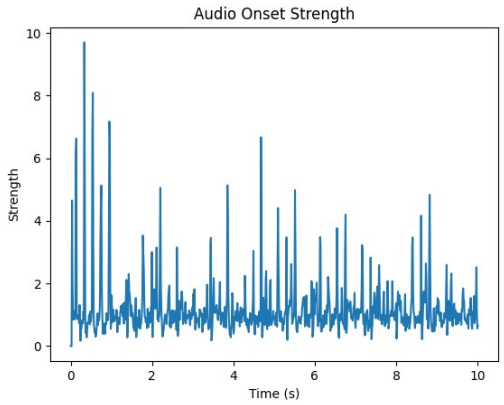
(b) Scene Change Score (Bhattacharyya Distance)



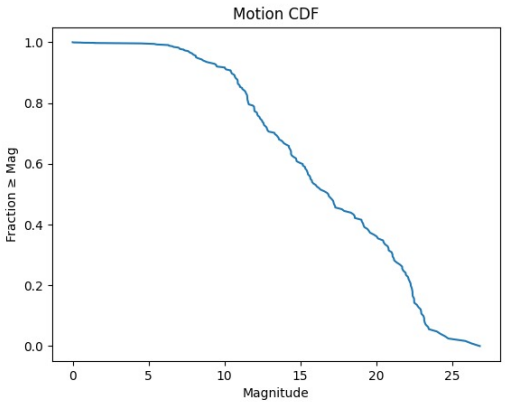
(c) Mask Coverage (Fraction of Pixels)



(d) Frame Shannon Entropy



(e) Audio Onset Strength



(f) Motion Distribution Histogram

Figure 4: Examples of analytical plots generated by the Feature-based Analyzer for a sample video clip after RVOS.

7 Conclusion

This paper successfully designed and deployed an end-to-end video summarization system that utilizes Referring Video Object Segmentation based on the MTTR model. A two-phase pipeline was constructed, which initially carries out query-specific object segmentation followed by sum-

marization. Two summarization approaches were investigated: a baseline uniform sampling scheme and an analysis module based on features that delivers detailed insights into the dynamics of the video content concerning the referred object. The system proves the viability and promise of utilizing state-of-the-art RVOS models such as MTTR to generate semantic, query-specific video summaries. Future research could involve designing sophisticated keyframe selection algorithms based on the recovered features (demonstrated by Algorithm 1), enhancing the pipeline towards real-time efficiency, as well as conducting user studies to test the perceptual quality of the generated summaries.

Acknowledgements

We would like to express our sincere gratitude to our project guide, **Dr. Santosh Kumar Vipparthi**, for his invaluable guidance and support throughout this work. We also thank **Amitabh Tripathi Sir** for his helpful discussions and insights during the project.

References

[1] Adam Botach, Evgenii Zheltonozhskii, Chaim Baskin. *End-to-End Referring Video Object Segmentation with Multimodal Transformers.* arXiv preprint arXiv:2111.14821, 2022.

[2] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, Cees G M Snoek. *Actor and action video segmentation from a sentence.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.