Filip Ilic, BSc

Combining Top-Down and Bottom-Up Processes to Extract Space-Time Volumes of Interest from Video

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Pinz, Axel, Ao.Univ.-Prof. Dipl.-Ing. Dr.techn.

Institute for Electrical Measurement and Measurement Signal Processing

Graz, January 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

As top-down based approaches of object recognition from video are getting more powerful, a structured way to combine them with bottom-up grouping processes becomes feasible. When done right, the resulting representation is able to describe objects and their decomposition into parts at appropriate spatio-temporal scales. We propose a method that uses a modern object detector that is able to provide rough masks of the salient object in question, and a dense optical flow estimator as top-down anchors to focus on salient structures in video. From these structures we extract space-time volumes of interest by smoothing in spatio-temporal Gaussian scale space that guides bottom-up grouping with Simple Linear Iterative Clustering. The resulting novel representation enables us to analyze and visualize the decomposition of an object into meaningful parts, and to describe the relative motion between object components, while preserving temporal object continuity. We propose two methods: one that relies on 'dense' top-down anchors, i.e. in every frame of the video, and a second one that only uses these anchors sparsely but also uses a region tracker to compensate for the loss of dense masks. We evaluate the segmentation performance of our two approaches on a video dataset that provides ground truth segmentation; we show that, while our method cannot outperform the state-of-the-art method, it does produce some results that are qualitatively better, with the additional benefit of modeling individual components of objects based on their appearance and motion. Moreover, our method is able to extract a representation at various scales, which might be useful for a variety of tasks. We also present a novel way to visualize interactions of object parts in space-time, which can be used to highlight the trajectory that an object is tracing in space-time.

Kurzfassung

Da Top-Down basierte Ansätze der Objekterkennung in Videos immer leistungsfähiger werden, ist es möglich sie mit Bottom-Up Gruppierungsmechanismen zu kombinieren um Objekte und deren Zerlegung in Einzelteile zu beschreiben. Unsere Methode erlaubt es uns Objekte und Objektteile, in geeigneten räumlichen und zeitlichen Maßstäben (Scales) zu repräsentieren. Im speziellen verwendet unsere Methode einen modernen Objektdetektor, der grobe Masken der auftretenden Objekte bereitstellt, und generieren dichten optischen Fluss um markante Objekte und Objektteile zu repräsentieren. Aus dem Aussehen und dem optischen Fluss des Objekts lassen sich Strukturen (Räumlich-Zeitliche Volumina von Interesse) extrahieren. Diese Stukturen werden durch gruppieren mit Simple Linear Iterative Clustering im Gauß'schen Scale Space erzeugt. Daraus ergibt sich die Zerlegung eines Objekts in sinnvolle Teile, die es uns ermöglicht Objekttrajektorien und die Relativbewegung zwischen Objektteilen zu analysieren und zu visualisieren.

Wir evaluieren zwei Methoden, welche die selben Mechanismen verwenden um Räumlich-Zeitliche Volumina von Interesse (Space-Time Volumes of Interest - STVIs) zu extrahieren, aber die Eingangsvideos anders vorverarbeiten: Die erste Methode verwendet 'dichte' Objektmasken, d.h. Masken in jedem Frame, wohingegen die zweite Methode Masken nur sporadisch verwendet und zusätzlich einen Objekttracker benutzt. Wir evaluieren beide Methoden auf einem Videodatensatz der Ground-Truth Segmentierungen enthält. Wir zeigen, dass sich unsere Methode dafür eignet einzelne Komponenten von Objekten zu modellieren, obwohl sie keine Verbesserung der Segmentierungsleistung erreicht. Darüber hinaus ist unsere Methode in der Lage, Objekte und Objektteile in verschiedenen Maßstäben zu repräsentieren, was für eine Vielzahl von Aufgaben nützlich sein kann. In unseren Visualisierungen zeigen wir auch die Interaktionen von Objekten und Objektteilen, was es uns ermöglicht, die Dynamik einer Scene zu veranschaulichen.

Contents

1 Introduction

> *What we see changes what we know.*
> *What we know changes what we see.*
>
> Jean Piaget

The straightforward extension from image based detection, representation and recognition of objects to the domain of videos is to treat each frame in the video individually, essentially ignoring the temporal correlation between two adjacent frames. While easy to implement, the fundamental flaw with such strategies is revealed when, from a perceptual point of view, two adjacent frames that look almost identical yield vastly different results w.r.t. to the detected objects. Current state-of-the-art methods that operate on video data are very powerful and enable us to detect object proposals, even segment the object with a mask in individual frames and try to establish a temporal correspondence between the individual frames, see e.g. [7]. These and similar approaches that build on top of Deep Convolutional Neural Networks (CNN) can detect and recognize object categories and track individual instances through the temporal sequence. However, they may not suffice when we aim at more complex analysis of e.g. interactions between objects, the decomposition of a global motion into specific motion patterns, or the decomposition of whole objects into their individual parts. To address these issues, it will be helpful to rely on a more explicit, tangible representation of what, where, when, and how things happen in video.

This thesis presents a general method to extract an *explicit* representation of scene dynamics from monocular video that offers a choice of the desired level of detail. We can decompose objects into spatio-temporally meaningful parts and switch between a viewer-centered, and object-centered representation. We call our novel representation ‘Space-Time Volume of Interest’, STVI. We

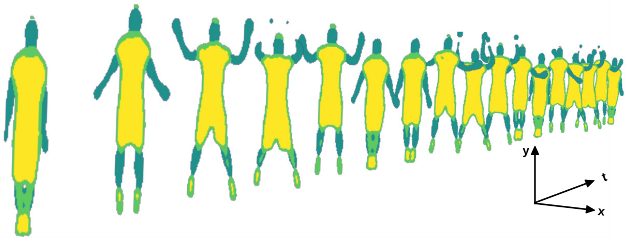


Figure 1.1: Viewer-centered Space-Time Volume of Interest extracted fully automatically from a UCF-101 jumping jack video, showing the decomposition into its individual parts (torso, head, limbs, feet). The extracted Space-Time Volume is visualized by taking sparse slices along the temporal dimension.

demonstrate the descriptiveness of STVIs in two teaser figures for a UCF101 [30] ‘jumping jack’ video sequence. Fig. 1.1 presents the viewer-centered decomposition of a human performing jumping jacks into trunk, limbs and head. The viewer-centered perspective supports the analysis of object trajectories and, when several objects share a scene, of object interactions. Fig. 1.2 shows the object-centered perspective of the same jumping jack action, with increasing level of detail from left to right. In general, the object-centered perspective supports the analysis of relative motion of body parts w.r.t. the global up-and-down pattern of the whole body.

Besides its theoretical appeal, such a representation at the individual object level will be highly desired in many applications of video understanding, e.g. to analyze motion patterns in sports or rehabilitation, to find similarities between specific actions and motions, to categorize a particular action based on features extracted from STVIs, or to describe complex scene dynamics in cluttered scenes with many moving objects. Previous efforts to represent objects in space-time have been either restricted to simple human actions that were extracted from rather clean video footage [9], or have tried to solve space-time correspondence and scale in a bottom-up fashion, see e.g. temporal superpixels (TSPs) [6]. The most consistent and theoretically grounded framework is Gaussian scale space representation as developed by Lindeberg [17], Laptev [13] and others [31]. Because TSPs and scale space are built in a bottom-up, fine-to-coarse manner, neither can explicitly address motion in video at the object level and relative motion between object parts.

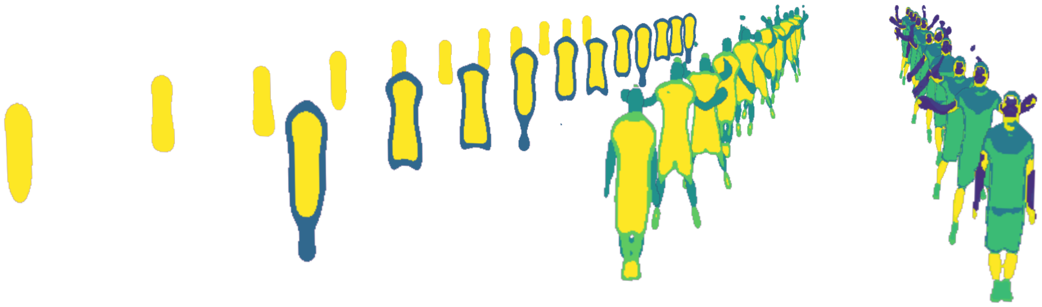


Figure 1.2: Object-centered representation of the jumping jack STVI from Fig. 1.1 at different spatio-temporal scales, showing increasing levels of detail from coarse to fine scales.

With the recent paradigm shift towards deep convolutional architectures for image and video analysis, we see many excellent solutions for image and object recognition, as well as two-stream architectures for video analysis that use appearance and motion information. These approaches offer close to human-like performance in object and action recognition, and object tracking; the stunning performance, however, comes at a price: Besides the huge effort required to train these networks, what they learn is represented implicitly in the millions of parameters that are tuned. Therefore, it is hard to perform explicit reasoning on top of the output of state-of-the-art ConvNets, as well as to switch between various spatio-temporal levels of detail in post-processing.

Problem statement & our contribution

This thesis addresses the problem of building a representation of salient objects from video, where the representation is able to capture the object of interest throughout its motion in the video. The object has to be represented in a spatio-temporally consistent manner, meaning that the extracted representation should not change abruptly w.r.t adjacent frames, but smoothly, and that the representation should be local in nature. From the locality of the representation, it follows that the represented objects should be decomposable into meaningful parts, at object-specific scales.

Our contribution offers a solution that builds on top of current state-of-the-art ConvNets for object segmentation and dense optical flow. These two ‘top-down anchors’, which can be thought of as a attention mechanism, guide the selection towards salient objects in the scene. We combine this ‘high-level’ information and select an appropriate spatio-temporal scale for the object of interest, which is used to drive a bottom-up grouping process for each object in the scene. Summarizing, our STVIs

1. bridge the gap between bottom-up and top-down approaches in representing video data,
2. combine ‘objectness’ and temporal consistency by using an object detector and optical flow, and
3. provide bottom-up, spatially and temporally scaled volumes that are constrained by these top-down anchors.

This novel representation provides the required richness to describe general object- and motion-dynamics in a scene.

2 Related Work

Superpixels Superpixels, as introduced by Ren and Malik [27], are non-overlaping groups of pixels. They are grouped together based on a similarity measure, which usually includes their color and spatial proximity. Superpixels are useful as they reduce the number of image elements, i.e. thousands of pixels, to be processed by orders of magnitude. There is a multitude of different superpixel algorithms that produce a superpixelation such as Felzenszwalb's algorithm [8], Quickshift [32], and SLIC superpixels [1]. These algorithms require a few parameters, which determine the compactness, size, and number of generated superpixels for a given image. SLIC superpixel segmentation performance outperformes other methods [2], and its rather straight-forward implementation allows for an easy examination and understanding of how changes in the parametrization affect the resulting superpixelation. An example of superpixels of varying granularities generated by SLIC is shown in Fig. 2.1.

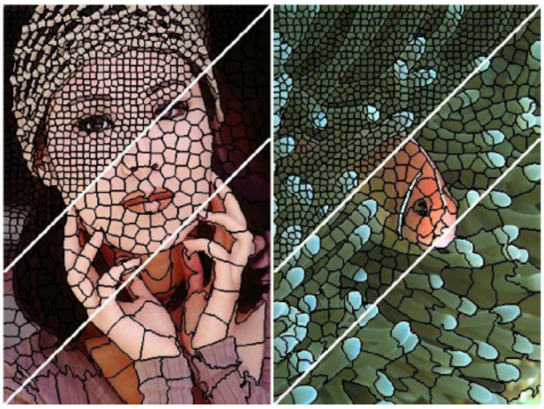


Figure 2.1: The parametrization of superpixel algorithms allows for different granularities of the generated oversegmentation. Figure is taken from the original work by Achanta et al. [1].

Chang et al. [6] extend this notion of superpixels for images to include the temporal domain. Their Temporal Superpixels (TSPs) are generated by assuming a Gaussian process so that superpixels of close spatial proximity exhibit a similar motion pattern. TSPs are able to track parts of an object over multiple frames as shown in Fig. 2.2 from their original paper. Note however that their approach has issues when dealing with occlusions of superpixels, from which it cannot recover. Also, keep in mind that this approach does not differentiate between salient objects and background. It models the whole image sequence with TSPs, with the considerable downside that it is inherently not a representation of objects in the scene, but rather of regions that exhibit similar motion and appearance.

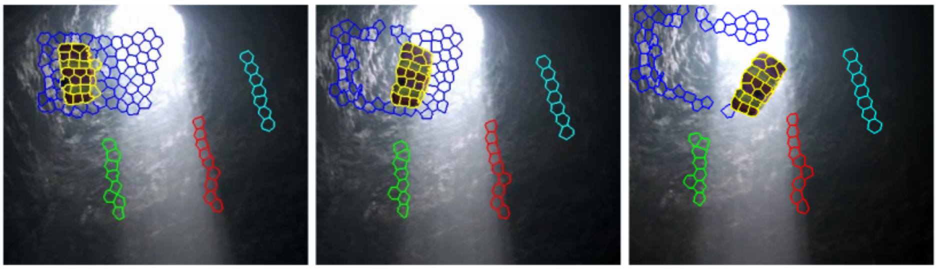


Figure 2.2: Individual Temporal Superpixels track the paragliding chute in the scene, while also modeling the background with superpixels that stay the same throughout the sequence. Figure from the original work by Chang et al. [6].

Motion cues A similar principle is employed by Levinshtein et al. [14]. A superpixel segmentation is produced by TurboPixels [15], and the Lucas-Kanade optical flow algorithm is used to estimate the motion of individual superpixels. From this information they build a spatio-temporal volume containing a superpixel graph with edges 1) along each superpixel in each frame to model the spatial coherence, and 2) edges between the frames determined by the optical flow. By using graph cuts on this spatio-temporal volume, it is possible to group superpixels into coherent groups, that can represent objects implicitly by detecting regions that exhibit similar motion and appearance patterns. In many applications this is not desirable; an *explicit* representation of objects and their motion is preferred.

Wang and Schmid [33] propose a different method to improve trajectories which are based on optical flow, by removing the clutter in the optical flow in the background, and estimating the camera motion. From these trajectories, they are able to extract features which can then be used for action recognition. A similar approach that uses optical flow to estimate point trajectories is proposed by Ochs and Brox [23]. Furthermore, they also employ superpixels, which in combination with sparse point trajectories can be used to create dense point trajectories. An example that shows the various stages of the method (input image, superpixel segmentation, sparse point trajectories, final segmentation) is shown in Fig. 2.3.

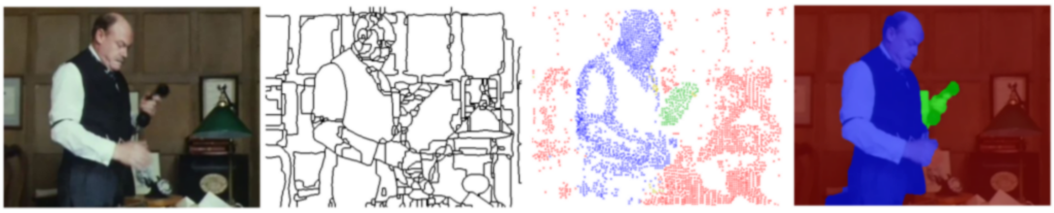


Figure 2.3: Turning point trajectories into dense regions. Figure from the original paper by Ochs and Brox [23].

Focus on salient objects This blindness to objects is contrasted by approaches such as [29] presented by Seguin et al. who propose to use a object detector to determine the coarse regions of interest in which objects are located. Their approach is to formulate the extraction of connected salient spatio-temporal regions as a constrained minimization problem. They introduce a grouping term which uses TSPs [6] from which a superpixel graph is built that enforces spatial and temporal consistency, a discriminative term that models background vs. foreground pixels, and a number of constraints to ensure a coherent representation. An example taken from their work is seen in Fig. 2.4. The visualization shows how the object tracks of four people are modeled as coherent space-time objects with well defined boundaries.

In the domain of space-time volumes and action recognition early work was done by Gorelick et al. [9]. They are able to create an 'action shape', see Fig. 2.5, from clean silhouettes of people performing actions such as running, jumping, walking, and others. These action shapes impressively visualize complex actions patterns, albeit implicitly (the object is not split into individual parts, with

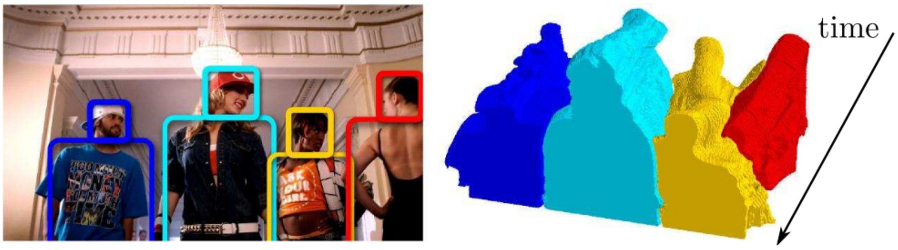


Figure 2.4: Instance-level video segmentation from object tracks. Figure from the original work by Seguin et al. [29]

individual partial action shapes), and create a representation from which they are able to classify the performed actions. To perform this classification they use a combination of local and global features extracted from the action shape.

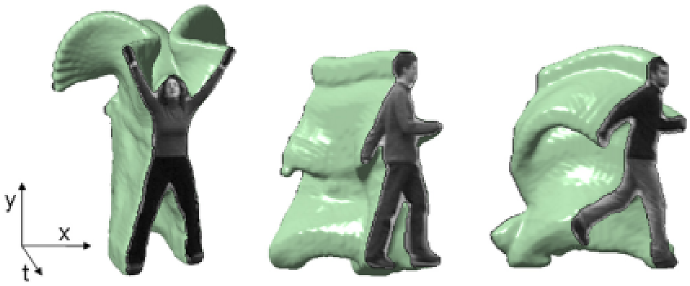


Figure 2.5: Action shapes which visualize the spatio-temporal extent of a performed action, extracted silhouettes from video. Figure from the original work of Gorelick et al. [9].

When talking about representing salient objects, it is important to mention frameworks that enable the recognition of objects in a scene. Such an object detector which provides per object bounding boxes and masks is proposed by He et al. [10]. An example of the segmentation performance of their proposed approach, Mask R-CNN, is shown in Fig. 2.6. While Mask R-CNN has not been (yet) extended to the video domain, it is feasible to individually stack Mask R-CNN masks together from which a similar video representation as shown in Fig. 2.4 could be achieved.

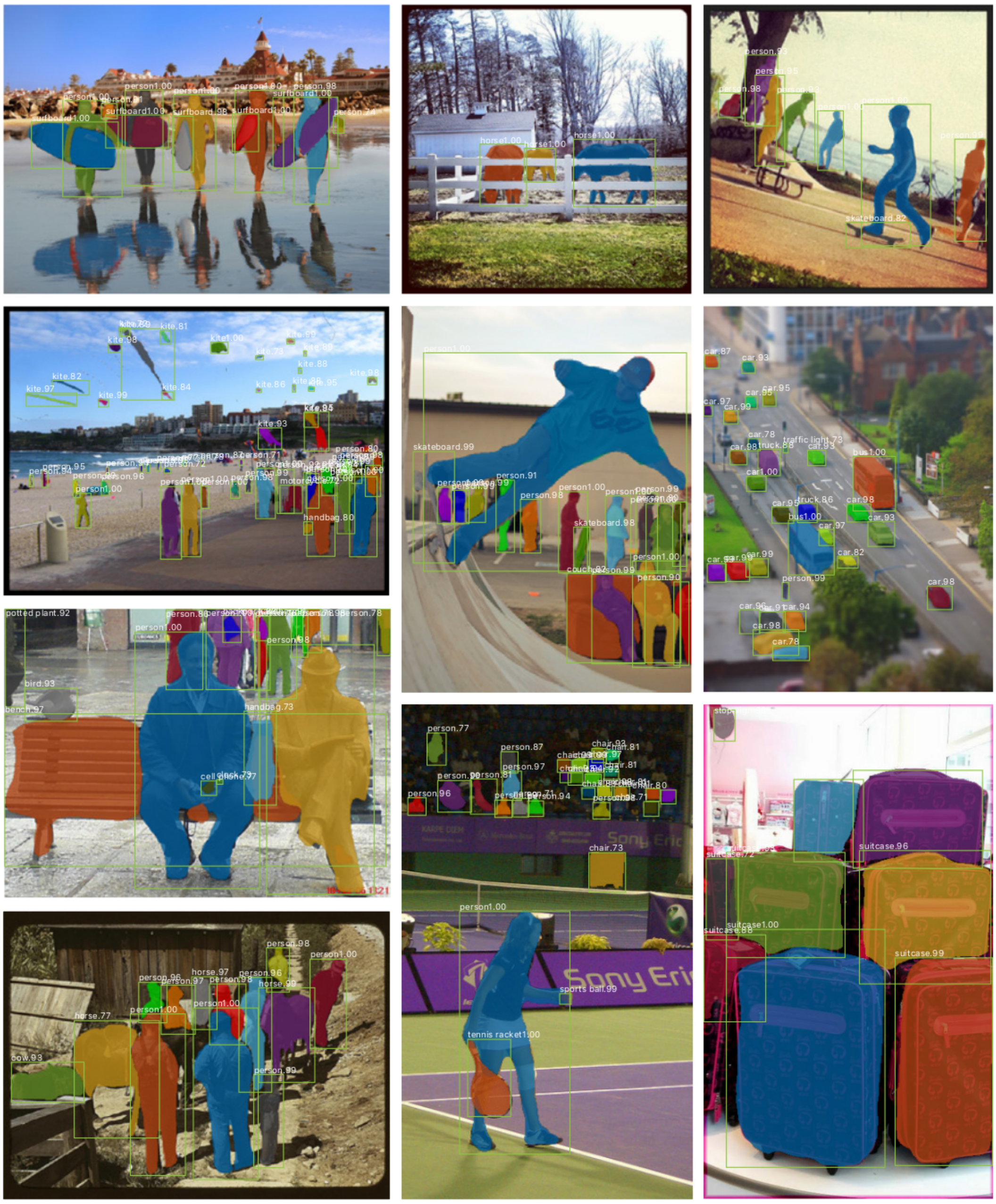


Figure 2.6: Segmentation performance of Mask R-CNN in complex scenes in which the detected objects are even partially occluded. Figure taken from the original work by He et al. [10]

3 Generating Space-Time Volumes of Interest: Methodology

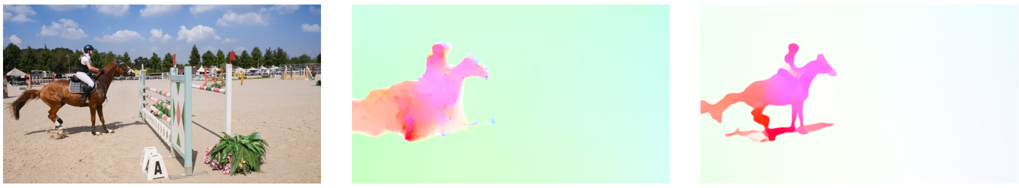
Our goal is to use high-level information about the position of objects, combined with their appearance and optical flow to build a representation of salient objects, the objects' components, and their motion in a bottom-up fashion. These spatio-temporally salient regions and structures extracted from the video model the dynamics of an object and its components over time. Our method uses object masks and the corresponding regions of interest provided by Mask R-CNN which was trained on the COCO databse [16], and optical flow. The video volume in combination with these two 'top-down anchors' helps us to constrain a bottom-up grouping, which yields temporally consistent regions of interest that correspond to object parts.

Before we start discussing our proposed method, we want to give a brief summary of the enabling technologies, and our reasons for choosing them.

3.1 Enabling Technologies

Object detection & masking Frame-based object proposals, e.g. the region proposal network RPN [26] will provide bounding boxes that inevitably contain background. We use Mask R-CNN [10], one of the best frame-based solutions currently available, which provides tight bounding boxes, COCO [16] object labels, and also a segmentation mask. However, these masks tend to be imprecise at the object boundaries and temporal correspondence is hard to obtain, so that a simple temporal stacking of masks obtained from individual frames of a video will not produce STVIs of the desired quality. Thus, we use Mask R-CNN masks as a per-frame initial segmentation which is subsequently refined.

Optical flow FlowNet2.0 [11] is a fast and reliable state-of-the-art solution to provide dense optical flow, and we use it as our second top-down anchor that can provide temporal coherence for moving objects. The reason for choosing FlowNet2.0 compared to other methods is that the optical flow generated by FlowNet2.0 tends to have sharp borders, see Fig. 3.1, which proves to be very useful in our proposed method. Optical flow alone will often fail to correctly segment objects of interest, for example with cluttered background, camera motion, and when several objects move jointly. It will fail completely whenever the object of interest temporarily does not move at all. In cases of articulated motion, e.g. a running person, there will be inconsistent motion of body parts w.r.t. the object centroid, and parts that touch the ground, e.g. the runner's leg, will temporarily vanish (zero motion while in touch with the stationary ground plane). Thus, we use optical flow as additional per-pixel and per-frame input to our bottom-up clustering.



(a) Reference (b) Warping based [20] (c) Flownet2.0 [11]

Figure 3.1: The flow computed by more classical approaches such as [20], which implements a method proposed by Brox et al.[5], compared to optical flow computed by Flownet2.0. As evident (c) has a tendency to produce very sharp borders at the objects' boundaries. This is useful for our approach where the optical flow should partially be used to segment moving objects from the background. Flownet2.0 is therefore chosen as our optical flow estimation method. Middlebury coding [3] is used for the visualization of the optical flow.

Clustering In terms of bottom-up methods, we need to deal with varying spatial and temporal scales that are related to size of objects and parts as well as to velocities of motion. This is achieved by smoothing with Gaussians of varying σ in space and τ in time, and has been researched in depth, see e.g. [19, 13]. For our STVIs, we use Gaussian scale space to achieve various levels of detail, as well as consistent tracking of parts through short partial occlusions. To abstract individual pixels into larger groups, various methods to generate superpixels have been proposed. In general, these methods were globally applied to images and videos with the typical result of a rather homogeneous distribution of compact superpixels of a certain size, where size is related to spatial scale. However, this is not what we aim at. We require superpixels of varying size and compactness, depending on the objects and their parts, but without modeling the background by superpixels at all. Out of many possible realizations of superpixelization, including e.g. TSPs [6], we prefer to use Simple Linear Iterative Clustering (SLIC) [2] due to a number of advantages, including simplicity, openness, and more importantly because the resulting superpixels suit our needs best as they yield superpixels with a well defined compactness over the whole image.

3.2 Method

We implement and evaluate two approaches that use the same common 'backbone' of extracting salient structures from image and optical flow sub-volumes centered around the salient object. The difference between the two approaches is the pre-processing of the input video.

The first approach that we will refer to as 'dense object masks' (in context shorthand **'dense'**) uses Mask R-CNN segmentation masks at every frame. This means that every frame of the video is passed through Mask R-CNN to obtain segmentation masks. From these masks the region of interest around the salient object is extracted, both in the appearance domain, i.e. intensity values, and in the optical flow. This approach *needs* these masks at every frame; should Mask R-CNN fail to detect the salient object even in one frame of the video, this approach fails. The schematic of the pipeline is shown in Fig. 3.2, where we can see a split into 3 distinct parts; the 'Pre-Processing', 'Backbone', and 'Post-Processing'.

The second approach 'sparse object masks with tracking' (in context shorthand **'sparse'**) is devised to alleviate these issues where a single failed detection leads to a total failure of our approach. Initially, a region around the salient object is selected, again with Mask R-CNN masks; but then a tracker is initialized with said salient region, which tracks the object over some amount of frames - in our case we chose a tracking duration of 15 frames. Additional masks are periodically used again to check if the tracking is still working correctly, and to adjust for small errors that might be introduced by partial occlusions. The difference in the pipeline is only in the 'Pre-processing' block. The details of the differences are discussed later in Section 3.2.2.

The benefit of the second 'sparse' approach is two-fold:

1. No failures when a mask (top-down anchor) fails in a single frame, meaning that this method should be more robust.
2. Considerable speedup as the calculation of the object masks is the most expensive operation in the 'dense' approach.

In the following we explain the implementation details of both proposed methods 'dense' and 'sparse', which refers to the frequency of their use of object masks.

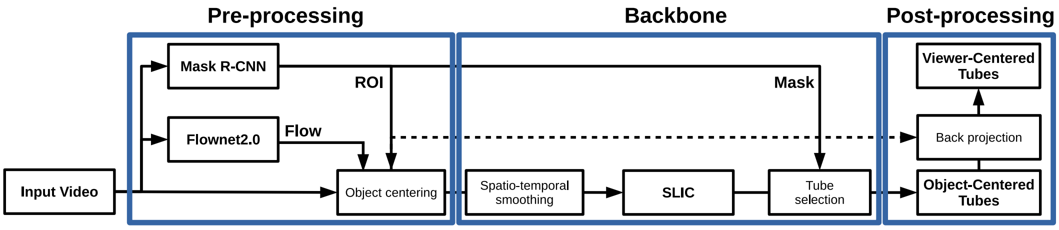


Figure 3.2: The Pipeline of our 'dense object masks' approach. It can be broken down in 3 major parts: pre-processing of the video, the extraction of salient space-time volumes by the 'Backbone', and the post-processing step which provides the visualization. The steps of the pipeline are such that Mask-R-CNN is used to extract the region of interest around the salient object, with Flownet2.0 providing optical flow information. The sub-volume of interest containing color- and flow-information is spatio-temporally smoothed to facilitate extraction of salient structures. SLIC clustering on the sub-volume is performed. From the clustering result, relevant tubes are selected that indeed correspond to objects of interest.

3.2.1 Dense object masks

In this section we describe a method that relies on the detection of correct Mask R-CNN masks in every frame of the video. The pipeline of this proposed approach is shown in Fig. 3.2. The pipeline shows that the input video is used to extract masks and region proposals for all object in each frame of the video while Flownet2.0 is simultaneously used to estimate the optical flow between all adjacent frame pairs of the video sequence. With this information available, our proposed method extracts salient spatio-temporal structures. These structures can then be displayed in a local, object-centered coordinate system, or in a viewer-centered coordinate system, which enables us to reason about the motion of the object and its individual components. The next paragraphs explain the pipeline, as seen in Fig. 3.2.

Object centering with regions of interest Starting from the input video, our first step is to use Mask R-CNN and obtain a region of interest (ROI) for the salient object in each frame. We use this ROI and create a sub-volume aligned along each of the ROI's center points, and padded to fit to the dimensions of the largest ROI in the video, see Fig. 3.3. With this processing step we create an object-centered sub-volume around the salient object in the scene.

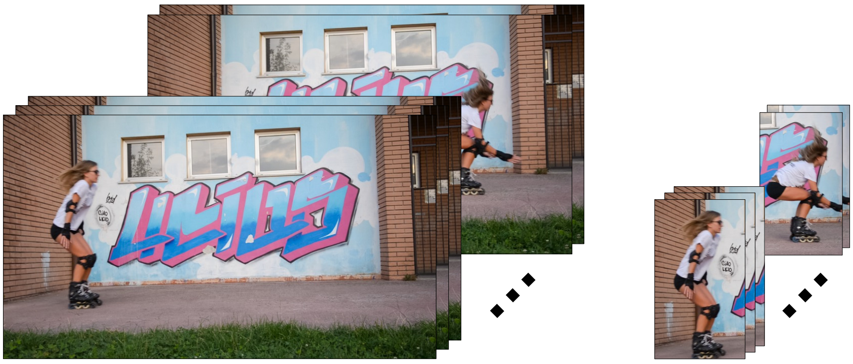


Figure 3.3: From the input video, we extract an object-centered sub-volume of interest (right) using Mask R-CNN region proposals.

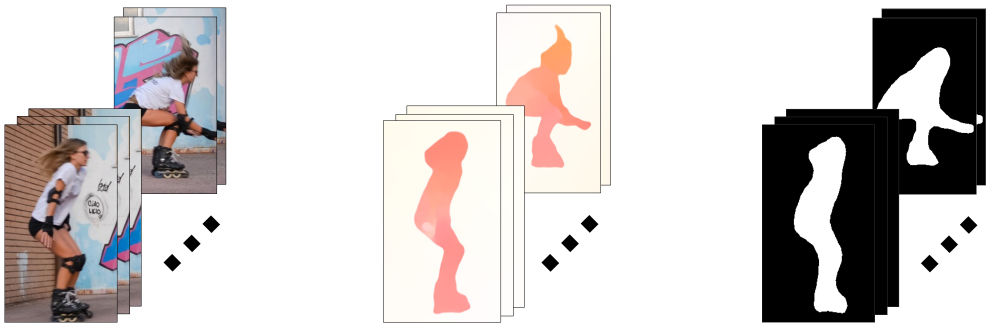
Optical flow of salient objects To provide temporal consistency of object and object-parts, we incorporate optical flow, generated by FlowNet2.0, into our pipeline. Optical flow deals with *apparent* motion in video, so that in the general case one cannot infer the motion of objects in the scene, but only the relative motion between scene, camera and homogeneously moving regions. There are scenarios - such as the camera tracking a moving object - where the object's optical flow might be close to $\vec{0}$, but the background optical flow reveals the tracking motion of the camera. The other extreme is a completely stationary camera where only the object is moving, producing $\vec{0}$ on each of the background pixels and non-zero flow on moving parts. Because we are only interested in representing salient objects, it seems natural to cancel the optical flow around the object of interest. We achieve this by calculating the average background optical flow in each frame, and subtracting it. This yields the relative motion of the salient object w.r.t. the background of each slice in the sub-volume:

$$\hat{F}_i = \left[F_i - mean(F_i \odot \neg M_i)\right]_{obj} \quad , \tag{3.1}$$

where F_i is the optical flow at frame i, and M_i the Mask R-CNN mask at frame i with 1's where an object is detected and 0's otherwise. The symbol $\odot$ denotes the element-wise multiplication, and $\left[\,\cdot\,\right]_{obj}$ denotes the cropping of the volume to the sub-volume around the salient object, cf. Fig. 3.3. $\hat{F}$ is therefore a sub-volume of the salient object-flow where, for each slice of the sub-volume, the mean background optical flow has been subtracted. For a visual intuition Fig. 3.4 shows the sub-volumes of image intensities, optical flow, and object masks, centered around the object of salience.

While the subtraction of the optical flow might not necessarily be needed for this approach to work, it does provide 1) a visual indication if the generated optical flow is good, and 2) close to $\vec{0}$ at any of the background pixels, which in the following steps will be useful so that background is discouraged to be recognized as a region of saliency.

Combining object masks and flow To summarize, the result of object-centering and object-flow calculations as explained above are two sub-volumes of identical extent in space-time:



(a) Image (b) Optical Flow (c) Object Masks

Figure 3.4: Sub-volumes of the image intensities (a), the corresponding optical flow centered around the object of salience(b), and the object masks (c) which were generated by Mask R-CNN.

1. The object-centered sub-volume, Fig. 3.4a, is cropped and centered around the salient object. We represent this sub-volume as a 6-dimensional entity $(x, y, t, L, a, b)^T$ with L, a, b referring to Lab color space.
2. The optical flow sub-volume, Fig. 3.5c, is represented as a 5-dimensional entity $(x, y, t, \hat{F}_x, \hat{F}_y)^T$, with $\hat{F}_x, \hat{F}_y$ denoting horizontal and vertical optical flow.

These two sub-volumes are now combined, resulting in one object-centered sub-volume that is represented in its 8 dimensions: $(x, y, t, L, a, b, \hat{F}_x, \hat{F}_y)^T$.

It is important to understand the role of these two top-down anchors: Object masks and optical flow allow us to extract a local sub-volume in a top-down fashion. There might be several objects of interest in one video snippet, exhibiting different object motion. For each object, we obtain an 8-dimensional sub-volume at the individual *object level*, i.e., we move from global space-time representation in a video to local, object-centered representations.

Spatio-temporal smoothing Each 8-dimensional sub-volume is now processed locally at the object level, in a bottom-up fashion. To facilitate the creation of connected intra-frame structures of an object, we smooth spatially; to create structures that are temporally consistent, even when parts of the object may be partially occluded, we smooth temporally. The combination of these smoothing operations provides a means to weigh and emphasize certain object- and motion-patterns in the sub-volume, if desired. We use Gaussian scale space, with its extension from spatial to temporal domain and the notation of a space-time scale space family $\mathscr{L}$ as proposed by Laptev and Lindeberg [12]:

$$\mathscr{L}(\cdot;\sigma^2,\tau^2) = g(\cdot;\sigma^2,\tau^2) * f(\cdot) \quad , \tag{3.2}$$

where the input f is convolved with a Gaussian kernel g; σ determines the spatial smoothing within each frame, and τ the smoothing in time. If the convolution is performed with large σ, τ, only large scale structures remain; convolution with small σ, τ preserves the higher-frequency structures in the sub-volume. While it is well-known that particular scales emphasize particular space-time structures, we explore the power of scale space *locally* to support the decomposition of the object at hand into meaningful components.

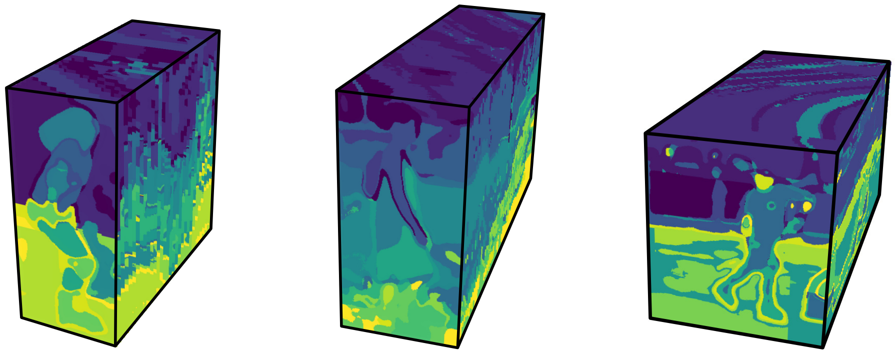
SLIC applied to appearance & flow After smoothing, we cluster the object into meaningful components. This is the second important bottom-up processing step in our pipeline Fig. 3.2. We achieve this by extending SLIC with its original 5 dimensions of two image coordinates and three color channels to the full 8 dimensions of our sub-volume by adding time and optical flow to construct an 8-dimensional descriptor:

$$\phi(x,y,t) = \begin{bmatrix} \alpha(x,y,t)^T \\ (L,a,b)^T \\ \beta(\hat{F}_x,\hat{F}_y)^T \end{bmatrix} \text{dimensionality: } 8 \times 1 \quad . \tag{3.3}$$

SLIC clustering is then performed in the smoothed sub-volume w.r.t. its spatio-temporal dimensions (x,y,t), its color (L,a,b), *and* object flow $(\hat{F}_x,\hat{F}_y)$, an example of resulting structures is shown in Fig. 3.5.

The scalar values α, β allow us to stretch and compress the sub-volume in which SLIC superpixels are generated, where high values of α create more compact regions by prioritizing spatial proximity. The scalar β is a trade-off between image intensities and flow components, with higher values prioritizing the flow component.

It is also noteworthy that we do not enforce spatial connectivity in our SLIC clustering; our reasoning is that since we work in the 2D projection of 3D data, objects that are connected in 3D might not appear connected in 2D.

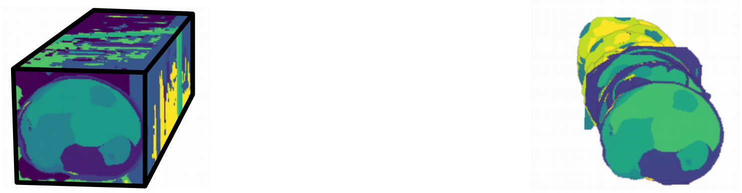


(a) Rollerblading (b) Walking (c) Skating

Figure 3.5: Volumetric structures generated from a spatio-temporally smoothed appearance and optical flow (as described in Eq. 3.3), followed by SLIC clustering. We can see that the clustering process is able to group together parts, that are similar based on their appearance and motion, e.g. the head, torso and limbs in (a).

Lastly, we need to decide on the parameter k which determines how many cluster-centers in the k-means clustering step of SLIC are placed. This parameter dictates to a very large degree the coarseness of the structures from the video sequence. Large values of k generate many small details in the sub-volume, whereas small values of k guide the focus toward large scale structures. This parameter, needs to be balanced with the spatial and temporal smoothing parameters, σ, τ respectively. Once that is done we can use our space-time variant of SLIC on a sub-volume of interest. However, this bottom-up clustering step alone will not readily represent the actual salient object, but the whole sub-volume will be filled with (hopefully) tube-like structures. These tube-like structures can be thought of as representing a certain region of the sub-volume over time; we need to find tubes that represent just the object of interest, which we solve in a top-down selection process.

Tube selection To select the relevant tubes which model the object and its motion we use the mask proposals from Mask R-CNN in each frame. We propose a simple yet effective way that can also deal with outliers; we stack all the masks in time, and select those tubes in the sub-volume that overlap with the mask-volume more than a certain percentage. The amount of overlap needed for a tube to be considered 'good' is selected via a threshold. Examples for successful tube selection are shown in Fig. 3.6.



(a) Structures

(b) Space-Time Volume of Interest

Figure 3.6: The clustering result of appearance and flow still contains lots of background. Additional masks from Mask R-CNN are used to select the structures that model the object more closely.

In our experiments, we verified that the results obtained are rather insensitive to this threshold parameter, when chosen between 50% - 90% overlap. We think that the robustness of the approach w.r.t. the overlap can be attributed to:

1. We work with short videos, where single tubes are able to capture and track parts of objects through the whole sequence, and
2. Even though Mask-RCNN masks are not pixel perfect they do detect the object with high accuracy and place well fitting masks over them.

In all our experiments reported in this work, the threshold has been set to 70% overlap.

As videos increase in length, the threshold parameter also needs to be tuned down, to accommodate tubes that approximate the object well through one section of the video but do not persist over the entirety of the video. One way to avoid the discarding of tubes that do not persist throughout the video but are generally good fits, is to consider only a reduced number of consecutive frames in the sub-volume, and to perform the clustering on these frames. The resulting tubes from this process will either present meaningful short-term volumes, or would need to be merged with other tubes based on some similarity measure. For our current purposes, where videos do not exceed a duration of a few seconds, we did not deem this partial tube merging necessary.

Object- and viewer-centered Space-Time Volumes We extract our STVIs from object-centered sub-volumes. So far we have argued in favor of this object-centered perspective. However, it may be interesting to switch back to the viewer-centered perspective to analyze what happens in general in a particular video. Once we have extracted objects and represent them as STVIs, we can back-project them into the space-time volume of the whole video to analyze the global motion of objects, and interactions between several objects. Fig. 3.7 shows an example for the differences between object- and viewer-centered STVIs: a roller-blade skater is skating through the scene from left to right, and jumping along the way. In the viewer-centered frame of reference, the jumping motion is clearly visible, whereas in the object-centered frame of reference, we see motion of body parts relative to the object centroid in local object coordinates, cf. also teaser Figs. 1.1 and 1.2.

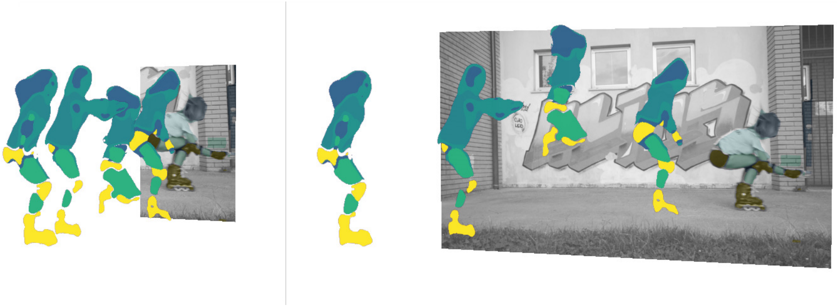


Figure 3.7: Changing from object-centered (left) to viewer-centered (right) coordinates can show the trajectory of the salient object, in this case the jump of the inline skater. The last frame of the video sequence is shown to provide a reference, and a visual cue on the segmentation accuracy provided by our generated space-time volume of interest.

Automatic parameter estimation Up to this point, we have explained all components of the general pipeline to extract Space-Time Volumes of Interest - STVIs - from video, but we have spared the actual choice of several important parameters: σ, τ for the spatio-temporal smoothing of the volume of interest; α, β for weighting the volumes proximity, color and flow components; and k, the number of cluster-centers during k-means clustering. Some of these parameters need to be set in accordance with each other. For instance as smoothing with large σ, τ kernels reveals low-frequency structures in the video volume, the number of clusters k that are needed to approximate the data in a meaningful way is lower. Unless explicitly noted for particular experiments, all parameters for the experimental results in this work were set according to the procedures described below.

σ: spatial smoothing is symmetric, meaning that x- and y-direction have the same value, and is estimated by the radial approximation, i.e., half of the diagonal of the average bounding box of the salient object:

$$\sigma = \frac{\sqrt{h^2 + w^2}}{2 \cdot b_1} \quad , \tag{3.4}$$

where h is the average height, and w is the average width of the bounding boxes of the salient object. The scaling factor is set to $b_1 = 25$. This metric emphasizes the size of an object and is rather robust to changes of the object's pose inside the bounding box.

τ: temporal smoothing is estimated by considering the maximum object flow in the sub-volume:

$$\tau = \frac{max(||\hat{F}||)}{b_2} \quad , \tag{3.5}$$

where $||\hat{F}||$ is the magnitude of the optical flow $\hat{F}$ in the sub-volume, according to Eq. 3.1, and the scaling factor is set to $b_2 = 100$.

α, β: clustering weights as introduced in Eq. 3.3 are set such that the spatial proximity component α depends on the selected spatial scale σ, and the optical flow weighting β depends on the selected temporal scale τ:

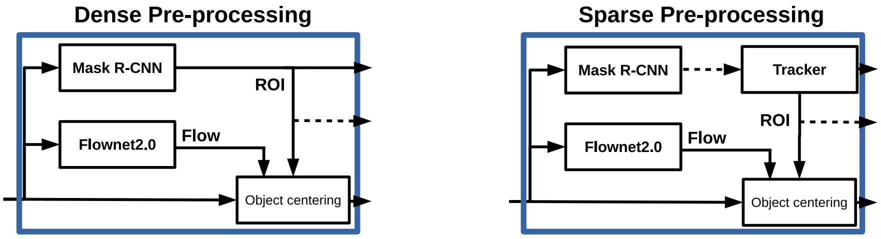
$$\alpha = \frac{\sigma}{b_3}, \ \beta = \frac{\tau}{b_3} \quad , \tag{3.6}$$

where the scaling factor is set to $b_3 = 10$.

k: number of cluster centers is set to $k = 15$, as any COCO class can be approximated reasonably well by 10-15 space-time tubes. Future work could entail setting k based on the class of the salient object, taking into account the complexity, e.g. the number of meaningful parts for a particular category.

3.2.2 Sparse object masks with tracking

In the previous subsection we described how our proposed approach works when Mask R-CNN is able to correctly identify each salient object in the video in



(a) Dense Method (b) Sparse Method

Figure 3.8: The only difference between the dense and the sparse method is the 'Pre-Processing' step in which coherent masks and regions of interest around each salient object in the video are extracted; the 'Backbone' and the 'Post-Processing' stay unchanged, cf. Fig.3.2. The modified pipeline utilizes a tracker, which is supplemented periodically with Mask R-CNN re-detections. This pipeline ensures that we are able to extract STVIs even with sporadic Mask R-CNN object detection failures.

every frame. Even if the object of interest is detected correctly in every frame, we need to be able to correctly find the correspondences between the individual objects in adjacent frames. Because Mask R-CNN does indeed have spurious failures to detect objects, especially through partial occlusions, we also propose a 'sparse' method that utilizes a tracker, so that we can extract STVIs from videos more robustly. The pipeline for the 'sparse' approach is very similar to the 'dense' approach (Fig. 3.2), and only differs in the 'Pre-processing' block. The differences are shown in Fig. 3.8.

We choose a variant of a discriminative correlation filter, the 'Discriminative Correlation Filter with Channel and Spatial Reliability' [21] (CSRT) tracker implemented in the OpenCV [4] library. Because of its reliability for short tracking sequences it is a good fit, as masks will be provided in periodic intervals to update and pinpoint the location of objects more accurately.

Tracking The tracker is initialized with the detections of Mask R-CNN from the first frame in the video. The region of interest is then tracked. Re-detection on the whole frame is performed every 15 frames, which is used for slight adjustments of the region of interest. For this it is necessary to perform the matching of bounding boxes between the last frame of the tracked sequence, and the re-detection. Usually this will be a matter of matching n tracked bounding boxes, to m bounding boxes obtained through the re-detection, where in general $n \neq m$. To select the best matches we need to determine the similarity between the two sets of bounding boxes. This is done by computing the intersecting area between the two bounding boxes. The match with the highest score is likely to be correct, if Mask R-CNN worked correctly in the re-detection. If it did not, it is problematic to update the location of the bounding box. To decide if the correct bounding box has been chosen, we additionally compute a confidence score; the intersection over the union of the two bounding boxes. If that confidence score is greater than 50%, we replace the currently tracked bounding box with the newly detected one. In case the confidence is smaller, we discard the re-detection information and continue to track the last region of interest known to be correct, and request a new mask every 2 frames for that object until we are able to ensure that we are still tracking the correct object. If this is not possible to determine while we are tracking the object within the next 10 frames, we abort the extraction of STVIs. This proposed tracking scheme

1. reliably tracks objects, even with spurious Mask R-CNN failures, and
2. compensates for drift of the tracked region over time, sometimes the result of partial occlusions.

The ‘sparse’ method makes our approach more robust, enabling us to extract STVIs from more videos.

Tube selection Tube selection, as discussed previously, is the procedure of extracting tubes of saliency which model the object of interest from the sub-volume that is created by clustering appearance and optical flow information. We use a similar technique as for the dense object masks to determine which tubes are a good fit, the intersection with the object masks from Mask R-CNN. Where in the dense approach we noted that extracted tubes are not sensitive w.r.t. the threshold parameter, this is quite different here; small changes in the threshold now drastically affect which tubes are selected. This makes sense, as now fewer masks are responsible for selecting salient tubes. This means that if the masks are imprecise the selected tubes will probably be imprecise as well. In our experimentation we set the tube selection threshold to 70%, the same as in the approach that uses dense masks.

3.3 Method summary

Our method extracts salient structures obtained from objects' appearance and their motion from videos using an interaction of top-down selection and bottom-up refinement processes. It relies on the detection of object bounding boxes for initialization of a tracker which is used to extract a sub-volume of interest. Spatio-temporal smoothing in Gaussian scale space of appearance and optical flow information in the sub-volume is used before SLIC clustering is applied. The Mask R-CNN masks are then used to select salient structures from that sub-volume containing appearance and optical flow, i.e. selecting structures that lie withing the boundaries of the masks. If multiple salient objects are present in the scene, the proposed algorithm can be run once for every object instance. After that it is possible to project the extracted STVIs back into the video.

Fig. 3.9 gives a simplified overview that shows partial results as they are processed in the pipeline.

Furthermore, we present two ways of obtaining coherent regions of interest from the input video. The first method, 'dense object masks', uses masks at every frame of the input video, whereas the second approach,'sparse object masks', only relies on them during initialization; a mask is used to initialize the tracker with a region of interest. This region of interest is tracked, and additional masks are periodically used to check for coherence. Even if masks are not correct during one of the periodic checks, our method is able to continue tracking and recover in most cases. This robustness to sporadic failures is what enables the 'sparse' method to work more reliably. Another benefit of using this method is that it is less hindered and influenced by partial occlusion of objects. This is because if an object is partially occluded it is less likely to be correctly masked by Mask R-CNN. If we use fewer of the masks, and preferably those where no occlusions are present, we can extract better STVIs, as the mask contains fewer 'background' areas.

We also found that the method used for tracking is not of critical importance, as the duration of the tracking is quite limited before another 'anchor' is supplied. More specifically, we did not see any noticeable difference between most common tracking algorithms provided by OpenCV [4].

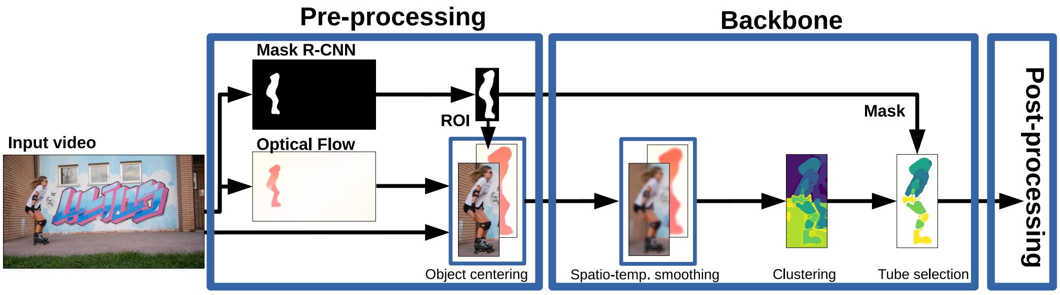


Figure 3.9: Pictographical overview of our proposed method, showing partial results of each processing step. The naming of the individual steps is such that it matches the pipeline described in Fig. 3.2.

It is also noteworthy to compare the computational cost for both approaches; see Table 3.1. Most of the computational cost comes from Mask R-CNN generating object masks. In a video of 80 frames, the 'dense' approach needs approximately 11 minutes to extract STVIs, where the 'sparse' approach only requires 5 minutes. The tracking interval is set to be 15 frames, meaning that new masks are supplied every 15 frames.

We can see that if the calculation of the object masks is performed less frequently that the computational cost is halved. Note that in the 'sparse' method the calculation of the optical flow is the bottleneck, using approximately 90% of the computational time. If one would need to increase the performance w.r.t. computational time, replacing Flownet2.0 with a faster methods would be the logical step. Note, that the resulting STVIs are likely to be worse, based on the fact that Flownet2.0 yields very sharp edges and contours around objects, which to our knowledge no other optical flow method can match. These sharp edges are one of the reasons why the object edges of our STVIs look 'natural'. We do not know of any other optical flow method whose run time is faster that can produce equally sharp contours.

Processing Step	**Time (in $\lfloor sec \rfloor$)**	
	Dense	**Sparse**
Calculating object masks	413	25
Calculating optical flow	302	302
Object centering	0	0
Object tracking	-	3
Spatio-temporal smoothing	2	2
SLIC	3	3
Tube Selection	0	0
Tube back projection	0	0
Total	720sec ≈ 11min	335sec ≈ 5 min

Table 3.1: Comparison of the computational cost of our two proposed methods of extracting Space-Time Volumes of Interest. The sparse method is shown to be around 55% faster, mainly because object masks do not need to be calculated every frame, and because the compensating mechanism (object tracking) is very fast. The entries that contain a '0', are in the order of a couple of milliseconds and negligible compare to the other steps. The entry '-' in row 'Object tracking' and the 'Dense' column means that object tracking is not performed. These results are obtained from the DAVIS [24] dataset, from a video that contains 80 frames. Mask R-CNN is run on a Intel i7-5939K processor, and Flownet2.0 on a setup of 4 NVIDIA 1080 Ti on a computer with 32gb RAM.

4 Evaluation

We begin with an overview of the used databases, Section 4.1, their video quality, video length, and the reason for choosing them. We structure the remainder of the evaluation of our method such that we:

1. summarize important aspects of our visualizations used throughout our work 4.2,
2. provide quantitative segmentation performance results using the automatic parameter selection, of our 'sparse' and 'dense' STVI method, compare it with Mask R-CNN 4.3.1, and
3. compare the quality of the segmentation achieved with STVIs at various spatio-temporal scales 4.3.2.

As our extracted representation is composed of smaller sub-components we will also discuss how well these sub-components resemble 'meaningful parts' of an object. This will be briefly touched upon in this section, and elaborated more in detail in Chapter 5 - 'Discussion'.

4.1 Databases

A representation should be able to capture the objects that it is trying to represent in their entirety; for this reason we choose to evaluate the segmentation performance of our method w.r.t. a ground truth segmentation. We therefore need a video database that contains ground truth labels for each frame of the video. Keep in mind that the object detector that we use was trained on the COCO [16] database; we therefore need to choose a database that contains classes of objects which our top-down anchor can reliably detect. The 'Densely Annotated VIdeo Segmentation' (DAVIS) [24] dataset meets all these requirements. It contains 50 videos with a length between 25 and 104 frames

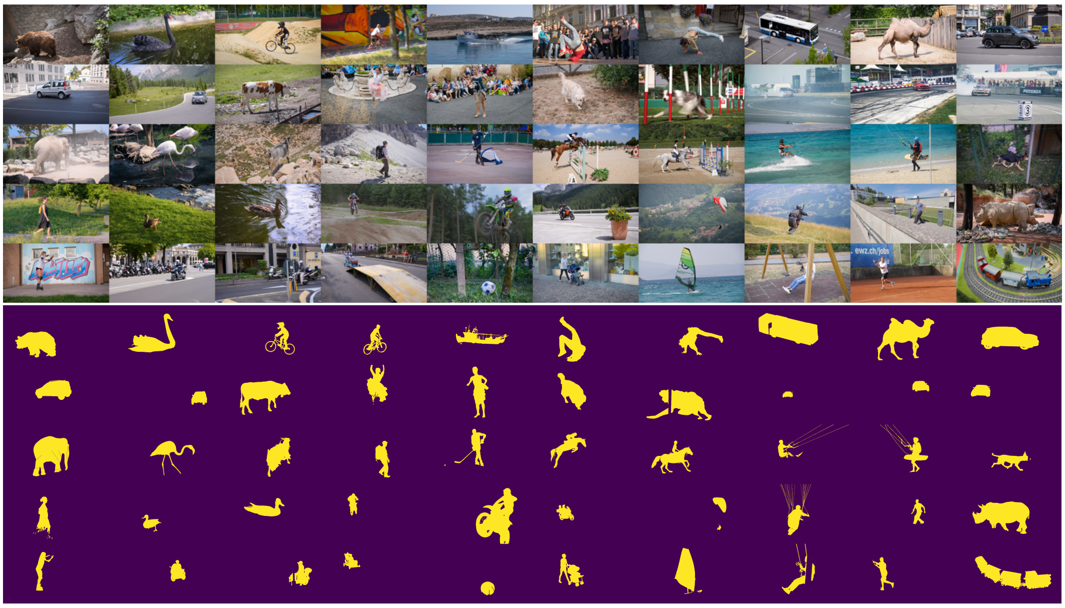


Figure 4.1: Sample still frames from DAVIS dataset, with their corresponding ground truth segmentation. The database has a variety of different subjects, such as humans, animals, cars and toys. Some of the sequences from the database contain many instances of other objects in the background, which are not contained in the ground truth labels.

with an average video length of 69 frames, with each video having a resolution of 854×480 pixels. Single frame samples from all videos of DAVIS are shown in Fig. 4.1. The corresponding ground truth segmentation is shown in the lower part. The ground truth labels are pixel perfect, as they are created partially manually. The database contains a variety of subjects that exhibit articulated motion, and rigid motion. Moreover, it contains different classes of subjects, with varying appearance; Table 4.1 shows the number of videos per class. This grouping was done by us. For example, we group scooters, motorbikes, and bicycles into the category ‘Bike-like’. Similarly, we group buses, cars, and boats into ‘Car-like’ objects.

Another database that we used is UCF101 [30]. It contains 101 classes of actions performed, some of which contain uncluttered scenes. Except for the UCF101 ‘jumping jack’ video used for STVI visualizations in Figs. 1.1, 1.2, 4.2

Group	Abbr.	Number of videos
Human	(H)	15
Animal	(A)	13
Car-like	(C)	8
Bike-like	(B)	7
Misc.	(M)	5
Animal + Human	(AH)	2
Total		50

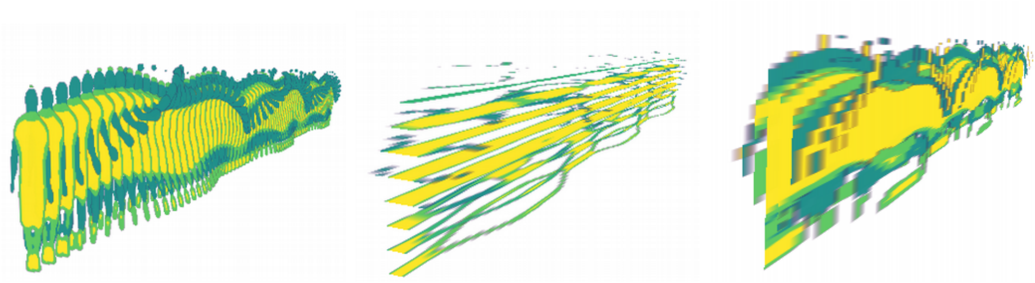
Table 4.1: Broad grouping of videos from DAVIS databse. Videos where there is only a single human or animal subject are the most represented group, comprising 60% of the database.

and 5.6, we use DAVIS videos for all our other examples and for quantitative evaluation, as DAVIS provides ground truth masks for all objects in each frame, and UCF101 does not. The reason for including the 'jumping jack' video is because it captures the full motion of the subject without any occlusions, with the background being a uniform color without texture. This yields very clean STVIs on which we can judge the performance of our method on a 'best case' scenario.

The DAVIS videos contain many humans and animals with a lot of variety in the actions they perform, their appearance, the backgrounds, the tracking motion of the camera, and the amount of partial occlusions occurring throughout the sequence. Because our approach aims to be as general as possible this variety should be able to reveal the benefits, draw-backs, and limitations of our proposed method.

4.2 Visualization

STVIs as visualized in our work show individual tubes that, when taken together, resemble the space-time extent of an object. 'False' color coding is such that one color corresponds to one tube, where we have already discussed that we do not enforce spatial connectivity in our SLIC clustering. Thus, individual tubes constitute the result of spatial and temporal smoothing of appearance and optical flow information to a particular scale, and subsequent bottom-up clustering. Our visualization inherently accounts for this aspect of an object's decomposition into spatio-temporally meaningful parts. This is in contrast to other approaches (e.g. [9, 35]) as discussed in Chapter 2, Figs. 2.5, and 2.4 where object- and motion-tracks are constructed as tight 3D meshes, effectively only showing the hull of the space-time object.



(a) Sliced along t-axis (b) Sliced along y-axis (c) Sliced along x-axis

Figure 4.2: Our Space-Time Volumes of Interest can be sliced along any of the dimensions, to emphasize structures in the volume, not easily visualized in previous approaches. Notice how the oscillating motion of the legs is emphasized differently depending on the slicing: in (b) the legs form tubes that split and then merge, whereas in (c) the merging of the legs seen as a yellow blob in the center plane is most prominent.

Note that [35] does not work fully automated but that particular aspects of the motion may be manually selected and visualized. Such volumetric hull-like visualizations do not suffice for our purposes, because one major feature of STVIs is their decomposition into object parts, which might be nested. Our visualizations highlight several important aspects of STVIs:

1. We visualize multiple sub-components of objects, where one component might enclose the other, or where a component might temporarily split or merge.
2. We provide slicing of the volume along any of its spatial or temporal dimensions, emphasizing motion patterns by looking at different cross-sections, see Fig. 4.2.
3. While we have presented automatic selection criteria for all our relevant parameters, we can also visualize what happens when clustering at various spatial and temporal scales, see Figs. 1.2 and 4.8.

Showing each slice in all (x, y, t)-dimensions would resemble a dense volumetric representation, but would limit possibilities to analyze the decomposition of

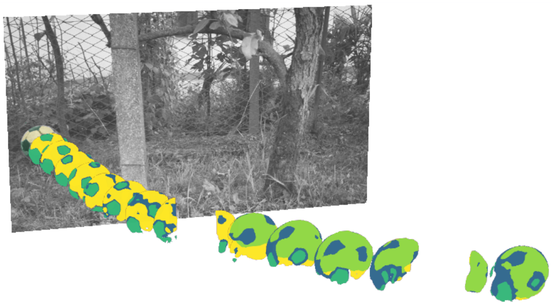


Figure 4.3: Ball rolling through the scene, slowing down towards the end. Note that (1) the ball is partially occluded twice, but object continuity is preserved, and (2) in the beginning the optical flow component is dominant and later on when the ball slows down the appearance weighs higher, hence the pentagons on its surface appear more pronounced.

an STVI into tubes. Thus, we have implemented a rendering of sparse slices of the volume. All figures in this work use sparse slicing along one of the (x, y, t)-dimensions of an STVI for a better visual understanding.

With the possibility to back-project the object-centered STVIs into the viewer-centered coordinate system, analysis of the motion of the object relative to the viewer is possible, which enables us to view the scene dynamics at the object level, see e.g. Fig. 3.7. This viewer-centered perspective is not only useful when representing interactions between several objects, but can also be useful to analyze the motion of a particular object, e.g. the ball in Fig. 4.3. Because the camera is almost stationary, the viewer-centered coordinate system stays aligned with the scene coordinate system.

This option to view STVIs w.r.t. different frames of reference can be quite useful: With a stationary camera the viewer-centered coordinate system makes the trajectory of the object immediately apparent, see Fig. 4.3 Combining this with a very coarse scale enables one to view the interactions of objects in a simplified manner over time. In contrast to this there is the object-centered perspective that lines up the slices of the STVIs, w.r.t. the centroid of each object in each slice. This makes the movement of individual parts w.r.t. each other more apparent.

4.3 Segmentation performance

To obtain quantitative results w.r.t DAVIS ground truth masks, we consider two metrics that measure segmentation performance as defined in [34]:

- Segmentation accuracy **(ACC)** that measures the overlap between our segmentation and the ground truth, and
- Under-segmentation error **(UE)** that captures the amount that overflows past the object's actual border.

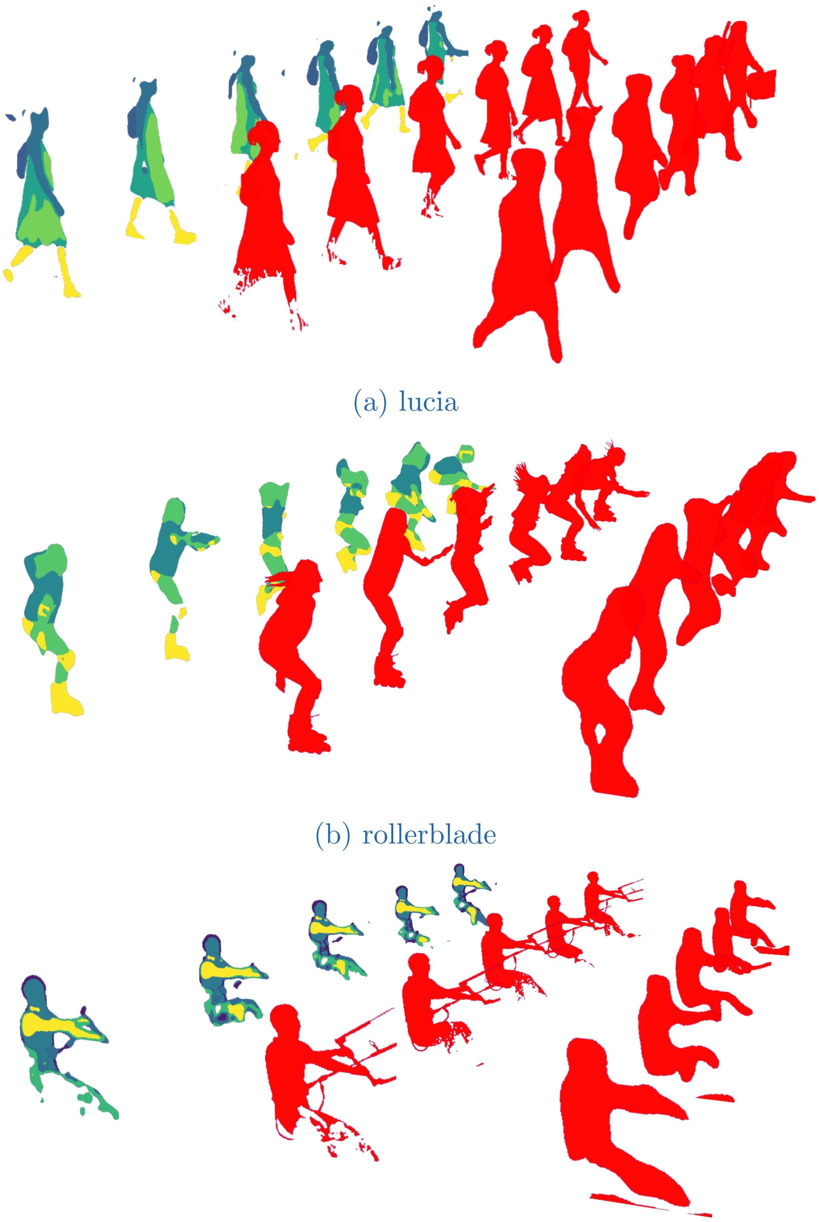
Both of these measures are normalized w.r.t. the ground truth masking, meaning that the best performance would be achieved with an accuracy of 100% meaning that all pixels that belong the the object are recognized correctly, and an under-segmentation error of 0% meanining that no pixels that do not belong to the object were recognized as the object.

In this section we measure these two metrics on the DAVIS dataset; first we investigate the segmentation performance of STVIs when using the proposed method of automatic parameter selection described in Section 3.2.1, and second we investigate the segmentation performance both qualitatively and quantitatively with a manual selection of the parameters.

4.3.1 Automatic parameter selection

Qualitative Results In the course of our experiments on DAVIS data, we noticed that Mask R-CNN tends to yield masks that are slightly too large, and that do not have natural looking object borders. Fig. 4.4 shows the qualitative comparison of our approach, DAVIS ground truth, and Mask R-CNN masks side by side on three videos. While Mask R-CNN masks tend to fully enclose the object, it produces sightly too large masks with imprecise object contours. Our approach produces natural looking object boundaries which are slightly too tight because of the non-edge-preserving smoothing that is applied. In general, we observe that Mask R-CNN masks err on the side of providing masks that are slightly too large, which means that they inevitably cover more of the object, increasing the probability for a higher segmentation accuracy score. The penalty for this choice is only revealed when we look at the under-segmentation error; because this error measures essentially how much background is mistakenly recognized as part of the object, it is on average higher than the UE that our STVIs yield. Additionally, note how in Fig. 4.4 Mask R-CNN is very apprehensive to creating 'holes' in the object mask, whereas our approach does not appear to exhibit this behavior. This is because, as mentioned in Section 3.2.1, we do not enforce full spatial connectivity during the SLIC clustering.

Because we smooth the sub-volume of interest, small foreground details that temporarily occlude the object are ignored. This can also be observed in Fig. 4.5: The per-frame ground truth correctly shows only parts of the legs and feet that are not occluded by the grass in the foreground, while STVIs capture the temporally consistent body part, whereas Mask R-CNN only yields unrefined blob-like mask regions. This effect is mostly dependent on the parametrization. We designed our automatic parameter estimation with the goal in mind, that it is able to represent objects at a 'medium' scale, meaning that it should be possible to represent the object by approximately 5-10 tubes; this number comes from the fact that we initialized k, the number of cluster-centers to be 15, and adjusted the spatio-temporal smoothing accordingly. From this it is apparent that it is expected that some parts of the background are still present in the region around the salient object, and are modeled by tubes which are later discarded in the 'tube selection' process. From this analysis we conclude that our automatic parameter selection seems to work as intended.

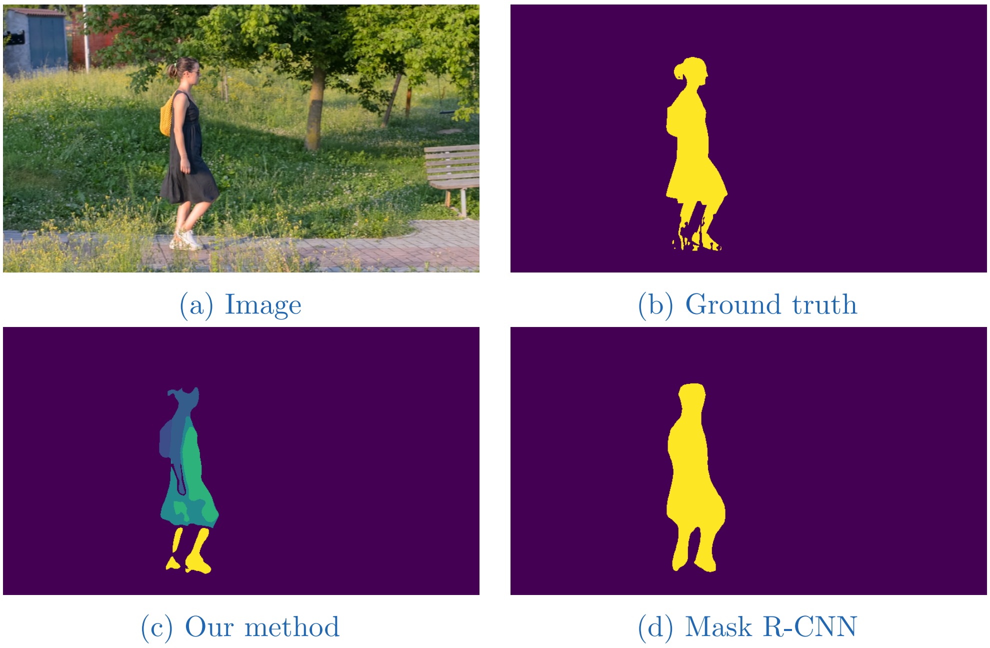


(a) lucia

(b) rollerblade

(c) kite-surf

Figure 4.4: Sample results of generated Space-Time Volumes of Interest for three videos from the DAVIS dataset. Each graphic shows the extracted STVIs (left), the ground truth segmentation (center), and the Mask R-CNN mask (right). The different 'parts' / 'tubes' that our method is able to extract are color coded, meaning that same colors comprise the same 'tube'. Note how our approach makes it more apparent what the represented object is, compared to Mask R-CNN masks, not only due to the decomposition into parts, but also because the contours tend to capture the shape of the object better.



(a) Image

(b) Ground truth

(c) Our method

(d) Mask R-CNN

Figure 4.5: Qualitative segmentation result of a single frame from the DAVIS dataset, 'lucia' video. The mask created by Mask R-CNN (d) is very rounded and looks overall very blob-like, whereas our approach (c) is able to capture more details with natural looking curves. Note especially how the feet are partially covered by grass even in the ground truth segmentation (b), which leads to even Mask R-CNN having troubles to recover the shape of the foot. Because of the spatio-temporal smoothing that our approach employs, it is able to deal with such small occlusions far better, and can recover the shape of the foot. Also note that our method does not capture the top part of the head of the walking person. This is because it is has a very similar color to the background, and is therefore (during clustering) modeled by a tube that consists mostly of background, and is later discarded in the tube selection step.

Quantitative Results We test on the full range of DAVIS videos, where we compare the two discussed approaches, 'dense object masks' and 'sparse object masks' w.r.t. accuracy and under-segmentation error. Table 4.3 shows the results, with an additional column for Mask R-CNN masks' segmentation performance. The table also lists the group (category) of the salient object, such as 'Human', 'Animal', 'Bike-like', etc., as introduced in Table 4.1. Note that if an approach was not able to correctly extract STVIs and failed, entries in the table are marked with a '-'. In the case of 'dense object masks' this might happen, because Mask R-CNN is not able to recognize and generate the object masks in one or more frames in the frame sequence. Because this approach relies on masks in every frame, even a single failure renders this approach useless. This already highlights the benefit of the 'sparse object masks' method; it does not rely on dense masks, and therefore works on more videos. More precisely: it enables us to evaluate 8 videos more, which is 16% of the total DAVIS database, see Table 4.2. This table also shows that the failures of extracting STVIs are not limited to one group, but may affect any group. Nevertheless, the 'sparse' method also fails on 3 videos of the DAVIS dataset. In our particular case these failures are caused by a cluttering of the scene by many objects of the same type and by partial occlusions of the most prominent object that is of actual interest. This proves a difficult combination of circumstances to reliably detect and track individual instances. In such cases where even the tracking fails and the resulting STVI is malformed, we abort the process as well, and mark the corresponding entry in Table 4.3 also with a '-'. The details about the types of videos which might cause our approach to struggle are discussed in Section 5.1.

Group	Abbr.	**Sparse**	**Dense**	**Total**
Human	(H)	15	11	15
Animal	(A)	12	11	13
Car-like	(C)	7	5	8
Bike-like	(B)	6	6	7
Misc.	(M)	5	4	5
Animal + Human	(AH)	2	2	2
Total		47	39	50

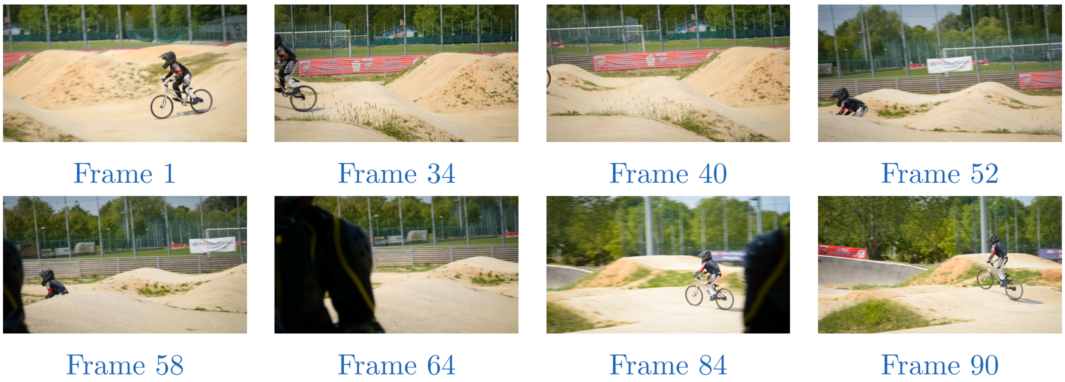
Table 4.2: Videos per group from which STVIs could be extracted. The 'sparse' method only fails in 3 videos, whereas the 'dense' method fails in 11, out of 50 videos available from the DAVIS dataset. Note that the failures in both methods are not limited to a single group, but rather spread evenly among the groups.

		Accuracy			Under-seg. error		
Dataset	**Group**[1]	**Sparse**	**Dense**	**MRCNN**	**Sparse**	**Dense**	**MRCNN**
bear	A	0.79	0.83	0.93	0.11	0.08	0.24
blackswan	A	0.78	0.79	0.90	0.10	0.03	0.03
bmx-bumps	B	-	-	-	-	-	-
bmx-trees	B	0.54	0.55	0.85	0.11	0.11	0.12
boat	C	0.75	0.72	0.90	0.18	0.48	0.35
breakdance	H	0.64	-	-	0.25	-	-
breakdance-flare	H	0.27	-	-	0.10	-	-
bus	C	0.70	0.74	0.95	0.05	0.14	0.13
camel	A	0.43	0.70	0.85	0.05	0.23	0.24
car-roundabout	C	0.87	0.37	0.96	0.08	0.11	0.41
car-shadow	C	0.76	0.76	0.96	0.03	0.02	0.16
car-turn	C	-	-	-	-	-	-
cows	A	0.56	0.56	0.92	0.06	0.05	0.08
dance-jump	H	0.17	-	-	0.01	-	-
dance-twirl	H	0.55	-	-	0.05	-	-
dog	A	0.82	0.50	0.94	0.07	0.01	0.06
dog-agility	A	0.38	0.32	0.77	0.04	0.03	0.28
drift-chicane	C	0.70	-	-	0.21	-	-
drift-straight	C	0.51	0.68	0.87	0.08	0.11	0.18
drift-turn	C	0.78	-	-	0.11	-	-
elephant	A	0.80	0.79	0.94	0.10	0.12	0.19
flamingo	A	0.61	0.61	0.83	0.10	0.15	0.11
goat	A	-	-	-	-	-	-
hike	H	0.70	0.70	0.93	0.05	0.04	0.07
hockey	H	0.68	0.76	0.86	0.10	0.08	0.17
horsejump-high	AH	0.58	0.68	0.83	0.04	0.09	0.13
horsejump-low	AH	0.81	0.77	0.86	0.09	0.05	0.08
kite-surf	H	0.75	0.78	0.72	0.04	0.07	0.22
kite-walk	H	0.42	0.48	0.71	0.00	0.00	0.07
libby	A	0.64	0.00	0.88	0.29	0.00	0.27
lucia	H	0.88	0.84	0.94	0.07	0.07	0.08
mallard-fly	A	0.50	-	-	0.11	-	-
mallard-water	A	0.44	0.52	0.95	0.07	0.05	0.07
motocross-bumps	B	0.73	0.41	0.89	0.13	0.10	0.14
motocross-jump	B	0.53	0.54	0.86	0.14	0.13	0.17
motorbike	B	0.76	0.61	0.86	0.21	0.31	0.26
paragliding	H	0.84	0.81	0.94	0.02	0.03	0.12
paragliding-launch	H	0.66	0.52	0.64	0.05	0.02	0.04
parkour	H	0.69	0.77	0.94	0.08	0.08	0.15
rhino	A	0.27	0.17	0.93	0.08	0.04	0.09
rollerblade	H	0.83	0.91	0.90	0.05	0.06	0.12
scooter-black	B	0.82	0.77	0.91	0.11	0.14	0.13
scooter-gray	B	0.77	0.68	0.82	0.12	0.10	0.15
soapbox	M	0.09	0.09	0.60	0.29	0.16	0.10
soccerball	M	0.93	-	-	0.01	-	-
stroller	M	0.38	0.36	0.75	0.02	0.02	0.17
surf	M	0.87	0.80	0.96	0.02	0.05	0.07
swing	H	0.41	0.61	0.84	0.02	0.02	0.18
tennis	H	0.72	0.73	0.94	0.10	0.09	0.17
train	M	0.52	0.30	0.89	0.05	0.01	0.10
Overlap[2]		0.61	0.57	0.82	0.08	0.08	0.15
Total		0.63	0.57	0.82	0.09	0.08	0.15

Table 4.3: Results of segmentation performance of our approach on DAVIS, compared to Mask R-CNN (MRCNN). Detailed analysis found in Section 4.3.1.
[1]Group abbreviation found in Table 4.2.
[2]Average over the videos which could be evaluated with both approaches.

For instance the video sequence ‘bmx-bumps’, example in Fig. 4.6, fails with every approach. The reason why it fails is multi-layered:

1. The ‘dense’ approach fails because Mask R-CNN is not able to extract valid masks in every frame, which is because there are many partial and full occlusions of the object.
2. Because there are relatively few masks which can be reliably extracted, the ‘sparse’ approach relies a lot on the tracking routine to compensate, which cannot cope very well with full occlusions either.
3. The object that is to be tracked is black, with very little texture. This makes the tracking particularly difficult as the other occluders in the scene are also black, which increases the probability of the tracker getting ‘stuck’ on the occluder.



Frame 1 Frame 34 Frame 40 Frame 52

Frame 58 Frame 64 Frame 84 Frame 90

Figure 4.6: Sample frames from DAVIS ‘bmx-bumps’ video sequence, which show that the object of salience is not present in every frame, and is in many of the frames partially, mostly, or fully occluded. These occlusions are part of the reason why both of our approaches, ‘sparse’ and ‘dense’, are not able to extract meaningful STVIs, which leads to a very bad segmentation performance.

In essence, most of the failures are due to occlusion problems; a possible solution to this is to split the video into smaller blocks which are processed individually. This results in block-wise STVIs. This way, failures during one block would only affect that particular block, leaving potential room for the other blocks to work correctly. This is discussed in more detail in Section 5.1.

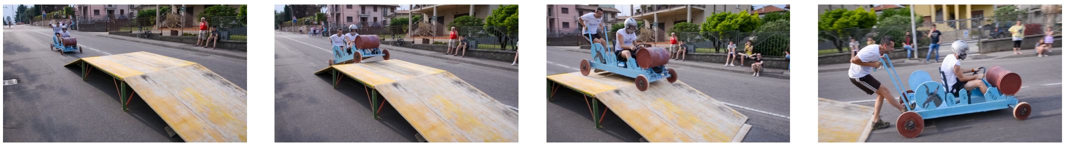
Table 4.3 also lists the average segmentation performance; our approaches have an accuracy that is worse than Mask R-CNN by $\approx 20\%$, and an under-segmentation error that is better by ≈ 10%. This shows that our initial observations - that Mask R-CNN yields slightly too large masks results in a higher accuracy, but also a higher under-segmentation error - are true.

Keep in mind that the two chosen measures, accuracy and under-segmentation error, are suited to determine the performance of a segmentation method. Our approach however does **not** aim to be a segmentation method, but rather a representation of salient objects which can be decomposed into meaningful parts, given an automatically determined scale. As we will discuss in Section 5.2, paragraph *'Object scale & decomposition into parts'*, this scale selection plays a vital role in the ability of our extracted STVIs to segment the object of interest well. While our automatic scale selection works quite well for many cases, see Table 4.3, it also means that the poor segmentation performance can also be attributed to a bad selection of scales in certain videos. The particularities of when the automatic scale selection fails are discussed in more detail in Section 5.1, paragraph *'Automatic parametrization not working'*.

		Average accuracy			Average under-seg. error		
Group		**Sparse**	**Dense**	**MRCNN**	**Sparse**	**Dense**	**MRCNN**
Human	(H)	0.61	0.71	0.85	0.06	0.05	0.12
Animal	(A)	0.58	0.52	0.89	0.09	0.07	0.15
Car-like	(C)	0.72	0.65	0.92	0.10	0.17	0.24
Bike-like	(B)	0.69	0.59	0.86	0.13	0.14	0.16
Misc.	(M)	0.55	0.38	0.80	0.07	0.06	0.11
Animal + Human	(AH)	0.69	0.72	0.84	0.06	0.07	0.10

Table 4.4: Segmentation accuracy and under-segmentation error, averaged by group. Mask R-CNN again shows that its Accuracy is far higher than any of our approaches, but at the cost of having a higher under-segmentation error.

While Table 4.3 gives insight about how each individual video performs, it does not reveal if our proposed approach works equally well on different types of videos. The segmentation performance for whole groups of videos is shown in Table 4.4. From this table we can see that the performance is rather constant throughout the groups. The only outlier is the accuracy score of the 'dense' method on the 'Misc' group, with 38% while the overall average accuracy is 57%. This outlier can be attributed to the fact that the category only contains 5 sample videos, where the performance of one video - 'soapbox', is the worst in the whole dataset: it only achieves an accuracy of 0.09%. The reason for this poor performance is that there are multiple objects in the scene for which valid anchors are found. Additionally, the ground truth segmentation segments three objects: the soapbox car, and two people pushing it. Because of this cluttering with multiple objects our STVIs latches onto the 'wrong' objects, which are in the background, and not the salient object. Samples from the video sequence are shown in Fig. 4.7.



Frame 1 Frame 24 Frame 50 Frame 83

Figure 4.7: The dataset 'soapbox' performs poorly and drags down the whole average accuracy score of the category 'Misc.', cf. Table 4.4. The reason for the poor performance is the drastic change of the object's size throughout the video.

When we compare the 'sparse' and 'dense' methods against each other we can see (result Table 4.3) that they perform approximately the same with a variation of $\approx 4\%$ accuracy and $\approx 1\%$ under-segmentation error. These differences are within an acceptably small range, and are indicative that there are no large discrepancies between the two approaches. Furthermore, Table 4.2 shows the per group success rate of being able to extract STVIs; from this we can see that the sparse approach is able to process 8 videos more, which in the case of a database only consisting of 50 videos is quite significant. It means that our goal of creating a method that

1. does not rely on Mask R-CNN masks in every frame, and therefore is more robust,
2. is computationally cheaper by a factor of 2, and
3. yields better or same results as the 'dense object masks'

was **successfully** met.

Because DAVIS contains only 50 videos it is difficult to anticipate how our approach would perform given a larger set of videos, where the video quality might not be as good. But we note that since there are no unexplained and surprising results observed in *any* of the 50 videos, we are confident that failures on other videos should be limited; an overview of possible improvements of our approach follows in Chapter 5.

4.3.2 Various spatio-temporal scales

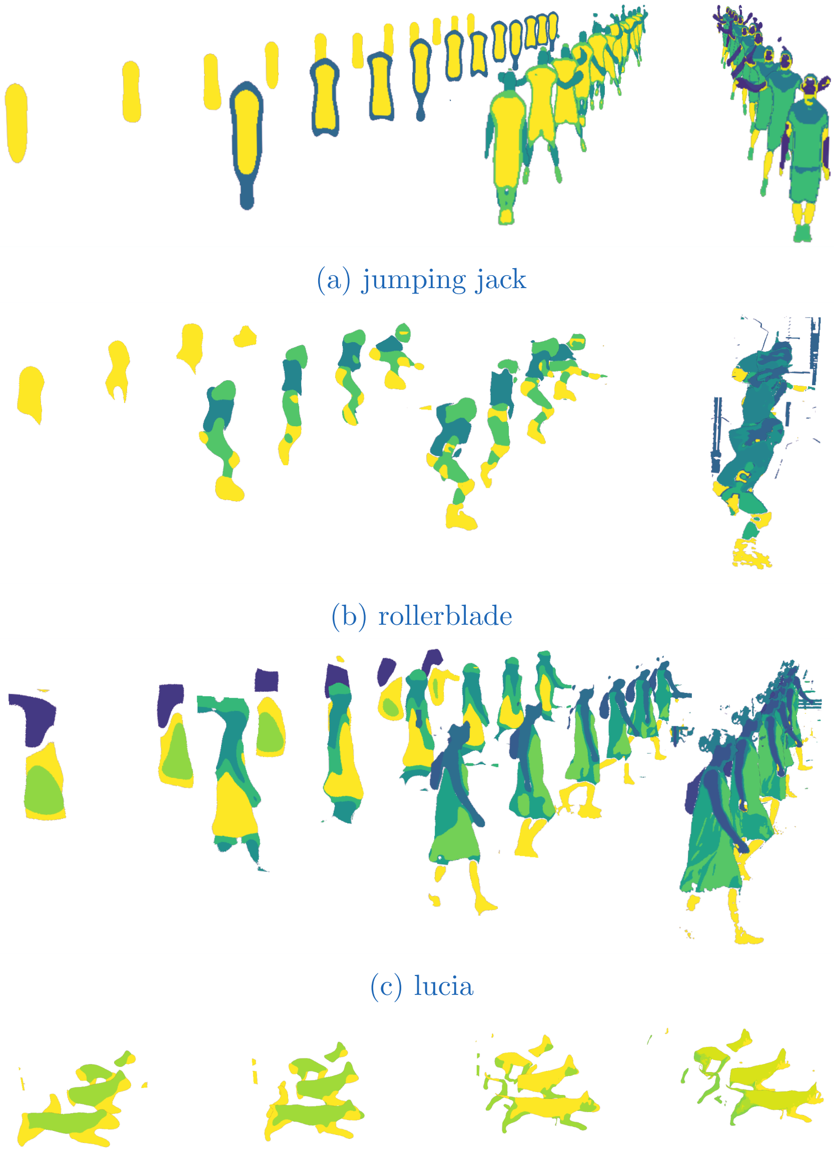
Our method for automatically generating STVIs provides a means to estimate an appropriate scale for the object of interest, both in spatial, and temporal extent. It is however possible to set the parameters that are responsible for the scale manually. These parameters on which the choice of scale depends are σ, τ, k, as explained in Section 3.2.1.

Our goal is to qualitatively assess how the generated STVIs behave and look at different scales. In Fig. 4.8 we show 4 granularities, from coarse to fine. In this Figure we can see how coarser levels correspond to blob-like structures, whereas finer levels reveal more delicate details. Especially note how the representation of a person at a very coarse scale reduces the number of tubes that are needed to approximate the essence of the spatio-temporal extent of the object. In our opinion the three representations of humans Fig. 4.8(a,b,c) exhibit similar structures at coarse scales, and differentiation only occurs at finer levels.

These observations are in line with our expectations towards a well-behaved spatio-temporally scalable representation.

Such a representation at coarse or fine scales might be needed depending on the type of application: Visualizing the movement of certain object in a scene might not require the full detail of arms, legs and head to be modeled; it might be more suited to simplify the shape into a blob, so that the scene which is being represented might be uncluttered and easier to understand.

Because our method seems to provide this well-behaved scalable representation, one could imagine building a scale pyramid of STVIs of the salient object. This pyramid could then be useful to determine at which scale the tubes exhibit some behavior that one might be particularly interested in. For instance consider the scale-pyramid representation of the 'jumping jack'. To recognize the up-and-down motion, it is only necessary to view the STVI at a very coarse scale, while if one is interested in the motion of the arms, a finer scale has to be selected.



(a) jumping jack

(b) rollerblade

(c) lucia

(d) libby

Figure 4.8: Space-Time Volumes of Interest at different scales. The scales are manually chosen with increasing granularity from left to right. Automatic parameter selection, as introduced in Section 3.2.1, is used in the third column of every Figure. We can see how coarser scales correspond to more blob-like shapes which do not model thin parts like arms and legs. We can also see that fine scales estimate small details better, but also have the downside of introducing some artifacts as can be seen especially in the finest scale of (b), and (c).

4.4 Evaluation summary

We have shown that our evaluation of the DAVIS dataset yields consistent results across different types of classes of input videos. While the accuracy of our two proposed approaches cannot compete with state-of-the-art segmentation methods, it does yield a better score w.r.t. under-segmentation error, meaning that less background is recognized wrongly as belonging to the object. Furthermore, keep in mind that our approach was not designed with the goal to be a segmentation method, but rather a method to extract a representation of a salient object, which is decomposable into different parts, at varying spatio-temporal scales. The segmentation performance is only used as a proxy to measure how well our representation fits the actual data. When we consider the qualitative results of our representation at varying scales, we can confidently say that the generated representation is well-behaved, meaning that there are no unpredictable jumps that occur when changing from one scale to a slightly coarser or finer one.

Our visualization of the generated volumes of interest is done by sampling from them along any of the dimensions. This allows us to visually inspect the resulting tubes that model certain parts of the object over time. This type of visualization also allows us to keep the generated representation in the object-centered perspective, where it is easier to see how the object's parts move w.r.t. each other, and it also allows us to back-project the volume in the viewer-centered perspective. The viewer-centered perspective emphasizes the motion of the object throughout the scene, but of course only w.r.t. the camera. This enables us to also show a visualization where the generated STVIs are superimposed on each frame of the input video.

5 Discussion

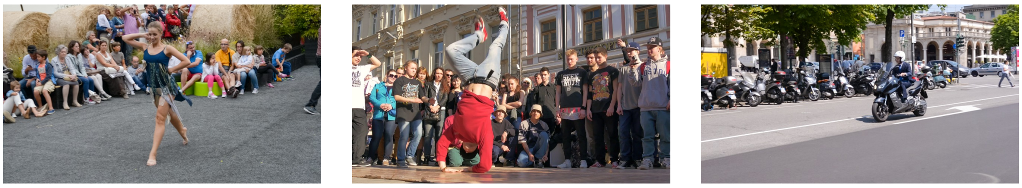
As briefly touched upon in Section 4.3, there are a number of things our novel representation tries to accomplish. The goal our representation tries to solve is to be focused on a salient object where the level of detail is variable, and the object decomposable into meaningful parts. We tried to evaluate these goals quantitatively with the performance defined by segmentation accuracy, and under-segmentation error. These measures of the segmentation performance might not be the most appropriate ones, because we are dealing with video data; the accuracy and the under-segmentation error are averaged over the whole video. It is therefore hard to tell if there might be parts of the video where the segmentation works good, and other parts of the same video where the performance is poor. Additionally, it is hard to evaluate a 'decomposition into meaningful parts' quantitatively, because of missing ground truth segmentation of said 'meaningful parts'. And even if such a database were available 'meaningful' is vague and hard to measure, while easy to understand intuitively by humans, given a video. We therefore restrict ourselves to a qualitative analysis of this 'meaningfulness'.

In general our method for extracting STVIs at different scales has been shown to work well for some videos, cf. Fig. 4.4 and others throughout this thesis. To paint a complete picture of our method we want to elaborate on why it worked well in those cases, and investigate the failure cases, especially w.r.t. DAVIS videos that showed poor segmentation scores, see Table 4.3.

5.1 Failure cases

In our experimentation, we were able to discern distinct types of videos which result in below average STVI performance. We discuss those issues, and show examples that are representative of them; keep in mind that a poor performance of our representation on a particular video might be the combination of multiple types of issues of varying severity.

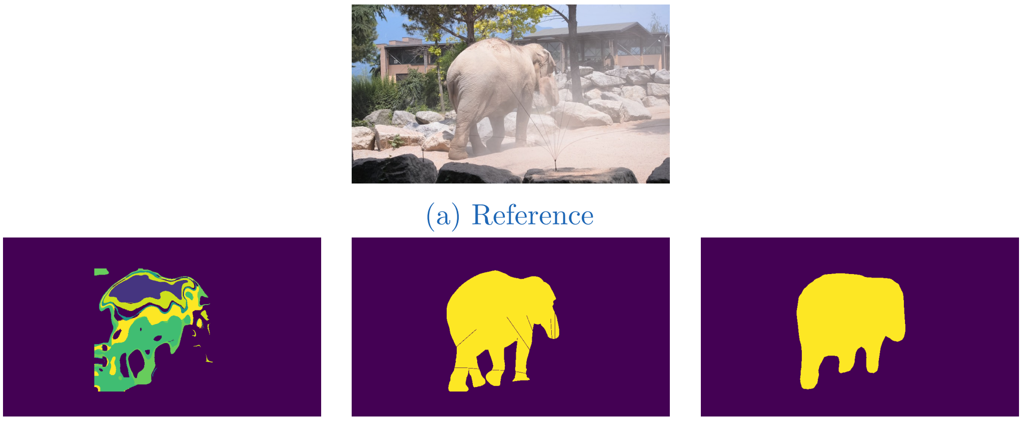
Detection & tracking in cluttered scenes Errors are introduced in the detection and tracking module, and related to the tracking performance of the object in question. These types of errors are especially apparent in the video sequences that are cluttered with many objects of the same type. This is due to the fact that our approach has difficulties keeping track of individual object instances if they are of the same class. Examples of this are shown in Fig. 5.1. In the particular case of ‘scooter-black’ Fig. 5.1 (c), we are able to extract STVIs even though the scene is very cluttered with same class objects. This is most likely because the background motorbikes do not have a human sitting on them and are therefore not as prominently recognized by Mask R-CNN. Furthermore, since the background objects are not moving nearly at the speed at which the salient object is moving, the optical flow helps the delineation of the object we are interested in during the SLIC clustering. This is in contrast to Fig. 5.1 (b), where the break dancer is often not recognized properly because of the ‘weird’ poses he is assuming.



(a) dance-twirl (b) breakdance (c) scooter-black

Figure 5.1: Videos from DAVIS, which contain objects of the same class as the salient object that is masked in the ground truth. The cluttering of the background leads to many Mask R-CNN detections, for which matching and tracking is performed. Because of the multitude of objects this results in wrong matching and tracking, and may lead to malformed STVIs.

Automatic parametrization not working Our proposed method for estimating the parameters from the objects' size and the optical flow might sometimes not be the most suitable way of choosing parameters. This is glaringly obvious when we are dealing with large homogeneous objects, that move slowly. Fig. 5.2 shows that case, where an elephant moves very slowly through the scene. As it is smoothed by a large amount, the already very homogeneous body is made even more uniform; because the elephant is also moving very slowly in the sequence there is also very little optical flow that can be used to anchor tubes. Therefore, the segmentation into 'meaningful parts' is not successful, and the STVI representation of the elephant is left with holes in the body, and is a generally poor segmentation.

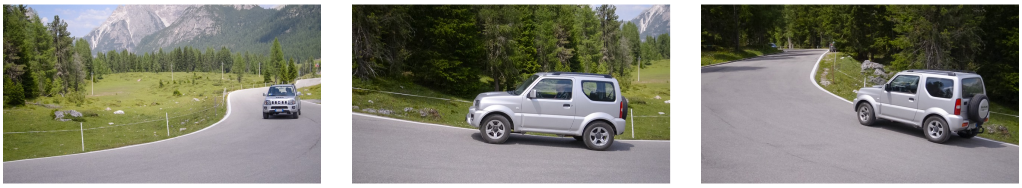


(a) Reference

(b) Our method (c) Ground truth (d) Mask R-CNN

Figure 5.2: Bad STVI: Example of a very homogeneous salient object (gray elephant) w.r.t. appearance, that also happens to move very slowly throughout the video. This results in malformed STVIs, as there is both very little appearance and flow information that differs from the background that can be used to segment the object successfully.

Changes of the object size A further type of detection, and tracking issue is shown to exist in the 'car-turn' dataset. The apparent size of the object changes drastically over time. Mask R-CNN has no issues with detecting the car in every frame, and our tracking solution also has no issues with extracting the correct correspondences, see Fig. 5.3. The problem arises because tube segments which in the beginning of the sequence might model the car well, do not model it well later, because of the drastic changes in appearance and size.



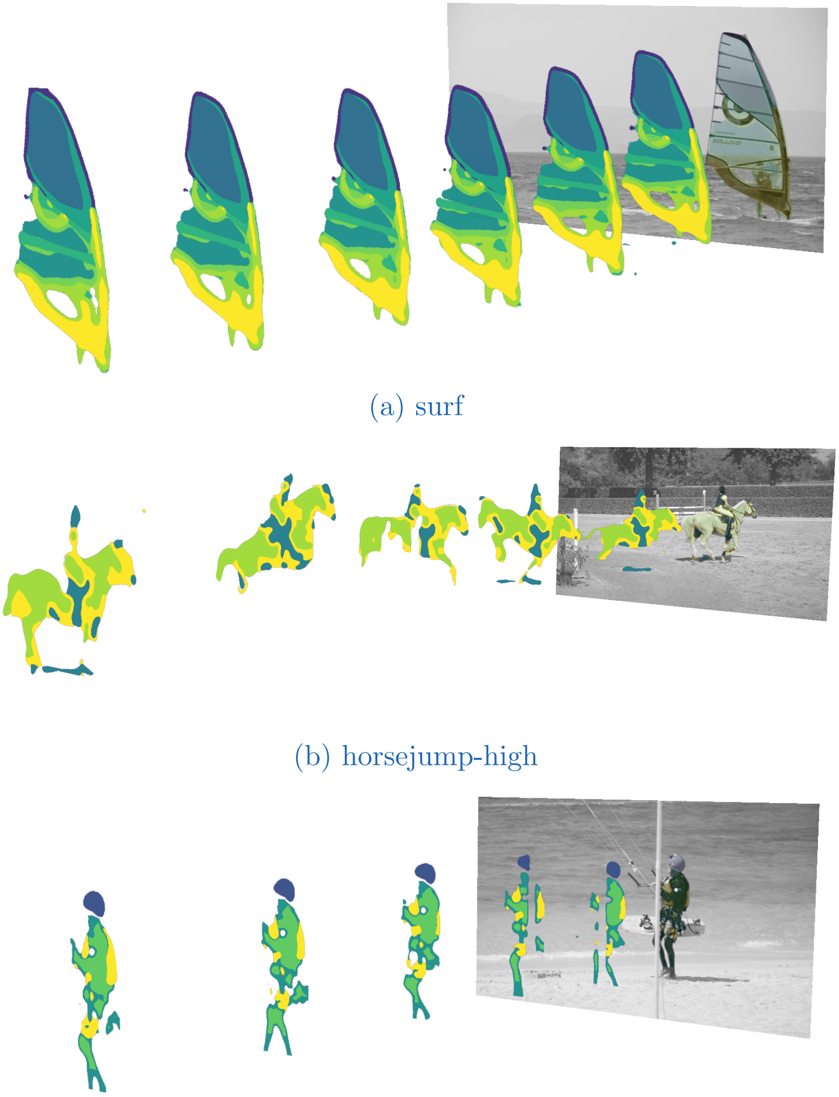
Frame 1 Frame 61 Frame 80

Figure 5.3: STVI fails: Drastic changes in shape and size of the object, in DAVIS 'car-turn'.

More generally: our approach works best if the tubes that model the object persist throughout the whole video. This is because our tube selection works by considering how much of a tube intersects the Mask R-CNN segmentation. This means that if the same part of an object is modeled in the beginning by one tube and at the end of the video modeled by the other, it is likely that one of the tubes will not be selected in the final representation. A possible solution for this is to not consider the whole video, but only partial segments at a time, extract tubes from the section of the video, and then stitch the appropriate ones together. This might alleviate issues such as the ones discussed but would need solving of other issues, such as, how the blocks of video and the resulting STVIs should be 'stitched' together.

5.2 Successful cases

Even though there are some videos that cause errors in our proposed method of extracting a representation of salient objects from video, there are many cases which demonstrate its usefulness. This section answers the question on which types of videos our approach works best, and highlights the particular usefulness of the representation that is generated. Examples of results which are not only quantitatively good, see Table 4.3, but also show a good quality w.r.t. accuracy at object borders, decomposition into meaningful parts, and object scale, are shown in Figure 5.4.



(a) surf

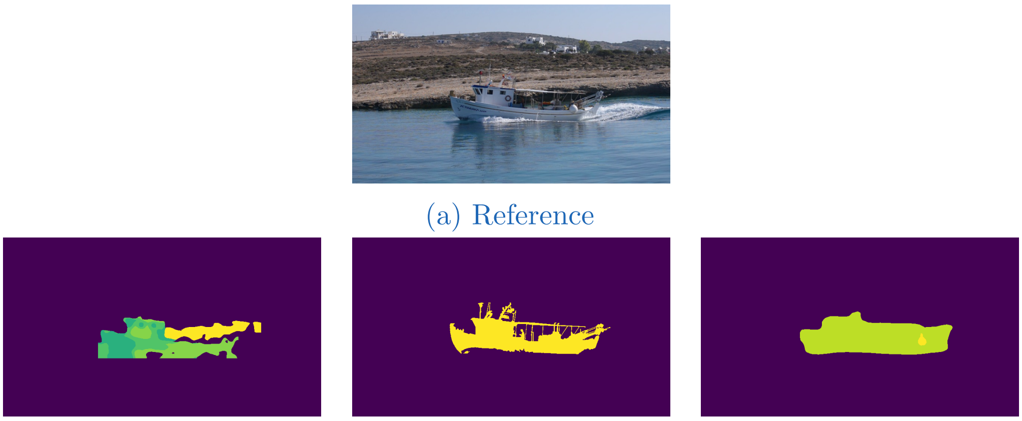
(b) horsejump-high

(c) kite-walk

Figure 5.4: Good STVIs: Examples from the DAVIS dataset for which the generated STVIs perform very well. The resulting representation captures the object boundaries rather accurately, stays consistent throughout the whole sequence, and is able to decompose the objects into meaningful parts. The main reason why these videos have such a good performance is that there is very little background clutter, and that the partial occlusions do not cover a lot of surface area of the object. In the last slice of the visualization the corresponding image in the video sequence is overlaid for reference.

Object scale & decomposition into parts The idea is that there is a 'correct' object scale, and the parts into which it is decomposed, which are tightly linked to the choice of the parametrization. This parametrization determines the spatio-temporal scale at which the object of salience is extracted. If the 'correct' scale is chosen the representation is able to model the object as good as possible w.r.t. human assessment[1]. As we have shown in Fig. 4.8, this seems to work quite well. Furthermore, one can observe that the segmentation into 'meaningful parts' works, as the generated same-color regions in Fig. 5.4 seem to resemble a frame-wise over-segmentation, comparable to superpixel algorithms.

Object segmentation borders When the extraction of STVIs is successful the borders of the object are generally sharper and less 'rounded' as the boundaries extracted with Mask R-CNN, see Figs. 5.5 and 4.4. This is in part due to Flownet2.0 yielding very sharp boundaries in the optical flow. Combined with the fact that our method works on image sequences and not individual frames, our method is able to determine the boundaries of an object more precisely. Fig. 5.5 is especially interesting in that regard because the segmentation


(a) Reference

(b) Our method

(c) Ground truth

(d) Mask R-CNN

Figure 5.5: Clear qualitative improvement compared to Mask R-CNN on video 'boat'. The object class is more recognizable from the STVIs generated from our method (a), than the 'blob-like' shape obtained from Mask R-CNN (d).

[1] Cf. also *'characteristic'* scale, as introduced by Lindeberg [18].

performance of Mask R-CNN is considerably better on that dataset (cf. Table 4.3), but the qualitative performance is not that convincing:

In our opinion it is much easier to recognize what the object is based on the representation that our STVIs provide than what Mask R-CNN delivers. This is another indicator that the two metrics chosen to measure the performance of our approach might not be optimal.

Object- and viewer-centered representation & visualization The object-centered view allows us to focus more on the objects and their components moving w.r.t. the object centroids, whereas the viewer-centered representation emphasizes the motion of the objects in the scene. One has to keep in mind that this is not the global motion of the object w.r.t. the scene, but the motion relative to camera and what it is capturing, i.e. viewer-centered[2] frame of reference.

Furthermore, the ability to slice our STVIs along any of their spatial- or temporal-dimension is highly useful to visualize the interaction of different components, see Fig. 5.6.

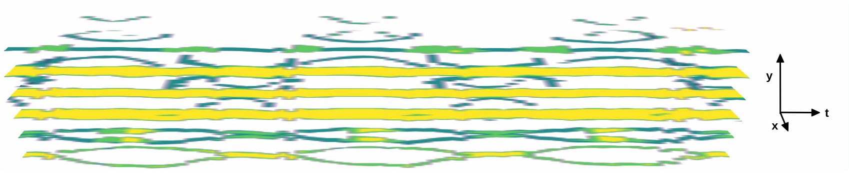


Figure 5.6: Object-centered slices along the y-dimension of the 'jumping jack' STVI shown in Fig. 1.1. Note how the tubes that model the legs and feet form oscillating patterns in space-time, whereas the tubes that model torso and head stay rather constant.

Interaction of multiple objects STVIs are extracted at object level, in the object-centered frame of reference of the salient object. In the case of a video that contains several objects of interest, we can extract several STVIs, one for each object. We demonstrate this on an example from DAVIS 2017 [25] that

[2] The terms *'viewer-centered'* and *'object-centered'* are used according to Marr [22].

shows two pedestrians and a truck, and process the video multiple times, each time considering a different object of interest, see Fig. 5.7. After extracting the three STVIs in their respective object-centered reference frames, we can switch to viewer-centered coordinates to combine all STVIs for this scene, which enables us to view their space-time interactions. Fig. 5.7 also shows the inherent limitations of working in 2D projections of 3D scenes: The groove that is 'carved' into the STVI of the truck is due to the occlusion by the two pedestrians in the foreground. Note that the spatio-temopral scale at which each object is processed varies, because it is individually selected at the object-level.

The ability to select the scale of each object in a scene individually can be used to draw attention to certain classes, which can be used to visualize interactions. Because Mask R-CNN can distinguish between classes such as 'human', 'animal' and 'car', it would be possible to represent animals at a very coarse scale, leaving them as blobs. For more important classes such as humans a finer scale, that still leaves the objects parts' intact might be more appropriate. This can then be used to represent the interactions of objects visually, which can be printed on paper and understood more easily, than showing frames side by side.

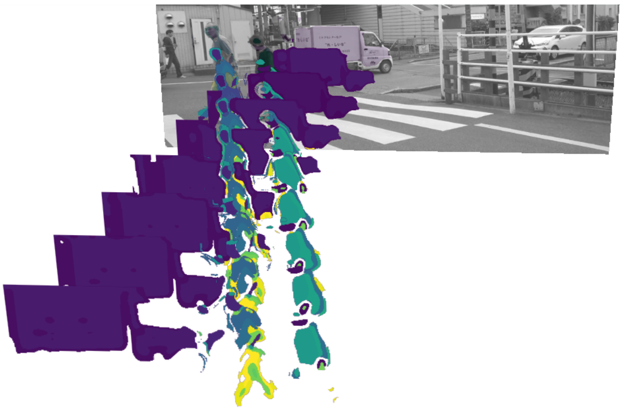


Figure 5.7: Multiple STVIs extracted for a scene with two pedestrians and a truck, and combined in the viewer-centered coordinate system. A coarse scale was chosen for each object, to remove high-frequency noise that would clutter the generated representation. One can clearly observe how the pedestrians and the truck are crossing in space-time.

5.3 Future work

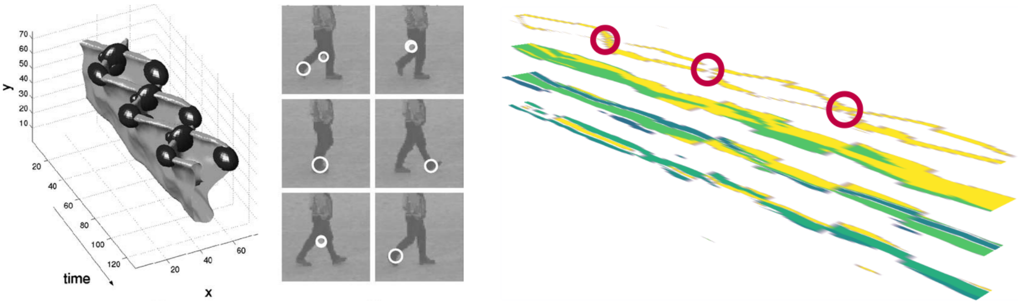
We have shown that a representation that combines top-down information with bottom up grouping processes can be useful to represent objects and their trajectories through the scene. Therefore, it might also be possible to improve our proposed method with a number of extensions. In this section we want to give an overview of possible directions which could be taken to enhance our work.

Additional information Information such as the estimate of a pose could be used to supplement Mask R-CNN, to ensure that the correct region is considered. Pose estimates could be obtained from pose estimation networks such as DensePose proposed by Guler et al. [28], with their downside being that they only work for humans.

Points of interest in the volume could also be used, to ensure that salient regions are modeled by our representation. An example of this could be Space-Time Interest Points [12] (STIPs). We already see that our proposed representation is able to capture regions that exhibit similar patterns to the ones which are recognized as STIPs, see Fig. 5.8.

Feature extraction Extracting features from the created STVIs that are similar to STIPs could be useful for tasks such as action recognition. In general, it might also be possible to extract STIPs of individual tubes enabling us to create *relative* STIPS, i.e. interest points that are meaningful w.r.t. a certain tube.

If it is possible to extract meaningful features from STVIs one could also imagine using it to cluster the features for videos of people that perform actions such as walking and running. If it can be shown that action-patterns performed with different frequencies (running is essentially walking with an increase in amplitude of certain body parts) exhibit similar STVI features it would be shown that the extracted features indeed correspond to intuitively understood patterns when categorizing actions.



(a) Space-Time Interest Points

(b) Motion patterns from STVIs

Figure 5.8: Space-Time Interest points extracted and visualized for a walking person are shown in (a) (Figure taken from the original work by Laptev and Lindeberg. [12]). Note the braid-like weaving of the legs, which corresponds to regions where STIPs are detected. Figure (b) shows an STVI, extracted from the video 'lucia' from DAVIS [24] which shows a person walking, sliced along its y-axis. We can see the similar weaving pattern of the tube that models the person's legs. Both Figures (a), and (b) are 'upside-down' so that the legs of the subject are visible, and not their upper body.

Reconstruction in 4D In our work we represent the full three spatial dimensions which are compressed to two in video, and the additional temporal dimension, i.e. instead of working in four dimensions we worked in three. One could imagine trying to reconstruct the full 4D scene. In a preliminary step this could either be done by using stereo video, or using depth estimation techniques for videos. If such an approach yields promising results it is likely that the represented Space-Time Volume of Interest in 4D would also be able to better represent objects and their interactions, especially since occlusions are easier detected if depth information is available.

6 Conclusion

We have presented a general method to extract Space-Time-Volumes of Interst, STVIs. We use top-down anchors which give us object proposals and track those objects of interest. We generate salient structures based on appearance and motion of the object, in the object-centered frame of reference of the salient object. This solves two problems *at the object level*: (1) selection of spatio-temporal scales, and (2) decomposition into meaningful parts. We can switch between object-centered analysis that reveals salient motion of components of an object, and viewer-centered analysis that shows the overall dynamics of a scene. Besides a fully automated scale selection, we can also select coarser scales that reveal general appearance and motion of an object, and finer scales that show details regarding object components. At the core of our method, we use theoretically well grounded bottom-up processes: spatio-temporal smoothing in Gaussian scale space to preserve object continuity and temporal consistency, and an extension of SLIC towards space-time clustering of object components. This interaction between top-down anchors and bottom-up aggregation presents the strength and the novelty of our approach. In addition, we provide a flexible visualization tool which allows us to view and analyze STVIs w.r.t. any of the dimensions of the volume which might reveal interesting patterns.

The state-of-the-art masking, and optical flow networks have shown their benefits, but might occasionally fail to correctly segment all frames of a video. It will be straightforward to replace the region proposal and masking, as well as the method that is used to extract optical flow by other components providing improved functionality that may become available in the future.

We have also shown that it is not necessary to rely on masks at every frame; with a smart tracking approach that uses masks as initialization we are able to increase the robustness of our approach when comparing it to the naive way of using masks in every frame. The tracking component occasionally requests a re-detection of the whole frame with Mask R-CNN to check if the tracking is still working correctly, and to adjust for slight alignment errors that might occur. Our qualitative evaluation has shown that there is no significant difference between the approach that uses masks in every frame and the approach that uses the masks sparsely.

Qualitatively, we have also shown that our extracted Space-Time Volumes of Interest are able to represent salient objects at different spatio-temporal scales. Our representation is a very *explict* one, making some aspects such as the grouping of parts of an object, and the choice of the scale easy to understand and extend. This ability to understand the generated STVIs might be particularly useful in future work, such as to categorize actions, and interactions between objects.

> *We are stuck with technology when*
> *what we really want is just stuff that works.*
>
> Douglas Adams

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. *SLIC superpixels*. Tech. rep. 2010.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282.

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. "A database and evaluation methodology for optical flow." In: *International Journal of Computer Vision* 92.1 (2011), pp. 1–31.

[4] G. Bradski. "The OpenCV Library." In: *Dr. Dobb's Journal of Software Tools* (2000).

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. "High accuracy optical flow estimation based on a theory for warping." In: *European Conference on Computer Vision*. Springer. 2004, pp. 25–36.

[6] J. Chang, D. Wei, and J. W. Fisher III. "A video representation using temporal superpixels." In: *In Proc. CVPR*. IEEE. 2013, pp. 2051–2058.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. "Detect to Track and Track to Detect." In: *Proc. ICCV*. 2017.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient graph-based image segmentation." In: *International journal of computer vision* 59.2 (2004), pp. 167–181.

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. "Actions as space-time shapes." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.12 (2007), pp. 2247–2253.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN." In: *In Proc. ICCV*. IEEE. 2017, pp. 2980–2988.

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. “Flownet 2.0: Evolution of optical flow estimation with deep networks.” In: *In Proc. CVPR)*. Vol. 2. 2017.

[12] I. Laptev and T. Lindeberg. “Space-time interest points.” In: *Proc. ICCV*. 2003.

[13] I. Laptev. “On space-time interest points.” In: *International Journal of Computer Vision* 64.2-3 (2005), pp. 107–123.

[14] A. Levinshtein, C. Sminchisescu, and S. Dickinson. “Spatiotemporal closure.” In: *Asian Conference on Computer Vision*. Springer. 2010, pp. 369–382.

[15] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. “Turbopixels: Fast superpixels using geometric flows.” In: *IEEE transactions on pattern analysis and machine intelligence* 31.12 (2009), pp. 2290–2297.

[16] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common Objects in Context.” In: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312.

[17] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Springer, 1994.

[18] T. Lindeberg. “Principles for automatic scale selection.” In: *Handbook of computer vision and applications*. Ed. by B. Jähne, H. Haussecker, and P. Geissler. Vol. 2. Academic Press, 1999. Chap. 11.

[19] T. Lindeberg and B. M. ter Haar Romeny. “Linear scale-space I: Basic theory.” In: *Geometry-Driven Diffusion in Computer Vision*. Springer, 1994, pp. 1–38.

[20] C. Liu et al. “Beyond pixels: exploring new representations and applications for motion analysis.” PhD thesis. Massachusetts Institute of Technology, 2009.

[21] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan. “Discriminative Correlation Filter with Channel and Spatial Reliability.” In: *CVPR*. Vol. 6. 2017, p. 8.

[22] D. Marr. *Vision: A computational investigation into*. WH Freeman, 1982.

[23] P. Ochs and T. Brox. “Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions.” In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1583–1590.

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

[25] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. “The 2017 DAVIS Challenge on Video Object Segmentation.” In: *arXiv:1704.00675* (2017).

[26] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks.” In: *In Proc. NIPS - Volume 1*. Montreal, Canada: MIT Press, 2015, pp. 91–99. URL: http://dl.acm.org/citation.cfm?id=2969239.2969250.

[27] X. Ren and J. Malik. “Learning a classification model for segmentation.” In: *null*. IEEE. 2003, p. 10.

[28] I. K. Riza Alp Guler Natalia Neverova. “DensePose: Dense Human Pose Estimation In The Wild.” In: 2018.

[29] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev. “Instance-level video segmentation from object tracks.” In: *In Proc. CVPR*. IEEE. 2016, pp. 3678–3687.

[30] K. Soomro, A. R. Zamir, and M. Shah. *UCF101: A dataset of 101 human actions classes from videos in the wild*. Tech. rep. Center for Research in Computer Vision (CRCV), Nov. 2012. [arXiv preprint arXiv:1212.0402].

[31] J. Sporring, M. Nielsen, L. Florack, and P. Johansen, eds. *Gaussian Scale-Space Theory*. Springer, 1993.

[32] A. Vedaldi and S. Soatto. “Quick shift and kernel methods for mode seeking.” In: *European Conference on Computer Vision*. Springer. 2008, pp. 705–718.

[33] H. Wang and C. Schmid. “Action recognition with improved trajectories.” In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE. 2013, pp. 3551–3558.

[34] C. Xu and J. J. Corso. “Evaluation of super-voxel methods for early video processing.” In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1202–1209.

[35] X. Zhang, T. Dekel, T. Xue, A. Owens, Q. He, J. Wu, S. Mueller, and W. T. Freeman. “MoSculp: Interactive Visualization of Shape and Time.” In: *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM. 2018, pp. 275–285.