DISSERTATION

UNSUPERVISED VIDEO SEGMENTATION USING TEMPORAL COHERENCE OF MOTION

Submitted by

Hessah Alsaaran

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2015

Doctoral Committee:

Advisor: Bruce A. Draper
Co-Advisor: J. Ross Beveridge

Darrell Whitley
Christopher Peterson

ABSTRACT

UNSUPERVISED VIDEO SEGMENTATION
USING TEMPORAL COHERENCE OF MOTION

Spatio-temporal video segmentation groups pixels with the goal of representing moving objects in scenes. It is a difficult task for many reasons: parts of an object may look very different from each other, while parts of different objects may look similar and/or overlap. Of particular importance to this dissertation, parts of non-rigid objects such as animals may move in different directions at the same time. While appearance models are good for segmenting visually distinct objects and traditional motion models are good for segmenting rigid objects, there is a need for a new technique to segment objects that move non-rigidly.

This dissertation presents a new unsupervised motion-based video segmentation approach. It segments non-rigid objects based on motion temporal coherence (i.e. the correlations of when points move), instead of motion magnitude and direction as in previous approaches. The hypothesis is that although non-rigid objects can move their parts in different directions, their parts tend to move at the same time. In the experiments, the proposed approach achieves better results than related state-of-the-art approaches on a video of zebras in the wild, and on 41 videos from the VSB100 dataset.

ACKNOWLEDGEMENTS

First, my deepest gratitude goes to my advisor, Professor Bruce Draper, for his generous guidance, support, patience, and expert knowledge during the development of this work. This dissertation would not been possible without his technical and editorial advice. I am thankful for his careful reading and detailed revisions of this dissertation. I have been fortunate to work with him, and I cannot thank him enough. Special thanks go to my co-advisor, Professor Beveridge, for his supervision and valuable advices. He was my first instructor in the Computer Science Department of Colorado State University, and I would like to thank him for introducing me to the computer vision group in the department. Professors Draper and Beveridge have always found time to meet, listen, discuss, and help overcome the roadblocks of this research, and I am always grateful for them. My sincere thanks also go to my colleague Mr. Dutta for his full assistance in the video stabilization process. Great appreciation goes to Professor Whitley and Professor Peterson for their kind acceptance to review my dissertation and be part of my doctoral committee.

Many thanks go to my family for their endless support and love. I would not have been able to overcome the obstacles in my dissertation journey without them. I am filled with gratitude for my parents, Nasser and Norah. They are the reason behind all the success in my life. I will always be grateful to my sisters, Rasha and Ruba, and my brothers, Abdulaziz and Omar, for their everyday encouraging and motivating words. I have been fortunate to have a loving son who was patient with me throughout my journey, and I am thankful to him. My son, Abdullah, is my bright sun that shines happiness and motivation to me.

Finally, great appreciation goes out to the Department of Computer Science at Colorado State University for all the excellent support I received from both the academic and administrative staff. I also acknowledge that this research would not have been possible without the full scholarship from King Saud University. So a very special thanks goes out to King Saud University for giving me the opportunity to pursue my Ph.D. degree.

DEDICATION

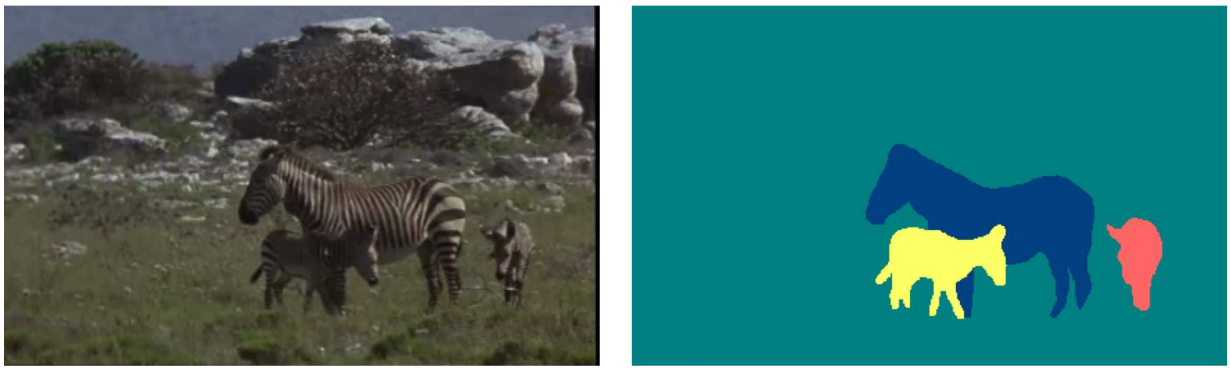
To my beloved family ...

TABLE OF CONTENTS

Chapter 1

Introduction

Spatio-temporal video segmentation groups pixels, with the goal of representing moving objects in scenes. For example, Figure 1.1 shows a family of zebras in the savannah [18], described in more details in Figure 1.2. The zebras all have similar colors and markings, and the babies stay close to their mother, circling her as she grazes. The result is a dynamic pattern of visual occlusions among similar objects. Worse still, zebras are non-rigid; their head, legs and tail typically move in different directions from the torso and from each other. The challenge is to develop a general-purpose motion segmentation algorithm capable of dividing this video into four regions, namely three zebras and the stationary background.



(a) (b)

Figure 1.1: A spatio-temporal video segmentation example. (a) A frame from a video [18] of three zebras walking around. The three zebras walk and stop continuously and independently in an uncorrelated manner. (b) The ideal segmentation of this video shown on this frame. Each color represents a segment, for a total of four segments.

This dissertation presents an unsupervised motion-based video segmentation algorithm designed for non-rigid objects such as animals, but that works for both rigid and non-rigid objects. Previous motion-based segmentation algorithms group pixels (or superpixels) based on consistent motion magnitudes and directions. Unfortunately, non-rigid objects may have

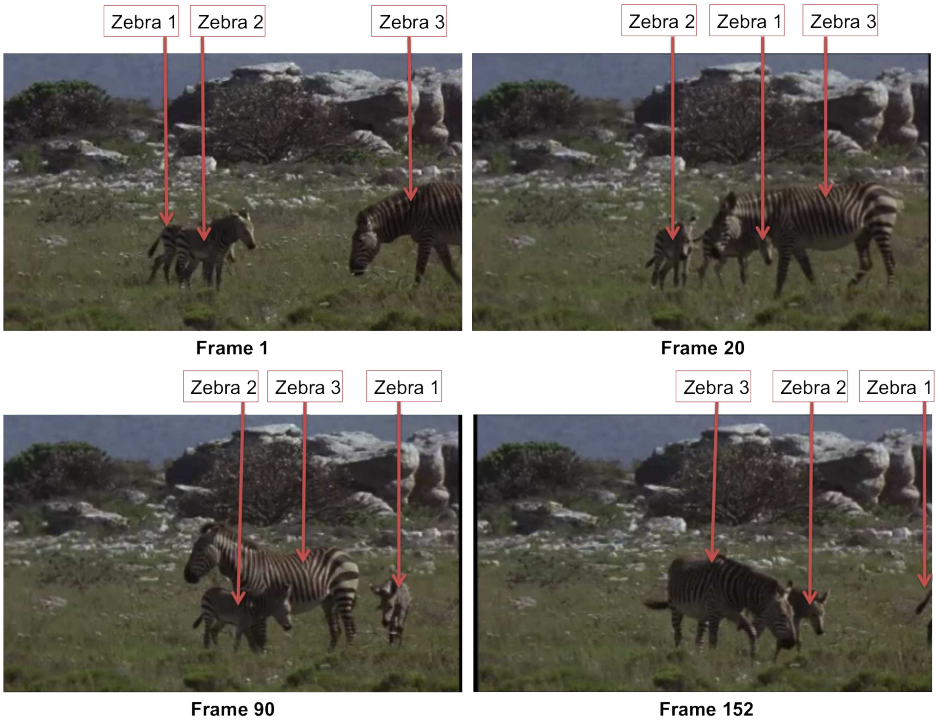


Figure 1.2: An example of three zebras [18], numbered from left to right in the first frame. At the beginning of the video, the third zebra is walking from right to left toward the middle of the frame, while the first zebra is walking in the opposite direction and passing behind the third zebra. Meanwhile, the second zebra pretty much remains in its place in approximately the first third of video. At that time, the third zebra stops briefly in the middle, while the second zebra turns around it in front of it, and the first zebra turns around it behind it. Finally, the third zebra starts to turn around the second zebra and becomes in front of it, while the first zebra walks to the right until it becomes out of view.

parts moving in different directions, as in the zebra example above. The hypothesis of this work is that although non-rigid objects may move their parts in different directions, connected parts tend to move at the same time. From this observation, the key is to identify *when* objects move instead of *how* they move. If two adjacent superpixels often move at the same time, their motions are temporally correlated and they are likely to be parts of a single object, even if they are moving in different directions. The emphasis on the *temporal coherence of motion* instead of motion magnitude and direction is the main contribution of this dissertation. The result is an algorithm capable of segmenting videos containing combinations of rigid and non-rigid objects. We called this algorithm **R**obust **A**nimate **M**otion **S**egmentation (**RAMS**).

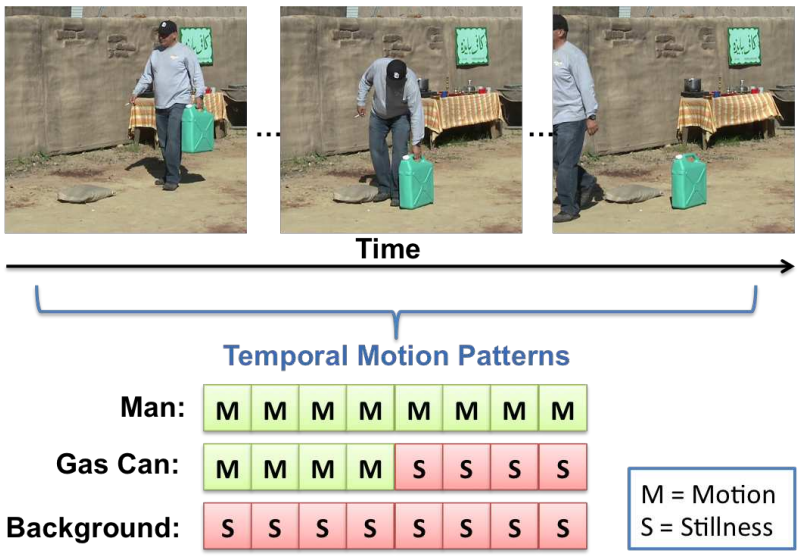


Figure 1.3: A simple example for illustrating how temporal coherence of motion is used in RAMS. Top: A video example of a man walking from right to left, holding a gas can that he places on the ground in the middle of the video. Bottom: An extracted temporal motion pattern for each of the three objects. The idea is that these objects are separable by comparing their temporal motion patterns.

Figure 1.3 shows an example of how temporal coherence can be used to group objects in a simpler video. A segmentation of the video in Figure 1.3 based on homogeneity should produce four moving superpixels: one for the man's shirt, another for his pants, one for the gas can he puts down, and one for the tablecloth he walks in front of. The key observation, however, is that the shirt and pants can be grouped, even though they don't look alike, because they tend to move at the same time. The gas can, on the other hand, is separate because after he puts it down, it stops moving. The tablecloth is yet a third object because it moves when the wind blows, and therefore has a third, distinct pattern of motion timings. Notice, however, that a rigid motion model will not group the shirt and pants, because although they move at the same time, they move in different directions. The insight of this paper is that temporal correlation alone is better than rigid motion models for animated objects, and is sufficient for other object types.

Returning to the more challenging example of the zebras shown in Figures 1.1, even though the zebras look alike, their temporal motion patterns are different; they start and

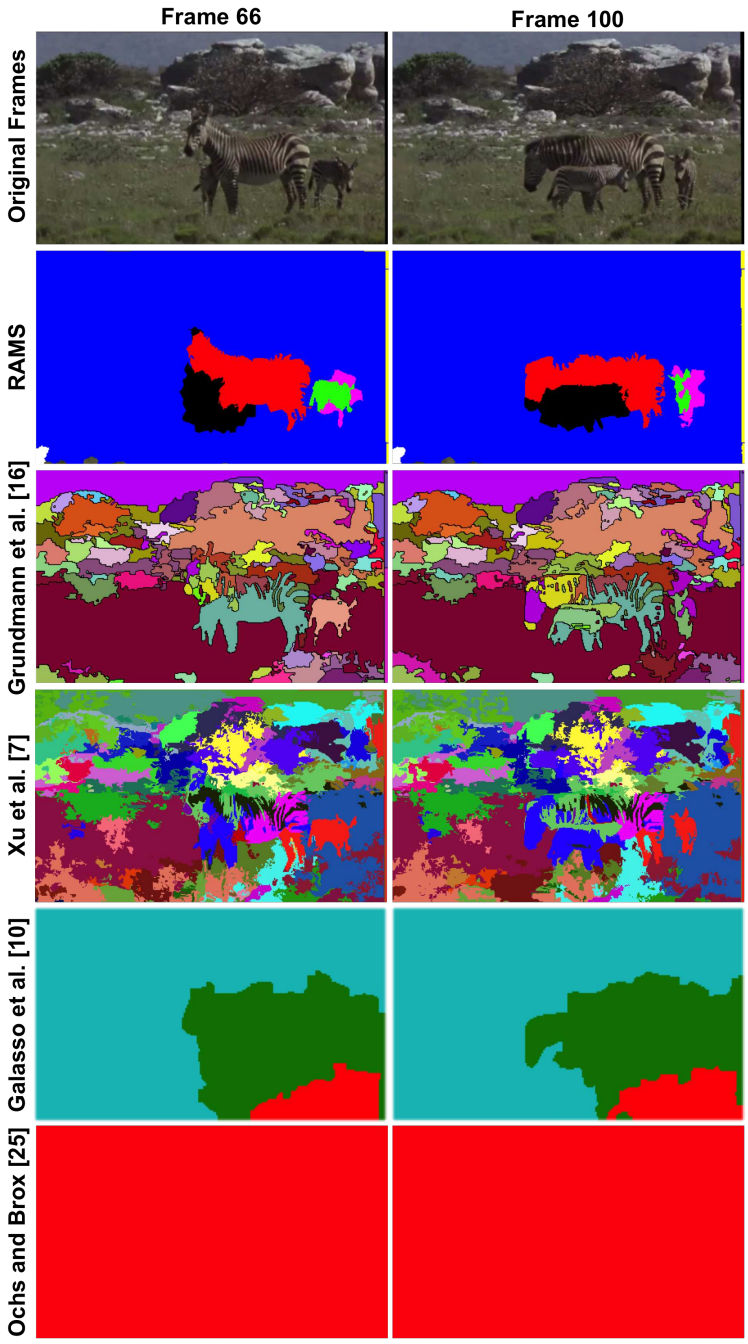


Figure 1.4: Segmentation results for the zebra video example for RAMS, and state-of-the-art approaches: Grundmann et al. [16], Xu et al. [7], Galasso et al. [10], and Ochs and Brox [25], respectively. For each algorithm, this figure shows the segmentation with the fewest clusters such that the zebras are visually separable (when possible).

stop at different times. The temporal coherence of their motions is sufficient to segment the overlapping zebras. The second row of Figure 1.4 shows the segmentation produced

by RAMS on two frames of this video. Rows 3, 5 and 6 of Figure 1.4 show the results of segmentation algorithms based on rigid motion models; they cannot group the zebras without over-segmenting the whole scene because the zebras are not rigid. Row 4 shows the no-more-successful result of an appearance based segmentation algorithm.

To compute temporal motion patterns, RAMS determines whether a point is moving or not. Currently, RAMS assumes a fixed camera to make motion detection simple. This is why we evaluate RAMS on 41 of the 100 VSB100 [9] videos: we test it on every video that does not have significant camera motion (we do compensate for jitter; see Chapter 3). This allows us to evaluate the effectiveness of temporal coherence models, rather than our ability to detect independent motion, although it limits the use of the current implementation to surveillance applications and other scenarios with fixed cameras. In the future, we expect to integrate RAMS with algorithms that detect independently moving points, for example [11, 31], to remove the fixed camera limitation, as discussed in Section 5.2.

Temporal coherence of motion may seem like a weak source of information. When segmenting videos, motion magnitudes and directions are generally thought of as the important video cues, based on intuitions from rigid objects. Given a video of cars on a street, for example, points that move with the same magnitude and direction should be grouped together, since they are likely to belong to the same car, while points that move in different directions are likely to belong to different cars. Notice that this is only true for rigid objects, however. Non-rigid objects, such as humans and animals, have parts that move in different directions. RAMS is the first approach that disregards motion magnitude and direction, and replaces them with a novel motion cue: temporal coherence. Figure 1.4 shows how well RAMS segments the zebras in the example of Figure 1.1 using this new cue. Just as important, the videos from VSB100 [9] show that temporal coherence outperforms rigid motion models and videos selected for other purposes, in part because some of the objects in these videos are non-rigid, and in part because temporal coherence is sufficient for rigid objects, too.

The term "temporal coherence of motion" has been used by previous video segmentation algorithms to mean "temporal coherence in motion type" (i.e. pixels of a segment temporally move together in the same direction and magnitude). In the proposed work, we mean the exact words of "temporal coherence of motion" as we use motion timings (i.e. pixels of a segment move at the same time, regardless of direction or magnitude). Although our use of the term is correct, there is a difference in definition that could confuse a reader. The idea of video segmentation based on when movements happen, not on how the movements happen, is counter-intuitive, but the idea is simpler and more general, and it works better for segmenting non-rigid objects.

The segments of a segmentation result can be further used for object recognition or action recognition. Segments provide localized and freely-shaped 3D regions for object recognition. Furthermore, the motion within a segment can also be analyzed for more localized object action recognition. The work in [35] by Ke et al. is an example of the use of video segments in action recognition. Other applications include the generation of automatic segmentations to reduce human effort. Automatic segmentations simplify video semantic labeling as a user can label a segment once, and then all labels are propagated to all frames. The annotation tool of [17] is an example of this application.

The next chapter describes some related work and background material. The third chapter describes the proposed approach, RAMS, in detail. The evaluation results are reported in the fourth chapter. The fifth and final chapter includes the conclusions and future work.

Chapter 2

Related Work

Image segmentation research preceded video segmentation research, and is a much more mature area. A video is an image sequence, but the introduction of motion in videos admits new sources of information. Thus, video segmentation is not a trivial extension of image segmentation. If a video segmentation approach segments each frame independently, it cannot use the motion information in videos. More importantly, If each video frame image is segmented independently, the frames' segmentations would suffer from segmentation inconsistency. Therefore video segmentation approaches process multiple frames at a time, either by processing the whole video at once, or by processing sliding windows of frames in a streaming mode while exploiting past segmentations. The first method has the advantage of acquiring all the motion information at once, and thus it has more information for better segmentation decisions. On the other hand, streaming methods have the advantage of breaking large videos into smaller parts to reduce memory and computational expenses. Section 2.2 presents some related video segmentation approaches.

Many video segmentation approaches, including the proposed approach (RAMS), use image over-segmentations called *superpixels* as a basis for video segmentation. Superpixels provide the segmentation unit for RAMS, presented in Chapter 3. Thus superpixels, superpixel methods, and their extension to videos, are described first in the following section.

2.1 Superpixels

Image segmentation partitions images into regions, using color and texture information, with the goal of representing the different objects in an image. Since image segmentation is

highly dependent on the domain area, many applications that use image segmentations resort to image over-segmentations. An over-segmentation of an image produces more and smaller regions, such that each region is a set of compact and homogeneous pixels that respects the boundaries of the image. These over-segmentations are called *superpixels*.

Since similar neighboring pixels are grouped together in a superpixel segmentation, many video segmentation approaches pre-process a video by obtaining its superpixel segmentation, and then segment the superpixels. This significantly reduces the number of units to segment when compared to segmenting at the pixel level. In addition, many neighboring pixels are similar and have redundant information, thus segmenting at the pixel level presents unnecessary workload since superpixel segmentations pre-group similar neighboring pixels. This section starts by describing some single image superpixel segmentation algorithms, and then it describes an approach that extends superpixels to videos by Chang, Wei, and Fisher III [12]. RAMS segments the superpixels of [12], so these algorithms are reviewed in detail.

2.1.1 Image Superpixel Methods

Two state-of-the-art and widely used superpixel algorithms are simple linear iterative clustering (SLIC) by Achanta et al. [27], and graph-based image segmentation (GBIS) by Felzenszwalb and Huttenlocher [22]. The following is a brief description of each of these two algorithms, and Figure 2.1 shows a segmentation example for both of these algorithms.

2.1.1.1 SLIC Superpixels [27]

The SLIC algorithm [27] represents pixels in terms of their Lab colors (l, a, b) and image locations (x, y), and performs a modified version of k-means clustering of the N pixels in this 5D space. The number of clusters (k) is a parameter for this algorithm, and it is set to the desired number of superpixels. The algorithm starts with initializing the clusters' centers by first placing them on a grid with spacing $S = \sqrt{\frac{N}{k}}$, and then adjusting these

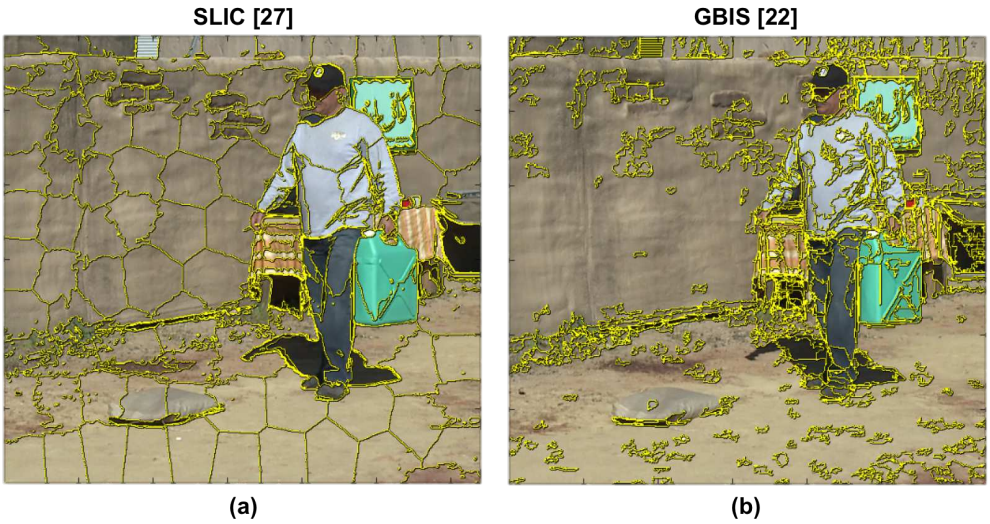


Figure 2.1: Superpixels methods results on the first frame of the video example of Figure 1.3: (a) SLIC [27] result. The two parameters were set as follows: (i) the approximate number of superpixels was set to 100, and (ii) the compactness ratio m was set to 20. (b) GBIS [22] result. The two parameters were set as follows: (i) scale, for region size, was set to 190, and (ii) minimum region size was set to 50.

locations to the lowest gradient location in a 3×3 neighborhood. These centers act as seeds, so the adjustment prevents them from being located on an image edge or a noisy pixel. After initialization,the algorithm iteratively performs two steps: the pixels' assignment step, and the centers' update step.

In the pixel assignment step, each pixel is assigned to the nearest cluster center, using a distance function D. Instead of comparing a pixel with all the clusters' centers, the algorithm only compares a pixel with cluster centers that have a search region that includes this pixel. The search region for each cluster center is the $2S \times 2S$ region centered at the cluster's center, because the approximate size of a superpixel is $S \times S$. For the distance function D, instead of using the euclidean distance in the 5D space, the algorithm computes the euclidean distance in color space (d_c) and spatial image space (d_s) independently, and then combines them in the final distance $D = \sqrt{d_c^2 + (\frac{d_s}{S})^2 m^2}$, where m is the weight for the relative importance between color similarity and spatial proximity, and is another parameter for the algorithm.

After the assignment step, each cluster center is updated to the mean in the 5D space for all the cluster's assigned pixels. The residual error between the new and old centers is computed. The iterations are stopped when this error converges.

Finally, a post-processing step is performed to ensure connectivity in superpixels. A connected components algorithm is performed on the pixels' labels. Each pixel that does not belong to the same connected component as their assigned cluster's center, is reassigned to the same label as the nearest cluster center.

2.1.1.2 GBIS Superpixels [22]

GBIS [22] is a graph-based approach, where an image is represented as an undirected graph. Each node corresponds to a pixel, and the edges connect neighboring pixels. Edge weights are set to the dissimilarity between the connected pixels. The graph is partitioned into a set of connected components/regions using an iterative and greedy region-merging process to obtain an image segmentation.

For each region, an internal difference is computed as the maximum weight in its minimum spanning tree. The difference between a pair of regions is computed as the minimum weight edge connecting these two regions. This difference is set to infinity if there are no connecting edges between the two regions. The algorithm uses a predicate for checking whether a boundary exists between two regions. The predicate evaluates to true if the difference between the pair of regions is larger than the internal difference of at least one of the regions, and evaluates to false otherwise. This predicate is used to control the merging process by preventing a merge of two regions if there was a boundary between them.

The algorithm starts by sorting the graphs edges by their weights in a non-decreasing order, and then greedily process each of these edges in the sorted order. If the edge's connected nodes belong to two different regions, then these two regions are merged only if the boundary predicate evaluates to false. Otherwise, they are not merged.

2.1.2 Extending Superpixels to Videos

Applying superpixel segmentation for each frame of the video independently results in inconsistencies between the frames' segmentations. Chang, Wei, and Fisher III [12] presented an extension of SLIC [27] that generates temporally consistent superpixels (TSP) from videos. RAMS uses these TSPs as a basis for video segmentation. Some other video segmentation approaches extend another superpixel algorithm called GBIS [22] to videos. Examples are: Grundmann et al. [16] and Xu et al. [7] video segmentation approaches. The following is a brief description of the TSP [12] approach. Figure 2.2 shows the consistency of the TSPs and the inconsistency of the per-frame SLIC segmentations for a video example.

2.1.2.1 TSP [12]

Chang, Wei, and Fisher III [12] presented an approach for generating temporally consistent superpixels from videos, and they call them Temporal SuperPixels (TSP). They are actually trajectories of superpixels, since they are tracked through time. RAMS clusters the superpixel trajectories to segment a video.

The TSP [12] approach starts with computing the superpixels in the first frame, and then propagating and updating this segmentation in subsequent frames, going forward in time. This section is divided into two parts: the first part describes how the initial superpixels are computed, and then the second part describes how the superpixels are propagated over time. In each of these two parts, the approach tries to maximize the likelihood of the labeling to arrive at a local optima. So in each of these parts, how the likelihood of a labeling is computed is described first, then how does it start and iterate to arrive to the local optima is described second. The approach takes one input, which is the desired number of superpixels per frame, and it is denoted with M.

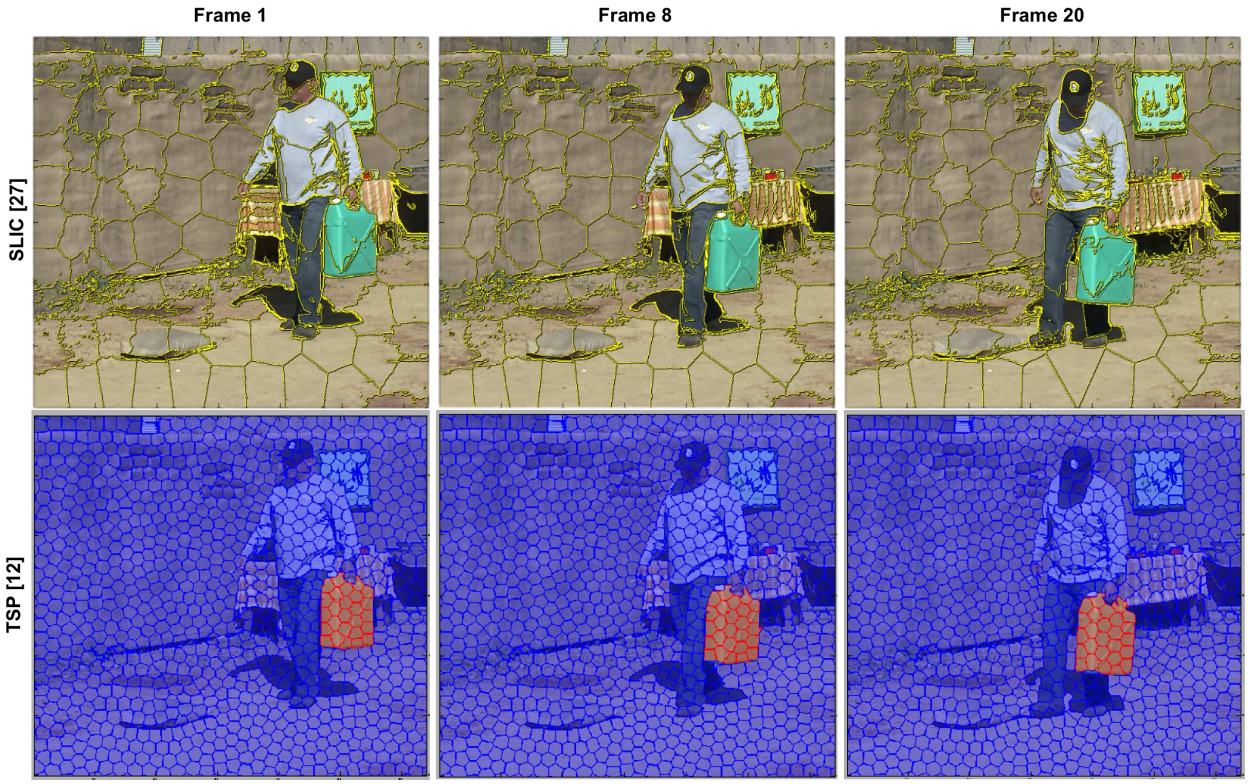


Figure 2.2: TSP [12] vs. SLIC [27] segmentation for the video example in Figure 1.3. The first row contains the per-frame SLIC segmentations for the frames 1, 8, and 20. By looking at the gas can in these frames, you can see that even though its shape and appearance did not differ much, its segmentation is not consistent; it was mostly 2 superpixels in the first frame, a single superpixel in the 8^{th} frame, and 3 superpixels in the 20^{th} frame. The second row contains the TSP segmentation for the same frames. The temporal superpixels of the gas can are highlighted in red in the first frame, and the color is propogated in subsequent frames showing temporal connectivity.

2.1.2.1.1 First Frame Segmentation

The TSP [12] approach is similar to SLIC [27] in producing the superpixels of the first frame, in that it transforms SLIC's k-means clustering to a Gaussian mixture model. Treating each superpixel as a Gaussian distribution with unknown parameters: mean and variance, and that pixels are generated from this mixture of distributions and their labels are the hidden variables, the approach tries to maximize the likelihood of the pixels labeling. So the superpixels' properties and pixels' labels are statistically inferred from the probabilistic model. In contrary to SLIC [27] that verifies connectivity as a post-processing step, the

connectivity in the TSP approach is ensured by incorporating a restriction on the labels' distribution. A proposed labeling z is considered valid only if all the superpixels are single 4-connected regions. So any invalid labeling will have a zero prior probability: $P(z) = 0$.

2.1.2.1.1.1 Likelihood of a Labeling

Given an image frame, each pixel i has a Lab color a_i and a location ℓ_i. Each pixel is therefore assigned a 5-dimensional vector. Let N denote the number of pixels in the frame, and K denote the number of superpixels. Each pixel i has a label z_i in $1..K$ corresponding to the label of the superpixel where it belongs. Each superpixel k has two means: μ_k^a and μ_k^ℓ, for color and location respectively. The approach assumes all superpixels have the same and fixed variances: σ_a^2 and σ_ℓ^2, for color and location respectively.

The prior probability of an event is the probability of this event occurring without seeing the observations and taking them into account, thus requires some prior knowledge. The prior probability of a labeling z (i.e. without looking at the observed pixels) is modeled by restricting the distribution of labels for connectivity, and the geometric distribution on the number of superpixels:

$$p(z) \propto \hat{\alpha}^K Valid(z) \tag{2.1}$$

where $Valid(z)$ is one if all the superpixels are single 4-connected regions, otherwise its zero. $\hat{\alpha}$ is a parameter that controls the coarseness of the superpixels. The coarseness refers to the color homogeneity; larger $\hat{\alpha}$ values promote fewer superpixels, thus larger and coarser superpixels. The prior probabilities of the superpixels' means is set to be from a uniform distribution:

$$p(\mu_k^\ell) = \frac{1}{N} \qquad , \qquad p(\mu_k^a) = (\frac{1}{256})^3 \tag{2.2}$$

assuming each color component has the value range [0,255].

The approach derives an optimal solution from an initial labeling. From a given labeling, a set of movements are proposed, and each proposal has its implicit optimal mean parameters for it. The approach accepts and performs the movement proposals that increases the labeling likelihood to arrive at a local optimum. The approach jointly optimizes the labels and the parameters, and the joint log likelihood is defined as:

$$\mathcal{L}(z) \stackrel{\text{C}}{=} \log[p(z) \prod_k \prod_d p(\mu_{k,d}) \prod_i p(x_{i,d}|z_i, \mu_d)] \tag{2.3}$$

where d indexes the 5 dimensions (color and location) of pixels x_i, and the symbol $\stackrel{\text{C}}{=}$ means equality up to an additive constant. If the topology of z is assumed to be valid (the meaning of the $\stackrel{\text{T}}{=}$ symbol), then joint log likelihood can be rewritten as:

$$\mathcal{L}(z) \stackrel{\text{T}}{=} \alpha K + \sum_k \sum_d \log p(x_{I_k,d}|z_i, \mu_{k,d}) \tag{2.4}$$

where constants are combined into α, and the pixels of a superpixel k is denoted by I_k. The number of pixels in a superpixel k is denoted by N_k. Since the prior probabilities of the means are from uniform distributions, the optimal means $\hat{\mu}_{k,d}$ for a labeling z are the empirical means computed from the corresponding pixels. By computing the log likelihood for superpixel k ($\log p(x_{I_k,d}|, \hat{\mu}_{k,d})$) and denoting it with $\mathcal{L}_n(x_{I_k,d})$, the joint log likelihood can be rewritten as:

$$\mathcal{L}(z) \stackrel{\text{C,T}}{=} \alpha K - \sum_k \sum_d \mathcal{L}_n(x_{I_k,d}) \tag{2.5}$$

2.1.2.1.1.2 Labeling Initialization and Possible Moves

The initial hypothesis labeling z to start from and propose changes is not clearly stated in the paper [12]. It appears to start with k randomly-centered superpixels, where k is the number of desired superpixels, and the labels of these centers are flooded to all remaining pixels using a watershed flooding technique.

From a current labeling, a neighborhood of moves is proposed. Since the connectivity of the superpixels are ensured in the model, a proposed move must maintain connectivity. So the approach considers only three possible movement types:

- Single-pixel move: Only "simple points" can change their labels. They are basically the points that lie on the edges of the superpixels, so changing their labels won't affect connectivity of the pixels within the superpixels.
- Merge move: Two superpixels can be merged if the merged result is a single 4-connected region.
- Split move: A superpixel can be split into two superpixels. The two superpixels are identified by performing k-means clustering and then ensuring connectivity, as in SLIC [27].

Considering only the valid changes, the approach accepts and performs the move that increases the likelihood of the labeling. This is done iteratively until there are no more moves, thus it arrived at a local optimum. The result is the superpixels of the first frame.

2.1.2.1.2 Subsequent Frames Segmentation

The superpixels in subsequent frames are inferred from previous frames. The following describes how likelihoods are computed, the initial state, and the neighborhood of possible moves. The approach goes forward through time, processing one frame after another.

2.1.2.1.2.1 Likelihood of a Labeling

Given a new frame t, its labeling is initialized by propagating the labels of the previous frame $t-1$ through optical flow as described in the next subsection. The appearance and locations of superpixels change over time. In addition, the set of superpixels from frame to frame can change; some superpixels can die while some new ones may appear. The superpixels that die, i.e. become occluded or leave the field of view, are called "dead superpixels". The superpixels that are still alive are called "old superpixels", and are denoted with the subscript "o", while the newly appeared superpixels are called "new superpixels" and are denoted with the subscript "n". So K_o and K_n are used to denote the number of old and new superpixels, respectively. The likelihood of a proposed labeling z, similar to Equation 2.5, depends on the likelihood of the old and new superpixels:

$$\mathcal{L}(z) \overset{\text{C,T}}{=} \beta K_o + \alpha K_n - \sum_k \frac{(N_k - \frac{N}{M})^2}{2\sigma_M^2} \sum_d \mathcal{L}_{s_k}(x_{I_k,d}) \tag{2.6}$$

where $s_k \in \{o, n\}$ and switches between the likelihood of new superpixels ($\mathcal{L}_n$) and the likelihood of old superpixels ($\mathcal{L}_o$). The likelihood of new superpixels ($\mathcal{L}_n$) is the same as in the case of the first frame. However, the likelihood of old superpixels ($\mathcal{L}_o$) is different since the old superpixels evolve through time and have past information. So Equation 2.6 can be broken down to the following basic probability computations for old superpixels. More details in the TSP paper [12].

The appearance and locations of old superpixels change over time. The approach models the color means separately from location means. The current color means are modeled using the previous means:

$$p(\mu_k^{a,t} | \mu_k^{a,t-1}) = \mathcal{N}(\mu_k^{a,t}; \mu_k^{a,t-1}, \delta_a^2 I) \tag{2.7}$$

For modeling the location means of the superpixels, the approach assumes that superpixels that are spatially close and are similar in color move together, assuming that the object is moving smoothly/rigidly. So the location means are modeled with a Gaussian process f^t, using the mean of the previous frame and a bilateral kernel that combines both color and location in the covariance matrix:

$$p(f^t|\mu^{t-1}) = \mathcal{N}(f^t; \mu^{\ell,t-1}, \Sigma(\mu^{t-1})) \tag{2.8}$$

such that

$$\Sigma_{k,j}(\mu^{t-1}) = \prod_d h(\mu^{t-1}_{k,d}, \mu^{t-1}_{j,d}) \tag{2.9}$$

and h is the squared exponential kernel:

$$h(\mu_k, \mu_j) = e^{-\frac{1}{2}|\mu_k - \mu_j|^2} \tag{2.10}$$

The covariance output of h gets closer to one if its inputs are very close, and it decreases as the inputs distance increase.

Using Gaussian process regression, a predication of f^t for old superpixels can be made:

$$f^t = \Sigma(\Sigma + \delta^2_\ell I)^{-1}(\mu^{\ell,t}_o - \mu^{\ell,t-1}_o) + \mu^{\ell,t-1}_o \tag{2.11}$$

It is used to compute the location mean probabilities:

$$p(\mu^{\ell,t}_k|f^t) = \mathcal{N}(\mu^{\ell,t}_o; f^t, \delta^2_\ell I) \tag{2.12}$$

The prior probability of a labeling z is changed to the form:

$$p(z) \propto \hat{\alpha}^{K_n}\hat{\beta}^{K_o} Valid(z) \prod_k \mathcal{N}(N_k; \frac{N}{M}, \sigma^2_M) \tag{2.13}$$

by splitting the geometric distribution on the number of superpixels into two: one for the number of old superpixels and one for the number of new superpixels. Setting $\hat{\alpha}$ and $\hat{\beta}$ differently controls the trade-off between the preference of using old superpixels and creating new superpixels. In addition, an area term is included for controlling the size of the superpixels.

2.1.2.1.2.2 Labeling Initialization and Possible Moves

The initial labeling of a frame is a propagation of the labels of its previous frame through optical flow. From the previous frame's labeling, each pixel gives its label to the pixel at the end of the optical flow vector in the current frame, if it was within the image domain. As a result, a pixel in the current frame can get zero, one, or multiple possible labels. If the pixel gets one label, then it takes this label. If the pixel gets multiple possible labels, it choses the label of the closest superpixel based on distances to means. If the pixel does not get any label, it is labeled with the new superpixel label. After that, the approach enforces connectivity resulting in some rejected pixels. The rejected pixels are then assigned to neighboring superpixels by iteratively looking at their 4-neighbors and flooding their labels to rejected pixels.

Similar to the case of the first frame, a set of movements from a given labeling are proposed. The approach accepts and performs the move that increases the labeling likelihood until there are no more accepted moves, thus it arrived at a local optimum. The three possible movements in the first frame case are also used here, with an additional fourth possible movement: the switch move. In this move, a new superpixel can be relabeled to be linked to a dead superpixel.

2.2 Related Video Segmentation Approaches

In the previous section, low-level superpixel algorithms are reviewed. RAMS and some other video segmentation approaches group superpixels to produce higher level segments.

This section reviews related video segmentation approaches. Some of them use superpixels, while others use different segmentation units as discussed below.

Video segmentation approaches vary in different aspects. The first aspect is the segmentation unit they use, which is the video parts that they cluster. The most common segmentation units are pixels, superpixels, and point trajectories. Clustering pixels or superpixels results in dense segmentations, while clustering point trajectories results in sparse segmentations. A second aspect where approaches vary is the video cues used for segmentation, which includes motion, color, and location. Some use all three, while others use two or one of them. The focus of this review is motion-based approaches, since the goal is motion-based segmentation. The third aspect where approaches vary is the clustering algorithm they use, where the most common are spectral clustering, bottom-up hierarchal clustering, and search and optimization. The related approaches are categorized in the following sections based on their used segmentation unit. Approaches that use the same segmentation unit are more similar than approaches with the same clustering algorithm.

Supervised vs. unsupervised approaches is another variation aspect. Despite the fact that supervised approaches can give high accuracy results, unsupervised approaches are sought after to minimize costs and time. A recent supervised approach [29] propagates a first frame hand-labeling to subsequent frames by tracking image patches. However, the focus of this review is unsupervised approaches, since the proposed approach is unsupervised segmentation.

Videos are 3D space-time volumes (2D spatial and 1D temporal), so another consideration is the order in which the approaches handle the video dimensions. Some approaches spatially segment each frame using motion and/or color first, and then match regions to get a more consistent temporal segmentation. Other approaches track segmentation units through time first, and then segment these trajectories using motion with/without color, such as the approaches that cluster point-trajectories. And then there are approaches that do spatial and temporal segmentation simultaneously, such as the graph-based approaches.

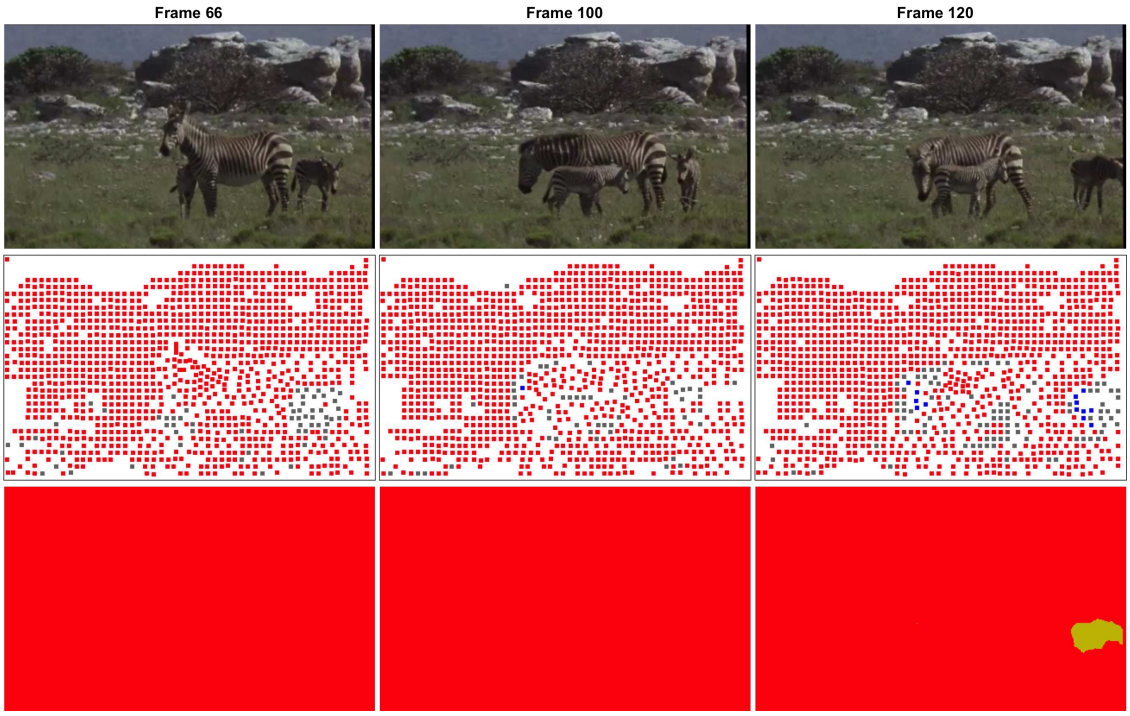


Figure 2.3: Segmentation results for Brox and Malik [32], and Ochs and Brox [25] approaches for the video example shown in Figure 1.2. The first row contains the original frames. The second row contains the point-trajectories segmentation of Brox and Malik [32] approach. The third row contains the corresponding densification of point-trajectories labels by Ochs and Brox [25] approach.

2.2.1 Clustering Point-Trajectories

Motion in videos is a rich source of information for video segmentation. Optical flow is used to densely estimate motion. Optical flow computation is the generation of flow vectors that map each pixel in a frame to a subsequent frame. However, optical flow only gives information about frame-to-frame motion. So many video segmentation approaches resort to point-trajectories for analyzing long-term motion in videos. These approaches segment videos by clustering their point-trajectories. Examples include [13], [15], [24], and [32].

Brox and Malik [32] presented an approach for segmenting point-trajectories using spectral clustering. The pairwise affinities are computed based on the motion and spatial distances between point-trajectories. The motion distance is computed as the maximum distance between the pair of trajectories translational motions at a shared frame among all

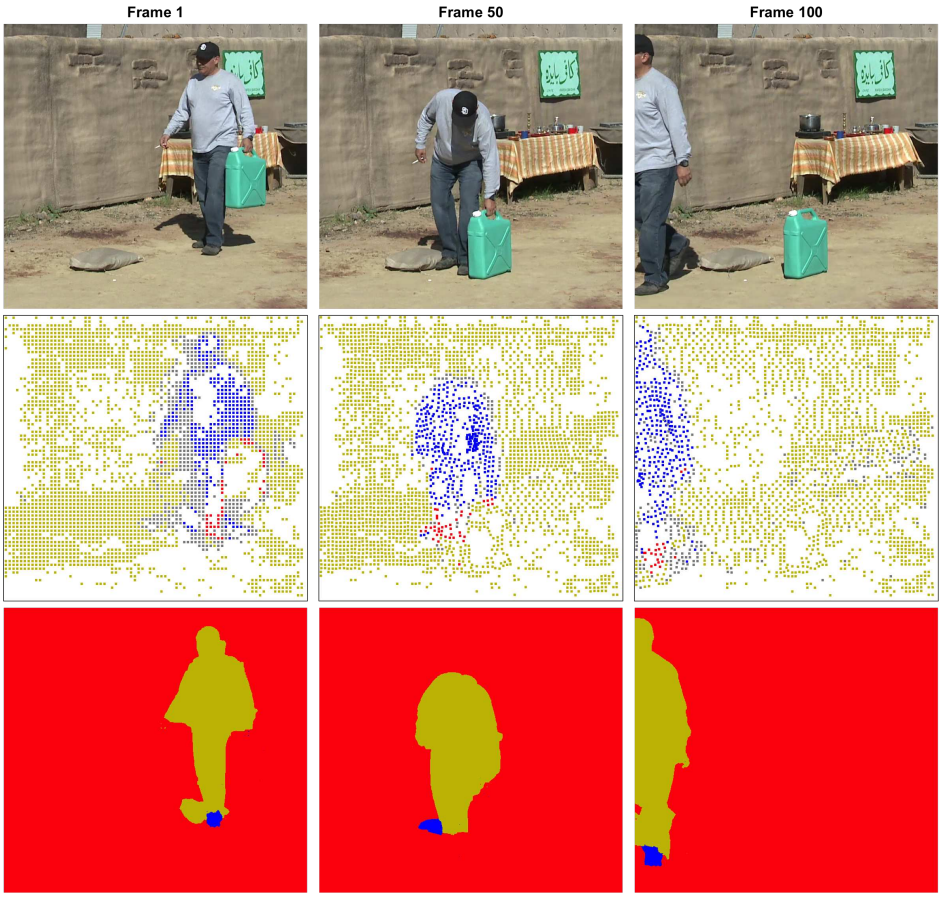


Figure 2.4: Segmentation results for Brox and Malik [32], and Ochs and Brox [25] approaches for the video example shown in Figure 1.3. The first row contains the original frames. The second row contains the point-trajectories segmentation of Brox and Malik [32] approach. The third row contains the corresponding densification of point-trajectories labels by Ochs and Brox [25] approach.

shared frames. Then this motion distance is scaled by the average spatial distance between the pair of trajectories at all shared frames, giving more weight to close pairs. Color is not used in the approach [32]. After clustering the point-trajectories based on these affinities, a post-processing step is performed to merge clusters that have similar motion models. The motion type that this approach [32] can handle is rigid motion only; non-rigid motion is not handled here. That is because they compute the distance between two trajectories as the maximum motion difference. So if one part of the object moves differently than the other part, then they will have higher distance and consequently lower affinity. Figures 2.3, 2.4, and 2.5 show some segmentation results for this approach.

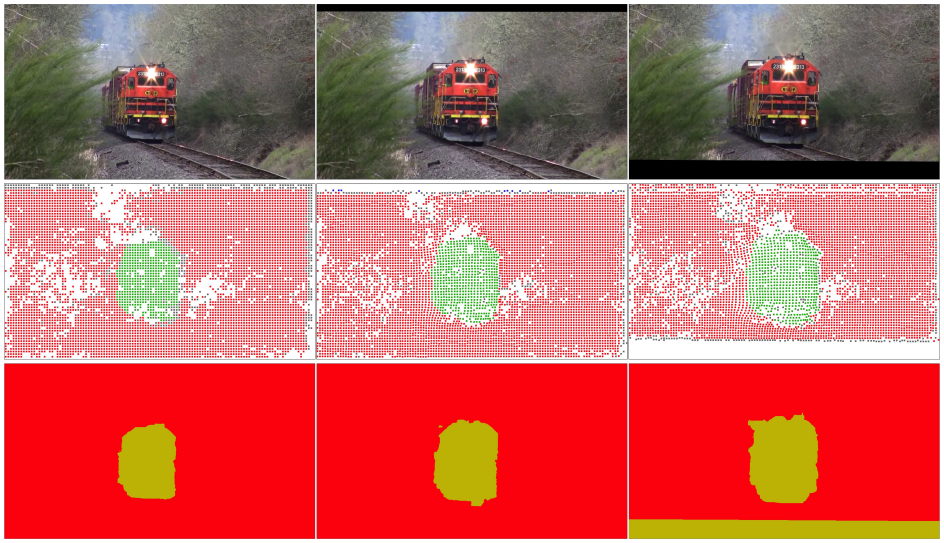


Figure 2.5: Segmentation results for Brox and Malik [32], and Ochs and Brox [25] approaches for the video "freight_train" from VSB100 [9] dataset. Frames 1, 61, and 101 are shown in consecutive columns. The first row contains the original frames. The video is stabilized with [28], and hence the black borders on the frames. The second row contains the point-trajectories segmentation of Brox and Malik [32] approach. The third row contains the corresponding densification of point-trajectories labels by Ochs and Brox [25] approach. The segmentation of the train is successful. However, the train is a rigid object.

The segmentation result of [32] is sparse; it does not cover the whole video. However, this problem was alleviated in [25]. Ochs and Brox [25] extended the point-trajectories clustering of [32] to all the pixels in the video for dense video segmentation, but they do so for each frame independently. First, a labeling of all pixels was computed by minimizing an energy function that is based on conserving the labels of the points along the already-clustered point-trajectories and distributing these labels in condensed regions. The result of this segmentation is then used as the lowest level in a k-level hierarchal clustering, where the levels use hierarchical superpixels and k is a user-set parameter (they set it to 3 experimentally). Then all the levels are jointly optimized to minimize an extended energy function with three terms. The first term objective is still to maintain the labels of the points along the already-clustered point-trajectories in the first level, but now these points are weighted by their distance to the coarsest superpixel boundary since points close to object boundaries have unreliable motion. The second term, as in the original energy function, has the objective of

distributing the labels in condensed regions, but it does that for each level. The third term is the one that ties the levels together by minimizing the difference between the consecutive levels labeling weighted by the color differences between their corresponding superpixels mean color. The weakness of this approach is that this dense segmentation is an interpolation of the sparse segmentation of point-trajectories. So the correctness of final results depends on the correctness of the point-trajectories clustering, and it also uses the same clusters. Figures 2.3, 2.4, and 2.5 show some segmentation results for this approach. Ochs, Malik, and Brox consolidated the approaches [32] and [25] in [23].

Then, Ochs and Brox [24] approach update the point-trajectories clustering of [32] by changing the pairwise affinities to tertiary affinities. This change was adopted to overcome the limitation of comparing two point-trajectories that can only reveal their translational motion similarity. Using three or more point-trajectories, a motion model (translation, rotation, and scaling) can be estimated using two point-trajectories and the other points are fitted to this model so the similarity will be based on the fitting error. Since the relationships are defined for more than two point-trajectories, a hyper-graph is constructed instead of a regular graph, and then this hyper-graph is projected to a regular graph in order to perform spectral clustering. The details of this process are as follows. First, a hyper-graph is constructed where the nodes are the point trajectories and the hyper-edges connect three nodes with weights that equals their motion similarity. The distance between a triplet in a shared frame is based on the fitting error of one of these point-trajectories to a motion model estimated using the other two point-trajectories. Note that this distance varies by the choice of the two point-trajectories used to compute the motion model, so they take the maximum distance among all distances from permuting the point-trajectories. The spatial distance between their points then weights this distance. So the final similarity of a triplet is based on their maximum distance in all shared frames. After constructing the hyper-graph, it is projected to a regular graph using a proposed regularized maximum projection, which basically sets the weight of an edge between two point-trajectories in the regular graph

with the maximum hyper-edge weight among all hyper-edges that include these two point-trajectories weighted by the length of their overlapping time. Since this is a computational expensive process due to the full-connectivity of the hyper-graph, they suggested sampling the hyper-edges considered in a projection. The sampling was a combination of random sampling and k-nearest neighbors sampling. Finally, spectral clustering was performed to obtain the clusters. As in [32], this method also doesn't handle non-rigid motion, and for the same reason.

Fradet, Robert, and Perez [15] presented another approach for sparse video segmentation by clustering point-trajectories. The basis for motion segmentation here is motion model fitting instead of pairwise similarities. A motion model is fitted for each cluster such that it is defined by a series of affine motion models for its lifetime. A trajectory is compared with a cluster's motion model through the use of motion residual, which is the dissimilarity between them computed as the mean geometric distance between the trajectory and its warped version (computed by applying the motion model of the cluster to the trajectory coordinates). So this approach starts by randomly sampling the trajectories set to create groups with minimum size of three trajectories. Then a motion model is fitted for all groups, and a preference set is computed for all trajectories (the groups where the motion residual between the trajectory and the group's motion model is less than a threshold). Based on these preference sets, a bottom-up hierarchical clustering algorithm starts with these sets as initial clusters, and then in each step, it merges the closest pair of clusters, where closeness is based on the Jaccard distance between the clusters. The clustering process stops when all clusters are disjoint sets, and then a motion model is fitted for all clusters and trajectories are assigned to their closest cluster based on the motion residual. The segmentation result is a sparse segmentation that only uses the motion cue of the video, and naturally suitable only for rigid motion.

Dimitriou and Delopoulos [19] also segmented point-trajectories, but operate on overlapping subsequences and then combine results. They first obtain an over-segmentation using spectral clustering with [32] affinities. For each segment, an affine motion is estimated and outliers are removed, and then similar segments are merged. In [20], Dimitriou and Delopoulos compute all possible motion models from the trajectories of a subsequence, when comparing its first and last frames. Then each trajectory is compared with all the possible motion models, resulting in a ranking vector for all models. These ranking vectors are used to compute affinities between trajectories, which are used in spectral clustering to obtain the subsequence segmentation. The segmentations of subsequences are combined, and similar segments are merged as in [19]. As previous approaches, both [19] and [20] segment point-trajectories based on affine motion models. What all these systems have in common is that they use rigid motion models to group points. RAMS uses only the temporal co-occurance of motion, not affine motion models, and therefore is able to group parts of non-rigid objects.

2.2.2 Clustering Pixels and Superpixels

Other approaches cluster pixels or superpixels. In these approaches, local image information such as color or texture supplement motion data. These approaches construct graphs connecting spatially and temporally neighboring nodes and segment the graph. Galasso, Cipolla, and Schiele [10] presented an approach that clusters superpixels of videos using spectral clustering. The authors presented six possible different affinities: (i) short-term-temporal affinity (STT), which basically measures pixel overlap for superpixels in neighboring frames; (ii) long-term-temporal affinity (LTT), which is point-trajectories overlap for the point-trajectories within the superpixels in different frames; (iii) spatio-temporal-appearance affinity (STA), which is based on the difference between the median colors of the superpixels; (iv) spatio-temporal-motion affinity (STM), which is based on the difference between the superpixels median optical flows; (v) across-boundary-appearance affinity (ABA), which is based on the common boundary of neighboring superpixels; and (vi) across-boundary-motion

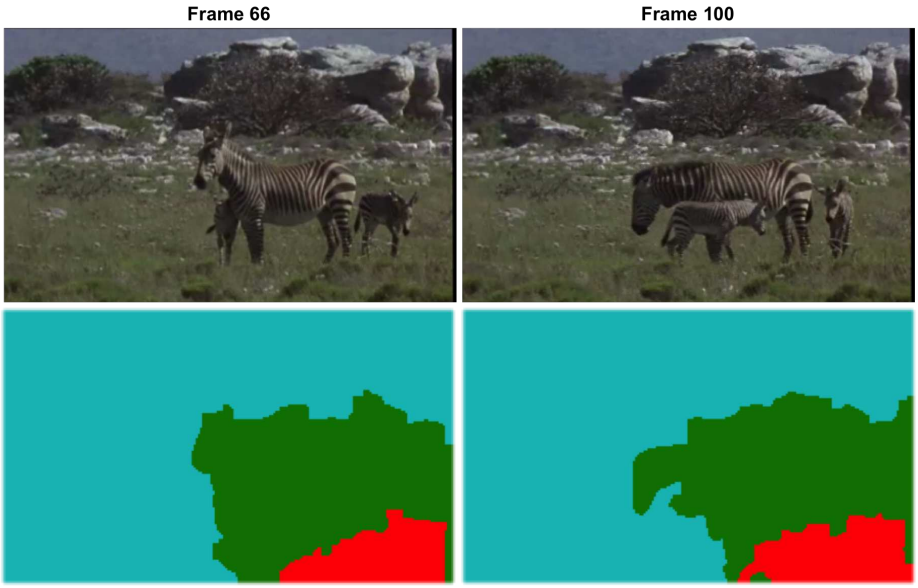


Figure 2.6: Segmentation result for Galasso et al. [10] approach for the video example shown in Figure 1.2. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The frames were resized by half for the approach to be able to process the video. The small zebras were not visually separable in any level, so the level where the big zebra was first visually separable from the background is the level presented in this figure.

affinity (ABM), which is based on the motion in the common boundary of the superpixels. The authors analyzed these six terms experimentally on a dataset and decided based on their segmentation performance that the minimal set is (STT+LTT+STM+STA).

The approach in [10] gives dense segmentation results, and incorporates multiple video cues. However, the authors reported that it fails when there is little to no motion in the video. In addition, this approach partially handles non-rigid motion on the boundaries only when the ABM affinity is used, but the authors did not include this term in the final minimal set. Figures 2.6 and 2.7 show some segmentation results for this approach.

Building on [10], Galasso, Keuper, Brox, and Schiele presented another approach in [8]. In [10], their approach uses superpixels of the first finest segmentation level in the image segmentation hierarchy produced by an extension of [21]. In [8], they use superpixels of the second coarser segmentation level in the image segmentation hierarchy produced by [21]. These superpixels are larger with uneven sizes. So they re-weighted their pairwise affinities

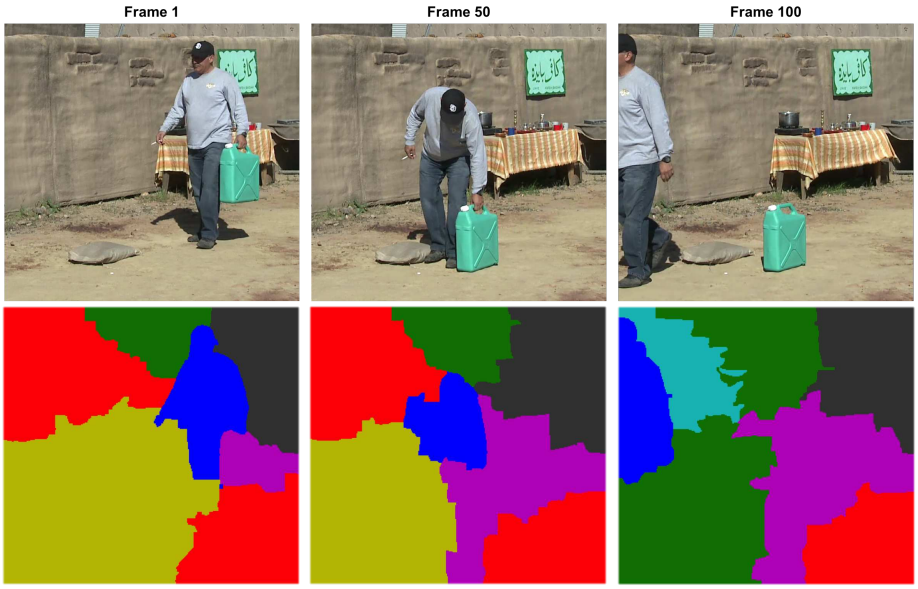


Figure 2.7: Segmentation result for Galasso et al. [10] approach for the video example shown in Figure 1.3. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The frames were resized by half for the approach to be able to process the video. The level in the segmentation hierarchy with the lowest number of clusters and still separates the man, the gas can, and the background, is the level presented in this figure.

by scaling them with the product of their corresponding superpixel sizes. In addition, they presented an algorithm for streaming videos, in which they update the superpixels graph as frames become available and defer merges if they had low certainty.

A graph-based approach is presented by Levinshtein, Sminchisescu, and Dickinson [1] to segment superpixels of a video, where edges connect spatially and temporally neighboring superpixels with weights based on either motion or color similarity. The color similarity is based on the histograms of the superpixel pair, while the motion similarity is based on the difference between their flow vectors. If two neighboring superpixels are in different frames, then their affinity is their color similarity only. Otherwise, if two neighboring superpixels are in the same frame, the affinity is either their color or motion similarity, whichever is smallest. Superpixels areas product is used to scale the final affinity, giving more weight to larger superpixels. However, the result is multiple solutions, where each is a two-cluster solution. Another reported limitation is the failure to segment small objects correctly.

The approach by Huang, Liu, and Metaxas [34] is a hyper-graph-based video segmentation. The nodes are image regions that are described with motion magnitudes and directions that were obtained by averaging over all the region's pixels motion. In addition, each image region is also described with a motion profile, which can be described as the probabilities of this region corresponding to a set of neighboring regions in the next frame based on color similarity. To create the hyper-edges, they first create two regular graphs and compute their first k eigenvectors using spectral analysis. These two graphs connect pairs of nodes that are spatial and temporal neighbors (8-connected). The first graph affinities are based on differences between the pairs' motion, while the second graph affinities are based on differences between their motion profiles. For each obtained eigenvector, they compute a two-way cut resulting in binary images. Then for each two-way cut, two hyper-edges are constructed in the hyper-graph where each of these hyper-edges connects the regions that lie in the same part of the cut. The weight of this hyper-edge is based on the motion difference between the two parts. Finally, a hyper-graph cut is computed based on hyper-edges weight and volume.

Grundmann, Kwatra, Han, and Essa [16] presented a graph-based approach. Their approach starts with a pixel-level graph connecting each pixel with its 26 spatial-temporal neighbors and performs an over-segmentation to get the regions. Then it performs a bottom-up hierarchical clustering of regions using region graphs, where regions are connected if they are incident, and iteratively merging regions based on the difference between their color histograms. They first proposed an approach that is based on color only, but then they proposed another one enhanced with optical flow as follows. Instead of connecting a pixel with its immediate neighbors in the previous and next frames in the first stage of processing, the pixel is connected with its neighbors along the optical flow. In addition to the color histogram descriptor, a per-frame flow histogram is computed for each region. This gives distances between regions that are based on both color and flow histogram distances. The addition of the motion cue improved the segmentation results significantly. Figures 2.8 and 2.9 show some segmentation results for this approach.

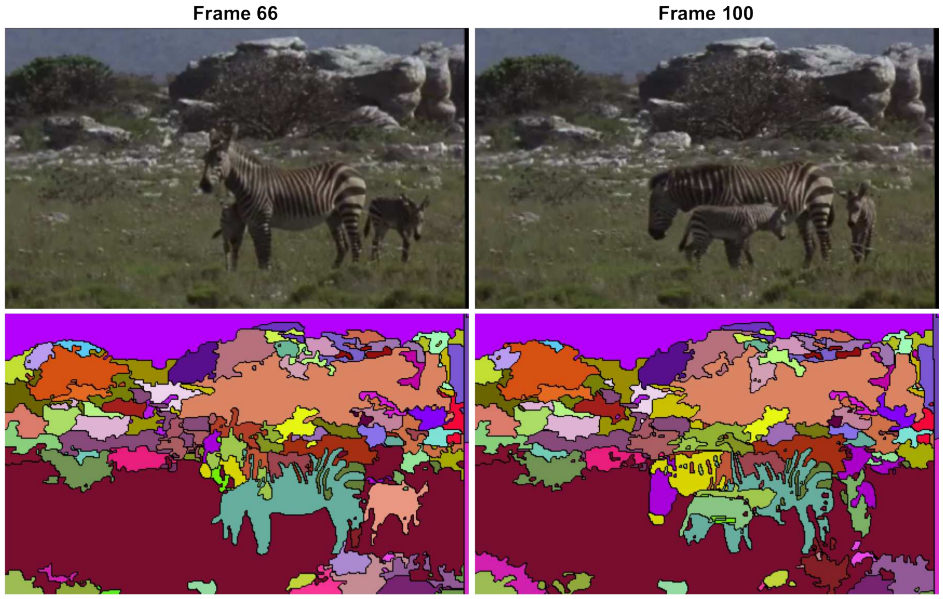


Figure 2.8: Segmentation result for Grundmann et al. [16] approach for the video example shown in Figure 1.2. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The level in the segmentation hierarchy with the lowest number of clusters and the zebras are visually separable, is the level presented in this figure.

Extending the previous approach of [16], and also building on Brox and Malik [32] point tracking and clustering, Lezama, Alahari, Sivic, and Laptev [13] presented an approach that starts with clustering point-trajectories, then extends that clustering to all the pixels of the video. In this approach, the sparse point-trajectories of the video get clustered by optimizing a labeling cost function with three terms. The first term penalizes for the between-cluster similarities encouraging similar tracks to have the same label. The second term penalizes for weak within-cluster similarities encouraging dissimilar tracks to have different labels. The similarity here is based on both spatial and velocity distances. By representing the relative depth of clusters in the clusters labels (i.e. $1 < 2$ means that cluster 2 occludes cluster 1), the third term enforces the order of labels by penalizing with an occlusion score for any track that is occluding another track but gets an equal or smaller label than that track. The occlusion score is based on the spatial and velocity distances between the two tracks at the time endpoints of the first track. Large spatial distance decreases the occlusion score, while

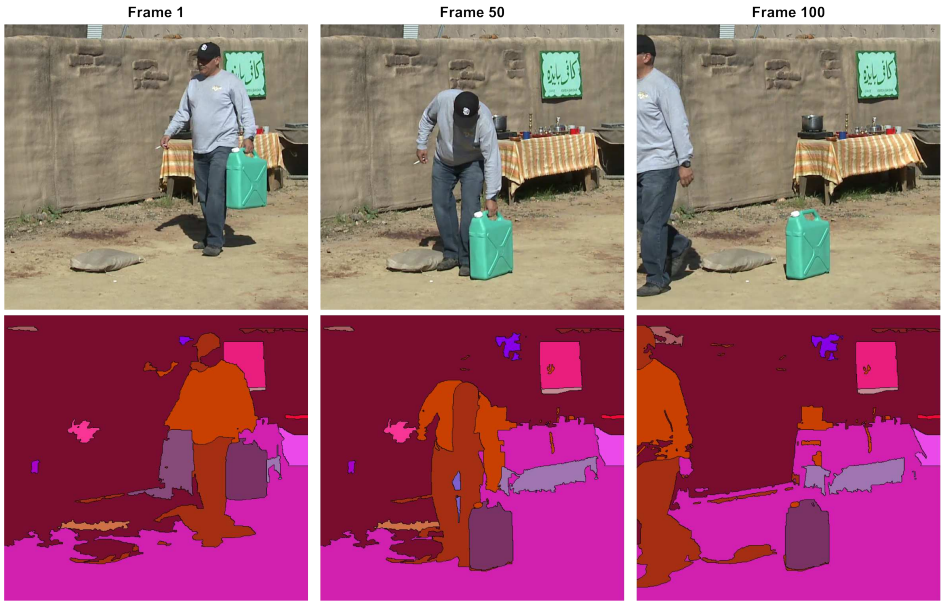


Figure 2.9: Segmentation result for Grundmann et al. [16] approach for the video example shown in Figure 1.3. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The level in the segmentation hierarchy with the lowest number of clusters and still separates the man, the gas can, and the background, is the level presented in this figure.

large motion distance increases the occlusion score. Note that this labeling cost function is only for all pairs that have some time overlap, and color information was not used here.

After that, the obtained result, which is a clustering of a set of point-trajectories, is incorporated into the previous graph-based segmentation [16] to segment all the video's pixels. The first difference between this approach and the approach [16] is in the first processing step. Instead of connecting a pixel with its 9 temporal neighbors along the optical flow in one time direction in the approach [16], the pixel is connected in this approach with only one temporal neighbor along the optical flow, and the velocity distances is also included in the weight of edges between the pixels in addition to the color distances. Once the graph has been constructed, the point-trajectories are incorporated in the graph such that the weight of the edges that lie along these trajectories is set to zero making them perform as seeds in the segmentation process. Also, the labels of the point-trajectories are incorporated here such that there is an additional pixel labeling in a way that pixels that lie along these

trajectories are labeled with the same point-trajectory label that they lie within, while the remaining pixels have -1 labels. Finally, in the iterative merge process, a merge cannot be done if the considered regions have different additional labels unless one of them has the -1 label. This enforces the sparse initial clustering results into the final dense segmentation. However, since it is an extension to [16], the results are also over-segmentations. Another limitation is the assumption of constant relative ordering. Even though it is not true in all cases, the authors maintain that it is a reasonable assumption. They also reported that errors in the initial sparse clustering will get propagated to the final dense clustering.

Similarly to [13], Silva and Scharcanski [14] adopted a video segmentation approach that also starts with point-trajectories clustering and then extends it to all pixels yielding a dense segmentation, but they approached this goal differently. The points along the point-trajectories are clustered in independent frame-level clusterings then integrated in an all-video-level using ensemble clustering. Finally, the point clustering is extended to pixels to achieve a dense video segmentation.

The details of the clustering process are as follows: In the frame-level clustering, three clusterings are performed for each frame comparing it with the frames that are one, two, and three frames apart by clustering the points within them based on their displacement vector. Then all these clustering are integrated to get one clustering of all the points using a meta-clustering algorithm in five steps. First, clusters are mapped to a hyper-graph in which the nodes are the points and there is a hyper-edge for each cluster connecting the nodes within it. Second, a similarity matrix is constructed for these hyper-edges where the similarity is computed as the Jaccard distance between each pair. Third, this similarity is used to hierarchically cluster these hyper-edges in meta-clusters (clusters of clusters). The level in this tree (i.e. the number of clusters) is chosen automatically such that it corresponds to the longest range of varying the threshold while the number of clusters remained constant. Fourth, for each meta-cluster, all its hyper-edges are collapsed into one meta-hyper-edge with a membership score for each point based on the percentage of hyper-edges in this

meta-cluster that include this point. Finally, each point is assigned to the meta-cluster with highest membership score.

Once this final point clustering is obtained, it gets validated based on the motion and spatial differences between the points and the meta-clusters, and re-labeling of points is performed where necessary. Then the results are spatially filtered based on connected components of Delauny triangulation. Finally, these results are extended to the video pixels by assigning each pixel to the meta-cluster of the most similar point based on motion and spatial distance. The entire segmentation process is based on motion and spatial distance without consideration of color.

Xu, Xiong, and Corso [7] presented the first segmentation approach for streaming videos. It is a graph-based hierarchical segmentation approach. It goes forward in time such that it uses previous segmentations for a current segmentation problem, but does not change these previous segmentations. It cuts the video into a set of non-overlapping chunks, and the computation of a chunk's segmentation (v_i) depends on its immediate predecessor video chunk segmentation (v_{i-1}) by processing the union of these two video chunks. They add an additional criteria for the merging process in the graph-based hierarchical segmentation of v_i, such that if two regions in the union and both include some regions from v_{i-1}, they cannot me merged if these included regions were not merged in the segmentation of v_{i-1} at a higher layer. For the graph-based hierarchical segmentation itself, they extended the graph-based image segmentation algorithm [22] to videos, similarly to [16]. However, they describe each region with a Lab color histogram and use histogram distances to compute edge weights. So they only use the color information with no motion information. This approach is included here only for evaluation purposes, since it was included in the comparison of [9]. Figures 2.10 and 2.11 show some segmentation results for this approach.

Tripathi, Hwang, Belongie, and Nguyen [30] improved [7] by utilizing motion. As in [16], they combine per frame flow histograms with color histograms. They use this supervoxel segmentation as a basis for motion layer segmentation, where they hierarchically merge

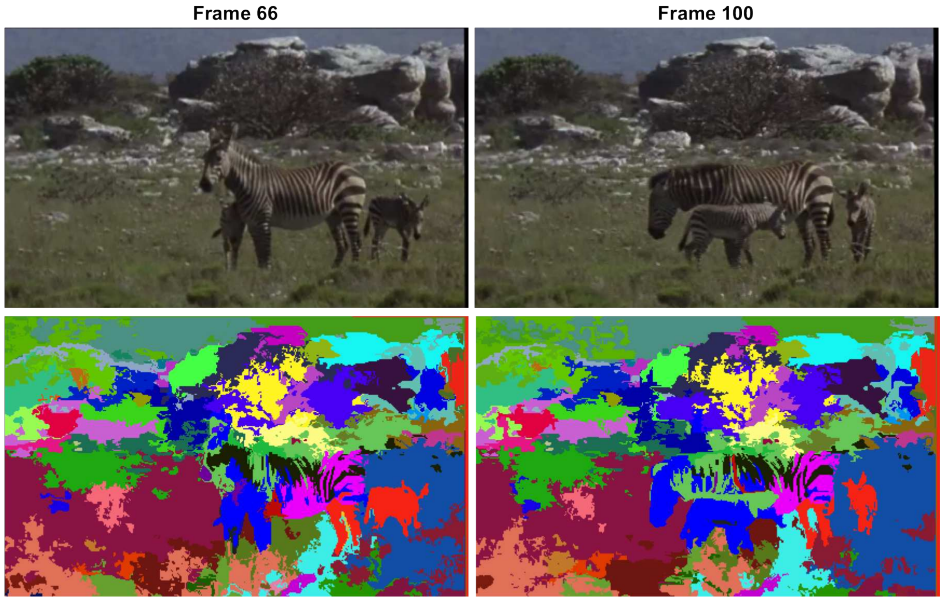


Figure 2.10: Segmentation result for Xu et al. [7] approach for the video example shown in Figure 1.2. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The level in the segmentation hierarchy with the lowest number of clusters and the zebras are visually separable, is the level presented in this figure.

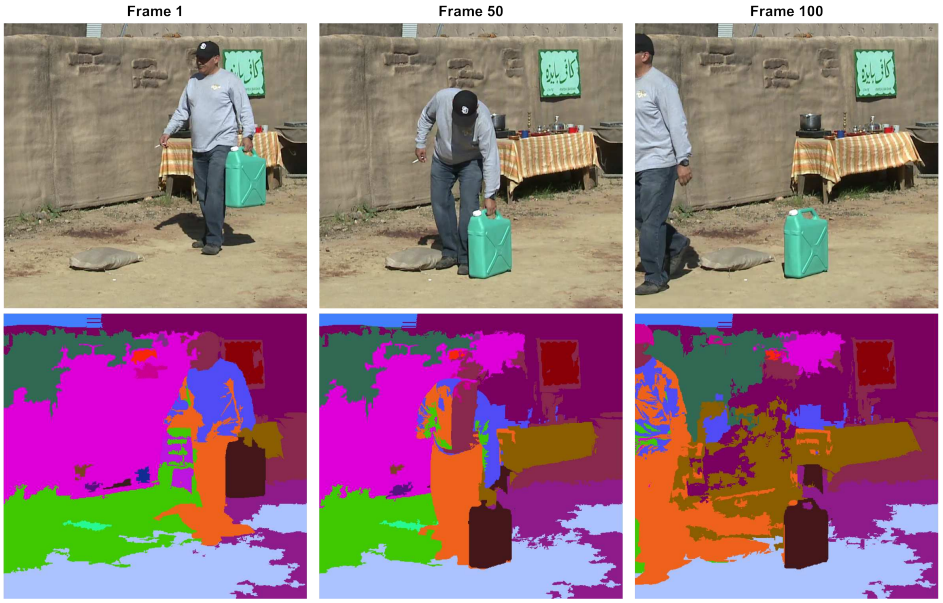


Figure 2.11: Segmentation result for Xu et al. [7] approach for the video example shown in Figure 1.3. The first row contains the original frames, while the second row contains the segmentation result shown for the corresponding frames. The level in the segmentation hierarchy with the lowest number of clusters and still separates the man, the gas can, and the background, is the level presented in this figure.

segments even further in a streaming manner too. Affine motion parameters are estimated for each segment, then neighboring segments are merged greedily based on their motion differences. Setyanto, Wood, and Ghanbary [2] also use a hierarchal video segmentation similar to [16], but it describes each region with a mean color in gray levels, and a temporal direction based on the motion of its centroid. Li, Lin, Zou, Yan, and Tang [6] presented a streaming approach that clusters a fine supervoxel segmentation of [16]. In addition to color, texture, and image edges, their motion features are histograms of optical flow. Collectively, these systems all supplement affine motion models with color and adjacency information, but the underlying motion models still assume rigid objects.

Another approach that segments pixels is by Taylor, Karasev, and Soatto [4] and it uses occlusion for segmentation. It processes one frame at a time, using its previous and next frames for occlusion cues, to create layers with depth orders. A depth order of zero is for the background layer, while higher depth orders are for foreground layers. The higher the value, the closer it is to the viewer. Objects are then obtained by determining the connected components of the foreground layers. Occlusions occur in regions where the forward and backward optical flow differ, so the approach determines the occluder/occluded relationships in order to determine depth order for the pixels in these regions. The depth order labeling is then optimized for all pixels using image and motion boundaries. The approach goes forward in time, but history of previous frames is used, especially when there is no motion to produce occlusion cues. The limitation of this approach occur when there is self occlusion, such as for example when a person moves his/her hand in front of his/her body. Since the hand occludes the body, these parts are placed on different depth layers, and thus ultimately segmented differently. This can break a non-rigid object into more than one segment.

RAMS differs from all the approaches above in that it does not use motion magnitude nor direction as a basis for motion comparison, thus dropping the rigid object assumption. RAMS clusters superpixel trajectories and compare their motion in terms of temporal co-occurrence. Another difference is the use of superpixel trajectories as segmentation units,

an approach that benefits from both the long-range motion information of point-trajectories and the large support areas of superpixels. RAMS builds on the work of Chang, Wei, and Fisher III [12] as it provides the superpixel-trajectories that RAMS clusters. By visually comparing results of RAMS with state-of-the-art approaches results, shown in Figure 1.4, RAMS significantly improves segmentation results.

Chapter 3

Robust Animate Motion Segmentation (RAMS)

The proposed approach (RAMS) is an unsupervised motion-based video segmentation algorithm. Its segments moving objects in videos based on motion, color, and location. Objects can move either rigidly or non-rigidly. A non-rigid object, for example the zebra shown in Figure 1.2, has parts that can move in different directions. Traditional rigid motion methods use motion magnitude and direction for segmentation, thus they are not suitable for segmenting non-rigid objects. The proposed approach does not assume rigid objects and uses temporal coherence of motion for segmentation. Even though the parts of a non-rigid object move independently in different directions, they are hypothesized to move at the same time. Using motion timing instead of motion magnitude and direction is proposed for segmenting moving objects.

As discussed in Chapter 2, image over-segmentations, called "superpixels", can create low-level segmentation units that can be clustered for obtaining video segmentations. Chang, Wei, and Fisher III [12] presented an approach for creating superpixel-trajectories of a video. By obtaining the superpixel trajectories as a preprocessing step, the proposed approach casts the video segmentation problem as the problem of clustering superpixel trajectories. This preprocessing step is not a contribution of this work, and RAMS is not limited to [12] in principle. However, [12] provides good superpixel trajectories and it is used in the experiments of the proposed approach.

Given superpixel trajectories, RAMS computes an affinity matrix for the trajectories in order to cluster them. Every pair of superpixel trajectories has an entry in the affinity matrix

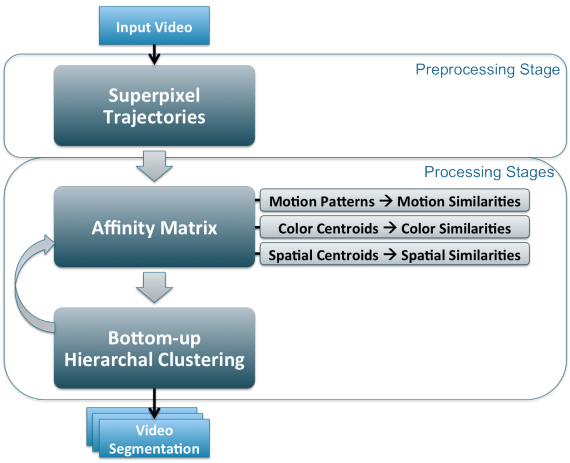


Figure 3.1: The basic program flow of RAMS. It starts with the input video frames, stabilized if needed, and obtains its superpixel trajectories using [12]. The superpixels are reduced in size from the boundaries, and some trajectories are discarded according to criteria discussed in the text. Then three descriptors are computed for each trajectory, and descriptors are used to compute the initial pairwise similarities. Finally, an iterative bottom-up hierarchical clustering is performed to cluster the superpixel trajectories.

that measures how similar they are. The affinity function for a pair of superpixel trajectories is based on temporal coherence of motion, color similarity, and spatial distance. Using these affinities, the superpixel trajectories are clustered to obtain the video segmentation.

Most other related video segmentation approaches cluster selected segmentation units through spectral clustering. Although spectral clustering can produce good results when segments are evenly sized, as pointed out in [3], usually this is not the case. That is because the goal of motion-based video segmentation is to segment moving objects, while combining all other non-moving objects into a single background segment. As a result, the background segment is usually much larger than the other segments. Spectral clustering tends to break this large background segment into multiple smaller pieces. To avoid this problem, RAMS uses bottom-up hierarchal clustering. The steps of RAMS are described in more details in the following sections. Figure 3.1 shows the basic flow of RAMS.

To compute temporal coherence of motion, motion must be determined for each superpixel trajectory to facilitate comparison. For simplicity, RAMS assumes that the background

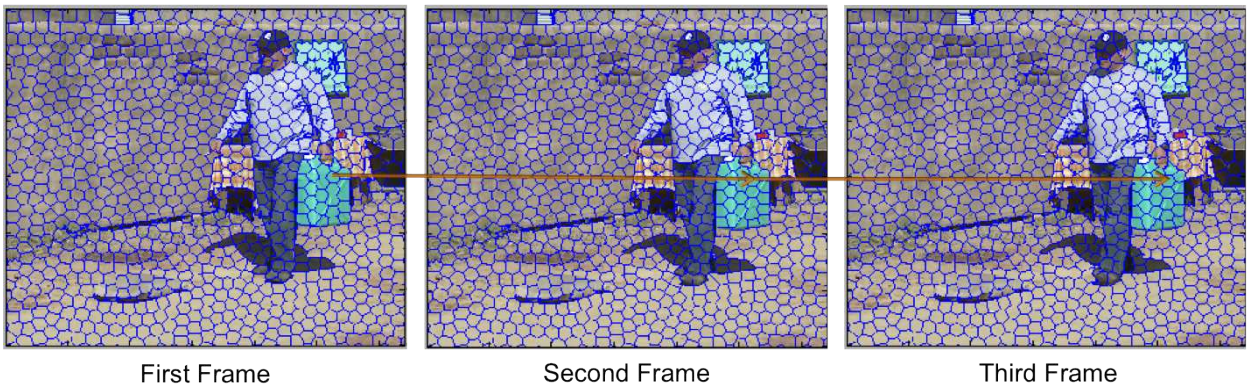


Figure 3.2: Superpixel segmentation of the video example shown in Figure 1.3, obtained using [12]. One of the superpixels trajectories is shown using an arrow between its corresponding superpixels in the first three frames.

is still. If a video has some jitter, it can be stabilized as a pre-processing step, and then RAMS proceeds with the stabilized frames. We use the video stabilization algorithm of Dutta et al. [28] for stabilizing the videos. However, videos with significant camera movement are not used. While it is possible to determine camera motion and thus the independent motions, this not done here since the focus of this work is on exploring the usability of motion temporal coherence in video segmentation instead of low-level motion detection. Extending RAMS to segment videos from cameras in motion is future work, and is discussed in more details in Chapter 5.

3.1 Superpixel Trajectories

The first step is to obtain a superpixel segmentation of the video. This is an over-segmentation of the video frames such that each superpixel is a small set of homogeneous pixels. These superpixels are tracked through time to obtain their trajectories. Note that these trajectories can have different lifetimes, i.e. they can start and end at different frames. Figure 3.2 demonstrates an example.

Superpixels were chosen over pixels or points for their larger support area in the image frames, which gives them more robust motion estimation and color information. Trajectories

are used to get the long-range motion information that is required for analyzing the objects motion in videos.

Chang, Wei, and Fisher III [12] presented an approach that generates temporally consistent superpixels from videos. We chose this algorithm for the temporal consistency of the superpixels and the length of their trajectories. Trajectories are the segmentation unit of RAMS, and their length is crucial for measuring temporal coherence. For example, if a long trajectory was broken into smaller pieces, the motion information from the smaller pieces is not complete. Consider a trajectory on the gas can in the video shown in Figure 1.3 for example. Consider having the trajectory in full video length as opposed to two separate parts. If the trajectory was long, then it can be inferred that it was moving in the first half of the video before it was placed on the ground in the middle of the video. Thus it has a motion pattern that is different from the motion pattern of the still background, so it can be segmented differently. On the other hand, if the trajectory was broken into two parts, the second trajectory will have a motion pattern that is similar to the background. As a result, the first trajectory may be clustered with the man segment, while the second trajectory might be clustered with the background segment. So longer trajectories have more complete motion information for measuring temporal coherence.

As a practical matter, optical flow is unreliable on the boundaries of moving objects due to occlusion; some regions may disappear, while new regions may appear. Since superpixel boundaries can lie on object boundaries, and are sometimes inaccurate, boundaries of superpixels may contain some pixels of another object or may have unreliable motion information. To avoid false motion along superpixels boundaries, each superpixel is reduced four pixels from the boundaries. These discarded pixels are not used in the segmentation algorithm. However, they are assigned the same label as their original superpixels during evaluation. In addition, trajectories that are less than three frames long, or have superpixels with less than a threshold number of pixels, are discarded in the segmentation process. Trajectories in the second case can be trimmed and used if the small superpixels are on either ends of

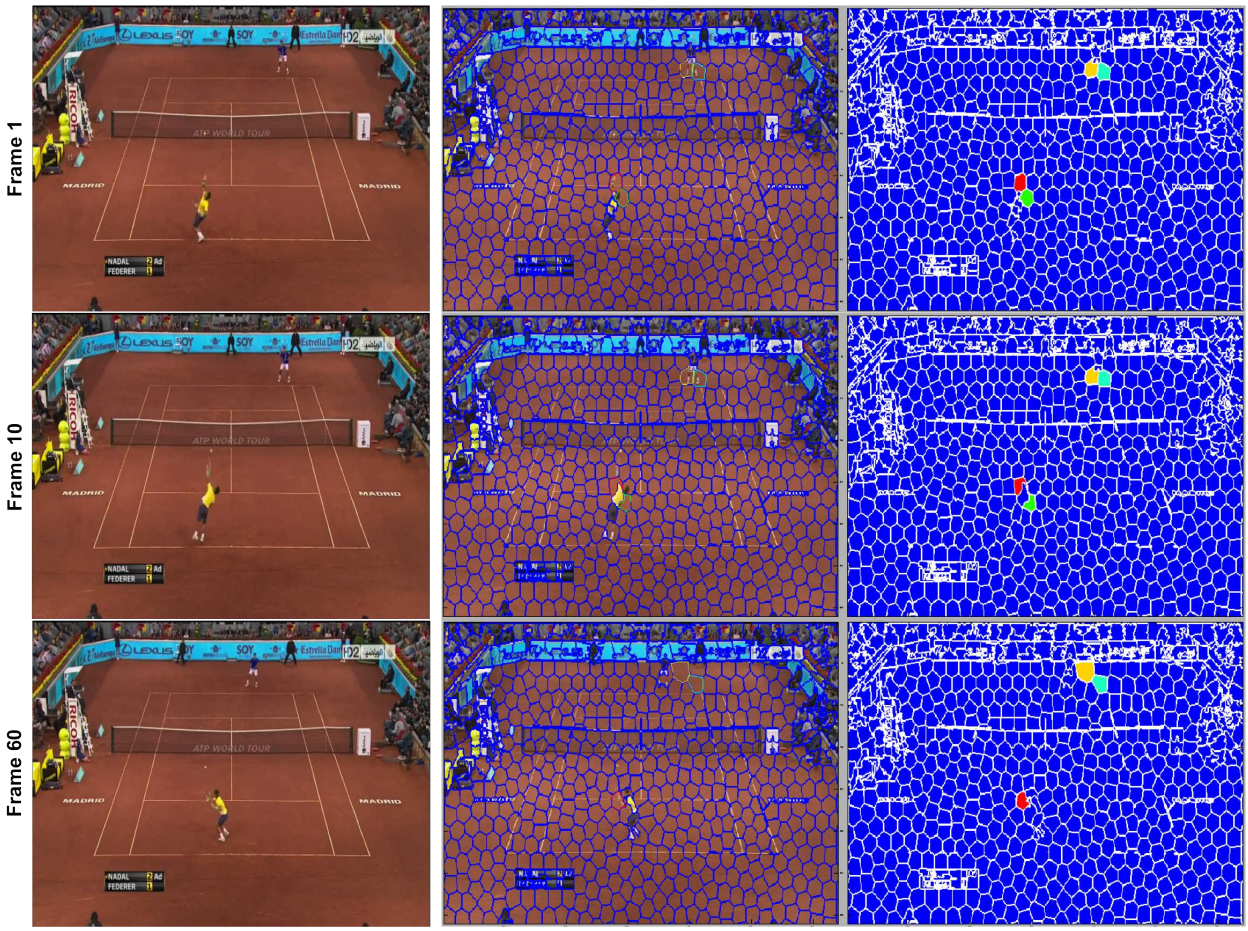


Figure 3.3: Superpixel segmentation of the "tennis_tr" video from the VSB100 dataset [9], obtained using [12] with the number of superpixels per frame parameter set to 500, is shown for the frames 1, 10, and 60. This video is 121-frames long, and resized by half to 640×360 pixels per frame. The first column is the original frames. The second and third columns are the superpixel segmentations where the first one is overlayed on the original image. Four superpixels were selected in colors: red, green, yellow, and aqua blue, to highlight the superpixels that have inaccurate boundaries. As seen in the first frame, they combine the two persons' limbs (legs, hand, and head) with some background regions. This can lead to inaccurate motion and color information. In addition, they cause tracking problems. These colored superpixels tracked to subsequent frames by the TSP algorithm [12], shown in frames 10 and 60, have shifted to background regions. These wrong tracks will probably have wrong motion patterns, affecting the segmentation result.

the track. The threshold for the number of pixels per superpixel is set to 10 in the experiments. Discarded superpixels are later merged with segments in a post-processing step for evaluation purposes, by assigning a discarded trajectory the same label as its most similar neighbor.

The temporal superpixels algorithm [12] takes the approximate number of superpixels per frame as a parameter. This parameter affects the superpixel accuracy depending on the video's dimensions and level of details. There is a trade-off between superpixel size and boundary accuracy. Large superpixels can include multiple visual regions, which affects motion estimation and color homogeneity. Smaller superpixels respect boundaries more, thus have better motion estimation and more homogeneous colors, and track for longer time. This segmentation approach aims at using fairly small superpixels for these reasons. But not so small that they loose their meaning as image regions and have weak and unreliable motion information. Figure 3.3 shows an example of inaccurate superpixels. The value of this parameter (approximate number of superpixels per frame) was set to 500 in the presented experiments. The video of the man and gas can is an exception, as this parameter was set to 800 because of its large size.

3.2 Clustering Superpixel Trajectories

RAMS uses bottom-up hierarchal clustering to group superpixel trajectories. A superpixel trajectory is a superpixel that is tracked through time, resulting in a sequence of superpixels from the first frame it appears in to the last frame where it exists. A superpixel trajectory t_i can be expressed as:

$t_i = \langle SP_{i,startFrame_{t_i}}, SP_{i,startFrame_{t_i}+1}, \ldots, SP_{i,endFrame_{t_i}-1}, SP_{i,endFrame_{t_i}} \rangle$,

where SP is short for superpixel, and $[startFrame_{t_i}, endFrame_{t_i}]$ is the time range over which the trajectory t_i exist. Clustering begins with each trajectory in its own cluster, forming the lowest level in the hierarchy, and then iteratively merges clusters greedily until only two clusters are left. The initial affinity between a pair of trajectories is a linear combination of all three similarities: motion similarity (ms), color similarity (cs), and spatial similarity (ss), as follows:

$$a_{t_i,t_j} = \alpha_1 ms_{t_i,t_j} + \alpha_2 cs_{t_i,t_j} + \alpha_3 ss_{t_i,t_j} \tag{3.1}$$

where the weights α_1, α_2, and α_3 are set proportional to the importance of each video cue, such that they all add to one.

For motion similarities, motion temporal coherence between a pair of trajectories is used. It looks at *when* superpixel trajectories move or don't move together. For example, if both are moving or not moving, then they are more likely to belong to the same object. On the other hand, if one is moving while the other is not, then they are more likely to belong to different objects. So each trajectory t_i is described with a temporal motion pattern (TMP_i): a sequence of motion probabilities for each of its compositing superpixels. A superpixel's motion probability ($mp_{i,frame}$) is computed as the fraction of its pixels that are moving. A pixel (p) is considered moving if its corresponding pixel in the subsequent frame ($next(p)$), as determined by the forward optical flow, lies within the subsequent superpixel of the trajectory, and the pixel's motion magnitude ($m(p, next(p))$) is above a threshold (mt). Notice that the motion pattern for a trajectory is always one frame shorter than the length of the trajectory since there is no motion information for the last frame. So for a trajectory t_i, its temporal motion pattern can be expressed as:

$$TMP_i = \langle mp_{i,startFrame_{t_i}}, mp_{i,startFrame_{t_i}+1}, \ldots, mp_{i,endFrame_{t_i}-1} \rangle$$

Each motion probability can be given as:

$$mp_{i,frame} = \frac{\text{Number of Moving Pixels}}{\text{Number of Considered Pixels}} ,$$

for superpixel $SP_{i,frame}$, where

$$\text{Number of Considered Pixels} = |\{p \in SP_{i,frame} | next(p) \in SP_{i,frame+1}\}| \text{ , and}$$

$$\text{Number of Moving Pixels} = |\{p \in SP_{i,frame} | next(p) \in SP_{i,frame+1} \wedge m(p, next(p)) \geq mt\}|.$$

The motion similarity (ms) between a pair of trajectories t_i and t_j is based on the city-block difference between their temporal motion patterns in the time they overlap (i.e. in the shared frames SF). Motion patterns begin and end in different times, so a set of shared frames where they co-existed must be determined for comparison. Motion similarity can be

given as:

$$ms_{t_i,t_j} = \begin{cases} 0 & \text{if } |SF| = 0, \\ 1 - \frac{\sum_{f \in SF} |mp_{i,f} - mp_{j,f}|}{|SF|} & \text{otherwise.} \end{cases} \tag{3.2}$$

It is worth mentioning that RAMS looks at motion patterns for the whole video at once. Motion patterns do not need to be exactly the same to have high similarities. A part that is different in a fraction of time can still have high similarities with the other parts of the object, compared to the background or other objects motions patterns. As an example, the big zebra in the zebras video [18] stops some time while moving its head independently, and yet the head is segmented with the rest of the body correctly by RAMS (the red region in the second row of Figure 1.4). So RAMS works well on zebras and other non-rigid objects because statistically over time motions of parts tend to occur together.

In addition to motion, RAMS also uses color and spatial distances to describe the affinities between trajectories. In regards to color, Lab color space was chosen for its perceptual uniformity. In other words, the perceptual difference between two colors is proportional to the Euclidean distance between the two colors' points in this space. This is also why it is used by other approaches, e.g. [12, 27], as described in Chapter 2. In RAMS, each trajectory is represented by a single 3-tuple Lab color point that is the mean of all pixels lying within the trajectory, assuming that the pixels are homogeneous. This assumption may not always hold, depending on the accuracy of the underlying superpixel segmentation. To reduce the effect of misclassified pixels and outliers, the centroid is computed by taking the mean in all the three color components. The color similarity (cs) between two trajectories is based on the Euclidean distance between their color centroids (cc) as follows:

$$cs_{t_i,t_j} = 1 - \frac{euclidean(cc_{t_i}, cc_{t_j})}{maxColorDist} \tag{3.3}$$

where $maxColorDist$ is the maximum possible distance based on the video's colors.

The spatial distance between any two trajectories is also incorporated in their pairwise affinity. Each trajectory is represented with a sequence of spatial centroids (sc) for each of its superpixels during its lifetime. The spatial distance between two trajectories is the maximum distance between their centroids in their overlapping time. The spatial similarity (ss) can be given as follows:

$$ss_{t_i,t_j} = \begin{cases} 0 & \text{if } |SF| = 0, \\ 1 - \frac{\max_{f \in SF} euclidean(sc_{i,f}, sc_{j,f})}{maxSpatialDist} & \text{otherwise,} \end{cases} \tag{3.4}$$

where $maxSpatialDist$ is the maximum distance based on the video's dimensions.

As clustering progresses, similarities are measured between clusters instead of individual trajectories. After merging superpixel trajectories in a cluster (C_k), the new cluster's temporal motion pattern is computed by combining the patterns of its compositing trajectories together. This is done by taking the unweighted average of their motion probabilities:

$$mp_{k,f} = \frac{\sum_{t_i \in \{t_i | t_i \in C_k \wedge f \in [startFrame_{t_i}, endFrame_{t_i} - 1]\}} mp_{i,f}}{|\{t_i | t_i \in C_k \wedge f \in [startFrame_{t_i}, endFrame_{t_i} - 1]\}|} \tag{3.5}$$

Using this combined temporal motion pattern TMP_k for the new cluster C_k, the motion similarity between it and the other clusters can be computed using the same equation of 3.2.

For the spatial centroids of the new cluster C_k, they are computed as the centroids of the trajectories' centroids:

$$sc_{k,f} = centroid(\{sc_{i,f} | t_i \in C_k \wedge f \in [startFrame_{t_i}, endFrame_{t_i}]\}) \tag{3.6}$$

where $centroid(points)$ is the centroid for the set of given points. Then, the spatial similarity between the cluster C_k and the other clusters can be computed using the same equation of 3.4.

The color pairwise similarity definition is slightly changed here to accommodate clusters of trajectories, so there is no need for a new color descriptor for each cluster. The reason for that is that it may not be appropriate to have a single Lab color point to represent a cluster, since the cluster may represent a multi-color object. So instead, the color similarity between two clusters C_k and C_r is the average of the color similarities, defined in Equation 3.3, for the trajectories of the two clusters as follows:

$$cs_{k,r} = \frac{\sum_{t_i \in C_k, t_j \in C_r} cs_{i,j}}{|C_k||C_r|} \tag{3.7}$$

The similarities between clusters slightly varies from similarities between individual trajectories, but the final affinity is given by the same linear combination as in 3.1. The affinity matrix is updated after every merge operation. Based on the affinity matrix, the most similar clusters are merged greedily until only two clusters are left.

Spectral clustering is considered state-of-the-art unsupervised clustering algorithm for its time efficiency. However, it is not suitable for motion-based video segmentation, where objects and background have different sizes. The goal of RAMS is to segment the different moving objects, while all other non-moving regions should be in a single segment. The background region is usually much larger than the moving objects. From our experiments with spectral clustering, the background was always split into multiple segments in the segmentation result. Figure 3.4 shows these results for the video shown in Figure 1.3. Although the background regions had high similarity values between each other, they were still split into multiple segments. After investigation, the reason turned out to be the spectral clustering algorithm. Spectral clustering is biased to create even-sized segments. That is what made backgrounds split, therefore spectral clustering is not suitable for motion-based video segmentation against backgrounds. Nadler and Galun [3] have shown the limitations of spectral clustering, and how it fails when a dataset contains multi-size segments. For this reason, bottom-up hierarchal clustering was adopted since it has no bias with regard to cluster size.

This chapter describes the details of RAMS. It describes how temporal motion patterns are computed and compared to cluster superpixel trajectories. Performance evaluation of RAMS is reported in the next chapter. Conclusions and future work are included in the last chapter.

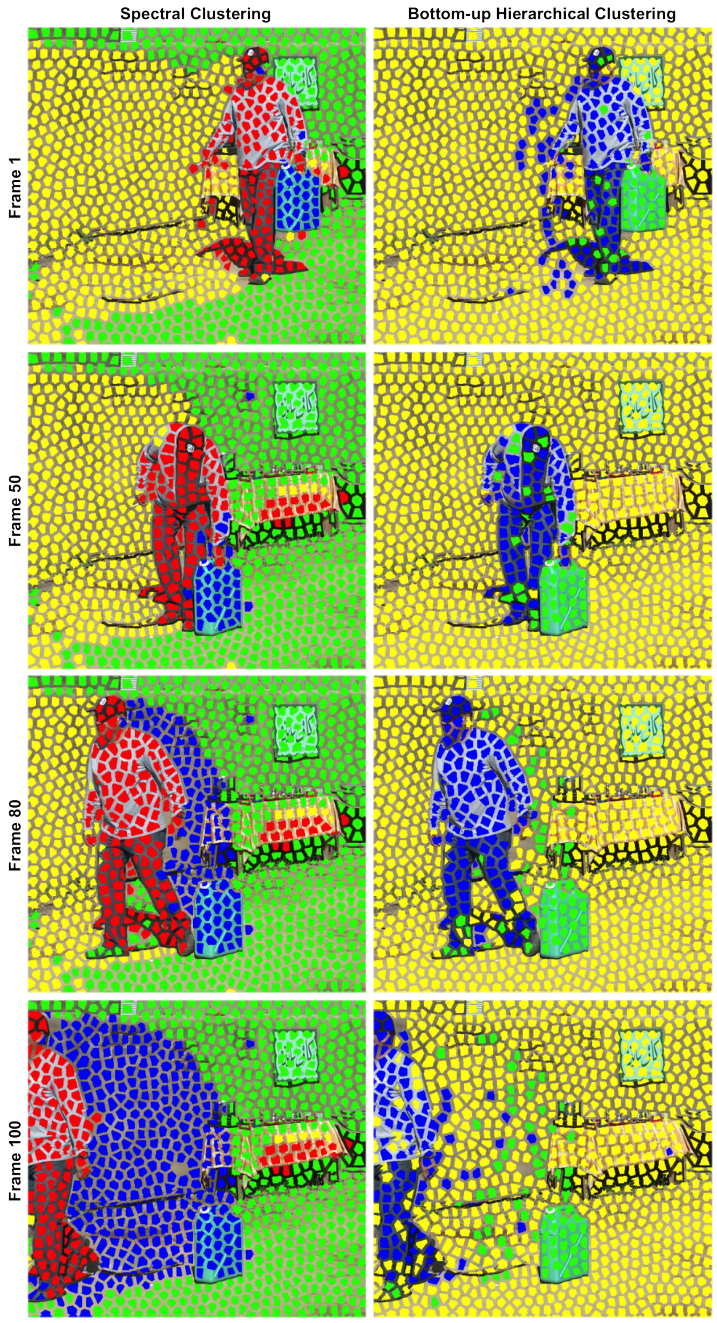


Figure 3.4: Spectral Clustering vs. Bottom-Hierarchal Clustering. Using *only* the *motion* similarities, the first column shows the spectral clustering result while the second column shows the bottom-up hierarchal clustering result, for the video shown in Figure 1.3. The number of clusters is four for the two settings.

Chapter 4

Performance Evaluation

In video segmentation evaluation, approaches are evaluated by comparing their segmentation results with hand-labeled segmentations called ground truth. A set of evaluation metrics are used for this comparison to generate some scores for an evaluated approach. Thus, approaches are compared by comparing their evaluation scores. The evaluation goal in this section is threefold:

- The first goal is to examine the performance of the proposed approach (RAMS). Are the results as expected, and is the approach performing as designed? RAMS differs from other related approaches in that it addresses the problem of non-rigid motion. The goal of the approach is to segment moving objects, whether moving rigidly or non-rigidly, as wholes by using motion temporal coherence. So the results should segment each moving object without breaking it into different parts. Thus the first goal of evaluation is testing how well RAMS succeeds in achieving its segmentation goals.

- The second goal is comparing the segmentation results of RAMS with human segmentations. Humans can visually detect and segment objects in videos. However, when a group of people are assigned the job of segmenting a video, they provide different answers to the same video segmentation task. Comparing the segmentation results of RAMS to what humans perceive as correct segmentations is the second goal of the evaluation.

- The third goal is comparing RAMS with other related approaches in the literature. The approaches are different in nature (group pixels, superpixels, or point-trajectories), goal (motion or color segmentation), output (single or hierarchal segmentation), and

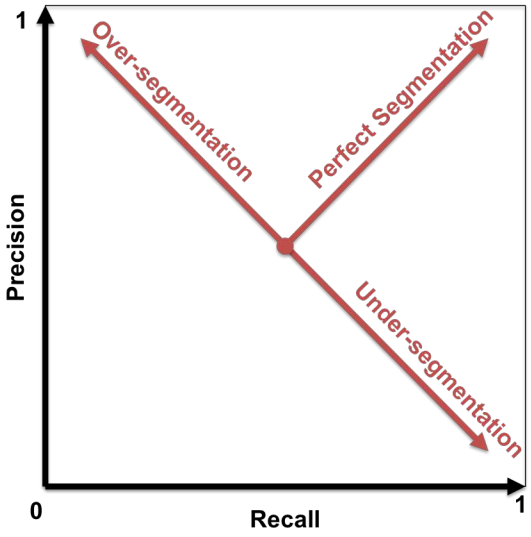


Figure 4.1: Interpretation of point locations in a VPR plot.

assumptions (rigid or non-rigid motion). So a comparison on a standard dataset is the third goal of the evaluation.

RAMS is evaluated using the evaluation measures originally suggested in [9], and described in the next section. The scores measure how well a segmentation result matches human-labeled ground truth. Of particular interest, Volume Precision-Recall (VPR) is used for evaluating video segmentations. Video segments are compared with ground truth video segments by inspecting the pixel overlap. Precision (P) measures how well a video segment is encapsulated within a ground truth segment. Perfect precision is scored by a video segmentation where none of its segments overlap with multiple ground truth segments. On the other hand, recall (R) measures how well the ground truth segments are covered by the tested segmentation segments. The equations are included in the next section, and more details with working examples are provided in Appendix A.

In general, over-segmentation results in high precision values but low recall values. Conversely, under-segmentation results in high recall values but low precision values. Good segmentations have high values for both precision and recall. Figure 4.1 shows how to interpret a point location for a segmentation result score. Most algorithms, including RAMS,

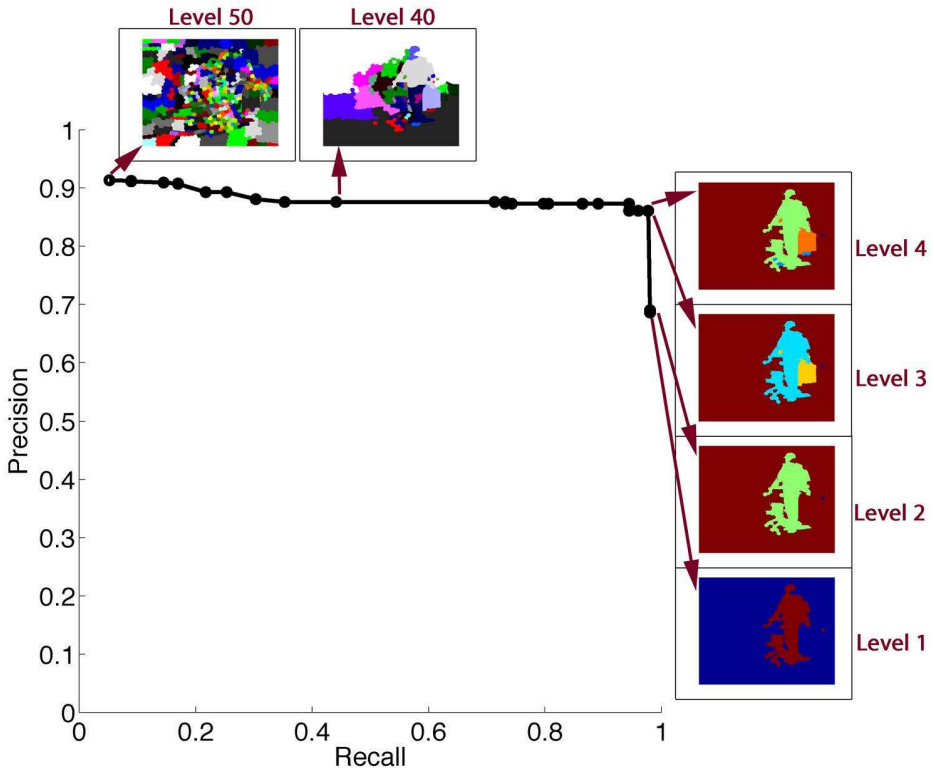


Figure 4.2: An example of a VPR plot.

produce a sequence of segmentations, ranging from over-segmented to under-segmented results. This creates a curve, starting in the upper left (over-segmented) corner of the graph, and heading toward the lower right (under-segmented) corner. The closer the curve gets to the upper right corner of the figure, the better. An example is shown in Figure 4.2 for RAMS result on the man and gas can video that have been previously shown in the introduction in Figure 1.3. It shows a curve with 50 points corresponding to 50 segmentations, each with fewer regions than the one before it. Segmentations 1, 2, 3, 4, 40, and 50 are shown in the figure for the first frame of the video. The best F-measure is achieved by the third level where the scores are F(R = 0.98, P = 0.86) = 0.92. After that, the gas can is merged with the background, which causes a drop in the precision scores.

The result of RAMS is a sequence of segmentations, where the number of clusters decreases from one segmentation to the next. The finest segmentation contains a cluster for every superpixel trajectory, and the coarsest segmentation contains only two clusters. From

these segmentations, 50 are selected for evaluation. Since we evaluate on the VSB100 [9] dataset, and the maximum number of ground truth labels for a video in this dataset is 21, the first 41 coarsest segmentations are included in the evaluation. In addition, the segmentations where the number of clusters is 50, 100, 150, 200, 250, 300, 400, 500, and 600, are also included in the evaluation.

We evaluate RAMS and three other state-of-the-art approaches that provide publicly available code: Grundmann et al. [16] using their improved code in [17], Ochs and Brox [25], and Xu et al. [7]. The evaluation of the superpixel-trajectories that RAMS starts with, provided by the TSP [12] algorithm, is also included for the VSB100 [9] dataset results. Galasso et al. [8] did not have a publicly available code at the time of this dissertation, while the Galasso et al. [10] code proved too computationally expensive to be feasible. Thus [8] and [10] were not included in the presented evaluation. Xu et al's. [7] approach only uses the color information with no motion information. This approach was included in this evaluation since it was included in the comparison of [9].

4.1 Evaluation Measures

The VSB100 [9] dataset is composed of 100 videos. Every twentieth frame from each video is human-labeled as ground truth. The selected frames are labeled by different people to encompass the different views and opinions of people to what are the objects in the video. Some hand labellings have more details than others. The benchmark aggregates an approach score over all the available ground truth segmentations, to accompany the different video segmentation approaches.

The VSB100 [9] benchmark evaluates the performance of algorithms using two types of metrics. The first one evaluates the boundaries in the segmentation result, and it is called Boundary Precision-Recall (BPR). However, this metric evaluates each frame independently; whether they are temporally consistent or not does not affect the score. So the benchmark

includes a second metric, called Volume Precision-Recall (VPR), that evaluates how well the segments' 3D volumes of the segmentation result fit the ground truth segments' volumes. The following describes these two metrics.

4.1.1 Volume Precision-Recall (VPR) [9]

The Volume Precision-Recall (VPR) [9] evaluates the pixel overlap between the segments of a segmentation result (S) generated by an algorithm, and the segments of a ground truth (G), with two metrics: precision (P) and recall (R). Precision maps the segments of S to the segments of G. Each segment s in S is matched to a ground truth segment g from G that has maximal overlap with s. Then the precision of the segmentation S is the average of the overlaps between its segments and their matched ground truth segments. Since multiple ground truth segmentations are provided for each video, the final precision is the average of precisions with each of these ground truth segmentations. So precision is expressed as:

$$P = \frac{1}{M} \sum_{i=1}^{M} \frac{\sum_{s \in S} max_{g \in G_i} |s \bigcap g|}{|S|}, \tag{4.1}$$

where the intersection $\bigcap$ is the pixel overlap, $|.|$ denotes the number of pixels, and M is the number of provided ground truth segmentations. In general, as a segment s becomes more fitted inside its matched ground segment g, its precision increases. So a perfect precision score is achieved by a segmentation when all its segments are fitted within the ground truth segments, no matter how small they are. Thus generally, over-segmentations usually achieve high precision scores.

On the contrary, recall maps the ground truth segments of G to the segments of S, in the same manner as precision. Each ground truth segment g from G is matched with a segment s in S that has maximal overlap with g. So recall is expressed as:

$$R = \sum_{i=1}^{M} \frac{\sum_{g \in G_i} max_{s \in S} |g \bigcap s|}{\sum_{i=1}^{M} |G_i|}. \tag{4.2}$$

In general, as a ground truth segment g gets covered by less segments from S, its pixel overlap with its matched segment s increases, and thus its recall increases. Note that it does not matter if the matched segment s exceeds the boundaries of g. Thus generally, under-segmentations usually achieve high recall scores.

However, there is an issue with these definitions. Consider the two degenerate cases: the first case is when each pixel is in its own segment, and the second case is when all pixels are in a single segment. The first case will result in a perfect precision score, while the second case will result in a perfect recall score. So the benchmark normalizes these values to avoid this problem, and thus precision and recall are re-expressed as:

$$P = \frac{\sum_{i=1}^{M}[(\sum_{s\in S} max_{g\in G_i}|s \bigcap g|) - max_{g\in G_i}|g|]}{M|S| - \sum_{i=1}^{M} max_{g\in G_i}|g|}, \text{ and} \tag{4.3}$$

$$R = \frac{\sum_{i=1}^{M}\sum_{g\in G_i}(max_{s\in S}|g \bigcap s| - 1)}{\sum_{i=1}^{M}(|G_i| - \Gamma_{G_i})}, \tag{4.4}$$

where Γ_{G_i} is the number of segments in the ground truth G_i. Appendix A provides an additional description of VPR by examples.

4.1.2 Boundary Precision-Recall (BPR) [9]

Boundary Precision-Recall (BPR) [9] measures the quality of boundary detection in the tested approach. It is usually used in image segmentation evaluation, but can be used for evaluating video segmentations. For a given frame with an associated ground truth segmentation, both the frame segmentation and the ground truth segmentation are transformed to binary boundary maps. A boundary map is a classification of each pixel being either a boundary pixel or not. Then the metric evaluates how well they overlap using a precision-recall framework. Letting S be the boundary map for the frame's segmentation result, and G_i be the boundary map for the i^{th} ground truth segmentation for M available ground truth

segmentations, the precision (P) and recall (R) can be defined as:

$$P = \frac{|S \bigcap (\bigcup_{i=1}^{M} G_i)|}{|S|}, \text{ and} \tag{4.5}$$

$$R = \frac{\sum_{i=1}^{M} |G_i \bigcap S|}{\sum_{i=1}^{M} |G_i|}, \tag{4.6}$$

where the intersection $\bigcap$ is the bipartite graph assignment between the given boundary maps, and $|.|$ denotes the number of matched pixels. The precision is the percentage of correct pixel assignment according to the ground truth, while recall is the percentage of ground truth that is covered by the segmentation.

Although BPR is generally a good measure for evaluating the quality of boundaries in segmentation results, it is less interesting for RAMS. That is because the accuracy of objects' boundaries is largely the responsibility of the underlying superpixel segmentation, which is TSP [12]. So the boundary accuracy of the final segmentation result is dependent on the boundary accuracy of the superpixels provided by TSP [12]. The goal of the proposed approach is to find temporally consistent object segmentation of a video by analyzing the objects' motion temporal patterns, and is not concerned with the accuracy of boundary detection. Thus VPR is more interesting for RAMS, and BPR is reported here for completeness.

4.1.3 F-measure Scores [9]

Precision and recall values are combined in the F-measure for aggregate performance evaluation and comparison:

$$F = \frac{2PR}{R + P}. \tag{4.7}$$

F-measure is used to report a final evaluation score of an evaluated approach, as in [9]. There are three scores reported for each approach: (i) Average Precision (AP), which is the area

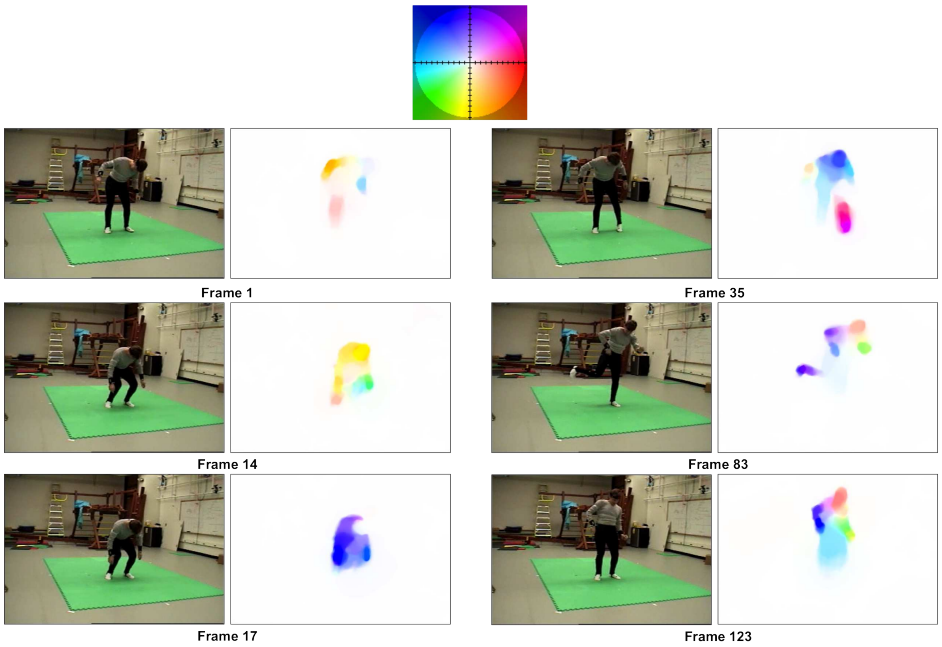


Figure 4.3: A video of a freely moving man [5], shown with six frames and their corresponding forward optical flow. This is an extreme case of non-rigid motion, and it is evident in the optical flow as the parts have different colors meaning that they are moving in different directions. The optical flow was part of the [32] result on this video, and it was color-coded with [33].

under the curve, (ii) Optimal Segmentation Scale (OSS) score, which is the F-measure when an optimal scale (level) is selected for each segmentation, and (iii) Optimal Dataset Scale (ODS) score, which is the F-measure when an optimal fixed scale (level) is selected for all segmentations.

4.2 New Dataset Results

A new small dataset is proposed to include videos with more challenging non-rigid motion than standard datasets. The purpose of this dataset is to demonstrate the strengths of the proposed approach, and to demonstrate where other approaches may fail. This dataset includes three videos: (i) the zebra video [18] shown before in Figure 1.2, (ii) the man with gas can video shown before in Figure 1.3, and (iii) another video of a freely moving man [5] shown in Figure 4.3. The lengths of these videos are 152, 110, and 150 frames,

respectively. Ground truth is hand-labeled for each video. The performance of RAMS and other related approaches is compared on this dataset. The evaluation method in [9] evaluates every twentieth frame for a video, so these are the frames that are shown for the results for visual comparison. The following sub-sections discuss the segmentation results on each of these videos.

When processing the zebras and freely moving man videos of this dataset, the boundary pixels of superpixels were not discarded as done for the other videos. Recall that pixels on superpixel boundaries are discarded during processing since they are unreliable. However, these two videos have very small image dimensions per frame, and thus discarding the boundary pixels ends up in discarding the important trajectories of the videos. So the boundary pixels of the superpixels of these two videos were kept during processing.

4.2.1 Zebras

The first video of the new dataset contains three zebras [18], described in detail in Figure 1.2. As discussed previously, zebras move non-rigidly; their parts can move in different directions at the same time. The ground truth for this video is hand-labeled and shown in the second column of Figure 4.4. The VPR and BPR evaluation scores are included in Table 4.1, and the plots are shown in Figure 4.5. The results can be visually compared in Figure 4.4. From these results, RAMS outperforms state-of-the-art approaches on the zebra video. It achieved higher scores, and was visually closer to the ground truth segmentation.

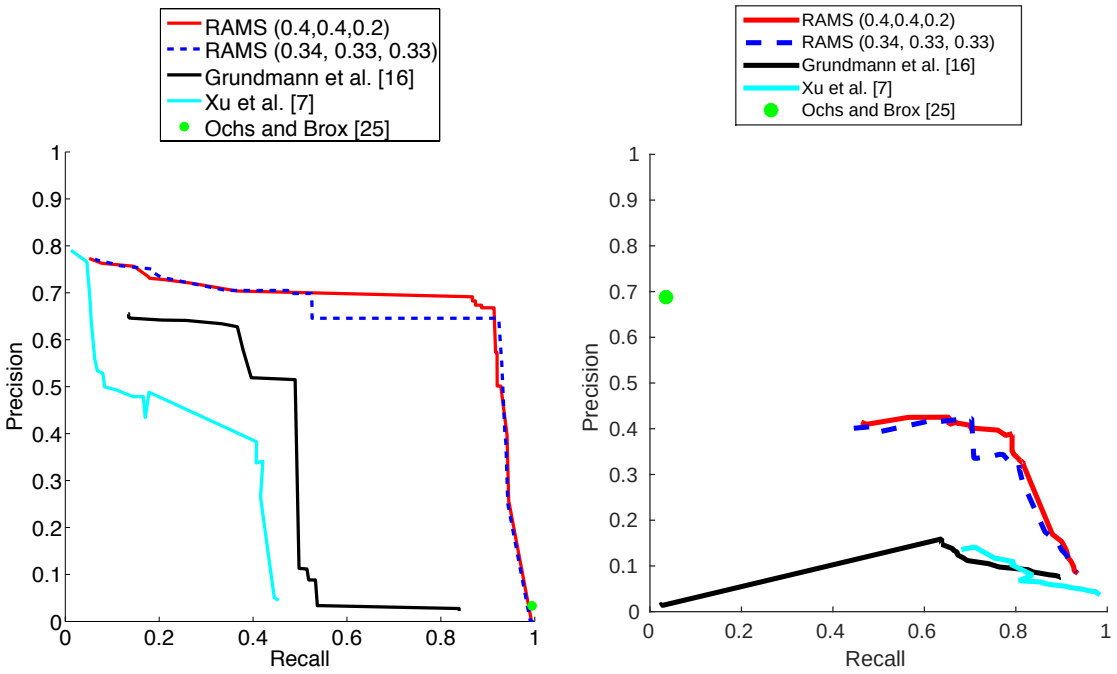
From Figure 4.4, Grundmann et al. [16] and Xu et al. [7] results are over-segmentations, while Ochs and Brox [25] result is an under-segmentation. Xu et al. [7] uses the color cue only, so it is extremely sensitive to color differences in the video so the background is highly over-segmented. Grundmann et al. [16] is also sensitive to color differences, so it also divides the background into many pieces. The zebras look alike, so they get merged when they get close to each other, and split when they separate, in the results of both approaches [16] and [7]. In Grundmann et al. [16] result, a large portion of the big zebra's back is merged with

the background in the middle frames of the video as the zebra stops walking. In addition, the legs of the big zebra segments differently in some frames because they move differently. The levels shown for both approaches [16] and [7] are the best visually because the separation of the zebras in some frames. In regards to Ochs and Brox [25] result, it labels the pixels of each frame independently by extending the point-trajectories labeling of Brox and Malik [32]. Figure 4.6 shows the segmentation of Brox and Malik [32]. From this figure, Brox and Malik [32] was not able to segment the zebras correctly, and that affects Ochs and Brox [25] result. In most frames, the majority of points within the zebras are merged with the background. Only at the end of the video when the labels on the big zebra get denser, and even then the zebra is divided into two segments because of their rigid motion assumption. However, when the corresponding frames of this time range, Ochs and Brox [25] result merges these two segments probably because they use superpixel boundaries in their optimization function. Note that in [32], color is not used for grouping point-trajectories.

Figure 4.4: The zebras video [18] and its segmentation results. The frames 1, 21, 41, 61, 81, 101, 121, and 141 are shown in consecutive rows. The first column contains the original frames, while the second column contains a hand-labeled ground truth. Columns 3-6 contain the segmentation results of RAMS (0.34, 0.33, 0.33), Grundmann et al. [16], Xu et al. [7], and Ochs and Brox [25], respectively. The visually best levels are the levels shown in columns 3-5.

Table 4.1: VPR and BPR evaluation scores for the zebras video [18]. The scores are in the format: **F**(R, P). The ODS and OSS are identical because this evaluation is for a single video; the dataset scale and segmentation scale are the same.

VPR - Zebras Video			
Approach	**ODS**	**OSS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.77** (0.91, 0.67)	**0.77** (0.91, 0.67)	**0.68**
RAMS (0.34, 0.33, 0.33)	**0.76** (0.92, 0.65)	**0.76** (0.92, 0.65)	**0.66**
Grundmann et al. [16]	**0.50** (0.49, 0.52)	**0.50** (0.49, 0.52)	**0.32**
Ochs and Brox [25]	**0.06** (0.99, 0.03)	**0.06** (0.99, 0.03)	**0.03**
Xu et al. [7]	**0.39** (0.41, 0.38)	**0.39** (0.41, 0.38)	**0.21**
BPR - Zebras Video			
Approach	**ODS**	**OIS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.52** (0.79, 0.39)	**0.52** (0.79, 0.39)	**0.27**
RAMS (0.34, 0.33, 0.33)	**0.53** (0.70, 0.42)	**0.53** (0.70, 0.42)	**0.26**
Grundmann et al. [16]	**0.25** (0.64, 0.16)	**0.25** (0.64, 0.16)	**0.08**
Ochs and Brox [25]	**0.07** (0.03, 0.69)	**0.07** (0.03, 0.69)	**0.02**
Xu et al. [7]	**0.24** (0.71, 0.14)	**0.24** (0.71, 0.14)	**0.12**



(a) VPR (b) BPR

Figure 4.5: VPR and BBR plots for the zebras video [18]. The solid red and dashed blue lines are for RAMS results, where the difference is the weight combination (α_1, α_2, α_3) used: the blue dashed line is when equal weights are used, and the red solid line is when the combination (0.4, 0.4, 0.2) is used. RAMS, Grundmann et al. [16], and Xu et al. [7], all produce multiple segmentations with different number of clusters. Therefore their results are represented with curves; a point for each produced segmentation. Ochs and Brox [25] produce a single segmentation, thus their result is represented with a single point (green circle).

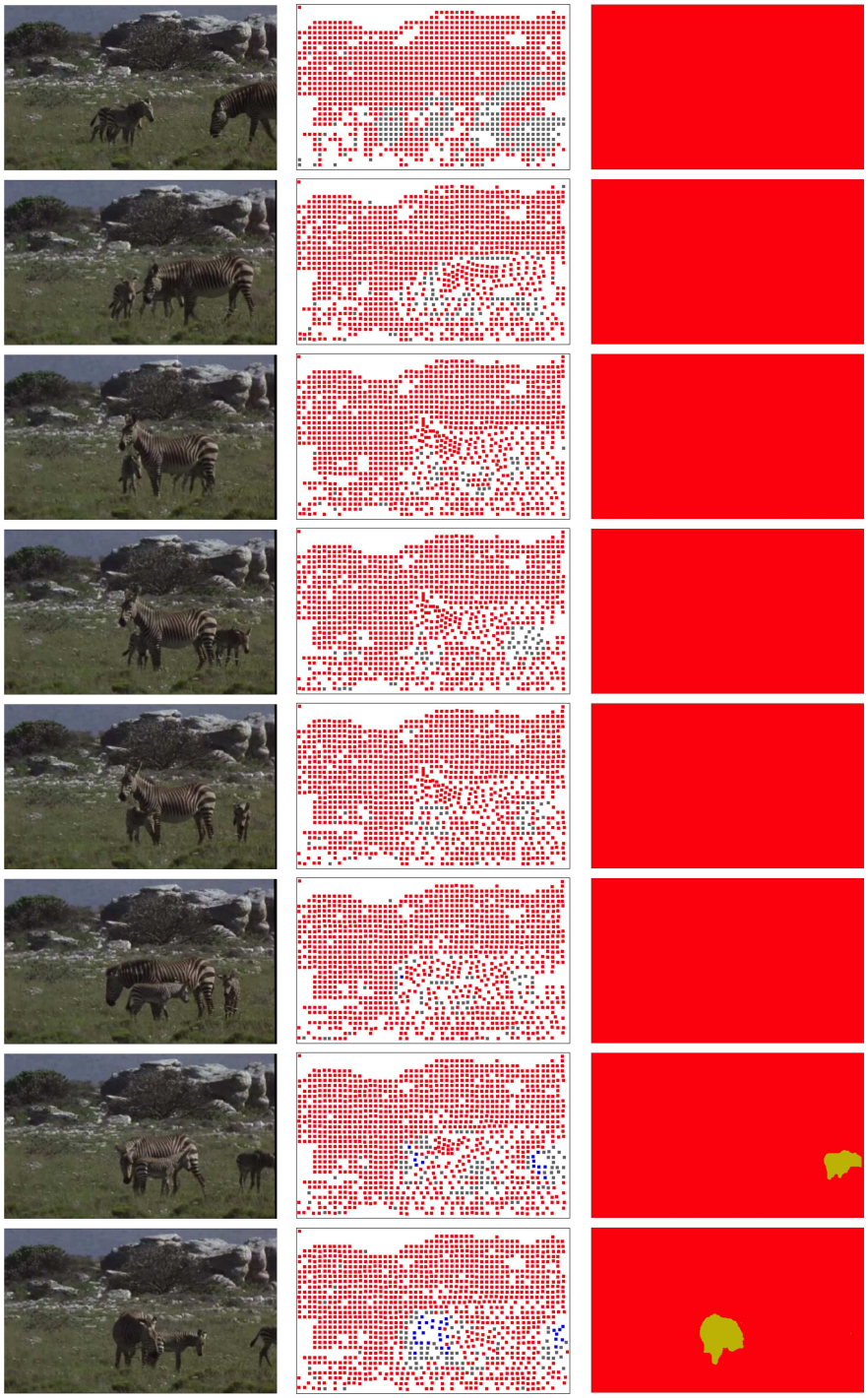


Figure 4.6: The comparison between the segmentation results of Brox and Malik [32], and Ochs and Brox [25] on the zebras video [18]. The frames 1, 21, 41, 101, 121, and 141 are shown in consecutive rows. The first column contains the original frames. The second and third columns contain the segmentation results of Brox and Malik [32], and Ochs and Brox [25], respectively. Brox and Malik [32] group point-trajectories, so each point is represented with a square in these images.

4.2.2 Man with Gas Can

The second video of the new dataset is a video of a man walking from right to left while carrying a gas can that he places on the ground midway through the video, as first shown in Figure 1.3, and discussed in the introduction. This video has a combination of motions; non-rigid motion by the man, and rigid motion by the gas can. Visual results are shown in Figure 4.7. From this figure, RAMS is able to segment the man and gas can correctly. The coarsest levels of Grundmann et al. [16] nd Xu et al. [7] results are shown in this figure, and they both divided the man into multiple parts. In addition, Xu et al. [7] overly segments the background. On the other hand, Ochs and Brox [25] was not able to segment the gas can. It also separates the legs because of their different motion.

Comparing to the hand-labeled ground truth shown in Figure 4.7, the VPR and BPR evaluation scores are reported in Table 4.2 for all approaches, and the plots are shown in Figure 4.8. RAMS achieved higher scores than the other approaches. The scores of RAMS (0.4, 0.4, 0.2) is less than RAMS (0.34, 0.33, 0.33) and Grundmann et al. [16] because of the occlusion issue discussed in Section 4.4.

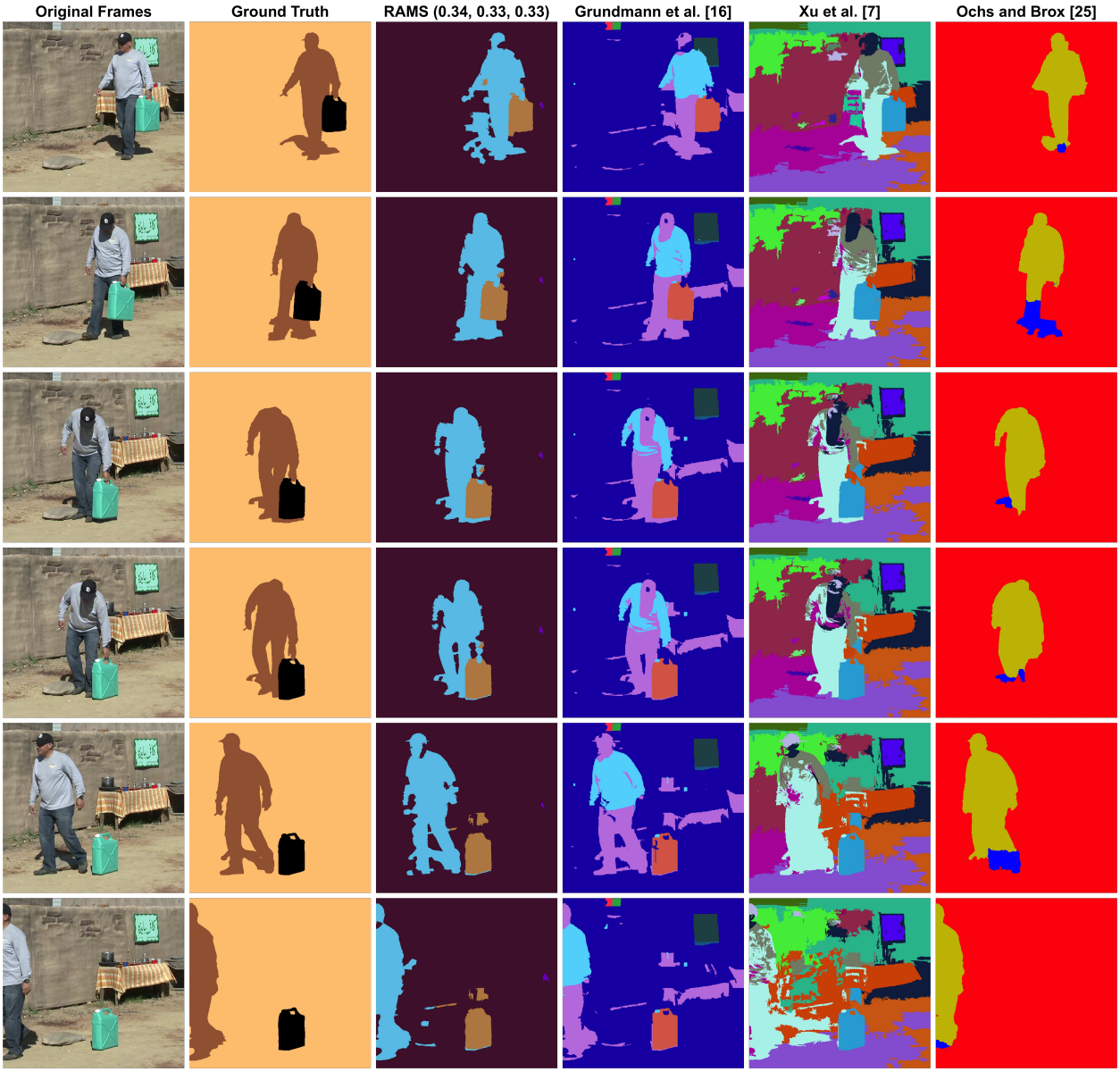
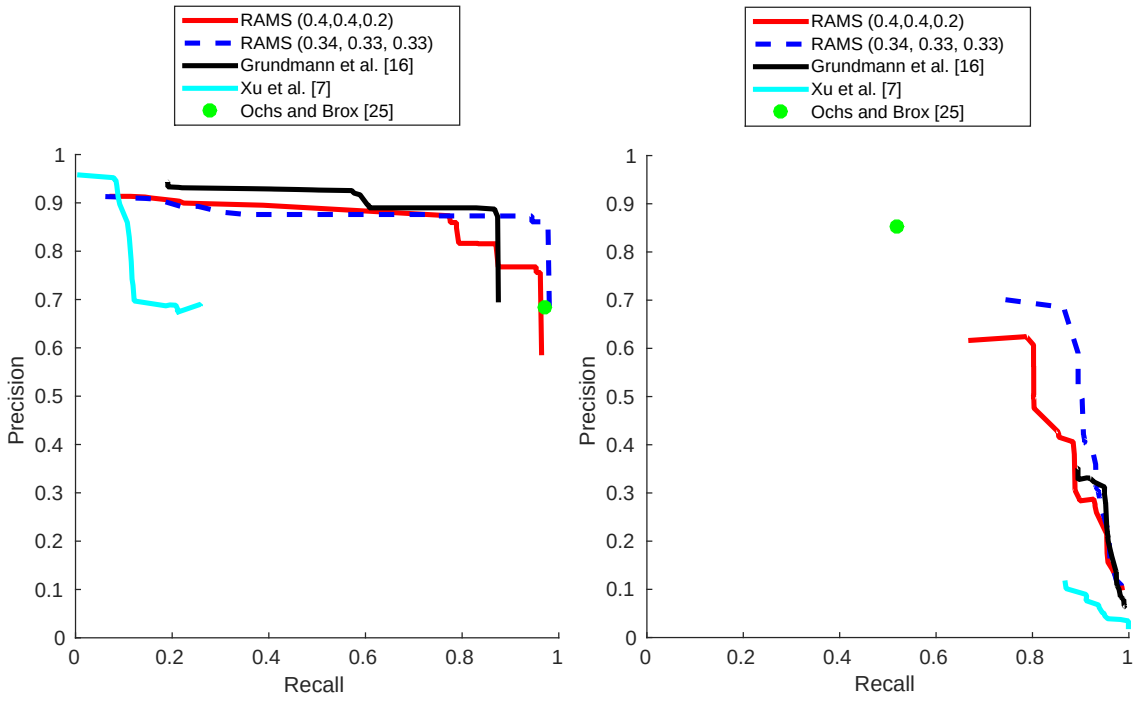


Figure 4.7: The man and gas can video and its segmentation results. The frames 1, 21, 41, 61, 81, 101, 121, and 141 are shown in consecutive rows. The first column contains the original frames, while the second column contains a hand-labeled ground truth. Columns 3-6 contain the segmentation results of RAMS (0.34, 0.33, 0.33), Grundmann et al. [16], Xu et al. [7], and Ochs and Brox [25], respectively. The coarsest levels are shown in columns 4 and 5.

Table 4.2: VPR and BPR evaluation scores for the man and gas can video. The scores are in the format: **F**(R, P). The ODS and OSS are identical because this evaluation is for a single video; the dataset scale and segmentation scale are the same.

VPR - Man and Gas Can Video			
Approach	**ODS**	**OSS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.85** (0.95, 0.77)	**0.85** (0.95, 0.77)	**0.85**
RAMS (0.34, 0.33, 0.33)	**0.92** (0.98, 0.86)	**0.92** (0.98, 0.86)	**0.87**
Grundmann et al. [16]	**0.88** (0.87, 0.89)	**0.88** (0.87, 0.89)	**0.81**
Ochs and Brox [25]	**0.80** (0.97, 0.68)	**0.80** (0.97, 0.68)	**0.66**
Xu et al. [7]	**0.38** (0.26, 0.69)	**0.38** (0.26, 0.69)	**0.21**
BPR - Man and Gas Can Video			
Approach	**ODS**	**OIS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.70** (0.79, 0.62)	**0.70** (0.79, 0.62)	**0.56**
RAMS (0.34, 0.33, 0.33)	**0.76** (0.87, 0.68)	**0.76** (0.87, 0.68)	**0.65**
Grundmann et al. [16]	**0.51** (0.89, 0.35)	**0.51** (0.89, 0.35)	**0.34**
Ochs and Brox [25]	**0.64** (0.52, 0.85)	**0.64** (0.52, 0.85)	**0.44**
Xu et al. [7]	**0.21** (0.87, 0.12)	**0.21** (0.87, 0.12)	**0.11**



(a) VPR (b) BPR

Figure 4.8: VPR and BPR plots for the man and gas can video.

4.2.3 Freely Moving Man

The third video contains a man with a wide range of motion [5]. He is bending his body while moving his arms and legs more freely than usual. This is an extreme case of non-rigid motion. Figure 4.9 shows every twentieth frame of the video in its first column. The ground truth segmentation for this video is two segments: a segment for the moving man, and a segment for the background, as shown in the second column of Figure 4.9. The segmentation results are visually shown in Figure 4.9, which contains the coarsest segmentations. RAMS was able to segment the man and the background. Even though the boundaries of the man segment is not accurate, RAMS did put the man in a single segment as opposed to some other approaches. Grundmann et al. [16] and Xu et al. [7] approaches divide the man and combine his parts with larger portions of the background. The approaches [16] and [7] also add more segments in the background such as the mat. However, the mat is stationary, and thus should be combined with the rest of the background. Ochs and Brox [25] segment the man at first, but then it starts to merge with the background. VPR and BPR evaluation scores are reported in Table 4.3, and the plots are shown in Figure 4.10. RAMS achieved higher scores than the other approaches.

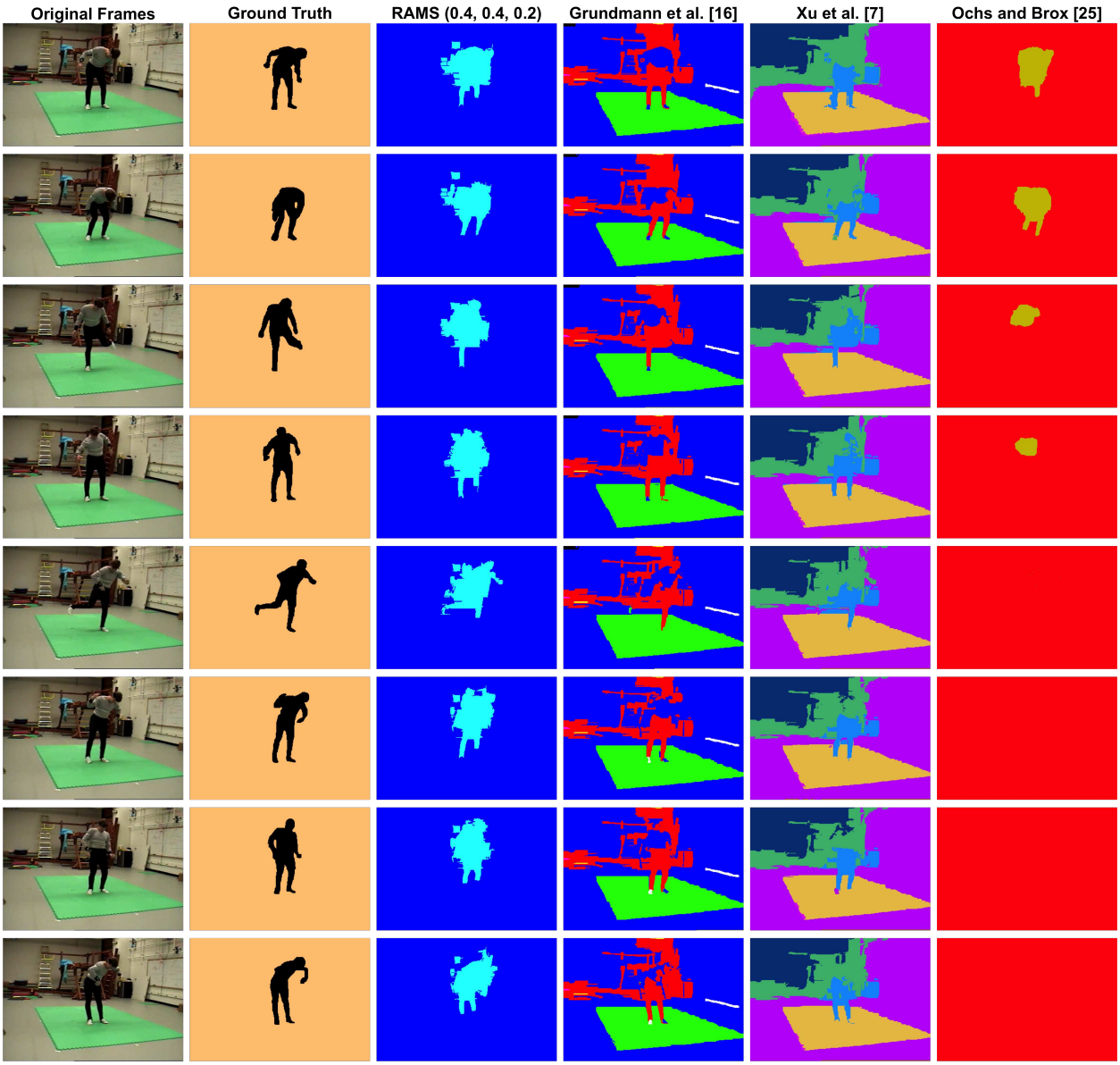
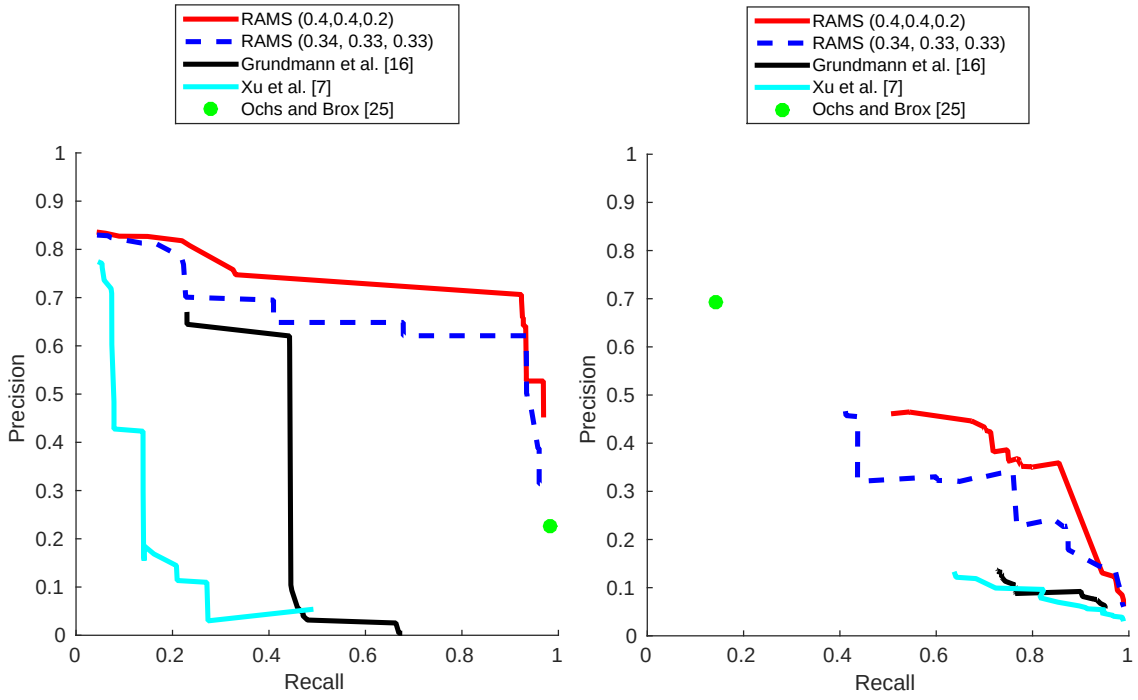


Figure 4.9: The freely moving man video [5] and its segmentation results. The frames 1, 21, 41, 61, 81, 101, 121, and 141 are shown in consecutive rows. The first column contains the original frames, while the second column contains a hand-labeled ground truth. Columns 3-6 contain the segmentation results of RAMS (0.4, 0.4, 0.2), Grundmann et al. [16], Xu et al. [7], and Ochs and Brox [25], respectively. The coarsest levels are shown in columns 3-5.

Table 4.3: VPR and BPR evaluation scores for the freely moving man video [5]. The scores are in the format: **F**(R, P). The ODS and OSS are identical because this evaluation is for a single video; the dataset scale and segmentation scale are the same.

VPR - Freely Moving Man Video			
Approach	**ODS**	**OSS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.80** (0.92, 0.71)	**0.80** (0.92, 0.71)	**0.73**
RAMS (0.34, 0.33, 0.33)	**0.75** (0.93, 0.62)	**0.75** (0.93, 0.62)	**0.66**
Grundmann et al. [16]	**0.52** (0.44, 0.62)	**0.52** (0.44, 0.62)	**0.30**
Ochs and Brox [25]	**0.37** (0.98, 0.23)	**0.37** (0.98, 0.23)	**0.22**
Xu et al. [7]	**0.21** (0.14, 0.42)	**0.21** (0.14, 0.42)	**0.12**
BPR - Freely Moving Man Video			
Approach	**ODS**	**OIS**	**AP**
RAMS (0.4, 0.4, 0.2)	**0.54** (0.67, 0.45)	**0.54** (0.67, 0.45)	**0.41**
RAMS (0.34, 0.33, 0.33)	**0.47** (0.76, 0.34)	**0.47** (0.76, 0.34)	**0.36**
Grundmann et al. [16]	**0.23** (0.73, 0.14)	**0.23** (0.73, 0.14)	**0.12**
Ochs and Brox [25]	**0.23** (0.14, 0.69)	**0.23** (0.14, 0.69)	**0.10**
Xu et al. [7]	**0.22** (0.64, 0.13)	**0.22** (0.64, 0.13)	**0.11**



(a) VPR

(b) BPR

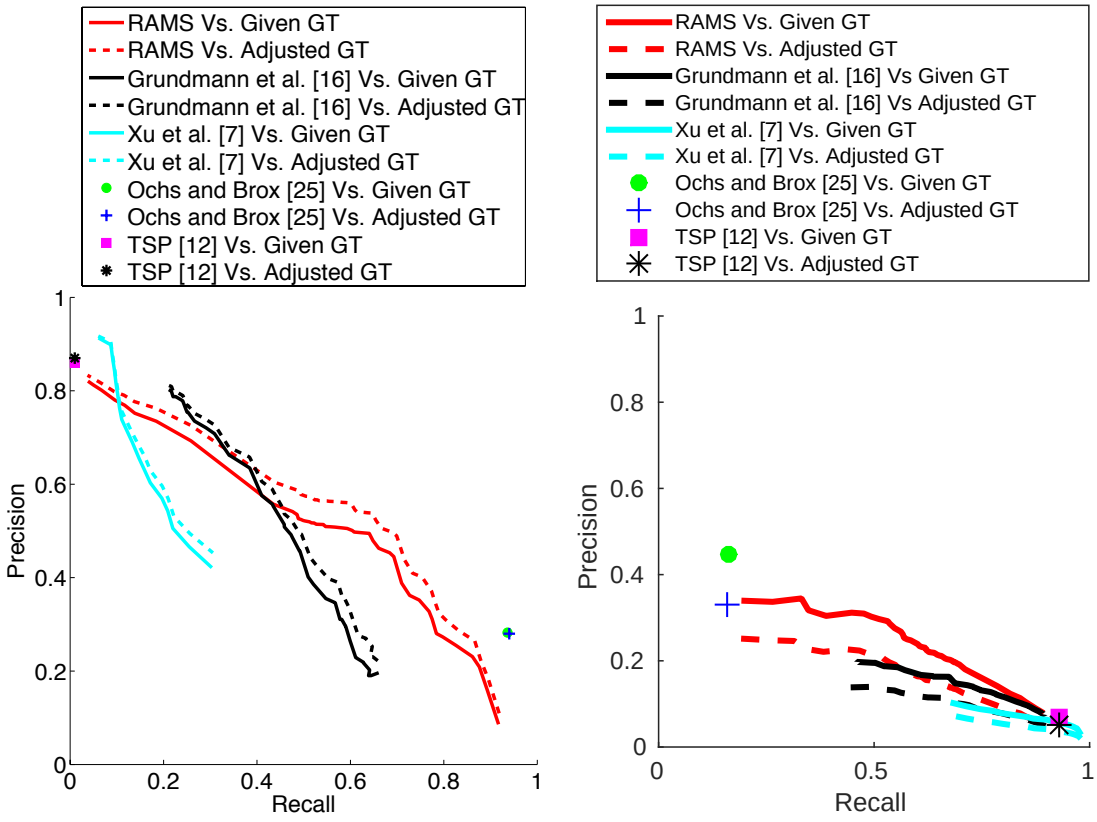
Figure 4.10: VPR and BPR plots for the freely moving man video [5].

4.3 VSB100 [9] Dataset Results

Galasso et al. [9], presented a video dataset called VSB100, consisting of 100 HD videos. This video set is originally from the Berkeley Video Dataset first introduced in [26], and its purpose is occlusion boundary detection. However, Galasso et al. [9] adopted this set as a general video segmentation benchmark because of its diversity, and provided the ground truth segmentations. The length of these videos range from 31 to 121 frames. The goal of RAMS is to segment moving objects as wholes, without breaking them up into different parts. Non-moving objects should be combined in a single background segment. However, some ground truth segmentations provided in VSB100 [9] do not fit with this segmentation goal; they either have less or more details. As a result, the evaluation on this dataset is performed twice: using the given ground truth, and using an adjusted ground truth. The adjustment is done by either merging or splitting some segments, to test how well RAMS succeeds in achieving its segmentation goals.

This dataset contains videos that were captured with moving cameras. However, as discussed above, RAMS assumes a fixed camera. If a video has jitter, it is stabilized using [28] and then processed by RAMS. Otherwise, it is not used for evaluation. As a result, 41 videos out of 100 are used in this evaluation, and their names are listed the first columns of Tables 4.5 and 4.6. In addition, as in [9], all videos are resized by half to reduce the computational expense.

The VPR and BPR evaluation scores for the VSB100 [9] dataset are shown in Table 4.4. VPR scores for each video individually are reported in Tables 4.5 and 4.6. The VPR and BPR plots are shown in Figure 4.11. From these results, RAMS outperforms state-of-the-art approaches on this dataset; it achieves higher F scores (see Table 4.4). Some approaches achieve either higher precision or higher recall, but not both. The F score combines both recall and precision in one measure, and requires a balance between them to achieve a high value. In addition, the VPR curve for RAMS covers different levels of segmentations from



(a) VPR

(b) BPR

Figure 4.11: VPR and BPR plots for the VSB100 [9] dataset, using the given and adjusted ground truth (GT).

over-segmentations to under-segmentations. Defining what is a "correct" segmentation is a matter of opinion. So this variation is an advantage that leaves the selection of the optimal level to the end-user or end-system.

RAMS depends on the accuracy of the underlying superpixel segmentation. If a superpixel boundary is not accurate, it may contain pixels of multiple objects within it. This causes mixed motion information for this superpixel, and also affects the correctness of its trajectory. This usually happens to very small objects, such as the tennis ball in the video "tennis_tr" of the VSB100 [9] dataset.

Table 4.4: VPR and BPR evaluation scores for the 41 videos from VSB100 [9] dataset with roughly stationary cameras, using the given ground truth and adjusted ground truth. The scores are in the format: **F**(R, P).

VPR - VSB100 Dataset						
Approach	**Using Given Ground Truth**			**Using Adjusted Ground Truth**		
	ODS	**OSS**	**AP**	**ODS**	**OSS**	**AP**
RAMS	**0.56** (0.64, 0.49)	**0.62** (0.68, 0.57)	**0.50**	**0.59** (0.65, 0.54)	**0.65** (0.71, 0.61)	**0.54**
TSP [12]	**0.01** (0.01, 0.86)	**0.01** (0.01, 0.86)	**0.01**	**0.01** (0.01, 0.87)	**0.01** (0.01, 0.87)	**0.01**
Grundmann et al. [16]	**0.49** (0.43, 0.56)	**0.52** (0.49, 0.56)	**0.41**	**0.50** (0.43, 0.59)	**0.54** (0.50, 0.60)	**0.42**
Ochs and Brox [25]	**0.43** (0.94, 0.28)	**0.43** (0.94, 0.28)	**0.26**	**0.43** (0.94, 0.28)	**0.43** (0.94, 0.28)	**0.26**
Xu et al. [7]	**0.35** (0.30, 0.42)	**0.35** (0.26, 0.54)	**0.21**	**0.37** (0.31, 0.45)	**0.36** (0.26, 0.57)	**0.22**
BPR - VSB100 Dataset						
Approach	**Using Given Ground Truth**			**Using Adjusted Ground Truth**		
	ODS	**OIS**	**AP**	**ODS**	**OIS**	**AP**
RAMS	**0.38** (0.53, 0.29)	**0.39** (0.60, 0.29)	**0.24**	**0.30** (0.47, 0.22)	**0.31** (0.59, 0.21)	**0.17**
TSP [12]	**0.13** (0.93, 0.07)	**0.13** (0.93, 0.07)	**0.06**	**0.09** (0.93, 0.05)	**0.09** (0.93, 0.05)	**0.04**
Grundmann et al. [16]	**0.28** (0.50, 0.20)	**0.29** (0.66, 0.19)	**0.16**	**0.22** (0.49, 0.14)	**0.22** (0.65, 0.13)	**0.11**
Ochs and Brox [25]	**0.24** (0.16, 0.45)	**0.24** (0.16, 0.45)	**0.07**	**0.21** (0.16, 0.33)	**0.21** (0.16, 0.33)	**0.05**
Xu et al. [7]	**0.18** (0.68, 0.10)	**0.18** (0.72, 0.10)	**0.09**	**0.13** (0.69, 0.07)	**0.13** (0.72, 0.07)	**0.06**

Table 4.5: VPR evaluation scores for the 41 videos from VSB100 [9] dataset, using the given ground truth. The evaluation is performed on each video individually. The ODS and OSS are identical since its a single video evaluation, and thus the dataset scale and the segmentation scale are the same. So only the OSS score is reported in this table.

Video	VPR - Using Given Ground Truth							
	RAMS		Grundmann et al. [16]		Ochs and Brox [25]		Xu et al. [7]	
	OSS	AP	OSS	AP	OSS	AP	OSS	AP
arctic_kayak	0.59	**0.51**	**0.62**	0.49	0.33	0.17	0.46	0.36
baseball	**0.53**	**0.43**	0.44	0.30	0.08	0.04	0.24	0.12
beach_volleyball	0.30	0.18	**0.38**	**0.21**	0.01	0.01	0.17	0.10
big_wheel	0.23	0.13	**0.34**	**0.22**	0.20	0.10	0.19	0.12
birds_of_paradise	**0.87**	**0.84**	0.54	0.5	0.36	0.20	0.42	0.26
buffalos	0.55	**0.52**	0.41	0.36	**0.60**	0.36	0.44	0.31
car_jump	**0.62**	**0.50**	0.38	0.22	0.01	0	0.4	0.25
chrome	0.28	**0.17**	**0.35**	0.16	0.04	0.02	0.24	0.11
dominoes	**0.65**	**0.67**	0.56	0.44	0.35	0.21	0.48	0.30
freight_train	0.73	**0.67**	0.67	0.49	**0.74**	0.58	0.38	0.21
frozen_lake	0.37	0.22	**0.51**	**0.37**	0.01	0	0.32	0.17
gray_squirrel	**0.71**	**0.60**	0.61	0.50	0.70	0.51	0.32	0.13
guitar	0.48	**0.48**	**0.52**	0.47	0.11	0.06	0.32	0.18
hummingbird	0.71	0.59	**0.76**	**0.63**	0.23	0.12	0.22	0.12
juggling	**0.90**	**0.85**	0.78	0.68	0	0	0.43	0.36
jungle_cat	**0.63**	**0.56**	0.5	0.47	0.02	0.01	0.37	0.28
kangaroo_fighting	0.49	**0.39**	**0.54**	0.36	0.44	0.27	0.33	0.22
kia_commercial	0.74	0.67	**0.83**	**0.72**	0.67	0.50	0.31	0.18
knot	**0.77**	**0.67**	0.74	0.63	0.02	0.01	0.41	0.23
koala	0.49	**0.46**	**0.52**	0.42	0.44	0.26	0.37	0.20
lion	0.12	0.05	0.19	0.06	0	0	**0.29**	**0.10**
palm_tree	**0.7**	**0.52**	0.50	0.33	0	0	0.20	0.09
penguins	**0.6**	**0.59**	0.37	0.37	0.25	0.14	0.41	0.31
pepsis_wasps	0.41	**0.20**	0.23	0.08	0	0	**0.42**	0.19
planet_earth_one	**0.74**	**0.69**	0.47	0.31	0.65	0.48	0.32	0.18
pouring_tea	**0.73**	**0.66**	0.42	0.28	0.47	0.29	0.28	0.14
rock_climbing	0.41	**0.26**	0.39	0.19	0	0	**0.43**	0.24
rock_climbingtwo	**0.71**	**0.60**	0.55	0.42	0.13	0.07	0.31	0.18
roller_coaster	0.34	**0.28**	**0.36**	0.18	0	0	0.1	0.05
rolling_pin	**0.73**	**0.64**	0.44	0.32	0.65	0.45	0.47	0.29
salsa	0.36	0.24	**0.45**	**0.26**	0.09	0.04	0.32	0.22
shark_attack	**0.56**	**0.38**	0.46	0.30	0	0	0.30	0.16
sitting_dog	**0.67**	**0.60**	0.49	0.38	0.64	0.45	0.39	0.28
sled_dog_race	**0.32**	**0.25**	0.28	0.16	0	0	0.11	0.05
snow_leopards	0.56	**0.53**	**0.62**	0.51	0.46	0.27	0.47	0.37
street_food	**0.79**	**0.75**	0.53	0.49	0.71	0.54	0.43	0.27
tennis_tr	0.38	0.24	**0.54**	**0.35**	0	0	0.21	0.09
trampoline	**0.84**	**0.72**	0.52	0.33	0.70	0.54	0.18	0.10
up_dug	**0.79**	**0.70**	0.62	0.49	0.22	0.12	0.33	0.18
white_tiger	**0.63**	0.54	0.56	0.43	0.44	0.28	0.21	0.11
zoo	**0.73**	**0.65**	0.50	0.44	0.53	0.32	0.47	0.31

Table 4.6: VPR evaluation scores for the 41 videos from VSB100 [9] dataset, using an adjusted ground truth. The evaluation is performed on each video individually. The ODS and OSS are identical since its a single video evaluation, and thus the dataset scale and the segmentation scale are the same. So only the OSS score is reported in this table.

Video	VPR - Using Adjusted Ground Truth							
	RAMS		Grundmann et al. [16]		Ochs and Brox [25]		Xu et al. [7]	
	OSS	AP	OSS	AP	OSS	AP	OSS	AP
arctic_kayak	0.59	**0.55**	**0.60**	0.48	0.48	0.28	0.46	0.35
baseball	**0.54**	**0.43**	0.44	0.30	0.08	0.04	0.24	0.12
beach_volleyball	0.29	0.17	**0.38**	**0.2**	0.01	0.01	0.17	0.10
big_wheel	0.25	0.14	**0.34**	**0.22**	0.24	0.13	0.20	0.12
birds_of_paradise	**0.90**	**0.88**	0.57	0.52	0.37	0.21	0.47	0.29
buffalos	0.55	**0.53**	0.42	0.37	**0.59**	0.35	0.45	0.32
car_jump	**0.61**	**0.49**	0.38	0.22	0.02	0.01	0.40	0.25
chrome	0.33	**0.20**	**0.34**	0.15	0	0	0.24	0.11
dominoes	**0.65**	**0.66**	0.56	0.44	0.35	0.21	0.48	0.30
freight_train	0.72	**0.66**	0.68	0.49	**0.74**	0.57	0.38	0.21
frozen_lake	0.37	0.22	**0.52**	**0.37**	0.01	0	0.32	0.17
gray_squirrel	**0.75**	**0.64**	0.64	0.51	0.72	0.54	0.32	0.15
guitar	0.64	0.63	**0.73**	**0.67**	0.08	0.04	0.41	0.26
hummingbird	0.71	0.59	**0.76**	0.63	0.22	0.12	0.22	0.12
juggling	**0.90**	**0.86**	0.78	0.68	0	0	0.44	0.36
jungle_cat	**0.63**	**0.56**	0.50	0.47	0.02	0.01	0.37	0.28
kangaroo_fighting	0.49	**0.39**	**0.55**	0.36	0.44	0.27	0.33	0.22
kia_commercial	**0.61**	**0.54**	0.49	0.48	0.29	0.16	0.37	0.24
knot	**0.77**	**0.68**	0.75	0.66	0.06	0.03	0.43	0.25
koala	0.49	**0.47**	**0.52**	0.42	0.44	0.26	0.37	0.20
lion	0.12	0.05	0.21	0.07	0	0	**0.30**	**0.11**
palm_tree	**0.71**	**0.53**	0.50	0.33	0	0	0.20	0.09
penguins	**0.60**	**0.59**	0.37	0.37	0.25	0.14	0.41	0.31
pepsis_wasps	**0.42**	**0.22**	0.31	0.12	0	0	**0.42**	0.19
planet_earth_one	**0.74**	**0.69**	0.47	0.31	0.65	0.48	0.32	0.18
pouring_tea	**0.74**	**0.67**	0.43	0.28	0.48	0.30	0.28	0.15
rock_climbing	0.43	**0.29**	0.40	0.2	0	0	**0.44**	0.25
rock_climbingtwo	**0.71**	**0.60**	0.55	0.41	0.15	0.08	0.32	0.18
roller_coaster	**0.39**	**0.31**	0.36	0.19	0	0	0.11	0.05
rolling_pin	**0.78**	**0.69**	0.44	0.33	0.72	0.54	0.49	0.30
salsa	0.36	0.24	**0.45**	**0.26**	0.09	0.04	0.32	0.21
shark_attack	**0.57**	**0.39**	0.47	0.31	0	0	0.33	0.18
sitting_dog	**0.67**	**0.60**	0.49	0.38	0.64	0.45	0.39	0.28
sled_dog_race	0.69	0.54	**0.75**	**0.62**	0	0	0.32	0.14
snow_leopards	**0.83**	**0.71**	0.64	0.48	0.73	0.55	0.35	0.23
street_food	**0.87**	**0.82**	0.52	0.49	0.70	0.53	0.46	0.29
tennis_tr	0.37	0.23	**0.53**	**0.34**	0	0	0.20	0.09
trampoline	**0.84**	**0.73**	0.53	0.33	0.70	0.54	0.19	0.10
up_dug	**0.83**	**0.74**	0.63	0.53	0.20	0.10	0.34	0.19
white_tiger	**0.60**	**0.52**	0.55	0.43	0.41	0.26	0.22	0.11
zoo	**0.70**	**0.65**	0.56	0.51	0.47	0.26	0.52	0.34

4.3.1 Ground Truth Adjustment

An evaluation dataset consists of a set of videos and a ground truth labeling for each video. The score of an evaluated approach measures how well its segmentation results match the ground truth. This traditional evaluation method has major shortcomings manifested in ground truths associated with videos. Ground truths are hand-labeled by people, and people have different opinions about what is a correct segmentation for a video. Consider, for example, a scene of a person walking in a park where there are some trees behind him/her. One might think that the correct segmentation consists of the person, some trees, the ground, and the sky. Another one might think that the correct segmentation is the person, and everything else is the background. Consider another example of a scene where a person is walking his/her dog in the park. Disregarding the background, one might think that the dog's leash is a different object than the dog, while another one will consider the dog and the leash as a single object. The person itself can be thought of as a single object, or multiple objects: shirt, pants, shoes, etc. Different people have different opinions about what is the the "correct" segmentation of a video, and the level of details it should include. There is no right or wrong answer.

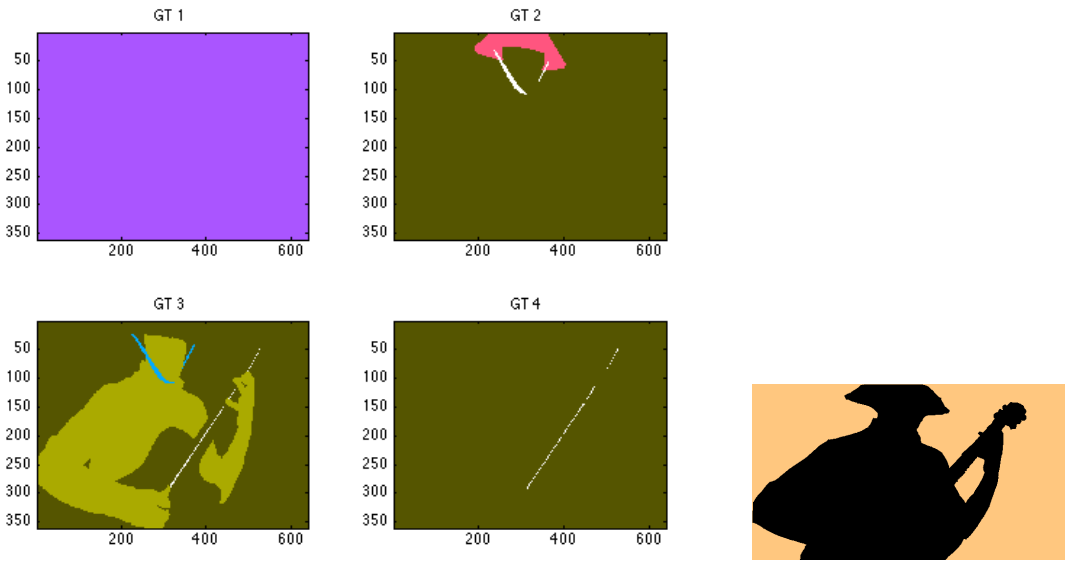
RAMS is a motion-based segmentation approach. Its goal is to segment every moving object as a whole. Some objects move rigidly, while others move non-rigidly. Non-rigid objects impose problems for traditional rigid motion segmentation methods because they have different parts that move in different directions. The hypothesis of this work is that the parts of a non-rigid object move at the same time, even if they move in different directions. So the proposed approach uses temporal coherence of motion for segmenting objects in videos. If spatially neighboring regions look alike and move temporally together, then they are likely to belong to the same object. This solves the problem of segmenting non-rigid objects. For example, people and animals have limbs that move in different directions. However, a person or an animal is a single object. Since a non-rigid object's parts move at the same time, using

motion temporal coherence can put the parts together in a single segment. In addition to non-rigid motion, an object can move rigidly, for example a flying airplane. A rigid object moves as a whole in one direction, and all its parts are also moving temporally together. So temporal coherence of motion is sufficient for segmenting moving objects in videos. Thus, the goal is segmenting every moving object as a whole, without breaking them down into different parts. All non-moving objects should be combined in a single background segment.

Some ground truth segmentations in the VSB100 [9] dataset do not fit with the segmentation goal above; they either have less or more details, as shown below. We therefore evaluate segmentation algorithms using both the original ground truth and ground truth segmentations adjusted to match the expected outcome for a true evaluation of the approach. This creates a second evaluation of the VSB100 [9] dataset, and its results are shown alongside the first evaluation (using the given ground truth) in the previous subsection.

The VSB100 [9] dataset provides up to four ground truth segmentations per video. The adjusted version is created by either selecting one of the provided four segmentations, or adjusting one of the provided segmentations by merging or splitting regions. By looking at the differences between the scores of RAMS per video, reported in Tables 4.5 and 4.6, the difference is insignificant for most videos. However, a few videos have significant difference.

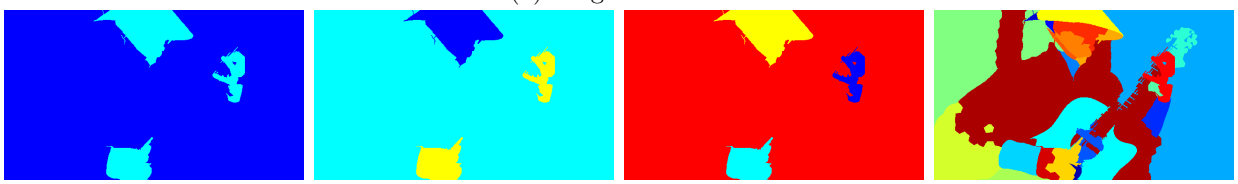
One such video is called "guitar"; it contains a man playing a guitar as shown in Figure 4.13. The movement is mainly in his hands with a little nodding from his head. The given ground truth segmentations are shown in Figure 4.12. There are three given ground truth segmentation for this video, and they are all very different. None of them capture the guitarist as a single segment. Semantically, there are two objects: the man and the guitar. However, in respect to motion, they are considered a single object since the man carries the guitar the whole time. Thus the new adjusted ground truth contains only one segment for the whole man and guitar, in addition to the background segment, as shown in Figure 4.12. The increase in OSS scores after adjusting the ground truth is 0.16. Figure 4.13 shows some segmentation results for this video by RAMS.



(a) Given GT (b) Adjusted GT

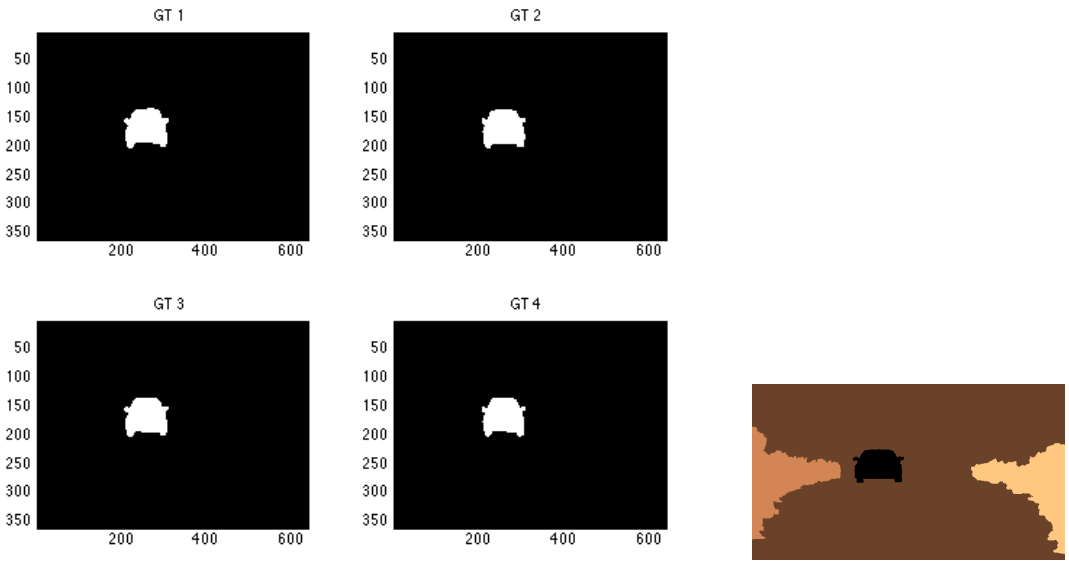
Figure 4.12: The given and adjusted ground truth for the "guitar" video of the VSB100 [9] dataset. The given ground truth segmentations GT 2, GT 3, and GT 4 contain 3, 4, and 2 segments, respectively.

(a) Original Frame



(b) Level 1 Segmentation (c) Level 2 Segmentation (d) Level 3 Segmentation (e) Level 22 Segmentation

Figure 4.13: The original first frame of the "guitar" video of the VSB100 [9] dataset, and RAMS segmentation result for this video. The levels shown are 1, 2, 3, and 22. In this video, a man is playing a guitar. The movement is mainly in his hands with a little nodding from his head. So it is correct for RAMS to segment the hands and head in the first levels. Since the torso and guitar were not moving, they were segmented differently in later levels, and that is expected.

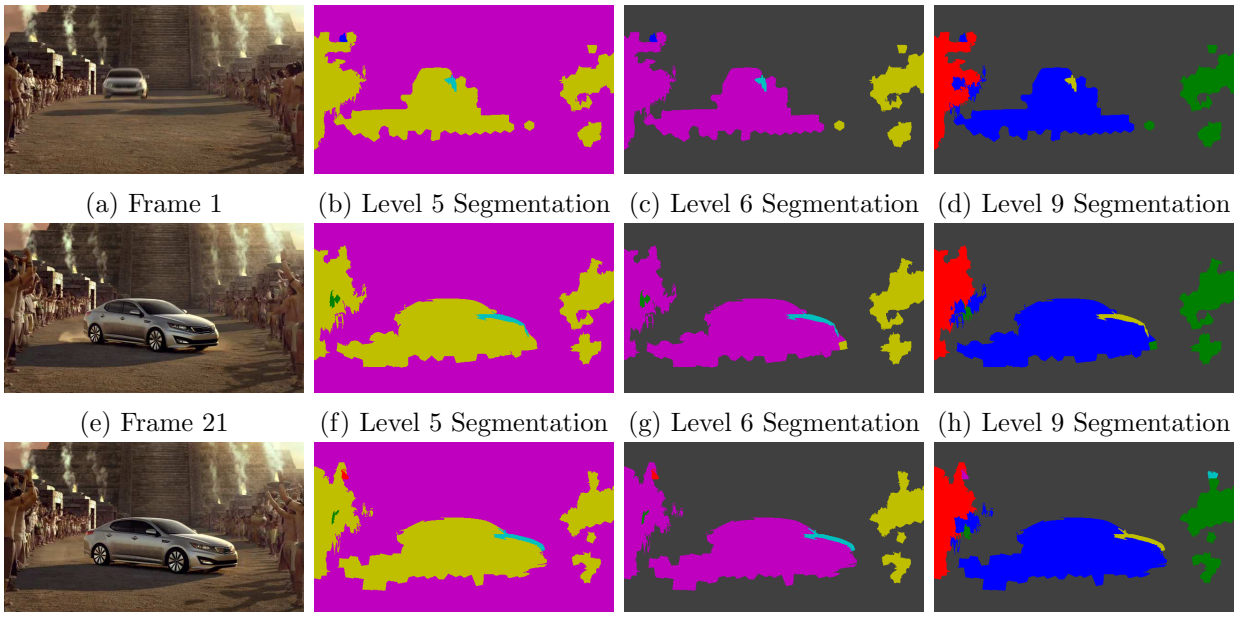


(a) Given GT (b) Adjusted GT

Figure 4.14: The given and adjusted ground truth for the "kia_commercial" video of the VSB100 [9] dataset.

A second video is called "kia_commercial". From its name, it is a commercial for a car where the car lands on the ground and slightly turns to the right of the viewer, with cheering crowds on the right and left. Some frames of this video are shown in Figure 4.15. Although the crowd is cheering by moving and waving their hands, all the given ground truth segmentation for this video contain a single a segment for the car, in addition to the background segment, as shown in Figure 4.14. However, the two crowds, on the right and left of the car, are moving and separate, so the new adjusted ground truth contains two additional segments for the two crowds, as shown in Figure 4.14. Figure 4.15 shows RAMS segmentation for this video.

After adjusting the ground truth for the "kia_commercial" video, the OSS score surprisingly decreased from F(R 0.73, P 0.76) = 0.74 to F(R 0.71, P 0.54) = 0.61. The recall decreased insignificantly by 0.02, but the precision decreased by 0.22. By visually looking at the segmentation result, as shown in Figure 4.15, the ninth level is the lowest level that separates the car and the two crowds, and it seems to be closer to the adjusted ground truth



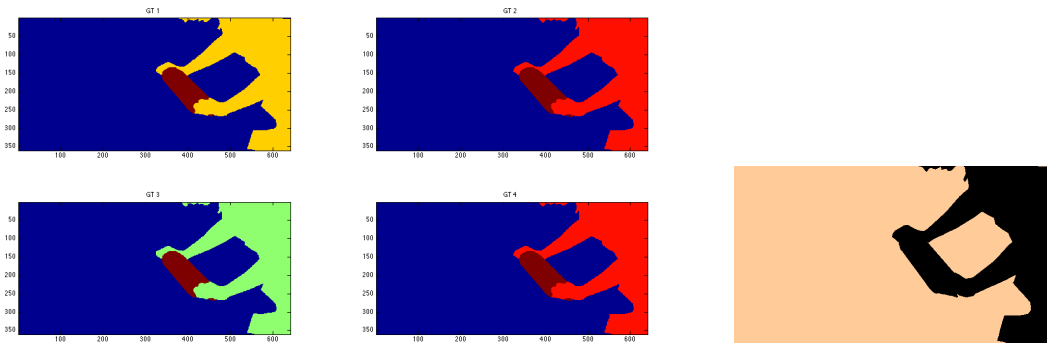
(a) Frame 1 (b) Level 5 Segmentation (c) Level 6 Segmentation (d) Level 9 Segmentation

(e) Frame 21 (f) Level 5 Segmentation (g) Level 6 Segmentation (h) Level 9 Segmentation

(i) Frame 41 (j) Level 5 Segmentation (k) Level 6 Segmentation (l) Level 9 Segmentation

Figure 4.15: The original frames of the "kia_commercial" video of the VSB100 [9] dataset, and RAMS segmentation result for this video. The levels shown are 5, 6, and 9. The car segment is spilling on the background in these levels, and that is normal since the shadow seems to move with car. The crowd segments are not complete, and that is because the motion of the people there is mainly in their arms and hand as they raise them up.

than the given ground truth. The car segment is spilling a little on the background, especially in the first frame, so this affects this segment precision, but this should also affect the score when given ground truth is used. By investigating the ninth level scores when using the adjusted ground truth, the reported score is F(R 0.82, P 0.25) = 0.38. Recall that precision measures how well a segment from the result is fitted within a ground truth segment. So each segment from the result is matched with a ground truth region with maximal pixel overlap, and then its precision is based on the number of pixels in their intersection. By manually calculating the precision (as in Equation 4.1) of the main segments in the ninth level segmentation: the car, the left crowd, the right crowd, and the background, the precisions were 48%, 83%, 72%, and 89%, respectively. The precision for all these segments collectively is 82%, which is very different from the reported precision for this level. However, after normalizing the precision score, as in Equation 4.3, by subtracting the number of pixels in the ground



(a) Given GT (b) Adjusted GT

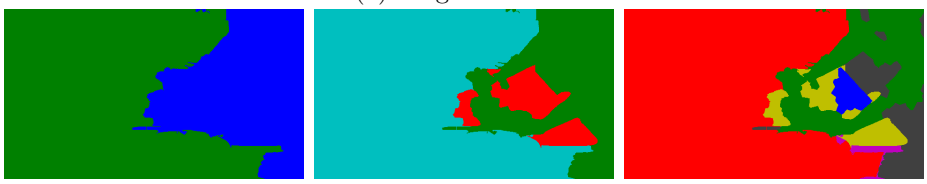
Figure 4.16: The given and adjusted ground truth for the "rolling_pin" video of the VSB100 [9] dataset.

truth background segment, the normalized precision is 0.23. Note that I did not include the small regions in this calculation, and thus the small difference from the reported score. So the normalization process of VSB100 [9] is what is affecting the scores for this video.

A third video called "rolling_pin" contains a woman who is rolling a pin on a table, as shown in Figure 4.17. All the given ground truth for this video is composed of a segment for the woman, a segment for the pin, and a segment for the background. However, the evaluation is for motion segmentation, and in respect to motion, the woman and the pin are moving together and the pin was in her hands all the time, so the pin is an extension of her. Thus the new adjusted ground truth merges the pin and woman segments together. The given and adjusted ground truth for this video shown in Figure 4.16. Figure 4.17 shows RAMS segmentation result for this video, and the increase in OSS scores after adjusting the ground truth is 0.05.

A fourth video is called "sled_dog_race", and it is a video of a pack of dogs pulling a sled with a person on it. The dogs in the pack have synchronized motion, spatially close to each other, and similar in color. To hand-label this video, one person may segment each dog independently, while another one may segment the dog pack together. Although there is no right or wrong answer for this video, the dogs in the team are co-temporally moving

(a) Original Frame

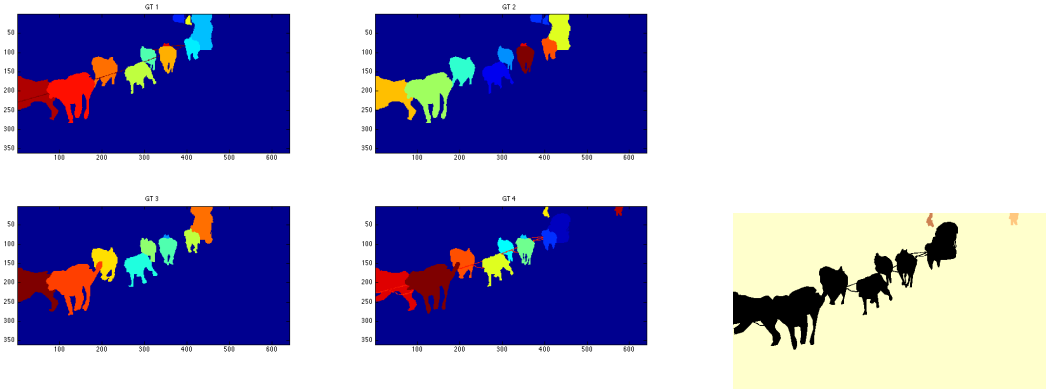


(b) Level 1 Segmentation (c) Level 3 Segmentation (d) Level 6 Segmentation

Figure 4.17: The original first frame of the "rolling_pin" video of the VSB100 [9] dataset, and RAMS segmentation result for this video. The levels shown are 1, 3, and 6.

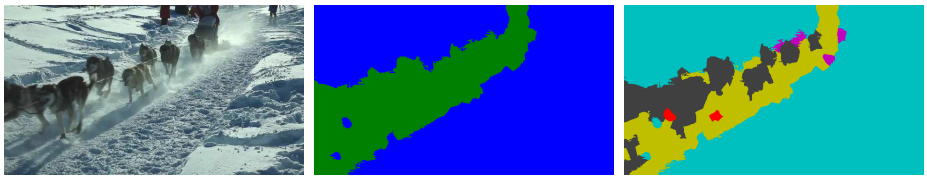
together, thus segmenting them together is the adopted answer for this approach. However, the given ground truth segmentations for this video all include the first answer; they separate each dog in a separate segment. So the adjusted ground truth contains a single segment for the pack. In addition, the sled that the dogs are pulling is inherently moving with them, so it is also merged with the dogs pack segment. Figure 4.18 shows the given and adjusted ground truth for this video. Figure 4.19 shows RAMS segmentation result for this video. The difference in OSS scores after adjusting the ground truth is 0.37.

The fifth and last video, called "snow_leopards", contains a leopard moving slightly from right to left on what seems to be a rocky mountain. The only moving object is the the leopard. However, the given ground truth breaks up the background by putting the stationary rocks in front of the leopard in separate segments. The rocks are not moving, and thus should be merged in a single background segment. This is done in the adjusted ground truth, which is shown with the given ground truth in Figure 4.20. RAMS segmentation result for this video is shown in Figure 4.21. The increase in OSS scores after adjusting the ground truth is 0.27.



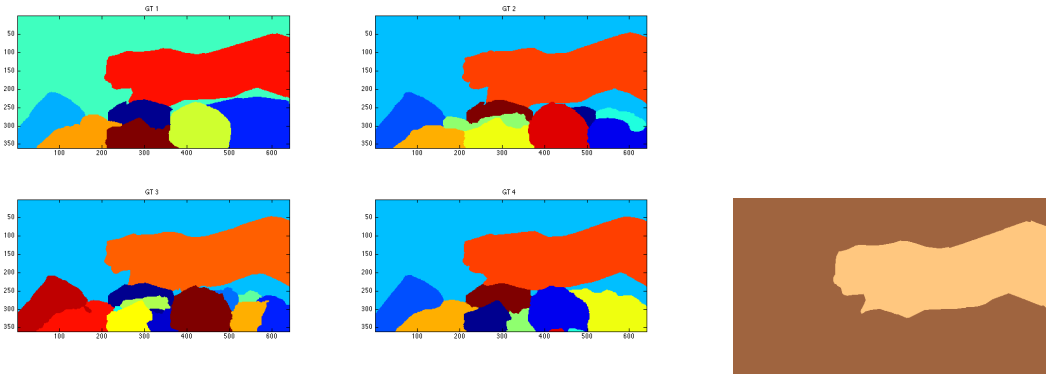
(a) Given GT (b) Adjusted GT

Figure 4.18: The given and adjusted ground truth for the "sled_dog_race" video of the VSB100 [9] dataset.



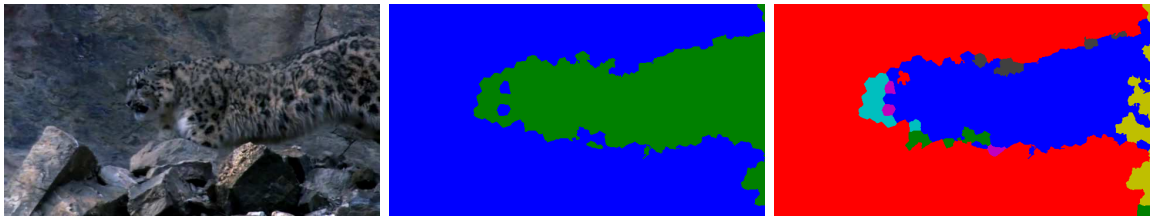
(a) Original Frame (b) Level 1 Segmentation (c) Level 6 Segmentation

Figure 4.19: An original frame of the "sled_dog_race" video of the VSB100 [9] dataset, and RAMS segmentation result for this video. The levels shown are 1 and 6.



(a) Given GT (b) Adjusted GT

Figure 4.20: The given and adjusted ground truth for the "snow_leopards" video of the VSB100 [9] dataset.



(a) Original Frame (b) Level 1 Segmentation (c) Level 9 Segmentation

Figure 4.21: The original first frame of the "snow_leopards" video of the VSB100 [9] dataset, and RAMS segmentation result for this video. The levels shown are 1 and 9.

4.3.2 Parameter Setting

The three similarity weights α_1, α_2, and α_3 used to compute the final similarities in Equation 3.1 were set to (0.4, 0.4, 0.2) respectively for all the videos in this evaluation, since this combination achieved the maximum score on the VSB100 [9] dataset. The contour plot of scores for all possible weight combinations is shown in Figure 4.22. From this plot, the large plateau (white region) is the weight space where high scores are achieved. This indicates that RAMS is not highly sensitive to these parameter values, as long as there is a balance of motion and color weights that is equal or higher than the spatial weight. Since this search was done to find the best weight combination for RAMS to achieve its segmentation goals, the adjusted ground truth is used in this evaluation. In addtion, this evaluation uses the first 41 coarsest levels of each segmentation.

The above weight search was done for the VSB100 [9] dataset as a whole. In addition, a weight search was done for each of its videos individually to see what is the best weight combination for each video. A histogram was computed to collect the votes of the individual videos about their best weight combination that achieved the highest score. A contour plot for this histogram is shown in Figure 4.23. From this histogram, it is clear that the videos do not agree on a weight combination; the different videos perform best with different weights. That does not mean that they cannot perform well with other weights. However, collectively, the weight combination (0.4, 0.4, 0.2) achieves better scores from the above weight search.

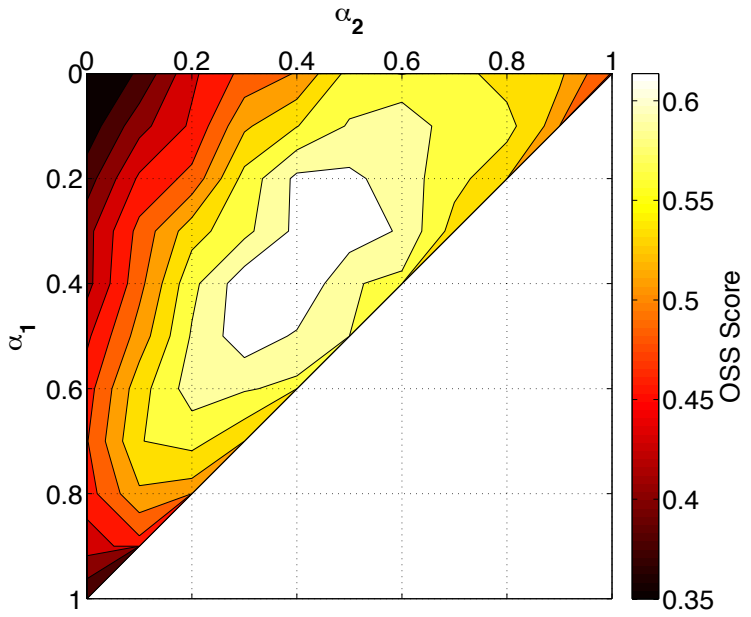


Figure 4.22: A contour plot of OSS scores for RAMS using all possible weight combinations (α_1, α_2, α_3), where increments per each weight is 0.1. The shown weights are α_1 and α_2, while α_3 is implicitly equal to:
$1.0 - \alpha_1 - \alpha_2$. Since all weights sum to 1.0, only the top left triangle contains the possible weight combinations.

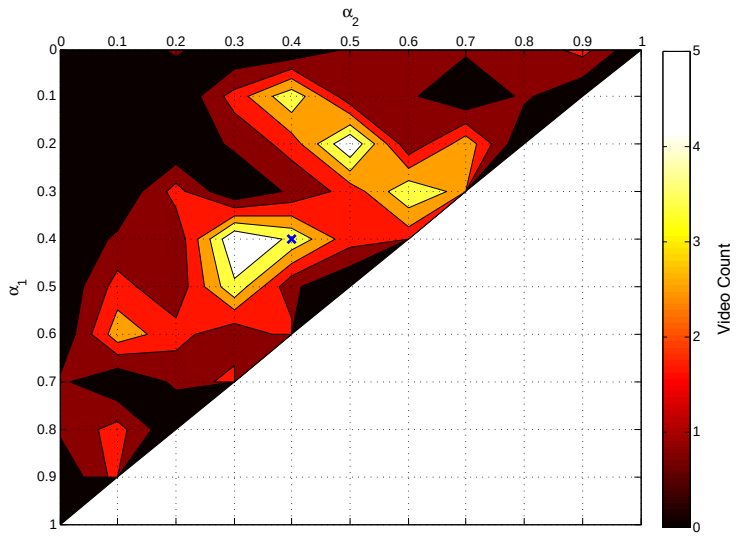


Figure 4.23: A contour plot of a histogram that collects the votes of the individual videos about their best weight combination that achieved the highest score. A blue cross mark is placed on the bin of (0.4, 0.4, 0.2) weight combination.

These similarity weights are used in both the clustering step and the post-processing step of RAMS. The clustering step is when the superpixel trajectories are grouped, while the post-processing step is when discarded trajectories are merged with their most similar neighbors. The same weights are used in both steps during the weight search. However, in the evaluation reported above, the post-processing weights were equally set. This did not affect the scores significantly. In addition, the discarded trajectories are problematic trajectories; they are either less than three frames long, or have very small superpixels. Thus using equal weights for merging them with similar neighbors is probably better.

4.4 Cases of Unexpected Results

There are some case when the results are not as expected by a user of a video segmentation algorithm. The first case is when a moving object has non-moving parts, such as for example a person or an animal that only moves his/her/its head. Any motion-based algorithm will not detect the non-moving parts. In this case, its up to the other visual cues, such as color and location, to help segmenting these parts. However, this could end up dividing the object into more than one segment. In addition, these parts usually appear in levels with larger number of clusters, as they tend to merge with the background in levels with smaller number of clusters. Figure 4.24 illustrates an example.

Another case where unexpected segments may appear involves occlusions. An example is a wall that gets occluded when an object passes in front of it. The occluded or dis-occluded areas of the video may get segmented with the occluding object, especially in the levels with small number of clusters as they can get separated in levels with larger number of clusters. Figure 4.25 illustrates an example. The reason causing this problem is that these occluded or dis-occluded areas usually end up having short trajectories as they are terminated with an occlusion or appear just after an occlusion. In addition, the superpixels on the boundary of the moving occluding object may not be accurate, spilling some pixels on these areas that

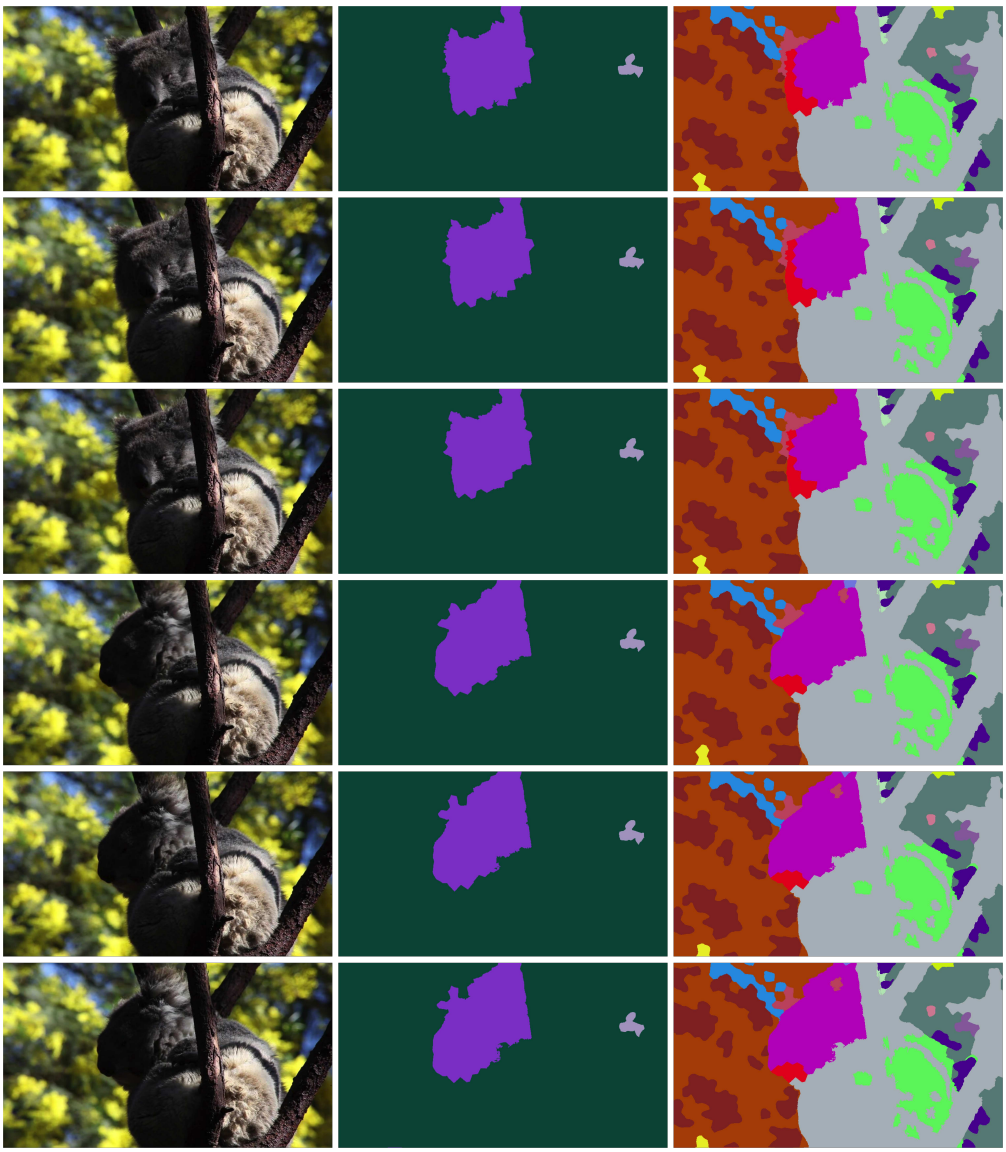


Figure 4.24: RAMS segementation result for the "koala" video of the VSB100 [9] dataset. The frames 1, 21, 41, 61, 81, and 101 are shown in consecutive rows. The columns contains the original frames, level 2 segmentation, and level 16 segmentation, respectively. Only the head of the koala is moving, so it is segmented in the first level shown. The rest of the koala's body did not move, so it got separated from the background in a later level.

have the same motion as the occluding object. So for the example shown in Figure 4.25, changing the weights of the motion, color, and spatial similarities, did change the effect of occlusion.

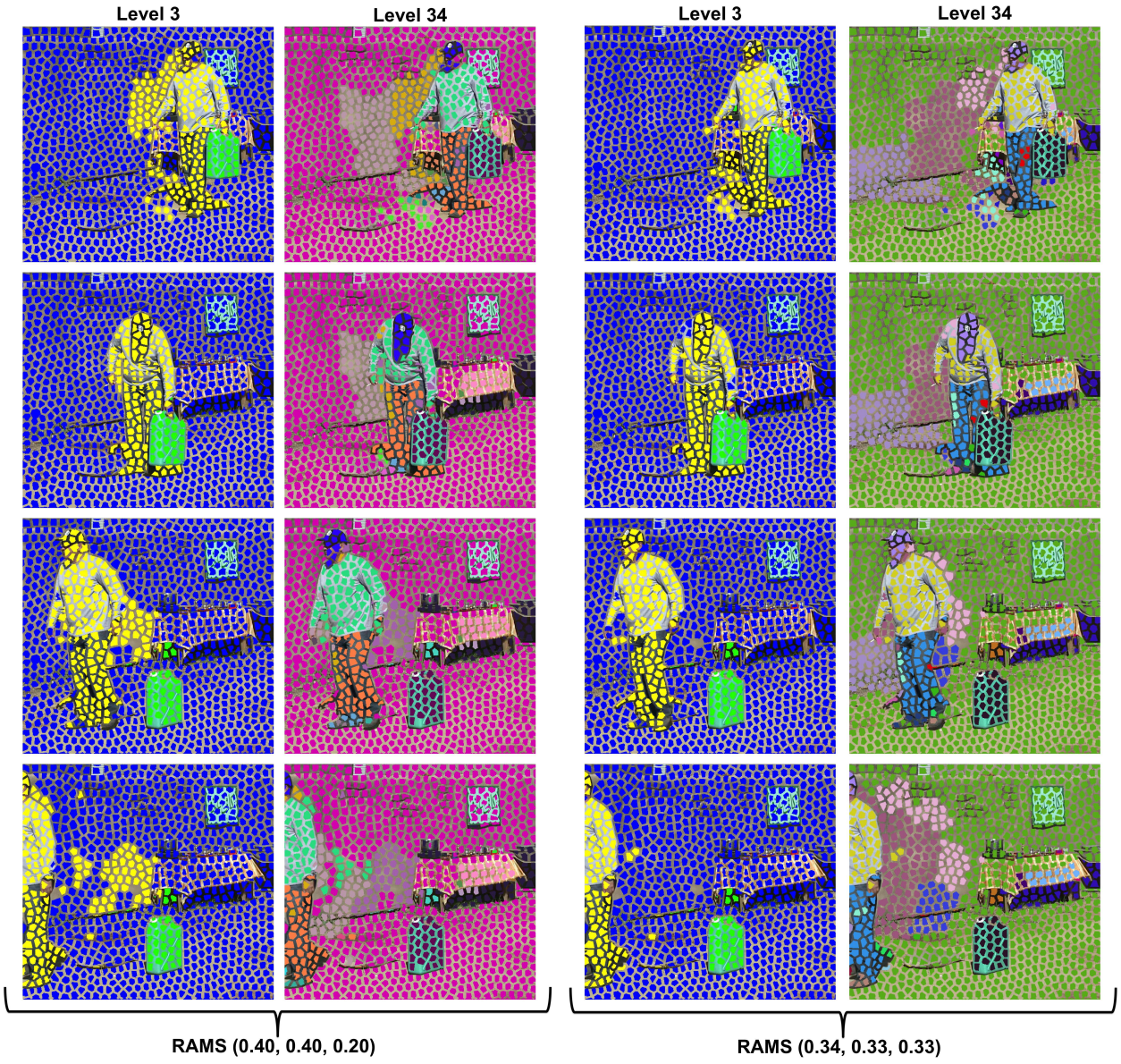


Figure 4.25: Different levels of the RAMS segmentation result of the man and gas can video, with different weight combinations. The frames 1, 36, 84, and 102 are shown in consecutive rows.This example shows the effect of occlusion on the wall behind the man as he passes in front of it. Merging the occluded wall area with the man happens at levels with small number of clusters. The occlusion effect was lessened by using a different weight combination that decreases the motion weight. It merged the occluded area of the wall with the background instead of merging it with the man segment.

4.5 Additional Discussion

RAMS outperforms state-of-the-art approaches on new videos that emphasize non-rigid motion, and the more traditional VSB100 [9] dataset as well. Although BPR is not as important as VPR for RAMS, RAMS also achieves good results with BPR. RAMS was designed for non-rigid motion, as in the zebra video example, but also works for rigid objects. The majority of objects of interest in real-world videos are non-rigid objects. In the tested videos of VSB100 [9], the approximate total number of main moving objects is 164, where 28 of them are rigid objects, and 136 of them are non-rigid objects. The rigid objects include three balls, two cars, a boat, and a train. However this video set contains more than 73 people and 62 animals.

An additional advantage of RAMS is its ability to put the background together in a single segment. Other approaches, such as [16] and [7], break up the background because of their color sensitivity. However, the background is still, thus a motion segmentation approach should put the background pieces together in a single segment.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

A new unsupervised motion-based video segmentation approach (RAMS) is proposed in this dissertation. It segments non-rigid objects based on temporal motion coherence, instead of coherence in motion magnitude and direction. Considering motion timing is a new perspective for looking at motion coherence between objects in videos. The experiments show significant improvement in video segmentation results, compared with related state-of-the-art approaches. These results prove that this new concept in motion coherence works for video segmentation.

In most videos, the objects of interest are people and animals, which move non-rigidly. Non-rigid objects have parts that can move in different directions at the same time. On the other hand, parts of a rigid object move at the same time and direction. So using temporal coherence of motion is suitable for segmenting all objects, whether they move rigidly or non-rigidly. Other approaches assume rigid motion, and thus can break an object into multiple segments. RAMS drops this assumption, and outperforms the other approaches.

RAMS hierarchically segments videos, producing multiple segmentations with different number of clusters. Defining what is a "correct" segmentation is a matter of opinion. So hierarchal segmentation is an advantage that leaves the selection of the optimal level to the end-user or end-system. The approaches [16] and [7] also produce hierarchal segmentations, but they tend to break up a stationary background into multiple segments even in their coarsest levels. RAMS, as opposed to these approaches, is able to put the background pieces together in a single segment.

RAMS sets parameters (α_1, α_2, α_3) to weigh the contribution of three video cues: motion, color, and location, respectively. In experiments, RAMS was able to perform relatively well under a wide range of weights. So RAMS is not highly sensitive to these parameters setting.

Other than the fixed camera assumption, RAMS is a general-purpose automatic video segmentation system. It makes no assumptions about the video contents. Its segmentations can be further used as a basis for object recognition or action recognition. Instead of looking at whole images or bounding boxes, segments provide more localized and freely-shaped regions for object recognition, and multi-frame regions provide time information. Furthermore, the motion of superpixel trajectories within a segment can also be analyzed for action recognition. The work in [35] by Ke et al. is an example that uses video segments in action recognition.

Other applications include the generation of automatic segmentations to save some human efforts. It simplifies video semantic labeling as a user can label a segment once, and then all labels are propagated to all frames. The annotation tool of [17] is an example of this application. In this context, tools can also be provided for correcting misclassified superpixel trajectories, and the user only needs to correct one superpixel to correct its whole trajectory.

5.2 Future Work

The biggest limitation of RAMS is that it assumes a fixed camera for detecting motion. Future work can use a motion detection algorithm to determine whether a superpixel is moving independently from the camera. Note that RAMS only needs to know whether a superpixel is moving, not how much nor in what direction. A future work will add independent motion detection to RAMS, in order to extend the domain of application to videos from moving cameras, while maintaining the principal of unsupervised segmentation through temporal motion coherence. However, this is not easy; motion detection algorithms can have assumptions about video content, or their inaccuracy affects the inferred motion timing, which in turn affects the segmentation results.

The video stabilization algorithm by Dutta et al. [28], uses a set of tracked points in every frame, segments them as background or foreground points, and uses the background points to estimate the transformation from one frame to the next. As a pilot study for adapting RAMS to camera motion, we use these points' foreground/background segmentation to estimate motion within superpixels without the need to transform or stabilize the frames; foreground points are moving, while background points are still. Rahul Dutta generously provided us with the foreground/background segmentation of points for two videos from the VSB100 [9] dataset with camera motion, namely the "hockey" and "space_shuttle" videos.

One issue is that points do not cover the entire image, so some superpixels do not have points within them to determine their motion. Table 5.1 contains the number of superpixels that do not have points within them at all, so they do constitute a big and important portion of the video and thus discarding them is not a good option. Thus another modified version of RAMS was implemented to take this problem into account. In this version, the motion probability for a superpixel is computed as the fraction of the points within it that are labeled as foreground. If the superpixel does not have any points within it, then its motion probability is undefined. These motion probabilities are what creates the motion

Table 5.1: The number of superpixels that do not have points (generated by [28]) within them, in the "hockey" and "space_shuttle" videos of VSB100 [9] dataset.

	"hockey"	**"space_shuttle"**
Total number of superpixels	78,254	82,437
Number of superpixels without any points	14,551	27,868
Approximate percentage	19%	34%

patterns for each superpixel trajectory. Then, comparing two superpixel trajectories in their overlapping time is done as usual. The exception is when two corresponding superpixels in a frame have undefined motion, both of them or either one of them, this frame is skipped during comparison. If two trajectories do not have any common frame with defined motion, their motion similarity is undefined and thus it is not used in the final similarity between the two trajectories. In this case, the weight of motion similarity is redistributed to the weights for the color and spatial similarities. For example, if the weights are (0.34, 0.33, 0.33) for motion, color, and spatial similarities respectively, then the weights become (0, 0.5, 0.5) for this pair of trajectories.

The "hockey" video is hard since the majority of frames in this video are dominated by foreground objects, and that is against the video stabilization algorithm assumption of the background being the majority of the frames. Figure 5.1 shows some frames of this video. However, although the video mainly consists of people, it could be sufficient to just distinguish between them regardless of whether they are considered foreground or background as long as adjacent people have different labels. Figure 5.1 shows the points and superpixel boundaries overlaid on the original images, and the point segmentation was able to differentiate between the main people in most frames. So an attempt to use this point segmentation was performed. The result of the modified version of RAMS for the "hockey" video is shown in Figure 5.1, and its VPR scores using the given ground truth shown in Figure 5.2 are shown in Table 5.2. RAMS was able to separate the main people in the visual result of this video.

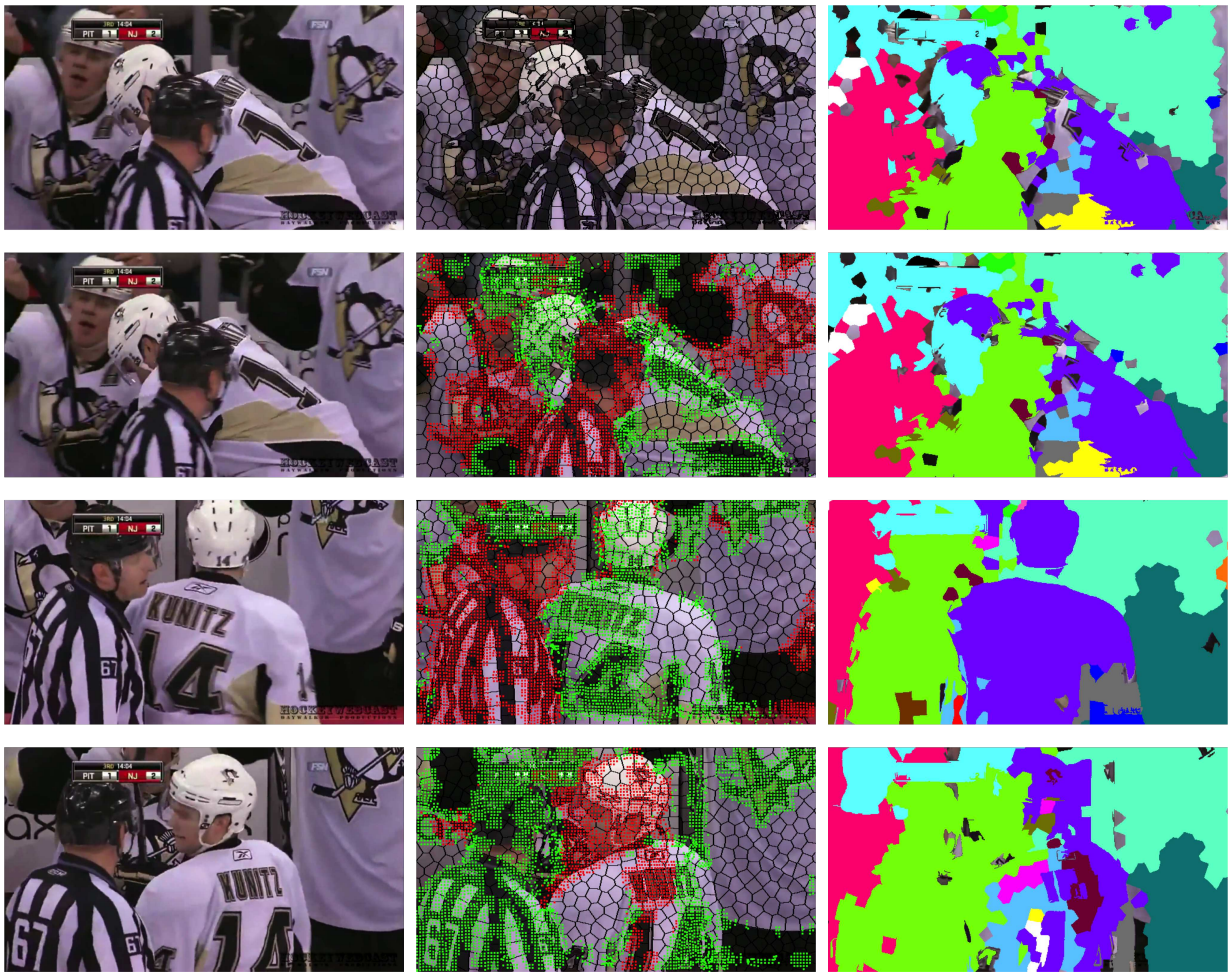
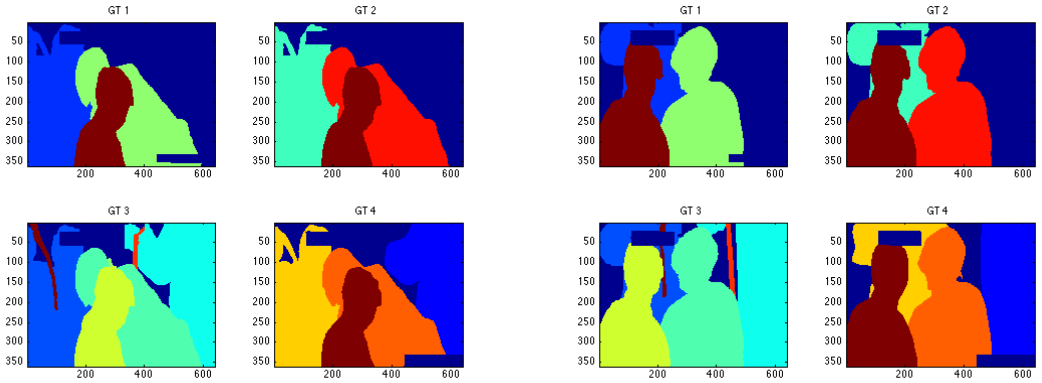


Figure 5.1: Original frames for the "hockey" video of VSB100 [9] dataset, and RAMS (0.34, 0.33, 0.33) modified version segmentation result for it. Frames 1, 2, 61, and 121 are shown in consecutive rows. The first column contains the original frame, while the second column overlays the superpixel boundaries and the points foreground/background segmentation provided by Rahul Dutta [28]. Red points correspond to the background segment, while green points correspond to the foreground segment. The points are thickened to be more visible. There is no point segmentation for the first frame by [28]. The third column contains a segmentation result from the modified version of RAMS (level 19). The segmentation result shown overlays the labeled superpixels (before post-processing) on the original images; labeled superpixels are colored according to their labels, while discarded superpixels retain their original colors.



(a) Given ground truth for the first frame.

(b) Given ground truth for the last frame

Figure 5.2: The given ground truth for the "hockey" video of the VSB100 [9] dataset.

Table 5.2: The VPR scores for the modified RAMS (0.34, 0.33, 0.33) version result on the "hockey" video of the VSB100 [9] dataset. In addition, the scores of Grundmann et al. [16, 17], and Ochs and Brox [25] are included for comparison.

Approach	**ODS**	**OSS**	**AP**
Modified RAMS	**0.68** (R 0.66, P 0.70)	**0.68** (R 0.66, P 0.70)	**0.64**
Grundmann et al. [16]	0.49 (R 0.39, P 0.67)	0.49 (R 0.39, P 0.67)	0.41
Ochs and Brox [25]	0.06 (R 0.98, P 0.03)	0.06 (R 0.98, P 0.03)	0.03

Table 5.3: The VPR scores for the modified RAMS (0.34, 0.33, 0.33) version result on the "space_shuttle" video of the VSB100 [9] dataset. In addition, the scores of Grundmann et al. [16, 17], and Ochs and Brox [25] are included for comparison. Score are in the format: F(R,P).

	Using Given Ground Truth			**Using Adjusted Ground Truth**		
Approach	**ODS**	**OSS**	**AP**	**ODS**	**OSS**	**AP**
Modified RAMS	0.26 (0.23, 0.29)	0.26 (0.23, 0.29)	0.16	0.41 (0.31, 0.62)	0.41 (0.31, 0.62)	0.26
Modified RAMS - Boundary Discarded	0.31 (0.39, 0.26)	0.31 (0.39, 0.26)	**0.21**	**0.57** (0.85, 0.43)	**0.57** (0.85, 0.43)	**0.51**
Grundmann et al. [16]	**0.35** (0.23, 0.78)	**0.35** (0.23, 0.78)	0.20	0.38 (0.27, 0.64)	0.38 (0.27, 0.64)	0.26
Ochs and Brox [25]	0 (0.74, 0.00)	0 (0.74, 0.00)	0	0 (0.77, 0.00)	0 (0.77, 0.00)	0

The second video, called "space_shuttle", is about a launch of a space shuttle. Figure 5.3 shows some frames and the point segmentation for this video. This video is hard because the camera zooms as well as pans. In addition, it has black strips on the borders of the image frames. Figure 5.3 shows a segmentation result from the modified version of RAMS in its third column. There were some points that were falsely labeled as foreground, especially in the black borders of the images. This lead to the spill of the space shuttle segment to the background. So another segmentation was attempted after discarding the superpixels on the image boundaries. This lessened the spill of the space shuttle segment to the background. Table 5.3 contains the VPR evaluation scores for the two results (with and without discarding boundary superpixels) using the given and adjusted ground truth shown in Figure 5.4.

The results of the "hockey" and "space_shuttle" videos were a little encouraging, so the modified RAMS version was tested on five of the previously tested videos of VSB100 [9] dataset: "dominoes", "juggling", "kia_commercial", "trampoline", and "up_dug". However, the scores for these videos significantly decreased. So it is clear that this version of RAMS is sensitive to mislabeling in the points foreground/background segmentation by [28]. As a result, future work will test other motion detection algorithms to find a more suitable one.

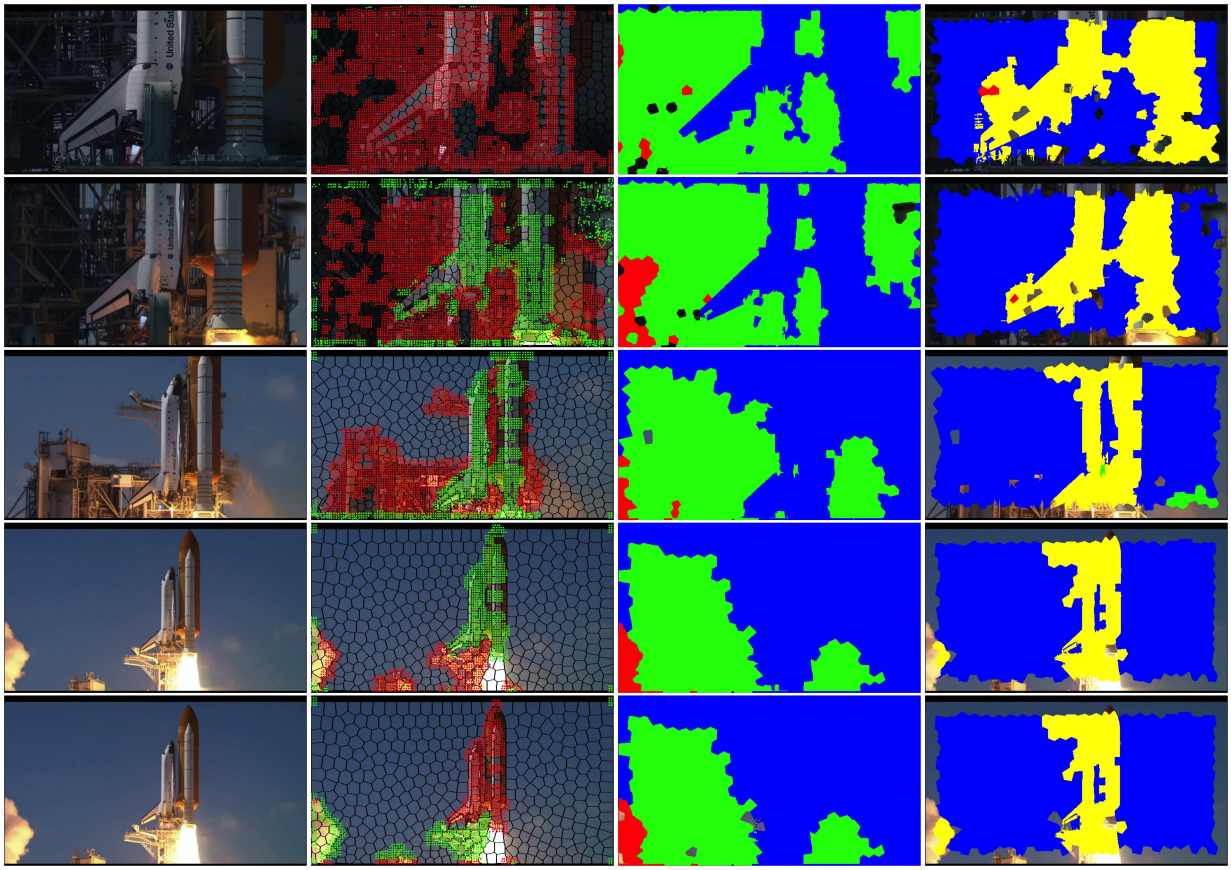
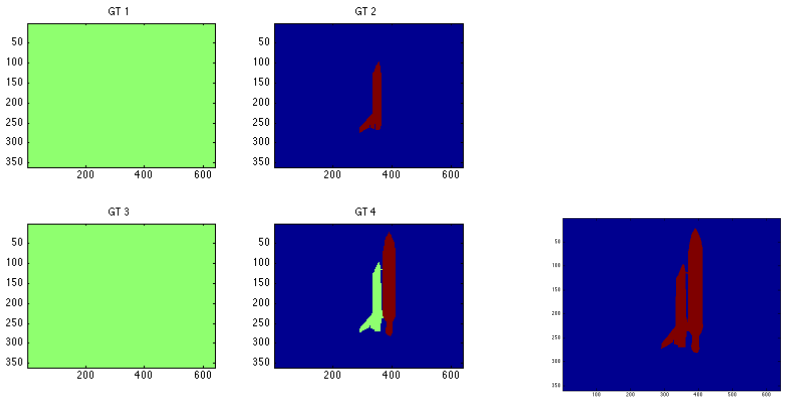


Figure 5.3: Original frames for the "space_shuttle" video of VSB100 [9] dataset, and RAMS (0.34, 0.33, 0.33) modified version segmentation result for it. Frames 2, 25, 71, 118, and 121 are shown in consecutive rows. The first column contains the original frame, while the second column overlays the superpixel boundaries and the points foreground/background segmentation provided by Rahul Dutta [28]. Red points correspond to the background segment, while green points correspond to the foreground segment. The points are thickened to be more visible. The third column contains a segmentation result from the modified version of RAMS (level 2). The fourth column contains a segmentation result from the modified version of RAMS (level 3) after discarding the boundary superpixels. The segmentation results shown overlays the labeled superpixels (before post-processing) on the original images; labeled superpixels are colored according to their labels, while discarded superpixels retain their original colors.



(a) Given ground truth for the last frame. (b) Adjusted ground truth for the last frame

Figure 5.4: The given and adjusted ground truth for the "space_shuttle" video of the VSB100 [9] dataset. The adjusted ground truth merges the two segments of the space shuttle into one since they are parts of the same object and are moving together in the video.

Conceptually, it is simple to adapt to moving cameras by changing the motion detection method. RAMS only needs to know when a superpixel is moving. This can be accomplished by detecting the camera's motion and then the independent motions. Examples of algorithms for this task include [11] and [31]. In the current version of RAMS, a superpixel's motion is dependent on the video's optical flow, as described in the third chapter. So replacing the optical flow with a version that compensates for camera motion, as in [11], is a simple solution to adapt RAMS to camera motion. However, Wang and Schmid [11] algorithm uses human detectors to remove points corresponding to humans for better background/camera motion estimation. The use of human detectors was optional for their algorithm, but it did significantly improve their results. The goal of their algorithm is to improve action recognition for humans, so their goal is different and may not be very sensitive to noisy optical flow. So a test is needed to check its compatibility with RAMS. Another solution is replacing pixels optical flow with point-trajectories motion, where the camera motion has been removed as in [31]. However, Wu et al. [31] requires all the trajectories to have the same length, and start at the first frame. If a trajectory gets out of view, the trajectory repeats its end point until the end of the video. This requirement is not reasonable as

objects can enter and leave at any time in the video. However, their results look promising, so perhaps the video can be divided into multiple small overlapping sub-videos and processed by their algorithm. This will be tested in future work. Both algorithms, [11] and [31], have assumptions about video content while RAMS does not. In the future, a preprocessing step will be added to RAMS to eliminate the fixed camera assumption, while keeping RAMS an unsupervised approach.

References

[1] A. Levinshtein, C. Sminchisescu, and S. Dickinson. "Spatiotemporal Closure," 10th Asian Conference on Computer Vision, Queenstown, New Zealand, Pages: 369-382, November 2010.

[2] A. Setyanto, J. C. Wood, and M. Ghanbary. "Evolution Analysis of Binary Partition Tree for Hierarchical Video Simplified Segmentation," 6th Computer Science and Electronic Engineering Conference, Colchester, Pages: 52-57, September 2014.

[3] B. Nadler, and M. Galun. "Fundamental Limitations of Spectral Clustering," Advances in Neural Information Processing Systems, Volume 19, Pages: 1017-1024, 2006.

[4] B. Taylor, V. Karasev, and S. Soatto. "Causal Video Object Segmentation From Persistence of Occlusions," IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, Pages: 4268-4276, June 2015.

[5] Carnegie Mellon University (CMU) Graphics Lab Motion Capture Database: http://mocap.cs.cmu.edu
Subject: 79, Trial: 22, Motion Description: range of motion,
URL: http://mocap.cs.cmu.edu/subjects/79/79_22.mpg

[6] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang. "SOLD: Sub-Optimal Low-rank Decomposition for Efficient Video Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, Pages: 5519-5527, June 2015.

[7] C. Xu, C. Xiong, and J. J. Corso. "Streaming Hierarchical Video Segmentation," 12th European Conference on Computer Vision, Florence, Italy, Pages: 626-639, October 2012.

[8] F. Galasso, M. Keuper, T. Brox, and B. Schiele. "Spectral Graph Reduction for Efficient Image and Streaming Video Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Pages: 49-56, June 2014.

[9] F. Galasso, N.S. Nagaraja, T.J. Cardenas, T. Brox, and B. Schiele. "A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis," IEEE International Conference on Computer Vision, Sydney, NSW, Pages: 3527-3534, December 2013.

[10] F. Galasso, R. Cipolla, and B. Schiele. "Video Segmentation with Superpixels," 11th Asian Conference on Computer Vision, Daejeon, Korea, Pages: 760-774, November 2012.

[11] H. Wang, and C. Schmid. "Action Recognition with Improved Trajectories," IEEE International Conference on Computer Vision, Sydney, NSW, Pages: 3551-3558, December 2013.

[12] J. Chang, D. Wei, and J. W. Fisher III. "A Video Representation Using Temporal Superpixels," IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, Pages: 2051-2058, June 2013.

[13] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. "Track to the Future: Spatio-Temporal Video Segmentation with Long-Range Motion Cues," IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, June 2011.

[14] L. S. Silva, and J. Scharcanski. "Video Segmentation Based on Motion Coherence of Particles in a Video Sequence," IEEE Transactions on Image Processing, Volume 19, Issue 4, Pages: 1036-1049, 2010.

[15] M. Fradet, P. Robert, and P. Perez. "Clustering Point Trajectories with Various Life-Spans," Conference for Visual Media Production, London, Pages: 7-14, November 2009.

[16] M. Grundmann, V. Kwatra, M. Han, and I. Essa. "Efficient Hierarchical Graph-Based Video Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, Pages: 2141-2148, June 2010.

[17] The Video Segmentation Project by Georgia Tech and Google Research.
URL: http://www.videosegmentation.com

[18] "Mountain Zebra" Video from www.arkive.org
URL: http://www.arkive.org/mountain-zebra/equus-zebra/video-08.html

[19] N. Dimitriou, and A. Delopoulos. "Fast, Robust and Occlusion Resilient Motion Based Video Segmentation," IEEE International Conference on Image Processing, Paris, Pages: 4398-4402, October 2014.

[20] N. Dimitriou, and A. Delopoulos. "Incorporating Higher Order Models for Occlusion Resilient Motion Segmentation in Streaming Videos," Image and Vision Computing, Volume 36, Pages: 70-82, 2015.

[21] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. "Contour Detection and Hierarchical Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 33, Issue 5, Pages: 898-916, 2011.

[22] P. F. Felzenszwalb, and D. P. Huttenlocher. "Efficient Graph-Based Image Segmentation," International Journal of Computer Vision, Volume 59, Issue 2, Pages: 167-181, 2004.

[23] P. Ochs, J. Malik, and T. Brox. "Segmentation of Moving Objects by Long Term Video Analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 36, Issue 6, Pages: 1187-1200, 2014.

[24] P. Ochs, and T. Brox. "Higher Order Motion Models and Spectral Clustering," IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, Pages: 614-621, June 2012.

[25] P. Ochs, and T. Brox. "Object Segmentation in Video: A Hierarchical Variational Approach for Turning Point Trajectories into Dense Regions," IEEE International Conference on Computer Vision, Barcelona, Pages: 1583-1590, November 2011.

[26] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. "Occlusion Boundary Detection and Figure/Ground Assignment from Optical Flow," IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, Pages: 2233-2240, June 2011.

[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 34, Issue 11, Pages: 2274-2282, 2012.

[28] R. Dutta, B. A. Draper, and J. R. Beveridge. "Video Alignment to a Common Reference," IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, Pages: 808-815, March 2014.

[29] S. A. Ramakanth, and R. V. Babu. "SeamSeg: Video Object Segmentation Using Patch Seams," IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Pages: 376-383, June 2014.

[30] S. Tripathi, Y. Hwang, S. Belongie, and T. Nguyen. "Improving Streaming Video Segmentation with Early and Mid-Level Visual Processing," IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, Pages: 477-484, March 2014.

[31] S. Wu, O. Oreifej, and M. Shah. "Action Recognition in Videos Acquired by a Moving Camera Using Motion Decomposition of Lagrangian Particle Trajectories," IEEE International Conference on Computer Vision, Barcelona, Pages: 1419-1426, November 2011.

[32] T. Brox, and J. Malik. "Object Segmentation by Long Term Analysis of Point Trajectories," 11th European Conference on Computer Vision, Heraklion, Crete, Greece, Pages: 282-295, September 2010.

[33] Utility for color-coding .flo images by Daniel Scharstein.
URL: vision.middlebury.edu/flow/

[34] Y. Huang, Q. Liu, and D. Metaxas. "Video Object Segmentation by Hypergraph Cut," IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, Pages: 1738-1745, June 2009.

[35] Y. Ke, R. Sukthankar, and M. Hebert. "Spatio-Temporal Shape and Flow Correlation for Action Recognition," IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, Pages: 1-8, June 2007.

Appendix A

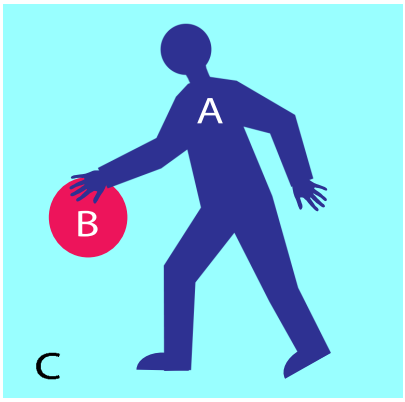
Performance Evaluation Measures

A video segmentation is evaluated by comparing it with a ground truth segmentation. Ground truth segmentations are human annotated segmentations. An evaluation score for a segmentation encompasses how much it matches a given ground truth. A set of evaluation measurements are used for this comparison. The following is a description by examples of the Volume Precision-Recall (VPR) [9] that is used for performance evaluation.

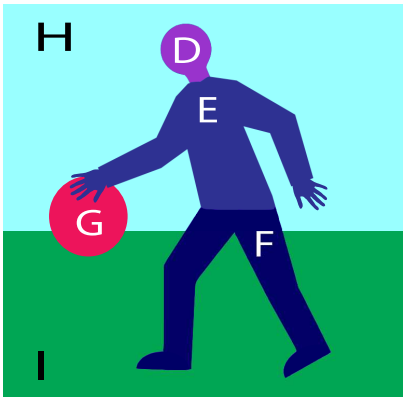
A.1 Volume Precision-Recall (VPR) [9] by Examples

Galasso et al. [9] introduced a precision-recall method to evaluate video segmentations. Each segment in a video segmentation is a 3D volume: two image spatial dimensions, in addition to the time dimension. Segments of a video segmentation result are compared with the segments of a ground truth segmentation by inspecting their pixel overlap. Precision measures how well a segment volume is encapsulated within a ground truth segment. A perfect precision is scored by a video segmentation where none of its segments overlap with multiple ground truth segments. On the other hand, recall measures how well the ground truth segments are covered by the tested segmentation segments. For both precision and recall, a matching procedure is performed to choose the corresponding pair of segments to be used for computing the scores. The following example explains this method.

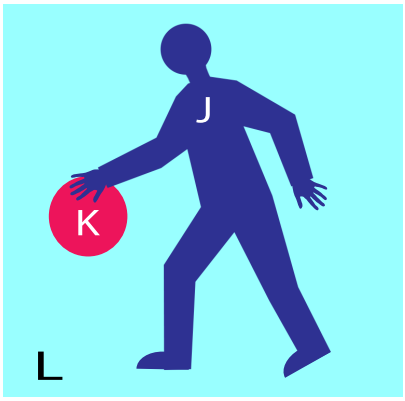
Consider an example video of a person playing with a ball and bouncing it on the ground. Assume that the given ground truth segmentation for this video is composite of three segments: person, ball, and background. A segmentation result, compared to this ground truth, can range from being either a correct segmentation, an over-segmentation where more details



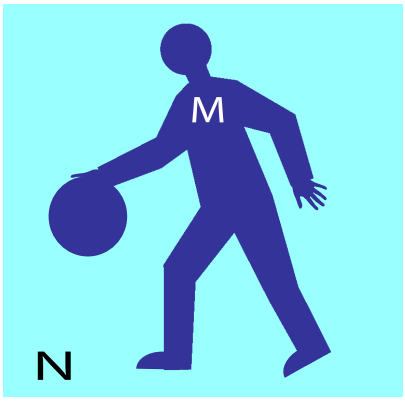
(a) Ground Truth Segmentation (S_0)



(b) First Segmentation (S_1)



(c) Second Segmentation (S_2)



(d) Third Segmentation (S_3)

Figure A.1: Segmentation examples for a video of a person playing with a ball. The segmentations are displayed in 2D for a frame. The first row is a ground truth segmentation provided by a human, while the second row contains three possible computer segmentation results.

are segmented, or an under-segmentation where it is more simplified. These segmentations are depicted in Figure A.1.

Using this video as a working example, the following is how precision and recall are computed in regards to the person. For computing the precision score of a segmentation, each of its segments is matched with a ground truth segment that has maximal pixel overlap. For example, in (S_2), segment J is matched with A. For recall, its the other way around, where each ground truth segment is matched with a segmentation segment that has maximal pixel overlap. Pixel overlap between a segment s_t and its matched segment s_m can be given as $\frac{|s_t \cap s_m|}{|s_t|}$, where $\cap$ is the intersection of pixels, and $|.|$ denotes the number of pixels. So in the case of (S_2), the person will have perfect precision and recall.

Considering the first segmentation (S_1) in Figure A.1, the person is over-segmented into three segments: D, E, and F. The precision for each of these three segments is perfect since each of them has a 100% pixel overlap with the ground truth segment A. However, the recall for the person in this segmentation is penalized since the ground truth segment A is matched to only one segment of these three segments (D, E, or F), whichever has the maximum overlap. If we assume E has the highest portion of the person in S_1 with 40% of its pixels, then A will be matched with E with a 40% recall score. So in general, using this VPR evaluation measurement, over-segmentation methods have high precision scores and low recall scores.

Considering the third segmentation (S_3) in Figure A.1, the person and the ball are in a single segment: M. For computing precision, this segment is matched with the ground truth segment A since it has maximum pixel overlap, assuming that the person is much larger than the ball (i.e. $|A| > |B|$). Since M is not entirely encapsulated within the ground truth segment, its precision is penalized. Assuming that 70% of M pixels corresponds to the person, while 30% corresponds to the ball, M will have a 70% precision score. On the other hand, the recall for the person in this segmentation is perfect since the person in the ground truth, A, is completely covered by M. So in general, under-segmentation methods have high recall scores and low precision scores.

For simplicity, the above example discussed the precision and recall scores for a single segment. The final precision and recall scores for a video segmentation S compared to a ground truth segmentation G is given as follows:

$$P = \frac{\sum_{s \in S} max_{g \in G} |s \cap g|}{|S|} \tag{A.1}$$

$$R = \frac{\sum_{g \in G} max_{s \in S} |s \cap g|}{|G|} \tag{A.2}$$

However, in VSB100 [9], multiple ground truth segmentations are provided for each video from different people to take into account the multiple levels of detail that people perceive for a video. This evaluation method is designed to accommodate multiple types of segmentation methods ranging from an over-segmentation to an under-segmentation. So the precision and recall scores are averaged over the M given ground truth segmentations, as follows:

$$P = \frac{1}{M} \sum_{i=1}^{M} \frac{\sum_{s \in S} max_{g \in G_i} |s \cap g|}{|S|} \tag{A.3}$$

$$R = \sum_{i=1}^{M} \frac{\sum_{g \in G_i} max_{s \in S} |s \cap g|}{\sum_{i=1}^{M} |G_i|} \tag{A.4}$$

A precision-recall (VPR) plot is used to plot the scores of an evaluated approach with (recall, precision) points. Figures 4.2 and 4.5 show examples of VPR plots. If an approach produces a single segmentation for a video, it will have a single point in this plot. But if an approach produces a hierarchy of segmentations for a video, it will have a point for each level in this hierarchy, with a curve line connecting these points. So the varying parameter for a curve is the level in segmentation hierarchy. If the evaluation is done on multiple videos, then these points are the average scores for each level.

Finally, F-measure is used to report a final evaluation score for the aggregate performance of an evaluated approach:

$$F = \frac{2PR}{P+R} \tag{A.5}$$

There are three scores reported for each evaluated approach: (i) Average Precision (AP), which is the area under the curve, (ii) Optimal Segmentation Scale (OSS) score, which is the F-measure when an optimal scale (level) is selected for each segmentation, and (iii) Optimal Dataset Scale (ODS) score, which is the F-measure when an optimal fixed scale (level) is selected for all segmentations.

There are two trivial cases for a segmentation: (i) all the pixels are in a single segment, and (ii) each pixel is a separate segment. The first case will result in a perfect recall score; any ground truth segment will be entirely covered by this single segment. The second case will result in a perfect precision score; every segment is completely encapsulated within a ground truth segment since it contains only one pixel. So in VSB100 [9], the scores are normalized to prevent the problem of obtaining high scores with these trivial cases. The following are the normalized scores:

$$P = \frac{\sum_{i=1}^{M}[\{\sum_{s\in S} max_{g\in G_i}|s\cap g|\} - max_{g\in G_i}|g|]}{M|S| - \sum_{i=1}^{M} max_{g\in G_i}|g|} \tag{A.6}$$

$$R = \frac{\sum_{i=1}^{M}\sum_{g\in G_i}\{max_{s\in S}|s\cap g| - 1\}}{\sum_{i=1}^{M}\{|G_i| - \Gamma_{G_i}\}} \tag{A.7}$$

where Γ_{G_i} is the number of segments in ground truth segmentation G_i. So for the first case, the recall will still be perfect, but the precision will be zero, and thus having a zero f-measure score. The opposite happens for the second case, where it will still have a perfect precision but zero recall, and thus zero F-measure score. The following example explains this normalization affect. Consider a video where its total number of pixels is 150 pixels. Assume it has only one ground truth segmentation, for simplicity, which is composite of

three segments and each of them contain 50 pixels. Table A.1 shows the computations of VPR scores for this example, both normalized and unnormalized, for these two trivial cases. As a result of this normalization, the final score is always $F = 0$ for these two cases.

Table A.1: A working example showing the difference between the normalized and unnormalized scores of two trivial cases of video segmentations: all pixels are in a single segment, and every pixel is in a separate segment. In this example, the video contains 150 pixels. It is assigned a single ground truth segmentation that is composite of three segments, each contain 50 pixels. Since there is only one ground truth segmentation in this example, normalized scores equations are simplified in the first block of this table, and used to compute the normalized scores of this table. Unnormalized scores are computed using the Equations A.1 and A.2. F is computed as in Equation A.5. The normalized scores for these two cases is: $F = 0$.

Simplified equations for this example	
$P = \frac{\sum_{s \in S} max_{g \in G} \lvert s \cap g \rvert - max_{g \in G} \lvert g \rvert}{\lvert S \rvert - max_{g \in G} \lvert g \rvert}$ $R = \frac{\sum_{g \in G} \{max_{s \in S} \lvert s \cap g \rvert - 1\}}{\lvert G \rvert - \Gamma_G}$	
First Case: all pixels = 1 segment	
Unnormalized Scores	**Normalized Scores**
$P = \frac{50}{150} = 0.33$ $R = \frac{3 \times 50}{150} = 1$ $F = \frac{2 \times 0.33 \times 1}{0.33+1} = 0.50$	$P = \frac{50-50}{150-50} = 0$ $R = \frac{3 \times (50-1)}{150-3} = 1$ $F = 0$
Second Case: each pixel = 1 segment	
Unnormalized Scores	**Normalized Scores**
$P = \frac{150 \times 1}{150} = 1$ $R = \frac{3 \times 1}{150} = 0.02$ $F = \frac{2 \times 1 \times 0.02}{1+0.02} = 0.04$	$P = \frac{150 \times 1 - 50}{150-50} = 1$ $R = \frac{3 \times (1-1)}{150-3} = 0$ $F = 0$