

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Instituto de Ciências Exatas  
Programa de Pós-Graduação em Ciência da Computação

Washington Luis de Souza Ramos

*Hyperlapse* Semântico para Vídeos Egocêntricos

Belo Horizonte  
2017

Washington Luis de Souza Ramos

*Hyperlapse* Semântico para Vídeos Egocêntricos

**Versão final**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Erickson Rangel do Nascimento  
Coorientador: Mario Fernando Montenegro Campos

Belo Horizonte  
2017

Washington Luis de Souza Ramos

**Semantic Hyperlapse for Egocentric Videos**

**Final version**

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Erickson Rangel do Nascimento  
Co-Advisor: Mario Fernando Montenegro Campos

Belo Horizonte  
2017

© 2017, Washington Luis de Souza Ramos.  
Todos os direitos reservados.

Ramos, Washington Luis de Souza.

R175s Semantic Hyperlapse for Egocentric Videos [manuscrito] /  
Washington Luis de Souza Ramos. — 2017.  
61 f. il.

Orientador(a): Erickson Rangel do Nascimento  
Dissertação (mestrado) — Universidade Federal de Minas  
Gerais, Instituto de Ciências Exatas, Departamento de Ciência  
da Computação  
Referências: f. 58-61

1. Computação – Teses. 2. Visão por computador – Teses. 3.  
Semântica - Processamento de Dados – Teses. 4. Vídeos em  
primeira pessoa – Teses. I. Nascimento, Erickson Rangel do. II.  
Campos, Mario Fernando Montenegro III. Universidade Federal  
de Minas Gerais, Instituto de Ciências Exatas, Departamento  
de Ciência da Computação. IV. Título.

CDU 519.6\*82.10(043)





UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

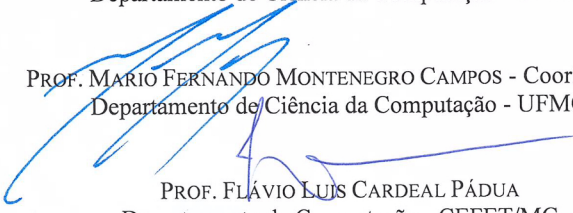
## FOLHA DE APROVAÇÃO

Semantic hyperlapse for egocentric videos

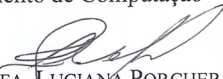
**WASHINGTON LUIS DE SOUZA RAMOS**


Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. ERICKSON RANGEL DO NASCIMENTO - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Coorientador  
Departamento de Ciência da Computação - UFMG

PROF. FLÁVIO LUIS CARDEAL PÁDUA  
Departamento de Computação - CEFET/MG

  
PROFA. LUCIANA PORCHER NEDEL  
Departamento de Informática Aplicada - UFRGS

  
PROF. WILLIAM ROBSON SCHWARTZ  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 03 de março de 2017.

*À toda minha família, aos meus colegas de projeto e aos meus professores, pois tornaram a realização deste trabalho possível.*

# Acknowledgments

Sou muito grato a todos que me ajudaram na composição deste trabalho. Seria muito difícil mencionar todos os que me apoiaram e contribuíram de alguma forma, seja ela direta ou indiretamente. Entretanto, gostaria de agradecer de uma maneira especial àqueles mais presentes no meu dia-a-dia e que compartilharam os meus momentos de nervosismo e preocupação, e de alegria e felicidade. Antecipo que, a ordem dos agradecimentos não tem nenhuma associação com a quantidade de importância para a composição do trabalho, pois a falta de uma dessas peças impactaria diretamente em sua completude.

Portanto, agradeço a Deus por me conceder os dons necessários para a realização deste trabalho e, por me iluminar durante toda essa caminhada. Agradeço aos meus pais e irmãos por me compreenderem pelos momentos em que não pude estar em casa e dar apoio quando foi preciso, mas mesmo assim me deram todo o carinho e amor necessários. Agradeço aos meus tios, tias, primos, primas, avó e amigos pelo suporte, pelas alegrias e tristezas e, sobretudo, pela paciência. À minha namorada e futura esposa, Jakeilane, por me oferecer um abraço confortante e um beijo carinhoso nos momentos mais difíceis e, peço a ela que me perdoe pelas falhas e ausências por causa dos estudos. Agradeço aos meus amigos/colegas de projeto, Michel, João e Felipe, pelas valiosas discussões, dúvidas tiradas, ajuda na composição dos datasets, criação da metodologia, etc.. Aos meus orientadores Prof. Erickson e Prof. Mario, pois foram sempre pacientes às minhas dúvidas e sempre confiaram em meu potencial, e também pela ajuda em geral (dúvidas na composição dos textos, metodologia, experimentação). Agradeço ao VerLab e DCC pela estrutura, e à CAPES, CNPq, e FAPEMIG pelo financiamento da minha pesquisa. Por fim, mas não menos importante, agradeço aos meus amigos e professores da PUC-Minas que sempre me disseram que eu tinha perfil e potencial para fazer o curso de mestrado, o que influenciou bastante em minha escolha.

# Resumo

O surgimento de dispositivos móveis pessoais de baixo custo e câmeras portáteis, e a crescente capacidade de armazenamento de sites de compartilhamento de vídeos têm impulsionado o interesse em vídeos em primeira pessoa, também conhecidos como vídeos egocêntricos. Câmeras vestíveis, em particular, podem operar por horas sem a necessidade de manuseio contínuo. Isso leva os vídeos egocêntricos a serem de longa duração com conteúdo não editado, o que os torna entediantes e visualmente desagradáveis, pois os movimentos naturais do corpo fazem com que o vídeo fique instável, causando até mesmo enjoos. Os algoritmos de *hyperlapse* visam transformar vídeos longos e monótonos em vídeos de curta duração e sem transições abruptas entre os quadros. No entanto, um aspecto importante é que algumas partes dos vídeos podem ser mais importantes do que outras, portanto devem ter a sua atenção adequada. Neste trabalho, propomos uma metodologia inovadora capaz de resumir e estabilizar vídeos egocêntricos extraíndo e analisando a informação semântica nos quadros. Este trabalho também descreve um novo conjunto de dados com vários vídeos rotulados e introduz uma nova métrica de avaliação de suavidade para vídeos egocêntricos. Diversos experimentos são conduzidos para mostrar a superioridade de nossa técnica sobre os algoritmos de *hyperlapse* do estado da arte no que diz respeito à informação semântica. De acordo com os resultados obtidos, nosso método é, em média, 10,67 pontos percentuais superior ao melhor competidor em relação à máxima quantidade semântica que pode ser obtida dado a taxa de aceleração desejada.

**Palavras-chave:** *Hyperlapse*, Aceleração de Vídeo, Informação Semântica, Vídeo em Primeira Pessoa.

# Abstract

The emergence of low-cost personal mobile devices and wearable cameras, and the increasing storage capacity of video-sharing websites have pushed forward a growing interest in first-person videos. Wearable cameras, in particular, can operate for hours without the need for continuous handling. That leads these videos to be generally long-running streams with unedited content, which makes them boring and visually unpalatable since the natural body movements cause the videos to be jerky and even nauseating. *Hyperlapse* algorithms aim to downsize long and monotonous videos into short fast-forward watchable videos with no abrupt transitions between the frames. However, an important aspect of such videos is that some parts of them may be more important than others, so they should have their proper attention. In this work, we propose a novel methodology capable of summarizing and stabilizing egocentric videos by extracting and analyzing the semantic information in the frames. This work also describes a dataset collection with several labeled videos and introduces a new smoothness evaluation metric for egocentric videos. Several experiments are conducted to show the superiority of our approach over the state-of-the-art *hyperlapse* algorithms as far as semantic information is concerned. According to the results obtained, our method is, on average, 10.67 percentage points higher than the best competitor with respect to the maximum amount of semantics that can be obtained given the required speed-up.

**Palavras-chave:** Hyperlapse, Fast-Forward, Semantic Information, First-Person Video.

# List of Figures

1.1	Wearable cameras examples . . . . .	13
1.2	Wearable cameras applications examples . . . . .	14
1.3	Example of an adaptive frame sampling with regard to the relevance to the recorder . . . . .	15
2.1	Example of a general adaptive frame sampling . . . . .	24
3.1	Overall steps of our semantic adaptive frame sampling process . . . . .	27
3.2	Our graph-building process for each video segment . . . . .	30
3.3	Examples of the Focus of Expansion (FOE) . . . . .	31
3.4	Stabilization methodology for fast forwarding egocentric videos . . . . .	32
3.5	Examples of possible distortions after applying weighted homography transformations . . . . .	33
3.6	The reconstruction process for a given warped frame . . . . .	34
4.1	Examples of the proposed semantic egocentric dataset . . . . .	39
4.2	Comparison among the epipole/FOE metric, the preference of the users and the Instability Index metric . . . . .	41
4.3	Semantic Content for the videos in Pub-Seq Dataset . . . . .	47
4.4	Semantic Content for the videos in Semantic Dataset . . . . .	48
4.5	Frames of the ‘Driving’ video, a failure case . . . . .	50
4.6	Instability Index for the videos of Pub-Seq Dataset . . . . .	51
4.7	Instability Index for the videos of Semantic Dataset . . . . .	52
4.8	Planes mismatches, failure case 1 . . . . .	54
4.9	Planes mismatches, failure case 2 . . . . .	55

# List of Tables

4.1	Pub-Seq Dataset Details . . . . .	37
4.2	Semantic Dataset Details . . . . .	38
4.3	Instability Index calculated for each experimental video (10× faster) . . . . .	42
4.4	Selected Speed-ups for the Pub-Seq Dataset . . . . .	45
4.5	Selected Speed-ups for the Semantic Dataset . . . . .	45
4.6	Instability comparison between the two major steps of our methodology for both datasets . . . . .	46
4.7	Output Speed-up for the videos of Pub-Seq Dataset . . . . .	49
4.8	Output Speed-up for the videos of Semantic Dataset . . . . .	49

# Contents

Acknowledgments	6
Resumo	7
Abstract	8
List of Figures	9
List of Tables	10
<b>1 Introduction</b>	<b>13</b>
1.1 Problem Definition . . . . .	15
1.2 Contributions . . . . .	17
1.3 Document Structure . . . . .	18
<b>2 Related Work</b>	<b>19</b>
2.1 Video Summarization . . . . .	19
2.1.1 Egocentric Video Summarization . . . . .	20
2.2 Hyperlapse . . . . .	22
2.2.1 3D Model Reconstruction . . . . .	22
2.2.2 Adaptive Frame Selection . . . . .	23
<b>3 Methodology</b>	<b>26</b>
3.1 Semantic Egocentric Fast-Forwarding . . . . .	26
3.1.1 Semantic Extraction . . . . .	27
3.1.2 Temporal Segmentation . . . . .	28
3.1.3 Speedup Rate Estimation . . . . .	28
3.1.4 Graph Building . . . . .	29
3.2 Egocentric Video Stabilization . . . . .	32
3.2.1 Master frames definition . . . . .	32
3.2.2 Transition smoothing . . . . .	33
3.2.3 Frames reconstruction . . . . .	34
<b>4 Experiments</b>	<b>36</b>
4.1 Datasets . . . . .	36
4.1.1 Public Sequences Dataset . . . . .	37



4.1.2	Semantic Egocentric Dataset . . . . .	37
4.2	Evaluation Metrics . . . . .	39
4.2.1	Instability Index Metric . . . . .	40
4.3	Parameters Setup . . . . .	42
4.3.1	$\lambda$ 's optimization via PSO . . . . .	43
4.4	Results & Discussions . . . . .	44
4.4.1	Instability improvement by the egocentric video stabilizer . . . . .	45
4.4.2	Comparison to other methodologies . . . . .	46
4.4.3	Running Times . . . . .	51
4.4.4	Concluding Remarks . . . . .	53
4.4.5	Limitations . . . . .	53
<b>5</b>	<b>Conclusions &amp; Future Work</b>	<b>56</b>
5.1	Conclusions . . . . .	56
5.2	Future Works . . . . .	57
	<b>Bibliography</b>	<b>58</b>

# Chapter 1

## Introduction

First-person Vision (FPV), also known as Egocentric Vision, is an emerging field in Computer Vision that consists of the automatic analysis of videos captured from a human-centric perspective [Li et al., 2013]. Thanks to advances in technology which constantly lead to the decreasing operational cost and the increasing storage capacity of mobile cameras, egocentric videos have shown to be an attractive way for people to document their lives. Due to this fact, the popularity of these videos has considerably increased on social media. Video-sharing services like YouTube and personal repositories are also responsible for such an increase since they provide extensive space for video storage.

Wearable devices such as GoPro™, Looxcie, and Google Glass™ cameras can be operated with no intervention, i.e., the camera operator just needs to press the “turn-on button” and thereafter, she/he is free to carry out her/his activities. It opens up unprecedented ways to record many continuous hours of regular activities such as walking, driving, and cooking, athletic activities like running and bicycling, and even working tasks



(a) GoPro™Hero 5 Black Edition



(b) Google Glass



(c) Looxcie LX2



(d) LG Action Cam

**Figure 1.1.** Wearable cameras examples.



(a) GoPro™ Hero 4 and Hero 5 used to record a sports action (mountain biking).



(b) Google Glass™ being used by one of the Google's marketing managers.



(c) An example of usage of the Looxcie LX2.



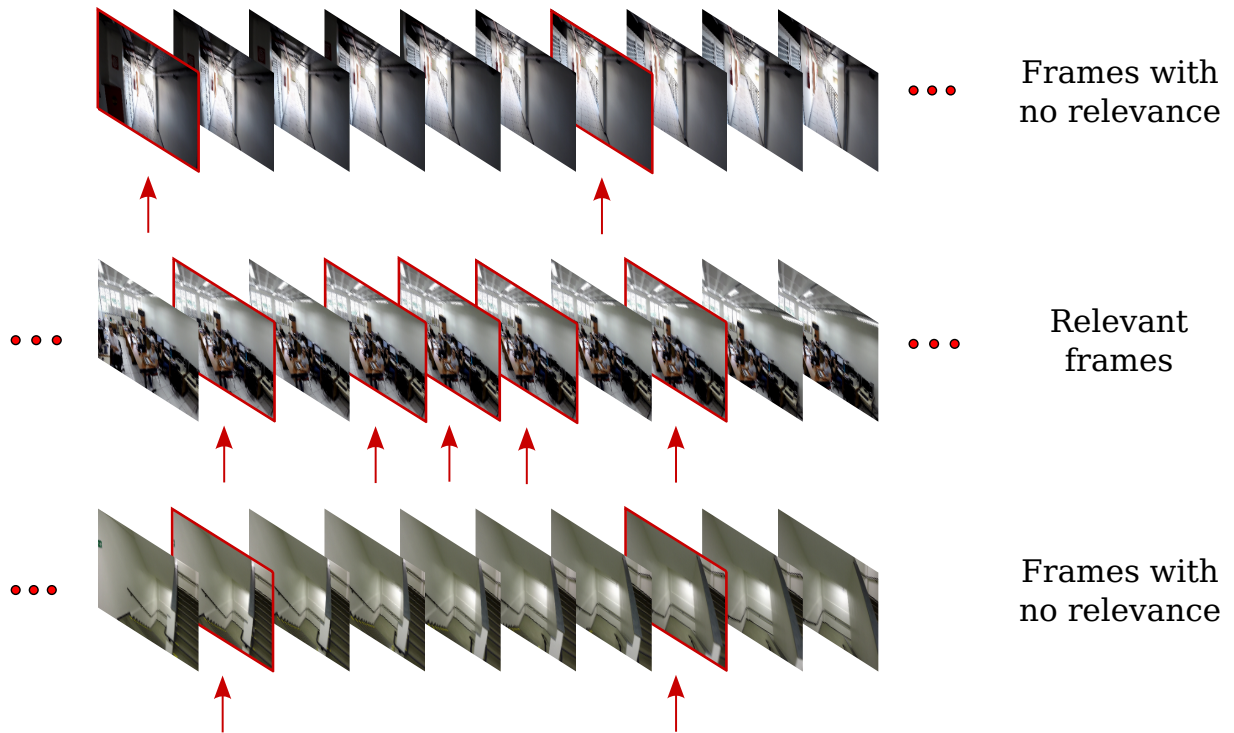
(d) LG Action Cam being used in a sports action (motocross).

**Figure 1.2.** Wearable cameras applications examples.

like event recordings (e.g., weddings, proms, birthdays, etc.) and monitoring (e.g., police patrol and lifeguarding). Examples of such devices and their usages are depicted in Figures 1.1 and 1.2, respectively.

Unlike third-person cameras such as in surveillance systems, egocentric cameras record the action like being the eye of the wearer; thus, they describe the real user's intentions and needs [Kanade and Hebert, 2012]. This is an open door for the creation of a wide range of applications in healthcare, security, education, and entertainment. The appealing and challenging environment provided by these videos has encouraged researchers from diverse study fields to engage. Examples of these fields include gaze prediction and tracking [Fathi et al., 2012; Li et al., 2013; Polatsek et al., 2016; Xu et al., 2015], recognition of activities [Fathi et al., 2011; Matsuo et al., 2014; Poleg et al., 2016; Singh et al., 2015], events [Lee et al., 2012; Lu and Grauman, 2013], objects [Ishihara et al., 2015; Ren and Gu, 2010; Wan and Aggarwal, 2015] and interactions [Yang et al., 2016] and, fast-forwarding/hyperlapse [Halperin et al., 2017; Joshi et al., 2015] which is the field this work belongs to.

In this work, we explore egocentric properties such as the focus of attention and interaction level in order to define the relevance of the scene to the recorder. Our method performs an adaptive fast-forward strategy in the egocentric video aiming to emphasize



**Figure 1.3.** Example of an adaptive frame sampling with regard to the relevance to the recorder. Three non-overlapping clips of an egocentric video are presented above. The first and last rows show the clips composed of frames with no relevance to the recorder. The clip in the second row is composed only of relevant frames. The frame selection to fast forward the video is represented by the red arrows and boxes on the frame. Note that in the relevant clip, the selection is denser than in the other clips, what leads it to be representative for the overall selection.

the sections where the recorder was more interested without removing the perception of continuity of the video. Figure 1.3 presents an example of this fast-forwarding process. Our method performs a denser frame selection in sections that are relevant to the recorder. In the next section, we detail the problem addressed in this work.

## 1.1 Problem Definition

Egocentric videos are hardly watched in their entirety because they are usually long and monotonous as a consequence of the common deferring of the users when it comes to editing and post-processing [Gygli et al., 2015]. Moreover, they contain shaky scene transitions due to natural body movements, causing visual discomfort [Bai and Reibman, 2016] and difficulty in extracting information [Poleg et al., 2016]. The use of simple fast-forward methods such as frame sub-sampling at a fixed rate is a naïve approach to reducing the video length since they do not require any understanding of

the content of the video. In contrast to the creation of fast-forward videos with carefully controlled cameras, where it is easy to track the movement between consecutive frames, in first-person videos, the significant camera shake leads the fast-forward videos to be jerky since the shakiness is increased. Consequently, the development of methods to speed up egocentric videos has become a research topic in Egocentric Vision over the last couple of years.

Several works have been proposed to address the instability of egocentric videos aiming to create a pleasant experience when watching the reduced version. Such works borrowed the term ‘hyperlapse’ from the exposure method in *timelapse* photography to name their methods. The term ‘hyperlapse’ refers to a photography technique where the camera is moved before each new photo is taken to form a moving *timelapse*. The camera moves through long distances, and the images are manually aligned to create a final video with smooth transitions along the acquisition time. Hyperlapse algorithms aim to downsize long and monotonous videos into short and watchable fast-forward videos with no abrupt transitions between the frames. In the remainder of this work, we will use hyperlapse, and smooth fast-forward as interchangeable terms.

One challenge involving the hyperlapse approaches is that some portions of the video may be more significant to the users than others. For instance, a camera installed on a police car could be recording all day long but with only a few events of interest, such as the officer interacting with someone or engaging in police activity (e.g., pursuit and capture). Most of the hyperlapse algorithms do not select frames according to their relevance to the viewer but instead treat all frames as equally relevant. Also, due to their nature of skipping stationary frames, the relevant frames may be missing in the fast-forward version.

Video summarization techniques come as the very first option to retain as much information as possible from the original video taking the minimum amount of viewing time from the user [Elkhattabi et al., 2015]. However, these techniques typically segment the video frames into shots and use features (e.g., color, edge, motion features) to find the most informative frames (namely *keyframes*) that also yield a good diversity of the original video. They cut out some parts of the video removing its continuity which is not convenient.

In this work, we propose a novel methodology capable of transforming raw egocentric videos into watchable fast-forward videos by considering both the pleasantness and relevance of frames to the viewer. Our approach analyzes the semantic information extracted from the frames and segments the video by selecting the set of pictures that maximizes the semantics, the required speed-up as well as the smoothness of the transition between the frames. We also present in this thesis a new dataset composed of semantically labeled videos and an evaluation metric to measure the egocentric videos’ smoothness. We conduct experiments on two datasets to evaluate the smoothness, the

accuracy of the required speed-up, and the overall semantic content of the video. Results show the superiority of our method as far as the semantic information is concerned. Our method is, on average, 10.67 percentage points higher than the best method in relation to the maximum amount of semantics that can be obtained, given the required speed-up. We name our method as SHEV (**S**emantic **H**yperlapse for **E**gocentric **V**ideos).

To the best of our knowledge, this is the first work that presents smooth fast-forwarding concerning the relevance of the video sections to the recorder.

## 1.2 Contributions

We can summarize our contributions as:

- i. a new adaptive fast-forwarding approach. Our method uses the disparity between the relevant and non-relevant parts to segment the input video and build graphs mapping the transition costs between pairs of frames to select those with the least cost adaptively through the shortest path algorithm;
- ii. an egocentric video stabilizer. Our algorithm stabilizes the segments by using homography transformations to match and align frames within a patch. It reconstructs the frames that were eventually too distorted to finally create a smooth output video;
- iii. a new dataset with several semantically labeled videos to fill the gap in the literature related to well-controlled datasets concerning the semantic information;
- iv. a new evaluation metric able to measure the smoothness of the egocentric videos. We demonstrate through qualitative results that the most used metric for this kind of video, which is the reduction of epipole/Focus of Expansion (FOE) jitter, is not accurate. We also present quantitative experiments to confirm the accuracy of the proposed metric.

Part of this work was published at the 2016 IEEE International Conference on Image Processing (ICIP) and at the First International Workshop on Egocentric Perception, Interaction and Computing at European Conference on Computer Vision (EPIC@ECCV) 2016.

## 1.3 Document Structure

This document is structured as follows. Next, in Section 2, we discuss the methods for video summarization and hyperlapse. In Section 3, we detail the two main steps of our methodology, the semantic fast-forwarding for egocentric video and the egocentric video stabilization. The dataset and metric contributions, experimental setup, and results are presented in Section 4. Finally, we conclude and present the future work in Section 5.

# Chapter 2

## Related Work

In this chapter, we present the related work from the Video Summarization and Hyperlapse areas. It is worth noting that video summarization and hyperlapse have some important differences. Video summarization methods are focused on creating compact visual summaries capable of presenting the most discriminative parts of the video as well as the most informative ones. Their output is generally a static storyboard of frames or dynamic skimming, which is a composition of complete video sections. On the other hand, hyperlapse methods are focused on creating a smooth, fast-forward version of the input video, i.e., the output video is sped up entirely, and no clips of the video are removed unless they are too similar. Furthermore, in hyperlapse, restrictions like suavity, continuity, and final video length play a key role in the frame selection.

### 2.1 Video Summarization

In the past several years, video summarization methods have been the main technique used to create a short summary from a long input video with the goal of maintaining essential information while saving the viewer a considerable amount of time [Lee et al., 2012; Mei et al., 2015; Zhang and Roy-Chowdhury, 2015]. It is an effective way to speed up browsing and retrieval tasks. Video summaries are typically presented in two forms: (i) static storyboard or still-image abstract, where the most representative keyframes are selected to represent the video as a whole, and (ii) dynamic video skimming or moving-image abstract, where a set of video clips compose the output.

The advantages of the still-image abstract over the video skimming are: (i) they can present, shortly, the most diversified moments once they are carefully selected through feature extraction [Kim et al., 2014], and (ii) they can be organized in many different ways for browsing or navigation, once they do not need to care about timing or synchronization issues [de Avila et al., 2008]. Early in 2008, de Avila et al. [2008] proposed a simple and efficient approach for video summarization. Their approach uses color histograms as



similarity features of the frames and clusters them via k-means. The selected keyframes are those most discriminative of each cluster based on their high representativeness. Kim et al. [2014] present their summarization output as a photo storyline graph. They collect photo streams from Flickr and user videos from YouTube to build similarity graphs in order to discover the underlying sequential structure of the photo streams by using the temporal information of the video frames.

Video skims are chosen as the output representation form in the majority of the cases because of their visual properties. Video skims are connected temporally and contain motion elements; hence they are more interesting to watch. Gygli et al. [2014] propose an approach based on the knapsack 0/1 optimization for the task of summarization. They first split the video into a series of dynamic blocks defined as superframes that have their boundaries adjusted in order to match the sequences with the least motion. These superframes are selected to compose the final set of skims via knapsack 0/1 optimization. Later, Gygli et al. [2015] proposed a learning model to jointly optimize the presence of the most interesting and discriminative parts of the video and the removal of redundancy. They combine linearly multiple objectives such as interestingness, representativeness, and uniformity using a submodular function model. It is worth noting that, unlike the majority of the video summarization approaches, the authors use an objective function to preserve the uniformity, once large skips can confuse the viewer.

Zhang et al. [2016] adopt both presentation forms as output. They propose the vsLSTM, a summarization method that uses a two-layered Long Short-Term Memory (LSTM) recurrent network in a bi-directional form in order to model long-range dependencies. It avoids deleting similar frames that are temporally distant. They further improve the diversity of the selected frames with a Determinantal Point Process (DPP) modeling. Their output is either a binary vector describing the selected frames for the storyboard or skims or a vector with probabilities for each of the frames to be chosen.

In this work, we propose an approach similar to video skims but with a stronger visual connection among the scenes to preserve the video’s continuity and content.

### 2.1.1 Egocentric Video Summarization

Regular summarization strategies are hard to be applied for the egocentric video summarization task since egocentric videos include diverse scene types, activities, and environments. Also, it is difficult to find important keyframes in such videos because of the severe camera motion, the varied illumination conditions, and the cluttered background [Lin et al., 2015].

Lee et al. [2012] present a video summarization of egocentric data, exploiting egocentric properties such as the interaction level, gaze, and object detection frequency. Their methodology produces a compact storyboard summary of the camera wearer’s day by training a regression model from annotated data to predict the relative importance of any region in a frame belonging to a person or object. Lu and Grauman [2013] present a video summarization approach that discovers the story of an egocentric video. They segment the input video into subshots, detect which objects appear in each one, and estimate the subshots individual importance. The subshots with the highest score compose the output.

Recent works focus on highlight detection. Lin et al. [2015] split their method into two sequential stages, offline and online. In the offline stage, they use a structured SVM to learn the highlight and the context models either sequentially or jointly. In the online stage, their method scans the input video, predicts the context of each segment, and scores each one with highlight confidence based on the predicted context. The final summary is composed of segments filtered from a threshold value. Bettadapura et al. [2016] present an approach for identifying picturesque moments in an egocentric video. They leverage GPS data in order to create nodes and filter them with scores assigned proportionally to the popularity of the location. The remaining frames are assigned to shots that will compose the final summary according to their aesthetics (composition, symmetry, and color vibrancy).

Although video summarization methods have been increasing their ability to select relevant frames to represent the whole video, the final result is, in general, a set of discontinuous frames. In contrast, our work manages to keep the video content entirely.

Probably, the works most related to ours in this category are the work of Okamoto and Yanai [2013] and the work of Yao et al. [2016]. In their methodology, Okamoto and Yanai generate walking route guidance videos by summarizing egocentric videos. They utilize ego-motion and pedestrian crosswalks to estimate the importance of each video section. Unlike most summarization methods, Okamoto and Yanai do not generate a summarized video. Their output, instead, is a playing scenario that determines the playing speed for each section based on their importance. Meaningful sections receive a smaller speed-up factor compared to the other sections. Although we share some of their ideas, our main goal is to provide the user with a nice and smooth experience when watching the fast-forward version. Therefore, in our methodology steps, we include the creation and stabilization of the fast-forward video. Furthermore, our algorithm is robust to various activities other than walking. We run experiments on videos of biking, driving, running, and walking activities (more details in Chapter 4).

Yao et al. propose a pairwise deep ranking model for detecting highlights in egocentric videos. The model learns the relationship between paired highlights and non-highlights segments to produce a score for each segment. The output is twofold: a com-

position of skims or a video timelapse. Yao et al. exploit a kernel temporal segmentation approach to define the segments for the video skimming. The skims are selected according to the highlight score until the desired length is achieved. For the video timelapse, they find a proper rate in order to play the highlight segments in slow motion, while the other segments are played in fast-forward to achieve a required final length. In comparison to their approach, we propose a lighter and modular one since we use the confidence assigned by the classifier and a threshold to identify the importance and the segments' boundaries. Also, we propose an adaptive frame selection approach, focusing on selecting frames that lead to a more stable video while they use uniform sampling. Moreover, the authors segment the video evenly and assume that the number of highlight segments is always smaller than the number of non-highlight segments, which might not always be valid, as we can see in examples of our proposed dataset (see Section 4.1.2 for details). Our segmentation strategy is capable of handling different configurations for the highlights lengths.

## 2.2 Hyperlapse

Hyperlapse is a recent term that refers to a cinematic technique derived from timelapse photography in which the camera pose changes at every exposure to provide an accelerated view of real-time. Recent efforts to create smooth, fast-forward videos from egocentric videos explore the automation of this technique. These approaches can be divided into two main categories: reconstruction of a 3D model of the scene along with the creation of a smooth path with a virtual camera and; adaptive selection of a set of frames that generates a smoother final video. Both categories are detailed in the next two sections.

### 2.2.1 3D Model Reconstruction

In the 3D model reconstruction category, methods use techniques to extract 3D information from multiple still images and compose a scene model, along with the camera poses, freely optimizing the camera's path for a new smooth virtual path.

A representative method in this category is the work of Kopf et al. [2014]. The authors present a technique that uses structure-from-motion (SfM) and a dense map interpolation to build a 3D model of the world. Using the camera positions and the geometric

model of the scene, they generate new virtual camera locations and orientations to make a new smooth path. Then, using image-based rendering techniques to generate the final video. Their results are stunning; however, the method creates many artifacts due to a large number of interpolated areas in the virtual camera’s path. The technique also requires camera motion and parallax to compute the 3D model of the scene. It is noteworthy that the high computational cost required by their method makes it unpractical. Moreover, the dynamics of the scene cause the SfM to fail.

### 2.2.2 Adaptive Frame Selection

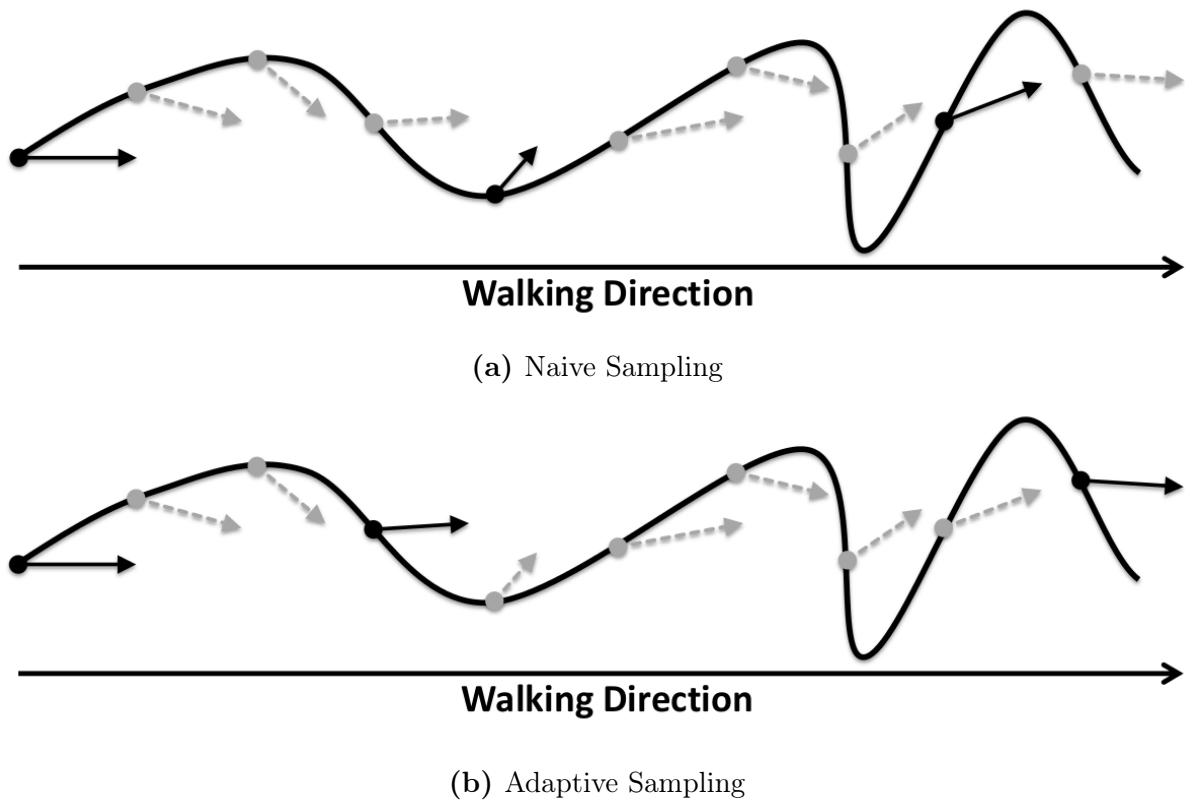
A natural approach to speed up videos is to keep in the final video the  $k$ -th frame from every consecutive set of  $n$  frames. Despite this naïve frame selection is able to reduce the video length, it increases the video shakiness, and it is not able to remove outlier frames, i.e., the frames that picture voluntary movements directions outward the main orientation of the camera wearer.

Adaptive frame selection adjusts the density of the frame selection according to the cognitive load. For instance, a denser selection could be made when the scene motion is too high, and, in turn, a sparser selection could be made when the camera wearer is stopped. In another example, a better frame selection algorithm would prefer to select forward-looking frames to reduce the shakiness, as shown in Figure 2.1. The Instagram Hypelapse App of Karpenko [2014], the work of Joshi et al. [2015] and the work of Poleg et al. [2015] are recent examples of this category.

Karpenko feeds into a video stabilizer [Karpenko et al., 2011] gyroscope samples and frames to obtain a new set of camera orientations as output. These orientations represent a smooth virtual camera motion. Thus, they apply a stabilization filter to produce *hyperlapse* videos. The major limitation of this approach is the need for inertial data, which makes it unfeasible to be used in videos recorded using a general camera.

Joshi et al. present a real-time method to create a hyperlapse video. Unlike the Instagram Hyperlapse App, their approach does not require any special sensor data. Thus, it can be used for general cameras. They use feature tracking to recover the camera motion and develop a Dynamic-Time-Warping (DTW) based algorithm to select frames subject to speed-up and smoothness restrictions in order to find an optimal smooth path. Then, the optimal set of frames is subject to 2D video stabilization, where the images are warped to render the resulting hyperlapse.

Poleg et al. propose an energy minimization model to sample the frames adaptively. Their approach focuses on skipping frames that do not represent the best viewing direction



**Figure 2.1.** Example of a general adaptive frame sampling. The curves represent a top view of a camera path with a walking direction from left to right, the arrows represent the frames, and the arrow directions represent the viewing direction. (a) Naive frame sampling for 5x fast-forward represented by the solid arrows. (b) Adaptive frame sampling, where it is preferable to select the forward-looking frames represented by the solid arrows. This figure was adapted from [Halperin et al., 2017].

to compose the final video. They create a graph from the original video where the frames are taken as nodes and edges are taken as the relation between frames. Three components are used in a linear combination to weight the edges of this graph: the shakiness cost, which assigns lower costs to forward-looking transitions; the velocity cost, which controls the playback speed of the video and; the appearance cost that prevents large visual changes between frames. Then, they compute the shortest path in order to find the best frames to compose the hyperlapse. The main contribution of their work is to model a complex problem in a graph formulation and then use a simple algorithm to perform the frame selection. Halperin et al. [2017] extended this work by expanding the field of view of the output video. They use a mosaicking approach on the input frames with single or multiple egocentric videos.

While the 3D category can generate highly smooth videos since virtual images are created based on the estimated 3D model to decrease the discontinuity between frames, the 2D category is faster and can provide similar smoothness if a judicious selection of frames is defined. Although the aforementioned solutions succeed in speeding up long

videos and producing a result that is pleasant to watch, they do not take into account the fact that some frames are more important than others, which is related to the semantics in regions of the scene. For instance, places where the camera stop moving, such as at a red light when riding a bike or stopping by to talk to a person at a family party, are taken as redundant in those methods. Therefore, they are removed from the resulting video. In this work, we present a new method that selects frames based on semantic information without degenerating the smoothness of the video.

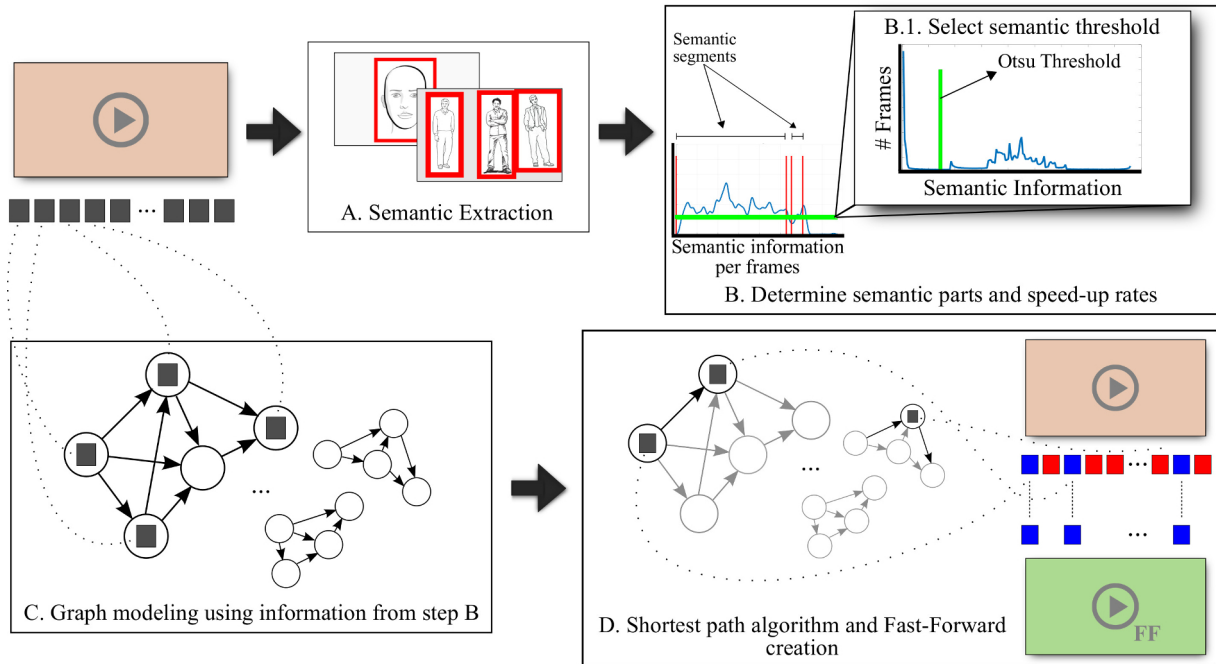
# Chapter 3

## Methodology

In the following two sections we detail our methodology to create semantic hyperlapse videos. We divide it into two main steps: semantic fast-forwarding and semantic egocentric stabilization. In the first step, the algorithm seeks the frames in the input video that maximize the semantic content, the smoothness and the proximity to the required speed-up. It segments the video into semantic and non-semantic segments and builds a graph for each type of segment mapping the frames and transition costs between pairs of frames. Then, it chooses the frames that minimize the overall cost via shortest path algorithm. The adaptive selection of frames is subject to an egocentric stabilization in the second step. Homography transformations are used to align the frames transitions and, an iterative stitching process is responsible for filling the frames that were excessively distorted by the homography transformations.

### 3.1 Semantic Egocentric Fast-Forwarding

This section presents the first step of our methodology: the semantic frame sampling process. It is composed of four sub-steps. We first extract the semantic information from each frame. Therefore, we split the video into semantic and non-semantic segments using the semantic values to define a threshold value. Then, we calculate different speed-up rates for each type of segment such that a lower speed-up rate emphasizes the semantic segments. Finally, the video frames and their relationships are used to construct a graph. We optimize the video shakiness, semantic content, and length by running the Dijkstra's shortest path algorithm. Figure 3.1 summarizes our frame sampling approach.



**Figure 3.1.** Overall steps of our semantic adaptive frame sampling process. From the input video, we extract ROIs containing the semantic information (A) in each frame and compute the semantic scores to define the semantic profile (B). We use the Otsu thresholding method to find a meaningful semantic threshold (B.1) in order to identify the semantic segments and calculate the speed-up rates based on the length of each segment. Then, we create a graph for each segment mapping the frames and their relations to the nodes and edges, respectively (C). Finally, we compute the shortest path and compose the final video with the selected nodes (D).

### 3.1.1 Semantic Extraction

In the first step of our sampling approach, we extract the semantic information present in each video frame according to the semantic selected by the user (e.g., pedestrian, face, car plate, etc.). The semantic information is encoded by the score function  $S : \mathbb{R} \rightarrow \mathbb{R}$ , which is composed of three components:

- i. the confidence of the extracted information. The user defines a classifier according to the application to analyze the frame and detect important regions, i.e., the Region of Interest (ROI). We use the confidence of the classifier as an important feature to compose the semantic score;
- ii. the size of the ROI. Larger areas mean that the object of interest is close to the recorder; therefore, it represents a higher probability of interaction;
- iii. the centrality of the ROI. Since the input is an egocentric video, the central area of the frame should have higher relevance to the viewer since it is where people are usually focused.



Formally, let  $k$  be the  $k$ -th ROI returned by the extractor in the frame  $f_x$ . The total semantic score is given by:

$$S_x = \sum_{k \in f_x} c_k \cdot a_k \cdot G_\sigma(k), \quad (3.1)$$

where  $c_k$  is the normalized confidence of the classifier for the ROI  $k$ , assigning relevance proportional to the reliability of the semantic information in frames, and  $a_k$  is the normalized area of the  $k$ -th ROI in pixels. To quantify the centrality of the object, we use a Gaussian mask with standard deviation  $\sigma$  and centered at the frame  $f_x$ .  $G_\sigma(k)$  is the value of the central point of the  $k$ -th ROI in the Gaussian function, which returns higher values to more centralized objects. Examples are illustrated in Figure 3.1-A.

### 3.1.2 Temporal Segmentation

The semantic score along the frames will define the semantic profile of the video as illustrated in Figure 3.1-B. Following most video summarization approaches, we split the video to create temporal segments by thresholding the semantic profile. We create a histogram with the semantic scores and, since we assume that this histogram has a bimodal distribution, we apply the Otsu thresholding method [Otsu, 1979] to find the threshold that better defines the disparity between the semantic and non-semantic frames. The value returned by Otsu (green line in Fig. 3.1-B.1) is used as the semantic threshold. Thus, every frame above this value is labeled as a semantic frame. Consecutive frames labeled as semantic will compose the semantic segments, and the remaining ones will compose the non-semantic segments.

### 3.1.3 Speedup Rate Estimation

To avoid losing relevant parts of the video, we calculate different speed-up rates for each type of segment defined in the previous step such that a lower speed-up rate,  $F_s$ , is applied to semantic segments. Consequently, in order to manage the whole video at the desired speed-up,  $F_d$ , the non-semantic segments receive a higher speed-up rate,  $F_{ns}$ . Estimating these speed-ups is not a trivial task once the total length of the semantic segments may vary. Therefore, given the total number of frames in semantic segments,

$L_s$ , and in the non-semantic segments,  $L_{ns}$ , the speed-up rates are computed by the minimization of the Equation 3.2:

$$D(F_{ns}, F_s) = \left| \frac{L_s + L_{ns}}{F_d} - \left( \frac{L_s}{F_s} + \frac{L_{ns}}{F_{ns}} \right) \right|. \quad (3.2)$$

The Equation 3.2 has many minimum points, once for every  $F_s$  there is a correspondent  $F_{ns}$  that leads the result to 0. We solve it by restricting the  $F_s$  and  $F_{ns}$  values so that the  $F_s$  is minimized as well as the difference between both.

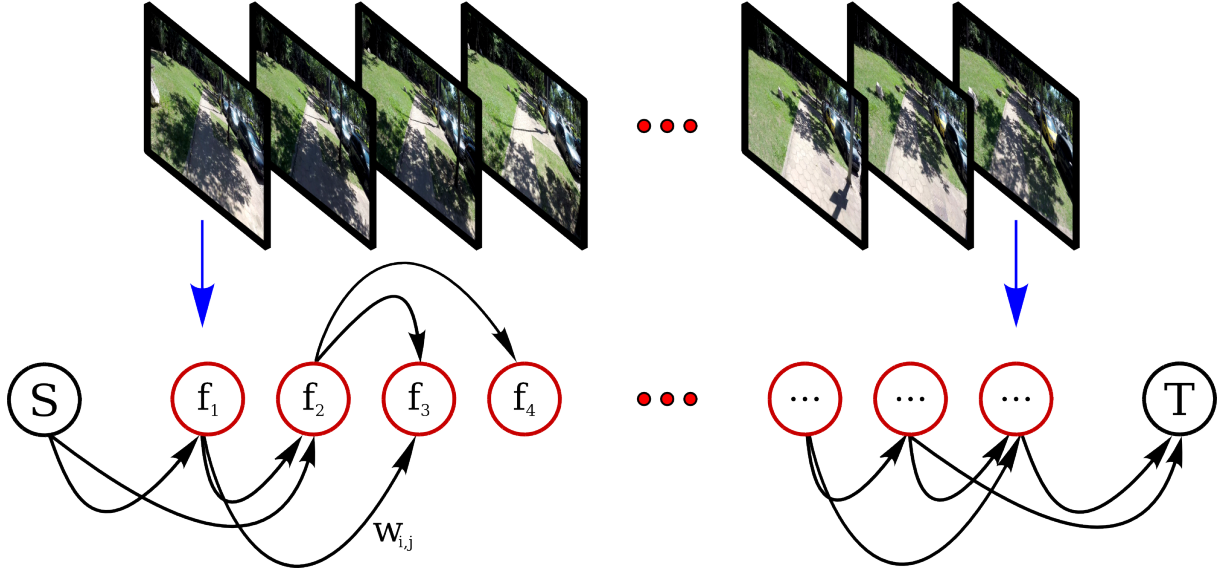
We also add some space restrictions: (i)  $F_s \leq F_d$ , once we want more emphasis in the semantic parts; (ii)  $F_{ns} \geq F_d$ , once we want to achieve desired speed-up in the fast-forward video and; (iii)  $F_s \geq p_s F_d$ , where  $p_s = L_s / (L_s + L_{ns})$ , once  $F_s < p_s F_d$  leads to an excessive number of frames. Given these restrictions and because  $F_{ns}$ ,  $F_s$  and  $F_d \in \mathbb{N}$ , the problem becomes easier to be solved, since the search space is finite and discrete. Thus, the optimization problem is represented by the Equation 3.3:

$$\begin{aligned} (F_s^*, F_{ns}^*) &= \arg \min_{F_s, F_{ns}} D(F_{ns}, F_s) + \lambda_1 |F_{ns} - F_s| + \lambda_2 |F_s| \\ &\text{subject to } F_s \leq F_d \\ &\quad F_{ns} \geq F_d \\ &\quad F_s \geq p_s F_d, \end{aligned} \quad (3.3)$$

where  $\lambda_1$  and  $\lambda_2$  are the regularization terms that give more importance to either keeping the speed-up rates close or taking the smaller  $F_s$ .

### 3.1.4 Graph Building

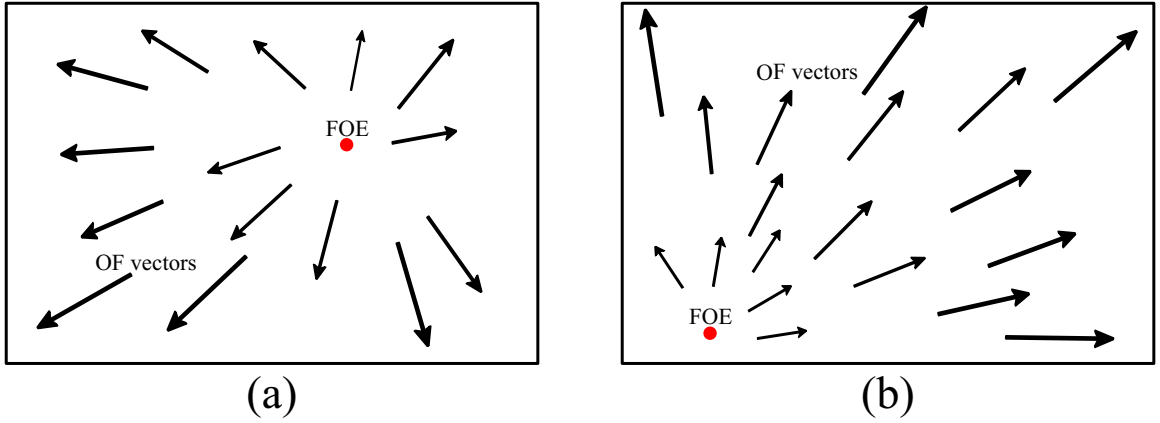
We model each video segment using a weighted graph similar to Poleg et al. [2015] and Halperin et al. [2017]. Figure 3.2 illustrates our graph building process. Each node of this graph represents a frame of the input video, and an edge connecting two nodes represents the existence of a temporal relation between the pair of frames. We connect the  $\tau_b$  border frames of each graph with one source and one sink node. The edges connecting the regular nodes are created up to a temporal distance  $\tau_{max}$  to reduce the graph complexity. The cost of the transitions from frames  $f_i$  to  $f_j$  are taken as the edges weight  $W_{i,j}$ . These costs are composed of a linear combination of four terms related to the shakiness, speed of motion, appearance change, and semantic gain/loss caused by the transition. The first three terms were previously proposed by Poleg et al. [2015] and Halperin et al. [2017] in



**Figure 3.2.** Our graph-building process for each video segment. Each frame of the video segment becomes a node in the graph, and the edges’ weights  $W_{i,j}$  indicate the cost when including the frame  $f_i$  right before the frame  $f_j$  in the fast-forward video. Nodes  $S$  and  $T$  represent the source and sink nodes of each graph, respectively.

their graph construction. The details of the four terms are presented as follows.

- **Instability Cost Term ( $I_{i,j}$ ).** To measure the instability of the transition, we first compute the motion direction of each frame by estimating the Focus of Expansion (FOE). The FOE is one particular point in the image extracted from the relative motion between two time-varying images [Sazbon et al., 2004]. Given the optical flow (OF) vectors, it can be understood as the point where the flow vectors seem to be flowing out. Examples of FOEs are depicted in Figure 3.3. The instability cost term prefers forward-looking frames; therefore, we use the difference between the FOE positions as the shakiness cost. Motivated by the good results achieved by Poleg et al. [2015] with sparse optical flow computations, our estimation of OF vectors and FOE are also obtained according to Poleg et al. [2014] and Sazbon et al. [2004], respectively.
- **Velocity Cost Term ( $V_{i,j}$ ).** This term controls the playback speed of the output video by skipping more frames where the camera motion is low and skipping less when the motion is high. We define the desired magnitude,  $D_{mag}$ , to act as a target for the average magnitude of the optical flow for consecutive output frames,  $A_{mag}(i, j)$ . It is preferable to choose the frames  $f_i$  and  $f_j$  to be consecutive in the output video if the average magnitude of the optical flow computed for this pair is closer to  $D_{mag}$ . Therefore, the velocity cost is computed as:  $V_{i,j} = A_{mag}(i, j) - D_{mag}$ .
- **Appearance Cost Term ( $A_{i,j}$ ).** More similar frame transitions make the video more enjoyable. We use the Earth Mover’s Distance (EMD) [Pele and Werman,



**Figure 3.3.** Examples of the Focus of Expansion (FOE). Black arrows represent the OF vectors, and the red dot represents the FOE position in images (a) and (b). Images (a) and (b) are not related to each other. Note that the OF vectors seem to be flowing out from the FOE. Thus, it can be used as an estimator for motion.

2009] between the color histograms of frames  $f_i$  and  $f_j$  as a resemblance measure. The EMD is a measure of distance between two distributions. It is defined as the minimum amount of “work” needed to change one distribution into the other. The appearance cost term value is proportional to the EMD value.

- **Semantic Cost Term ( $S_{i,j}$ ).** This term is used to penalize the transitions that are not composed of frames with relevant semantic information. Given the semantic score  $S_i$  of the frame  $f_i$  and the semantic score  $S_j$  of the frame  $f_j$ , the semantic cost is given by Equation 3.4:

$$S_{i,j} = \frac{1}{S_i + S_j + \epsilon}. \quad (3.4)$$

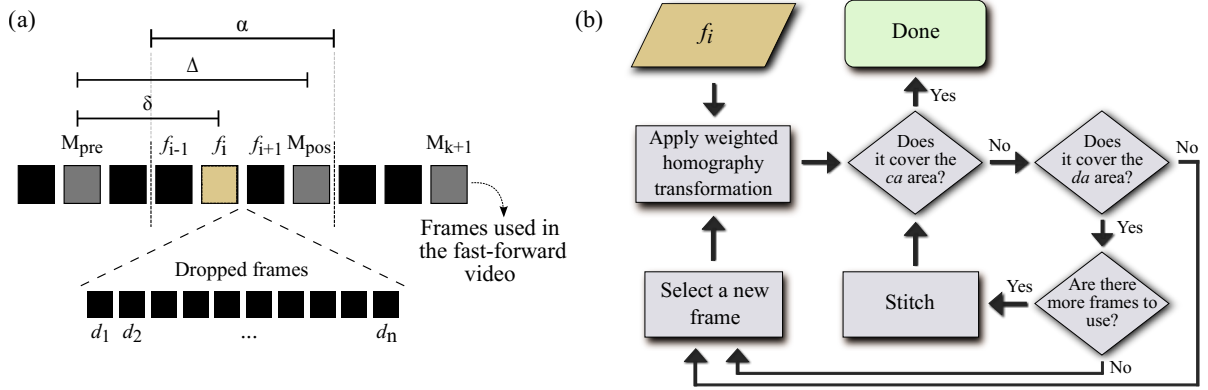
The value  $\epsilon$  avoids dividing by zero when there is no semantic information in both frames.

The final weight  $W_{i,j}$  of the edge  $E_{i,j}$  is given by:

$$W_{i,j} = (\lambda_I \cdot I_{i,j} + \lambda_V \cdot V_{i,j} + \lambda_A \cdot A_{i,j} + \lambda_S \cdot S_{i,j}) \cdot \left\lceil \frac{(j-i)}{F} \right\rceil, \quad (3.5)$$

where the values of  $\lambda$  coefficients are the regularization factors for each one of the terms of the costs. We add a proportional factor to enhance transitions between frames with lower distance, where  $F \in \{F_s, F_{ns}\}$ .

The best frame selection in our modeling is obtained by running the Dijkstra’s shortest path algorithm in each graph separately. The frames related to the selected nodes will compose the final fast-forward video.



**Figure 3.4.** Stabilization methodology for fast-forwarding egocentric videos. (a) Illustration of how the video is segmented into temporal patches, dropped frames, and the terms  $\alpha$ ,  $\Delta$ , and  $\delta$ . (b) The diagram of the stabilization process.

## 3.2 Egocentric Video Stabilization

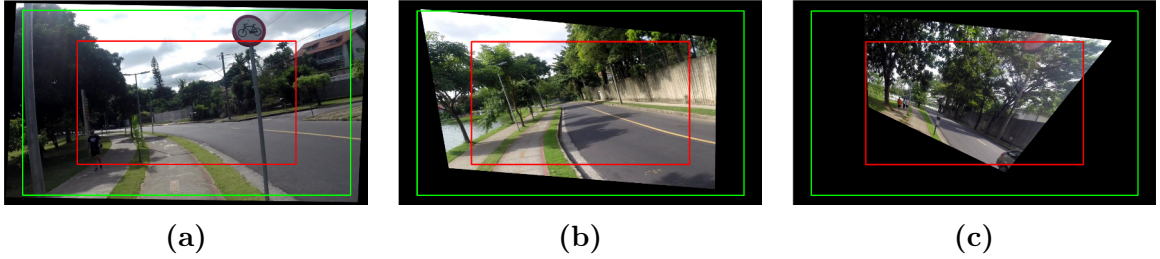
As noted by Kopf et al. [2014], traditional video stabilization algorithms do not achieve good results on egocentric videos. This can be assigned to the difficulty of tracking the motion between successive frames, which is increased in the fast-forward version. In this section, we present our egocentric stabilization method for semantic fast-forward egocentric videos. We first segment the video into temporal patches and, for each patch, look for the frame that contains the best features for matching the other frames in the patch, the master frame. Then, we apply weighted homography transformations to the intermediate frames for every pair of masters, intending to create smooth transitions along with the output video. Finally, we reconstruct the frames “over-warped” by the homography transformations by an image stitching process.

### 3.2.1 Master frames definition

The first step of the stabilization methodology consists of segmenting the video into temporal patches of length  $\alpha$  and selecting one master frame  $M_k$  for each patch (Fig. 3.4-a). We select as the master of the  $k$ -th patch, the frame  $M_k$  in this patch that maximizes the Equation 3.6:

$$M_k^* = \arg \max_{M_k \in p_k} \sum_{f_i \in p_k} R(f_i, M_k), \quad (3.6)$$

where  $p_k$  is the  $k$ -th patch and the  $f_i$  is the  $i$ -th frame of the fast-forward video. The



**Figure 3.5.** Examples of possible distortions after applying weighted homography transformations. The green outer boxes in the images represent the crop area  $ca$ , and the red inner boxes represent the drop area  $da$ . The leftmost frame (a) covers the  $ca$  area, the middle frame (b) covers the  $da$  area, and the rightmost frame (c) does not cover any area.

function  $R(x, y)$  calculates the number of *inliers* in the RANSAC method [Fischler and Bolles, 1981] when computing the homography transformation from the image  $x$  to  $y$ .

### 3.2.2 Transition smoothing

The second step is to smooth the transitions between the selected master frames. We share with Hsu et al. [2012] the idea of alleviating the transitions with weighted homography transformations. However, instead of using homography consistency and smoothing the transitions between segments, we propose to segment the video into temporal patches and alleviate the transitions between the master frames.

For each frame  $f_i$ , we calculate two homography matrices,  $H_{f_i, M_{pre}}$  and  $H_{f_i, M_{pos}}$ .  $M_{pre} = f_b$  stands for the previous master frame, which is the one that is temporally closer to the frame  $f_i$ , s.t.  $b < i$ .  $M_{pos} = f_a$  stands for the posterior master frame, which is the one that is temporally closer to the frame  $f_i$ , s.t.  $a > i$ . Both homography transformations are applied with weights set according to the temporal distance to the masters. The  $i$ -th frame of the stabilized video ( $\hat{f}_i$ ) is given by:

$$\hat{f}_i = H_{f_i, M_{pre}}^{1-w} H_{f_i, M_{pos}}^w f_i. \quad (3.7)$$

The term  $H_{x,y}^p$  in Equation 3.7 represents the  $p$ -th power of the homography transformation matrix from the image  $x$  to the image  $y$ .  $w = (\delta(2\alpha)/\Delta)$  is the weight that composes the  $p$ -th power, where  $\delta$  is the temporal distance from the frame  $f_i$  to  $M_{pre}$ , and  $\Delta$  is the distance between  $M_{pre}$  and  $M_{pos}$  (Fig. 3.4-a). As stated by Hsu et al., choosing the  $\alpha$  value to be a power of 2 makes the root calculation feasible by consecutive square roots.



**Figure 3.6.** The reconstruction process for a given warped frame. The images from left to right represent the stitching being applied to an intermediate frame, i.e., a frame between two masters.

### 3.2.3 Frames reconstruction

As expected, after applying the homography transformations estimated in Equation 3.7, black areas are generated due to the fact that the camera movements are abrupt and the elapsed time between consecutive frames in the fast-forward videos is large. Thus, the last step is to reconstruct these corrupted regions.

To reconstruct these frames, we define two image areas centered in the frame: i) the drop area ( $da$ ) equals to  $dp\%$  size of the frame and; ii) the crop area ( $ca$ ) equals to  $cp\%$  size of the frame, where  $cp > dp$ . Figure 3.5 depicts such areas. The  $da$  area is the center of the image, where the viewer focuses on the majority of the time. Therefore, it is not allowed present any black or reconstructed areas. On the other hand, the area between the  $ca$  and  $da$  is the peripheral vision, which is allowed to present artifacts but not black areas. The  $ca$  area is the cut region; thus, regions outside this area are removed in the final video. Therefore, having these black areas outside does not cause any issues. The reconstruction procedure is an iterative process represented by the flowchart in Figure 3.4-b and described by the Algorithm 1.

The stitching step is performed as follows. We use the SURF detector to select feature points in the frame  $\hat{f}_i$  and in the  $j$ -th frame dropped from the original video,  $d_j$ . To calculate the homography transformation we match feature points between the images by describing all feature points of  $d_j$  and  $\hat{f}_i$  with SURF descriptors and applying the brute force matching strategy. Given the matched points, we calculate the homography matrix  $H_{d_j, \hat{f}_i}$  using the RANSAC method. The  $\hat{d}_j = H_{d_j, \hat{f}_i} d_j$  is now aligned and stitched with  $\hat{f}_i$  to compose the reconstructed image. Figure 3.6 illustrates the stitching step in a frame that covers the  $da$  area (inner red box) but not the  $ca$  area (external green box).

If it is necessary to select a new frame, it means that the  $\hat{f}_i$  does not yield a good transition in the final video. The algorithm selects a new frame  $d_j$  that belongs to the interval  $[f_{i-1}, f_{i+1}]$  in the original video and maximizes the Equation 3.8:

$$d_j^* = \arg \max_{d_j} (G_\sigma(p)(R(d_j, \hat{f}_{i-1}) + R(d_j, \hat{f}_{i+1}))(\eta + S(d_j))), \quad (3.8)$$

---

**Algorithm 1** Reconstruction Procedure

---

**Requires:** The set of frames  $\mathcal{F}$  of the fast-forward video; The set of frames  $\mathcal{D}$  dropped in the fast-forward process; The *da* and *ca* areas; The set of the selected masters  $\mathcal{M}$ .

**Ensures:** The set of stabilized frames  $\mathcal{S}$ .

```

1: function STABILIZEEGOVIDEO( $\mathcal{F}$ )
2:    $\mathcal{S} \leftarrow \{\}$  ▷ The set of output frames
3:   for each  $f_i \in \mathcal{F}$  do
4:      $\hat{f}_i \leftarrow \text{ApplyWeightedHomography}(f_i, \mathcal{M})$  ▷ According to Equation 3.7
5:     while  $\neg \text{ItCovers}(\hat{f}_i, ca)$  do
6:       if  $\text{ItCovers}(\hat{f}_i, da)$  &  $\text{ExistUnusedFrames}(f_i, \mathcal{D})$  then
7:          $\hat{f}_i \leftarrow \text{DoStitching}(\hat{f}_i, d_j)$ 
8:       else
9:          $d_j \leftarrow \text{SelectNewFrame}(\mathcal{M}, f_i)$  ▷ According to Equation 3.8
10:         $\hat{f}_i \leftarrow \text{ApplyWeightedHomography}(d_j, \mathcal{M})$ 
11:       end if
12:     end while
13:      $\mathcal{S} \leftarrow \mathcal{S} + \{\hat{f}_i\}$ 
14:   end foreach
15: end function

```

---

where,  $G_\sigma(x)$  is the value of the Gaussian function with zero mean and standard deviation  $\sigma$  in the position  $x$ ;  $p$  is the percentage of area covered by  $d_j$ ;  $\eta$  is a value used to prevent multiplication by zero, in case the function  $S(d_j)$  that calculates the semantic score in the frame  $d_j$  returns zero. The final stabilized video is composed of all frames that achieve the Done step.



# Chapter 4

## Experiments

This chapter presents the whole experimental setup and evaluations for our proposed methodology. We first present details about the datasets used in Section 4.1, including specific aspects of our proposed dataset composition. Then, in Section 4.2 we present the evaluation metrics used for comparison, including details about the new instability evaluation metric. The parameters configuration are presented in Section 4.3. Finally, we show the comparison with literature work and the results in Section 4.4.

The experiments were executed using an Intel<sup>®</sup>Core<sup>™</sup> i7-3770 CPU at 3.40GHz with 8 cores and 32GB of RAM. All the semantic fast-forwarding algorithm was implemented in MATLAB due to the simple matrix manipulations provided by the language. We used the C++ implementation of OpenCV in our egocentric video stabilization, which is also implemented in C++.

### 4.1 Datasets

In the following two sections, we describe the datasets we used to conduct our experiments. We present in the next section the Pub-Seq Dataset, which is a collection of publicly available videos that other authors previously used to evaluate their hyperlapse methods. Then, we present details about the Semantic Dataset, which is a collection of videos that we recorded aiming to achieve a certain level of semantics to further verify the effectiveness of our method. We use people as semantic information since people are usually considered relevant to the wearers and watchers.

**Table 4.1.** Pub-Seq Dataset Details. Sequences were collected from the sources in the ‘Source’ column. All sequences were filmed at 30 frames per second (fps), except ‘Running’ and ‘Walking 3’, which were filmed at 24 and 15 fps, respectively.

Name	Source	Resolution	Camera	Number of Frames
Bike 1	[Kopf et al., 2014]	1280 × 960	Hero3	10,786
Bike 2	[Kopf et al., 2014]	1280 × 960	Hero3	7,049
Bike 3	[Kopf et al., 2014]	1280 × 960	Hero3	23,700
Running	[Poleg et al., 2015]	1280 × 720	Hero3+	12,900
Driving	[Poleg et al., 2015]	1280 × 720	Hero2	10,200
Walking 1	[Kopf et al., 2014]	1280 × 960	Hero2	17,249
Walking 2	[Kopf et al., 2014]	1280 × 720	Hero	6,900
Walking 3	[Poleg et al., 2015]	1920 × 1080	Hero3+	7,999
Walking 4	[Poleg et al., 2014]	1920 × 1080	Hero3+	15,667

#### 4.1.1 Public Sequences Dataset

This dataset is composed of 9 publicly available sequences which were used by Kopf et al. [2014], Joshi et al. [2015], and Poleg et al. [2015] to evaluate their hyperlapse methodologies. All sequences were filmed with GoPro™Hero series cameras at 30 frames per second (fps), except ‘Running’ and ‘Walking 3’, which were filmed at 24 and 15 fps, respectively. Details about these sequences are shown in Table 4.1.

During our experiments, we found a limitation in this dataset. None of the videos has a considerable amount of semantics. This prevents us from testing our method in scenarios where most part of the video is composed of semantic content. Therefore, we propose a new labeled dataset which is presented in the next section.

#### 4.1.2 Semantic Egocentric Dataset

We propose a new labeled dataset to run the experiments and validate our methodology along with the Pub-Seq Dataset since no semantically controlled egocentric datasets were found in the literature. A dataset of such kind provides us with richer details about the behavior of our algorithm in diverse cases.

The dataset comprises 11 videos divided into 3 categories of different activities: Biking; Driving and Walking. The videos under each one of these categories are classified according to their amount of semantic information. The classes are: 0p, which represents

**Table 4.2.** Semantic Dataset Details. Videos with the resolution of 1280x720 were filmed at 60 fps, and the others were filmed at 30 fps. All videos were recorded with a GoPro™ Hero3+ camera.

Name	Resolution	Number of Frames
Biking 0p	1280 × 720	17,949
Biking 25p	1920 × 1080	17,071
Biking 50p	1280 × 720	26,954
Biking 50p2	1280 × 720	14,939
Driving 0p	1920 × 1080	9,463
Driving 25p	1920 × 1080	7,989
Driving 50p	1920 × 1080	10,379
Walking 0p	1920 × 1080	8,219
Walking 25p	1920 × 1080	10,982
Walking 50p	1920 × 1080	11,570
Walking 75p	1920 × 1080	15,481

the videos with approximately no semantic information present (Biking 0p, Driving 0p, and Walking 0p); 25p, for the videos containing relevant semantic information in approximately 25% of its frames (Biking 25p, Driving 25p and Walking 25p); 50p, for the ones with around a half of their frames composed by semantics (Biking 50p, Biking 50p2, Driving 50p and Walking 50p) and; 75p, which represents videos with approximately 75% of their frames containing relevant semantic information (Walking 75p).

We defined people as the relevant object for this dataset since people play an important role in most of the egocentric recordings, either in casual or in security applications. To find the people in the videos, we used the Normalized Pixel Difference (NPD) Face Detector [Liao et al., 2016] (the state-of-the-art face detector) for the videos of the Walking category or a pedestrian detector [Dollár, 2016] for the videos of the other categories. We tried to use faces as the semantic information for all videos, but the usage of the pedestrian detector was necessary because the videos when biking or driving present a higher motion speed, which prevents the face detector from achieving a substantial accuracy. The semantic information used to measure and classify the videos in the categories was obtained according to the Equation 3.1. We apply the same process as in Section 3.1.2 in order to define which frames are classified as relevant and which are not.

The videos were recorded with a GoPro™Hero 3+ camera mounted in a helmet for the Biking and Walking videos and attached to a head strap for the Driving videos. All videos were recorded in daylight so that the detectors could achieve better accuracy. Table 4.2 shows some details about the videos in the dataset and Figure 4.1 shows some frame examples. The complete dataset, including videos and the semantic labels, is



**Figure 4.1.** Examples of the proposed semantic egocentric dataset. Frames in the first row represent the videos of the ‘Biking’ category. Frames in the second row represent the videos of the ‘Walking’ category. Frames in the third row represent the videos of the ‘Driving’ category.

publicly available to the research community <sup>1</sup>.

## 4.2 Evaluation Metrics

Our final goal is to create a semantic smooth fast-forward egocentric video with a given number of frames. A satisfying result depends on the amount of semantic information, the achieved speed-up factor, and the visual smoothness of the final video. Measuring the visual smoothness is complex. Most hyperlapse methodologies either use qualitative metrics, which involve human evaluation of the videos, or the reduction of epipole/FOE jitter in the final video as a quantitative metric. Even though the qualitative measurement is appropriate, once the final video has to be enjoyable to the viewers, during our preliminary experiments, we noticed that the reduction of epipole/FOE jitter occasionally assigns better scores for videos evidently more shaky. Based on that, we devised a quantitative metric that takes into account the preference of the viewers. We used the following metrics to quantify the accuracy of the evaluated methodologies:

- I. **Semantic Content.** This metric indicates the amount of semantic information in the output video. Output videos with higher values demonstrate that the technique managed to keep more semantic information from the input video. We calculate it

<sup>1</sup><https://www.verlab.dcc.ufmg.br/semantic-hyperlapse>

through the Equation 4.1:

$$Semantics = \sum_{i=1}^{\widehat{L}} \widehat{S}_i, \quad (4.1)$$

where  $\widehat{S}_i$  is the semantic score (see Eq. 3.1) of the  $i$ -th output video frame.

- II. **Output Speed-up.** This metric indicates the speed-up achieved by the output video. It is better to have a speed-up close to the required speed-up. We calculate the achieved speed-up according to the following equation:

$$Speedup = \frac{L}{\widehat{L}}, \quad (4.2)$$

where  $L$  and  $\widehat{L}$  are the number of frames of the input and the output videos, respectively.

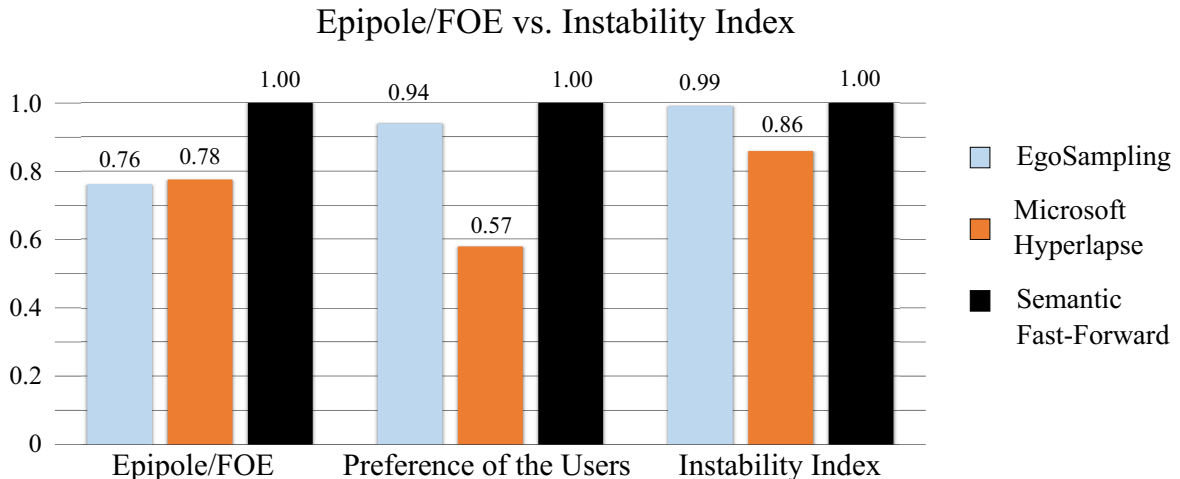
- III. **Instability Index.** This metric quantifies the shakiness of the output video. It is detailed in Section 4.2.1. The lower is the Instability Index, the smoother is the video.

### 4.2.1 Instability Index Metric

Hyperlapse methodologies focus on producing smooth fast-forward egocentric videos. In order to evaluate the smoothness of the output videos, we need an evaluation metric that accurately expresses this value. The most popular quantitative measure present in the literature is the reduction of the epipole/FOE jitter [Halperin et al., 2017; Poleg et al., 2015], which is not accurate.

Inspired by the qualitative comparison between videos made by Joshi et al. [2015], where they made side-by-side comparisons using only the mean and standard deviation of consecutive output frames, we devised a quantitative metric to calculate the smoothness of videos. We assume that, in shaky videos, the pixels should present more distant values in a range of consecutive frames when compared to smoother videos. The shakiness estimation is computed as in Equation 4.3, which presents a value for the instability of the video.

$$I = M \left( \frac{1}{\overline{N}} \cdot \sum_{i=1}^N \frac{\sum_{j \in B_i} (f_j - \bar{f}_i)^2}{(N_B - 1)} \right), \quad (4.3)$$



**Figure 4.2.** Comparison among the epipole/FOE metric, the preference of the users, and the Instability Index metric. The smaller the value, the better it is. Results are normalized by the highest mean value for better visualization. The epipole/FOE metric presents a low mean for the ES algorithm, which does not match the users’ preference, unlike the Instability Index, which seems to be a better match.

where  $N$  is the number of frames of the video,  $B_i$  is the  $i$ -th buffer composed by  $N_B$  temporal neighbor frames,  $f_j$  is the  $j$ -th frame of the video, and  $\bar{f}_i$  is the average frame of the buffer  $B_i$ .  $M(\cdot)$  is a function that returns the mean value for the pixels of a given image, and  $I$  indicates the instability index of the video. A smoother video yields a smaller  $I$  value.

Once the viewers are whom the methodologies concern, we conducted a user study to verify the real smoothness of the videos and to assess the quality of the metric. For the qualitative evaluation, we generated output videos with an average length of 35 seconds from the 9 sequences present in Table 4.1 using the following smooth fast-forwarding techniques: EgoSampling (ES) [Poleg et al., 2015]; Microsoft Hyperlapse (MH) [Joshi et al., 2015] and Ours (semantic fast-forward step only), with a speed-up factor of 10. Then, we asked for 33 subjects to watch the videos (randomly and with no labels) and grade each video instability with respect to its smoothness in an assessment questionnaire. The format of the questionnaire is a five-level Likert item for the question “How shaky is the video?”. The items are as follows: (1) Not a bit shaky; (2) A little shaky; (3) Shaky; (4) Very shaky; (5) Too shaky. We demonstrate the accuracy of our evaluators by low average standard deviations, which represent a small divergence in the user responses:  $\sigma = 0.93$  for the ES output videos;  $\sigma = 0.97$  for the MH output videos and;  $\sigma = 0.78$  for our output videos.

Figure 4.2 shows the mean values of the 9 sequences, normalized by the highest mean of each metric. Unlike the quantitative measure of epipole/FOE locations differentiation, where the ES technique is superior to the other two techniques, the majority of the subjects preferred watching the MH output video. Results reveal that the proposed

**Table 4.3.** Instability Index calculated for each experimental video (10× faster).

$N_B$ \ Video	Head01	Gimbal01	Head02	Gimbal02	Head03	Gimbal03
5	28.535	<b>26.145</b>	31.780	<b>27.192</b>	33.775	<b>29.917</b>
7	31.242	<b>29.164</b>	34.900	<b>30.779</b>	36.124	<b>32.785</b>
11	34.919	<b>33.262</b>	39.089	<b>35.768</b>	39.278	<b>36.620</b>
15	37.472	<b>36.030</b>	42.005	<b>39.183</b>	41.467	<b>39.173</b>

metric really reflects the preference of the subjects since it is more similar.

We realized a quantitative experiment to consolidate the accuracy of the Instability Index metric. In this experiment, we used two GoPro™ Hero series cameras, one attached to a 3-axis handheld gimbal and the other to a head strap. Since the gimbal is a hardware stabilizer, we expect the videos recorded with it to be more stable.

We recorded three pairs of videos ((Head01, Gimbal01), (Head02, Gimbal02), and (Head03, Gimbal03)) with an average length of 9 minutes and applied a 10x naive fast-forwarding to reduce the videos. For each pair of videos, we calculated the instability index for both with different buffer sizes ( $N_B$  in Eq. 4.3) to compare their smoothness. Results of these experiments are presented in Table 4.3. The smallest instability index values are in bold. As expected, the videos recorded with the gimbal present the smallest values since it causes the camera to become more stable in acquisition time.

### 4.3 Parameters Setup

In this section, we show details of our experimental setup, which includes the datasets used, the methods and metrics that were chosen for comparison, and the parameters configuration for both main steps of our methodology.

Most parameters were defined empirically, but some were optimized via a bio-inspired algorithm since they are simple to implement and can find reasonable solutions efficiently. We tried two kinds of algorithms: Genetic Algorithms (GAs) [Man et al., 1996], which are inspired by the process of natural selection and use a population of candidate solutions, and the Swarm Intelligence algorithms, more specifically the Particle Swarm Optimization (PSO) [Kennedy and Eberhart, 1995], which shares many similarities with the GAs, but instead of evolution operators such as crossover and mutation it uses particles moving through the solution space. We chose PSO over a GA to use in our experiments because PSO is less complex and could find solutions with less computational costs than GAs. In this section, we present the values that we defined empirically. Details about

our optimization via PSO are presented in Section 4.3.1.

In our semantic fast-forwarding methodology, we used as the semantic extractors the same detectors used for the labeled dataset composition: the Liao et al.’s NPD Face Detector [Liao et al., 2016] in videos where the wearer is walking and the Dollár’s pedestrian detector [Dollár, 2016] in videos where the motion speed is higher (running, driving and biking). These detectors are responsible for giving us the confidence value ( $c_k$ ) and the ROI size ( $a_k$ ) of the Equation 3.1. We considered any  $c_k < 60$  as false face detections and  $c_k < 100$  as false pedestrian detections. The Gaussian function (Equation 3.1) is a Normal with parameters  $\mu = 0$  and  $\sigma = \min(W/2, H/2)$ , where  $W$  is the frame width and  $H$  is the frame height.

In the temporal segmentation (Sec. 3.1.2), we filtered the semantic profile with a Gaussian function with  $\sigma = 5 \cdot fps$ , where  $fps$  stands for frames per second and ‘ $\cdot$ ’ is the multiplication operator. We only considered ranges with 3 seconds or over, once short ranges would result in a flash in the fast-forward video. We also connected every semantic segment separated by a span of 5 seconds or less because we consider this gap as a misdetection range. Such actions change the  $p_s$  value (Sec. 3.1.3) of each video, once some frames below the semantic threshold can be part of semantic segments and vice-versa. However, it reduces the number of transitions between different segments and makes the segments more solid.

For the construction of the graph, we set the values of the border frames  $\tau_b$  and the maximum allowed skip  $\tau_{max}$  to be 1 and 100, respectively. In the velocity cost term, we set the value of the desired magnitude for the optical flows  $D_{mag}$  to be 10 times the average optical flow magnitude of the sequence. We tested different values for the  $\epsilon$  in the semantic cost term equation (Equation 3.4). The value with the best results was  $\epsilon = 1$ . Finally, for all experiments, we set the desired speed-up to  $F_d = 10$ .

In our egocentric video stabilization methodology, the size of the patches for selection of the master frames was defined as  $\alpha = 4$ . We set the area of  $da$  as  $dp = 50\%$  of the frame and the area of  $ca$  as  $cp = 90\%$ . The parameter  $\sigma$  of the Gaussian function in the Equation 3.8 and the value of  $\eta$  in the same equation were defined as  $\sigma = 10$  and  $\eta = 0.5$ . We used the OpenCV implementation of SURF and RANSAC.

### 4.3.1 $\lambda$ ’s optimization via PSO

The PSO algorithm comprises a group of particles arranged randomly in the search space. The optimization process occurs iteratively. In every iteration, the particles’ positions (parameters values) are updated to follow the local and global best particles.



Local best is the particle that achieved the best solution in the current iteration, and global best is the particle with the overall best solution. The solution is given by a fitness equation which is defined according to the problem.

For the optimization of the parameters  $\lambda_1$  and  $\lambda_2$  of the Equation 3.3, we designed the following fitness equation:

$$fitness_{\lambda_1\lambda_2} = c \left| F_s^* - \frac{F_d + p_s F_d}{2} \right| + |\widehat{F}_d - F_d| + p_{ns} |F_s^* - F_{ns}^*|, \quad (4.4)$$

where  $F_s^*$  and  $F_{ns}^*$  are the best values of  $F_s$  and  $F_{ns}$  when replacing  $\lambda_1$  and  $\lambda_2$  with the particle position.  $p_s = L_s/(L_s + L_{ns})$  represents the semantic percentage of the video,  $p_{ns} = L_{ns}/(L_s + L_{ns})$  is non-semantic percentage,  $c = 2$  is a constant value to control the importance of selecting a lower semantic speedup and  $\widehat{F}_d = (L_s + L_{ns})/(L_s/F_s^* + L_{ns}/F_{ns}^*)$  is the speedup achieved with the selected speedups.

The fitness equation for obtaining the best values for  $\lambda_I$ ,  $\lambda_V$ ,  $\lambda_A$  and  $\lambda_S$  is:

$$fitness_{\lambda_I\lambda_V\lambda_A\lambda_S} = \frac{J}{Max_J} + \left| \frac{\widehat{L} - E_L}{E_L} \right| + \frac{\widehat{S}^* - Semantics}{\widehat{S}^*}, \quad (4.5)$$

where  $J$  is the jitter of the generated fast-forward video, which is obtained by the magnitude mean deviation of the FOE locations along the selected frames;  $Max_J$  is the maximum jitter possible for the video, obtained by the  $J$  value of a hypothetical video where for every frame the FOE is as far as possible from the previous;  $E_L$  is the expected number of frames for the fast-forward video,  $\widehat{L} = L/\widehat{F}_d$  is the fast-forward video length and  $L$  is the original video length;  $\widehat{S}^*$  is the maximum value for the Semantic Score of the fast-forward video that could be obtained given the required speed-up; and  $Semantics$  is the value calculated as in Equation 4.1 for the fast-forward video. Due to performance restrictions, we use the jitter measure instead of the proposed Instability Index metric.

## 4.4 Results & Discussions

We first compare the improvement in stability when applying our egocentric stabilizer to the semantic fast-forward videos. Then, we present an overall comparison against the naïve frame selection, the work of Poleg et al. [2015] and the work of Joshi et al. [2015].

Tables 4.4 and 4.5 present the semantic percentage ( $ps$ ), the speed-ups obtained by minimization of the Equation 3.3 ( $F_s$  and  $F_{ns}$ ) using the PSO algorithm and, the theoretic final speed-up calculated for the both datasets used in our experiments ( $S$ ). It is

**Table 4.4.** Selected Speed-ups for the Pub-Seq Dataset.

Name	ps	$F_s$	$F_{ns}$	S
Bike 1	17.48%	5	12	9.640
Bike 2	2.79%	2	10	8.995
Bike 3	21.93%	6	12	9.842
Driving	0.00%	10	10	10.000
Running	7.01%	3	12	9.914
Walking 1	1.84%	1	12	9.977
Walking 2	0.00%	1	10	10.000
Walking 3	11.77%	4	12	9.713
Walking 4	6.94%	3	12	9.933

**Table 4.5.** Selected Speed-ups for the Semantic Dataset.

Name	ps	$F_s$	$F_{ns}$	S
Biking 0p	1.06%	1	11	9.947
Biking 25p	24.81%	6	13	10.081
Biking 50p	54.73%	8	14	9.926
Biking 50p2	51.72%	8	13	9.824
Driving 0p	2.25%	1	12	9.619
Driving 25p	24.23%	6	11	9.152
Driving 50p	46.33%	8	12	9.743
Walking 0p	0.00%	10	10	10.000
Walking 25p	25.60%	6	13	10.011
Walking 50p	50.11%	8	13	9.899
Walking 75p	75.20%	9	15	9.991

noticeable that our designed fitness function could manage different amounts of semantics since the overall speed-up remains closer to the desired speed-up, which is  $F_d = 10$ , and the selected speed-ups are not too distant nor too close, except for the cases where the semantic percentage is close to 0%. Note that in order to get the perfect overall speed-up, the value of  $F_{ns}$  should be very high since we intend to get the smallest possible value for  $F_s$  without degrading the video continuity.

#### 4.4.1 Instability improvement by the egocentric video stabilizer

We calculated the Instability Index for the videos of both datasets before and after the stabilization step of our methodology. Table 4.6 summarizes our results. In general, the output videos produced by the complete methodology are more stable than the videos produced by the semantic fast-forward step only. Failure cases are observed in Driving

**Table 4.6.** Instability comparison between the two major steps of our methodology for both datasets. The column **SHEV-Stb** presents the instability index for the output video of the semantic fast-forward step only, and the column **SHEV+Stb** presents the results for the output video of the complete methodology.

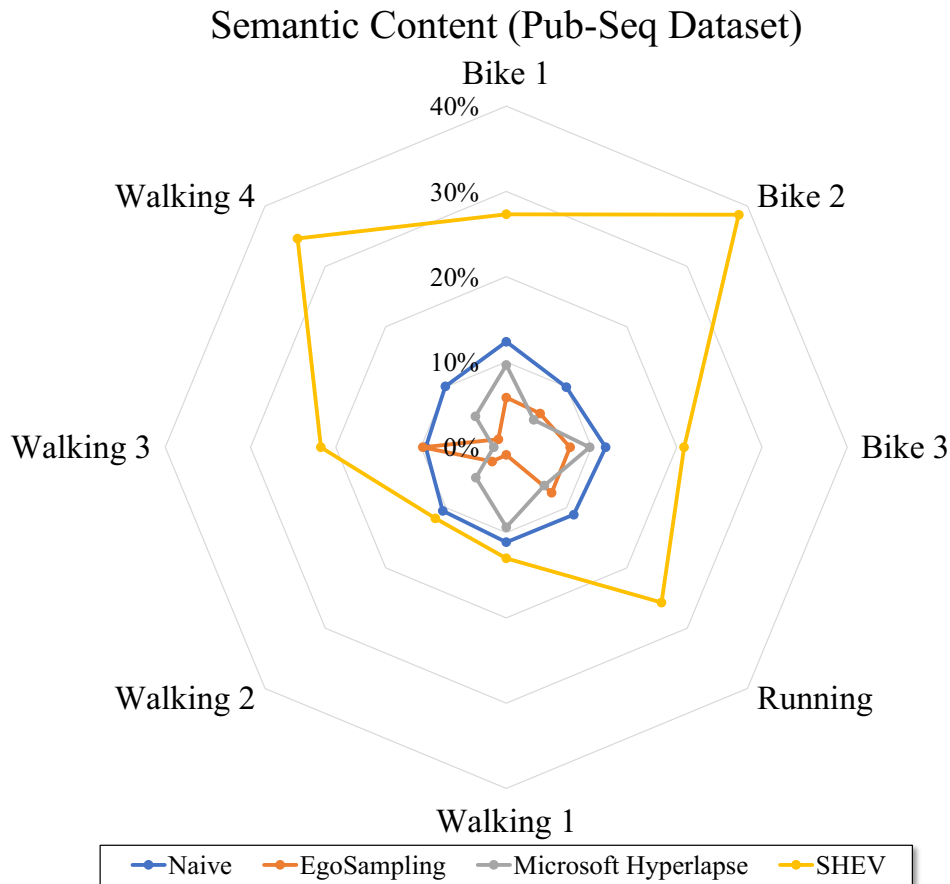
Semantic Dataset			Pub-Seq Dataset		
Name	SHEV-Stb	SHEV+Stb	Name	SHEV-Stb	SHEV+Stb
Biking 0p	28.77	<b>26.81</b>	Bike 1	37.68	<b>36.58</b>
Biking 25p	52.36	<b>50.28</b>	Bike 2	36.68	<b>35.68</b>
Biking 50p	36.23	<b>32.91</b>	Bike 3	36.76	<b>36.12</b>
Biking 50p2	31.64	<b>29.20</b>	Driving	41.44	<b>39.00</b>
Driving 0p	<b>46.59</b>	48.09	Running	39.21	<b>38.28</b>
Driving 25p	<b>42.21</b>	43.39	Walking 1	30.47	<b>27.18</b>
Driving 50p	42.84	<b>42.24</b>	Walking 2	36.98	<b>35.73</b>
Walking 0p	35.54	<b>35.43</b>	Walking 3	37.51	<b>35.56</b>
Walking 25p	37.73	<b>37.38</b>	Walking 4	35.35	<b>34.67</b>
Walking 50p	40.94	<b>38.24</b>			
Walking 75p	38.52	<b>35.95</b>			

0p and Driving 25p. The videos recorded in a car are more challenging for our egocentric stabilizer since many scene changes occur in a short interval because of the car’s speed. A large number of skipped frames in the fast-forward step causes the sequential frames in the fast-forward video to be spatially distant. Thus, the key-point correspondences are not valid, leading the homography calculation to be erroneous.

#### 4.4.2 Comparison to other methodologies

We compared the results of our complete methodology against three different techniques:

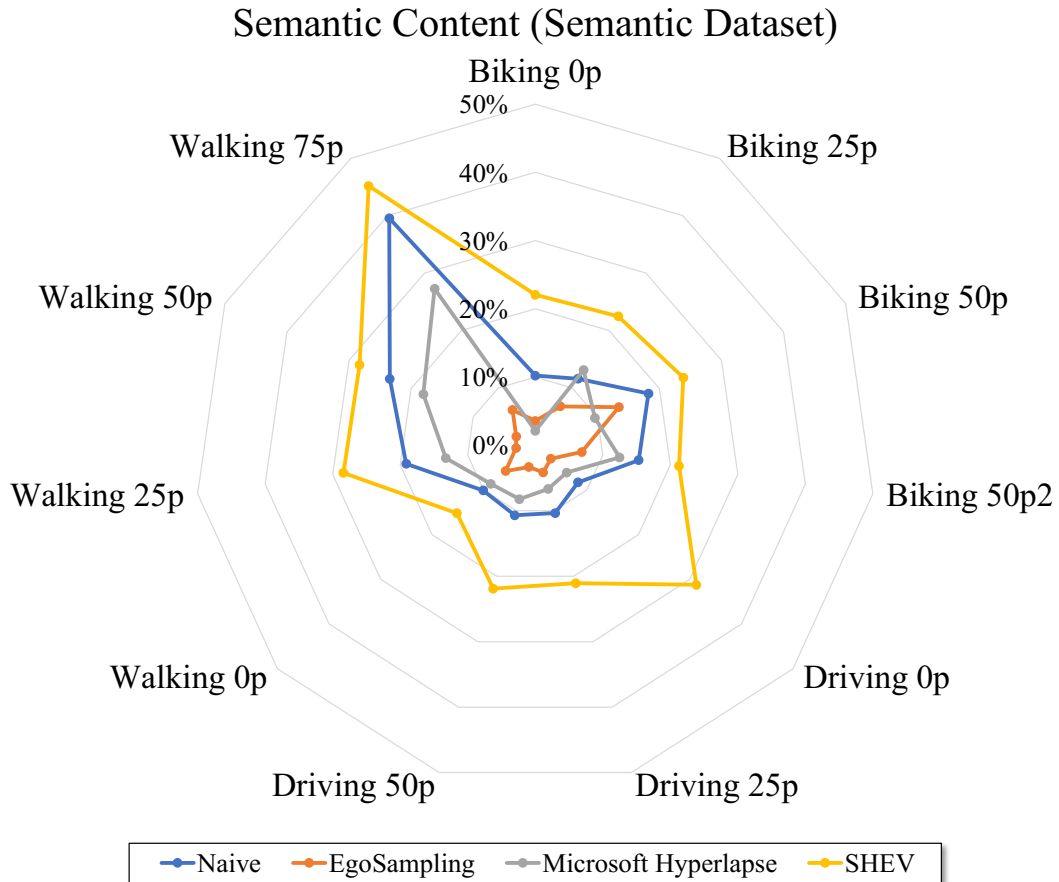
- (i) *Naïve* (N), which simply creates a video by taking every  $n$ -th frame of the input video. This selection gives us the perfect Output Speed-up, which is the exact value required, but the Instability Index and the Semantic Content are video-dependent values;
- (ii) *EgoSampling* (ES) [Poleg et al., 2015], which creates a video by using the Poleg et al.’s technique with parameters defined according to the best values of their work;
- (iii) *Microsoft Hyperlapse* (MH) [Joshi et al., 2015], where we used the released desktop version of their algorithm to create the output videos.



**Figure 4.3.** Semantic Content for the videos in Pub-Seq Dataset. The results are related to the highest Semantic Content that could be achieved given the required speed-up. Results for the ‘Driving’ video were removed since this video has no semantic information. Our method is on average 11.88 percentage points better than the Naïve approach, which is the competitor with the highest average semantic content.

**Semantic Evaluation.** Figures 4.3 and 4.4 depict the semantic content value normalized by the number of frames of the output video for the videos of both tested datasets. We present the results concerning the maximum semantic content achievable given the desired speed-up. The maximum semantic content is given by the sum of the semantic score of the  $k$  frames with the highest values, where  $k$  is the ideal number of frames for the output video. Results for the ‘Driving’ video were removed since it has no semantic information. Therefore, all algorithms would present the same value of semantic content for their output videos.

Our method outperforms all other methodologies as far as semantic information is concerned. In Figure 4.3, it is noteworthy the small values obtained by the hyperlapse techniques that are even smaller than the Naïve ones. This fact is due to the uniformity of frame selection of the Naïve approach along with the skipping strategy of the hyperlapse techniques. Hyperlapse algorithms tend to make larger skips when the motion is low, for example, when the recorder is stopped. This might have led the techniques to exclude frames with more semantic information. Our technique stands out in this aspect since, in



**Figure 4.4.** Semantic Content for the videos in Semantic Dataset. The results are related to the highest semantic content that could be achieved given the required speed-up. Our method is, on average, 9.46 percentage points better than the Naïve approach, which is the competitor with the highest average semantic content.

addition to reducing the speed-up factor in semantic segments, the semantic term balances the selection in non-semantic segments. Our method is, on average, 11.88 percentage points better than the Naïve approach, which is the competitor with the highest average semantic content.

We repeat our analysis for the Figure 4.4 which presents the results for the semantic dataset. It is important to note that our technique is robust to different amounts of semantics. The main reason for this is the restrictions imposed by Equation 3.3 that demands the minimization of the semantic speed-up and its difference from the non-semantic speed-up independently of the length of the semantic segments. Our method is, on average, 9.46 percentage points better than the Naïve approach, which is the competitor with the highest average semantic content.

**Speed-up Evaluation.** Tables 4.7 and 4.8 present the Output Speed-up achieved by the techniques in Pub-Seq Dataset and Semantic Dataset, respectively. For a better analysis of the data, we removed the Naïve technique since it always achieves the required speed-up.

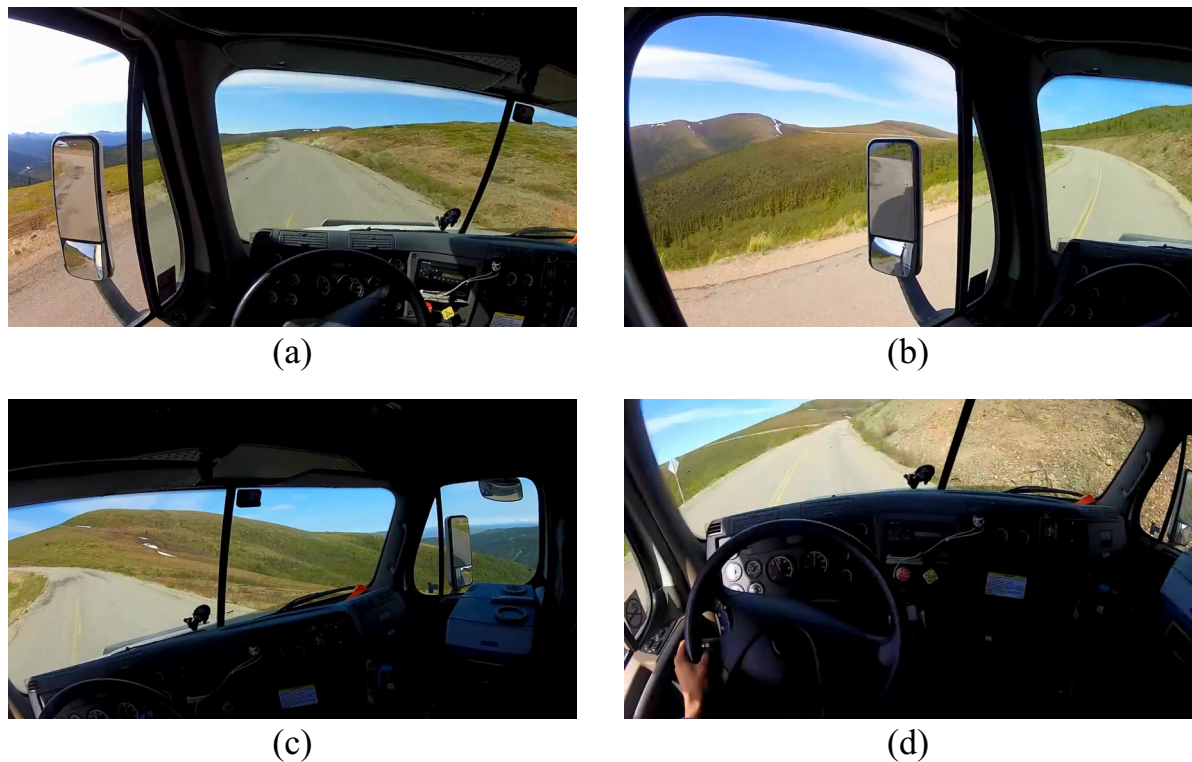
**Table 4.7.** Output Speed-up for the videos of Pub-Seq Dataset. Better results are those with the values closer to  $F_d = 10$ .

Video Name	EgoSampling	Microsoft Hyperlapse	SHEV
Bike 1	15.344	11.052	<b>10.110</b>
Bike 2	13.663	10.586	<b>10.352</b>
Bike 3	13.383	11.122	<b>9.996</b>
Driving	57.309	<b>11.272</b>	24.002
Running	13.495	10.841	<b>9.970</b>
Walking 1	54.245	11.115	<b>11.001</b>
Walking 2	24.824	8.735	<b>10.425</b>
Walking 3	12.520	7.929	<b>10.000</b>
Walking 4	25.770	9.255	<b>9.999</b>
<b>Mean</b>	25.617	10.212	11.762
<b>St. Dev.</b>	17.823	1.241	4.602

**Table 4.8.** Output Speed-up for the videos of Semantic Dataset. Better results are those with the values closer to  $F_d = 10$ .

Video Name	EgoSampling	Microsoft Hyperlapse	SHEV
Biking 0p	24.028	<b>10.152</b>	11.926
Biking 25p	11.388	8.389	<b>10.006</b>
Biking 50p	14.024	10.760	<b>10.002</b>
Biking 50p2	18.086	8.397	<b>9.979</b>
Driving 0p	32.187	<b>10.024</b>	11.814
Driving 25p	25.938	10.430	<b>10.049</b>
Driving 50p	26.078	11.221	<b>10.038</b>
Walking 0p	14.244	7.391	<b>10.011</b>
Walking 25p	13.328	8.307	<b>9.993</b>
Walking 50p	24.256	7.632	<b>10.000</b>
Walking 75p	27.160	9.199	<b>9.994</b>
<b>Mean</b>	20.974	9.264	10.347
<b>St. Dev.</b>	6.979	1.319	0.754

In general, our technique produces hyperlapse videos with the speed-up closest to the desired one. A failure case is present in the ‘Driving’ output video shown in Table 4.7. This is a challenging video where the driver with a camera attached to his head alternates between looking ahead and looking in the left rear-view mirror often. This leads to larger frame skips aiming to eliminate outlier frames, which are those where the driver is looking in the rear-view mirror. Although we have a factor to penalize distant skips (see Eq. 3.5), the graph may have low edge weights for higher temporal relationships. We believe this occurred in this experiment because the video has a homogeneous content, i.e., the frames have similar structure and appearance. Figure 4.5 presents some frames of this video.

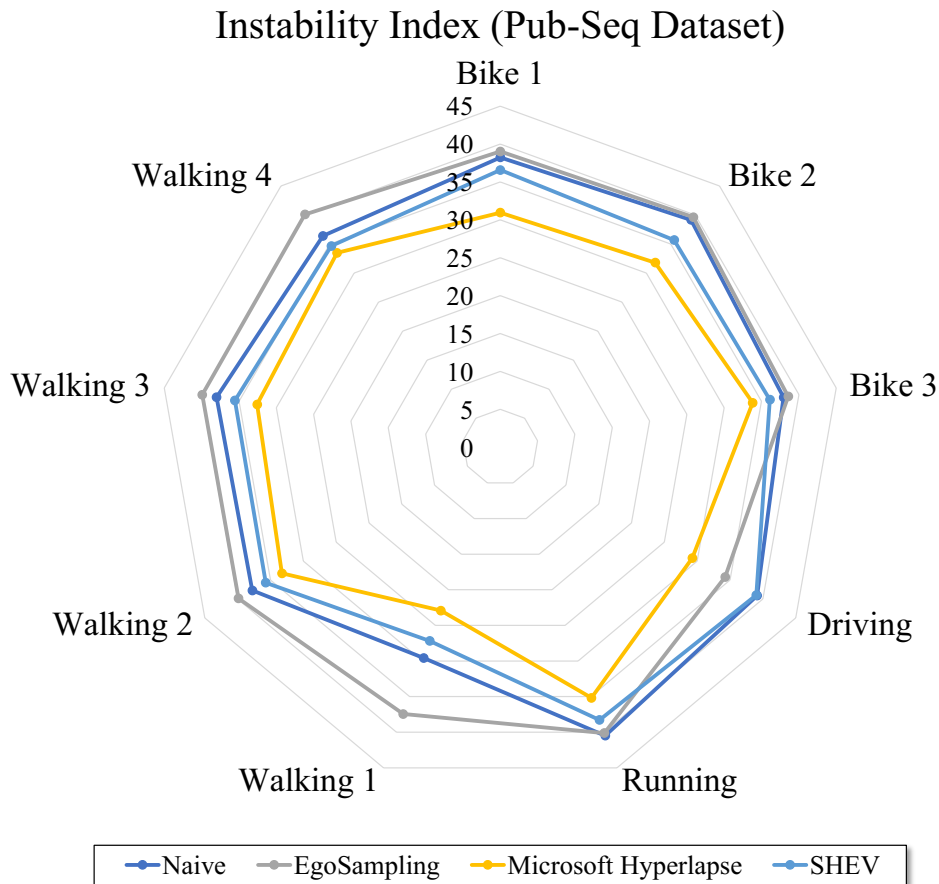


**Figure 4.5.** Frames of the ‘Driving’ video, a failure case. The top left corner frame (a) depicts the forward-looking frame. The frames (b), (c), and (d) are outlier frames, which are often in this video. Removing many outlier frames causes the output video to have a high speed-up factor.

Another notable case is the speed-ups achieved by the EgoSampling technique. The mean value for the Output Speed-up of this technique in both datasets is far from ideal. We believe this is the absence of control of the speed-up rate in their graph building step. In most cases, it is more advantageous for the shortest path algorithm to select a smaller number of frames in order to reduce the overall cost.

Note that the ‘Driving’ video experiment in Table 4.7 presents the higher output speed-up rates even for the Microsoft Hyperlapse algorithm, which has the best result for this experiment. Therefore, we should consider it as an outlier. By considering this experiment as an outlier, the new mean and standard deviation values for each of the techniques drop to 21.655 and 14.199 for EgoSampling, 10.079 and 1.257 for Microsoft Hyperlapse, and 10.232 and 0.357 for ours (SHEV), what lead us to the most accurate results for the Pub-Seq Dataset.

**Instability Evaluation.** We present in Figures 4.6 and 4.7 the Instability Index values for all videos. As expected, the Microsoft Hyperlapse algorithm presents the best results once its optimization technique is entirely focused on the smoothness of the final video. Our approach presents the second-best values for smoothness in all cases, except in the ‘Driving’ video, where EgoSampling presents a smoother video. In this specific video, the EgoSampling algorithm did not allow for a speed-up rate closer to the ideal to avoid



**Figure 4.6.** Instability Index for the videos of Pub-Seq Dataset. Smaller values represent a smoother video.

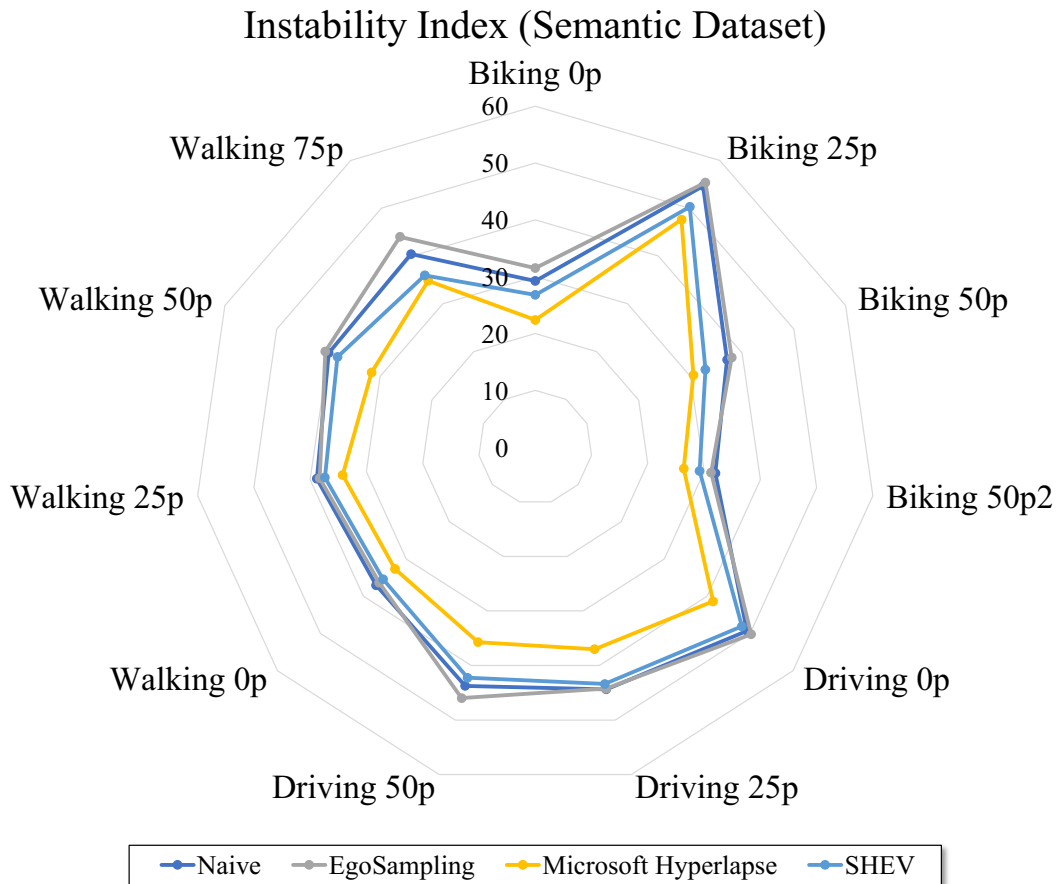
introducing shakiness into the final video.

### 4.4.3 Running Times

We measured the running times for each step of our methodology. It is important to mention that producing hyperlapse videos in real-time is not one of our objectives. However, we consider the running times a piece of relevant information for those interested in reproducing our methodology.

In the semantic fast-forwarding stage, we used either a face or a pedestrian detector as semantic extractors, which took approximately 0.7 and 1.2 seconds per input frame, respectively. These values relate to a frame with a resolution of  $1920 \times 1080$ . We ran the temporal segmentation and the speed-up estimation steps 30 times per video to obtain the average running time of these tasks. We used the PSO algorithm with 30 particles and 50 iterations to estimate our speed-up rates. The temporal segmentation step took around





**Figure 4.7.** Instability Index for the videos of Semantic Dataset. Smaller values represent a smoother video.

7 milliseconds, and the speed-up estimation task took approximately 150 milliseconds per video. The average cost to calculate the terms used in the graph building is around 600 milliseconds per input frame. Once the graph is built, we run the PSO algorithm with 8 particles and 30 iterations, which takes around 1 second per iteration. It includes the Dijkstra’s shortest path algorithm running time. These values are also considering a frame with a resolution of  $1920 \times 1080$ .

The bottleneck of our methodology is the stabilization stage. It is highly dependent on the frames selected by the fast-forward stage since the reconstruction step is expensive. The best case, which was the Biking 0p video, took around 2.4 seconds per frame in the fast-forward version, while the worst case (Driving 25p) took around 17.13 seconds per frame in the fast-forward version. The master frames definition step took on average 2.8 seconds per patch in the worst-case experiment.

#### 4.4.4 Concluding Remarks

The results acquired in our tests reveal the robustness of our method to diverse amounts of semantics. Independently of the semantic information in the input video, our method remains the one with the highest Semantic Content without degenerating the Output Speed-up. Although the Microsoft Hyperlapse achieved the best results for the smoothness, they present poor results for the Semantic Content, even worse than the Naïve approach in both tested datasets. Results also show that the ‘Driving’ video is very challenging to our methodology because it leads us to our worst results for the Output Speed-up and Instability Index metrics. The main reason for these results is the number of outlier frames, i.e., those where the driver is looking in the rear-view mirror.

#### 4.4.5 Limitations

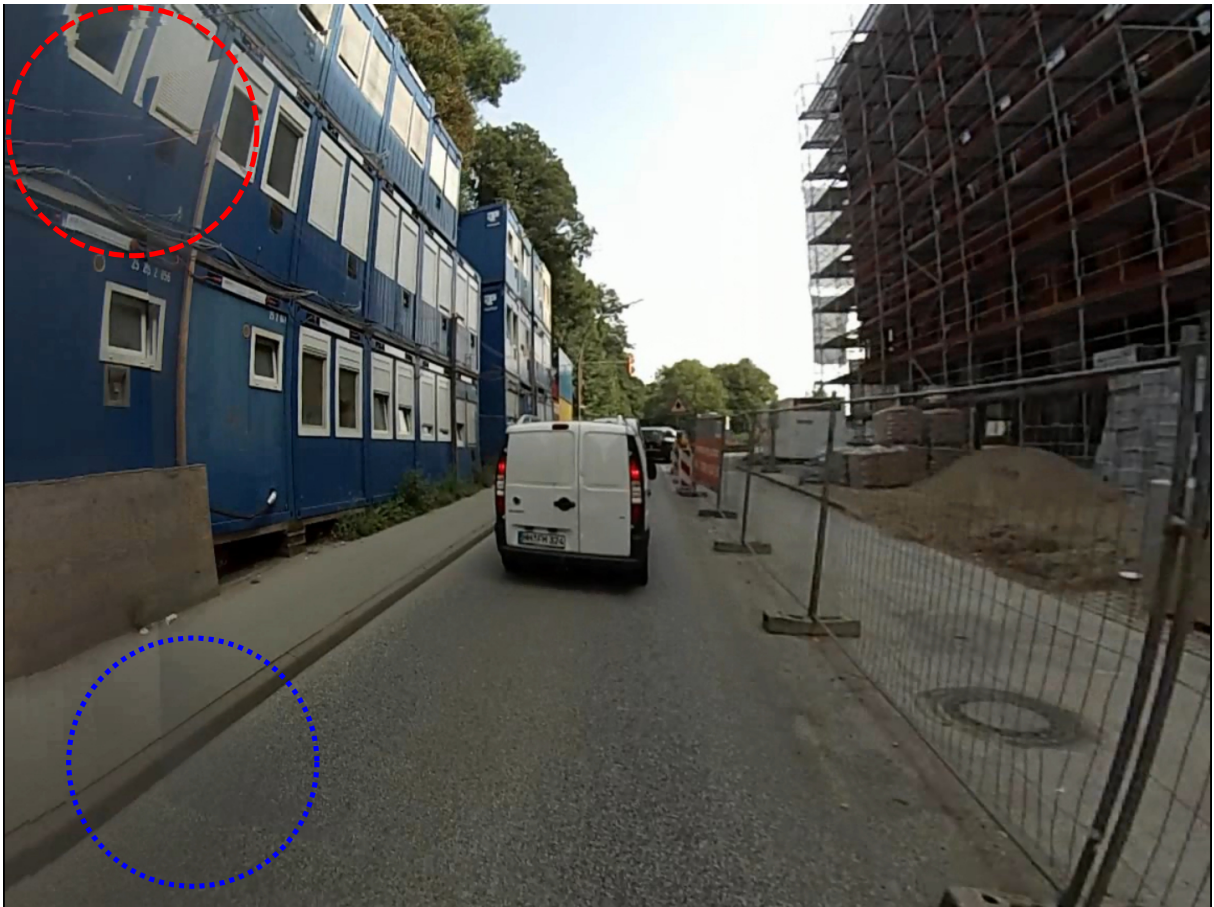
Our approach has some limitations, which we present following:

- We use in our methodology user-defined semantics in order to do the semantic extraction and define the segments. Although this method works well for specific applications, the ideal scenario for the general users would be the automatic definition of the semantics, which could be defined according to the video content.
- In our adaptive frame selection step, we assume a perfect estimation for the FOE. We use a sparse optical flow estimation proposed by Polog et al. [2014], where the images are divided into blocks, and only one vector is taken as representative for each block. This estimation is not always accurate since contradictory movements within the same block can be found. Thus, the optical flow vectors would negatively influence the FOE estimation.
- In the egocentric video stabilizer, when using a homography matrix to describe the transition from one frame to another, we are based on the assumption that the detected key-points are in the same plane on the scene, which is not always true. This leads the stitching process to present visual discontinuities since some planes do not match. Figure 4.8 shows a good match among the planes on the right side of the figure (blue dotted circle), while the planes on the left side are compromised (red dashed circle). Another example is depicted in Figure 4.9, where the bottom



**Figure 4.8.** Planes mismatches, failure case 1. The Figure depicts a frame reconstructed by the stitching process. The usage of homography restricts the correct matches for only one plane per image. The blue dotted circle presents a region where the planes match correctly. The red dotted circle presents a region where the planes do not match correctly.

left planes of the images used in the stitching match correctly (blue dotted circle), while in the upper left planes, the match is not correct (red dashed circle).



**Figure 4.9.** Planes mismatches, failure case 2. The Figure depicts a frame reconstructed by the stitching process. The usage of homography restricts the correct matches for only one plane per image. The blue dotted circle presents a region where the planes match correctly. The red dotted circle presents a region where the planes do not match correctly.

# Chapter 5

## Conclusions & Future Work

### 5.1 Conclusions

Several factors have influenced the emergence of the smooth fast-forward, or hyperlapse, methodologies in recent years. The main factor is the mass usage of wearable and mobile device cameras for life-logging since many continuous hours of video are generated for such purpose, and no edition or preprocessing is done to turn these recordings into watchable videos. Despite the great results achieved by hyperlapse methodologies, some applications need the information usually lost along the fast-forwarding process.

In this work, we were inspired by the video summarization strategies, where the goal is to create a compact summary of the video with the key components for understanding the overall content. However, unlike video summarization, we aimed to maintain the continuity of the video. To the best of our knowledge, this is the first work focused on a semantic hyperlapse.

We have presented an approach with two steps: semantic fast-forwarding and egocentric stabilization. In the first step, we split the video into semantic and non-semantic segments. For each type of segment, we calculated different speed-up rates such that the semantic segments were emphasized by a lower speed-up. The final video speed-up should achieve the speed-up required by the user. In the second step, we stabilize the video by applying homography transformations estimated from consecutive fast-forward frames. We used neighbor frames from the original video to fill the images “over-warped” by the homography transformations.

We compared our results to the state-of-the-art hyperlapse techniques. We performed the experiments in two datasets, one of them our contribution to the research community. We also contributed with a metric to measure the smoothness of egocentric videos since the most used metric in the literature does not reflect the real preference of the watchers. We measured the semantic content, speed-up achieved, and the smoothness. The results show the superiority of our approach over the state-of-the-art hyperlapse algorithms as far as the semantic information is concerned. According to the results obtained,

our method is, on average, 10.67 percentage points higher than the best method with respect to the maximum amount of semantics that can be obtained, given the required speed-up.

## 5.2 Future Works

Many possibilities arise with this new research field. Some extensions and modifications to our pipeline are:

- Use the concept of multi-importance to define different levels of semantic segments. Although we label some segments only as semantic, some parts can be even more important within the semantic segments. So, our methodology could be adapted to define multiple thresholds and play each one of them at different speed-up rates.
- Use the power of Convolutional Neural Networks (CNNs) to extract the semantic information instead of simple detectors. CNNs have been on the rise in the latest years, achieving state-of-the-art results in image classification. The usage of CNNs would automatize part of our pipeline and eventually present better results for general watchers, i.e., those who are not only looking for a specific object like faces but overall important scenes.
- Use Image/Video Captioning and Natural Language Processing (NLP) together to define the semantic segments. We could attach to the beginning of the pipeline an NLP system that would receive a sentence from the user, in natural language, that expresses her/his choice of which clip of the video she/he would like to watch in an emphasized way. That would be possible with Video Captioning algorithms, which have substantial results to define accurate labels for video clips.

# Bibliography

- Bai, C. and Reibman, A. R. (2016). Characterizing distortions in first-person videos. In *IEEE International Conference on Image Processing*, pages 2440--2444.
- Bettadapura, V., Castro, D., and Essa, I. (2016). Discovering picturesque highlights from egocentric vacation videos. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1--9.
- de Avila, S. E. F., d. Luz, A., de A. Araújo, A., and Cord, M. (2008). Vsumm: An approach for automatic video summarization and quantitative evaluation. In *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 103--110. ISSN 1530-1834.
- Dollár, P. (2016). Piotr's Computer Vision Matlab Toolbox (PMT). <https://github.com/pdollar/toolbox>.
- Elkhattabi, Z., Tabii, Y., and Benkaddour, A. (2015). Video summarization: techniques and applications. *International Journal of Computer and Information Engineering*, 9(4):928--933.
- Fathi, A., Farhadi, A., and Rehg, J. M. (2011). Understanding egocentric activities. In *IEEE International Conference on Computer Vision*, pages 407--414. IEEE.
- Fathi, A., Li, Y., and Rehg, J. M. (2012). Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, pages 314--327. Springer.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381--395.
- Gygli, M., Grabner, H., and Gool, L. V. (2015). Video summarization by learning sub-modular mixtures of objectives. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3090--3098. ISSN 1063-6919.
- Gygli, M., Grabner, H., Riemenschneider, H., and Gool, L. V. (2014). Creating summaries from user videos. In *European Conference on Computer Vision*, pages 505--520. Springer.



- Halperin, T., Poley, Y., Arora, C., and Peleg, S. (2017). Egosampling: Wide view hyperlapse from egocentric videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(5):1248--1259.
- Hsu, Y.-F., Chou, C.-C., and Shih, M.-Y. (2012). Moving camera video stabilization using homography consistency. In *IEEE International Conference on Image Processing*, pages 2761--2764.
- Ishihara, T., Kitani, K. M., Ma, W. C., Takagi, H., and Asakawa, C. (2015). Recognizing hand-object interactions in wearable camera videos. In *IEEE International Conference on Image Processing*, pages 1349--1353.
- Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., and Cohen, M. F. (2015). Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics*, 34(4):1--9.
- Kanade, T. and Hebert, M. (2012). First-person vision. *Proceedings of the IEEE*, 100(8):2442--2453.
- Karpenko, A. (2014). The technology behind hyperlapse from instagram. <http://instagram-engineering.tumblr.com/post/95922900787/hyperlapse>. Accessed: 2016-05-12.
- Karpenko, A., Jacobs, D., Baek, J., and Levoy, M. (2011). Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*, 1(2):13.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942--1948.
- Kim, G., Sigal, L., and Xing, E. P. (2014). Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4225--4232. ISSN 1063-6919.
- Kopf, J., Cohen, M. F., and Szeliski, R. (2014). First-person hyper-lapse videos. *ACM Transactions on Graphics*, 33(4):78:1--78:10. ISSN 0730-0301.
- Lee, Y. J., Ghosh, J., and Grauman, K. (2012). Discovering important people and objects for egocentric video summarization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346--1353.
- Li, Y., Fathi, A., and Rehg, J. M. (2013). Learning to predict gaze in egocentric video. In *IEEE International Conference on Computer Vision*, pages 3216--3223. ISSN 1550-5499.



- Liao, S., Jain, A. K., and Li, S. Z. (2016). A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Mach. Intelligence*, 38(2):211--223. ISSN 0162-8828.
- Lin, Y. L., Morariu, V. I., and Hsu, W. (2015). Summarizing while recording: Context-based highlight detection for egocentric videos. In *IEEE International Conference on Computer Vision Workshop*, pages 443--451.
- Lu, Z. and Grauman, K. (2013). Story-driven summarization for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714--2721. ISSN 1063-6919.
- Man, K. F., Tang, K. S., and Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, 43(5):519--534. ISSN 0278-0046.
- Matsuo, K., Yamada, K., Ueno, S., and Naito, S. (2014). An attention-based activity recognition for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 565--570. ISSN 2160-7508.
- Mei, S., Guan, G., Wang, Z., Wan, S., He, M., and Feng, D. D. (2015). Video summarization via minimum sparse reconstruction. *Pattern Recognition*, 48(2):522--533. ISSN 0031-3203.
- Okamoto, M. and Yanai, K. (2013). Summarization of egocentric moving videos for generating walking route guidance. In *Pacific-Rim Symposium on Image and Video Technology*, pages 431--442. Springer.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62--66. ISSN 0018-9472.
- Pele, O. and Werman, M. (2009). Fast and robust earth mover's distances. In *IEEE International Conference on Computer Vision*, pages 460--467. ISSN 1550-5499.
- Polatsek, P., Benesova, W., Paletta, L., and Perko, R. (2016). Novelty-based spatiotemporal saliency detection for prediction of gaze in egocentric video. *IEEE Signal Processing Letters*, 23(3):394--398. ISSN 1070-9908.
- Poleg, Y., Arora, C., and Peleg, S. (2014). Temporal segmentation of egocentric videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537--2544.
- Poleg, Y., Ephrat, A., Peleg, S., and Arora, C. (2016). Compact cnn for indexing egocentric videos. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1--9.

- Poleg, Y., Halperin, T., Arora, C., and Peleg, S. (2015). Egosampling: Fast-forward and stereo for egocentric videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768--4776.
- Ren, X. and Gu, C. (2010). Figure-ground segmentation improves handled object recognition in egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3137--3144. ISSN 1063-6919.
- Sazbon, D., Rotstein, H., and Rivlin, E. (2004). Finding the focus of expansion and estimating range using optical flow images and a matched filter. *Machine Vision and Applications*, 15(4):229--236. ISSN 1432-1769.
- Singh, S., Arora, C., and Jawahar, C. V. (2015). Generic action recognition from egocentric videos. In *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1--4.
- Wan, S. and Aggarwal, J. K. (2015). Robust object recognition in rgb-d egocentric videos based on sparse affine hull kernel. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 97--104. ISSN 2160-7508.
- Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J. M., and Singh, V. (2015). Gaze-enabled egocentric video summarization via constrained submodular maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2235--2244. ISSN 1063-6919.
- Yang, J. A., Lee, C. H., Yang, S. W., Somayazulu, V. S., Chen, Y. K., and Chien, S. Y. (2016). Wearable social camera: Egocentric video summarization for social interaction. In *IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1--6.
- Yao, T., Mei, T., and Rui, Y. (2016). Highlight detection with pairwise deep ranking for first-person video summarization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 982--990.
- Zhang, K., Chao, W.-L., Sha, F., and Grauman, K. (2016). Video summarization with long short-term memory. In *European Conference on Computer Vision*, pages 766--782. Springer.
- Zhang, S. and Roy-Chowdhury, A. K. (2015). Video summarization through change detection in a non-overlapping camera network. In *IEEE International Conference on Image Processing*, pages 3832--3836.