Semantic Hyperlapse for Egocentric Videos

Washington L. S. Ramos, Mario F. M. Campos, Erickson R. Nascimento
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
Email: {washington.ramos, mario, erickson}@dcc.ufmg.br

***Abstract*—The emergence of low-cost personal mobiles devices and wearable cameras and, the increasing storage capacity of video-sharing websites have pushed forward a growing interest towards first-person videos. Wearable cameras can operate for hours without the need for continuous handling. These videos are 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 create a shorter watchable version with no abrupt transitions between the frames. However, an important aspect of such videos is the relevance of the frames, usually ignored in *hyperlapse* videos. 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, our method is on average 10.67 percentage points higher than the second best in relation to the maximum amount of semantics that can be obtained, given the required speed-up [1]. More information can be found in our supplementary video: https://youtu.be/_TU8KPaA8aU.**

I. Introduction

Thanks to advances in technology which constantly leads 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 in social media such as video-sharing services like YouTube, and personal repositories, since they provide extensive space for storage.

Wearable devices such as GoPro HERO™, Looxcie, and Google Glass™ cameras can be operated with no intervention, thus the camera operator is free to carry out his/her activities. It opens up unprecedented ways to record many continuous hours of regular activities like walking, driving, and cooking, athletic activities (e.g. running and bicycling), and even working tasks like event recordings (e.g. weddings, proms, birthdays, etc.) and monitoring (e.g. police patrol and lifeguarding).

Problem Definition. Egocentric videos are hardly watched in their entirety because they are usually long and monotonous. Moreover, they contain shaky scene transitions due to natural body movements, causing visual discomfort [1] and difficulty on extracting information [2]. The use of simple fast-forward methods such as frame sub-sampling at a fixed rate is a naïve approach to reduce the video length since they do not require any understanding of the video content. In contrast to the creation of fast-forward videos with carefully controlled cameras, where it is easy to track the movement, in first-person videos the significant camera shake leads the fast-forward videos to be jerky since the shakiness is increased.

Several works have been proposed to tackle the instability of egocentric videos aiming to create a pleasant experience when watching the reduced version, usually called hyperlapse [3]–[6]. 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. 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 final version.

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 segment the video by selecting the set of pictures which maximizes the semantics, the required speed-up as well as the smoothness of the transition between the frames. We name our method as SHEV (**S**emantic **H**yperlapse for **E**gocentric **V**ideos).

Contributions. We can summarize our contributions as:

i) a new adaptive fast-forwarding approach. Our method segments the input video into relevant and non-relevant parts and, it builds 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;
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, which is able to measure the egocentric videos smoothness.

II. Related Work

Egocentric Video Summarization. Regular summarization strategies are hard to be applied to the egocentric video

[1]This work relates to an M.Sc. dissertation.

summarization task, once 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 [7]. Probably, the works most related to ours in this category are the work of Okamoto and Yanai [8], and the work of Yao *et al.* [9].

The Okamoto and Yanai's methodology generates walking route guidance videos by summarizing egocentric videos. They utilize ego-motion and pedestrian crosswalk to estimate the importance of each video section. Unlike most summarization methods, they 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 to the user a nice and smooth experience when watching the fast-forward version.

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 composition of skims or a video timelapse. 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 a 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.

Hyperlapse. Recent efforts to create smooth fast-forward egocentric videos 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 frame set that generates a smoother final video.

A representative method in the 3D model reconstruction category is the work of Kopf *et al.* [3]. 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. Image-based rendering techniques are used 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 path. The technique also requires camera motion and parallax to compute the 3D model of the scene. It is noteworthy the high computational cost required by their method, which makes it unpractical. Moreover, the dynamics of the scene causes the SfM to fail.

Adaptive frame selection adjusts the density of the frame selection according to the cognitive load. For instance, a denser selection could be done when the scene motion is too high and, in turn, a sparser selection could be done when the camera wearer is stopped. The works of Joshi *et al.* [5] and Poleg *et al.* [4] are recent examples of this category.

Joshi *et al.* present a real-time method to create a hyperlapse video. 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 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. They compute the shortest path to find the best frames to compose the hyperlapse. Halperin *et al.* [6] 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 solutions mentioned above 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 semantic in regions of the scene. Therefore, they are removed from resulting video.

III. Methodology

We divide our methodology into two major steps: semantic fast-forwarding and semantic egocentric stabilization. In the first step, the algorithm seeks the input video frames that maximize the semantic content, the smoothness and the proximity to the required speed-up. In the second step, homography transformations are used to align the frames transitions. Then, an iterative stitching process is responsible for filling the frames that were excessively distorted by the transformations.

A. Semantic Egocentric Fast-Forwarding

In this section, we present the first step of our methodology, which is composed of four sub-steps detailed as follows.

1) Semantic Extraction: In the first step of our sampling approach, we extract the semantic information present in each frame of the video according to the semantic selected by the user (e.g. pedestrian, face, car plate, etc.). The semantic information is encoded by the score function $S : \mathbb{R} \rightarrow \mathbb{R}$, given by $S_x = \sum_{k \in f_x} c_k \cdot a_k \cdot G_\sigma(k)$, where c_k is the normalized confidence of the extractor for the region of interest (ROI) k and a_k is the normalized area of the k-th ROI in pixels. $G_\sigma(k)$ is the value of the central point of the k-th ROI in the Gaussian function with standard deviation σ and centered at the frame f_x. This function returns higher values to more centralized objects. Examples are illustrated in the Figure 1-A.

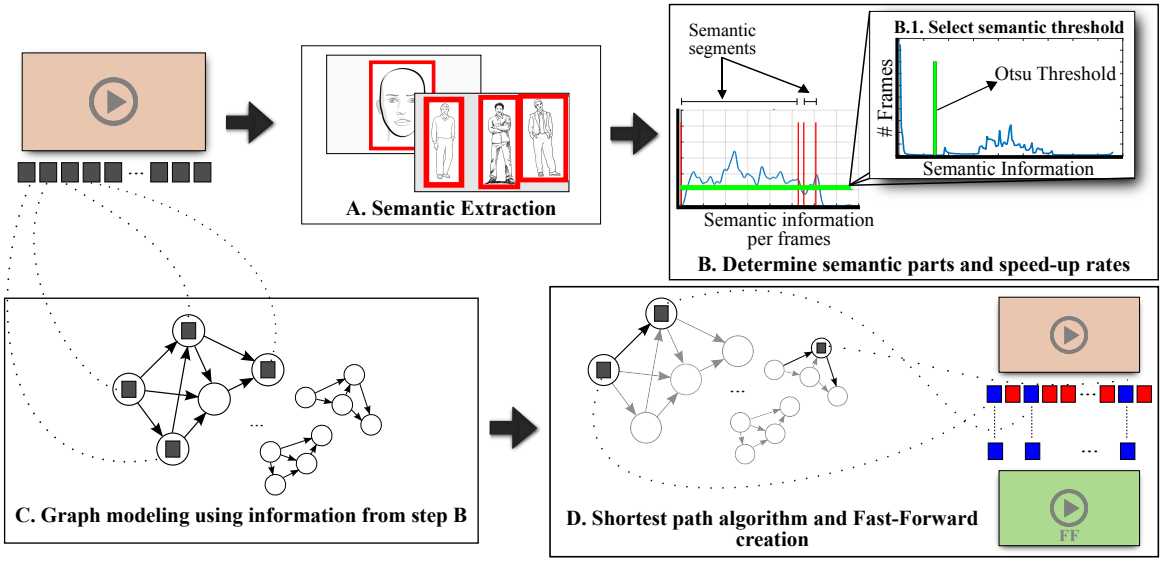

Fig. 1. Overall steps of our semantic adaptive frame sampling process. From the input video, we extract ROIs containing the semantic information (A) in each frame and compute the semantic scores to define the semantic profile (B). We use the Otsu thresholding method to find a meaningful semantic threshold (B.1) in order to identify the semantic segments and to 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).

2) Temporal Segmentation: The semantic score along the frames defines the semantic profile of the video as illustrated in Figure 1-B. We create a histogram of the semantic scores and apply the Otsu thresholding method to find the threshold that better define the disparity between the semantic and non-semantic frames. The value returned by Otsu (green line in Fig. 1-B.1) is used as the semantic threshold. Thus, every frame above this value is labeled as relevant.

3) Speedup Rate Estimation: 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 in 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 following equation: $D(F_{ns}, F_s) = \left| \frac{L_s + L_{ns}}{F_d} - \left(\frac{L_s}{F_s} + \frac{L_{ns}}{F_{ns}} \right) \right|$. Note that, for every F_s there is a correspondent F_{ns} that leads the result to 0. We solve it by restricting their values so that the F_s is minimized as well as the difference between both.

We also add some space restrictions: (i) $F_s \leqslant F_d$, once we want more emphasis in the semantic parts; (ii) $F_{ns} \geqslant F_d$, once we want to achieve desired speed-up in the fast-forward video and; (iii) $F_s \geqslant p_s F_d$, 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 1:

$$\underset{F_s,\, F_{ns}}{\arg\min}\, D\left(F_{ns}, F_s\right) + \lambda_1 |F_{ns} - F_s| + \lambda_2 |F_s|, \qquad (1)$$

where λ_1 and λ_2 are the regularization terms that give more importance either to keep the speed-up rates close or take the smaller F_s.

4) Graph Building: We model each video segment using a weighted graph similar to Poleg *et al.* [4] and Halperin *et al.* [6]. 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 τ_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 τ_{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 ($I_{i,j}$), speed of motion ($V_{i,j}$), appearance change ($A_{i,j}$) and semantic gain/loss ($S_{i,j}$) caused by the transition.

The first three terms were previously proposed by Poleg *et al.* and Halperin *et al.* in their graph construction. The semantic cost is given by: $S_{i,j} = 1/(S_i + S_j + \epsilon)$, where S_x is the semantic score of the frame f_x. The value ϵ 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:

$$W_{i,j} = (\lambda_I \cdot I_{i,j} + \lambda_V \cdot V_{i,j} + \lambda_A \cdot A_{i,j} + \lambda_S \cdot S_{i,j}) \cdot \left\lceil \frac{(j-i)}{F} \right\rceil, \qquad (2)$$

where the values of λ coefficients are the regularization factors for each one of the costs terms. We add a proportional factor to enhance transitions between frames with lower distance, where

$F \in \{F_s, F_{ns}\}$ is the speed-up rate applied to the graph which the edge $E_{i,j}$ belongs.

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 compose the final fast-forward video.

B. Egocentric Video Stabilization

The frames selected in the previous major step are subject to a stabilization process which consists of three sub-steps.

1) Master frames definition: The first step of the stabilization methodology consists of segmenting the video into temporal patches of length α and selecting one master frame M_k for each patch. We select as the master of the k-th patch, the frame M_k in this patch that maximizes the Equation 3:

$$\arg\max_{M_k \in p_k} \sum_{f_i \in p_k} R(f_i, M_k), \tag{3}$$

where p_k is the k-th patch and the f_i is the i-th frame of the fast-forward video. The function $R(x, y)$ calculates the number of *inliers* in the RANSAC method when computing the homography transformation from the image x to y.

2) Transition smoothing: The second step is to smooth the transitions between the selected 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 temporally closer to f_i, s.t. $b < i$. Analogously, $M_{pos} = f_a$ stands for the posterior master frame, 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 ($\widehat{f_i}$) is given by $\widehat{f_i} = H^{1-w}_{f_i,M_{pre}} \cdot H^{w}_{f_i,M_{pos}} \cdot f_i$. The term $H^p_{x,y}$ 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 δ is the temporal distance from the frame f_i to M_{pre}, and Δ is the distance between M_{pre} and M_{pos}.

3) Frames reconstruction: As expected, after applying the homography transformations, black areas are generated because the camera movements are abrupt and the elapsed time between consecutive frames in fast-forward videos are large. Thus, the last step is to reconstruct corrupted these regions.

To reconstruct these frames, we first define two static image areas centered in the frame: i) the drop area (da) equals to $dp\%$ size of the frame, which is the area where the viewer focuses on the majority of the time and; ii) the crop area (ca) equals to $cp\%$ size of the frame, which defines the boundary for the peripheral vision. The area da has a smaller size compared to da, therefore $cp > dp$. Regions outside the ca area are removed in the final video.

The reconstruction procedure is an iterative process. Firstly, we check if the warped frame $\widehat{f_i}$ covers ca. If it does, the frame is ready to compose the final video. We drop the warped frames that do not cover da. We apply stitching in the frames that cover da but do not cover ca. The stitching process is performed as follows. We use the SURF detector to select feature points in the frame $\widehat{f_i}$ and in the j-th frame dropped from the original video, d_j. To calculate the homography transformation matrix we match feature points between the images by describing all feature points of d_j and $\widehat{f_i}$ with SURF and applying the brute force matching strategy. Given the matched points, we calculate the homography matrix $H_{d_j,\widehat{f_i}}$ using RANSAC. The $\widehat{d_j} = H_{d_j,\widehat{f_i}} \cdot d_j$ is now aligned and stitched with $\widehat{f_i}$ to compose the reconstructed image. The process ends when the reconstructed frame covers ca or the number of frames used for reconstruction is too large.

Whenever we drop a frame in the reconstruction process, we select a new frame d_j that belongs to the interval $[f_{i-1}, f_{i+1}]$ in the original video and maximizes the Equation 4:

$$\arg\max_{d_j} (G_\sigma(p)(R(d_j, \widehat{f}_{i-1}) + R(d_j, \widehat{f}_{i+1}))(\eta + S(d_j))), \tag{4}$$

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

IV. Experiments

Datasets. We used two datasets to conduct our experiments. The first is the Pub-Seq Dataset, which is a collection of publicly available videos that were previously used by other authors to evaluate their hyperlapse methods: Bike 1, Bike 2, Bike 3, Walking 1 and Walking 2 [3]; Running, Driving and Walking 3 [4] and; Walking 4 [10]. We propose a new labeled dataset to run the experiments and validate our methodology since no semantically controlled egocentric datasets were found in the literature, the Semantic Dataset.

The Semantic Dataset is composed of 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, where the number in the pattern <number>p indicates the percentage of semantic information in the videos. The videos are: Biking 0p, Driving 0p and Walking 0p (0%); Biking 25p, Driving 25p and Walking 25p (25%); Biking 50p, Biking 50p2, Driving 50p and Walking 50p (50%) and; Walking 75p (75%). The complete dataset, including videos and the semantic labels, are publicly available to the research community [2].

Evaluation Metrics. We quantified the performance of the evaluated methodologies according to the following metrics: (i) Semantic Content, which is the sum of the semantic scores over the frames of the output video; (ii) Output Speed-up, which is the rate of acceleration achieved in the output video and; (iii) Instability Index, which is a metric that we devised based on a user study and inspired by the qualitative comparison made by Joshi *et al.* [5]. We used it to quantify

[2]http://www.verlab.dcc.ufmg.br/fast-forward-video-based-on-semantic-extraction/dataset

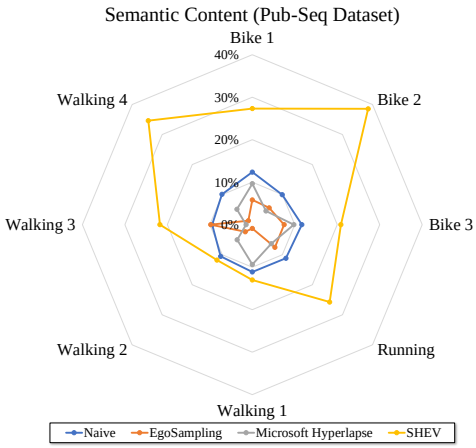

Fig. 2. Semantic Content for the videos in Pub-Seq Dataset. Results for the 'Driving' video were removed, once this video has no semantic information. Our method (SHEV) is on average 11.88 percentage points better than the Naïve approach, which has the higher average semantic content.

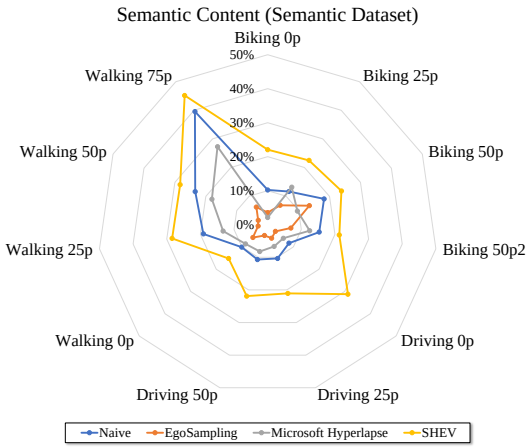

Fig. 3. Semantic Content for the videos in Semantic Dataset. Our method (SHEV) is on average 9.46 percentage points better than the Naïve approach, which has the higher average semantic content.

the shakiness in the output video. The shakiness estimation is computed as in the Equation 5:

$$I = M\left(\frac{1}{N}\cdot\sum_{i=1}^{N}\frac{\sum_{j\in B_i}(f_j-\bar{f}_i)^2}{(N_B-1)}\right), \quad (5)$$

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.

Parameters Setup. Most parameters were defined empirically. In our semantic fast-forwarding methodology, we used as the semantic extractors the Liao *et al.*'s NPD Face Detector [11], which is the current state of the art, in videos where the wearer is walking and, the Piotr Dollár's pedestrian detector [12] in videos where the motion speed is higher. We considered any $c_k < 60$ as false face detections and $c_k < 100$ as false pedestrian detections. For the construction of the graph, we set the values of the border frames τ_b and the maximum allowed skip τ_{max} to be 1 and 100, respectively. To achieve better visual results, we optimized our λ parameters (in Eqs. 1 and 2) by using Particle Swarm Optimization (PSO). 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 σ of the Gaussian function in the Equation 4 and the value of η in the same equation were defined as $\sigma = 10$ and $\eta = 0.5$.

Results and Discussions. 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; (ii) *EgoSampling* (ES) [4], which creates a video by using the Poleg *et al.*'s technique with parameters defined according to the best values of their work and; (iii) *Microsoft Hyperlapse* (MH) [5], where we used the released desktop version of their algorithm to create the videos.

Figures 2 and 3 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 with relation to the maximum semantic content that could be achieved given the desired speed-up. Our method outperforms all other methodologies as far as the semantic information is concerned. 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, once in addition to the reduction of the speed-up factor in semantic segments, the semantic term balances the selection in non-semantic segments.

We measured the output speed-up for videos in both datasets with a desired speed-up $F_d = 10$. We omitted the results for Naïve technique since it always achieves the required speed-up. In the Pub-Seq Dataset, the respective mean and standard deviation values reported were: 25.617 and 17.823 for the ES algorithm; 10.212 and 1.241 for MH and; 11.762 and 4.602 for ours (SHEV). The MH algorithm is the most accurate since it presents the smallest standard deviation and the mean value which is the closest to F_d. A failure case that led our algorithm to a higher mean is the 'Driving' output video. This is a challenging video where the driver often alternates between looking ahead and looking in the left rear-view mirror. This leads to larger frame skips aiming to eliminate the outlier frames.

In the Semantic Dataset, the values reported were: 20.974 and 6.979 for the ES algorithm; 9.264 and 1.319 for MH and; 10.347 and 0.754 for ours (SHEV). In this case, our technique produces the hyperlapse videos with the speed-up closest to the desired one. The usage of the PSO algorithm to optimize

TABLE I
INSTABILITY INDEX IN THE PUB-SEQ DATASET (BEST IN BOLD)

Video	Naive	EgoSampling	Microsoft Hyperlapse	SHEV
Bike 1	38,24	39,02	**30,95**	36,58
Bike 2	39,15	39,62	**31,79**	35,68
Bike 3	37,95	38,58	**33,81**	36,12
Driving	39,13	34,25	**29,25**	39,00
Running	40,48	40,12	**35,18**	38,28
Walking 1	29,58	37,45	**22,92**	27,18
Walking 2	37,76	39,87	**33,26**	35,73
Walking 3	38,00	39,92	**32,57**	35,56
Walking 4	36,39	40,09	**33,49**	34,67

TABLE II
INSTABILITY INDEX IN THE SEMANTIC DATASET (BEST IN BOLD)

Video	Naive	EgoSampling	Microsoft Hyperlapse	SHEV
Biking 0p	29,26	31,52	**22,39**	26,81
Biking 25p	54,61	55,36	**47,60**	50,28
Biking 50p	37,09	38,00	**30,59**	32,91
Biking 50p2	32,00	31,25	**26,39**	29,20
Driving 0p	49,30	50,24	**41,38**	48,09
Driving 25p	44,36	44,24	**37,03**	43,39
Driving 50p	43,74	45,98	**35,72**	42,24
Walking 0p	37,05	36,34	**32,66**	35,43
Walking 25p	38,81	38,31	**34,23**	37,38
Walking 50p	39,93	40,60	**31,67**	38,24
Walking 75p	40,40	44,01	**34,82**	35,95

the λ's is the main factor for such results, once the λ's control the selected speed-ups and the weights for the graph terms.

Tables I and II present the Instability Index of the output videos produced by the methodologies. As expected, the MH 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 ES presents a smoother video. In this specific video, the ES algorithm did not allow for a speed-up rate closer from the ideal to avoid introducing shakiness into the final video.

Limitations. We use in our methodology a user-defined semantics to extract semantic and define the segments. Although this method works well for certain applications, the ideal scenario for the general users would be the automatic definition of the semantics where it could be defined according to the video content. In addition, in the stabilization step, when using homography matrix to describe the transition from one frame to another, we are based on the assumption that the detected keypoints are in the same plane on the scene, which is not always true. This leads the stitching process to present visual discontinuities, once some planes do not match.

V. CONCLUSIONS

In this work, we presented a new approach for producing hyperlapse videos focusing on the semantic content. In the first step of our methodology, we split the video into semantic and non-semantic segments and calculate different speed-up rates such that the semantic segments were emphasized by a lower speed-up. In the second step, we stabilize the video by applying homography transformations estimated from consecutive fast-forward frames to generate the final hyperlapse video.

VI. AWARDS & PUBLICATIONS

Part of this work was published on the 2016 IEEE International Conference on Image Processing (ICIP) [13] and on the First International Workshop on Egocentric Perception, Interaction and Computing at European Conference on Computer Vision (EPIC@ECCV) 2016 [14]. A journal extension of this is under review in the Journal of Visual Communication and Image Representation (JVCI). This work has also been awarded as the best master's work presented in the Week of Graduate Seminars held by the computer science department (DCC-UFMG).

REFERENCES

[1] C. Bai and A. R. Reibman, "Characterizing distortions in first-person videos," in *Image Processing (ICIP), 2016 IEEE International Conference on*. Phoenix, AZ, USA: IEEE, Sep 2016, pp. 2440–2444.

[2] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, "Compact cnn for indexing egocentric videos," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2016, pp. 1–9.

[3] J. Kopf, M. F. Cohen, and R. Szeliski, "First-person hyper-lapse videos," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 78:1–78:10, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601195

[4] Y. Poleg, T. Halperin, C. Arora, and S. Peleg, "Egosampling: Fast-forward and stereo for egocentric videos," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015, pp. 4768–4776.

[5] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen, "Real-time hyperlapse creation via optimal frame selection," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 63:1–63:9, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766954

[6] T. Halperin, Y. Poleg, C. Arora, and S. Peleg, "Egosampling: Wide view hyperlapse from egocentric videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[7] Y. L. Lin, V. I. Morariu, and W. Hsu, "Summarizing while recording: Context-based highlight detection for egocentric videos," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 443–451.

[8] M. Okamoto and K. Yanai, *Summarization of Egocentric Moving Videos for Generating Walking Route Guidance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 431–442. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-53842-1_37

[9] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 2537–2544.

[11] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 38, no. 2, pp. 211–223, Feb. 2016.

[12] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," https://github.com/pdollar/toolbox, May 2016.

[13] W. L. S. Ramos, M. M. Silva, M. F. M. Campos, and E. R. Nascimento, "Fast-forward video based on semantic extraction," in *IEEE International Conference on Image Processing*, Sept 2016, pp. 3334–3338.

[14] M. M. Silva, W. L. S. Ramos, J. P. K. Ferreira, M. F. M. Campos, and E. R. Nascimento, "Towards semantic fast-forward and stabilized egocentric videos," in *European Conference on Computer Vision Workshops*. Amsterdam, NL: Springer International Publishing, October 2016, pp. 557–571.