Personalized Cinemagraphs using Semantic Understanding and Collaborative Learning*

Tae-Hyun Oh[1,2†] Kyungdon Joo[2†] Neel Joshi[3] Baoyuan Wang[3] In So Kweon[2] Sing Bing Kang[3]
[1]MIT CSAIL, Boston, MA [2]KAIST, South Korea [3]Microsoft Research, Redmond, WA

Abstract

Cinemagraphs are a compelling way to convey dynamic aspects of a scene. In these media, dynamic and still elements are juxtaposed to create an artistic and narrative experience. Creating a high-quality, aesthetically pleasing cinemagraph requires isolating objects in a semantically meaningful way and then selecting good start times and looping periods for those objects to minimize visual artifacts (such a tearing). To achieve this, we present a new technique that uses object recognition and semantic segmentation as part of an optimization method to automatically create cinemagraphs from videos that are both visually appealing and semantically meaningful. Given a scene with multiple objects, there are many cinemagraphs one could create. Our method evaluates these multiple candidates and presents the best one, as determined by a model trained to predict human preferences in a collaborative way. We demonstrate the effectiveness of our approach with multiple results and a user study.

1. Introduction

With modern cameras, it is quite easy to take short, high resolution videos or image bursts to capture the important and interesting moments. These small, dynamic snippets of time convey more richness than a still photo, without being as heavyweight as a longer video clip. The popularity of this type of media has spawned numerous approaches to capture and create them. The most straightforward methods make it as easy to capture this imagery as it is to take a photo (e.g., Apple Live Photo). To make these bursts more compelling and watchable, several techniques exist to stabilize (a survey can be found in [33]), or loop the video to create video textures [29] or "cinemagraphs" [1], a media where dynamic and still elements are juxtaposed, as a way to focus the viewer's attention or create an artistic effect.

The existing work in the space of cinemagraph and live image capture and creation has focused on ways to ease user burden, but these methods still require significant user control [2, 17]. There are also methods that automate the creation of the loops such that they are the most visually seamless [23], but they need user input to create aesthetic effects such as cinemagraphs.

We propose a novel, scalable approach for automatically creating semantically meaningful and pleasing cinemagraphs. Our approach has two components: (1) a new computational model that creates meaningful and consistent cinemagraphs using high-level semantics and (2) a new model for predicting person-dependent interestingness and visual appeal of a cinemagraph given its semantics. These two problems must be considered together in order to deliver a practical end-to-end system.

For the first component, our system makes use of semantic information by using object detection and semantic segmentation to improve the visual quality of cinemagraphs. Specifically, we reduce artifacts such as whole objects being separated into multiple looping regions, which can lead to tearing artifacts.

In the second component, our approach uses semantic information to generate a range of candidate cinemagraphs, each of which involves animation of a different object, *e.g.*, tree or person, and uses a machine learning approach to pick which would be most pleasing to a user, which allows us to present the most aesthetically pleasing and interesting cinemagraphs automatically. This is done by learning a how to rate a cinemagraph based on interestingness and visual appeal. Our rating function is trained using data from an extensive user study where subjects rate different cinemagraphs. As the user ratings are highly subjective, due to individual personal preference, we propose a collaborative filtering approach that allows us to generalize preferences of sub-populations to novel users. The overall pipeline of our system is shown in Fig. 1.

In summary, our technical contributions include: (1) a novel algorithm for creating semantically meaningful cinemagraphs, (2) a computational model that learns to rate (i.e., predict human preference for) cinemagraphs, and (3) a collaborative filtering approach that allows us to generalize and predict ratings for multiple novel user populations.

*[*] *Acknowledgment* We would like to thank all the participants in our user study. We are also grateful to Jian Sun and Jinwoo Shin for the helpful discussions. This work was mostly done while the first author was an intern at Microsoft Research, Redmond. It was completed at KAIST with the support of the Technology Innovation Program (No. 10048320), which is funded by the Korean government (MOTIE).

[†] The first and second authors contributed equally to this work.

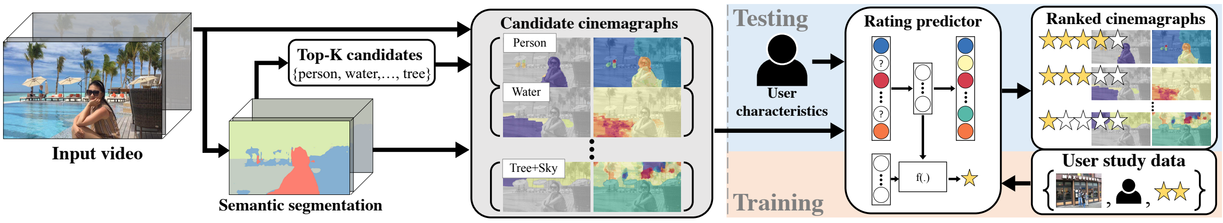


Figure 1: Overview of our semantic aware cinemagraph creation and suggestion system: 1) applying a semantic segmentation on the input video to recover semantic information, 2) selecting top-K candidate objects, each of which will be dynamic in a corresponding candidate cinemagraph, 3) solving semantic aware Markov Random Field (MRF) for multiple candidate cinemagraph generation (Sec. 3). 4) selecting or ranking the best candidate cinemagraphs by a model learned to predict subjective preference from a database we acquire of user preferences for numerous cinemagraphs in an off-line process (Sec. 4).

2. Related Work

There is a range of types of imagery that can be considered a "live image", "live photo", or "living portrait". In this section, we briefly survey techniques for creating these types imagery, categorized roughly as video textures (whole frame looping), video looping (independent region looping), and content-based animation (or cinemagraphs).

Video Textures Video textures [29, 20, 24, 10] refer to the technique of optimizing full-frame looping given a short video. It involves the construction of a frame transition graph that minimizes appearance changes between adjacent frames. While the above methods are restricted to frame-by-frame transition of a video, the notion of video re-framing has inspired many video effect applications, *e.g.*, independent region-based video looping and cinemagraphs.

Video Looping Liao *et al.* [23] developed an automatic video-loop generation method that allows independently looping regions with separate periodicity and starting frames (optimized in a follow-up work [22]). The representation used in [22, 23] conveys a wide spectrum of dynamism that a user can optionally select in the generated video loop. However, the output video loop is generated without any knowledge of the scene semantics; the dynamics of looping is computed based on continuity in appearance over space and time. This may result in physically incoherent motion for a single object region (e.g., parts of a face may be animated independently). Our work builds directly on these approaches, by incorporating semantic information into cost functions.

Interactive Cinemagraph Creation The term "cinemagraph" was coined and popularized by photographer Jamie Beck and designer Kevin Burg [1], who used significant planning and still images shot with a stationary camera for creating cinemagraphs.

A number of interactive tools have been developed to make it easier to create cinemagraphs [35, 17, 2]. These approaches focus on developing a convenient interactive representation to allow user to composite a cinemagraph by manual strokes. Commercial and mobile apps such as Microsoft Pix, Loopwall, Vimeo's Echograph[1] and Flixel's Cinemagraph Pro[2] are also available, with varying degrees of automation. The primary difference between all these previous works and ours is that user input is not necessary for our method to create a cinemagraph effect.

Automatic and Content-based Creation Closely related to our work are techniques that perform automatic cinemagraph creation in a restricted fashion [40, 7, 39, 3, 30].

Bai *et al.* [3] track faces to create portrait cinemagraphs, while Yeh *et al.* [40, 39] characterize "interestingness" of candidate regions using low-level features such as cumulative motion magnitudes and color distinctness over subregions. More recently, Sevilla-Lara *et al.* [30] use nonrigid morphing to create a video-loop for the case of videos having a contiguous foreground that can be segmented from its background. Yan *et al.* [38] create a cinemagraph from a video (captured with a moving camera) by warping to a reference viewpoint and detecting looping regions as those with static geometry and dynamic appearance.

By comparison, our method is not restricted to specific target objects; we generate a cinemagraph as part of an optimization instead of directly from low-level features or very specific objects (*e.g.*, faces [3]). Our approach is to produce independent dynamic segments as with Liao *et al.* [23], but we encourage them to correspond as much as possible with semantically clustered segments. Given the possible candidates, each with a different looping object, we select the best cinemagraph by learned user preferences.

Rating of Videos and Cinemagraphs There are a few approaches to rank or rate automatically-generated videos. Gygli *et al.* [15] propose an automatic GIF generation method from a video, where it suggests ranked segments from a video in an order of popularity learned from GIFs on the web; however, their method does not actually *generate* an animated GIF or a video loop. Li *et al.* [21] create a benchmark dataset and propose a method to rank animated GIFs, but do not create them. Chan *et al.* [7] rank scene "beauty" in cinemagraphs based on low-level information

[1] https://vimeo.com/echograph

[2] http://www.flixel.com/

(the size of the region of interest, motion magnitude, and duration of motion). We are not aware of any work that rates cinemagraphs based on user and high-level visual contexts.

3. Semantic Aware Cinemagraph Generation

A semantic segmentation of the scene allows us to model semantically meaningful looping motion in cinemagraph. In the following sections, we describe how we extract the semantic information for cinemagraph, and then how we instill it into an MRF optimization.

Note that throughout this paper, we assume that the input video is either shot on a tripod or stabilized using off-the-shelf video stabilization (*e.g.*, Adobe After Effect). Due to the space limit, we present details, *e.g.*, implementation, all the parameter values we used and setups if not specified, in the supplementary material.

3.1. Semantic Information Extraction

Semantic Segmentation We use semantic segmentation responses obtained from a model, FCN-8 [25][3] learned with PASCAL$_{\text{Context}}$[27], which predicts 60-classes per pixel. We run it on each frame of the input video independently, which forms the semantic response $F \in [0,1]^{C\times S\times T}$, where C, S and T denote the numbers of channels (or semantic category), spatial pixels and input video frames, respectively.

Empirically, we found that using macro-partitioned semantic categories causes over-segmentation, which is often undesirable for cinemagraph generation. We re-define categories that exhibit different types of cinemagraph motions and alleviate FCN's imperfect prediction that are easily confused by FCN. We combined some categories to generate a smaller number of representative higher-level categories which are roughly classified by similar semantics as well as similar cinemagraph motion characteristics, *e.g.*, {ground, floor, sidewalk} to be in *background*. We reduced the number of categories from 60 to 32 including background (C=32); all these mapping of categories are listed in the supplementary.

Top-K Candidate Label Selection Unfortunately, this 32-dimensional (in short, dim.) feature introduces significant computational complexity in subsequent optimization. To reduce the complexity and memory usage, we only store the top-K class responses to form semantic response. These top-K classes are used in the optimization described later and determining what objects should be dynamic (*i.e.*, looping) in each candidate cinemagraph.

We select the top-K by the number of pixels associated with each category with simple filtering. The procedure to select candidate objects is as follows:

1. Given $F \in [0,1]^{C\times S\times T}$, construct a global histogram $h_g(c) = \sum_{x,t} \delta\left[c = \operatorname{argmax}_{c'} F(c', x, t)\right]$, where $\delta[\cdot]$ denotes the indicator function to return 1 for true argument, otherwise 0,
2. Discard classes from h_g that satisfy the following criteria:
   (a) Static object categories with common sense (*i.e.*, objects that do not ordinarily move by themselves, such as roads and buildings. The full lists are in the supplementary),
   (b) Object classes of which the standard deviation of intensity variation across time is ≤ 0.05 (*i.e.*, low dynamicity),
   (c) Object classes of which connected component blob sizes are too small on average ($\leq 20\times 20$ pixels),
3. Pick top-K labels which are K highest values in the histogram h_g, and with this, pick the channel dim. of F to be K as $F \in [0,1]^{K\times S\times T}$. We set $K = 4$.[4]

Spatial Candidate Map π Given top-K candidate objects, we maintain another form of candidate information that allows our technique to decide which regions should appear as being dynamic in each candidate cinemagraph.

We use a rough per-pixel binary map $\pi_i \in \{0,1\}^S$ for each category i. Let $m[\cdot]:\{1,\cdots,K\}\rightarrow\{1,\cdots,C\}$ be the mapping from an index of the top-K classes to an original class index. Then, we compute $\pi_{m[k]}$ by thresholding the number of occurrences of the specified candidate object k across time as $\pi_{m[k]}(x) = \delta\left[h_t(k,x) \geq thr.\right]$, where $h_t(k,x) = \sum_t \delta\left[k = \operatorname{argmax}_{k'} F(k', x, t)\right]$ is a histogram across the temporal axis. The candidate region information from π is propagated through subsequent MRF optimization.

3.2. Markov Random Field Model

Our MRF model builds on Liao *et al.* [23]. Their approach solves for an optimal looping period p_x and start frame s_x at each pixel, so that the input RGB video $\vec{V}(x,t) \in [0,1]^3$ is converted to an endless video-loop $\vec{L}(x,t) = \vec{V}(x, \phi(x,t))$ with a time-mapping function $\phi(x,t) = s_x + (t - s_x) \bmod p_x$. Following this formulation, we formulate the problem as 2D MRF optimization.

Liao *et al.*'s approach uses terms to minimize color difference between immediate spatiotemporal pixel neighbors, but it does not incorporate any high level information. Thus, while the resulting loops are seamless in terms of having minimal color differences of neighbors, it is common that the resulting video loops have significant artifacts due to the violation of semantic relationships in the video, *e.g.*, parts of objects like animal or person are often broken apart in resulting video-loops, which looks unnatural and awkward.

We extend the method of Liao *et al.* such that semantic consistency is also considered in the energy terms of the optimization along with photometric consistency. In addition to creating results that have fewer semantic-related artifacts, we use the semantically meaningful segments to create a variety of cinemagraph outputs where we can control

[3] We explain with FCN as a reference in this work, but it can be seamlessly replaced with an alternative one and all the technical details remain same.

[4] It has practical reasons: (1) A multiple of 4 allows word alignment for the memory bus. Bus transfer speed is important because the semantic feature vector is frequently evaluated during optimization. (2) Through many experiments, we found that four categories are enough to cover a wide range of dynamic scenes.

the dynamic/static behavior on a per-object basis. Lastly, we adaptively adjust parameters according to semantic contexts, *e.g.*, enforce greater spatial consistency for a person, and require less consistency for stochastic non-object textures such as water and grass.

Cost Function Denoting start frames $\mathbf{s} = \{s_x\}$, periods $\mathbf{p} = \{p_x\}$, and labels $\mathbf{l} = \{l_x\}$, where $l_x = \{p_x, s_x\}$, we formulate the semantic aware video-loop problem as:

$$\underset{\mathbf{s},\mathbf{p}}{\operatorname{argmin}} \sum_x \Big\{ E_{\text{temp.}}(l_x) + \alpha_1 E_{\text{label}}(l_x) + \alpha_2 \sum_{z \in \mathcal{N}(x)} E_{\text{spa.}}(l_x, l_z) \Big\}, \quad (1)$$

where $z \in \mathcal{N}(x)$ indicates neighbor pixels. The basic ideas for the label term E_{label}, spatial and temporal consistency terms $E_{\text{spa.}}$ and $E_{\text{temp.}}$ are the same with those described in [23]. However, there are significant differences in our work, *i.e.*, our semantic aware cost function.

Hyper-Classes for Semantic Aware Cost Our empirical observation is that depending on types of object motion characteristics, qualities of resulting cinemagraphs vary as mentioned above. In this regard, a single constant value for each parameter in cost function limits the extent of its applicability. To allow the object specific adaptation, we control the dynamicity of resulting loops according to the class.

Assigning object dependent parameters for all the classes leads to parameter tuning on the high dimension parameter space, which is challenging. As a trade-off, we use another set of hyper-class by simply classifying C-classes into *natural* / *non-natural* texture to encourage the diversity of loop labels or to synchronize loop labels, respectively. The natural set $\mathcal{H}_{\text{nat.}}$ denotes the objects like tree, water, grass, waterfall, *etc*., which are natural objects that have textual motion easily loopable and generally require less spatial coherence. The non-natural set $\mathcal{H}_{\text{non.}}$ denotes the objects like a person, animal, car, *etc*., which have rigid or non-rigid motion and are very sensitive to incoherence. The full natural and non-natural category list is in the supplementary. The separation into “natural” and “non-natural” empirically allows us to enjoy few parameters but enough adaptation effectively.

Temporal consistency term Both consistency terms incorporate semantic and photometric consistency measures. The term $E_{\text{temp.}}$ measures the consistency across the loop start frame s_x and the end frame $s_x + p_x$ as

$$E_{\text{temp.}}(l_x) = \gamma_t(x) \left[(1-w)\Phi_V(x) + w\Phi_F(x) \right], \quad (2)$$

where w is the semantic balance parameter, the temporal photometric consistency $\Phi_V(x)$ and the temporal semantic consistency $\Phi_F(x)$ are defined as follows:

$$\Phi_V(x) = \tfrac{1}{3} \begin{pmatrix} \|\vec{V}(x, s_x) - \vec{V}(x, s_x + p_x)\|^2 + \\ \|\vec{V}(x, s_x - 1) - \vec{V}(x, s_x + p_x - 1)\|^2 \end{pmatrix},$$

$$\Phi_F(x) = \tfrac{1}{K} \begin{pmatrix} \|\vec{F}(x, s_x) - \vec{F}(x, s_x + p_x)\|^2 + \\ \|\vec{F}(x, s_x - 1) - \vec{F}(x, s_x + p_x - 1)\|^2 \end{pmatrix},$$

so that the loop is not only visually loopable, but also semantically loopable. We represent the semantic response F in a vector form, $\vec{F}(x,t) \in [0,1]^K$.[5] The factor $\gamma_t(x)$ [20, 23] is defined as

$$\gamma_t(x) = 1 / \big(1 + \lambda_t(x) \, \mathrm{MAD}_{t'} \|\vec{V}(x,t') - \vec{V}(x,t'+1)\|\big). \quad (3)$$

This factor estimates temporal intensity variation at x based on the median absolute deviation (MAD). The factor $\gamma_t(x)$ slightly relaxes $E_{\text{temp.}}$ when the intensity variation is large, based on the observation that looping discontinuities are less perceptible in that case. The spatially varying weight $\lambda_t(x)$ is determined depending on semantic information as $\lambda_t(x) = 125$ if $(\vee_{i \in \mathcal{H}_{\text{nat.}}} \pi_i(x)) = 1$, where $\vee$ denotes the logical disjunction operator, otherwise it is half the value. By this, we reduce $\gamma_t(x)$ for the natural objects, as the loop discontinuity is less perceptible for the natural one.

Spatial consistency term The term $E_{\text{spa.}}$ also measures semantic and photometric consistency between neighbors as well. Specifically, $E_{\text{spa.}}$ is defined as

$$E_{\text{spa.}}(l_x, l_z) = \gamma_s(x,z) \left[(1-w)\Psi_V(x,z) + w\Psi_F(x,z) \right]. \quad (4)$$

The spatial photometric consistency $\Psi_V(x,z)$ and the spatial semantic consistency $\Psi_F(x,z)$ are defined as follows:

$$\Psi_V(x,z) = \tfrac{1}{3 \cdot \mathtt{LCM}} \sum_{t=0}^{T-1} \begin{pmatrix} \|\vec{V}(x, \phi(x,t)) - \vec{V}(x, \phi(z,t))\|^2 + \\ \|\vec{V}(z, \phi(x,t)) - \vec{V}(z, \phi(z,t))\|^2 \end{pmatrix},$$

$$\Psi_F(x,z) = \tfrac{1}{K \cdot \mathtt{LCM}} \sum_{t=0}^{T-1} \begin{pmatrix} \|\vec{F}(x, \phi(x,t)) - \vec{F}(x, \phi(z,t))\|^2 + \\ \|\vec{F}(z, \phi(x,t)) - \vec{F}(z, \phi(z,t))\|^2 \end{pmatrix},$$

where $\mathtt{LCM}$ is the least common multiple of per-pixel periods [23]. This cost can be evaluated efficiently by separating cases *w.r.t.* l_x and l_z and using an integral image technique in a constant time similar to Liao *et al.* [23].

We also define the connectivity potential, $\gamma_s(x,z)$, in a semantic aware way, to maintain coherence within objects. We introduce a label occurrence $\vec{h}_t(x) = [h_t(k,x)]_{k=1}^{K}$, where the histogram $h_t(k,x)$ was defined in Sec. 3.1. If two histograms between neighbor pixels are similar, it indicates that two pixels have a similar semantic occurrence behavior. We measure the connectivity potential by computing the difference of semantic label occurrence distribution as

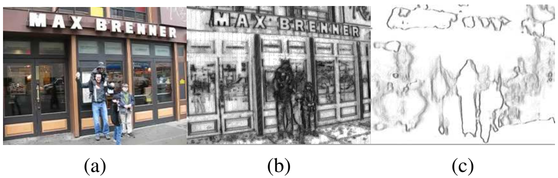


Figure 2: Comparison on connectivity potential $\gamma_s(x,z)$. (a) Selected frame. (b) $\gamma_s(x,z)$ in Liao *et al.* [23] (deviation of intensity difference across time). (b) Our version of $\gamma_s(x,z)$ (difference of semantic label occurrence distribution).

[5] When feeding semantic response F into the subsequent optimization, we re-normalize each vector across the channel axis to sum to one.

Algorithm 1 Procedure for Candidate Cinemagraph Generation.

```
1: Input : Video, semantic responses, spatial candidate map $\pi$.
2: Stage 1 (Initialization): Solve MRFs for $\mathbf{s}$, given each $p{>}1$ fixed (i.e., $\mathbf{s}^*_{|p}$).
3: (Multiple Candidate Cinemagraph Generation)
4: for each candidate label $\mathbb{ID}$ do
5:     Stage 2: Solve MRF for $\{\mathbf{p}{>}1, \mathbf{s}'\}$ given $\mathbb{ID}$, where each $p_x$ is paired as $(p_x, s^*_{x|p_x})$ from the step 2, $s'_x$ denotes all possible frames for the static case, $p{=}1$.
6:     Stage 3: Solve MRF for $\mathbf{s}$ given $\mathbb{ID}$ and fixed $\{\mathbf{p}^*\}$.
7:     Render the candidate cinemagraph result as described in Liao et al. [23, 22].
8: end for
9: Output : Candidate cinemagraphs.
```

$$\gamma_s(x, z) = 1 \Big/ \Big(1 + \lambda_s \|\hat{h}_t(x) - \hat{h}_t(z)\|_2\Big), \tag{5}$$

where $\hat{h}_t(\cdot)$ is the normalized version of $\vec{h}_t(x)$. As shown in Fig. 2, it preserves real motion boundaries better than the one proposed by Liao *et al.*

Label term We define the label term, E_{label}, to assign an object-dependent spatial penalty in addition to discouraging a trivial all-static solution as in Liao *et al.* This is key in generating object-specific candidate cinemagraphs that allows us to vary which objects are static vs. looping.

Our label term E_{label} is defined as:

$$E_{\text{label}}(l_x) = \begin{cases} E_{\text{static}}(x) \cdot \delta[\pi_{\mathbb{ID}}(x)], & p_x{=}1, \\ \alpha_\infty \cdot \delta[\vee_{i \in \mathcal{H}_{\text{nat.}}} \pi_i(x)], & 1{<}p_x{\le}P_{\text{short}}, \\ 0, & P_{\text{short}}{<}p_x, \end{cases} \tag{6}$$

where $\mathbb{ID}$ represents the current target candidate category index the algorithm will generate, and P_{short} defines the range of short periods. The label term E_{label} has three cases. When $p_x = 1$, *i.e.*, static, the cost imposes the static penalty E_{static} only when the semantic index at the pixel is the target label we want to make it dynamic. The static term $E_{\text{static}}(x)$ is defined as

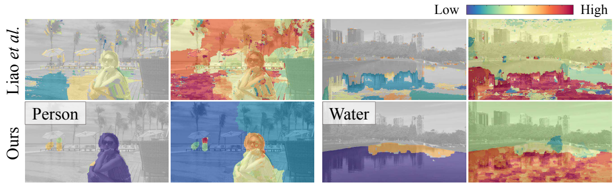
$$E_{\text{static}}(x){=}\alpha_{\text{sta.}} \min(1, \lambda_{\text{sta.}}\, \text{MAD}_{t'} \|N(x,t'){-}N(x,t'{+}1)\|), \tag{7}$$

where N is a Gaussian-weighted spatio-temporal neighborhood. The static term E_{static} penalizes significant temporal variance of the pixels neighborhood in the input video, and also prevents a trivial solution which assigns all the pixel to be static that attains perfect spatio-temporal consistency.

We also observe that long periods look more natural for natural objects. To encourage long period, we add high penalty on natural object regions for short period labels ($1{<}p_x{\le}P_{\text{short}}$) with a large α_∞. Otherwise, E_{label} is 0.

3.3. Optimization Procedure

The multi-label 2D MRF optimization in Eq. (1) can be solved by α-expansion graph cut [19]. Due to the size of the label space, *i.e.*, $|\mathbf{s}|{\times}|\mathbf{p}|$, directly optimizing Eq. (1) may stuck in poor local minima. This is because a graph cut α-expansion only deals with a single new candidate label at a time. Also, a video clip typically consists of multiple semantic regions, whereby several candidate cinemagraphs are generated. We present an efficient procedure for multiple candidates in Alg. 1, which is regarded as a block coordinate descent. The stages (1) and (2) in Alg. 1 are similar to the procedure of Liao *et al.* [23] except the $\mathbb{ID}$ dependency involved. Moreover, due to the restriction of the paired label, $s_{x|p}$, in the stage (1), the solution can be still restricted up to the stage (2); hence we additionally introduce the stage (3).



(a) Period $\mathbf{p}$ (b) Start frame $\mathbf{s}$ (c) Period $\mathbf{p}$ (d) Start frame $\mathbf{s}$

Figure 3: Visualization of $\{\mathbf{p},\mathbf{s}\}$ estimated by Liao *et al.* [23] (top) and ours (bottom). Values of $\mathbf{p}$ and $\mathbf{s}$ are presented by a color map on the top right corner, with gray indicating static pixels.

Since the terms related to candidate-specific regularization by $\mathbb{ID}$ are not involved in the stage (1), the initial paired label sets $\{(p_x, s_{x|p_x})\}$ obtained from the stage (1) are shared across all other stages. The complexities of each stage are proportion to the number of labels: $|\mathbf{s}|$, $|\mathbf{p}| + |\mathbf{s}|$ and $|\mathbf{s}|$ in the stages (1,2) and (3) respectively, which are significantly lower than directly optimizing the problem with $|\mathbf{p}|{\times}|\mathbf{s}|$ labels. The number of total candidate cinemagraphs generated is restricted to K. To obtain more diverse candidates, we allow the target $\mathbb{ID}$ to involve combination of multiple objects, *e.g.*, $\{$Person, Tree$\}$ in $\mathbb{ID}$, so that both are dynamic in a candidate cinemagraph.

Fig. 3 visualizes the labels $\{\mathbf{p},\mathbf{s}\}$ obtained by our semantic-based cinemagraphs, which show strong spatial coherence along the semantic regions.

4. Learning to Predict Human Preference

Given a set of candidate cinemagraphs generated from a video clip, we want to automate suggesting a single *best* cinemagraph or predicting a ranking for a specific user. To this end, we investigate a computational model to predict human perceptual preference for cinemagraphs. This model is trained on rating scores we collected from a user study.

4.1. User Study

Our study consisted of a dataset of 459 cinemagraphs,[6] of which mean video length is about 1 sec. The 459 cinemagraphs are the multiple candidates generated from 244 input video clips. The study consisted of 59 subjects; each was shown one cinemagraph at a time in random order, which is loop play-backed until a user provides a rating from 1 to

[6] As we are only interested in understanding semantic and subjective preference, we chose cinemagraphs that did not have any significant visual artifacts, so as not to bias the ratings.

5 using the following guideline: 1) rate each cinemagraph based on interestingness/appeal of the cinemagraph itself, 2) if it is not appealing at all (*i.e.*, you would delete it in an instant), rate it a 1, 3) if it is extremely appealing, (*i.e.*, you would share it in an instant), rate it a 5, 4) otherwise, give intermediate scores according to your preference. Before starting the study, each user was instructed, and carried out a short pilot test. In a pilot study, we found that asking users to rate all cinemagraphs was too fatiguing, which affected the rating quality over time. Instead, in our final user study, we limit the total time spent to 20 mins. On average, each subject ended up rating 289 cinemagraphs.

We conducted a simple statistical analysis to see the characteristics, which suggests that user rating behaviors are very diverse in terms of rate distribution shapes and little consensus among users for each cinemagraph. For instance, 72.66% of cinemagraphs in the dataset have the rates of the standard deviation $\sigma{>}1$ among users, while the ones having $\sigma{<}0.5$ is actually close to 0%,[7] implying strong personal preference for cinemagraphs. Thus, user-dependent preference may not be modeled using a single model across all users (refer to *global model*). We instead learn a local preference model for each user. In addition, we have to handle partial information, since every subject rated only about 63% of all the cinemagraphs.

4.2. Preference Prediction Model

Given the user-study data, our goal is to predict subjective preference rating for a user. A basic approach we can consider is to model subjective preference by associating a regression model to each user independently (refer to *individual model*). However, it is not practical due to two issues on this model: (1) for a new user, we need to train a new model from the scratch, and (2) it requires a lot of data for each user to achieve reasonable generalization. To handle these issues, we use a collaborative style learning to process multi-user information.

To develop a model depending on user and context (cinemagraph), we formulate the problem as a regression, $y{=}f(\mathbf{v},\mathbf{u})$, where y, $\mathbf{v}$ and $\mathbf{u}$ denote a predicted rating, context and user features respectively. In what follows, we describe the context and user features, and the model f.

Context Feature The context feature $\mathbf{v}$ can be easily extracted from cinemagraphs, which may be relevant to its preference. We use and concatenate three types of features: hand designed, motion, and semantic features. The hand designed feature consists of quantities related to face, sharpness, trajectory, objectness and loopability. For the motion, we use C3D [36], which is a deep motion feature. For the semantic feature, we use two semantic label occurrence measures for static and dynamic regions. These detail specifications and lists refer to the supplementary.

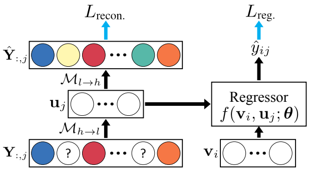


Figure 4: Diagram for architecture and variable dependency of the proposed joint model. The left and right towers denote an auto-encoder and a regression model for rate prediction, respectively.

User Feature Contrary to the context feature, it has not been researched which and what user profiles are related to user's preference for cinemagraph, *i.e.*, undefined. In this regard, we do not use any explicit profile, *e.g.*, age, gender, but instead we leverage rating behavior to reveal user's latent feature. Motivated by collaborative learning [32, 18], we assume that a user's characteristics can be modeled by similar preference characteristics of other users and so it is for similar cinemagraphs. We observed that this is also valid for our scenario (through evidences in Sec. 5 and the supplementary). This allows us to model group behavior and to obtain compact user representation from user rate data without any external information.

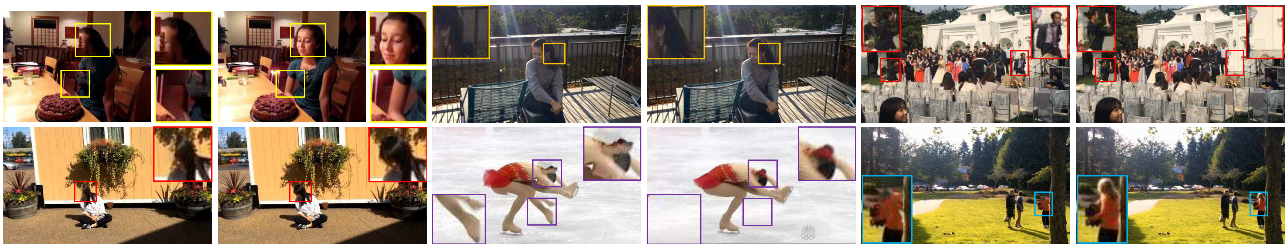
We are motivated by an unsupervised approach using auto-encoder [16] to learn the latent user feature such that users with similar preferences have similar features. It has known to have an implicit clustering-effect by enforcing embedding of data to be low-dimensional, called bottleneck [14]. Formally, we represent the multi-user rating information as a matrix $\mathbf{Y}{\in}\mathbb{R}^{m\times n}$ with m cinemagraphs and n users, of which entry y_{ij} is a rate $\{1,\cdots,5\}$ of i-th cinemagraph by j-th user. Given a rating vector for a user, $\mathbf{y}_j = \mathbf{Y}_{:,j}$,[8] we consider two mappings $\{\mathcal{M}\}$ for the auto-encoder, one of which maps a high-dimension vector to low-dimensional space,[9] as $\mathbf{u}{=}\mathcal{M}_{h\to l}(\mathbf{y})$ and the other is the inverse map as $\mathbf{y}{=}\mathcal{M}_{l\to h}(\mathbf{u})$. Thus, the auto-encoder can be trained by minimizing self-reconstruction loss, $\|\mathbf{y}{-}\mathcal{M}_{l\to h}(\mathcal{M}_{h\to l}(\mathbf{y}))\|$. Through this procedure, we can obtain the latent user feature $\mathbf{u}$ from the intermediate embedding. Unfortunately, this is not directly applicable to our problem due to incomplete data (partial ratings by a user). Thus, we leverage a model suggested by Carreira *et al.* [6], *i.e.*, an auto-encoder with missing values (AEm), depicted as the left tower in Fig. 4, whereby rating vectors with missing values are completed and simultaneously nonlinear low-dimensional embeddings of rating vectors are learned. Now, we have the latent user feature $\mathbf{u}$. The mappings for $\{\mathcal{M}\}$ make use of a Gaussian radial basis function (RBF) network [4] as suggested by Carreira *et al.*

Model 1) A Simple User Aware Model Since we have the described features $\mathbf{u}$ and $\mathbf{v}$, now we can train a regression model such that $y{=}f(\mathbf{v}{,}\mathbf{u})$. For the simple baseline model, we use the random forests (RF) regression [11] as a regres-

[7] Statistics of user ratings can be found in the supplementary due to space limitation.

[8] We borrow a MATLAB like matrix-vector representation.

[9] $\mathcal{M}$ applies vector-wise to each column.



(a) Liao *et al.* (b) Proposed (c) Liao *et al.* (d) Proposed (e) Liao *et al.* (f) Proposed

Figure 5: Comparisons with Liao *et al.* [23]. The shown sampled frames are from the cinemagraphs generated by each method. We can observe severe artifacts such as distorted or tearing faces or bodies in (a,c,e), while ours shows artifact-free and semantic preserving results.

sion function $f(\cdot)$. The RF model is proper for this purpose in that we have limited amount of training data. We use 10 number of ensembles for generalized performance. We call this model as subjective aware RF (S-RF).

Model 2) A Joint and End-to-End Model When we learn $\mathbf{u}$ by Carreira *et al.* [6], the context feature information is not used; hence, any link between the user and context information may not be reflected to $\mathbf{u}$. To learn $\mathbf{u}$ reflecting context information, we formulate a *joint model* for both regression and auto-encoder that are entangled by user latent feature as a medium variable, of which loss is defined as

$$\underset{\mathbf{U},\mathbf{Y}_{\overline{\Omega}},\{\mathcal{M}\},\theta}{\arg\min} \; L_{\text{reg.}}(\mathbf{U},\boldsymbol{\theta}) + \lambda L_{\text{recon.}}(\mathbf{U},\mathbf{Y}_{\overline{\Omega}}, \underset{h\to l}{\mathcal{M}}, \underset{l\to h}{\mathcal{M}}), \quad (8)$$

where Ω denotes the index set for known entries and $\overline{\Omega}$ is its complementary set, *i.e.*, missing entries, $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$, θ denotes regression model parameters, and $\lambda = \frac{1}{nm}$ is the balance parameter. $L_{\text{reg.}}$ and $L_{\text{recon.}}$ are respective common l_2 regression loss and the the auto-encoder loss of AEm by Carreira *et al*. As with Carreira *et al.*, $L_{\text{recon.}}(\cdot)$ incorporates missing values,[10] and defined as:

$$L_{\text{recon.}}(\mathbf{U},\mathbf{Y}_{\overline{\Omega}}, \underset{h\to l}{\mathcal{M}}, \underset{l\to h}{\mathcal{M}}) = \quad (9)$$
$$\|\mathbf{U} - \underset{h\to l}{\mathcal{M}}(\mathbf{Y})\|_F^2 + \|\mathbf{Y} - \underset{l\to h}{\mathcal{M}}(\mathbf{U})\|_F^2 + R_{\mathcal{M}}(\underset{h\to l}{\mathcal{M}}, \underset{l\to h}{\mathcal{M}}),$$

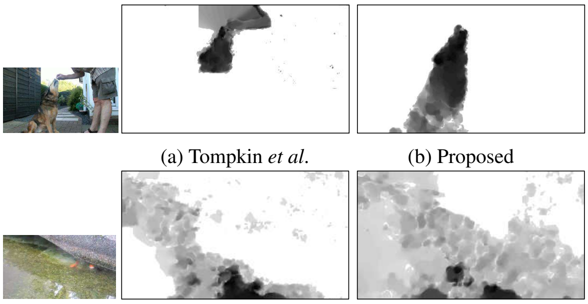
where $\|\cdot\|_F$ denotes Frobenius norm and $R_{\mathcal{M}}(\cdot,\cdot)$ is the l_2 regularization term for two mappings. The same user feature $\mathbf{U}$ is also fed into $L_{\text{reg.}}(\cdot)$:

$$L_{\text{reg.}}(\mathbf{U},\boldsymbol{\theta}) = \textstyle\sum_{(i,j)\in\Omega} (y_{ij} - f(\mathbf{v}_i,\mathbf{u}_j,\boldsymbol{\theta}))^2 + R_f(\boldsymbol{\theta}), \quad (10)$$

where $R_f(\cdot)$ is the l_2 regularization term for the rating regressor f, and we use a linear regression for $f(\cdot)$ as $f(\mathbf{u},\mathbf{v},\boldsymbol{\theta}) = \boldsymbol{\theta}^\top[\mathbf{u};\mathbf{v};1]$. The variable dependency and overall architecture are shown in Fig. 4. We optimize Eq. (8) by the Gauss-Newton method in an alternating strategy. Its optimization details can be found in the supplementary.

Having two loss functions on the same rating may seem redundant, but the information flow during optimization is significant. The sum of two gradients, back-propagated through the rating regressor $f(\cdot)$ to $\mathbf{U}$ (see Fig. 4) and from $L_{\text{recon.}}$, encourages $\mathbf{U}$ to be learned from auto-encoding with missing completion and context aware regression. This can be regarded as multi-task learning, which has regularization effect [34] that mitigates the problems of partial and limited number of measurements. This is because it collaboratively uses all the ratings provided by all the users, whereas the *individual model* does not.

For new user scenario, it can be dealt with in a way similar to [18, 37, 31] by finding a similar other user in database.



(a) Tompkin *et al.* (b) Proposed

(c) Yeh and Li (d) Proposed

Figure 6: Comparison with Tompkin *et al.* [35] and Yeh and Li [39]. The intensity maps indicate average magnitude of optical flow (darker represents larger magnitude). The dynamic areas in our results are better aligned along semantic boundaries of moving objects ("animal" in (a,b), "water" in (c,d)), than other methods.

5. Results

Implementation and Run-time Speed We implemented our approach on a PC with 3.4GHz CPU, 32GB RAM and

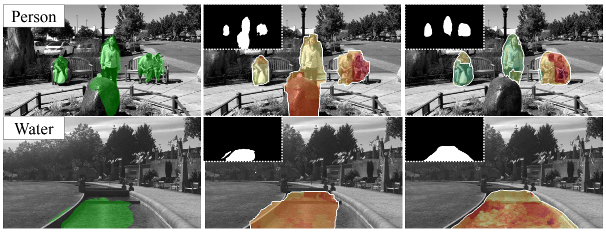


Figure 7: Comparison without/with user editing for our method. [Left] Sampled frames overlaid with semantic segmentation mask for a selected object by green color, [Middle] Color coded $\{\mathbf{s}\}$ label obtained by our method without user editing. [Right] Results with user editing. Each superposed black-white mask shows a semantic binary map $\pi_{(\cdot)}$, on which user edits. Color coding of $\{\mathbf{s}\}$ is referred to Fig. 3.

[10] Note that we assume there is no case where all entries in a column vector $\mathbf{y}$ are missing.

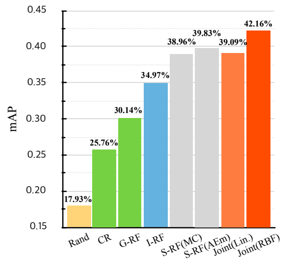
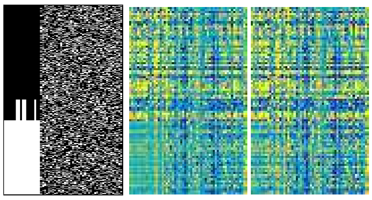


Figure 8: mAP comparison for rating prediction. Rand: random guess, CR: constant prediction with rate 3, {G,I,S}-RF: {global, individual, subjective} RFs, Joint: Joint model with either linear or RBF mappings. MC and AEm indicates user feature obtained from either matrix completion [5] or AEm [6].

NVIDIA GTX1080 GPU, and applied it over a hundred of casually shot videos acquired by ourselves or collected from previous works. For speed purposes, we downsampled and trimed the videos so that the maximum resolution does not exceed 960×540 and the duration is less than 5-sec long. Without careful engineering level code optimization, our system typcially takes a few minutes to generate all the semantically meaningful cinemagraphs for each video.

The Importance of Semantic Information As we argued before, semantic information plays a key role in the process of candiate generation to suppress any semantic meaningless video loops. Thanks to our novel semantic aware cost function (described in Sec. 3) embedded in the MRF framework, the generated cinemagraphs all trend to be more meaningful compared with the ones generated by previous work such as [23] in which only low-level information is considered. Fig. 5 shows a few typical videos that semantic information is crucial to avoid severe artifacts. As indicated by the comparison, the results for Liao *et al*. tend to have artifacts such as distortions or ghosting effects, as highlighted in the close-up views, while our method preserves the boundary region of objects well with more natural looping. Figs. 3 and 6 show another examples of what happens if semantic-based looping is not applied.

The Effectiveness of Collaborative Learning We compare the several baselines for cinemagraph preference prediction in Fig. 8 in terms of mean average precision (mAP). Interestingly, S-RF and Joint outperform I-RF (individual learning per a user), which suggests collaboratively learning the preference behavior is beneficial. The best performance of Joint shows learning the user feature in a context aware manner can improve the quality of preference prediction for cinemagraph. Another example in Fig. 9 shows the completed rating matrix for missing entries by a matrix completion (MC) [5] (as a reference that does not use context feature) and ours. The completed regions in each left bottom region of matrices clearly show that our method predicts preference ratings more plausibly and diversely than MC by virtue of context aware feature. We visualize 2D embedding of latent user features by t-SNE [26] in Fig. 10, which suggests that users can be modeled by a few types for cinemagraph preference. Refer to supplmentary material for additional results.



(a) Missing pattern of rates (b) MC [5] (c) Joint (RBF)

Figure 9: Completed rating matrices (rows: cinemagraphs, cols.: users). White color indicates missing entries, and rate scores are color-coded through the *parula* color map built in MATLAB.

User Interaction We have showed our results in cases where semantic segmentation worked well. While significant progress has been made on semantic segmentation, the semantic segmentation that we use does not always produce object regions with perfect boundaries or labeling as shown in Fig. 7-[Left], which produces loop labels violating the semantics of the scene (Fig. 7-[Middle]). Using a more advanced semantic segmentation approach such as [13, 12, 9, 8] is one way to improve. However, with simple manual interaction to roughly correct the mask $\pi_{\mathbb{ID}}$, we can quickly fix the issues and output semantically meaningful cinemagraphs (Fig. 7-[Right], where each example took about 19 sec. on average for the editing). This simple optional procedure is seamlessly and efficiently compatible to our MRF optimization (details in the supplementary).

6. Discussion and Future Work

We create cinemagraphs using a semantic aware per-pixel optimization and human preference prediction. These allow our method to create cinemagraphs without user input; however, the automatic results are limited by the quality of the semantic segmentation. Semantic segmentation itself remains a open research issue beyond the scope of this work, and as these methods improve, they can be used in our approach to improve the results. As an alternative, we optionally allow the user to correct imperfections of semantic segmentation and thus improve the quality of the output. Our system is flexible in that the semantic segmentation part can be seamlessly replaced with an advanced or heterogeneous (*e.g*., face segmentation) one to improve semantic knowledge or speed, *e.g*., [28].

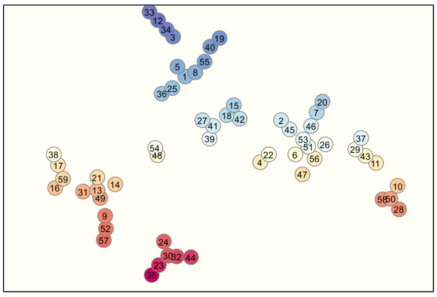


Figure 10: t-SNE visualization for 59 latent user features $\{\mathbf{u}\}$ obtained by Joint(RBF). This plot clearly shows clustered positions of users, which may imply that the intrinsic dimensionality of user space holds the low-dimensionality assumption.

References

[1] Cinemagraph. http://www.cinemagraph.com/, 2012. 1, 2

[2] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Selectively de-animating video. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):66, 2012. 1, 2

[3] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Automatic cinemagraph portraits. *Computer Graphics Forum*, 32(4):17–25, 2013. 2

[4] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006. 6

[5] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 2009. 8

[6] M. A. Carreira-Perpin, Z. Lu, et al. Manifold learning and missing data recovery through unsupervised regression. In *IEEE International Conference on Data Mining (ICDM)*, pages 1014–1019, 2011. 6, 7, 8

[7] Y.-T. Chan, H.-C. Hsu, P.-Y. Li, and M.-C. Yeh. Automatic cinemagraphs for ranking beautiful scenes. In *ACM International Conference on Multimedia*, pages 1361–1362, 2012. 2

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations (ICLR)*, 2015. 8

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 8

[10] J. Choi, T.-H. Oh, and I. So Kweon. Video-story composition via plot analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3122–3130, 2016. 2

[11] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013. 6

[12] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8

[13] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[14] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. 6

[15] M. Gygli, Y. Song, and L. Cao. Video2gif: Automatic generation of animated gifs from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[16] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 6

[17] N. Joshi, S. Mehta, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen. Cliplets: Juxtaposing still and dynamic imagery. In *Symposium on User Interface Software and Technology (UIST)*. ACM, 2012. 1, 2

[18] A. Kapoor, J. C. Caicedo, D. Lischinski, and S. B. Kang. Collaborative personalization of image enhancement. *International Journal of Computer Vision (IJCV)*, 108(1-2):148–164, 2014. 6, 7

[19] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):147–159, 2004. 5

[20] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 22(3):277–286, 2003. 2, 4

[21] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A new dataset and benchmark on animated gif description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650, June 2016. 2

[22] J. Liao, M. Finch, and H. Hoppe. Fast computation of seamless video loops. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 34(6):197, 2015. 2, 5

[23] Z. Liao, N. Joshi, and H. Hoppe. Automated video looping with progressive dynamism. *ACM Transactions on Graphics (SIGGRAPH)*, 32(4):77, 2013. 1, 2, 3, 4, 5, 7, 8

[24] Z. Lin, L. Wang, Y. Wang, S. B. Kang, and T. Fang. High resolution animated scenes from stills. *IEEE Transactions on Visualization & Computer Graphics (TVCG)*, 3:562–568, 2007. 2

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 3

[26] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008. 8

[27] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 891–898, 2014. 3

[28] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 8

[29] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. *ACM SIGGRAPH*, pages 489–498, 2000. 1, 2

[30] L. Sevilla-Lara, J. Wulff, K. Sunkavalli, and E. Shechtman. Smooth loops from unconstrained video. *Computer Graphics Forum*, 34(4):99–107, 2015. 2

[31] D. Song, C. E. Lee, Y. Li, and D. Shah. Blind regression: Nonparametric regression for latent variable models via collaborative filtering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2155–2163, 2016. 7

[32] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009. 6

[33] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. 1

[34] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems (NIPS)*, pages 640–646, 1996. 7

[35] J. Tompkin, F. Pece, K. Subr, and J. Kautz. Towards moment imagery: Automatic cinemagraphs. In *Conference for Visual Media Production*, pages 87–93, 2011. 2, 7

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 6

[37] W. Wang, M. A. Carreira-Perpinán, and Z. Lu. A denoising view of matrix completion. In *Advances in Neural Information Processing Systems (NIPS)*, pages 334–342, 2011. 7

[38] H. Yan, Y. Liu, and Y. Furukawa. Turning an urban scene video into a cinemagraph. *arXiv preprint arXiv:1612.01235*, 2016. 2

[39] M.-C. Yeh. Selecting interesting image regions to automatically create cinemagraphs. *IEEE MultiMedia*, 23:72–81, 2016. 2, 7

[40] M.-C. Yeh and P.-Y. Li. An approach to automatic creation of cinemagraphs. In *ACM International Conference on Multimedia*, pages 1153–1156. ACM, 2012. 2