

Supplementary Material: Personalized Cinemagraphs using Semantic Understanding and Collaborative Learning

Tae-Hyun Oh^{1,2*} Kyungdon Joo^{2*} Neel Joshi³ Baoyuan Wang³ In So Kweon² Sing Bing Kang³
¹MIT CSAIL, Boston, MA ²KAIST, South Korea ³Microsoft Research, Redmond, WA

Summary

This is a part of the supplementary material. The contents of this supplementary material include user study information, implementation details including parameter setups, additional results for the cinemagraph generation and the human preference prediction, and supplementary tables, which have not been shown in the main paper due to the space limit. The supplementary material for resulting videos (comparison with other methods [10, 6, 5, 12], user editing effects, qualitative results) can be found in the project web page.

1. User Study Information

During the user study, each cinemagraph is replayed again and again until a user provides a rating for it. The user spends about 4 seconds per cinemagraph on average (we did not limit the time for individual samples but limit the total time by about 20 min.). Before starting the user study, each user was instructed by us, and carried out short pilot tests. The users used the interface provided by us as shown in Fig. 1. On user demographics, the age range is 23-35 years old. About 85% were engineering students and researchers, with the others being non-technical people.

The preference rating could be regarded as an open-ended question. Since the relationships between specific features and user’s cinemagraph preference have not been studied, we do not limit any specific preference criteria to avoid bias but capture natural behaviors.

Statistics of User Ratings Fig. 2-(a) shows rating distributions for a random sample of users. The graph shows a very diverse set of rating distributions; the skew and shapes are all quite different. Some of users have a fairly uniform distribution for their ratings, while others clearly favor a certain value (even though few users strongly biased, their ratings are still distributed and express preferences to some extent).

Fig. 2-(b) shows a measure of the diversity of user rating

*The first and second authors contributed equally to this work.

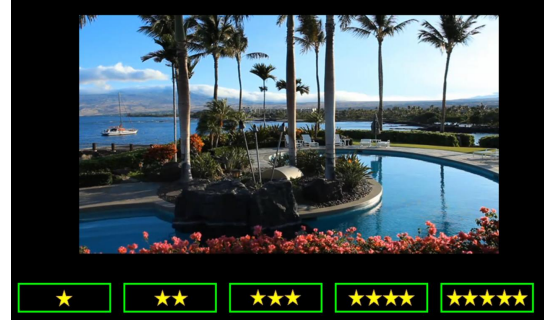


Figure 1: Interface for our user study. The subject is asked to rate a randomly shown cinemagraph (from 1 star to 5 stars).

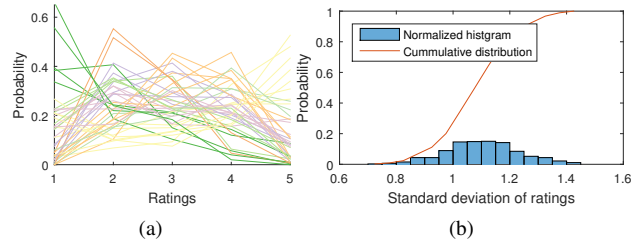


Figure 2: Statistics of user ratings. (a) Rating distributions for sampled users (color encoded by clustering users having similar distribution). Different users have diverse tendencies for providing ratings. (b) Distribution of standard deviations of ratings for each candidate cinemagraph across users. Normalized histogram of the standard deviation and its cumulative distribution are overlaid.

scores per cinemagraph. For each candidate cinemagraph, we measure the standard deviation σ of user ratings. The histogram in Fig. 2-(b) is constructed by binning the standard deviations for all cinemagraphs.

If the histogram has a pick at $\sigma = 0$, it means all the users gave the same rates for all the cinemagraph, *i.e.*, perfect consensus by common sense. The way of analyzing data may not be same with any traditional statistical test, the presented statistic plot actually implies subjectiveness of rating behavior for cinemagraph. Looking at the overlaid, cumulative distribution curve, it is interesting to see that 72.66% of cinemagraphs in the dataset have $\sigma > 1$, while

the percentage of cinemagraphs having $\sigma < 0.5$ is actually close to 0%. This represents the diversity of rating tendencies that are user-dependent for a cinemagraph.

2. Implementation Details

In this section, we provide the detail information that allows to reproduce our implementation.

Parameter Setup All the parameters used in our experiments are listed in the following table:

Related terms	Parameters
$\pi(x)$ in Sec. 3.1	$thr. = 0.15T$
E_{label} in Sec. 3.2	$\alpha_1 = 1$
$E_{spa.}$	$\alpha_2 = 15$
$E_{temp.}$ and $E_{spa.}$	$w = 0.2$
$\gamma_t(x)$ in $E_{temp.}$	$\lambda_t(x) = \begin{cases} 125, & \text{if } (\forall i \in \mathcal{H}_{int}, \pi_i(x)) = 1, \\ 125/2, & \text{otherwise.} \end{cases}$
$\gamma_s(x, z)$ in $E_{spa.}$	$\lambda_s = 10/\sqrt{K}$
E_{label}	$\alpha_{\infty} = 1000$
E_{label}	$P_{short} = 20$
E_{static}	$\lambda_{sta.} = 100$
E_{static}	$\alpha_{sta.} = 0.03$
$N(\cdot)$ (Gaussian kernel) in E_{static}	$\sigma_x = 0.9$ and $\sigma_t = 1.2$

Candidate Cinemagraph Generation The procedure for MRF optimization is as follows:

1. For each looping period label $p > 1$, we solve Eq. (1) only for the per-pixel start times $s_{x|p}$ given the fixed p , saying $L|p$, by solving a multi-label graph cut with the start frame initialization $s_{x|p}$ that minimizes $E_{temp.}$ per pixel independently.
2. Given a candidate object label $\mathbb{I}\mathbb{D}$, we solve for per-pixel periods $p_x \geq 1$ that define the best video-loop $(p_x, s_{x|p_x})$ where $s_{x|p_x}$ is obtained from the stage (1), again by solving a multi-label graph cut. In this stage, the set of labels are $\{p > 1, s'_x\}$, where s'_x denotes all possible frames for the static case, $p=1$.
3. Due to the restriction of the paired label, $s_{x|p}$, in the stage (1), the solution can be restricted. In this stage, we fix p_x from the stage (2) and solve a multi-label graph cut only for s_x .

Conceptually, we should alternate the stages (2) and (3). However, in practice, we need to perform the optimization only once, and even then it produces a better solution than the two-stage approach suggested by Liao *et al.* The other difference over Liao *et al.* is that since we generate several candidate cinemagraphs (each representing a different semantic object), we must solve the multi-label graph cut several times.

In MRF optimization, we parallelize the graph cut optimizations using OpenMP and only use a few iterations through all candidate α -expansion labels. We find that two iterations are sufficient for the stage (2) and a single iteration is sufficient for all the other stages. To reduce computational cost, we quantize the loop start time and period labels

to be multiples of 4 frames. We also set a minimum period length of 32 frames.

User Editing To edit the cinemagraph, the user selects a candidate class $\mathbb{I}\mathbb{D}$ and a representative frame having regions in which bad boundaries occur.¹ Then, the boundary shape of binary map $\pi_{\mathbb{I}\mathbb{D}}$ is edited on overlaid selected frame.

Once the editing is done, the edited $\pi_{\mathbb{I}\mathbb{D}}$ is fed into MRF optimization and re-run the stages (2, 3) in the Algorithm 1 with the parameter $\alpha_{sta.}$ in $E_{static}(\cdot)$ being doubled, so that the edited regions are strongly encouraged to be dynamic. Note that despite increasing $\alpha_{sta.}$ a non-loopable region will remain static. Rerunning the stages (2, 3) requires initialization and pre-computed $\{s_{x|p}\}$, but we can re-use these pre-computed quantities from the stage (1).

Context Feature For the context feature, we use three types of features: hand designed, motion, and semantic features. We extract 55-dimensional hand designed features, which consist of face, sharpness, trajectory, objectness and loopability (its details are listed in Sec. 4 of this supplementary material). We use C3D [11] as the motion feature, which is a deep motion feature obtained from 3D convolutional neural network. We apply C3D with the stride of 16 frames and 8 frame overlap, and average pooling is applied, so that we have a 4096 dimensional representative motion feature for each cinemagraph. For the semantic feature, we use two semantic label occurrence measures for static and dynamic regions as $\vec{h}_{static} = \sum_{x \in static} \vec{h}(x)$ and $\vec{h}_{dyn.} = \sum_{x \in dynamic} \vec{h}(x)$ respectively. The final context feature for a cinemagraph is formed by concatenating all the mentioned feature vectors, where each feature is independently normalized by the infinity norm, *i.e.*, the largest absolute value, before concatenation.

Model 2) A Joint and End-to-End Model We apply an alternating optimization strategy iteratively over $(\mathbf{U}, \mathbf{Y}_{\bar{\Omega}})$ and $(\{\mathcal{M}\}, \theta)$; we first fix $(\{\mathcal{M}\}, \theta)$ during optimizing $(\mathbf{U}, \mathbf{Y}_{\bar{\Omega}})$ and followed by $(\{\mathcal{M}\}, \theta)$ while fixing $(\mathbf{U}, \mathbf{Y}_{\bar{\Omega}})$ until convergence. When fixing $(\{\mathcal{M}\}, \theta)$, optimizing $(\mathbf{U}, \mathbf{Y}_{\bar{\Omega}})$ is the non-linear least square problem. We optimize it using the Gauss-Newton method, where $\frac{\partial f(\mathbf{u}, \mathbf{v}; \theta)}{\partial \mathbf{u}}$ is added when updating \mathbf{U} . In the process of minimizing $L_{recon.}$, missing values $\mathbf{Y}_{\bar{\Omega}}$ are regarded as optimization variables while \mathbf{Y}_{Ω} is kept constant.

When we solve for $(\{\mathcal{M}\}, \theta)$, we separately solve three regressions for $\{\mathcal{M}\}$ and $f(\cdot; \theta)$. The mappings for $\{\mathcal{M}\}$ use Gaussian radial basis function (RBF) network [2] to provide a non-linear mapping, $\mathcal{M}(\mathbf{x}) = \mathbf{W}\mathcal{K}(\mathbf{x})$ where $\mathcal{K}(\mathbf{x}) = [\kappa_1(\mathbf{x}), \dots, \kappa_d(\mathbf{x})]$ ($d \ll \min(m, n)$), where $\mathbf{Y} \in \mathbb{R}^{m \times n}$, and $\kappa_i(\mathbf{x}) = \exp(\frac{1}{2\sigma_i^2} \|\mathbf{x} - \mu_i\|_F^2)$.² For the

¹ Since it is used as a guide, it does not have to be exact.

² When we use a linear mapping for \mathcal{M} , it reduces to a linear model that forms matrix factorization.

regressions for $\{\mathcal{M}\}$ between \mathbf{U} and \mathbf{Y} , we update respective $\{\mu\}$ by k -means and $\{\sigma\}$ by cross validation with a subset that is split from the training set used for RBF training. Then, $\{\mathbf{W}\}$ is solved for by a least square fit. With this RBF mapping, the regularization term is defined as

$$R_{\mathcal{M}}(\mathcal{M}_{h \rightarrow l}, \mathcal{M}_{l \rightarrow h}) = \lambda_R (\|\mathbf{W}_{h \rightarrow l}\|_F^2 + \|\mathbf{W}_{l \rightarrow h}\|_F^2).$$

The rating regressor $f(\cdot)$ uses a linear function as $y = f(\mathbf{u}, \mathbf{v}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top [\mathbf{u}; \mathbf{v}]$. Again, the parameter $\boldsymbol{\theta}$ is updated by least square fit with its regularization term $R_f(\boldsymbol{\theta}) = \lambda_\theta \|\boldsymbol{\theta}\|_F^2$. The regularization parameters are set as $\lambda_R = \lambda_\theta = 0.1$. The number of RBF basis functions is set as $d = 25$. These parameters are chosen by running the algorithm on the separated validation set (more details are described in Sec. 3.2 of this supplementary material), which was not used for test in all experiments. We use a validation dataset for parameter tuning with the parameter sets $\lambda_R = \lambda_\theta = \{1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}, 0.1, 1\}$ and $d = \{5, 10, 15, 20, 25, 30, 35, 40, 45\}$.

In our method, we initialize $\mathbf{Y}_{\overline{\mathcal{Q}}}$ from the convex matrix completion (MC) [3] with speeding up by [9], \mathbf{U} from Laplacian eigenmap [1] on \mathbf{Y} obtained from MC with 25 dim as mentioned above. Then, with this initialization, the mappings $\{\mathcal{M}\}$ and the rating regressor $f(\cdot)$ are fit.

3. Additional Results

In this section, we present additional qualitative results for semantic cinemagraph generation, followed by extensive evaluation on the computational model for human preference prediction.

3.1. Evaluation on Semantic Cinemagraph Generation

Computational Time Profile In our experiments, the input videos are at most 5 seconds long, with maximum rate of 30 frames/second. The resolution is at most 960×540 pixels; higher resolutions are down-sampled. The processing time for a 3-sec 960×540 video takes a few minutes, depending on the number of candidates. Here is the breakdown in timing: initialization ≈ 10 secs (the stage (1) in Algorithm 1, MRF solving ≈ 50 secs per candidate (the stages (2, 3) in Algorithm 1), and rendering ≈ 10 secs.

Additional Qualitative Comparison Figure 3 shows a comparison with Tomkin *et al.* [10]. The method of Tomkin *et al.* allows user to select the region and loop to be animated, but has no synchronization feature. The example of Tomkin *et al.* have not only the desynchronized animation on eye blink and visual artifacts on that region, which shows what happens if semantic-based looping is not applied. The differences are clearer in our supplementary video, which we encourage the reader to view.

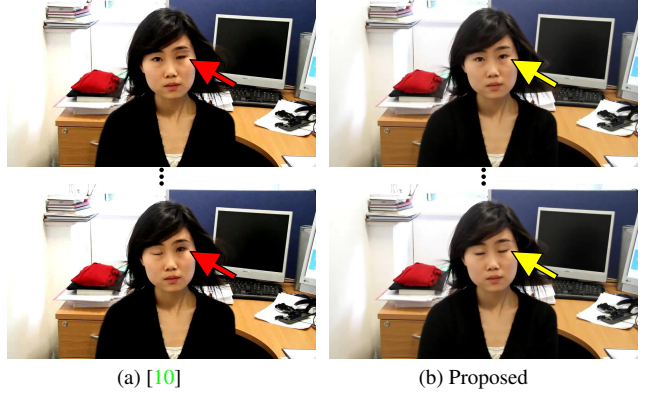


Figure 3: Comparisons of our cinemagraph generation with Tomkin *et al.* [10]. In the result (a) of Tomkin *et al.*, although a winking effect on the eyes is intentionally introduced by user editing, it generates unsynchronized one (red arrow) with visual artifact, while our result in (b) shows synchronized eye blinking of person (yellow arrow).

Cinemagraph Visualization Figure 4 shows representative examples of cinemagraphs rendered using different periods and start frames ($\{\mathbf{p}, \mathbf{s}\}$ respectively). Each row is of the same scene, and each column represents a candidate cinemagraph (*i.e.*, a different object to animate). The heat map indicates how dynamic the region is, with gray being static. The preference prediction results in Fig. 4 will be explained in the subsequent section.

3.2. Evaluation on Human Preference Prediction

In this section, we evaluate the preference prediction model described in Sec. 4 of the main paper in the following ways: performance and visualization of grouping effect. Throughout our experiments, we randomly sampled 10% rating data as the validation set, and tune parameters of methods using this set. We use the rest of the data for 9-fold cross validation, so that the amount of test set is same with the validation set.

Performance In Fig. 8 of the main paper, we consider other regression methods to understand the effects of several factors, and especially choose randomized forests (RF) [4] as the main competitor.³ Fig. 8 of the main paper shows the performance comparison: Rand: random guess (a lower bound of the performance), CR: constant prediction model with rate 3, G-RF: a single global RF model for all users, I-RF: RFs individually learned for each user, S-RF+ $\{\text{MC}, \text{Ours}\}$: a single RF model for all users with subjective user feature obtained from either MC or Ours (for both user features, we use 25 dimensions), Ours: the proposed method with either linear or RBF mapping func-

³We tested other regression methods, such as linear, support vector, Gaussian process, multi-layer perceptron, for the rate prediction given context and user features. In our scenario with limited amount of training data, RF performed best; hence we only report RF based results for simplicity.
























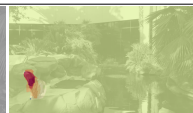













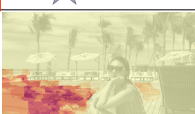
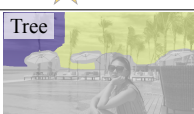















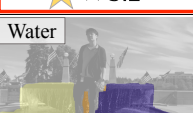





















































						
	Water		Tree		Person	
USER A	 × 3.1	 × 4.0	 × 1.0	 × 1.0	 × 3.0	 × 3.0
USER B	 × 4.0	 × 4.0	 × 3.3	 × 3.0	 × 2.8	 × ?
						
	Water		Tree		Animal	
USER A	 × 3.3	 × 3.0	 × 2.0	 × 2.0	 × 3.0	 × 3.0
USER B	 × 3.6	 × 4.0	 × 1.5	 × ?	 × 1.9	 × 2.0
						
	Water		Tree		Person	
USER A	 × 4.0	 × 4.0	 × 3.5	 × ?	 × 4.5	 × 5.0
USER B	 × 3.2	 × ?	 × 2.1	 × 3.0	 × 2.5	 × 3.0
						
	Water		Tree		Person	
USER A	 × 4.0	 × 4.0	 × 3.6	 × 3.0	 × 3.4	 × ?
USER B	 × 3.0	 × 3.0	 × 3.1	 × 1.0	 × 5.0	 × 5.0
						
	Grass		Person		Tree	
USER A	 × 1.8	 × 2.0	 × 2.4	 × 1.0	 × 2.5	 × 1.0
USER B	 × 3.0	 × 3.0	 × 3.5	 × 4.0	 × 4.0	 × 4.0
						
	Water		Tree		Person	
USER A	 × 2.9	 × 2.0	 × 3.0	 × ?	 × 0.6	 × 1.0
USER B	 × 2.7	 × 3.0	 × 1.4	 × 3.0	 × 2.0	 × ?

Figure 4: Visualization of $\{p, s\}$ and predicted ratings for unseen cinemagraphs by our prediction model. Each row presents three different candidate cinemagraphs generated from a single video input, and subsequent two columns are a pair of $\{p, s\}$, whose value is presented by a color map ranging from blue to through yellow to red as values increase, with gray indicating static pixels. Note that the presented cinemagraphs are unseen data during training. Preferences are not observed for every combination of users and cinemagraphs, which is indicated by the symbol ‘?’ as unknown ground truth. Red highlights indicate the selected best cinemagraph for each user according to the predicted preference rates, and blue highlights indicate the true preference according to the surveyed preference rate.

tions. G-RF and I-RF require context feature only, while S-RF’s require both context and user features. For RF based methods, we use 10 number of ensembles.

It is worthwhile to see the learnability of human prefer-

ence by comparing simple regression, *i.e.*, G-RF and I-RF. As mentioned in Sec. 7.1 of the main paper, we cannot find any common sense from the statistics of user ratings, rather it reveals the fact that users’ preferences are too subjective;

it can be deduced from low mAP of G-RF. Note that modeling of G-RF can be regarded as an attempt to learn a common sense of human preference. In order to show the importance of the user feature, we compare S-RF, which uses both user and context features, with G-RF and I-RF. The improvement of S-RF over G-RF and I-RF clearly shows the importance of the user feature. On the other hand, the importance of context feature is shown by comparing S-RF and MC which do not use context feature. Notice that S-RF can be used only when user feature is given by other methods that can learn user feature in an unsupervised manner such as MC or Ours. Thus, S-RF is an ideal comparison in the setup without given user feature. Nonetheless, Ours (RBF) achieves the best performance over S-RF by virtue of joint approach to learn user representation and regression. Lastly, comparing to Ours (Lin.) shows that the non-linear dimension reduction is crucial for implicit user relational modeling in a collaborative learning regime. Running time of Ours (RBF) takes about 72 seconds in unoptimized MATLAB implementation with a matrix of 459×59 .

Qualitative Examples of the Predicted Rating We present rate prediction examples in Fig. 4, and highlight the selected best cinemagraph for each user by colors. Note that the presented cinemagraphs are unseen data during training. Since preferences are not observed (surveyed) for every combination of users and cinemagraphs, unknown ground truth is indicated by the symbol “?”. It is well reflected by the proposed method that each user has their own subjective for best preferred cinemagraph, and overall the predictions have good matches with the selected best cinemagraphs by ground-truth.

Grouping Effect Given the user representation obtained by Ours (RBF), we visualize its 2-dimension embedding by t-SNE [7] in Fig. 5. The plot clearly shows clustered positions of users, which may imply that the intrinsic dimensionality of user space holds the low-dimensionality assumption. To see tendencies among neighbor users in the embedding space, we display true ratings of sampled users in Fig. 6. The users and groups are sampled by considering the proximity in the 2D embedding, and the cinemagraphs are sampled from a set in which entries are rated by all the presented users directly (none of them are inferred). The user IDs correspond to the node IDs in Fig. 5. It shows that each group has similar preference tendency, which implies that the users located at similar embedding space have similar preference characteristics.

4. Supplementary Tables

Hand-Designed Feature List Figure 7 is the hand-designed feature list used in the human preference learning part. The low-level hand designed feature has total 55-

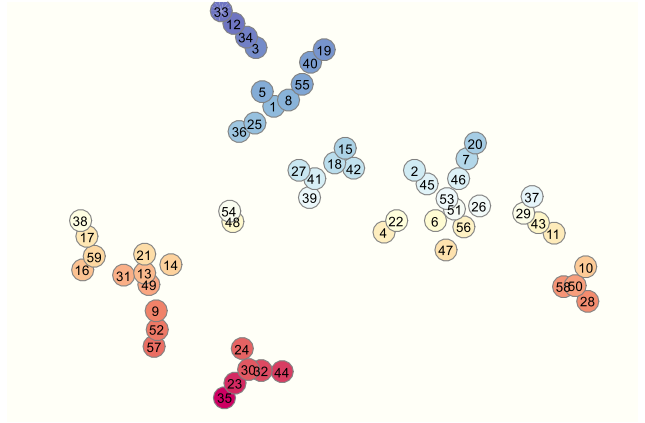


Figure 5: t-SNE visualization for 59 latent user features.

UIDs	Group A			Group B			Group C		
Cinema.	23	30	32	1	5	8	26	51	53
	1	1	1	3	1	4	4	4	5
	2	1	1	5	5	5	2	3	3
	2	2	3	5	5	5	2	3	2
	2	3	2	4	1	3	3	3	3
	3	1	1	5	5	5	2	5	5
	2	1	2	1	1	1	2	1	1
	2	3	5	2	1	1	2	2	2
	3	4	4	5	5	4	1	3	5
	3	4	4	2	1	2	3	2	2

Figure 6: Group behavior of user preference among intra- and inter-groups. The presented ratings are the numbers directly provided by each user. The users are sampled according to the proximity of embeddings in Fig. 5, and the presented cinemagraphs are sampled as those are rated by all the listed users, i.e., intersection set. Green color overlay indicates dynamic looping regions, otherwise static.

dimension. The presented order of this list is identical to the order of feature vector entries.

Semantic Class Mapping Table These semantic classes are based on PASCAL-Context [8]. This class mapping table in Fig. 8 is used to combine some categories and classify natural/non-natural categories in the semantic-based cinemagraph generation method. The dot . in the mapping

Type	Dimension	Feature
Face	15	facesizeMin facesizeMax facesizeMean facesizeMedian facesizeStd facexsMin facexsMax facexsMean facexsMedian facexsStd faceysMin faceysMax faceysMean faceysMedian faceysStd
Texture	5	sharpnessMin sharpnessMax sharpnessMean sharpnessMedian sharpnessStd
Motion flow	10	motionMin motionMax motionMean motionMedian motionStd motionSurroundMin motionSurroundMax motionSurroundMean motionSurroundMedian motionSurroundStd
Trajectory	15	tracklengthMin tracklengthMax tracklengthMean tracklengthMedian tracklengthStd trackBoundingBoxMin trackBoundingBoxMax trackBoundingBoxMean trackBoundingBoxMedian trackBoundingBoxStd trackTravelsMin trackTravelsMax trackTravelsMean trackTravelsMedian trackTravelsStd
Global loopability	5	globalLoopCostsMin globalLoopCostsMax globalLoopCostsMean globalLoopCostsMedian globalLoopCostsStd
Face ratio	5	faceRatiosMin faceRatiosMax faceRatiosMean faceRatiosMedian faceRatiosStd

Figure 7: Hand-designed feature list used in the human preference learning part. It has total 55 dimension.

class denotes that original class name is used and left intact.

References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- [2] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.
- [3] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 2009.
- [4] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.
- [5] 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.
- [6] Z. Liao, N. Joshi, and H. Hoppe. Automated video looping with progressive dynamism. *ACM Transactions on Graphics (SIGGRAPH)*, 32(4):77, 2013.
- [7] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [8] 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*, pages 891–898, 2014.
- [9] T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. S. Kweon. Fast randomized singular value thresholding for low-rank optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [10] J. Tompkin, F. Pece, K. Subr, and J. Kautz. Towards moment imagery: Automatic cinemagraphs. In *Conference for Visual Media Production (CVMP)*, pages 87–93, 2011.
- [11] 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.
- [12] M.-C. Yeh. Selecting interesting image regions to automatically create cinemagraphs. *IEEE MultiMedia*, 23:72–81, 2016.

ID	PASCAL-Context	Mapping class	Natural category
1	background	background	natural
2	aeroplane	.	
3	bicycle	bike	
4	bird	animal	
5	boat	.	
6	bottle	household item	
7	bus	.	
8	car	.	
9	cat	animal	
10	chair	chair	
11	cow	animal	
12	diningtable	household item	
13	dog	animal	
14	horse	animal	
15	motorbike	bike	
16	person	person	
17	pottedplant	grass	natural
18	sheep	animal	
19	sofa	chair	
20	train	.	
21	tvmonitor	.	
22	bag	.	
23	bed	.	
24	bench	chair	
25	book	.	
26	building	background	natural
27	cabinet	household item	
28	ceiling	background	natural
29	clothes	person	
30	computer	.	
31	cup	household item	
32	door	.	
33	fence	.	natural
34	floor	background	natural
35	flower	grass	natural
36	food	household item	
37	grass	grass	natural
38	ground	background	natural
39	keyboard	.	
40	light	.	natural
41	mountain	.	natural
42	mouse	.	
43	curtain	.	natural
44	platform	background	natural
45	sign	.	
46	plate	household item	
47	road	background	natural
48	rock	.	natural
49	shelves	household item	
50	sidewalk	background	natural
51	sky	sky & tree	natural
52	snow	water	natural
53	bedcloth	.	
54	track	background	natural
55	tree	sky & tree	natural
56	truck	.	
57	wall	background	natural
58	water	water	natural
59	window	.	
60	wood	.	natural

Figure 8: Class mapping table used in the semantic-based cinema-graph generation method.