Turning an Urban Scene Video into a Cinemagraph

Hang Yan
Washington University
St. Louis, USA
yanhang@wustl.edu

Yebin Liu
Tsinghua University
Beijing, China
liuyebin@mail.tsinghua.edu.cn

Yasutaka Furukawa
Washington University
St. Louis, USA
furukawa@wustl.edu

Abstract

This paper proposes an algorithm that turns a regular video capturing urban scenes into a high-quality endless animation, known as a Cinemagraph. The creation of a Cinemagraph usually requires a static camera in a carefully configured scene. The task becomes challenging for a regular video with a moving camera and objects. Our approach first warps an input video into the viewpoint of a reference camera. Based on the warped video, we propose effective temporal analysis algorithms to detect regions with static geometry and dynamic appearance, where geometric modeling is reliable and visually attractive animations can be created. Lastly, the algorithm applies a sequence of video processing techniques to produce a Cinemagraph movie. We have tested the proposed approach on numerous challenging real scenes. To our knowledge, this work is the first to automatically generate Cinemagraph animations from regular movies in the wild.

1. Introduction

Our world is dynamic. Imagine you are standing in the middle of Times Square surrounded by constant noise, cars passing by, or flashy billboards showing advertisements at every second. A fundamental challenge in Computer Vision is to model and visualize dynamic environments. Cinemagraphs, still photographs containing minor and repeated animations [1], are one of the most successful examples in capturing such scene dynamics. Their subtle animations are effective in capturing the "moment" with striking visual effects.

The generation of high-quality Cinemagraphs have so far required static cameras with carefully configured scenes [1, 2] or interactive tools [14]. No compelling techniques exist in the automatic conversion of regular videos into Cinemagraphs. Online photo storage services, such as Google Photos, automatically produce short animations from user images and movies. However, their animations are either a simple image slide-show or a trimmed movie segment looping forward and backward unnaturally.

We seek to make the first step towards automated Cinemagraph generation from regular movies with moving cameras in the wild. The key insight is that even subtle animations yield striking visual effects, where our approach is to selectively and precisely segment regions that lead to high-quality animations. In particular, we focus on urban environments or night-time settings, where neon-signs, displays, or flashy billboards decorate a scenery. Such a geometry is static, making the modeling task significantly easier, while their appearance adds effective dynamics to the scene visualization.

Formally, this paper turns a regular video capturing urban scenes into a Cinemagraph-style animation in three steps. First, we utilize existing 3D reconstruction techniques to warp an input video into the viewpoint of a reference frame. Second, novel temporal analysis algorithms are applied to the warped video to give regions where high-quality animations can be produced. These regions have static geometry with cyclically or non-cyclically changing appearance. Third, we perform a sequence of video processing techniques to generate high-quality animations for the segmented regions, while fixing the rest of the pixels to the reference frame.

The contributions of this paper are two fold. The technical contribution lies in the effective temporal analysis of noisy warped videos to enable the segmentation and classification of visually interesting regions. The system contribution is the fact that this is the first effective system automatically generating Cinemagraph animations from regular movies.

2. Related work

Dynamic scene reconstruction has been a fundamental problem for Computer Vision. Significant progress has been made for lab-environments, where multiple calibrated and synchronized video cameras are the input. A successful system has been demonstrated for a human body [26], a human face [4], or multiple people with interactions [13, 8]. Dynamic scene reconstruction from YouTube videos has

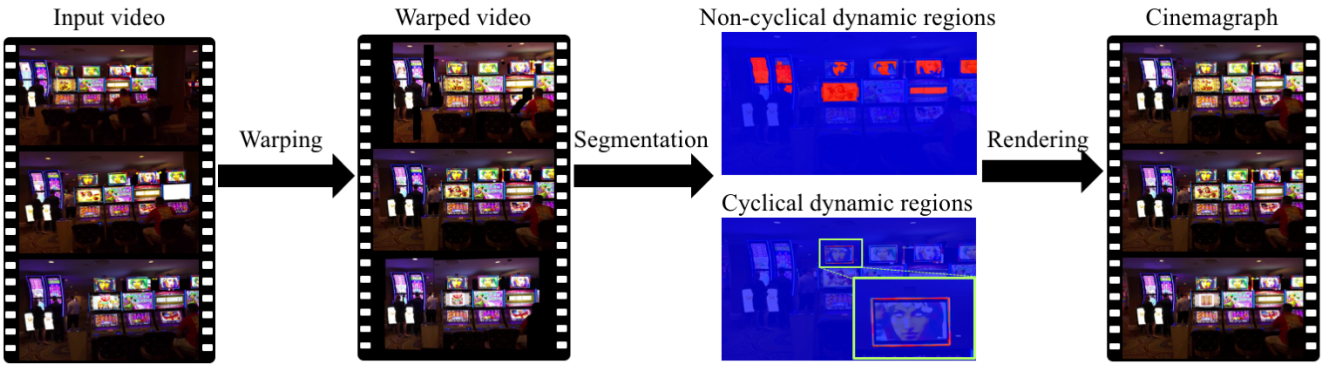

Figure 1: System overview. Given an input video clip (left), the system first warps all frames into the reference view (left middle) by computing camera poses and the reference depth-map. Regions that are visually attractive are selected, namely regions with static geometry and dynamic (cyclical or non-cyclical) appearance (right middle). The system then creates a cinemagraph rendering by only animating those regions while fixing other pixels to the reference frame (right).

been recently proposed [12]. They jointly reconstruct the static background and dynamic foreground objects. However, the adopted visual hull reconstruction leaves noticeable protruding artifacts on their models. Dynamic Fusion reconstructs non-rigidly deforming objects from an RGBD stream, but does not focus on dynamic appearances [21].

More macro scale dynamics (e.g., scene changes over months or years) can be detected by analyzing a set of images acquired by standard cameras [30], stationary surveillance cameras [23], vehicle-mounted cameras [29], or community photo sharing websites [20, 18]. In particular, the time lapse reconstruction [18] has produced impressive spatio temporal 4d models. They are one of the most successful examples of varying geometry and appearance over space and time. However, they require massive amount of photographs, limiting their applications to a small number of landmarks in the world. In contrast, we seek to realize Cinemagraph-quality dynamic scene visualization from a single regular movie.

The Cinemagraph creation has also been studied. Impressive results have been obtained by semi-automatic systems [14, 2], or with templated input videos [3]. Automatic cinemagraphs systems [28, 32] create the mask for animated regions by motion analysis. However, these systems are successful only when the camera does not move and the scene is mostly static. A data-driven single-image approach has produced impressive animations [16], but the dependence on the database currently limits their application ranges. A simple yet appealing system [24] has been proposed to create endless loops by randomly jumping between similar frames. However, their system requires the video to be cyclic. In contrast, this paper proposes an automated approach for regular movies with moving cameras.

3. System overview

This paper proposes a novel system that allows us to convert standard movies with moving cameras capturing urban scenes into Cinemagraph animations. Our system consists of three steps: warping, segmentation, and rendering (See Figure 1). First, we use Structure from Motion (SfM) and Multi-View Stereo (MVS) algorithms to warp the input video into the viewpoint of a reference frame via a depth-map based image morphing. Second, effective temporal analysis and segmentation algorithms are applied on the warped video to give regions that lead to good animations. Finally, we render a high-quality Cinemagraph movie by a sequence of video processing techniques in each segmented region to mitigate artifacts from warping, while fixing the rest of the pixels to the reference frame. The warping step is an application of standard techniques, while the last two steps, in particular the segmentation step, exhibit technical contributions in this paper.

4. Spatial alignment by image warping

The video warping is based on standard 3D reconstruction techniques with minor modifications for being robust against scene dynamics. First, we use a SfM software TheiaSFM [27] to estimate camera poses. Given a reference frame I_r, we estimate a depth-map based on 100 neighboring frames (i.e., 50 frames before and after) via a standard MRF formulation. The range of scene depth at I_r is estimated from visible 3D points provided by SfM, with 1% nearest and farest points discarded for robustness. The inverse depth space is then uniformly discretized into 128 labels.

Following the idea in [15], we compute a matching score between I_r and each neighboring image, then sum up the best half of these scores as the overall unary term to com-

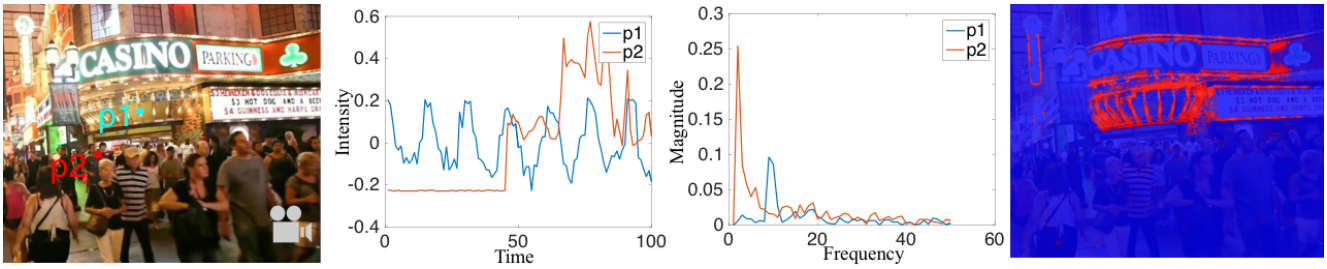

Figure 2: Fourier analysis for cyclical dynamic pixels. Left: Two pixels $p1$ and $p2$ are marked for illustration in the reference frame. Left middle: the intensity patterns of two pixels over time. The intensities are preprocessed to have zero means. Right middle: the frequency magnitudes of temporal pattern. Since $p2$ has cyclical dynamic appearance, the peak occurs at a high frequency, while for $p1$ the peak occurs at a low frequency. Right: the resulting cyclicity score, encoded in the red color channel.

pensate for occlusions and scene dynamics. The matching score is defined as one minus the Normalized Cross Correlation over 5×5 image patches, truncated at 0.3 for robustness. The pairwise term is a truncated linear function of the absolute label differences with a truncation at 4. We multiply 0.15 to the pairwise term. Given the estimated depth-map, we warp all the neighboring frames into the viewpoint of I_r via standard backward warping, while taking into account occlusions via Z-buffering. Occluded pixels are ignored in the next segmentation process and will be in-painted in the last rendering process.

5. Dynamic appearance segmentation

Carefully choosing regions to animate is the key to successful Cinemagraph creation. Our approach is to conduct temporal analysis on the spatially aligned warped-video, while focusing on two types of appearances common in urban scenes.

5.1. Non-cyclical dynamic appearance

Digital displays or billboards are popular visual attractions in urban downtowns, night clubs, or store-fronts in shopping malls. Detection and segmentation of displays pose challenges to existing techniques as they could show arbitrary contents. Our system detects these regions by 1) segmenting the warped video into 2D segments by a novel feature vector encoding characteristic appearance changes, and 2) classifying these regions by a random forest trained from manually annotated video clips.

Warped video segmentation: We perform hierarchical bottom-up 2D segmentation of a warped video (See Fig. 4). [1] Our contribution lies in the novel distance metric for a pair of spatial regions, defined as $D_T + 0.1D_A$. The metric utilizes the warped video to analyze temporal appearance changes D_T in addition to the pixel appearance changes D_A.

[1] The approach starts from pixels and greedily merges the closest pair if the distance of their feature vectors is below a threshold. It iterates between merging and increasing the threshold until everything merges to a single segment. The original algorithm [10] operates on 3D pixel volumes, while our system performs 2D segmentation by treating each pixel as a 1D array.

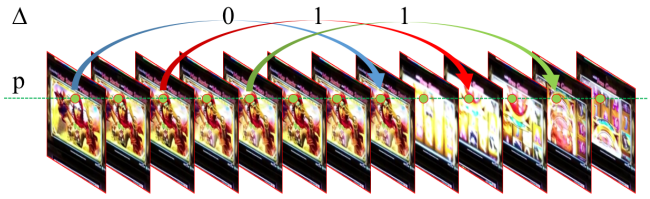

Figure 3: Our temporal binary pattern encodes the temporal characteristic of a pixel (p) by checking once in every α frames if there will be significant color change after β frames. The figure illustrates a case when $\alpha = 2$ and $\beta = 7$.

The inspiration of D_T comes from local binary pattern [7, 22] for feature matching, which we employ in the temporal domain. Let $I_p(f)$ be the mean color of a pixel p (or a region after merging) at frame f in the warped video. We will drop p from the notation below for simplicity. Our binary temporal pattern descriptor checks once in every α frames if there will be significant color change β frames ahead (See Figure 3). More precisely, the i_{th} bit of the binary pattern is defined as

$$\Delta(i) = \mathbb{1}(\|I(\alpha i + \beta) - I(\alpha i)\|_2 > \theta). \quad (1)$$

$\mathbb{1}$ denotes the indicator function and checks if the color difference in RGB space is more than $\theta = 100$.

To capture both local and long term appearance changes, we use two sets of parameters: $\alpha_1 = 4, \beta_1 = 4$ for $\Delta_1(i)$, and $\alpha_2 = 2, \beta_2 = N/2$ for $\Delta_2(i)$, where N is the number of input frames. The final binary descriptor is the concatenation of the two:

$$T = [\Delta_1(0), \Delta_1(1), \ldots, \Delta_2(0), \Delta_2(1) \ldots].$$

The distance between two binary temporal descriptors D_T

is computed by the Hamming distance of the two binary vectors normalized by the feature dimension.

D_A measures the pixel appearance difference between two segments by computing the one minus the normalized cross correlation of two color histograms. The histogram for each segment is constructed from the LAB values of all pixels inside the 2D region from all frames, with 8 bins per channel.

We start the process with the initial merging threshold set to 0.2 and increase it by a factor of 1.5 every time. We use segmentation results at three different levels of granularity to generate (overlapping) image segments for robustness, in particular, at 60%, 70% and 80% levels [10].

Classification: We build a binary random forest classifier based on appearance, temporal changes, shape, and position features. We have obtained the training data by downloading stationary video clips of popular urban scenes from YouTube. Then, we have manually annotated 2D regions such as displays and billboards. Please refer to the supplementary material for detailed feature design and training process. At test time, we pass all segments from three granularity levels into the classifier and take the union of all positive segments. We ignore mostly invisible segments, that is, if more than half the pixels are invisible (i.e., project outside the view or fail in the Z-buffering test during warping) in more than half the neighboring frames.

5.2. Cyclical dynamic appearances

Repeated advertisements or flashing neon-signs are also symbolic structures in many urban scenes, especially at night time. Due to the fact that these regions are often small and isolated, standard motion analysis and segmentation algorithms perform poorly. We propose a simple but powerful temporal analysis algorithm based on Discrete Fourier Transform (DFT) to recognize these pixels.

For each pixel of the warped video, we compute the 1D DFT of its intensities over all frames, then conduct a frequency analysis (See Fig. 2). For ideal cyclical intensity patterns, we should observe a clean peak among high frequency components, while low magnitudes at low frequencies. To be robust against errors from warping, instead of computing a single score using all the N input frames, we look for the optimal interval of at least $N/2$ frames. More precisely, we compute the cyclicity score of a pixel from frame i to j as

$$C_{cyc}(i,j) = \frac{\max_{k>\tau} |F_k|}{\max_{k\leq\tau} |F_k|}.$$

$|F_i|$ denotes the magnitude of a DFT component. [2] τ is the boundary between the low and high frequency components, which is set to 4 throughout the experiments. The final score is the maximum over all the possible frame intervals containing at least $N/2$ frames:

$$C_{cyc} = \max_{(j-i)\geq N/2} C_{rep}(i,j). \tag{2}$$

We animate a pixel if 1) its score (2) is greater than 2.5 and 2) its 80th percentile of intensities over all frames is greater than 127.

[2] We only use the first half of the DFT coefficients as their magnitudes are symmetric for real-valued arrays. We also discard the direct component by subtracting the mean intensity before DFT.

6. Cinemagraph rendering

Our system renders Cinemagraph animations in the detected regions using frames from the warped video, while keeping the remaining pixels fixed to the reference frame. The cyclical pattern often consists of small segments and the animation in the optimal interval computed by the formula (2) looks natural without any post-processing.

Non-cyclical appearance segments are more challenging. We first in-paint visibility holes by Laplacian smoothing over space and time. Next, we apply geometric stabilization by homograph warping using static feature points. More concretely, feature tracks are generated [25, 6] and filtered by the constraints that 1) the track has to last for at least 10 frames and 2) the standard deviation of the tracked pixel coordinates must be less than 2 pixels in both x and y. Linear least square are used to compute the homography warping from each frame to the reference.

As in existing literature [5, 18], we apply intensity regularization. This is crucial for our warped video, which suffers from severe rendering artifacts. Standard techniques such as temporal median filtering [5] or global least squares optimization [18] have produced compelling results for many of our examples. However, they show two typical failure modes when a segment exhibits rapid optical flow motions and/or abrupt temporal changes. First, they over-regularize high frequency temporal signals. Second, inconsistencies arise across pixels due to the lack of spatial regularization.

Our approach is to rearrange pixel colors of a segment throughout the frames as a 2D matrix and obtain a low-rank approximation. This method achieves moderate temporal and spatial regularization. RPCA [31] is the choice of our machinery, which has been successfully used for various image and video analysis tasks, but not for high-quality movie rendering to our knowledge.

More concretely, we concatenate pixels of a segment in a single frame to a row vector, and stack them across the frames to form a matrix P. We use Accelerated Proximal Gradient [17] method to minimize the standard RPCA formulation:

$$\|A\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad A + E = P. \tag{3}$$

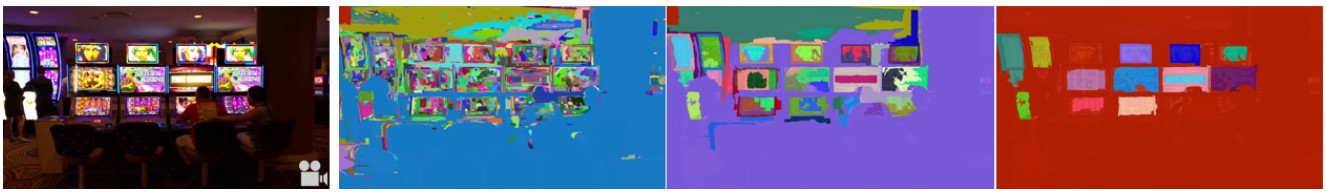

Figure 4: Hierarchical 2D video segmentation. The left shows the reference frame. The right shows the segmentation results at 0%(lowest), 60%, and 80% hierarchy levels, respectively.

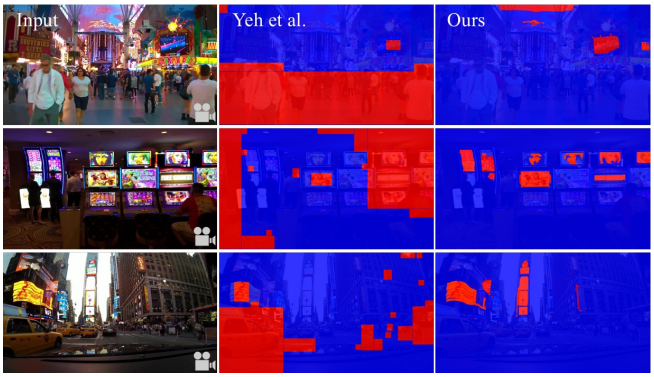

Figure 5: Segmented regions for Cinemagraph animation by Yeh et al. [32] and our approach. The method by Yeh et al. assumes a static camera as in any other Cinemagraph creation methods. It simply identifies a region with large optical flow motions, and fail to identify effective regions for Cinemagraph animations.

Figure 6: Adaptive appearance regularization. Left: with $\lambda = 0.02$. Notice the occluders inside the red circles. Middle: output of RPCA with $\lambda_r = 0.011$, which is automatically selected. The occluding artifact is mitigated. Right: output of RPCA with $\lambda_r = 0.005$. The frames are overly smoothed.

The minimization of the nuclear norm $\|A\|_*$ achieves spatial and temporal regularization. We solve the problem for each channel independently and rearrange A as the output pixel values. We have found that it is important to adaptively tune the scalar weight λ depending on the video content, which varies significantly across examples. Intuitively, a rich video content with fast optical flows or temporal changes should still have large nuclear norm. Therefore, we set λ to be proportional to the "richness" of the video content, characterized by Δ_1 from Section 5.1. More precisely, we set λ to be 0.005 + 0.015 γ, where γ is the number of '1' in Δ_1 divided by its dimension (See Figure 6).

To create endless loops, we render segments with cyclical appearance over and over again inside the interval found by the optimization (2). For non-cyclical dynamic segments, we create loops by playing the video forward and backward. Each segment is looped independently.

7. Experimental results

We have implemented the proposed system in C++ and used Intel Core I7 CPU with 32GB RAM and NVIDIA Titan X GPU (for stereo matching score evaluation). We have downloaded various footages from YouTube such as walkthroughs or drive-throughs of urban scenes. We have also recorded walk-through videos by ourselves. Most of the input videos are 10 seconds long (i.e., 300 input frames), while some last for 2 minutes. Our movie collections span indoors/outdoors, day/night, and various places such as urban downtowns, city streets, casinos, shopping malls, or university buildings. Notice that SfM processes all the input frames, but our algorithm only needs a reference frame and 100 neighboring frames. The running time of our system after the SfM step ranges from thirty minutes to an hour, depending on the frame resolution and the number of display segments, where the bottleneck lies in stereo computation and RPCA. Our main technical contribution, namely the segmentation, finishes in few seconds.

Figure 10 shows four of the input videos, segmentation results of the two algorithms, and the output Cinemagraphs. Our system enables automatic high-quality Cinemagraph creation from videos with moving cameras in the wild, where all the existing approaches require a static camera with a clean scene and/or human manual interventions, to our knowledge. Please refer to the supplementary material for complete experimental results and movies.

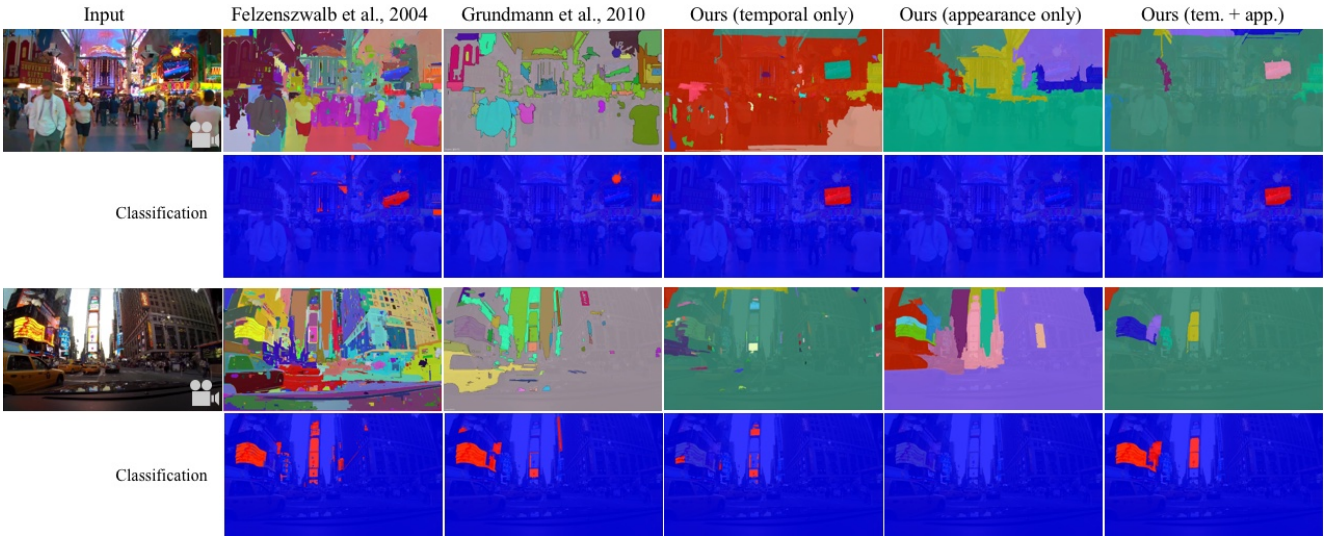

Figure 7: We evaluate five different segmentation algorithms on two examples. For each example, the segmentation result is shown in the first row and the corresponding classification result is shown in the second row. The combination of the temporal and appearance (temp. + app.) information allows us to effectively segment regions that lead to good animation. Algorithms of Felzenszwalb et al. [9] (with threshold parameter set to 500) and Grundmann et al. [10] lacks in temporal appearance information and fail to group pixels in the display in the top example. Only using the temporal information (temporal only) produces incomplete segments for partially dynamic displays (yellow display on the left of the bottom example). Only using appearance information (appearance only) fails to capture display segment as in Felzenszwalb et al. and Grundmann et al. Segmentation result from 80% hierarchy are shown in the right four columns.

Figure 8: Comparison of different intensity regularization algorithms. Left: input frame. Right: rendering using temporal median filter with radius of 5, global least-square optimization with the smoothing weight 50, and our adaptive RPCA. The first two algorithms regularize each pixel independently, causing inconsistency across pixels under fast motion. In contrast, our algorithm jointly regularize all pixels inside a region.

Figure 5 demonstrates that our segmentation process effectively identifies regions that lead to high quality animations. Since no automated Cinemagraph generation method exists for a general movie, we have supplied our warped videos to an existing algorithm assuming a static camera [32] for comparison. However, their algorithm simply looks at a rectangular regions with large optical flow motions, and cannot handle severe rendering artifacts or rich dynamics in our movies.

Figure 7 shows the effectiveness of our new pixel dis-

Figure 9: Failure cases. Left: the appearance inside the red oval is highly distorted in the second example due to geometric errors from SfM and stereo. Middle: we fail to detect this display, which is mostly static with minimal appearance dynamics. We need more training data. Right: defects caused by segmentation and large occluders. Although we handle small and fast occluders by intensity regularization, large and slow occluders still cause visual defects in the final rendering.

tance metric for display segmentation. The feature vector allows us to extract image segments that have similar temporal changing patterns. We compare our algorithm with Felzenszwalb et al. [9] on the reference image and Grundmann et al. [10] on our warped video. Both methods fail to extract segments that have similar temporal appearance characteristics.

Figure 6 shows the effectiveness of our adaptive appearance regularization, where we control the low-rank regularization weight depending on the richness of the video content. Figure 8 shows our rendering results against two standard intensity regularization techniques: temporal median filter [5] and global least-squares optimization [18]. These methods have no spatial regularization (i.e., per-pixel operation), causing inconsistencies across pixels in the presence of fast optical flow motions. Our low-rank approximation technique outperforms in such cases with moderate spatio-temporal regularization.

Applications: The capability to turn general videos into Cinemagraph animations opens up potentials for novel applications. For instance, in the field of scene visualization, image-based rendering navigation has become the golden standard (e.g., Google Maps Street View) [11, 19], where a user looks at a real photograph at one location, and jumps between locations via transition rendering. However, photographs are all static without any dynamics in these systems. While directly serving videos might be a solution to visualize scene dynamics, they require a lot more data space/transfer and constraint the navigation strictly on the video path. Replacing images with Cinemagraphs allows one to experience scene dynamics at each location as well as free navigation in a scene. Cinemagraphs animate only a fraction of an image and requires minimal extra data space. In particular, we demonstrate this next-generation Cinemagraph-based rendering navigation, by taking a long walk-through video, generating Cinemagraph animations at sub-sampled frames, then form a navigation graph by connecting these frames. Furthermore, it is also easy to replace animating contents by another media for virtual advertisement, which might prevail in the near future with the emerging VR and AR. Please see the supplementary video for examples.

8. Limitations and future work

This paper proposes the first effective system that turns regular movies with moving cameras into high-quality Cinemagraph animations at urban environments. Automatic Cinemagraph creation from regular video is still a very challenging problem, and we have observed a few major failure modes (See Figure 9). First, our system expects that SfM utilizes a static part of a scene to produce camera poses and MVS interpolates rough geometry over dynamic regions. These assumptions might fail at highly dynamic regions, causing unnatural distortions in the animated contents. The second failure mode is in the classification. Our training data come from movie clips by stationary cameras, which look different from the warped videos. We need more training data, potentially, annotating the warped videos from our algorithm for training. The last failure mode is in the rendering. While RPCA is very powerful in suppressing artifacts, it still fails under the presence of severe occluders, such as the long appearance of pedestrians in front of a camera. Utilization of semantic segmentation techniques is our future work to make our system further robust against occluders.

This paper makes a first important step towards automated high-quality dynamic scene visualization from regular movies by mass consumers. We hope that this paper will fuel a round of new research, tackling more diverse set of dynamics in our world.

Acknowledgement

This research is partially supported by National Science Foundation under grant IIS 1540012 and IIS 1618685, and Microsoft Azure Research Award.

Figure 10: Each column shows sample input frames (the middle frame as the reference), cyclical dynamic segments, non-cyclical dynamic segments, and sample output frames. Segmentations for cyclical dynamic pixels and output frames are cropped to the red and green bounding boxes to better illustrate the details. Please see our supplementary video for the full assessment of our results and more examples.

References

[1] Cinemagraph. http://cinemagraphs.com/. 1

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

[3] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Automatic cinemagraph portraits. In *Computer Graphics Forum*, volume 32, pages 17–25. Wiley Online Library, 2013. 2

[4] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph.*, 30:75:1–75:10, August 2011. 1

[5] E. P. Bennett and L. McMillan. Computational time-lapse video. In *ACM Transactions on Graphics (TOG)*, volume 26, page 102. ACM, 2007. 4, 7

[6] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001. 4

[7] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012. 3

[8] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015. 1

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal on Computer Vision*, 59(2):167–181, 2004. 6, 7

[10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2141–2148. IEEE, 2010. 3, 4, 6, 7

[11] G. Inc. Google maps. http://maps.google.com. 7

[12] D. Ji, E. Dunn, and J.-M. Frahm. 3d reconstruction of dynamic textures in crowd sourced data. In *ECCV*, pages 143–158. Springer, 2014. 2

[13] H. Joo, H. S. Park, and Y. Sheikh. Map visibility estimation for large-scale dynamic 3d reconstruction. In *Computer Vision and Pattern Recognition*. IEEE, 2014. 1

[14] N. Joshi, S. Mehta, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen. Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 251–260. ACM, 2012. 1, 2

[15] S. B. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58(2):139–163, 2004. 2

[16] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Graphics (TOG)*, 33(4):127, 2014. 2

[17] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 61, 2009. 4

[18] R. Martin-Brualla, D. Gallup, and S. M. Seitz. Time-lapse mining from internet photos. *ACM Transactions on Graphics (TOG)*, 34(4):62, 2015. 2, 4, 7

[19] Matterport. Matterport: 3d for the real world. http://matterport.com. 7

[20] K. Matzen and N. Snavely. Scene chronology. In *ECCV*, pages 615–630. Springer, 2014. 2

[21] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015. 2

[22] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE, 1994. 3

[23] K. Sakurada, T. Okatani, and K. Deguchi. Detecting changes in 3d structure of a scene from multi-view images captured by a vehicle-mounted camera. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 137–144. IEEE, 2013. 2

[24] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press/Addison-Wesley Publishing Co., 2000. 2

[25] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994. 4

[26] J. Starck and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications, IEEE*, 27(3):21–31, 2007. 1

[27] C. Sweeney. Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org. 2

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

[29] A. O. Ulusoy and J. L. Mundy. Image-based 4-d reconstruction using 3-d change detection. In *Computer Vision–ECCV 2014*, pages 31–45. Springer, 2014. 2

[30] T. Y. Wang, P. Kohli, and N. J. Mitra. Dynamic sfm: Detecting scene changes from image pairs. *Comput. Graph. Forum*, 34(5):177–189, 2015. 2

[31] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009. 4

[32] M.-C. Yeh and P.-Y. Li. An approach to automatic creation of cinemagraphs. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1153–1156. ACM, 2012. 2, 5, 6