An Approach to Automatic Creation of Cinemagraphs

Mei-Chen Yeh Po-Yi Li
Department of Computer Science and Information Engineering
National Taiwan Normal University, Taipei, Taiwan
{myeh, 699470668}@ntnu.edu.tw

ABSTRACT

A cinemagraph is a new type of medium that infuses a static image with the dynamics of one particular region. It is in many ways intermediate between a photograph and a video, and has a number of attractive potential applications, such as the creation of dynamic scenes for games and interactive environments. However, creating cinemagraphs is time consuming and requires certain level of proficiency on photo editing techniques. In this paper, we present a fully automatic approach that creates cinemagraphs from video sequences. Specifically, we view cinemagraph construction as a constrained optimization problem that seeks a sub-volume in video with the maximum cumulative flow fields. The problem can be efficiently solved by a branch-and-bound search scheme. A user survey is conducted to understand user preferences and demonstrate the performance of the proposed approach. The findings of this study should provide information for various design choices for an easy and versatile authoring tool for cinemagraphs.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications.

General Terms

Algorithms, Experimentation

Keywords

Cinemagraph, motion analysis, user study

1. INTRODUCTION

A cinemagraph is a new form of digital photography, recently exploded by a fashion photographer Jamie Beck and a visual designer Kevin Burg [10]. A cinemagraph is neither a photograph nor a video; however, it is in many ways intermediate between them. A cinemagraph contains mostly a static picture with a touch of movements that capture dynamic entities of an object. For example, the *Daily Prophet* (i.e. the wizard newspaper) in the Harry Potter movies can be viewed as an example of cinemagraph. By freezing most of the moving elements and animating one or just a few, cinemagraphs are able to draw attention to a certain object in a more creative and effective manner compared to traditional mediums such as photographs and videos.

Given a short video clip, cinemagraphs can be manually made with image processing software such as Photoshop and After Effects [10]. For end users, the manual process of creating a cinemagraph is sometimes tedious and annoying (Section 2), and requires certain level of proficiency on photo editing techniques. Therefore, it would be beneficial to devise a computational approach for such a process. Among the cinemagraph construction steps, the major challenge is the generation of masks and layers, in which a user has to carefully select image parts and video frames to animate, and a still image to be used as the background. To automate the process, the following problems need to be addressed: (1) How to detect and analyze the dynamic characteristics of a scene or event? (2) How to determine the moving regions that give a most visually appealing cinemagraph? (3) How to sample video frames to generate a seamless loop? These questions are nontrivial as input video clips, especially those captured with a smartphone, may have very complex motion patterns combined from ego-motion and scene motion.

In this paper, we present a fully automatic approach for constructing cinemagraphs from video clips. Our emphasis is on a selection framework that analyzes the dynamic characteristics from frame sequences, with the goal to generate masks and layers that composite a beautiful cinemagraph. More specifically, we formulate an optimization problem that seeks the best selection strategy by maximizing the cumulative local motion dynamics within a time period, with the conditions that the sampling points are determined to ensure a seamless loop, which we refer to as the smoothness constraint. To handle the large search space in video, we have developed a fast algorithm based on the branch-and-bound scheme [2] (Section 3). Figure 1 illustrates an overview of our method. The exemplar video contains two children playing on a swing set. Each frame is first characterized by a collection of motion primitives, from which we compute a score for each pixel that indicates its likelihood of belonging to an interesting motion trajectory. By searching the regions of highest accumulated scores that also meet the smoothness constraint, we locate the motion masks and the static background layer, and generate the cinemagraph.

There are several advantages of our method. First, the method is fully automatic and requires no training phrase. Second, the method is data-driven and does not rely on object tracking or background subtraction. Empirical results illustrate the applicability of our method to real-world videos (Section 4).

In the remainder of this paper, we first describe the manual process and some related works in this field. Section 3 presents the main approach that automates the construction of cinemagraphs. We then present experiments and conclude the paper with a short discussion and a few future directions.

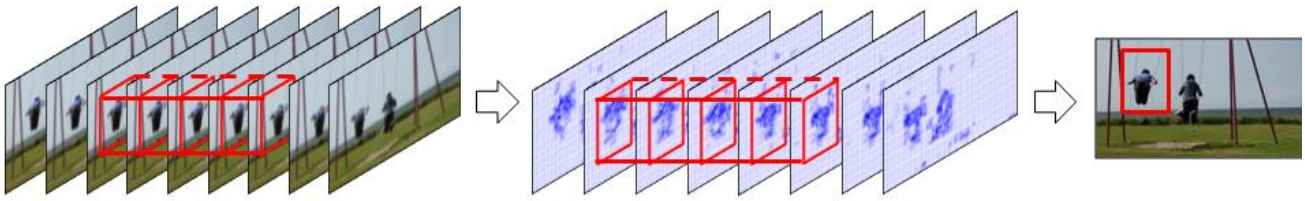

Figure 1. The automatic cinemagraph construction problem is formulated as searching for a sub-volume in video that has the maximum cumulative score. Each blue dot represents a spatiotemporal feature point which contributes a vote based on its own optical flow.

2. RELATED WORK

2.1 Manual Process

Cinemagraphs are usually made with professional image editing software. Interesting readers may find a number of online tutorials for creating cinemagraphs. A general guideline includes the following steps: (1) frame the input video—import the entire or a portion of the video and make every frame into its own separate layer; (2) catch the moving moment—figure out the frames that capture the desired movement, and choose one layer that shows the consistent, non-moving elements of the cinemagraph; (3) mask—edit the static layer by using a vector mask that filters out the moving elements; (4) synthesize—place a adjustment mask over all layers and generate the animation. One of the challenges a user is often confronted in the initial test runs is the creation of a fairly smooth loop in the movements. Trial and error has usually been the main method of finding how to smoothly loop a cinemagraph.

2.2 Previous Work

The work in [6] is probably the first that explores the automatic creation of cinemagraphs where the authors termed the type of medium a *video texture*. The method identifies a number of transition points where a frame can jump to another (not necessary an adjacent frame) in the video sequence and generates an infinite loop from a finite length video. Recently, Tompkin *et al.* created a cinemagraph authoring tool that combines several video processing techniques including video motion stabilization and segmentation [7]. The approach is semi-automatic as a user is still required to select the moving regions in the process. More recently, iPhone Apps and tools that create cinemagraphs have been commercially launched, such as Cliplets [1], Kinotopic [11] and iCinegraph [12]. Although these tools aim to simplify the creation process, they are still semi-automatic and the movement selection process is somewhat cumbersome. In this paper, we present a new formulation of the problem, which seeks the best selection strategy for masks and layers from a video sequence.

3. APPROACH

3.1 Problem Formulation

Given a video sequence $[I_1, I_2, \ldots, I_N]$, the task of automatic cinemagraph construction is to identify both where (the spatial location) and when (the temporal location) of the interesting dynamic behavior of an object in a video. We consider the problem as a constrained local spatiotemporal score maximization problem. Suppose each frame is characterized by n by m values, each represents a score that estimates a pixel's likelihood of belonging to the motion mask (Section 3.2), our objective is to find a spatiotemporal cuboid V^* that maximizes the cumulative score:

$$V^* = \arg\max_{V \subseteq \mathcal{V}} \sum_{p \in V} \mathrm{s}(p) = \arg\max_{V \in \mathbf{\Lambda}} f(V), \quad (1)$$

where $f(V) = \sum_{p \in V} \mathrm{s}(p)$ is the objective function and $\mathbf{\Lambda}$ denotes the candidate set of all valid cuboids in $\mathcal{V}$. The optimal solution has six parameters $\{t^*, b^*, l^*, r^*, s^*, e^*\}$ to be determined, where t^*, b^*, l^*, r^* denote the spatial location of the bounding box in the frame, and s^*, e^* denote the start and end positions on timeline. Furthermore, the following conditions must be held:

$$d(R_{s^*}, R_{e^*}) \leq \epsilon, \quad (2)$$

$$|s^* - e^*| \geq T. \quad (3)$$

Equation (2) indicates that the cropped areas in I_{s^*} and I_{e^*} should be similar to ensure a seamless loop, where R_i denotes the region specified by t^*, b^*, l^*, r^* in I_i. Equation (3) suggests that the extracted dynamic behavior should be lasting at least for T frames. Given a $n \times m \times t$ video clip, the solution space is in the order of $O(n^2m^2t^2)$. In the following, we present the objective function used in Eq. (1), and describe an implementation of the branch-and-bound method [2] for searching V^* efficiently.

3.2 Motion-based Scoring

The score computation takes as input a set of frame-to-frame optical flow fields. Let $F(p) = (F_x(p), F_y(p))$ denote an optical flow vector field that defines a 2D vector at each pixel location $p = [x\ y]$. A frame is characterized by the magnitude of the optical flow, i.e.,

$$s(p) = \sqrt{F_x(p)^2 + F_y(p)^2}. \quad (4)$$

Each score is subtracted a constant c, such that we introduce positive and negative score values and obtain a representation where positive score values imply relatively large motions. Next, individual pixel can vote for the motion mask—the score of a bounding box is the summation of the pixel scores. Now we have a 3D array of positive and negative scores which are passed to the sub-volume search for localizing the optimal mask.

3.3 Efficient Sub-volume Search

The branch-and-bound scheme has been shown to enable efficient maximization of a quality function over all possible sub-images in [2]. This method outperforms sliding-window-based methods as it always examines the rectangle set that looks most promising in terms of its quality bound, and does not impose restrictions on the values that the rectangle coordinates can take.

The idea can be extended to find the optimal 3D sub-volume in videos. However, the search of 3D sub-volume introduces two

additional parameters (start and end points in the time dimension). As the complexity of the branch-and-bound method grows exponentially in the number of dimensions, the method is too slow for our application. For example, it takes minutes even to hours to process a 10-second video clip. In our implementation, we apply the efficient search method in [9], which decomposes the 6D parameter space into a 4D spatial parameter space and a 2D temporal parameter space. We first employ the branch-and-bound strategy to search the spatial parameter space. Once the spatial window is determined, we can easily search for the optimal temporal segment. With the objective function previously described, the upper bound estimation of the branch-and-bound search is

$$f(R) \leq \min\{\hat{F}_1(R), \hat{F}_2(R)\}, \quad (5)$$

where $\hat{F}_1(R) = F(R_{\min}) + \sum_{i \in R_{\max}, i \notin R_{\min}} F^+(i)$ and $\hat{F}_2(R) = F(R_{\max}) - \sum_{i \in R_{\max}, i \notin R_{\min}} G^-(i)$. $R_{\max}$ and $R_{\min}$ denote the largest (or the smallest) possible region, and $F(\cdot)$ and $G(\cdot)$ denote the maximum (or the minimum) sum of a region (or a pixel i) along the temporal direction. $F^+(i) = \max(F(i), 0)$ and $G^-(i) = \min(G(i), 0)$. Please refer to [9] for more details. Unlike the method in [9] that takes different search strategies in the two subspaces, we use the branch-and-bound method in both subspaces for implementation simplicity.

3.4 Post Processing

Once the motion regions and the frame segment have been selected, the next step is to generate a seamless video loop. Similar to the methods in [6][7], we compute the sum of squared difference (SSD) between pairs of moving regions with an interval larger than or equal to a threshold *T*. Using the values returned by the efficient search method as the initial values, the two parameters s^* and e^* are refined to meet the constraints (Eq. (2) and Eq. (3)). A small SSD between R_{s^*} and R_{e^*} is required to obtain a video loop that has unnoticeable temporal artifacts. Furthermore, I_{s^*} is selected as the background layer. In case the footage does not lend itself to loop well, we interpolate frames at the end of the looping cinemagraph to smoothly return to the starting frame. The final step is to composite the moving region onto the background layer, per frame in the interval $[I_{s^*}, I_{e^*}]$, and output the file in the Graphics Interchange Format (GIF) format.

4. EXPERIMENTAL RESULTS

As rating a cinemagraph is subjective, we conducted a user survey to understand user preferences and demonstrate the performance of the proposed approach. Ten video clips were used in the experiment, mostly collected from the web. Our video dataset contains a variety of motion patterns. Based on the framework described in [5], we categorized each motion pattern into one of the following cases: flash, pulse, swing, spin, turn, shuttle, drift, and thrust. Note that the motion patterns in our dataset are complex because they may result from the appearance variation—intensity change and deformation—, or the perceived path of an object—rotation and translation. For example, our video dataset includes motion patterns from traffic lights and barber's poles. Moreover, the object in our video dataset can either move forwardly, or create an oscillation; the flow field created by the variation can be either continuous or intermittent. Finally, the motion can be induced either by the object motion or the motion of the surrounding background. Figure 2 shows a snapshot and the motion types of each video in the dataset.

The user study involves 153 participants. For each video, we manually created a cinemagraph for each observed motion pattern. Shown together with the automatically generated cinemagraph, each evaluator is required to select the most preferred or enjoyable one from the options.

Figure 2 shows the result of the subjective test. The number on top of a bar represents the number of votes a method receives, and the automatic approach is green-colored. The proposed approach has better satisfactions than others in Fig. 2 (a)-(c), (g)-(i). It is interesting to observe that users do not prefer a particular motion patterns; however, they tend to select the one with a *relative large motions* when the observed motion patterns differ in magnitude. Our approach seeks a sub-volume that has the maximum cumulative flow fields. Therefore, it can successfully localize the spatiotemporal mask that creates a visually interesting result.

However, our approach fails in the case shown in Fig. 2 (e), where the video captures two children playing on a swing set. The automatic approach selects the child who swings higher; however, the localized spatial mask does not contain the whole swing chains. In this case, the moving subject (i.e. the swing) is cut into two pieces. It is somewhat surprising that the unnatural, funny cinemagraph receives 59 votes. The example suggests the need for exploring semantic analysis techniques for better understanding the scene or event under consideration.

Finally, there are a few cases where a tie appears. For example, Fig. 2 (d) contains a flashing traffic light and a mirror that reflects a passing car. The automatic approach selects the mirror. Fig. 2 (f) shows a barber's pole and some passing cars and the barber's pole is selected by the approach. Fig. 2 (j) shows four people dribbling on the court. The automatic approach selects the person in the middle, while a few more votes go to the mask that contains the rightmost two people. In these cases, motion patterns with similar votes in a video are either of the same type, or have a similar motion magnitude. It is not clear how users favor a cinemagraph when motion patterns are alike in terms of their types or magnitudes.

5. CONCLUSION

In this paper, we propose an approach for automatically constructing a cinemagraph from a short video clip. In particular, we view the construction process as an optimization problem that searches a sub-volume in video with the maximum cumulative score based on optical flow. As a data-driven approach, our method is fully unsupervised and does not need to train or maintain a background model.

There are a few directions we may explore to enhance the approach. For example, the current scoring function is based on motion amplitude alone, and may not be robust to scenes with highly dynamic backgrounds and moving cameras. One interesting direction we are pursuing is the investigation of spatiotemporal saliency techniques [4], such as motion consistency [8] and motion contrast [3], which might provide the frame characterization with additional robustness against distracting motion patterns. Another research direction is the integration of semantic analysis techniques into the approach. Suppose we have some understanding of the scene, semantically meaningful motion masks may be better selected by the automatic approach.

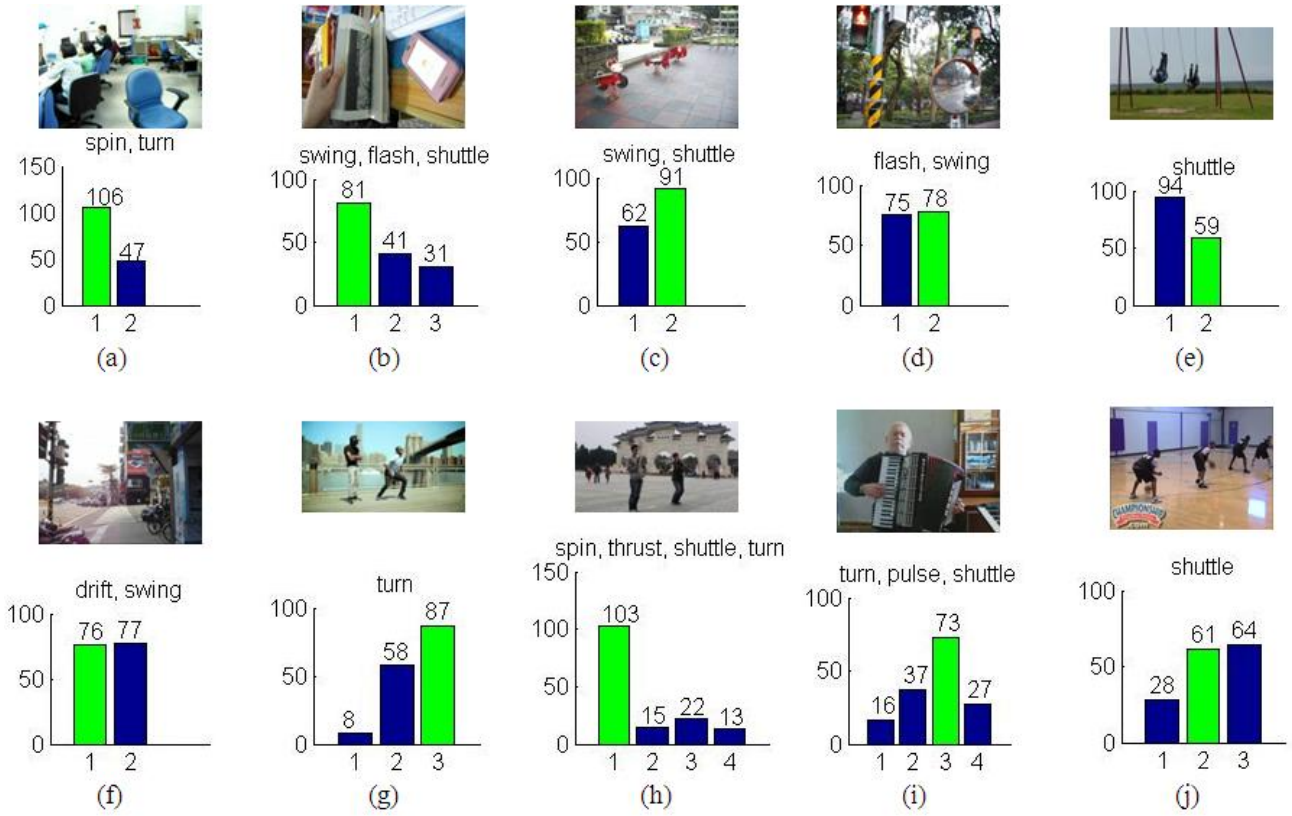

Figure 2. Results of the subjective test. Ten videos were used in the user study. The top panel shows a snapshot of the video, and bottom panel gives the motion types and users' preferences. The number on top of a bar represents the number of votes a method receives, and the automatic approach is green-colored.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Council, Taiwan, under Grants NSC 101-2221-E-003-023.

7. REFERENCES

[1] N. Joshi, S. Metha, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen, "Cliplets: juxtaposing still and dynamic imagery," Technical Report MSR-TR-2012-52, Microsoft Research, 2012.

[2] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: a branch and bound framework for object localization," *IEEE Tran. Pattern Analysis and Machine Intelligence*, 31(12):2129-2342, 2009.

[3] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Tran. Pattern Analysis and Machine Intelligence*, 32(1):171-177, 2010.

[4] H. C. Nothdurft, "The role of features in preattentive vision: comparison of orientation, motion and color cues," *Vision Research*, 33(14), 1937-1958, 1993.

[5] E. Pogalin, A.W.M. Smeulders, and A.H.C. Thean, "Visual quasi-periodicity," *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2008.

[6] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," *ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques* (*SIGGRAPH*), 2000.

[7] J. Tompkin, F. Pece, K. Subr, and J. Kautz, "Towards moment imagery: automatic cinemagraphs," *European Conference on Visual Media Production* (*CVMP*), 2011.

[8] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Tran. Pattern Analysis and Machine Intelligence*, 22(8), 774-780, 2000.

[9] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2009.

[10] http://cinemagraphs.com/

[11] http://kinotopic.com/

[12] http://www.icinegraph.com/