AUTOMATIC CINEMAGRAPHS FROM VIDEO

Ronald Wotzlaw
Li Wei

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation
Wotzlaw, Ronald and Wei, Li, 'AUTOMATIC CINEMAGRAPHS FROM VIDEO', Technical Disclosure Commons, (August 22, 2016)
http://www.tdcommons.org/dpubs_series/237

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
AUTOMATIC CINEMAGRAPHS FROM VIDEO

ABSTRACT

This document describes techniques to generate a cinemagraph from a video automatically and without user input. The techniques enable accurate determination of moving regions of pixels in the video which are suitable for animation, by determining mask regions using a dense optical flow technique. Described techniques enable selection of frame sequences from the video that provide smooth and engaging animations for the cinemagraph, by looking for pixel motion under particular criteria. Techniques enable multiple independent cinemagraph animations to be smoothly provided in a single cinemagraph image.

KEYWORDS

- Cinemagraph animation
- Automatic loop selection
- Synchronized animations
- Optical flow

BACKGROUND

A cinemagraph (e.g., animated GIF) is a still image in which a repeated (looping) animation occurs. For example, one or more objects, a background, or other content in the image can appear to move based on sequentially displaying frames of the animation and replaying the frames continuously. Compared to static images, cinemagraphs can attract more attention and interest from viewers and can create an illusion that the viewer is watching a video.

Creating a cinemagraph from a video can including determining moving regions in an image, where the moving regions are to be animated. Existing limitations for creation of cinemagraphs may include the need for user input to designate such regions, the accuracy of such
regions, restrictions in the extent or type of content to be animated, the amount of shake allowed in the video, etc.

DESCRIPTION

Techniques of the present disclosure allow compelling cinemagraphs to be automatically created from videos, where the cinemagraphs include accurately-determined animated regions and smooth animation loops. Techniques described herein can be implemented on any type of electronic device or computer, server or client device, handheld device, etc., and the cinemagraphs can be displayed on any of a variety of display devices (display screens, projectors, etc.)

Figure 1 illustrates an example process for creating a cinemagraph in which techniques of the present disclosure may be implemented. To create a cinemagraph animation from a video, the video is decoded and stabilized, the regions in the video in which movements occur are determined (mask generation), repeat movements are selected in these regions (loop selection), and cinemagraph animation is encoded into a file (e.g., a GIF file), where the animation smoothly transitions between its start and its end.

An input video is received from which the cinemagraph is determined. The input video should meet several requirements. For example, the video should be reasonably stable, e.g., without large changes in the scene between successive frames of the video, e.g., caused by movement of the capturing camera. In addition, the video should not have large changes in view angle and large changes in zoom level. If such unstable characteristics are present in the video, then a stable portion of the video can be selected, removed from the video, and used as the input video. If there are multiple stable portions, some implementations can provide each stable portion as an output.
portion as a separate input video. Some implementations can use a sequence or "burst" of images instead of a video.

Video decoding is then performed on the input video to provide video frame images for further processing. In some examples, the decoder can be "ffmpeg," and the video frames can be stored in AVFrame format.

Video stabilization can then be performed on the video frames. The video stabilization is applied to compensate for minor instability (e.g., movements) in the frames, such as movement that was introduced into the video by camera shaking during capture of the video. Motions of the video between frames are estimated from the input image sequence, e.g., using optical flow or other techniques, and the frames are modified to reduce the movements. In some implementations, all the video frames are warped to a midpoint frame of the video sequence. The resulting image frames can be cropped in order to remove any black areas at the boundaries of the frames, which may be generated by homography projection. The stabilized image frames are resized back to the original video frame size after cropping.

Mask generation is then performed on the stabilized image frames. The mask generation is performed to separate the scene depicted by the input video into static regions and animated regions. In some implementations, dense optical flow magnitude can be used to estimate the movement of each pixel through the frames of the input video. For example, an average magnitude of movement can be determined for each pixel. Binary thresholding (e.g., using Otsu's algorithm) can be applied to segment the input video scene into either high-magnitude regions (areas that have pixels with larger amounts of movement, e.g., over a threshold) and low-magnitude regions (areas that have pixels with no movement or smaller amounts of movement) in a binary fashion. The high-magnitude regions and low-magnitude regions can define a mask.
The binary thresholding technique may be noisy, and so the mask can be cleaned up by removing small regions (e.g., areas having below a threshold number of contiguous pixels) from the mask so that these small regions are no longer considered high-magnitude regions. In addition, holes inside large high-magnitude regions can be filled, so that the holes are included in the high-magnitude regions. This clean-up process can create a lower number of larger high-magnitude regions in the mask. Each distinct high-magnitude region of the mask can be considered individually for loop selection and cinemagraph creation.

For example, Fig. 2 shows an example frame from an example input video. Fig. 3 shows a resulting mask determined from the video of Fig. 2, where the different colored regions of the mask have been determined based on dense optical flow, binary thresholding, and cleanup of the resulting regions as described above. Some implementations can use additional techniques to find regions in a video, e.g., object recognition techniques or other image recognition techniques, if consent has been obtained from the users associated with the analyzed images or videos.

Loop selection can then be performed. A high-magnitude region of the mask can be selected and analyzed to determine which frames of the input video should be selected to be included in a looping animation for a cinemagraph for that region. In one example, one of the regions shown in Fig. 3 can be selected. The technique can examine the video for qualifying movements that may create a desirable loop for the selected region.

Repeat motion is generated by looping a sub-sequence of the stabilized video frames, focusing on the selected high-magnitude region in the video frames. The loop can be determined by selecting a sequence of frames having a start frame and an end frame.

One type of loop can be a forward-backward loop, which animates the video frames of the sequence in a forward direction from a first frame to a last frame of the sequence, and then
reverses direction and animates the video frames from the last frame to the first frame. This entire animation is then repeated for new loops. Another type of loop can be a forward loop, which animates the video frames in a forward direction from the first frame to the last frame of the sequence, where the last frame is the end frame of the loop. A new loop then jumps back to the first frame. The forward loop may have significant and noticeable visual artifacts if the first frame and last frame are sufficiently different from each other.

In forward-backward loop selection, the loop will always appear continuous regardless of the first and last frames selected for the frame sequence, since the loop ends at the first frame for the next loop. However, the visual quality of animation is higher if the frames near to the first frame of the sequence are relatively static. This is because, at the point when the animation switches from forward to backward, the direction of object motion is reversed. If this reversal occurs when the object is moving fast, the animation may appear unnatural because the object suddenly changes direction without any transition (e.g., slowing of movement before direction reversal). Thus, an object that is moving more slowly can be reversed without causing the unnatural appearance.

The amount of movement in the selected region can be measured, e.g., using dense optical flow magnitude. The sum of optical flow magnitude can be determined within the selected region for the entire length of the input video frames. Local minimum points can be determined as potential first and last frames of a sequence that can be used for the loop.

In one example, each pair of local minimum points can be examined, and a pair of minimum points can be selected as first frame and last frame if particular criteria are met. The criteria can include: (1) the length (e.g., number of frames) between the first and last frames is longer than a particular threshold length, and (2) the magnitude of pixel flow at or near both the
first and last frames is relatively low compared to the pixel flow at frames between the first and last frames. For example, a value \( V(a, b) \) can be determined as follows:

\[
V(a, b) = \sum_{i=a}^{b} M_i \div 0.5(b - a)(M_a + M_b),
\]

where \( a \) and \( b \) are the first and last frames, with \( a < b \). \( M_i \) is the optical flow magnitude at frame \( i \). If \( V(a,b) \) is greater than a threshold, then it is noted that a forward-backward loop is valid between frames \( a \) and \( b \). In another example, the optical flow magnitude for the region in each frame is determined, the magnitudes of optical flow vectors are summed and divided by the area of the region, a filter is applied (e.g., a 1D Savitzky Golay filter) to smooth the data of optical flow magnitudes, and local minima are found and the two smallest are selected for the first and last frames of the sequence used for the loop. The forward-backward loop is created by copying and reversing the sequence and appending the reversed sequence to the end of the sequence, as described above.

For example, Fig. 4 shows a graph of the magnitude of dense optical flow magnitude in a selected region over the frames of an input video. The shown curve indicates several minimum points where the movement of pixels is low. The first two local minimum points of the curve that meet the criteria described above are indicated as arrows, where these arrows can indicate the first and last frames of a sequence used to create a forward-backward loop.

Forward loop selection can alternatively be used for a loop selected from the input video frames. In one example, for a particular region, if no loops are found which qualify for a forward-backward loop as described above, then a forward loop can be determined from the input video frames (alternatively, a forward loop can be checked before a forward-backward loop). In a forward loop, the first frame of the sequence can be the first frame of the video (e.g.,
if no frame is specified by a user), and the last frame is selected by searching all the video frames that meet the forward loop requirements, including having a length (e.g., number of frames) between the first and last frames that is longer than a particular threshold length. Of the frames searched, one frame is selected as the last frame, which is the frame having the most similarity (e.g., minimum 2-norm image distance) to the first frame. In some implementations, if the selected last frame is still very different from the first frame (e.g., the 2-norm distance is greater than a threshold), the selected region can be denoted to have failed to generate a forward loop, and no cinemagraph animation is created for the selected region.

In some implementations, loop blending can also be implemented in the forward loop to enable a smooth transition from a last frame back to a first frame. For example, the last few frames (e.g., last two frames) of the loop can be blended with the first few (e.g., two) frames of the loop so that the transition from last frame to first frame is smoother.

After selection of an appropriate loop from the input video, encoding of the video data can be performed to create an output file providing a cinemagraph for the selected region. In some examples, the output file can be a GIF file (e.g., animated GIF), or other format of file.

For example, the background image (e.g., the first frame of video, or other selected frame) can be blended with the input video frames according to the animation area and the selected loop, to generate the cinemagraph animation frames. The blending can include copying the pixel values of the background image to non-animated areas of each cinemagraph frame, and copying the pixel values of animated regions in video frames of the loop to corresponding frames of the cinemagraph. In some examples, a forward-backward loop can be encoded in the output file by encoding the animated portions of the selected loop sequence of frames from the first frame of the sequence to the last frame of the sequence, and then encoding the frames from the
last frame to the first frame. A forward loop can be created by encoding the animation portions of the loop sequence of frames. If loop blending is used in the forward loop, some frames at the start of the loop can be ignored in the encoding since these frames are included in the blending with the frames at the end of the loop. The resulting output file can be displayed as a looping sequence of cinemagraph frames.

A different region of the mask can similarly be selected, followed by loop selection and cinemagraph encoding as described above. Each region can be processed to create a different cinemagraph output file associated with each region of the mask.

An advantage of using dense optical flow techniques to generate a mask is that high quality masks are determined for animated regions, since movement of all pixels of the video frames are considered. Dense optical flow is also advantageous for selecting the most qualifying loops from a video. For example, dense optical flow can be used with forward-backward loop selection to avoid abrupt switches in motion when the loop transitions from forward to backward. By utilizing the results of dense optical flow for multiple steps in the process, computation time can be reduced.

**Multiple animated regions in a single cinemagraph**

Multiple independent cinemagraph animations can also be included in a single cinemagraph image, e.g., for a single background image scene. For example, each of the regions shown in Fig. 3 can be provided with its own independent looping animation and included in a single cinemagraph animation image and file. In some examples, the multiple regions can each be processed individually as described above, and the results combined. For example, a single background image from the input video can be used as the background for all of the animated
(high-magnitude) regions, and each animated region can include copied pixel values of video frames of the associated loop for that region.

In some cases, the different animations may have different frame rates or different numbers of frames. This can occur based on the motion of pixels in the regions, where different lengths of video sequences were selected for the loops of different regions. This may also occur based on the type of loop used, e.g., some forward-backward loops may require more frames than forward loops. For example, if the animations are started at the same time, these different looping rates may cause one animation to reach its end frame before a different animation reaches its end frame, causing some animations to become static at different times or have other visual artifacts.

The loops of the various regions can be synchronized so that they include the same number of frames, allowing all the animations to depict looping motion continuously and smoothly in a single cinemagraph image.

In one example, a loop associated with a region that has less frames can be provided with additional interpolated frames so that the loop has the same number of frames as a different loop having the highest number of frames. Interpolation techniques can determine additional in-between frames based on the two frames surrounding the interpolated frame. For example, an interpolated position of an object can be an average of its positions in the surrounding frames.

In another example, the least common multiple of the number of frames for each region loop can be determined, and each loop can be repeated such that the number of frames for each region loop is the least common multiple. For example, if there are region loops with 3, 4, and 12 frames, then repeating the first loop 4 times will provide 12 frames, repeating the second loop 3 times will provide 12 frames, and repeating the third loop one time will provide 12 frames. In
general, if $L$ is the least common multiple and $F_i$ is the number of frames of region loop $i$ (that is, $L$ is the least common multiple of $F_1$, $F_2$, $F_n$, assuming $n$ is the total number of region loops), then $L / F_i$ would be the number of repeats needed for frame $i$.

In a least common multiple implementation, the total number of frames in each region loop may grow to a large number quickly. This can be reduced by limiting the number of region loops that are used (e.g., no more than two). In some implementations, the least common multiple can be used approximately and loops can be stretched or reduced in length by interpolating frames to a different number of frames, allowing reduction of the number repeated loops. For example, if region loop R1 has $F_1 = 12$ frames and region loop R2 has $F_2 = 13$ frames, then shortening the second region loop to 12 frames by interpolating the 13 frames to 12 frames reduces the number of frames compared to using $12 \times 13 = 156$ frames.

The described techniques may be implemented in a software program, by computer hardware, or by a combination of software and hardware. In situations in which the techniques described herein may collect or use personal information about users (e.g., user images or videos, user presence status or information, information about a user's social network, location, biometric information, and/or activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the techniques described herein collect, store and/or use user personal information only upon receiving explicit authorization from the relevant users to do so. For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or
more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more such control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. In some examples, a user’s identity may be treated so that no personally identifiable information can be determined.

FIGURES

![Diagram](image.png)

FIG. 1

![Image](image.png)

FIG. 2
FIG. 3

FIG. 4