Video Textures Exploiting Symmetric Movements

William Van Haevre † and Frank Van Reeth ‡

Hasselt University, Expertise Centre for Digital Media
Wetenschapspark 2, B-3590 Diepenbeek (Belgium)

Abstract

In this paper, an extension to the traditional method for creating video textures is presented. By exploiting the symmetric properties of the frames in a given video sequence, a new endless video stream or video loop is generated by reversing video playback when appropriate. We aim at a larger set of possible input video's for video texture creation, including captured motions containing no smooth transitions (eg. turbulent plant movements). To achieve this, good turn points, instead of transitions are extracted from the video data, after which a simple algorithm synthesizes a new video sequence while optimally exploiting the original frame data. Visual artifacts caused by changed lighting conditions are reduced significantly.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: video I.3.3 [Picture/Image Generation]: display algorithms

1. Introduction

Images are often used to add more visual features to traditional information exchange. To extend this with dynamic or time dependent properties of the communicated information, video has become an important medium. Due to the restriction in data transport and storage for most computer related applications, the use of large video sequences is unacceptable and requires an alternative approach.

In addition, a more data driven approach for the creation of animations is demanded, due to the success of video based rendering. Based on existing video material, new footage is created while reusing or extending available data.

In response, video textures [SSSE00] were introduced. Video textures are short video sequences, often repeatable over time without noticeable artifacts. They are synthesized from existing video data and contain the most important dynamic and visual features of an object. Several methods for the creation of video textures have been proposed, but the applicability of the technique is still restricted to a small set of possible input videos. The most important reason for this, is the specific requirement of the original data to satisfy certain requirements to be acceptable as input (ex. a sufficient amount of similar frames).

A first algorithm to create video textures was presented by Schödl et al. [SSSE00]. Using the L_2 norm, all frames of a given video sequence are compared. Sufficiently matching frames act as transition points from which the original video playback can deviate to reach other parts of the video data. They extended their work to the use of video sprites within constrained animations and machine learning for video based rendering [SE01, SE02], resulting in more interactive animations. In [KSEB03], video texture synthesis is performed using a graph cut technique.

An alternative method was presented by Campbell [CDGT02], who synthesized video textures using PCA dimensionality reduction and used the auto-regressive process to move through the lower dimensional manifold to which the videoframes were transformed. In a similar way, the applicability of video textures was extended by de Juan [dJB04] towards cartoon textures.

To increase the usability of video textures to other fields of interest, an extension to Shödl's original algorithm is required, allowing video data with no repetitive occurrences of similar frames, to be acceptable as input for video texture creation. Observation of the movements of many objects in

† e-mail: william.vanhaevre@uhasselt.be
‡ e-mail: frank.vanreeth@uhasselt.be

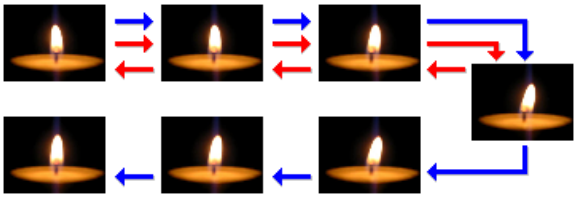


Figure 1: *reversing video playback at a symmetric frame position: (blue) original frame sequence, (red) an almost identical sequence, reusing the previous frames.*

our environment motivates the exploitation of their symmetric properties for video synthesis.

2. Video textures of symmetric movements

An important property of a video texture is the possibility to repeat the generated sequence over time without noticeable, disturbing repetitions of the same movements. This requires the displayed motions to be made of actions that remain natural looking when repeated continuously. Such movements are often focused around one ore more fixed positions to which the object returns from time to time.

Standard video textures exploit these reference locations by means of transitions. Whenever a similar position of an object or set of different objects is detected on different videoframes, a transition can be made, resulting in a rearrangement of the frames of the original video sequence.

In the case of symmetric motions, the movement away from a reference position and the returning movement are (sometimes exact) opposites. This is not always true for the complete recorded motion, but most of the time it holds for several frames in the neighbourhood of the point where the motion turns. This implies that while playing the video sequence, a large number of frames can be reused when an extreme position in the captured motion is reached, resulting in a reversed playback of that action. (figure 1).

The use of video textures can thus be extended to motions containing no smooth traditional transitions. A typical subset of this category includes the movements of plants. It is very unlikely to find exactly the same configuration of the plant structure on different locations in a video, due to their complex structure, turbulent motions and the incoherency of the movements of the individual plant parts. On the other hand, plants tend to move very symmetrically. Their branching structure forces them to return to their original state whenever an external force is applied to them, by means of successive decreasing symmetric movements.

This paper describes a method for the creation of video textures, by exploiting the symmetric properties of short sequences of videoframes, and can be divided into three steps:

1. locate highly symmetric frames within the video data (sections 3.1 and 3.2);
2. derive probabilities from this data, indicating the chance for each frame to reverse video playback (section 3.3);
3. use these probabilities to generate an infinite videostream or video loop from the original data (section 4).

3. Video analysis

As mentioned before, we are interested in finding the specific positions in a video sequence where playback can be reversed, without noticeable visual inconsistencies. With traditional video textures [SSSE00], a good transition is found when the succeeding frames before and after the transition almost completely match.

3.1. Locating highly symmetric frame locations

In the case of symmetric movements, a similar procedure can be used. To locate highly symmetric frame locations, we compare the frames located over the same distance before and after the test position. If they match sufficiently, according to a user-specified threshold, a symmetric position is marked. We use the L_2-norm (eq. 1) to grade the equivalence of the corresponding frames. Other metrics can be used as well.

$$D_{i,j} = \|I_i - I_j\|_2 \tag{1}$$

The symmetric property of a location within a video sequence can sometimes be very local. Just a few frames should be included in the test. Also, the influence of nearby frames is often more important than the similarity of more distant frames. To accomplish this, a weighted sum of the L_2 distances is used (eq. 2), by means of binomial coëfficients.

$$Sym_i = \sum_{r=1}^{n} \binom{2(n-s)}{n-r} \cdot D_{i-r,i+r} \qquad \text{with } s < \frac{n}{2} \tag{2}$$

In this equation n equals the amount of frames included in the test, s influences the smoothness of the binomial coëfficients and $D_{i-r,i+r}$ is the L_2 distance between the frames located r positions before and after testframe i. Higher s values relate to higher weights for nearby frames and lesser influence for more distant frames.

To make these values more meaningful and easier to handle, we map then onto the unit interval $[0..1]$ (eq. 3). A resulting value of 0 indicates absence of symmetry, 1 equivalents to a frame location with high local symmetry relative to the other frames. Sym_{max} holds the highest Sym value computed during the previous calculations (figure 2, gray).

$$Sym_i = 1 - \frac{Sym_i}{Sym_{max}} \tag{3}$$

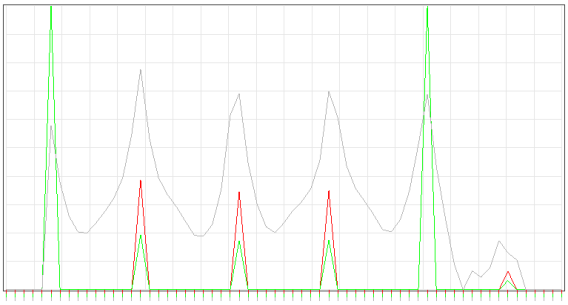


Figure 2: *a graphical representation of the clock sequence (section 5) with 63 frames: (gray) calculated sym-values, indicating the summetric property of each frame; (red) maximum derived probabilities, representing the chance for a frame to act as a turn position; (green) initial probabilities, used for video texture generation.*

3.2. Reducing the candidates

At this stage, we estimated for each frame in a videosequence how well it mirrors the actions taken before and after its position. Next, we need to limit the set of frames, according to two rules: (i) only the frames with the highest *Sym* values are required; (ii) the symmetric points need to be positioned far enough from each other, to prevent rapid repetitions of the same short action. To achieve these goals, we keep a subset of the original frames containing only those frames that have a *Sym* value which is a local maximum within a user specified range r (eq. 4). The other values are reduced to 0. The frame positions closest to the beginning and the end of the video, with a sufficient *sym*-value get a new value of 1.0 to indicate that video playback should always be reversed at these positions, to prevent reaching a dead end.

$$Sym_i = \begin{cases} Sym_i & \text{if } Sym_i = max\ (Sym_{i-r}, ..., Sym_{i+r}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

3.3. Extracting probabilities

The calculated *sym*-values can be transformed into probabilities, indicating the possibility for a frameposition to act as a turnpoint. A fully symmetric frame (*sym*-value of 1.0) should have a 50% chance to reverse video playback and equal probability to proceed with the original video sequence. For all frame positions we state (eq. 5):

$$Pmax_i = \frac{Sym_i}{2} \tag{5}$$

which indicates that a lower *sym*-value corresponds to a smaller probability to reverse playback and a higher probability to continue (figure 2, red). The outmost turn positions receive maximum probability: $Pmax_a = Pmax_z = 1.0$.

4. Video synthesis

A straightforward algorithm to synthesize new video from the original sequence is simply to start video playback at a random position and use the precalculated probabilities to decide if a reverse is going to be made at a certain frame position. The problem with this approach, is that most of the time the same turns will be taken, that is at the positions with the highest probabilities. Also, the same part of the original video sequence will be reused several times, leaving a portion of the available frames unused.

4.1. Extended algorithm

To exploit the full range of available frames, we start our video synthesis with initial turn probabilities equal to 50% of the maximum allowed probabilities (eq. 6). This increases the chance to continue with the original video playback at the turn positions, making it possible to travel further within the frame space. The values of the outmost turns always remain unchanged: $P_a = P_z = 1$ (figure 2, green).

$$P_i = \frac{Pmax_i}{2} \tag{6}$$

We try to encourage a changing behaviour of the turnpoints, by increasing or decreasing the probability of a point whenever it is reached. When a turn is taken, the probability to turn again at this point is lowered with a fixed or relative amount, increasing the possibility to continue the next time without a turn. When a turn is not taken, the opposite behaviour is provoked, by increasing the probability (eq. 7).

$$P_i = \begin{cases} max\ (P_i - inc, 0) & \text{if turned at frame } i \\ min\ (P_i + inc, Pmax_i) & \text{otherwise} \end{cases} \tag{7}$$

The resulting rearrangement of the videoframes can be done in real-time and results in a completely random videostream.

5. Results

The presented retrieval of the symmetric properties of the video frames and the algorithm for video synthesis are both limited to an $O(n)$ complexity, resulting in a very short calculation time. The examples described below were processed within a few seconds, on a Pentium 4, 3.2 GHz processor with 1GB of memory.

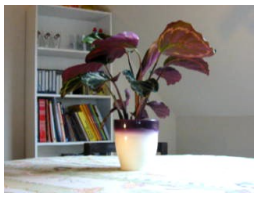
Traditional video textures [SSSE00] suffer from small visual inconsistencies due to slightly changed environmental conditions before and after the transition was made, in terms of lighting. Because no transitions are taken with the new algorithm, and any two frames in the synthesized video match with already succeeding or preceding frames in the original video, no visual lighting artifacts are visible.

Video textures are only applicable under certain conditions. The captured actions must contain repetitive motions

and be isolated from other actions in the background. We extend this range of applications to a larger set, including symmetric motions without actual repetitive actions, but still require the motion to be significantly isolated from other moving parts.

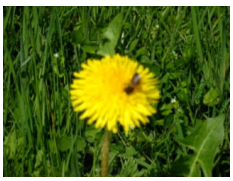
Clock This first input video displays a clock with a swinging pendulum, earlier also used as an example for the original method. The forward and backward swinging motions are almost identical, resulting in a typical high symmetric turn point. The result after applying our algorithm demonstrates how the technique manages to produce a new sequence with a quality that minimally matches the results of the traditional method. The synthesized motions are very realistic and the video contains no visual artifacts.

Plant This example shows a generated video sequence of a small plant subjected to wind. Even though the movements of the individual plantparts are not isolated and the overall motion is turbulent, the video synthesis reveals no disturbing or unrealistic motions. Due to the resulting absence of good transitions, the traditional method fails to create a good result.



Flame The observation of a candle flame reveals another typical example of symmetric movements. This is a very good candidate for a hybrid method, in which the use of transitions is mixed with occasional turns of the video playback to increase randomness in the resulting video. Using only our new method already gives a very satisfying result, with no unrealistic motions or disturbing visual drawbacks.

Nature A lot of symmetric motions can be observed in our environment. A flower, for example, is moved by the wind. Meanwhile, an insect may be crawling on its surface using small successive displacements. Both movements have a high symmetric appearance. Even in combination, the resulting synthesized video sequence shows no immediate unnatural behaviour. Because the influence of subsequent small camera movements is negligible on the overall symmetric properties of the video content, this example also demonstrates the applicability of our method on input data captured without a fixed camera position.



6. Conclusions

Video textures based on symmetric movements are an extention to the traditional method [SSSE00] for creating endless video output from a given video sequence. The presented algorithms exploit the similarity of the surrounding frames at specific video positions, to reverse video playback and extend the original data to a new synthesized version, without visual artifacts or unrealistic content.

A technique is described to determine the symmetric properties of each frame in a given video sequence. From this information probabilities are derived, indicating how suitable the frames are to act as a turn point. An algorithm is presented to synthesize a new endless video stream or video loop, maximally exploiting the available video frames.

In future work we would like to focus on a hybrid method in which the traditional technique is extended with our algorithm, to increase the randomness of the generated video sequence. The combination of random transitions and reverses in the video playback should allow for sufficient reorganisations of the original video sequence, without visual inconsistencies.

Acknowledgements

We gratefully express our gratitude to the European Fund for Regional Development (ERDF), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT), which are kindly funding part of the research reported in this paper.

References

[CDGT02] CAMPBELL N. W., DALTON C., GIBSON D., THOMAS. B.: Practical generation of video textures using the auto-regressive process. *British Machive Vision Conference* (sep 2002), 434–443.

[dJB04] DE JUAN C., BODENHEIMER B.: Cartoon textures. In *Eurographics Symposium on Computer Animation* (Grenoble, France, aug 2004), ACM Press.

[KSEB03] KWATRA V., SCHÖDL A., ESSA I., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. In *Siggraph 2003, Computer Graphics Proceedings* (San Diego, California, jul 2003), ACM Press.

[SE01] SCHÖDL A., ESSA I.: Machine learning for video-based rendering. *Advances in Neural Information Processing Systems 13* (jul 2001), 1002–1008.

[SE02] SCHÖDL A., ESSA I.: Controlled animation of video sprites. In *ACM Symposium on Computer Animation on ACM Siggraph 2002* (San Antonio, Texas, jul 2002), ACM Press.

[SSSE00] SCHÖDL A., SZELISKI R., SALESIN D. H., ESSA I.: Video textures. In *Siggraph 2000, Computer Graphics Proceedings* (New Orleans, Louisiana, jul 2000), Akeley K., (Ed.), ACM Press, pp. 489–498.