FAST VIDEO ARTISTIC TRANSFER VIA MOTION COMPENSATION

Pedro Garcia Freitas[1] and Mylène C.Q. Farias[2]

[1]Department of Computer Science, University of Brasília, Brasília, Brazil
[2]Department of Electrical Engineering, University of Brasília, Brasília, Brazil

ABSTRACT

Techniques for conversion of natural video scenes into drawing-style videos are frequently used to produce animated movies. In the past, the conversion was manually performed, what demanded a lot of time and a high production cost. Recently, with the advancement of computer vision techniques and the development of new deep learning algorithms, 'drawing' can be automatically performed. Nevertheless, current `drawing' algorithms are computationally expensive and require a high processing time. In this letter, we present a simple, but effective `drawing' algorithm that is capable of reducing the processing time.

KEYWORDS

Computer-guided Rotoscopy, Style Transfer, Motion Compensation, Video Processing

1. INTRODUCTION

Rotoscoping is a technique [1] that converts 'natural' video frames into cartoon or artistic animated frames. In the early days, rotoscoping was performed manually, demanding an enormous amount of time and a high number of artists, what naturally increased the cost. In recent years, algorithms that perform an automatic conversion of filmed scenes into artistic scenes have been developed to decrease costs. More recently, machine learning algorithms have been used to perform the conversion of natural images into artistic images. For example, Gatys et al. [2] developed an algorithm based on neural networks, which captures the style of paintings (or art works) and transfers it to natural images. Although effective, their approach uses a deep convolutional network to mimic the artistic style and transfer it to the natural content, what is computationally expensive. To increase computational performance, Johnson et al. [3] proposed an algorithm that uses perceptual loss functions based on high-level features from pre-trained neural networks.

Since Gatys' [2] and Johnson's [3] methods were developed for images, they produce flickering and discontinuities distortions when applied to videos. To reduce these distortions, Ruder et al. [4] developed an algorithm that preserves the smooth frame transitions by using temporal constraints that penalize discontinuities between two consecutive frames. Unfortunately, although this algorithm is able to reduce distortions, it requires a large processing time. In this letter, we propose an algorithm that uses motion compensation to eliminate frame discontinuities in converted artistic videos. The algorithm eliminates the unnecessary processing of redundant information in consecutive frames, what reduces the overall processing time. Visual results are good and comparable to previous works [2, 3, 4].

DOI : 10.5121/ijma.2017.9202

2. FAST VIDEO ARTISTIC STYLE TRANSFER

The proposed algorithm has four stages: (1) temporal segmentation, (2) intra-frame style transfer, (3) motion estimation, and (4) inter-frame style transfer. The original and artistic videos are represented by x^t and y^t, respectively.

In the temporal segmentation stage, video frames are classified as intra-frames (x^t_a) or inter-frames (x^t_e). The classification avoids introducing errors due to scene changes (shot boundaries) in the motion compensation stage. This classification is performed by computing the normalized histogram (**H**) of the pixel distributions of two consecutive frames, i.e. **H**(x^t) and **H**(x^{t+1}) . To compare the histograms, we use the following metric to compute their distances:

$$d=\sqrt{\Delta(x^{t+1},x^t)\cdot\Delta(x^t,x^{t+1})}, \quad (1)$$

where

$$\Delta(x^i,x^j)=H(x^i)-H(x^j) \quad (2)$$

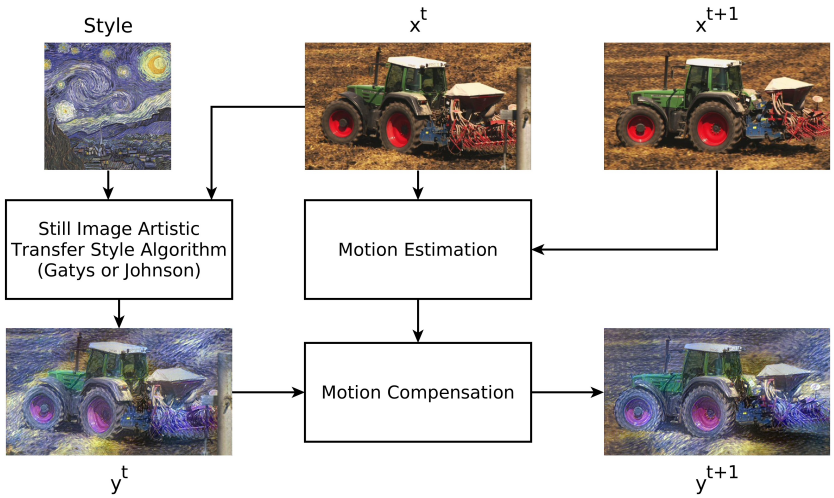

Figure 1.Block diagram of the inter-frame style transfer stage.

If d is higher than a threshold τ, the frame x^{t+1} is classified as intra (x_a^{t+1}). Otherwise, it is classified as inter (x_e^{t+1}).

In the intra-frame style transfer stage, a style transfer algorithm (e.g. Gaty's [2] or Johnson's [3]) is applied to convert the original natural intra-frames into an artistic style frames. More specifically, for each x_a^t reference frame, we compute the corresponding artistic style frame y_a^t . Since the style transfer algorithms are computationally complex, representing a bottleneck for time performance, only the intra-frames y_a^t are processed at this stage.

In the motion estimation stage, the original frames, x^t , are divided into 'macroblocks' of n×n pixels. To estimate the relative motion of each macroblock, we search for macroblocks with similar content in the next video frame x^{t+1} . To estimate the content similarity, we use the Mean Absolute Difference (MAD). The search is restricted to an area p×p surrounding the macroblock position in x^{t+1} . After motion vectors for all macroblocks of x^t are computed, we obtain a motion estimation map for the whole frame. We can reconstruct x^{t+1} from x^t using the motion vectors and the estimated macroblocks.

Figure 1 shows a block diagram of the inter-frame style transfer stage. In this diagram, $y^t = y_a^t$ is the previous stylized intra-frame, which was generated from the previous natural intra-frame

Figure 2: Frames of tested video frames and their stylized versions using Ruder'smethod.

$x^t=x_a^t$ using a still-image transfer style algorithm. The stylized inter-frame $y^{t+1}=y_e^{t+1}$ is obtained using only y^t and the motion estimation map computed from the original video frames x^t and x^{t+1}. Therefore, the process of generating stylized inter-frames does not require a style transfer algorithm.

If two consecutive frames are classified as inter-frames, we first convert x^t into y^t , by performing an inter-frame style transfer using y^{t-1} . Then, the process is repeated to generate y^{t+1} , i.e. an inter-frame style transfer is performed using y^t. For several consecutive inter-frames, the motion estimation is performed always using the previous converted frame. In other words, the proposed method is based on the re-utilization of previous converted frames, in order to avoid using computationally costly style transfer algorithms. Therefore, the higher the number of frames classified as inter-frames, the faster the overall conversion process will be.

3. EXPERIMENTAL SETUP

Experiments were performed using a laptop with an Intel i7-4700mq processor, 32GB of RAM, and a Nvidia GeForce GTX 770m running Linux CentOS 7. The temporal segmentation and motion estimation steps of the proposed method were implemented in Matlab R2015a. For comparison, we used Python implementations of the style transfer methods provided by their authors in [2, 3, 4].

Figure 3: Frame 245 of sequence 'VIDEO 1' and the their stylized versions.

To test the proposed algorithm, we used three videos of the LIVE Quality Video Database [5]. These videos are all high-quality 10 seconds videos, with 768×432 at 50 or 25 frames per second (fps). Also, three copyright-free artistic images were used as style. The first row of Figure 2 shows thumbnails of the natural videos, while the first column of this figure shows images with three different artistic styles. The other images in the figure correspond to the stylized video frames. For example, the image in the 3rd column of the 2nd row corresponds to the stylized version of a frame of video 2, using the artistic style 1.

4. EXPERIMENTAL RESULTS

Figure 3 shows results of applying the algorithms proposed by Gatys [2], Johnson [3], and Ruder [4], alongside with the corresponding proposed method adaptations, on 'Video 1' (non-distorted version of MC sequence in LIVE database). In this figure, the results correspond to the 245-th frame of 'Video 1', classified as an inter- frame. Fig. 3-(a) shows the original frame, while Figs. 3-(b) shows the result obtained using Ruder's algorithm [4], which is designed for videos and, therefore, cannot be adapted to be used with the proposed methodology. Figs. 3-(c) and (e) show the results obtained with Gatys' and Johnson's algorithms. Figs. 3-(d) and (f) correspond to the results obtained adapting Gatys' and Johnson's algorithms for videos using the proposed methodology, which are identified as 'Gatys+MC' and 'Johnson+MC', respectively. Notice that this adaptation produces results that are similar to the corresponding original methods.

Table 1 shows the average runtime (in minutes) for all tested artistic transfer algorithms. Since Gatys' [2] and Johnson's [3] methods are designed for images, the average runtime corresponds to the sum of the average runtime necessary to process each video frame. From Table 1, we can notice that Gatys' [2] and Ruder's [4] algorithms perform similarly, what is surprising since Ruder's algorithm pre-processes the frames to generate an optical flow map. On the other hand, as expected, Johnson's algorithm is significantly faster than Gatys' and Ruder's algorithms.

However, the processing runtime of proposed approaches ('Gatys+MC' and 'Johnson+MC') is much smaller, leading to a better performance. On average, the speedup is about 4.94 for Gatys' method and 3.43 for Johnson's method.

Table 1. Average runtime (in minutes) for style transfer algorithms.

Video	Style	Method				
		Gatys	Johnson	Ruder	Gatys+MC	Johnson+MC
Video 1	Style 1	3,081.293	37.391	4,274.627	627.316	12.448
Video 1	Style 2	4,822.257	35.934	4,353.927	977.445	10.608
Video 1	Style 3	4,963.427	38.289	3,290.406	1,006.049	11.171
Video 2	Style 1	5,069.397	39.486	3,920.698	1,036.859	10.757
Video 2	Style 2	3,117.162	37.030	4,002.351	638.510	10.164
Video 2	Style 3	4,064.445	39.011	4,512.369	831.939	10.751
Video 3	Style 1	2,367.307	18.463	2,280.406	465.955	5.582
Video 3	Style 2	2,717.732	17.719	2,501.406	534.448	5.245
Video 3	Style 3	2,821.691	19.388	2,917.321	554.906	5.655
Average		3,669.412	31.412	3,561.501	741.491	9.153

It is worth mentioning that, in our implementation, the average time for computing motion vectors is 0.006 minutes (0.361 seconds) per frame. For this reason, the overhead caused by the motion estimation stage is almost negligible, specially when compared to the processing time required by a style transfer algorithm. After computing the overall runtime of the proposed methodology (including vector estimation + style transfer), we observe that the time necessary to convert a video with the 'Gatys+MC' approach is 1.779 minutes per frame, while time necessary for converting a video with the 'Johnson+MC' approach is 0.021 minutes per frame. Without the proposed methodology, required times are 8.806, 0.075, and 8.547 minutes per frame for Gatys', Johnson's, and Ruder's algorithms, respectively.

5. DISCUSSION

The proposed method was implemented in an unoptimized Matlab program, which contains several code fragments. However, even with this limited implementation, the proposed approach showed a better time performance than regular style transfer algorithms. Therefore, further improvements can be made to the proposed methodology to make it more attractive to practical multimedia applications. First, more efficient motion estimation algorithms can be used [6]. Second, by using specific hardware resources, better implementations of the algorithm can be developed. For example, Matlab can be replaced by a compiled language (e.g. C++). Finally, the use of parallel programming techniques can further reduce the computation time of the proposed methodology [7, 8].

6. CONCLUSIONS

In this paper, we converted natural videos into artistic style videos by combining motion estimation techniques with frame style transfer methods. Experimental results show that the proposed approach reduces the amount of processing time required to generate an artistic style video, while maintaining comparable visual results. Future works include a parallel implementation of the motion estimation algorithm using GPUs. We also plan to investigate the effect of spatial and temporal resolution on style transfer algorithms. If the resolution does not have an effect on the quality of converted videos, it can be reduced to decrease the overall runtime performance [9].

ACKNOWLEDGEMENTS

This work was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and by the University of Brasília.

REFERENCES

[1] Sturman, D. J. (1994). A brief history of motion capture for computer character animation. SIGGRAPH 94, Character Motion Systems, Course notes, 1.
[2] Gatys LA, Ecker AS, Bethge M. A neural algorithm of artistic style. ArXiv preprint arXiv:1508.06576. 2015 Aug 26.
[3] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. arXiv preprint arXiv:1603.08155. 2016 Mar 27.
[4] Ruder M, Dosovitskiy A, Brox T. Artistic style transfer for videos. arXiv preprint arXiv:1604.08610. 2016 Apr 28.
[5] Seshadrinathan, K., Soundararajan, R., Bovik, A. C., and Cormack, L. K. (2010). Study of subjective and objective quality assessment of video. IEEE transactions on image processing, 19(6), 1427-1441.
[6] Barjatya A. Block matching algorithms for motion estimation. IEEE Transactions Evolution Computation. 2004 Apr;8(3):225-39.
[7] Garcia-Rodriguez, J., Orts-Escolano, S., Angelopoulou, A., Psarrou, A., Azorin-Lopez, J., Garcia-Chamizo, J. M. (2016). Real time motion estimation using a neural architecture implemented on GPUs. Journal of Real-Time Image Processing, 11(4), 731-749.
[8] Kao, H. C., Wang, I. C., Lee, C. R., Lo, C. W., Kang, H. P. (2016, April). Accelerating HEVC Motion Estimation Using GPU. In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on (pp. 255-258). IEEE.
[9] Schulter, S., Leistner, C., Bischof, H. (2015). Fast and accurate image upscaling with super-resolution forests. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3791-3799).

AUTHORS

Pedro Garcia Freitas, received his B.Sc. degree in Computational Physics in 2010 and his M.Sc. in Computer Science in 2013, both from the University of Braslia (UnB), Brazil. He is currently pursuing a Ph.D. in Computer Science at the University of Brasilia (UnB), Brazil. His interests are parallel computing models, machine learning, and video quality assessment, signal processing, and programming languages.

Mylène C.Q. Farias, received her B.Sc. degree in electrical engineering from Universidade Federal de Pernambuco (UFPE), Brazil, in 1995 and her M.Sc. degree in electrical engineering from the Universidade Estadual de Campinas (UNICAMP), Brazil, in 1998. She received her Ph.D. in electrical and computer engineering from the University of California Santa Barbara, USA, in 2004 for work in no-reference video quality metrics. Dr. Farias has worked as a research engineer at CPqD (Brazil) in video quality assessment and validation of video quality metrics. She has also worked as for Philips Research Laboratories (The Netherlands) in video quality assessment of sharpness algorithms and for Intel Corporation (Phoenix, USA) developing no-reference video quality metrics. She is currently an assistant professor at the Department of Electrical Engineering of the Universidade de Brasilia (UnB), Brazil. Her current interests include video quality metrics, video processing, multimedia, watermarking, and information theory. Dr. Farias is a member of IEEE and IEEE Signal Processing Society.