Animated Non-photorealistic Rendering in Multiple Styles

Ting-Yen Chen and Reinhard Klette

Department of Computer Science
The University of Auckland, New Zealand

Abstract. This paper presents an algorithm and its applications for artistic rendering of recorded video data following ideas of non-photorealistic rendering. The proposed algorithm does not only work on a variety of artistic rendering styles for static photography but it can also be applied to the creation of artistic videos. Cartoon-like and comic-like styles are the two artistic styles considered in this paper. For creating successfully an artistic video, three key challenges are addressed: temporal (or colour) consistency, stylistic flexibility, and scalability. Our work on addressing these challenges started with collecting samples of image and video data. Based on experimental results, we designed a method for video-based non-photorealistic rendering from those input data, either in cartoon-like or in comics-like style. The paper demonstrates the benefit of the designed video-based rendering framework by comparing its results with results obtained from existing Android apps.

Keywords: Artistic rendering, non-photorealistic rendering, animations, cartoon-like style, comic-like style.

1 Introduction

Recently there have been impressive advances in computer animation technology. Techniques of *non-photorealistic rendering* (NPR) have been developed for interpreting photographs in some artistic style. In this paper we combine both, animation technology and NPR for producing cartoon-like and comic-like style videos. NPR in computer graphics focuses on achieving a variety of expressive styles for digital art, such as oil-painting, water-painting, or cartoon-like artwork. Graphic editing software already includes many features in this direction. The goal of developers in this field is to help artists to create their own artwork, not to create automatically data which may compete with creations by artists.

At present, animation in art is known as artistic video. NPR has been extended from static images to animations. Animated NPR aims at producing video in any possible artistic style automatically, to be finalized then by interactive editing. Animated NPR is not only artwork in a series of static images; it also needs to ensure consistency between consecutive frames in the resulting video. According to J. Wang et al., there are three criteria for evaluating a successful *video tooning* (i.e. cartoon) algorithm [3]:

F. Huang and A. Sugimoto (Eds.): PSIVT 2013 Workshops, LNCS 8334, pp. 12–23, 2014.

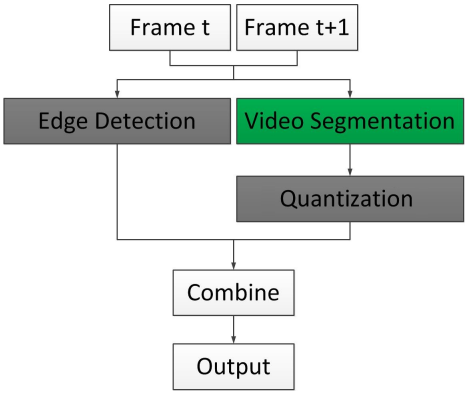

Fig. 1. Block diagram of the proposed algorithm. The green and gray boxes show techniques which need to be designed. Video segmentation, followed by a quantization method, addresses temporal consistency. The output image combines results from this quantization process and an edge detection method.

- The final image sequence or video should maintain temporal consistency to avoid significant jumps in frame transitions.
- The content of image or video data should be abstracted in a way as to provide a higher-level semantic representation.
- An artist should be able to have control over the style of the result.

Our creation of cartoon-like animation follows those criteria. The paper proposes an algorithm for animation in cartoon-like style.

The Proposed Algorithm. Fig. 1 shows a block diagram of our algorithm. The algorithm takes as input only the recorded sequence and generates the rendered output automatically. The process has three procedural components. First, in each frame we find edges of objects in the scene in order to achieve a *hand-drawing style*. Video segmentation aims at achieving colour consistency and a diffusion effect. The third procedure applies colour quantization to the results of the video segmentation process. This procedure aims at higher visual quality. The combination of these three procedures can be easily adapted to produce cartoon-like video automatically. The key feature of our approach is that both colour consistency and diffusion effects are achieved within the procedure of video segmentation. This allows us to improve time performance.

Comic-like style applies typically a hand-drawing style in black and white. Our chosen style, the XDoG technique, is easy to implement and provides good results.

Outline of the Paper. The paper focuses on artistic rendering of recorded video data. Section 2 briefly discusses the challenges for producing an image or a video in artistic style. Section 3 describes the video segmentation step in our algorithm for achieving colour consistency and a diffusion effect. Section 4

describes the edge detection step for producing a hand-drawing style, and also introduces a technique for creating a comic-like style. Section 5 describes the quantization step for achieving a higher quality of the cartoon effect. Section 6 reports on obtained results and evaluates rendered video data; it also compares results obtained by an Android app (called *Cartoon Camera*) with those obtained by our algorithm. Section 7 concludes.

2 Challenges

Challenges can be related to NPR of static images, to an animation subject, or to both combined. Before starting the discussion of possible approaches, we discuss below a few potential problems which occur if we would proceed just in some straightforward way.

Binarization converts a grayscale image into a black and white image, typically by using one threshold globally. An example is shown in Figure 2 (*top-left*).

Posterization maps a grayscale image into an image using only a few colour values. An example is shown in Fig. 2 (*top-right*). We conclude that NPR of a static image cannot be done simply by applying a standard posterization procedure.

Related to the three criteria suggested in [3], the following three challenges are considered in this paper for animated NPR:

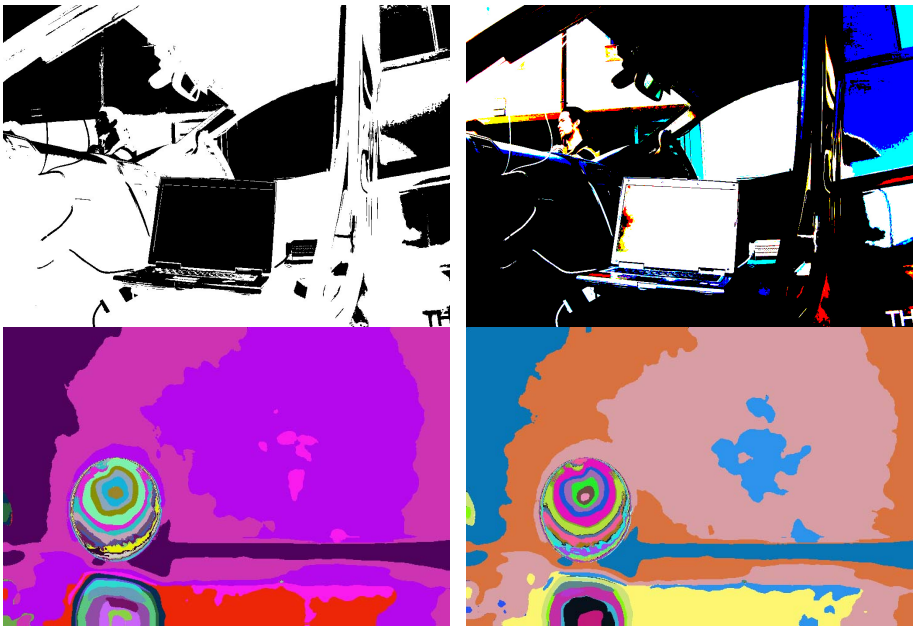

Fig. 2. *Top*: A static grayscale image binarized, and also shown as in posterised form using two levels of intensities in each of the three colour channels; neither result is satisfactory. *Bottom*: Examples of two subsequent video frames individually rendered after segmentation, illustrating missing colour consistency.

The first challenge is related to ensuring temporal coherence. Image segmentation approaches applied to each frame independently produce unstable results as we cannot expect that there are only minor frame-to-frame changes; these changes cannot be expressed as a continuous function [1]. The images in Fig. 2 (*bottom*) illustrate that a standard image segmentation process did not achieve colour consistency for video data. Segmented regions in a dynamic scene need to be tracked over time.

The second challenge is achieving stylistic flexibility. Users should be able to experiment with different visual styles related to their aesthetic requirements or individual preferences.

The last challenge is scalability. Given the large amount of pixels in a video, video segmentation tends to be slow. For example, high-resolution images have high visual quality, but their use is inefficient for video segmentation. In conclusion, a parallel implementation is proposed to achieve real-time for scalability in NPR.

3 Video Segmentation

Mean-Shift Video Segmentation. For achieving temporal coherence and diffusion effects, we follow the original algorithm proposed by Y. Zhang [6]. This algorithm is based on region matching between consecutive frames combined with dynamically generating new regions by the use of a mean-shift filtering procedure. The mean-shift procedure for video segmentation is defined by the following steps:

1. Define a fixed kernel window for each data point.
2. Compute a new center point as the mean of the data within the kernel window.
3. Shift the kernel window to a new center point.
4. Repeat till convergence.
5. Apply a matching algorithm.

For a given image sequence, f_t represents the frame at time t in the sequence. Let $\mathbf{x}_s(t) = [x(t), y(t)]^\top$ be the pixel coordinates (i.e. the spatial features) in frame f_t, and $\mathbf{x}_r(t) = f(\mathbf{x}_d(t)) = [R(\mathbf{x}_s(t)), G(\mathbf{x}_s(t)), B(\mathbf{x}_s(t))]^\top$ be the range features (i.e. colour values) at pixel in frame f_t. A feature vector $\mathbf{x}(t) = [\mathbf{x}_s(t), \mathbf{x}_r(t)]^\top$ combines domain and range features. A mean-shift vector is defined as follows:

$$m(\mathbf{x}(t+1), \mathbf{x}_i(t)) = \mathbf{x}'(\mathbf{x}(t+1), \mathbf{x}_i(t)) - \mathbf{x}(t+1) \tag{1}$$

$$\mathbf{x}'(\mathbf{x}(t+1), \mathbf{x}_i(t)) = \frac{\sum_{i=1}^{N} \mathbf{x}_i(t) \cdot g_s(\| \frac{\mathbf{x}_s(t+1) - \mathbf{x}_{s\,i}(t)}{h_s} \|^2) \cdot g_r(\| \frac{\mathbf{x}_r(t+1) - \mathbf{x}_{r\,i}(t)}{h_r} \|^2)}{\sum_{i=1}^{N} g_s(\| \frac{\mathbf{x}_s(t+1) - \mathbf{x}_{s\,i}(t)}{h_s} \|^2) \cdot g_r(\| \frac{\mathbf{x}_r(t+1) - \mathbf{x}_{r\,i}(t)}{h_r} \|^2)} \tag{2}$$

where $g(\cdot)$ is the *Epanechnikov* kernel window and $x'(\cdot)$ is the new center point; h_s and h_r employed the size of kernel window (kernel bandwidth). Note that *Gaussian kernel* can also be used for computing the mean shift.

Let the corresponding feature vector $\mathbf{x}(t+1)$ be in the kernel center, and the set of feature vectors $\mathbf{x}_i(t)$, where $i = 1, \ldots, N$, be in the previous frame within the bandwidth h_d from f_t. For each pixel $\mathbf{x}_d(t+1)$ at f_{t+1} and the pixels within a kernel window at f_t, apply the mean-shift filter to find the mode of the region by calculating the mean-shift vector from Eq. (1) repeatedly until it converges, where $\|m(\cdot)\| < \varepsilon$ (pre-defined threshold). ε is set to avoid high computational cost due to excessive iterations. In this paper, we set ε to 100. Since a new center of mean-shift is generated from each iteration, a new set of features $\mathbf{x}_i(t)$ (from frame f_t) is used in the next iteration.

Local Mode Matching. Once this iterative algorithm converges, the converged mode $\mathbf{x}(t)$ from f_t is computed. The process of *local mode matching* is sought between a pixel in the current frame and a variable set of features in the previous frame. If a similar mode is found, then the colour pixel of the resultant image is replaced by the colour pixel of the convergence point. Otherwise, a mean-shift filter is applied to the current frame $t+1$ to obtain the filtered pixels as the new region or object.

A feature vector $\mathbf{x}(t+1)$ is used to find the matching region from the previous frame if the convergence point $\mathbf{z}(t)$ has a small range distance to $\mathbf{x}(t+1)$ (i.e. $\|f_{t+1}(\mathbf{x}(t+1)) - f_t(\mathbf{z}(t))\| \leq h_r$). Otherwise, the pixel of convergence belongs to a new region or object that is introduced in the current frame.

4 Edge Detection

The difference of Gaussians (DoG) operator is a common technique used in computer vision for edge detection. DoG involves the subtraction of two differently blurred versions of an original image. The blurred images are obtained by convolving the original gray-scale images with Gaussian kernels with different standard deviations. The DoG kernel is used as an approximation to a scale-normalized Laplacian of Gaussians (LoG) kernel.

LoG aims at detecting zero-crossing in second-order derivatives of an image. This essentially captures the rate of intensity changes. In an ideal continuous case, this detection would capture local maxima in the gradient. The basic principle of LoG is to compute an image by applying different Gaussian filters with different scale factors for standard deviations (or sigma), also called the blur radius. The LoG has two main effects. First, the noise is reduced and the differentiation is regularized. Second, the kernel size is restricted, which means the range of possible scales, at which edges can occur, is reduced. The LoG filter was proposed in [2]. The LoG operator is commonly approximated by the DoG as follows:

$$F_{DoG}(\mathbf{x}) = G_\sigma(\mathbf{x}) - G_{k\sigma}(\mathbf{x}) \tag{3}$$

where k is the scale factor and $\mathbf{x}$ is a pixel location.

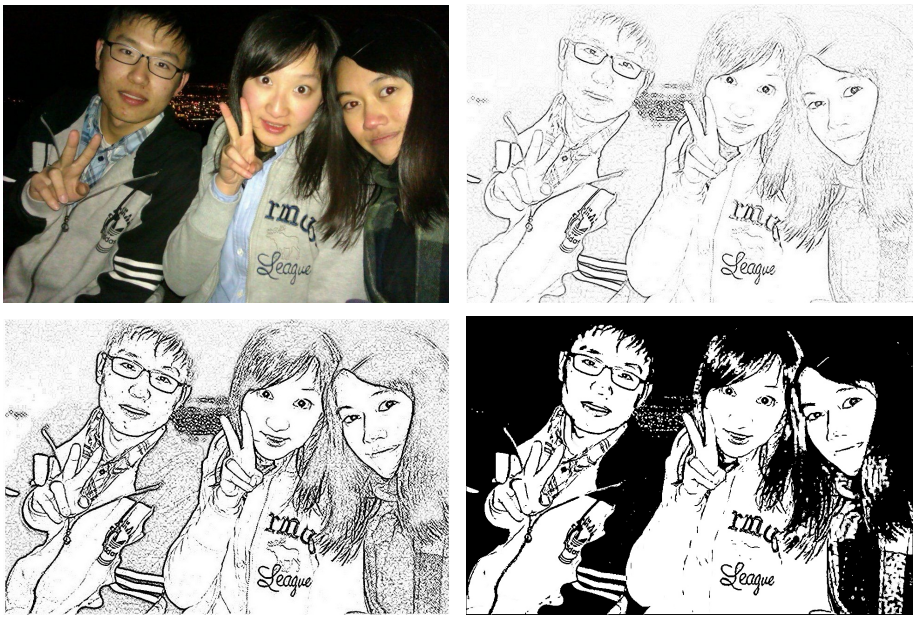

Fig. 3. *Top*: Original image and edge map of reduced sharpness. *Bottom*: Sharper edge map, and XDoG filtered image.

DoG. The DoG has been extended by H. Winnemöller [5] to produce a filter capable of generating a hand-drawing style. This DoG allows users to control the sharpness of the edge. The approach is based on Eq. (3) and defined as follows:

$$F_{DoG}(\mathbf{x}, \sigma, k, \tau) = G_{\sigma}(\mathbf{x}) - \tau \cdot G_{k\sigma}(\mathbf{x}) \tag{4}$$

$$G(\mathbf{x}, \sigma, k) = \frac{1}{W(\mathbf{x})} \sum_{i=0}^{N} f(\mathbf{x}) \cdot e^{-\frac{1}{2}\left(\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right)} \tag{5}$$

$$E(\mathbf{x}, \sigma, k, \tau, \varphi) = \begin{cases} 1 & \text{if } (F_{DoG}(\mathbf{x}, \sigma, k, \tau)) > 0 \\ 1 + \tanh\left(\varphi \cdot F_{DoG}(\mathbf{x}, \sigma, k, \tau)\right) & \text{otherwise} \end{cases} \tag{6}$$

where Eq. (6) is used to determine the edges from DoG functions with parameter τ which controls the amount of center-surround differences required for cell activation, and φ controls the sharpness of the activation falloff [5]. Value σ determines the spatial scale for edge detection (see examples in Fig. 3). For small values of τ, less noise is detected, but real edges become less visible. This threshold parameter τ is used to determine the sensitivity of the edge detector. Eq. (5) is a Gaussian blur function. Each pixel and its adjacent pixels within a radius of N do the convolution with a Gaussian kernel. $W(\mathbf{x})$ is the total weight of the Gaussian kernel at pixel location $\mathbf{x}$.

XDoG. The XDoG filter can be used for edge detection with comic-like style rendering based on the DoG algorithm. This filter is capable of generating NPR

styles such as comic-like, hatching, charcoal and pastel. The recently proposed XDoG filter by H. Winnemöller [4] further refines the thresholding processing by introducing an additional parameter ε into the DoG filter and is defined by

$$E(\mathbf{x}, \sigma, k, \tau, \varphi) = \begin{cases} 1 & \text{if } (F_{DoG}(\mathbf{x}, \sigma, k, \tau)) > \varepsilon \\ 1 + \tanh\left(\varphi \cdot (F_{DoG}(\mathbf{x}, \sigma, k, \tau) - \varepsilon)\right) & \text{otherwise} \end{cases} \tag{7}$$

According to [4], this filter is still quite sensitive to noise. To some extent, the threshold ε can be used to reduce sensitivity, but a more effective way is to apply the diffusion filter in one or two iterations before applying an XDoG filter. This XDoG also allows users to control ε to produce their own artworks. An example is shown in Fig. 3 (*bottom-right*).

5 Colour Quantization

The cartoon effect is an important artistic style for NPR images. Colour quantization [5] is a method that reduces the number of distinct colours used in an image. This quantization effect is used with the intention that the new image is still visually similar to the original image. The colour quantization step on abstracted image can produce cartoon-like or paint-like effects, and it is defined as follows [5]:

$$Q(x, q, \varphi) = q_{nearest} + \frac{\Delta q}{2} \tanh\left(\varphi \cdot (f(x) - q_{nearest})\right) \tag{8}$$

where Δq is the width (or size) of a bin, $q_{nearest}$ is the bin boundary closest to the intensity of x, and a parameter φ controls the sharpness of the transition across the boundary. The term '*bin*' is defined for the colour range used to represent the image.

To minimize jarring transitions [5], φ is a function of the luminance gradient in the abstracted image. Hard bin boundaries only appear where the luminance

Fig. 4. A cartoon-like image combines a diffusion effect and a quantization effect

gradient is high. In low gradient regions, bin boundaries are spread out over a larger area. Thus, the algorithm gives the user a trade-off between increased quantization artifacts and reduced colour variation by defining a target sharpness range $[\Lambda_\varphi, \Omega_\varphi]$ and a gradient range $[\Lambda_\delta, \Omega_\delta]$. Then, the calculated gradients are clamped to $[\Lambda_\delta, \Omega_\delta]$. Then, φ is generated by mapping linearly to $[\Lambda_\varphi, \Omega_\varphi]$.

For a standard quantization, H. Winnemöller [5] mentioned that an arbitrarily small luminance change can push a value to a different bin, thus causing a large output change for a small input change. On the other hand, a soft quantization provides an effect which makes a luminance change less noticeable.

The main idea of this colour quantization process is to use a sharpness value and a user controlled variable Δq to distort the luminance channel of the image. This sharpness value is derived from the magnitude of the image gradient and scaled by a user controlled parameter φ in order to create the variable sharpness. An example is shown in Fig. 4.

6 Experiments and Evaluations

We demonstrate the workflow of our proposed approach with video data. The video data used for the experiments has a static background an moving object(s). This section also presents a performance comparison of visual quality, efficiency and effectiveness with *Cartoon Camera* which is a publicly available mobile application.

Used Data for Evaluation and Comparison. We illustrate three video data, recorded by a handheld SONY HD camera (7.1 mega pixels). The video data is outdoor or indoor. Names of datasets are *graduation*, *laboratory*, and *home*, consisting of 200 frames each. We run the proposed algorithm on those video data and perform a visual evaluation. We are interested in colour consistency and visual quality of static or non-static objects in consecutive frames.

Cartoon-Like Animation Experiment. When using the DoG algorithm of [5], we decided for light-weighted edge detection with a threshold value of 0.989 and a thickness value of 1. This threshold value cuts off unnecessary noise in an image. The threshold value is predefined because light conditions vary for each frame. By testing the threshold in any condition on each frame, the range of suitable values is between 0.98 and 0.99. The thickness value presents the sharpness of the edge. Choose a suitable thickness value is in $[0, 1]$.

Video segmentation plays an important role in the temporal direction process. This is used for keeping constant the matched regions of colours between frames. Up to this stage, images are light-weight cartoon-like style due to the diffusion effect. This *light-weight* style means that slightly changed colour and edges in the image are constant but this effect gives a low-level cartoon effect (see *top-right* in Fig. 5). Optionally, applying a quantization process may be needed to achieve a high-level cartoon effect (see *bottom-right* in Fig. 5).

Evaluation. Fig. 5 shows results obtained by applying our proposed algorithm for *home* video data. The top-left image shows the application of a mean-shift

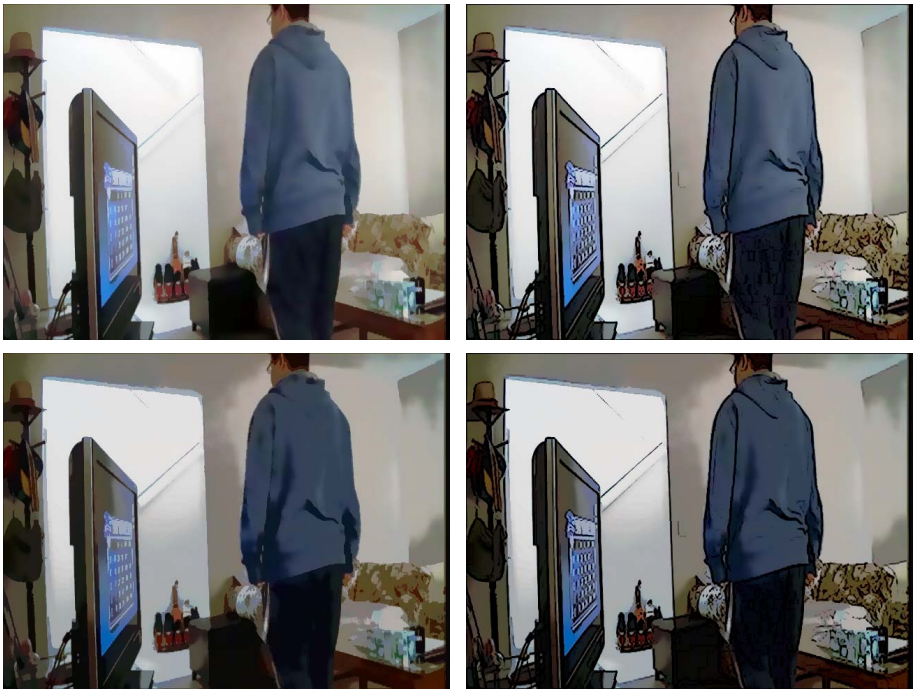

Fig. 5. Frames from the *home* sequence. *Top*: Diffusion effect, and diffusion effect with a hand-drawing style. *Bottom*: Diffusion and quantization effect, and diffusion and quantization effect with hand-drawing style. The radius for the diffusion process is 25 pixels. The quantization parameters are set with a sharpness range between 2.8 and 9.6.

filter with the temporal direction known by video segmentation. This gives an interesting effect with a variety of kernel sizes. With influence parameters at this level, the diffusion effect ensures that colours of the background and foreground remain constant between frames. This cartoon effect depends on kernel size, while colour distortions make this effect more artistic.

Another advantage of the cartoon effect is the use of a quantization technique. The bottom-left image shows the combination of diffusion and quantization techniques. In standard quantization, an arbitrarily small luminance change can push a value to a different mode of colour. This causes that the colour in the region becomes brighter and the shadow of the object becomes more visible than before. Also, the visual quality is much better than with the use of diffusion. Soft quantization can also achieve temporal coherence. In soft quantization, the change is spread over a larger area, making it less noticeable. The combination successfully blends the two techniques together to create a higher quality.

The two images on the right show the variance of results for the hand-drawing style by using the edge detection technique. The clarity of the object is important for achieving an artistic style. Based on the idea of the cartoon, reorganization is more important than the style of painting. Without the edge of the object,

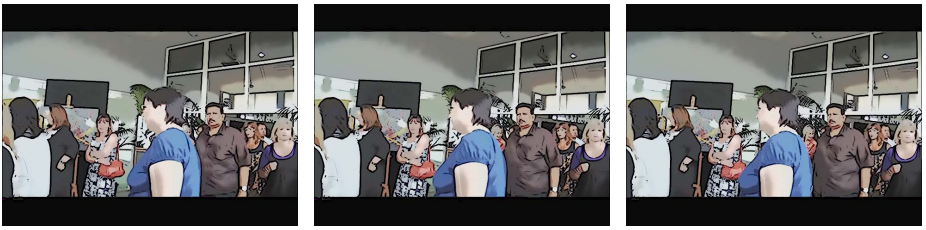

Fig. 6. Frames from *graduation* video. People walk from the right to the left. *Left* to *right*: Three subsequent frames of a cartoon-like video; quantization parameters are set with a sharpness range between 2.8 and 9.6.

it is hard to recognize the contents of the scene. With the influence parameters of the DoG algorithm, all edges in the scene can be adjusted by the user. This means that the visual quality can also be adjusted by the user. However, the techniques used in the bottom-right image produce a better cartoon effect.

Fig. 6 shows the results obtained by applying our algorithm to *graduation* video data. The dataset is taken on a sunny day which means the effect of light is reasonably high. However, the visualization of each moving object is still clear along the temporal direction, although some objects in the scene have poor visual quality. For example, a human face has the same colour as the hair. In this case, some contents in the scene are destroyed by the light.

Comparison with Cartoon Camera. *Cartoon Camera* (CC) is a free Android camera app. It enables the camera to create a photograph in cartoon-like style. A comparison between CC and our algorithm in the experiments is carried out quantitatively using defined performance measures (e.g. numbers of segments, or for colour consistency), or qualitative evaluations. The evaluation will focus on three categories: *effectiveness*, *efficiency* and *quality*.

Effectiveness measurement evaluates the processed output in either "well performance" or "poor performance". Fig. 7 shows a result from our algorithm and a result from CC. Each segment area for both obtained results is well segmented

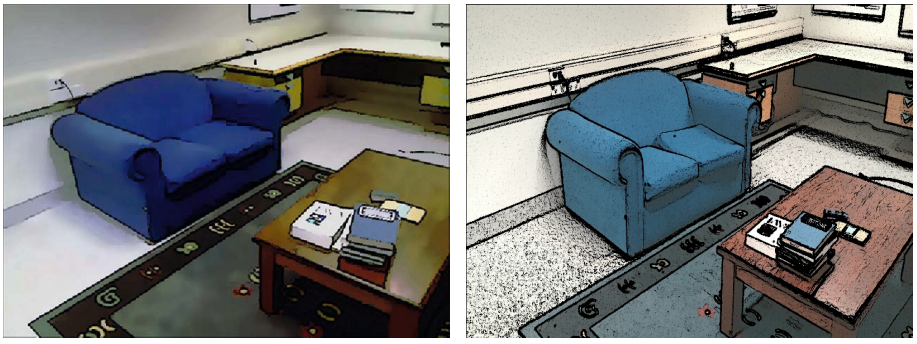

Fig. 7. *Left* to *right*: Resultant images using our algorithm or CC. Our result gives a better abstract effect then CC. The used image is from the *laboratory* video.

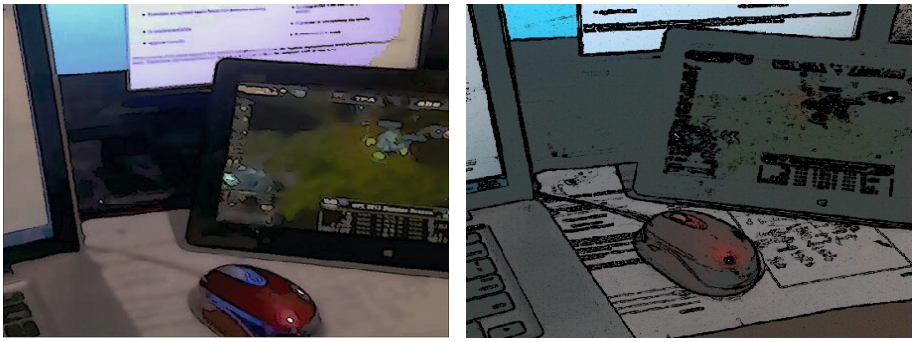

Fig. 8. *Left* to *right* : Resultant images using our algorithm or CC. Our result gives more abstract effect than CC. CC does not give a good performance under this condition.

in the expected colour. This illustrates that both techniques work well under normal light conditions. Colour consistency for both techniques is also achieved. In some cases, CC does not provide a satisfactory performance; see Fig. 8. Compared to our result, the objects in the CC image are less recognizable and with unexpected colour. This illustrates that CC has an issue when dealing with the light from monitors. This means that this light may destroy some properties of objects such as colours and edges. Our algorithm produces more abstract effects and gives a better quality of art than CC.

To evaluate the efficiency we need to determine the cost for producing the required output. The output with the lower cost of the process will have higher efficiency. The cost can be defined as time consumed for processing. CC gives a better real-time performance for producing a cartoon-like image or video. This illustrates that CC has better time efficiency. CC produces a light-weight technique to achieve the real-time performance and high efficiency. Our algorithm is based on video-based NPR and it takes approximate 10 seconds to produce a rendered image based on the records in Table 1.

The last criteria is based on the quality of an application. Users should be able to experiment with different artistic styles that meet their aesthetic requirements or personal preferences. CC provides a user interface which allows the user to change colour mode and level of edges. Our application allows the user to change the level of diffusion, quantization and edges. Both techniques achieve good quality in applications. This is helpful for supporting artists in achieving different possible styles.

Table 1. Performance of the four NPR methods for a frame from *laboratory* video data with size 1920×1088

Method	**VS**	**Quantization**	**DoG**	**XDoG**
Frames per second	7.332	0.348	2.518	2.626

7 Conclusions

The paper focused on the creation of animations ensuring temporal coherence between generated frames. Lack of temporal consistency may affect the smoothness of the video. For this reason, temporal consistency must be considered for creating animations. Our proposed algorithm can be used for different situations of indoor or outdoor video data. The proposed algorithm can create cartoon-like animations automatically. The mean-shift video segmentation procedure in our algorithm reduces the sampled feature space and generates the mode for each region which achieves colour consistency between frames. Finally, by using edge detection and quantization, we resample the computed modes of the original dataset to obtain a final cartoon-like result.

Our chosen algorithm is significantly more accurate than CC while sacrificing the performance. However, the algorithm is especially useful for any kind of images with any resolution and long-time video sequences. Moreover, the style can be adjusted by the user.

Issues with our proposed algorithm: First, face features are a critical issue. From an artist's point of view, any feature within the scene can affect the story in the scene. Our experiments produced reasonable results for facial features. In some cases, under strong light conditions, the features are not easy to detect. This issue might be solved by using edge consistency in the temporal direction. For example, the edge should be kept within the scene if the edge appears more than four times in consecutive frames. For facial features, facial recognition can optimize the performance.

A second issue is scalability. Our goal is to generate a cartoon-like animation and our video-based algorithm is not yet robust. Therefore, future work can involve real-time processing for supporting a camera in the real word. Our project is run on a CPU system. Using a GPU system or other parallel implementations can accelerate the performance of our algorithms without losing accuracy.

References

1. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: Proc. Computer Vision and Pattern Recognition, pp. 2141–2148 (2010)
2. Marr, D., Hildreth, R.C.: Theory of edge detection. In: Proc. R. Soc. Lond. Series B, Biological Sciences, vol. 207, pp. 187–217 (1980)
3. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. Association for Computing Machinery Transactions on Graphics 23(3), 574–583 (2004)
4. Winnemöller, H.: XDoG: Advanced image stylization with eXtended difference-of-Gaussians. In: Proc. NPAR, pp. V147–V155 (2011)
5. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. In: Proc. Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques, pp. 1221–1226 (2006)
6. Zhang, Y.: Advances in image and video segmentation. IGI Global (2006)