Coherency Algorithms for Pen and Ink converted Real Time Video

Fahad Zafar*
University of Maryland Baltimore County

Figure 1: *(a) Original (b) NPR Version*

Abstract

Non-Photo Realistic Rendering (NPR) algorithms remove the realistic rendering features from an existing source. Particularly for the Pen and Ink styles it is correct to say that they add a pattern of strokes that remove the mechanical layout of a machine image, converting them into a picture most similar to humanly art. When dealing with videos, and converting them into some existing NPR version, the strokes added percframe add an incoherency to the image stream. The focus of this paper is to put forward algorithms that are independant of the NPR implementation and can convert existing real life videos into a coherent image streams.

1 Introduction

The images sampled from the video are independantly converted to a $Sketch$ NPR image. Since the algorithm in its original state contains no coherency handling for the images, the $wriggly$ affect is introduced in the image stream. The affect occurs due to the rotation and placement of numerous strokes by the conversion algorithm. One caution to be taken care of when solving this problem is to take care of the fine line between the limits of coherency and incoherency. If you tend to apply a smoothing filter and entirely smoothen the images, the NPR affect is lost. On the other hand, if you add too less of a coherency, the pixels appear incohereny and no noticable coherency is added in the NPR image stack.

2 Related Work

Most NPR techniques convert models existing in $WorldSpace$ to an NPR implementaion, Others work with lightings in order to change the realistic look and feel of the object into a $cartoonish$ display. $Gooch$ uses a technique to extract an NPR view from an existing real life video source. It adds coherency to the images using different stages of conversion and works for colored NPR implementations. Another approach tried to extact the stroke orientation from the image in order to add a synchronous feel to the images being converted, but also alters the conversion algorithm to maintain such consistency.

*e-mail: fahadzafar@umbc.edu

Figure 2: *Erosion in Space and Time*

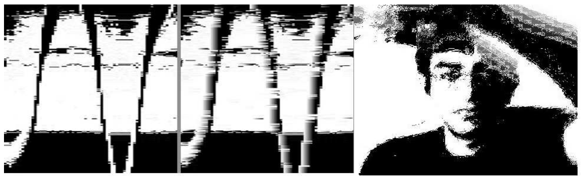

Figure 3: *Time Trail Affect, erosion in time domain only*

3 Erosion

Applying erosion in time and space domain would firstly allow pixels to last for a longer duration in the image stack. Secondly, isolated pixels will most probably combine with some existing pixel in space domain. Even if they dont, the new pixels added will remove the affect of tiny spots appearing and disappearing from the screen ($Fig.2$).

The introduction of new pixels in order to produce coherency removes a lot of detail from the stream. Even though the spread factor is small, the pixles loose the amount of detail they are adding.

Another approach is to only erode in the time domain. Since no pixels in the space domain are added, the amount of detail presented by pixels is least affected. Working this scenario out does remove the problem of loosing detail, however it introduces a new one.

A time trail affect appears for the moving objects ($groupofpixles$) is left behind in the scene as the video progresses ($Fig.3$).

Figure 4: *Smoothing filter applied across time axis*

Figure 5: *Filter applciation to Grey Scale Stroke based NPR*

float t = .099
float r = t - .07

$$\begin{bmatrix} r & 0 & 0 & 0 & 0 & 0 & 0 & r \\ 0 & r & 0 & 0 & 0 & 0 & r & 0 \\ 0 & 0 & t & 0 & 0 & t & 0 & 0 \\ r & 0 & 0 & .1 & .25 & 0 & 0 & r \\ r & 0 & 0 & .25 & .5 & 0 & 0 & r \\ 0 & 0 & t & 0 & 0 & t & 0 & 0 \\ 0 & r & 0 & 0 & 0 & 0 & r & 0 \\ r & 0 & 0 & 0 & 0 & 0 & 0 & r \end{bmatrix}$$

Figure 6: *The Specialized Convolution Filter*

3.1 Implementation Outline

The coherency algorithm works in 3 steps. First we convert the sampled video into an image stream with respect to time. Then apply the coherency technique, and lastly convert the images back in to the space dimension.This step is actually the inverse of Step-1. The images are in reverse order since their perspective has shifted. A simple flip and frame reversal will bring back the images into the original order.

3.2 The Coherency Step

Apply erosion to the pixels in these images along the time dimension using a spread factor k. k represents the number of new pixels to be added along the timeline in the images. Access the image from left to right in order to find white pixels following black ones. When a white pixel is found at location i, add pixels till location $i + k$. The color variable C stores the color to which the ith location color will transition to. Initially it is white to add a fade out affect along the image. Reduce the color along the pixel locations from i to $i + k$ using a linear gray scale dimming.

If a collision occurs during the k pixel addition i-e Some x pixel is encountered that is not white, a new color transition is added. Now C is the color of the xth pixel and a new color $Color_m$, where

$$Color_m = (Color_i + Color_x)/2 \qquad (1)$$

is added to pixel location $i + d/2$ (Where d = number of pixels between i and x). Write the pixels to transition from $Color_i$ into $Color_m$ and then to C causing a Fade in - Fade out affect.

Sometimes $d = 0$ or less than $k/2$. If this case occurs very often then the coherency will be lost. In such a case, resample the image stack from the video at twice the number of images as used previously.

4 The Convolution Approach

Applying a filter that simply smoothens the images does not produce desirable results. The purpose of the entire algorithm is to maintain the same affect on the images while adding some consistency to the frame to frame shift. A normal smoothing filter simply smoothens the image ($Fig.4$).

It turns out that the best results achieved are by a convolution solution compared to the other approaches mentioned. The results are quiet pleasing as they provide sufficient coherency and maintain the originality of the images($Fig.5$). A specialized filter is used for this conversion ($Fig.6$). An $8*8$ filter with specific properties to maintain the NPR look and feel and at the same time making the scene as coherent as possible.

The filter shown in ($Fig.6$) is one that adds coherency to the images, while keeping the orientation of the strokes intact along all directions. A relationship of two weight variables in the filter tend to increase or decrease the blend of strokes in order to produce coherency.

5 Conclusion

The best results are achieved w the convolution filter is applied. It keeps the essence of the original NPR video intact. A noticable feature was found during the research that applying an algorithm in time domain or pixel domain has exactly the same affect. The only difference is that when worknig in time domain, a short trail affects persists in the video. A convolution approach seems to be disirable and potentially pleasing. The variables in the filters can be changed according to the requirements. The results thus depend on the manual input of those variables.

6 References

Strothotte, Schlechtweg ”Non-Photorealistic Computer Graphics (Modeling, Rendering and Animation)”

Gooch and Gooch ”Non-Photorealistic Rendering”

Peter Litwinowicz Apple Computer, Inc. ”Processing Images and Video for an Impressionist Effect

Holger Winnemoller, Sven C. Olsen, Bruce Gooch ”Real-Time Video Abstraction”

Georges Winkenbach and David Salesin, ”Computer-Generated Pen-and-Ink Illustration,” SIGGRAPH 1994, pp. 91 - 98.

Barbara J. Meier, ”Painterly Rendering for Animation,” SIGGRAPH 1996, pp. 477 - 484.

Michael P. Salisbury, Michael T. Wong, John F. Hughes and David

H. Salesin, ”Orientable Textures for Image-Based Pen-and-Ink Illustration,” SIGGRAPH 1997, pp. 401 - 406.

Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen, ”A Non-Photorealistic Lighting Model for Automatic Technical Illustration,” SIGGRAPH 1998, pp. 447 - 452.

Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden and John F. Hughes, ”Art-Based Rendering of Fur, Grass, and Trees,” SIGGRAPH 1999, pp. 433 - 438.

Aaron Hertzmann and Denis Zorin, ”Illustrating Smooth Surfaces,” SIGGRAPH 2000, pp. 517 - 526.

Doug DeCarlo, Anthony Santella, ”Stylization and abstraction of photographs,” ACM Transactions on Graphics, v21n3 (Proceedings of SIGGRAPH 2002), pp. 769 - 776.

Holger Winnemller, Sven C. Olsen, and Bruce Gooch, ”Real-time video abstraction,” ACM Transactions on Graphics, v25n3 (Proceedings of SIGGRAPH 2006), pp. 1221 - 1226.