Adaptive Abstraction of 3D Scenes in Real-Time

Niall Redmond[1] and John Dingliana[1]

[1]Graphics, Vision and Visualization Group, Trinity College Dublin

Abstract

We present an approach for creating abstracted renderings of 3D scenes in real-time. We create painterly renderings with edge detail using varying levels of abstraction dependent on regions of interest within a scene. Image space techniques are used to make the system real-time and as non-invasive as possible. This approach can also use object space information to segment the scene into visually important objects and unimportant data and background. Interactive frame rates are achieved by using graphics hardware to perform the computations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Colour, shading, shadowing and texture

1. Introduction

Non-photorealistic rendering (NPR) has become a popular technique for visualizing complex data and for improving the aesthetics and effectiveness of computer images. NPR can create simple renderings of complicated objects to make scenes faster and simpler to comprehend. NPR might also help to simplify a complex scene for display on a smaller device where the original scene would be impossible to show.

One of the main motivations of this paper is to use NPR abstraction to remove less important detail in a 3D scene and increase the saliency of relevant detail. A semi-automated system for doing this for single images is presented in [SD02] using eye-tracking data to guide the NP rendering. The drawback of this system is that use is limited to single images.

Current NPR techniques for 3D scenes are usually reliant on using the 3D scene data to generate NP renderings. Among these are particle systems for generating brushstrokes, 3D edge detectors and various types of NP shaders. However many of these approaches are tailored for data modeled specifically for that NPR algorithm or for simple scenes. A motivation of this paper is to provide a more generic system which does not depend on a specific data type or scene size and can be easily bound to a non-specific application. An example of a non-invasive NPR system was put forward in [MG01] where the OpenGL pipeline is intercepted and altered. However this system produced results containing a high level of flicker when NP textures were applied upon a scene.

We present a system which produces different levels of abstraction in a scene. The system we propose adapts to any change in focus region to smoothly move between high and low levels of abstraction. This adaption can be made more meaningful by using the 3D object data to identify which object is at a focus point. This enables the focus objects to be rendered with increased prominency in the scene.

To demonstrate our system we use Virtual Dublin [HO03], a virtual urban environment which runs in real-time. NPR abstraction in an application such as this could be used in tourism for highlighting popular sights or historical landmarks from the rest of the scene. Adaptive abstraction could also be used in many interactive environments such as games or route finding applications. However, for use in applications such as these, it is necessary for the system to be able to run in real-time. We achieve this by using image space NPR methods implemented on GPU pixel shaders. Past research shows that image space NPR techniques can be used effectively in real-time.

2. Related Work

There have been many publications investigating automatic painterly NPR techniques but the majority of these are applicable to images and video but not to 3D animation in

Figure 1: *Example of various renderings of the same part of a scene. (a) A normal rendering with no abstraction (b) The edge image retrieved from Difference of Gaussian edge detection (c) Output image from the Kuwahara filter using 3x3 sampled regions (d) The edge image overlaid on the Kuwahara image.*

real-time. Expensive techniques exist for painterly rendering of single images such as the one proposed in [Her98]. This type of technique, which simulates each brushstroke, is relatively popular but impossible to run at interactive rates. Meier [Mei96] was among the first to present an automated system for rendering 3D animation in a painterly style. Meier uses a 3D particle system as the basis for her technique. However this method is slow due to the processing time for particles in large scenes which contain many models, such as a city scene.

A real-time method for image abstraction is presented in [WOG06]. Here image based techniques are used on video and single images to produce painterly abstraction in real-time. Each frame goes through a series of image processing steps. First the image is put through an anisotropic diffusion filter, followed by edge detection and colour quantization. In earlier work an image based technique for creating painterly renderings is put forward in [MKK76]. Here an edge preserving blur filter is used to create a painterly effect upon images. This filter, known as the Kuwahara filter, is very similar to the anisotropic bilateral diffusion filter described in [WOG06].

Edges play an important part in various types of non-photorealistic rendering. Adding edges to the scene can create clearly distinct regions and bold edges can help to direct focus to a particular object or area. Much work has gone into finding edges from an image as accurately and efficiently as possible. Object space algorithms can find edges very effectively but are very expensive and dependant on the complexity of the scene. Image space edge detectors, despite not being as precise as object space edge detectors, can achieve good results and run at interactive rates.

The Canny edge detector [Can86] is known as the optimal image space edge detector due to its accurate and noise-free results but the complexity associated with it makes it impossible to run in real-time. The Marr-Hildreth algorithm [MH80] for detecting edges uses the Laplacian of Gaussian function on an image and then looks for the zero crossings in the filtered result. The Difference of Gaussian method is a fast approximation of the Marr-Hildreth algorithm.

Santella and DeCarlo [SD02] use eye tracking data to guide automatic painterly renderings of images. They used an eye-tracker to retrieve the meaningful parts of an image then abstraction is performed with small, detailed strokes created for visually important data and large brush strokes created for less important parts of the image. While the renderings are effective they can only be used for single images and they suffer from unwanted background detail near important foreground objects.

Our goal for this paper is to create images which use visually important points and objects in the scene as a guide for image abstraction. We aim to create a non-invasive system which can run on a variety of 3D applications in real-time.

3. Rendering Techniques

We apply image based techniques to first abstract the scene and then add edges. First the Kuwahara filter is used to abstract the original image from the framebuffer. A Difference of Gaussian edge detector is then used to find the significant edges in the image.

3.1. Painterly Abstraction

The Kuwahara filter works on a per-pixel basis, it calculates the mean colour and variance for each of four regions surrounding each pixel. These regions are generated by dividing the surrounding area into four overlapping square windows each containing the centre pixel. The output colour for each pixel is the mean colour for the surrounding region with the smallest variance. This has the effect of blurring the image while keeping the edges between colours sharp. The Kuwahara filter gives good coherence between frames with little or no flicker.

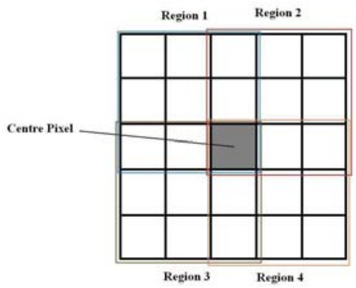

Figure 2: *Example of a Kuwahara Filter with 3x3 sampled regions, each indicated by a separate colour.*

Depending on the size of the sampled regions, a different level of abstraction can be achieved. While large sampled regions (5x5 and above) give more blur they don't preserve edges as accurately. If more abstraction is required it is preferable to instead apply a smaller filter iteratively, such as 3x3, which still preserves edges but increases the abstraction effect.

3.2. Edge Detection

Difference of Gaussian (DoG) edge detection was chosen above other edge detection algorithms as it gives a good compromise between accuracy and speed. Whilst simplistic filters such as Sobel are quick they are not accurate enough and more complex techniques such as Canny provide excellent results but are too complex for interactive frame rates.

DoG edge detection applies a Gaussian blur to an image using two different sigma values. The normal Gaussian blur function is as follows:

$$G(u,v) = (\frac{1}{2\pi\sigma})e^{\frac{-(u^2+v^2)}{2\pi^2}}$$

The edge image can then be discovered by finding the difference between the two blurred images. The Marr-Hildreth ratio [MH80] of 1.6 between the sigma values is used to achieve the most accurate edge detection.

4. Adaptive Abstraction

Using the NPR techniques described, different levels of abstraction can be created around regions of interest in the scene. These regions in a scene can be rendered using less abstraction and bolder edges to increase prominence. They can also be used to calculate the important objects in a scene and increase their salience from the rest of the scene.

4.1. Abstraction Levels

The Kuwahara filter can be used with varying sampled region sizes to create different levels of abstraction. These filters can also be applied iteratively to achieve more abstraction. These different options for the Kuwahara filter can then be used to achieve a smooth transition of increasing abstraction from visually important points to extraneous areas. The filter used at each pixel is chosen by calculating the distance to the point of focus using the simple distance formula. Figure 3 shows the transition between regions of low abstraction and high abstraction.

We can also add adaptive edges by taking the filtered abstracted image and performing full DoG edge detection on it. The edges in the resulting image can then be alpha blended with our abstracted colour image. The amount of transparency is dependant on the distance from the point of focus in the frame, making the edges in focus bolder and more noticeable.

To further increase the contrast in abstraction within the scene the white background from the edge image can also be alpha blended with the colour image. This causes the unimportant areas to be smoothly bleached out.

Figure 3: *Example of a rendering which uses fading to increase saliency of the region of interest.*

4.2. Scene Segmentation

To create meaningful non-photorealistic renderings which fully highlight important content it is necessary to use the available object information. The current point of focus can be used to calculate which pixels in the screen belong to the object at that point. The scene is first rendered with each object in a distinct colour using flat shading. The colour image retrieved from this rendering is then used by the pixel shaders to extract the pixels of interest from a frame. Focus objects can then be rendered in single Kuwahara and edge abstraction levels to make the object distinct from the rest of the scene. An example of this can be seen in figure 4.

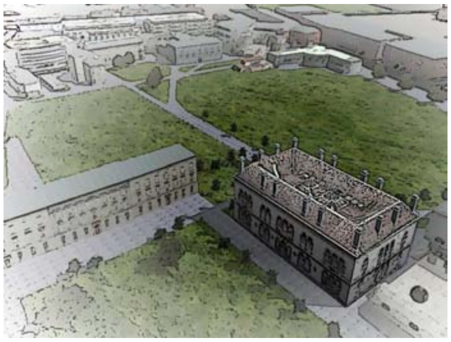

Figure 4: *Example of a rendering which uses object information to abstract the image.*

5. Discussion and Future Work

In this paper we have proposed a novel technique for generating non-photorealistic renderings of 3D scenes in real-time. The system adapts to a point within a scene to create different levels of abstraction which can be used to direct attention to particular points or objects within a scene.

The major advantage of image space methods such as the ones described is that the speed of these methods are dependent on screen resolution rather than scene complexity. The system was tested on a Pentium D, 3.72GHz computer with 2GB of RAM using a NVidia GeForce 7950 GX2 GPU. Highly interactive rates were achieved using our system on Virtual Dublin which contains over 50,000 polygons. This can be seen from the frame times in table 1.

RenderingStyle	640*x*480	800*x*600	1200*x*800
Regular rendering (no abstraction)	45	45	45
DoG edge detection	44	44	44
Kuwahara filter (3x3 sampled regions)	42	36	23
Kuwahara filter (5x5 sampled regions)	23	18	10
Adaptive abstraction (Varied Kuwahara and DoG)	46	38	24
Adaptive abstraction with scene segmentation	32	25	17

Table 1: *Frames per second achieved by the system using various abstraction options on Virtual Dublin at a different resolutions.*

The Kuwahara and edge abstraction are image space methods and can be implemented with an application in a completely non-invasive manner. However the scene segmentation algorithm requires some knowledge of the 3D model data. To segment each frame using our system direct access to the application code is required. While this does not comply with our goal of a completely non-invasive system the scene segmentation is necessary for meaningful abstractions and the invasiveness of the unique colour rendering can be kept to a minimum.

The regions of interest discussed can be created in a number of ways. They could be predefined to encourage the user of an application to look at certain points. These regions could also adapt to a users goal in an application to highlight a path or object of current interest. They could also be created using an eye-tracking device to constantly adapt to a users eye movement. Future research will include using an eye-tracker to investigate the effect of a users focus driving the abstraction on-screen in real-time. An eye-tracker will also be used to evaluate the effectiveness of the techniques used at drawing users attention to particular scene points and objects.

References

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence 8* (1986), 769–798.

[Her98] HERTZMANN A.: Painterly rendering with curved brush strokes of multiple sizes. *Computer Graphics 32*, Annual Conference Series (1998), 453–460.

[HO03] HAMILL J., O'SULLIVAN C.: Virtual Dublin - A Framework for Real-Time Urban Simulation. *Journal of WSCG 11* (Feb. 2003).

[Mei96] MEIER B. J.: Painterly rendering for animation. *Computer Graphics 30*, Annual Conference Series (1996), 477–484.

[MG01] MOHR A., GLEICHER M.: Non-invasive, interactive, stylized rendering. In *Symposium on Interactive 3D Graphics* (2001), pp. 175–178.

[MH80] MARR D., HILDRETH E.: Theory of Edge Detection. *Royal Society of London Proceedings Series B 207* (Feb. 1980), 187–217.

[MKK76] M. KUWAHARA K. HACHIMURA S. E., KINOSHITA M.: *Digital processing of biomedical images*. Plenum Press, 1976.

[SD02] SANTELLA A., DECARLO D.: Abstracted painterly renderings using eye-tracking data. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (2002), ACM Press, pp. 75–ff.

[WOG06] WINNEMOLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), ACM Press, pp. 1221–1226.