A Hybrid Technique for Creating Meaningful Abstractions of Dynamic 3D Scenes in Real-Time

Niall Redmond
Graphics, Vision and Visualization Group, Trinity College Dublin, Ireland
redmondn@cs.tcd.ie

John Dingliana
Graphics, Vision and Visualization Group, Trinity College Dublin, Ireland
john.dingliana@cs.tcd.ie

ABSTRACT

We present an approach for creating non-photorealistic renderings of 3D scenes in real-time. We employ a hybrid system which uses both image-space and object-space techniques for creating fast and effective results. A reliable method of edge detection is presented to find all important edges within a scene. This edge detection technique is combined with a painterly renderer to render the scene using different levels of abstraction. This is used to increase the saliency of important objects and remove extraneous detail. 3D object information is used to apply an object-based segmentation technique which allows each scene object to be rendered using a single abstraction level depending on its scene importance. The abstraction techniques are implemented on the GPU which helps the system achieve interactive rates.

Keywords: NPR, Adaptive, Real-time, GPU.

1 INTRODUCTION

Non-photorealistic rendering (NPR) has become a popular technique for both stylization of images and also for emphasizing important communication content within images or scenes. NPR can create simple renderings of complicated scenes to make them faster and simpler to comprehend. This can be done by borrowing techniques which have long been used by artists to control a viewers focus on an image. NPR is therefore heavily concerned with human perception of scenes and images.

Our main goal is to use non-photorealistic abstraction to produce meaningful renderings which emphasize vital features of an image and reduce the impact of unimportant information. We present a system which uses 3D object information in the scene to create meaningful and natural non-photorealistic renderings. By doing this we aim to allow for user judgment to be effectively influenced and guided in interactive scenes. Preliminary results for the system were presented in [19].

The majority of existing automatic NPR techniques for 3D scenes are not applicable to generic scenes as they are too expensive to run in real-time, are tailored for limited scene sizes or will only work for a single type of 3D model, such as polygon models [16] or volumetric models [23]. These types of object based methods require complete knowledge of the geometry detail in the scene and they are often unsuitable where important detail is only available in texture. Image-based abstraction techniques can solve these problems as access to the model and scene data is not necessary and texture detail can be taken into account. However most existing image based abstraction techniques are suited only for single images or video and do not exploit available 3D object information. We present a system which uses a hybrid approach to create non-photorealistic renderings. Image-space techniques are used to ensure interactive rates while object-space data is also used to create renderings which can adapt to the scene and create natural looking results.

Figure 1: Example of our object based abstraction

To demonstrate our system we use Virtual Dublin [7], a virtual urban environment, developed in Trinity College, which runs in real-time. This is a typical interactive application where our system could be used to emphasize or omit detail in order to influence a users

WSCG 2008 conference proceedings, ISBN
WSCG'2008, February 4 – February 7, 2008
Plzen, Czech Republic.

choices in the scene. It is also an example of a large complex scene, containing over 80,000 polygons. The building objects themselves are relatively coarse in their geometry but the textures contain much detail. Our abstraction techniques could be used to highlight particular historical landmarks or popular sights in Virtual Dublin for tourism purposes.

For our system to be able to run effectively in interactive applications it is necessary for it to run in real-time. We achieve this using image space NPR methods which are implemented on GPU pixel shaders. Past research shows that image space NPR techniques can be used effectively to produce real-time applications [25] [11].

The paper is organised as follows. Section 2 reviews past research in various types of non-photorealistic rendering, studies of NPR with different levels of abstraction and also research in the relationship between psychology and NPR. The non-photorealistic rendering methods used are described in Section 3. Section 4 contains information about the different levels of abstraction created, the method of segmenting the scene and the system used to determine each objects' abstraction levels. Section 5 contains results obtained from our system while Section 6 discusses the work and puts forward possible future work.

2 RELATED WORK

There has been much research into various types of automatic non-photorealistic rendering, both using image-space and object-space techniques [8] [16] [25]. The most popular styles include painterly rendering, pencil shading, silhouette drawing and cel-shading. Painterly rendering is the most heavily researched style of NPR. Many different techniques exist for creating painterly renderings although the majority of these are only suitable for single images and not for real-time applications due to processing time and frame to frame coherence.

Among the earliest work on painterly rendering was the research by Haeberli in 1990 [4]. Haeberli introduced a semi-automated system which gave the user control over the painterly image that was being created. While it was a simplistic method which required a lot of user interaction it proved that convincing painterly renderings were possible. Following on from that automatic methods were presented for creating painterly renderings for both single images and 3D scenes. For image based painterly renderings [13] [8] the automatic simulation of various size brushstrokes meant a large amount of processing. Similarly the 3D particle system presented in [16] for creating painterly renderings for 3D scenes was an expensive one and impossible to run for large scenes in real-time.

Real-time methods for obtaining NPR results are presented in [25] where image based techniques are used on video to produce non-photorealistic results. Each frame of the video is put through a series of image processing filters. First an anisotropic diffusion filter is used to produce a painterly like effect, then an edge detector and finally a colour quantization filter. A simliar filter to the anisotropic diffusion filter used by Winnemoller et al. was earlier put forward in [14]. Here an edge preserving smooth filter is presented which gives a painterly effect. This filter, known as the Kuwahara filter, works on a per pixel basis and is both fast and effective.

Santella and DeCarlo have done much research on the creation and evaluation of non-photorealistic images with various levels of detail. In [21] eye tracking data is used to guide automatic painterly renderings of images. The eye-tracker is used to first retrieve the visually important parts of an image then abstraction is performed. The abstraction uses brushstrokes of various sizes depending on the visual importance of each point. Following on from that paper Santella and DeCarlo evaluated their work in [22]. Eye-tracking data was again used, this time to investigate whether images created using the original system successfully focussed a users interest on particular parts of an image. The results validated the design of their original system and showed that a users focus can be guided using NPR. The renderings produced by Santella and DeCarlo are quite effective although the technique is only applicable to single images. The images also suffer from unwanted background detail near focus points in the image due to the constrains of the purely image-space technique.

In [6] the relationship between NPR and psychology is explored. Here it is found that NPR can influence user perception and judgement. Users tended to select objects with bold edges over other objects within a scene. It was also shown that NPR can be used to guide user behaviour in navigation and exploration, users viewed paths with higher detail as more interesting for exploration relative to paths with low detail.

Edges can be an important part of non-photorealistic rendering. The addition of edges to a scene can help create clearly distinct regions and bold edges can help to direct user focus to particular objects. Through perceptual tests in [5] it is shown that humans are good at inferring the shape of an object from its line drawing alone. Edges can therefore be useful when rendered on their own and much research has gone into the generation and stylization of edges [17] [12] [3].

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 accurate as object space edge detectors, can achieve good results and run at interactive rates. Some hybrid techniques exist which use image-space edge detectors to find discontinuities in the depth buffer and therefore find 3D edges [20]. These methods can find object silhouettes quite well although they are dependant on the size of the depth buffer and miss edges between objects at the same depth level.

The Canny edge detector [2] 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 [15] 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.

NPR has also been used successfully in visualization. [1] presents a technique for using NPR for medical visualizations with emphasis and focussing methods. Silhouettes and hatching are used among other methods. The results are effective although impossible to run in interactive applications. A method for pen-and-ink rendering for volumetric visualization is described in [23] where NPR is shown to be an effective mechanism for improving interpretation of images. NPR could be used for other types of visualization such as vehicle or housing projects.

Virtual Dublin [7], a large-scale immersive urban simulation developed in Trinity College, is used to demonstrate our system. It simulates Dublin city centre and the surrounding areas and contains over 80,000 polygons. Virtual Dublin is an ongoing project with many research areas, such as crowd simulation, traffic simulation, and multisensory perception. Our abstraction system could be used effectively with an application such as Virtual Dublin to help guide a user around the city.

3 RENDERING TECHNIQUES

Our system uses image space techniques implemented on the GPU, which work on a frame to frame basis. To effectively highlight important content within a scene a painterly renderer and an edge detector were used. First the Kuwahara filter is used to abstract the scene in a painterly style and then a Difference of Gaussian edge detector is used to highlight the significant edges in the image.

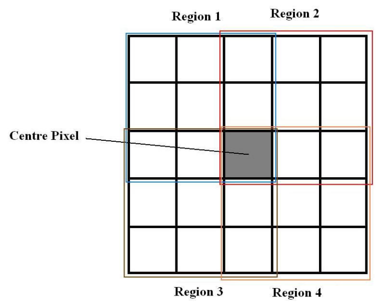

Figure 2: Example of the layout of a 5x5 Kuwahara filter containing 3x3 sampled regions. Each of the four regions is indicated using a seperate colour

3.1 Painterly Abstraction

To produce effective non-photorealistic renderings we implemented a painterly abstraction technique to obtain various levels of detail within an image. The Kuwahara filter, an edge-preserving smooth filter, was chosen as it is suitable for real-time applications, it gives a convincing painterly effect and it has good frame-to-frame coherence with no flicker.

The Kuwahara filter works on a per-pixel basis throughout the image. The filter calculates the mean colour and variance in each of four adjacent regions for each pixel. These regions are generated by dividing the surrounding area into four overlapping windows, each containing the centre pixel itself (see Figure 2). The output colour for each pixel is the mean colour for the adjacent window with the smallest variance. This has the effect of smoothing internal regions while keeping the edges between colours sharp.

Using varying sampled region size can give different levels of abstraction. Large filters, such as 9x9, produce more smoothing but don't preserve edges as accurately. Large filters are also significantly slower. If a larger level of abstraction is required and edges still need to be preserved it is preferable to apply a smaller filter iteratively (e.g. 5x5). This has the effect of preserving edges and increasing the abstraction. Using various filter sizes in combination with applying filters iteratively can produce many different levels of abstraction (see Figure 4).

3.2 Edge Detection

As discussed in Section 2, edges are an excellent way of directing user focus. Image-space edge detectors are not as accurate as object-space edge detectors but are needed for speed and also for finding edges within textures which may appear in a scene. Texture edges are important within a large environment such as Virtual Dublin because large amounts of complex textures are

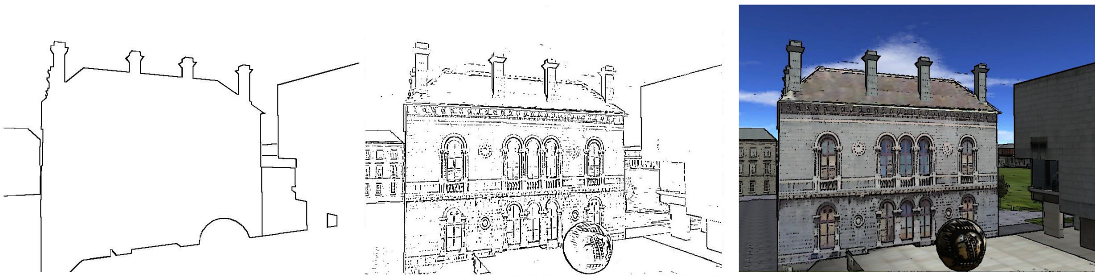

Figure 3: Edge detection example: 3D Silhouette edge image(Left), Image space edge image(Middle) and the two overlaid on the original image to display all scene edges(Right)

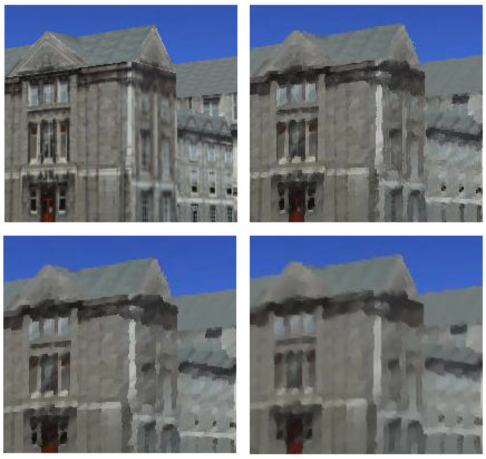

Figure 4: A section of an image when there is no abstraction(top left), a 3x3 Kuwahara filter(top right), a 3x3 Kuwahara filter applied twice(bottom left) and a 5x5 Kuwahara filter(bottom right)

typical of this sort of application. These textures may contain important information which may be emphasized to the user.

Image-space edge detectors can achieve good results and work at interactive rates although they can be unreliable due to the fact that some 3D edges can be missed due to colour similarities. We present a hybrid edge detection system which uses both object-space and image-space techniques to reliably find all edges in the scene, including 3D object silhouettes and edges which exist within textures.

Difference of Gaussian (DoG) edge detection was chosen to be the image-space edge detector used in our system. More accurate but expensive detectors exist such as Canny, while the Sobel filter is faster but too unreliable. DoG was seen as a good compromise between speed and accuracy. Difference of Gaussian edge detection works by simply applying a gaussian blur to an image using two different blur levels. The resulting edges can then found by calculating the difference between the two blurred images and finding all pixels with a difference under a threshold.

The Gaussian blur function can be seen in Equation 1 below, the function is used for creating a Gaussian mask from a sigma value. The mask is then passed through the image to blur each pixel. Each of the two blur images is created using a different sigma value. The Marr-Hildreth [15] ratio of 1.6 between the blurred images was used to give the most effective result.

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

σ = standard deviation of the Gaussian distribution

(u,v) = distance from centre of mask

The Difference of Gaussian edge detection is run for the framebuffer image to find the majority of the scene edges, including any edges which might occur within object textures. The same DoG edge detection process is then run on the false colour image generated from the first-pass rendering (see Section 4.2). By doing this all 3D silhouettes edges are found for scene objects. As can be seen from Figure 3 the 3D silhouette image finds edges which the image-space detector might miss, in the example edges are missed particularly around the roof. Implementing these two edge detectors gives a reliable method of finding all significant edges within a scene.

While DoG edge detection is run for both the framebuffer image and the false colour image, a different edge threshold is used for each image. This is because all edges from the simple false colour image are desirable whereas only the strongest edges from the framebuffer image are needed. DoG edge detection can be implemented on a pixel-by-pixel basis making it ideal for GPU pixel shaders. The edges found in the edge detection process can then be overlaid on the original image to give convincing results containing all scene edges (see Figure 3).

4 SCENE ABSTRACTION

4.1 Abstraction Levels

Using the two NPR techniques described, it is possible to produce different levels of abstraction. These levels

Figure 5: Examples of our abstraction levels. Left (level 1, most salient), Middle (level3) and Right (level 5)

can be used to increase the contrasts between scene objects and therefore guide a users focus within a scene more successfully. In addition to these methods the implementation of a fading algorithm is also introduced to increase object saliency.

The fading algorithm is done in two parts, first the colour of an object can be altered by utilizing simple operations on the RGB values of the object. This fades the objects colour and therefore lowers its visual saliency in the scene. Edges can also be faded out by altering the alpha value of each edge pixel to increase its transparency and blend it with the colour from the underlying normal rendering. This serves to make some edges bolder than others and give them more prominence in an image. Combining these techniques enable objects of high importance to be rendered with increased saliency by using clearer abstraction levels and bolder edges. Similarly objects of lesser relevance can be rendered with large abstraction levels and weak edges.

Our system can use five levels of abstraction within scenes depending on object significance. Application of the Kuwahara filter with a combination of varying filter size and multiple applications of the filter allows the generation of the different abstraction levels. These multiple degrees of painterly rendering allow for smooth abstraction transitions within a scene.

Five levels of abstraction for both colour and edge fading are again used to match with the five painterly abstraction levels. Five levels were chosen as we found it to be a sufficient number to successfully abstract a scene smoothly. It is possible to introduce more levels of abstraction if necessary. An example of three of the finished abstraction levels can be seen in Figure 5 and a detailed description of the five degrees of abstraction we use can be seen in Table 1.

Level	*Kuwahara*	*RGBFade*	*EdgeFade*
1	None	No change	No transparency
2	5x5 filter	0.12 addition to each colour	0.08 transparency with alpha value
3	5x5 filter applied twice	0.18 addition to each colour	0.16 transparency with alpha value
4	9x9 filter	0.24 addition to each colour	0.24 transparency with alpha value
5	9x9 filter applied twice	0.30 addition to each colour	0.30 transparency with alpha value

Table 1: Detailed description of the five levels of abstraction used

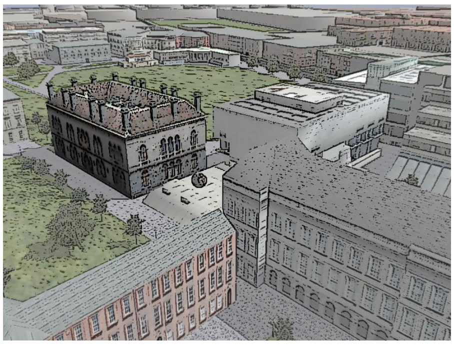

Figure 6: An example of a scene with object based-segmentation. One object is rendered clearly while the rest of the scene is abstracted

4.2 Scene Segmentation

To create meaningful non-photorealistic renderings which fully highlight important objects and find all edges it is necessary to use the available 3D object information. We implemented a method which uses false colour rendering of a scene to find edges and guide abstraction. False colour rendering is a popular technique used in synthetic vision for object identification [18] and similar techniques in NPR use an ID-buffer to segment a scene e.g. in [24] to give each object a different halftoned texture.

When rendering each frame of the scene we use a two pass system. We first render the scene with each object rendered in a distinct colour. The amount of unique colours available depends on the depth of each component in the colour buffer. When rendering during the false colour pass it must be ensured that all objects are rendered in flat colour, therefore lighting, textures etc. must be disabled. The simplicity of the rendering pass makes it fast enough for interactive applications. The false colour rendering pass is a useful tool for object identification on each frame.

4.3 Object-based Abstraction

It is desirable to render each object in an abstraction style which suits that objects importance in the scene. This allows the effective rendering of scenes which we propose will draw an users attention to certain objects and paths. We put forward an object classification algorithm for doing this based around the most visually important objects within the scene.

The information gained from the false colour rendering can be used to calculate which abstraction level each object should be rendered in. The first step to doing this is calculating the centre point of each visible salient object. These visually important objects can be pre-defined, set by the current mouse position or image centre, or be dependant on a users current goal or interest. For each pixel in the image the distance to the nearest important objects centre point is found.

After the image has been fully traversed the abstraction levels for each scene object must be calculated. This is necessary for creating meaningful NPR renderings which increase saliency of certain objects but also fit in naturally with the scene. As can be seen from Figure 9 there is an obvious difference between the rendering done using only distance as a guide for abstraction compared with the rendering which does the abstraction on a per-object basis.

The per-object abstraction uses the false colour image and distance information for each pixel to calculate the average distance for each scene object. This is done by first inspecting the false colour image to discover which pixels belong to a certain object and then using the distance information for those pixels to find an average. Depending on the result each object can be defined as belonging to one of the five abstraction levels. This information can then be passed to the pixel shader to do the necessary abstraction.

The pixel shader, written in CG, is responsible for implementing the abstraction itself. Only one pixel shader is used to do all abstraction. The work performed by the shader depends on which abstraction level has been defined for each pixel. The shader first performs the Kuwahara filtering followed by the edge detection process. It is done in this order as some edges may be altered in the painterly process, especially for high levels of abstraction. Finally the edge and colour fade are performed on the pixels values, again dependant on the abstraction level for the pixel. The final output pixel

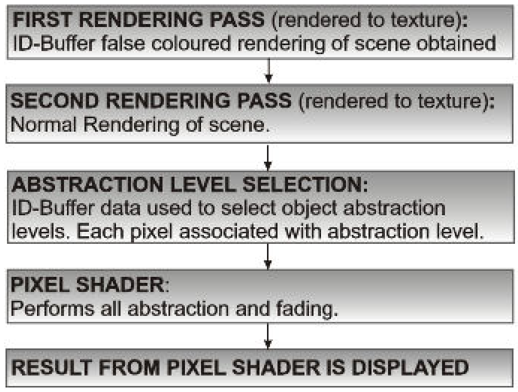

Figure 7: Overall algorithm for the system

is then written to a texture which can be displayed on-screen when the shader has completed. The overall algorithm for our system can be seen in Figure 7.

5 RESULTS

We implemented our system using C++ with OpenGl and the GPU pixel shader was written in CG. The system was tested on a Pentium D, 3.72GHz computer with 2GB of RAM using an NVidia GeForce 7950 GX2 GPU.

The implementation of the abstraction techniques on the GPU coupled with the fast false colour rendering techniques mean that interactive frame times are achieved, which can be seen from table 2, below. The major advantage of using image space methods such as the ones implemented here is that they are dependant on screen resolution rather than scene complexity which means frame rates can be guaranteed. The object segmentation algorithm does depend on scene complexity but the speed of the false colour rendering makes the slow-down minimal, even for large complex scenes such as Virtual Dublin, on which this system was tested.

RenderingStyle	640*x*480	800*x*600	1200*x*800
Regular rendering (no abstraction)	102	100	98
2 pass rendering (false colour pass)	54	53	52
Edge detection (with 2 pass)	50	47	45
Painterly rendering (with 2 pass)	51	48	47
Full abstraction system	46	43	42

Table 2: Frames per second achieved by the system using various abstraction options on Virtual Dublin at a different resolutions.

As can be seen from the frame times, the biggest slowdown of the system is the false colour rendering

Figure 8: Example of two scene images with and without abstraction. The yellow markers indicate the most salient points of each image

although Virtual Dublin still runs at interactive rates. The two pass rendering method is also vital for our system and is the fastest way of achieving object segmented renderings. The table also shows the fact that the image based abstraction methods do depend on image resolution although the slowdown is minimal for large changes in resolution. This is largely thanks to the speed of the algorithm when implemented on the GPU.

The results from our system were evaluated using a saliency mapping algorithm [9]. This technique uses a multi-scale approach for finding the location of salient regions based on local image structure. Regions of high contrast are found for colour, intensity and orientation of an image on a number of scales. The results are then normalised and summed to create the final saliency map for an image.

We used the saliency mapping system to confirm that our techniques can successfully increase the saliency of certain objects within a scene. Figure 8 shows two scene images both with and without our abstraction methods. This figure proves that adding abstraction to a scene image can change the most salient point to the object we want to emphasize in the scene. While user studies would provide more worthwhile information about the effectiveness of our system these preliminary tests show that our methods are successful and could be used to guide a users focus in a scene.

The objects of interest in each scene could be chosen in a number of ways depending on the application. They could be pre-defined to encourage a user to take certain routes or to look at certain objects within the scene. In an application such as Virtual Dublin this could be highlighting historical monuments or popular museums. The visually important object at any one time could also be dependant on a users goal in an interactive application at any particular time to make the current objective clear. Along with tourism the system would be useful in games or route-finding applications.

6 CONCLUSION AND FUTURE WORK

We have presented a novel technique for generating meaningful non-photorealistic renderings of large 3D scenes in real-time. Our main contribution is a hybrid method for rendering each scene object in a particular abstraction level which is suitable for its importance in the scene. We have shown that our methods could be used to create more salient objects and guide a users focus, thereby directing and influencing user judgement. We also present a novel hybrid method for effectively combining object space edges and edges within textures to find all scene edges.

Our system successfuly provides interesting variable abstractions of a scene in real-time which was our main goal. We also validate our abstraction methods by showing that scene saliency can be altered successfully. To fully investigate the effectiveness of the system at directing user attention, we propose future work in the field of perception.

An eye-tracker may be used such as in previous work like [22] where it is proved that eye-tracking data can be a useful tool for the evaluation of NPR systems or in [10] where saliency-guided enhancement was evaluated with eye-tracking data. In addition, eye-tracking experiments may be useful in tuning the abstraction parameters to create the most effective abstractions of the scene. Further research could include task based experiments to evaluate the success of the renderings. Comparison and evaluation of tasks performed both with and without abstraction would give an accurate measure of the effectiveness of the object based abstraction system.

REFERENCES

[1] Christian Tietjen Bernhard Preim and Christian Dorge. Npr, focussing and emphasis in medical visualizations. *Simulation and Visualisierung*, pages 139–152, 2005.

[2] J.F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:769–798, 1986.

[3] Cassidy J. Curtis. Loose and sketchy animation. In *SIGGRAPH '98: ACM SIGGRAPH 98 Electronic art and animation catalog*, page 145. ACM Press, 1998.

[4] Paul Haeberli. Paint by numbers: abstract image representations. In *SIGGRAPH '90*, pages 207–214. ACM Press, 1990.

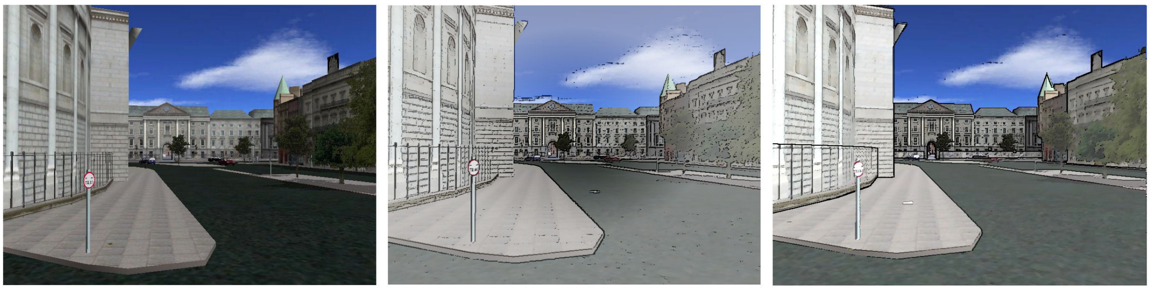

Figure 9: A comparison between the original rendering, a purely distanced based abstractive rendering and our finished hybrid abstraction method, which focuses on the far building at the centre of the image

[5] Nick Halper, Mara Mellin, Christoph S. Herrmann, Volker Linneweber, and Thomas Strothotte. Surface gradients, contours and the perception of surface attitude in images of complex secenes. *Perception*, 25:701, 1996.

[6] Nick Halper, Mara Mellin, Christoph S. Herrmann, Volker Linneweber, and Thomas Strothotte. Psychology and non-photorealistic rendering: The beginning of a beautiful relationship. pages 277–286, 2003.

[7] John Hamill and Carol O'Sullivan. Virtual Dublin - A Framework for Real-Time Urban Simulation. *Journal of WSCG*, 11:221–225, February 2003.

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

[9] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[10] Youngmin Kim and Amitabh Varshney. Saliency-guided enhancement for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):925–932, 2006.

[11] Allison W. Klein, Wilmot Li, Michael M. Kazhdan, Wagner T. Corrêa, Adam Finkelstein, and Thomas A. Funkhouser. Non-photorealistic virtual environments. In *SIGGRAPH '00*, pages 527–534. ACM Press/Addison-Wesley Publishing Co., 2000.

[12] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3d animation. In *NPAR '00:*, pages 13–20. ACM Press, 2000.

[13] Peter Litwinowicz. Processing images and video for an impressionist effect. In *SIGGRAPH '97*, pages 407–414. ACM Press/Addison-Wesley Publishing Co., 1997.

[14] S. Eiho M. Kuwahara, K. Hachimura and M. Kinoshita. *Digital processing of biomedical images*. Plenum Press, 1976.

[15] D. Marr and E. Hildreth. Theory of Edge Detection. *Royal Society of London Proceedings Series B*, 207:187–217, February 1980.

[16] Barbara J. Meier. Painterly rendering for animation. *Computer Graphics*, 30(Annual Conference Series):477–484, 1996.

[17] J. D. Northrup and Lee Markosian. Artistic silhouettes: a hybrid approach. In *NPAR '00:*, pages 31–37. ACM Press, 2000.

[18] Hansrudi Noser, Olivier Renault, Daniel Thalmann, and Nadia Magnenat Thalmann. Navigation for digital actors based on synthetic vision, memory, and learning. *Computers and Graphics*, 19(1):7–19, 1995.

[19] Niall Redmond and John Dingliana. Adaptive abstraction of 3d scenes in real-time. In *EUROGRAPHICS '07: Proceedings of the Short Paper Programme*, pages 77–80, September 2007.

[20] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. *SIGGRAPH Comput. Graph.*, 24(4):197–206, 1990.

[21] Anthony Santella and Doug DeCarlo. Abstracted painterly renderings using eye-tracking data. In *NPAR '02:*, pages 75–ff. ACM Press, 2002.

[22] Anthony Santella and Doug DeCarlo. Visual interest and npr: an evaluation and manifesto. In *NPAR '04:*, pages 71–150. ACM Press, 2004.

[23] S. M. F. Treavett and M. Chen. Pen-and-ink rendering in volume visualisation. In *VIS '00*, pages 203–210. IEEE Computer Society Press, 2000.

[24] O. Veryovka and J. Buchanan. Comprehensive halftoning of 3D scenes. In *Computer Graphics Forum (Eurographics '99)*, volume 18(3), pages 13–22. The Eurographics Association and Blackwell Publishers, 1999.

[25] Holger Winnemoller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. In *SIGGRAPH '06*, pages 1221–1226. ACM Press, 2006.