Towards a perceptual method of blending for image-based models.

Gordon Watson, Patrick O'Brien† and Mark Wright

Edinburgh Virtual Environment Centre
University of Edinburgh
JCMB, Mayfield Road,
Edinburgh EH9 3JZ
<g.c.watson, mark.wright>@ed.ac.uk

†Now at Sony Entertainment Europe.
patrick@optikal.demon.co.uk

Abstract
We present a technique for view-dependent texture mapping that attempts to minimise the perceived artefacts that can arise from image blending. These artefacts result from imperfect image registration, typically due to un-representative scene geometry, or scene motion during capture. Our method draws inspiration from work in image mosaicing, but uses a metric based on perception of Mach bands that we also use to quantitatively evaluate the method. We discuss the implications of this work to creating a fully perception-based method of image-based rendering.

Keywords: image blending, image-based rendering, perceptual methods.

1 Introduction

The image-based rendering field of computer graphics attempts to produce novel views of a scene by reconstructing the plenoptic function from discrete samples [1, 10]. The samples usually take the form of photographs of the scene, and reconstruction involves sample interpolation. The approach has several advantages over traditional three-dimensional graphics, including constant rendering cost regardless of scene complexity and photorealistic output.

A key operation in IBR is therefore interpolating the discrete samples to produce a new approximation of the plenoptic function for the novel view. To date, image-based rendering methods have employed interpolation schemes with simple mathematical formulations that pay little attention to how the resulting images are actually perceived. We believe an interpolation scheme based upon image perception would both improve the quality of rendered images and extend the scope of image-based rendering, for example, to the capture and rendering of natural outdoors environments.

In this work we focus on a specific image-based rendering technique referred to in [4] as view-dependent texture mapping (VDTM). A geometric model of the scene (termed a geometric proxy by Buehler et al. [2]) is recovered from the input images using photogrammetric modelling techniques. The geometric proxy is texture mapped by projecting the photographs back onto it, producing novel views of the scene, VDTM assumes only an approximate geometric model of the scene is available but can give the appearance of more complex geometry being present.

In common with other image-based rendering techniques, VDTM employs either linear interpolation between the images, or nearest-neighbour.

We will show that in VDTM these interpolation techniques can generate artefacts when geometric features in the real scene are not modelled by the proxy. A highly accurate proxy is difficult to recover from a sparse set of input images, so we seek to reduce these artefacts by controlling interpolation using a perception based metric.

The metric allows us to classify sample components and interpolate them at different rates thus reducing interpolation artefacts. We use the same perception based metric to verify the quality of our results.

2. Background and Related Work

IBR attempts to model a scene using a continuous representation of the plenoptic function [10]. The plenoptic function was first proposed by Adelson and Bergen in [1] and describes all the rays of any wavelength that are visible from a point in space.

Producing a continuous representation of the plenoptic function requires some way of processing discrete samples of the function to provide a reasonable approximation. Many methods of doing this have been proposed. For example, Lippman's Movie Map technique [8] simply chooses the sample which is closest to the current viewing position (equivalent to nearest neighbour interpolation). Other methods are based on interpolating the samples using image flow fields [3, 7, 10].

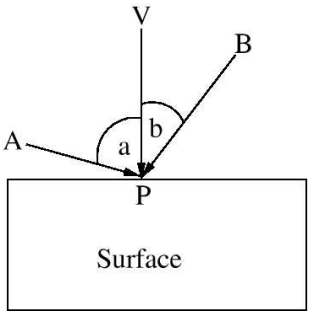

Figure 1: The correct choice of photograph to use for texturing point P depends on the relative positions of the real cameras A & B, and the virtual camera V.

View-dependent texture mapping, presented by Debevec et al. in [4], uses a fitness function to assess the suitability of each input image for texturing the geometric proxy.

The fitness function considers the relative gaze angles of the viewer and the camera that captured the image under consideration. The smaller the divergence between the angles, the more suitable the image for texturing. Figure 1 illustrates this with two cameras. Intuitively, it is desirable to use the image captured by camera B. The fitness function weights each image's contribution to the final colour of the point being textured.

In [2], Buehler et al. present an IBR algorithm which generalizes many existing IBR approaches. With a sparse set of input images and an accurate geometric proxy, the algorithm behaves like VDTM. This approach uses a more advanced fitness function, considering sample resolution in addition to angle similarity.

Both [4] and [2] use the fitness function to linearly interpolate the samples. We will show that when the geometric proxy is approximate (as is often the case due to the low number of input images), piecewise linear interpolation generates artefacts.

Pollard et al. have shown that IBR artefacts can be reduced by interpolating different texture frequencies at different rates [11]. In this paper we examine the nature of these artefacts and extend the technique. We use a metric based on the human visual system to identify the texture components which should be blended at different rates. The new metric is then used to evaluate the quality of results.

2.1 The Mach Band Effect

An edge-ramp in an image corresponds to high spatial frequencies. The human vision system is well adapted to identifying these high frequencies that cause a perceptual phenomenon known as the Mach band effect (discovered in 1865 by Mach, an Austrian physicist), which emphasizes the border between two regions of differing intensities. Close to the border, the region on the light side looks lighter and the region on the dark side looks darker. These regions only appear close to the border, and they are called Mach bands. A detailed study of Mach bands can be found in [12].

Mach bands are relevant in computer graphics because the eye is naturally drawn to the regions containing them. Of course, Mach bands in a computer generated image are not necessarily artefacts because we see them around edges in the real world. However, a Mach band may greatly emphasize existing artefacts. For example, in computer graphics, smoothly curving surfaces are often approximated using polygons. When the polygonal surface is illuminated the polygons in the surface will have slightly different intensities due to their differing orientations relative to the light source. These differing intensities produce obvious edges between the polygons producing a faceted appearance. The edges are emphasized by Mach bands and the illusion that we are viewing a smoothly

curving surface breaks down. The obvious solution of increasing the number of polygons fails since it increases the number of high frequencies in the image, and therefore creates more Mach bands.

Gouraud showed that interpolating the vertex shading values allowed smooth shading across polygon borders. This eliminates intensity discontinuities and minimises Mach bands [5].

3. Existing Interpolation Techniques

3.1. Why Interpolate ?

Images of a scene are discrete samples of the plenoptic function. As described in section 2, the aim of image based rendering is to reconstruct a continuous representation of the plenoptic function from samples. In our case, the reconstruction takes place in the texture blending stage of view dependent texture mapping; the most appropriate samples are selected and interpolated to give an approximation of the plenoptic function for the current viewpoint.

3.2. Nearest Neighbour Interpolation

The easiest way to achieve the reconstruction is to use '*nearest neighbour*' interpolation. This simply means using the sample with the highest *fitness* for the current point. That is, choose the sample whose parameters (such as view direction and position) most closely match the novel view parameters.

Clearly, this means that the samples will not be blended. When the viewer moves, the most suitable sample may change, causing the surface texture map to be instantly swapped for the new 'nearest neighbour'. When this form of interpolation is used, the texture changes are very obvious due to incongruent textures. The causes of texture incongruence are explored in section 4.

3.3. Piecewise Linear Interpolation

In approaches such as [4] and [2], the plenoptic function is reconstructed using piecewise linear interpolation of the samples. Multiple textures may be blended together to give the colour for a surface point, using the fitness of an image to weight its contribution. For example, imagine we are texturing point P in figure 1 using images captured from viewpoints A and B. Call these Sample α and Sample β respectively. As the viewer moves from viewpoint A towards viewpoint B, Sample α will contribute progressively more to the point, while Sample β will contribute progressively less. When the viewer is halfway between A and B, each sample will contribute equally to the colour of the point.

Piecewise linear interpolation produces smooth changes between textures as the viewpoint changes and avoids the sudden texture switching and 'seams' which occur with nearest neighbour interpolation. However, texture incongruence may still cause artefacts with this interpolation technique.

4. Sources of Texture Incongruence

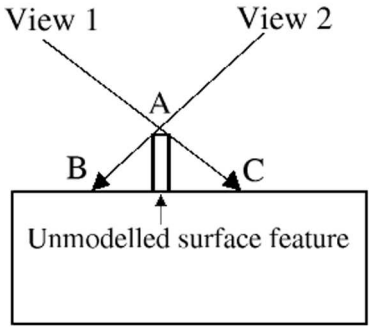

Figure 2: Depending on the viewing position, point *A* projects to a different point on the surface.

4.1 Unmodelled geometric features.

Figure 2 illustrates what happens when a photograph is taken of a surface with a protrusion. Depending on where the photograph is taken from, the protrusion will project onto a different position on the surface. If a perfect geometric proxy has been recovered, the photographs will project back onto the model in the correct manner when texture mapping is performed.

Unfortunately in practice it is not possible to recover a perfect geometric model of the scene. If the protrusion in figure 2 is not recovered, the part of the texture containing the protrusion will be projected onto the model surface behind it. The exact position on the surface where it appears depends on the viewpoint of the original photograph. Therefore photographs taken from different positions will disagree as to where the protrusion appears.

4.2 Non-lambertian Surfaces.

In the real-world many surfaces exhibit non-lambertian reflections. If a surface is slightly specular, it may appear different shades when viewed from different angles. As a result, two different photographs of the same surface under the same lighting conditions may record dramatically different colours for the surface.

4.3 Motion during capture.

In practice it is not possible to capture all the images required for an image-based model instantaneously. This implies motion may occur during capture, either to elements of the scene, or to the position or brightness of the light sources. This is a very real problem in the case of capture out of doors.

Changes in the brightness of a light source will cause artefacts similar to a non-lambertian surface, wheras motion of the scene itself will cause an artefact similar to un-modelled scene geometry – due to the images disagreeing as to the location of a scene element.

5. Implications for Blending

Nearest Neighbour Interpolation always produces the same class of artefact regardless of the cause of texture incongruence. The artefacts take the form of 'seams' on the model surface at the border between the textures. In a movie sequence, this causes an effect known as 'popping', when the current texture is abruptly changed for the new 'most fit' texture.

Piecewise Linear Interpolation copes well with texture incongruence caused by slightly specular surfaces. The transition from one texture to another is smooth with no obvious seams or popping. Texture incongruence caused by unmodelled geometry is not handled as effectively. If the two photographs in figure 2 are linearly interpolated as the viewer moves from View1 to View2, an image of the unmodelled protrusion will be simultaneously projected onto two different places on the surface. The edges in the textures being interpolated will not align, even though they are associated with the same edge in the real scene. This creates spurious high frequencies in the synthetic image, which generates undesired Mach bands. The artefacts will be at their worst when the viewer is halfway between view1 and view2, an example of this can be seen in figure 3b.

6. The Mach Band Metric.

As previously described, linear interpolation of textures that are incongruent due to unmodelled geometry produces spurious high spatial frequencies, and hence more Mach bands. The presence of spurious Mach bands can dramatically alter the perceived quality of a synthetic image [5].

If n textures are to be blended, each containing m Mach bands, the ideal situation is one in which no texture incongruence occurs – i.e the Mach bands from each image align perfectly with each other. This produces a synthetic image with m Mach bands Conversely in the worst case scenario an image with $n*m$ Mach bands could be generated in the case that no Mach bands align at all between the textures.

We propose that measuring the number of generated Mach bands in a synthetic image provides a perception based metric for comparing the relative proficiencies of different interpolation techniques. Clearly, as the Mach bands tend to m, less artefacts will be generated. An obvious way of minimising these is to use nearest neighbour interpolation. However, as previously discussed, this solution can suffer from significant artefacts in the form of `seams' on the model surface.

6.1. Desired Properties of a texture interpolation technique.

This motivates the following requirements for a texture interpolation technique:

1. Produces smooth changes between textures for non-Lambertian surfaces
2. Minimises the number of Mach bands around areas of unmodelled geometry.

7. Frequency Dependent Interpolation

We attempt to meet the requirements stated in section 6.1 by separating a texture $\mathbf{t}$ into two parts:

1. The high frequency (ie Mach band generating) component (call this $\mathbf{t_m}$).
2. The component containing no high frequencies (call this $\mathbf{t_n}$)

These components are then interpolated at different rates.

If the $\mathbf{t_m}$ components are interpolated using a nearest neighbour scheme, the high spatial frequencies from incongruent textures cannot be simaultaneously projected onto the same point, thereby minimising spurious Mach bands. Linear interpolation of the tn components produce smooth overall changes between textures.

We call this technique Frequency Dependent Interpolation. It requires a method of identifying the regions that cause Mach bands, a method for splitting the texture into the two regions, and a method of recombing the texture components on the model surface.

7.1 Identifying Mach band regions

Marr-Hildreth edge detection [9] was used to identify the high spatial frequencies that give rise to Mach bands. The Marr-Hildreth method is isotropic and closely matches the human perception of edges. The filter is equivalent to applying a high-pass filter to the texture, with the cut-off set to the response of the eye.

The result was thresholded to provide a binary map of high frequency edges.

7.2 Separating texture components.

We require a process that separate the texture into tm and tn components such that $\mathbf{t_m} + \mathbf{t_n} = \mathbf{t}$.

Marr-Hildreth filtering identifies those pixels that give rise to Mach bands. The image is dilated to form a small region around each edge, gaussian filtering is performed inside that region to remove all high frequency components. The size of the dilation filter depends only on the texture size and is simply required to give adequate support to the gaussian filter.

This result is now $\mathbf{t_n}$ and contains no Mach bands, $\mathbf{t_n}$ is then subtracted from $\mathbf{t}$ to form $\mathbf{t_m}$, which is re-scaled into a signed 8-bit image.

7.3 Texture recombination.

Now that we have $\mathbf{t_n}$ and $\mathbf{t_m}$ for each texture we must interpolate all the components in such a manner that minimises Mach bands and ensures a continuous surface colour. We use a fitness function related to the angular difference between the sample camera and the viewing position. The low frequency components are linearly interpolated using these fitness weights, wheras the high frequency components are interpolated using a nearest neighbour scheme that ensures only a single set of Mach band producing edges appear on the final image at any time.

8. Implementation.

A renderman complient ray-tracer called the *Blue Moon Rendering Tools* was used to generate the synthetic images. Textures were projected into the scene using custom light source shaders which could be configured with the intrinsic and extrinsic parameters of the camera that captured the original image. Two texture projectors replaced each camera from the original scene, one projecting tm and the other projecting tn. In addition a third projector was used to project an *alpha* channel. This allowed the weight of each image to be reduced towards the edge and allowed occluding foreground objects to be masked out.

Interpolation was achieved using custom surface shaders assigned to the model surface. Surface shaders which performed nearest neighbour, linear, and frequency dependent interpolation were implemented. This allowed different algorithms to be easily compared.

9. Results

Figure 4 shows an original image and the corresponding low and high frequency components after filtering. Figure 3a shows a frame from an animation rendered with frequency dependent blending, linear blending is shown in figure 3b. Figure 5 shows a plot of Marr-Hildreth filter response – a proxy to Mach band number, for a simple cross-fade between two images. As can be seen the frequency-dependent method reduces the overall number of Mach bands compared with the linear method except in the transition region where the two are equal.

10. Discussion & Conclusions.

The results have showed a blending method based on perception of edges can reduce visible artefacts for an image based model.

However seams still appear in the images. These are due to the algorithm 'cross-fading' between the two highest weighted images, the seam representing a transition from one '2nd best' image to another. The problem occurs if the capture cameras are not arranged in a plane. Cross-fading between more than two images reduces the appearance of visual seams, but at the expense of more cumulative errors and more Mach bands.

The solution to this problem isn't clear with the current implementation since the ray-tracer employed casts rays that are completely independent from each other, and hence a point on the surface has no knowledge of adjacent surface points. This may be a computationally desirable property for parallel execution, but it means we are not able to take into account *visual field* perception or 'hole fill' areas unseen by any camera.

Animation frames generated by the system are similarly independent from previous or future frames meaning we are unable to take into account *temporal* perception and coherence.

Clearly the eye does not perceive the world in this manner, our perception-based approach to rendering therefore raises the question as to what would be the requirements for a renderer to use a comprehensive model of visual perception, and the challenge of implemention - particularly in an interactive context.

11 Acknowledgements

The authors wish to thank Dr Paul Debevec for access to the Facade photogrammetric modelling system.

12 References

[1] E. Adelson and J. Bergen. *Computational models of visual processing*, chapter 1. The MIT Press, Cambridge Mass. 1991.

[2] C. Buehler, M. Bosse, S. Gortler, M. Cohen, and L. McMilllan. Unstructured Lumigraph rendering. *Proceedings of SIGGRAPH 2001.* 2001

[3] S. Chen and L. Williams. View interpolation for image synthesis. *Proceedings of SIGGRAPH 1993,* 1993.

[4] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Proceedings of SIGGRAPH 1996*, pages 11-20, 1996.

[5] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, pages 623-629, 1971.

[11] S. Pollard and S. Hayes. 3d video sprites. *Report HPL-98-25*, 1998.

[12] F. Ratliff. Mach Bands: Quantitative Studies on Neural Networks in the Retina. *San Fransisco: Holden-Day*, 1965.

[13] Zakia, Richard D. Perception and Imaging. *Focal Press,* 2002.

[6] L. Gritz and J. Hahn. Bmrt: A global illumination implementation of the renderman standard. *Journal of Graphics Tools*, 1(3), 1996.

[7] S. Laveau and O. Faugeras. 3d scene representation as a collection of images and fundamental matrices. *INRIA, Technical Report* No. 2205, 1994.

[8] A. Lippman. Movie-maps: An application of the optical video-disc to computer graphics. *Proceedings of SIGGRAPH 1980*, 1980.

[9] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings Royal Society of London Bulletin*, 207:187-217, 1980.

[10] L. McMillan and G. Bishop. Plenoptic modelling: An image-based rendering system. *Proceedings of SIGGRAPH 1995*, 1995.

Figure 3: A frame blended using frequency-dependent blending on the left, and linear blending on the right. Looking at the perspective at the two ends of the building it is evident they are derived from different source photographs. The geometric proxy for this façade is just a plane.

Figure 4: The original image is shown on the left, the low-frequency image in the middle, and the high frequency image (normalised) on the right.

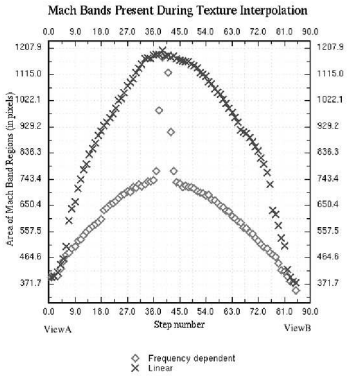

Figure 5: A plot of Marr-Hildreth filter response shows the frequency-dependent interpolation reduces the appearance of Mach bands except for the centre 'change-over' region