High Quality 3D Image Warping by Separating Visibility from Reconstruction

Voicu S. Popescu, Anselmo A. Lastra

TR99-017

Computer Science Department

University of North Carolina at Chapel Hill

High Quality 3D Image Warping by Separating Visibility from Reconstruction

Voicu S. Popescu, Anselmo A. Lastra

{popescu|lastra}@cs.unc.edu, Computer Science Dept., UNC Chapel Hill

Abstract

Image-based rendering using 3D warping (WIBR) appeals since one hopes that the quality of the reference images used can be conveyed to the warped images. However, until now, the only WIBR method that comes close to achieving this goal is based on a mesh of micro-triangles that after being warped is fed into a polygon-rendering engine. We propose a new WIBR method that overcomes the disadvantages of the mesh and produces high-quality warped images by exploiting the idea of separating visibility resolution from reconstruction. Thus it proceeds in two steps: first we determine the samples visible in the desired image and then the desired image is reconstructed from the visible samples. The reconstruction is done with the help of an offset buffer. Separating the two incompatible tasks allows for better control over each of them individually, with the benefit of a very good overall result as it can be seen from the color plate and accompanying video.

Keywords

rendering, image-based rendering, antialiasing

1. Introduction

Rendering by 3D warping of images with depth (WIBR) is promising since one hopes to transfer the quality of a reference image to warped images for several desired views [McMillan95].

However, until now, the only WIBR method that comes close to realizing this hope is the *mesh* method. This method treats the reference image as a mesh of micro-triangles and after the mesh is transformed (warped) it is fed into a polygon-rendering engine. The good quality of the desired view is due to the minute scan-conversion of the micro-triangles. This strategy both benefits *and* requires help from polygon-rendering hardware. Unfortunately, rendering as polygons fails when you're trying to use multiple reference image locations. Triangles interpenetrate and coincide, causing flashing as the viewpoint moves. One solution is to pre-process the data to build a single mesh, but this is not only difficult, but maybe also prohibitive for warping of data acquired in real time.

A commonly used WIBR method is inspired by the *splatting* done in volume rendering [Westover91]. The essence of the WIBR splatting method is to approximate the area of the desired image that is influenced by a warped sample; the area is called a splat. The desired image is the result of applying the splats corresponding to all the warped pixels. Splatting is mainly used when no support from polygon-rendering hardware is available. However, the quality of the warped images quickly degrades as the view changes from the view of the reference image.

With a closer analysis of the splatting method one recognizes that the splats have two distinct tasks that they have to perform simultaneously:

- resolving *visibility*, that is overwriting the samples that are not visible in the desired view
- and *reconstructing* the desired image out of the visible samples

For correct visibility resolution, the splats have to be *opaque* so that they completely overwrite the back-surface samples. For reconstruction purposes however, the splats need to be *semi-transparent* in order to blend the visible samples together. The blending cannot be done before visibility is fully resolved since visible samples become contaminated by back-surface samples, which should have no role in the final image. Also, underestimation of the size of the splat can allow back-surface samples to erroneously appear in the final image. To prevent this, researchers ensured that splat sizes were approximated to excess [Shade98, Rafferty98]. However, splats that are too big worsen the reconstruction of the desired image since they incorrectly erase visible samples. This leads to aliasing of edges and of high frequency textures.

Thus, resolving visibility and reconstructing the desired image are incompatible with each other. *Separating* the two tasks enabled us to develop a WIBR method that produces high-quality images and is amenable to hardware implementation.

2. First visibility and then reconstruction

So the problems that need to be solved are finding the visible samples and properly reconstructing the desired image from them.

Previous work in the domain of antialiasing and reconstruction points out that sixteen or even as little as nine or five quality samples per pixel are enough for a high-quality reconstruction [Molnar91].

The first step is obviously to increase the resolution of the warp buffer so that it can accommodate multiple samples per final-image pixel. We doubled the warp buffer's resolution in each direction, so as many as sixteen samples can participate in generating the color of an output pixel when using a reconstruction / resampling kernel that is two final-image pixels wide.

Note that increasing the resolution of the warp buffer doesn't change the cost of the 3D warping equation. Now, image-based rendering methods based on warping cannot benefit from a large number of images that together sample the scene at a much higher resolution than the desired views, since that would defeat the purpose of warping. Consequently, simply warping the reference image samples does not produce the required number of samples per desired image pixel. The splatting method simplistically replicates the warped sample throughout the coarsely approximated splat. The mesh method overcomes this problem by an expensive scan-conversion of the micro-triangles, done at appropriate higher resolution. Here is our solution.

2.1. Scan-conversion before transformation

We propose creating all the necessary samples *before warping* by interpolating in the reference image both in depth and in color (see Figure 1). The newly created sub-samples are then 3D warped individually, which implicitly computes and scan-converts what used to be the splats of the initial samples for the splatting method or the micro-triangles for the mesh method. Unlike the regular samples, the sub-samples can safely be considered points (constant in shape and size) since they are of sub-pixel size and their squareness doesn't affect the quality of the reconstruction of the final image. The warping of the sub-samples still has to resolve visibility. Under the assumption that the maximum growth in area of the samples through warping has an upper bound, the sub-samples can be recorded with only one write into the warp buffer without any sub-sample splat or micro-triangle evaluation. This is possible by interpolating the reference image at a higher resolution than the warp buffer. The difference in resolutions allows for sample expansion, up to the preset upper limit. We obtained good results when we assumed that original samples couldn't grow through warping more than twice in each direction. This is not as strong of an assumption as it might appear since a good sample (taken by an incident ray close to normal to the surface) doesn't grow more than twice in each direction until the viewpoint gets twice as close to the surface as it originally was. Bigger growths occur when the angle at which a surface is viewed changes from acute to normal. But this is an under-sampling problem rather than a rendering problem and the missing information can only be found in another, appropriate, reference image. With the warp buffer twice as fine as the input and output images (see above) it means that in order to allow for a maximum sample expansion of two we need to interpolate four times in between the original pixels in each direction.

Special care is taken in order to avoid interpolating across depth discontinuities in the reference image, which mark the physical separation between scene surfaces. Interpolating depth between samples of two unconnected surfaces creates false samples that, once warped, will incorrectly trail from one surface to the other. Interpolating color between unconnected samples also produces artifacts. Once a front surface slides apart from the background surface due to parallax, the old silhouette of the front surface incorrectly persists on the background and on the front surface.

The artifacts described above do not occur only when one interpolates in the reference image. Depth images acquired from the real world (using cameras and active (laser) range finders) do not have pure samples at silhouette edges. Both the depth and color of the edge samples are an average of the surface fragments seen by the bundle of rays that gathered that sample. Although the reference images used in this work were synthetic, they are as close to real photographs as the current geometry-based rendering techniques allow (with reasonable time constraints).

Our method of detecting the depth discontinuities is inexpensive but proved to be very robust. The depth images encode the depth information as generalized disparities [McMillan97]. The generalized disparity at a pixel is the length of the ray to the pixel on the image plane divided by the range (the length of the ray to the surface sampled). The observation we made is that *generalized disparities of equidistant, collinear pixels are in a linear progression iff they sample the same planar surface*. The claim can be formally proved relatively easily, so we will just note that the property claimed is not that surprising since the generalized disparity differs from the inverse of z only by a multiplicative constant. We proceed at detecting the depth discontinuities by convolving the generalized disparity map with a second derivative kernel, which exposes large deviations from local planarity in the image. The detected edge samples are eliminated from the reference image, which is not a perfect solution since thin surfaces disappear.

So we have shown how by interpolating in reference image space we provide the samples necessary for reconstruction. The splats are at most a few pixels big so there are no concerns for artifacts due to perspectively incorrect colors caused by screen-space interpolation instead of model-space interpolation. The quality of the samples used for reconstruction depends not only on their color but also on their accurate location in the buffer used to reconstruct. The accuracy of the location of the warped samples depends on the round-off warping errors that are due to coercing warped samples to integer warp buffer coordinates. We virtually eliminate these errors with very little additional cost.

2.2. Offset buffer

We use a few additional bits per warp buffer location to improve the precision of the position of the warped samples. These bits form the *offset buffer* and whenever a sample (sub-sample) is warped they are written with the offset between the actual warping location and the center of the warp buffer location (see Figure 1). In our work we used two bits for the offsets in the u and v direction. Considering that the warp buffer is already twice as refined as the output image, this brings the warping round-off error to within one eighth of a pixel width.

The offset buffer is used at the reconstruction stage. The kernel is still two output pixels wide but instead of storing four weights per row it now stores sixteen weights per row in order to take advantage of the more precise location of the samples. Note that the cost of the convolution with the reconstruction kernel doesn't change much since the number of samples used per output pixel is the same. The only difference is having to compute the sum of the weights used at each pixel, which is the denominator in computing the color of the output pixel as a weighted average.

The offset buffer is a direct consequence of separating visibility from reconstruction: visibility and reconstruction are solved at different resolutions. Reconstructing with the help of the offset buffer is equivalent to reconstructing from a very fine but sparsely populated buffer that however has enough samples for a good reconstruction. Continuing the analogy, the warp buffer / offset buffer combination has the advantage of providing instant access to the populated locations of the sparsely populated buffer, since the warp buffer acts like bins from which the colors can be directly accessed.

3. Everything together: the algorithm

Here is a pseudo-code summary of our rendering algorithm:

```
1. For every desired image
  1.1.  Clear buffers
  1.2.  For every reference image of
  resolution w by h
    1.2.1. Detect depth
    discontinuities
    1.2.2. Interpolate in color and
    depth in the reference image
    creating a x w by a x h sub-
    samples where a is the super-
    sampling factor (e.g. a = 4)
    1.2.3. Warp every sub-sample in a
    warp buffer of resolution b x w by
    b x h where a = b x c and c is the
    maximum expansion through warping
    (e.g. b = 2, c = 2, a = 4)
    1.2.4. Save o-bit offsets in both
    u and v direction in the offset
    buffer (e.g. o = 2)
    1.2.5. Reconstruct/ resample the
    desired image from the warp and
    offset buffers with a 2 x b x o by
    2 x b x o kernel
2. Done
```

4. Implementation and Results

To illustrate our rendering method we used reference images of geometric models. Since the steps 1.2.1 and 1.2.2 of the algorithm do not depend on the current viewpoint and reference images were available ahead

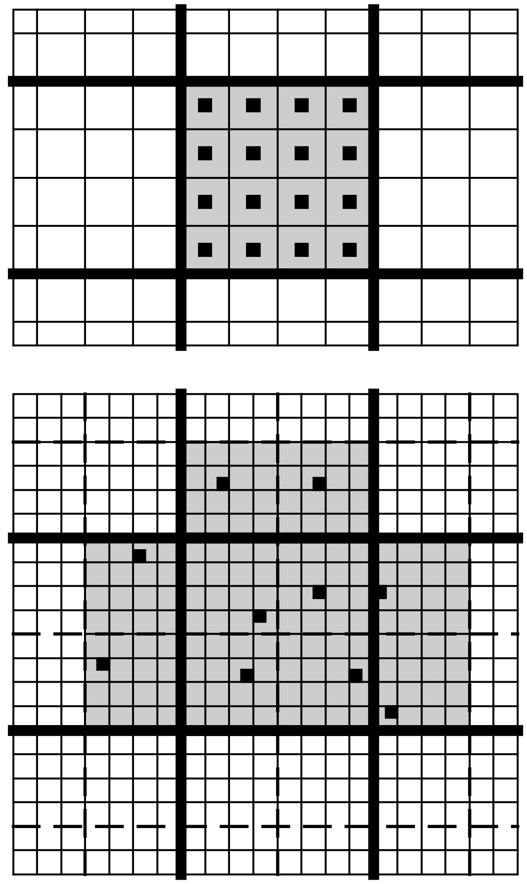


Figure 1. The figure shows a fragment of reference image (top) and a fragment of the output image (down). The thick lines delimit the pixels in both images. The gray-shaded area in the reference image represents an original sample. The reference image is super-sampled by interpolation four times in each direction (see little black squares). Warping the sub-samples is equivalent to computing and scan-converting the corresponding splat (gray-shaded area in the lower grid). The dotted line in the lower grid marks the warp buffer locations and none of them contains more than one warped sub-sample. The offset buffer records with two bits the warping location of the sub-samples inside the warp-buffer location. The most refined grid shows the imaginary refined sparse buffer from which the final image is reconstructed. One can see that for sub-samples that land near the border between two warp-buffer locations the warping round-off error would be significant in the absence of the offset buffer.

of time, they were done as a pre-process. However if a real-time depth images are available they can be done at run-time since they are easy to parallelize.

The accompanying video shows paths rendered offline using our method and played back at 30Hz. For the city model we rendered 600 depth images placed on two regular grids at different altitudes that sandwich the viewpoint path. (The only exceptions are a few extra images needed to sample some of the benches from the train station plaza; those benches were insufficiently sampled by the grid reference images because they are horizontal and at height close to the viewpoint's height.) Every grid node has three images, looking left, straight, and right. At run time the only reference images considered are the ones taken from locations close to the current position of the viewpoint. The images are warped one at a time. In order to avoid the flickering of high frequency surfaces sampled in several reference images, we used a fuzzy z-buffer algorithm that gives preference to the earlier warped images. This is only a partial solution since when the warper switches the set of reference images one can notice an abrupt change.

The warper is not heavily optimized, but it can warp a 720 by 486 image in about one-fifth of a second, in parallel on 32 processors.

The Kamov helicopter was captured with only nine reference images.

The reconstruction of the final image was done using a two-pixel-wide raised-cosine kernel that tapers off completely at the corners.

5. Conclusions and Future Work

Once we decided to treat visibility and reconstruction separately, we gained tight control over each task individually with the benefit of high quality warped images. Visibility and reconstruction are each solved at their respective appropriate resolutions. The offset buffer effectively increases the resolution for reconstruction. The scan-conversion of the splats or micro-triangles is done implicitly by transformation of the reference-image-space interpolated sub-samples.

Separating visibility from reconstruction is new as far as warping is concerned. Geometry-based rendering obviously does separate visibility from reconstruction, since the final image is reconstructed from a super-sampled buffer after visibility is fully resolved. But schemes like the A-buffer [Carpenter84] do not take full advantage of separating visibility from reconstruction since the resolutions at which visibility and reconstruction are resolved are the same: from reconstruction considerations. The resolution is unnecessarily high for visibility. We speculate that the A-buffer scheme can be redesigned and made cheaper using different resolutions for the two tasks.

We were concerned with the quality of the individual samples input to the reconstruction but their relative position is also important. In the future we will investigate jittering the interpolation locations of the samples in the reference images.

Thin surfaces are typically not well sampled in any of the available reference images. A pessimistic approach would be to declare the thin surfaces too close to the maximum frequency that can be represented at the current resolution. However, efforts of trying to reverse the antialiasing of the thin surfaces in order to create some pure samples that can then be blended with the new background surface might be rewarded with interesting results.

This work did not investigate combining several reference images either at run time nor as a pre-process (since the application permits it). Depth images have been pre-combined in order to efficiently eliminate disocclusion errors [Shade98], but the solutions have not been extended for full scenes nor was quality the major concern. We believe that researching the combing of reference images could make possible large-model or natural-scene fly-throughs of uninterrupted high quality.

We will investigate a hardware implementation and believe that it will much simpler than current polygonal renderers that drive the mesh method since the only significant computation is evaluating the 3D warping equation.

6. References

[McMillan95] McMillan L. and Bishop G., "Plenoptic Modeling: An Image-Based Rendering System", *Proc. SIGGRAPH '95*, 39-46 (1995).

[Rafferty98] Rafferty M., Aliaga D., Popescu V. and Lastra A., "Images to Accelerate Architectural Walkthroughs", *Computer Graphics & Applications*, Vol. 18, No. 6, November-December, 38-45 (1998).

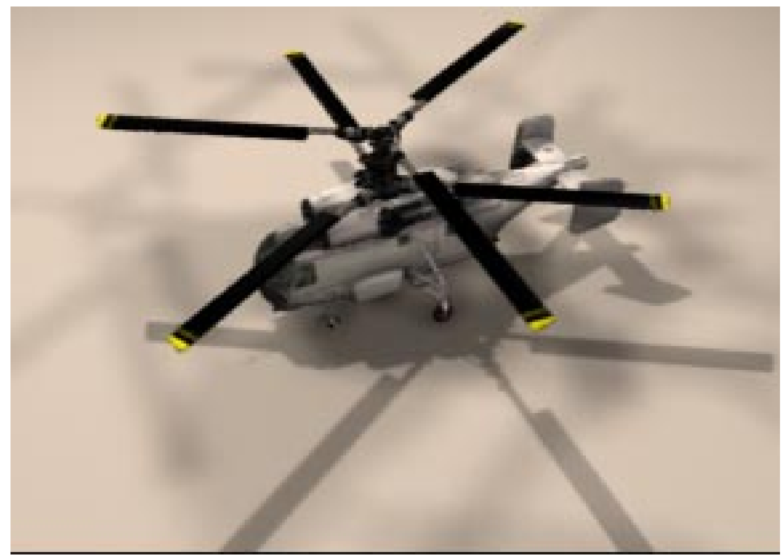
[Shade98] Shade J., Gortler S., He L., and Szeliski R., "Layered Depth Images", *Proc. SIGGRAPH '98*, 231-242 (1998).

[Westover91] Westover L., "SPLATTING: A Parallel, Feed-Forward Volume Rendering Algorithm" *Ph.D., UNC*. (1991)

[Carpenter84] Carpenter L., "The A-buffer, an Antialiased Hidden Surface Method", *Computer Graphics* Vol. 18, No. 3, July, 103-108, 1984

[Molnar91] Molnar S., "Efficient Supersampling Antialiasing for High-Performance Architectures", *TR91-023 University of North Carolina at Chapel Hill*, (1991)



Color Plate
Paper: *"High Quality 3D Image Warping by Separating Visibility from Reconstruction"*
The top two images show the same view of the city model. The left one was generated with our method (it is one of the frames of the path shown in the video) and the right one was generated from geometry with the renderer that was used to generate the reference images, for comparison. The bottom two images are other examples of warped images created with our method.