Interleaved Sampling

Alexander Keller[1], Wolfgang Heidrich[2]

[1] University of Kaiserslautern, keller@informatik.uni-kl.de

[2] The University of British Columbia, heidrich@cs.ubc.ca

Abstract. The known sampling methods can roughly be grouped into regular and irregular sampling. While regular sampling can be realized efficiently in graphics hardware, it is prone to inter-pixel aliasing. On the other hand these artifacts can easily be masked by noise using irregular sampling which, however, is more expensive to evaluate as it lacks the high coherence of a regular approach. We bridge this gap by introducing a generalized sampling scheme that smoothly blends between regular and irregular sampling. By interleaving the samples of regular grids in an irregular way, we preserve the high coherence and efficiently reduce inter-pixel aliasing thus significantly improving the rendering quality as compared to previous approaches.

1 Introduction

Sampling of functions is one of the most fundamental tasks in computer graphics. It has to be performed for applications as diverse as anti-aliased scan-conversion of scenes, illumination from area light sources, cameras with depth-of-field effects, motion blur, and volume rendering. The existing sampling schemes for these applications can be grouped into regular and irregular methods.

On the one hand, sampling on regular grids is efficient and simple, and also well suited for hardware rasterization. Unfortunately, the signals that need to be sampled in computer graphics typically are not band limited and often can contain arbitrarily large frequencies. Therefore some amount of aliasing is unavoidable, and sampling by regular grids emphasizes these artifacts.

On the other hand, Monte Carlo methods use randomized sample positions instead thus masking the aliasing artifacts by noise that is more visually pleasing. However, per pixel irregular sampling does not exhibit the kind of coherence known from regular sampling. Consequently, these methods are more expensive and not well suited for hardware implementations.

The new sampling scheme of *interleaved sampling* allows to smoothly blend between regular and irregular sampling thus yielding the best and most efficient compromise. To this end we interleave samples taken from a number of independent regular grids, and merge them into a single sampling pattern. The relative positions of the regular grids are determined by irregular offsets. Each regular grid can be interpreted as a low resolution image of the scene that can be rendered using the efficient rasterization algorithms. The samples of all regular grids are interleaved such that in the final high-quality, high-resolution image adjacent pixels are not correlated. Only minimal changes to existing hardware algorithms such as the Accumulation Buffer or multisampling allow to exploit the benefits of the new sampling scheme.

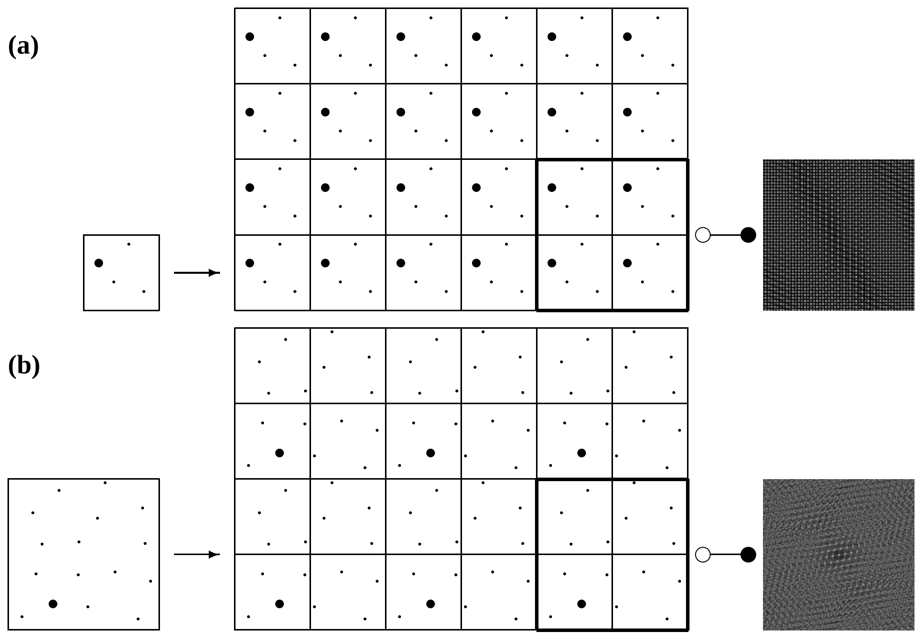

Fig. 1. The principle of interleaved sampling illustrated by example: In **(a)** the classical Accumulation Buffer is illustrated. The same sampling pattern is repeated for all pixels. The Fourier transform of this repeated pattern clearly shows a high potential for inter-pixel aliasing. In **(b)** a 2×2 pixel basis pattern is replicated. Thus the sampling patterns in adjacent pixels are different by interleaved sampling and the Fourier transform reveals a much reduced potential for inter-pixel aliasing. To illustrate the regular grids involved, one sample from the irregular basis pattern is emphasized as well as its corresponding replications. Also note that both approaches require exactly the same number of samples.

2 The Algorithm

The basic idea of interleaved sampling is intriguingly simple and instantly becomes obvious from Figure 1: in the Accumulation Buffer method [HA90] the irregular sampling pattern is affixed to *one pixel* and in consequence periodically repeated for all pixels such bearing a high potential for inter-pixel aliasing (see Figure 1a). For interleaved sampling an irregular offsetting pattern is chosen that covers *multiple pixels*. By periodically repeating this pattern regular grids still persist as emphasized in Figure 1b, however adjacent pixels are sampled by different patterns. Note that although the number of regular grids involved is increasing the inter-pixel aliasing potential is effectively decreased and the total number of samples remains the same.

2.1 Blending between Regular and Irregular Sampling

In a first step an arbitrary irregular sampling pattern is selected. By periodically tiling the image plane, each sample of the irregular basis pattern generates a regular grid. In a second step the pixel resolution is fixed. Visually spoken this means to superimpose the pixel graphpaper onto the periodic sampling pattern. Each regular grid then is rendered using a fast rasterization algorithm, and its samples are accumulated in the pixels determined by the superposition of the graphpaper. Finally each pixel is averaged by a box filter over the number of samples that fell inside.

By interleaving the regular grids by an irregular basis pattern, a high level of coherence is preserved and can still be exploited by raster graphics hardware, yet the high risk of local aliasing, i.e. inter-pixel aliasing, is reduced drastically since adjacent pixels use different sampling patterns.

Note that this general interleaving scheme does not require an integer ratio of the number of pixels covered by the irregular basis pattern as in the example of Figure 1. Depending on the irregular basis pattern and the graphpaper resolution it may also happen, that the pixels have different oversampling rates. The ratio of the support of a pixel and the support of the irregular basis pattern determines the amount of correlation. We can blend between two limit cases (see also Figure 4): In the first the basis pattern consists of only one sample, which results in correlated sampling on one regular grid. For the second case the size of the basis pattern is chosen as large as the image graphpaper which corresponds to the completely uncorrelated case where no regular grid structure is present.

2.2 Theoretical Considerations

The method of dependent tests [Sob62, FC62] was introduced as a generalization of the Monte Carlo method of integration. The basic idea was to use the *same* set of samples for the estimation of multiple integrals instead of using a new realization of the set of samples for each integral.

The Accumulation Buffer is the best example for the realization of the method of dependent tests: the same sampling pattern is used for all pixel integrals. However due to the repetition of the sampling pattern for each pixel, adjacent pixels can be highly correlated causing inter-pixel aliasing (see the periodogram in Figure 1a). This effect is not observed in the uncorrelated approach as taken by e.g. a software ray tracer that is applying a different sampling pattern to each pixel. The specific sampling pattern for the method of dependent tests can be chosen from a wide variety of different methods (for a survey see [Gla95]). A summary of the approaches that have been used for graphics hardware can be found in a report by Goss and Wu [GW99]. We use samples with blue noise characteristics that perfectly tile periodically and are generated by a relaxation scheme similar to the one introduced in [MF92].

By interleaving the method of dependent tests adjacent pixels are sampled in an uncorrelated way. The periodicity by the replication of the irregular basis pattern is much reduced as compared to the per pixel approach. Thus the inter-pixel aliasing potential is attenuated, and lower sampling rates as compared to the Accumulation Buffer can be used. This partially was recognized in [DW85], however without the awareness of the method of dependent tests and consequently without exploiting the coherence induced by the grid structure. Interleaved sampling is a generalization of the method of dependent test considering spectral issues.

3 Applications

Interleaved sampling as introduced above can be applied to a number of different sampling tasks, in particular those for which the Accumulation Buffer [HA90] has been used before. This includes soft shadows based on the method by Brotman and Badler [BB84], limited depth-of-field [HA90, KHM95, HSS97], and Instant Radiosity [Kel97].

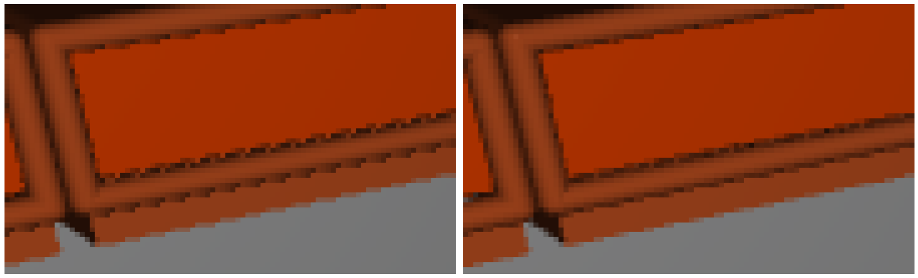

Fig. 2. On the left we see the standard Accumulation Buffer approach for anti-aliasing at 4 samples per pixel as depicted in Figure 1a. Using the interleaved approach (see Figure 1b) on the right, the aliasing artifacts are spatially separated yielding a more visually pleasing appearance.

In addition to these applications, we will also consider interleaved sampling for texture based volume rendering along the lines of Cabral et al. [CCF94].

3.1 Anti-Aliasing by Supersampling

The first application we will look at is supersample anti-aliasing along the lines of multisampling [Ake93] and the Accumulation Buffer [HA90]. Existing multisampling implementations can be interpreted as a special case for interleaved sampling. If we consider only the i^{th} sample of every pixel in an image, we see that all these samples are located on a regular grid with a spacing that is identical to the spacing of the pixels (see the emphasized sample in Figure 1a). Therefore, efficient incremental scan-conversion algorithms can be applied on this grid for rasterizing geometric primitives.

With interleaved sampling we stretch out the grid spacing of the samples so that they are no longer identical to the pixel spacing. Therefore, every sample on a particular grid corresponds to a different sub-pixel position in one of the pixels (see the emphasized sample in Figure 1b). Stretching out the grid spacing also stretches out aliasing structures thus very much improving the visual quality as can be seen in Figure 2.

There are different ways of implementing interleaved sampling in hardware rendering applications:

- **Modification of Multisample Hardware.** The most generally applicable way would be to add direct hardware support for interleaved sampling in a similar fashion multisampling is supported today. The hardware for scan conversion does not need to be modified for this. The only difference to multisampling is that not all the pixels will have the same sub-pixel masks, but the mask is one of a predetermined and implementation-specific number of masks. With these modifications the geometry only needs to be processed once, while individual scan converters take care of rasterizing the regular grids.
  The relative positions of the regular grids, which also determine the masks for the individual pixels, would be an implementation-specific arrangement that could be wired into the hardware. These positions would be generated in the design phase of the hardware. Since interleaved sampling is composed of regular grids, the hardware implementation will be less expensive than the 3Dlabs Wildcat multisample extension [3Dl], which randomizes the sample selection on a higher resolution grid.
- **Implementation on Current Hardware.** Another way to implement interleaved

sampling is to individually render lower resolution images corresponding to the regular grids, and to then interleave the samples obtained this way by hand. To this end, the individual regular grids (low resolution images) are rendered directly into a texture map (an operation that is supported on many contemporary platforms). With these regular grids as textures, we can then render full-screen polygons while a second texture or a screen-space stipple pattern is used to mask out the pixels that should not be affected by the regular grid.

If the whole application is bounded by fill rate, then rendering multiple copies of the geometry will not degenerate the rendering performance. In this case, the method introduces only a constant overhead for interleaving the samples after the low-resolution regular grids have been rendered. The cost of this process does not depend on the shading complexity of every sample in the image. Thus, if the application is limited by the rasterization performance, this algorithm has only a constant overhead over supersampling with the same sampling pattern for each pixel. It is therefore a feasible algorithm that is interesting for applications that have high shading costs for each individual sample, such as complex procedural shaders [POAU00], or volume rendering (also see Section 3.3).

3.2 Motion Blur

The basic concept of interleaved sampling can easily be extended to arbitrary dimensions. For example, by simply assigning each of the interleaved images one moment in time, the above algorithm can support motion blur: then the irregular basis sampling pattern is 3-dimensional, where the first two dimensions are used for interleaving and the third dimension specifies the moment in time. Interleaved sampling so enables efficient and correct motion blur simulation for e.g. the REYES architecture [CCC87] or the photon map [JC98] using a very small set of different time samples (i.e. just the samples in the basis pattern), where each image to be interleaved is rendered by the original algorithm, however for the specified moment in time.

In Figure 4 the results of a motion blur simulation at 16 samples per pixel are shown. Using uncorrelated random samples no aliasing artifacts are visible, whereas the standard correlated Accumulation Buffer technique exhibits visible artifacts. Using a 2×2 interleaved irregular basis pattern of 64 samples with blue noise characteristics (yielding 16 samples per pixel as in the other cases), we blend between uncorrelated and correlated sampling and are able to exploit rasterization on regular grids. Despite the low sampling rate the aliasing artifacts are clearly reduced at the same amount of sampling work.

3.3 Volume Rendering

The next application we consider is texture-based volume rendering as described by Cabral [CCF94]. This algorithm uses a simple emission-absorption model for the volume, so that the rendering is reduced to a 1D integration along the viewing ray for every pixel in the final image. The texture-based approach for computing this 1D integral works as follows: the volume is sliced with equidistant planes. These are sorted back-to-front and blended together in the framebuffer. As a result of this algorithm, we obtain a sampling pattern that corresponds to a regular 3D grid in clip coordinates (i.e. after the projective transform).

Typically, on the order of 100 slices are used for this kind of application, and each slice yields a polygon that covers a large portion of the screen and is textured with a

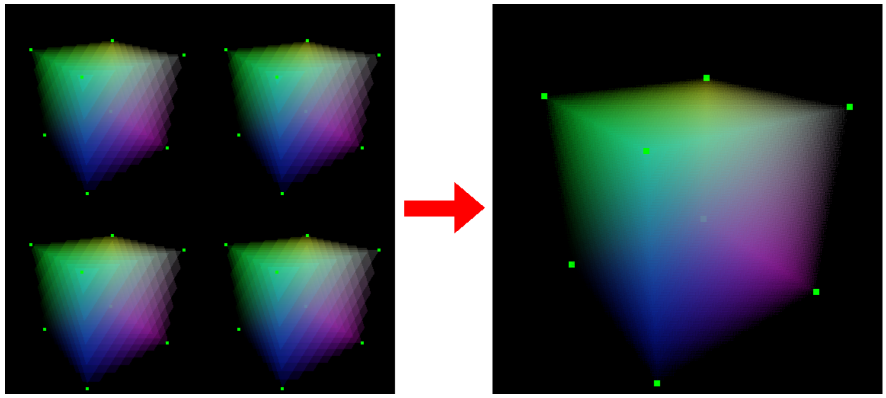

Fig. 3. For volume rendering, we combine a collection of small resolution images with slightly offset sample planes into a large resolution image by interleaving the samples from the smaller images. Except for a final combining operation, the fill rate required is the same as for rendering a large image in the first place, but the rendering quality is much better.

3D texture with tri-linear interpolation. Thus, the fill-rate required by this method is quite high, while the cost of geometric transformations is negligible. But even with 100 slices, there are often still some serious aliasing artifacts left. After all, modern medical imaging devices can generate data sets in the order of 512^3, and therefore 100 slices would result in an undersampling of a factor of 5. On the other hand it is not possible to increase the number of slices arbitrarily, both because of the performance implications and because of the limited depth of the framebuffer that results in quantization artifacts when using alpha blending. This could be avoided by hierarchical summation, which however requires additional framebuffers for the intermediate results and in consequence cannot be realized efficiently.

Interleaved sampling can be used to improve on this situation. Since we do not have the fill rate to do supersampling at interactive rates, we are only going to interleave the samples in one dimension; the z-coordinate in camera space. Interleaving in that dimension corresponds to varying the offsets for the samples in z direction. These offsets will be different for adjacent pixels, but since we are using interleaved sampling, they will repeat after a number of pixels, thus providing us with the coherence we need for exploiting graphics hardware. The method can be considered as a hardware implementation of the equidistant ray marcher from [PKK00] with interleaved jittered offsets.

Using existing hardware, the method then proceeds similarly to the method described in Section 3.1. First, we use the traditional texture-based volume rendering algorithm [CCF94] to render lower resolution regular grids (i.e. images), each with the sampling planes slightly offset compared to the other regular grids. The cost of this operation is identical to the original texture-based volume rendering algorithm, since volume rendering is completely determined by the rasterization cost.

As described in Section 3.1, we then combine the low-resolution regular grids into a high-resolution image using stipple patterns or 2D textures to mask out the pixels that are not affected by the individual regular grid. Figure 3 illustrates this method for the configuration with 4 regular grids. The cost of this second step is constant, that is, it does not depend on the number of sampling planes. During our tests with an SGI Octane VPro, we found this constant cost to be about the same as rendering 8 additional planes with the original method. The quality improvements we can achieve with interleaved sampling are much higher than the improvements we get by adding this number of additional samples.

Figure 5 shows a comparison of the algorithm with and without interleaved sampling for 20, 60, and 120 sampling planes. Since we are not supersampling in this application, interleaved sampling replaces the aliasing effects with noise that resembles the patterns known from ordered dithering. Note, however, that we are jittering sample locations instead of dithering color values.

On an Octane VPro, the frame rate for the 20 plane version is > 30 frames per second both with and without interleaving. For the 120 plane version we get around 18 frames per second in the interleaved case, and 20 frames per second in the non-interleaved case. Note that even for the 120 plane version there is still some aliasing left, and the 60 plane version with interleaved sampling looks better than the 120 plane version without. Also note that renderings with 120 sample planes already exhibit quite serious quantization artifacts due to the limited depth of the framebuffer. Thus, further increasing the number of sampling planes will not improve the quality.

Some more comparisons between the traditional volume texture-based volume rendering and interleaved sampling can be found in Figure 6.

3.4 Other Applications

The basic techniques of interleaved sampling presented in this paper naturally generalize to all Accumulation Buffer [HA90] and multi-pass frame-buffer [POAU00] techniques while preserving the good spectral properties and the fast convergence. So it is straightforward to apply interleaved sampling to weighted sampling [HA90], to the simulation of extended light sources using deep shadow maps [LV00], and to global illumination simulations [Kel97, HDKS00]. Here interleaved sampling can replace the N*-shadows-artifact* by noise. Finally it is very appealing to apply interleaved sampling to CCD-chip designs.

4 Conclusion

We have presented a new sampling scheme called *interleaved sampling* that exhibits excellent spectral behavior and yields excellent quality especially at low sampling rates, while at the same time it is simple and efficient to implement.

The fundamental idea of the new method is to interleave samples from several regular grids to form a locally irregular sampling pattern. By stretching out the periodicity of the per-pixel sampling pattern, aliasing structures are spread out, too, and in consequence much less perceivable. The high coherence of the involved regular grids allows for very efficient implementations in software, and in new as well as current hardware. The feasibility of interleaved sampling has been demonstrated by a number of examples ranging from supersampling over motion blur to volume rendering.

Acknowledgements

For the image in Figure 6 we used an MR data set kindly provided by Christoph Rezk-Salama, University of Erlangen-Nürnberg. Part of this work was done in August 2000, while both authors stayed at the Max-Planck-Institut für Informatik in Saarbrücken, Germany. We would like to thank Hans-Peter Seidel for granting the research stay of the first author, who was also supported by mental images, Germany.

References

3Dl. 3Dlabs, *Wildcat: SuperScene Antialiasing*, http://www.3dlabs.com/product/technology.

Ake93. K. Akeley, *RealityEngine Graphics*, Computer Graphics (SIGGRAPH '93 Conference Proceedings), 1993, pp. 109–116.

BB84. L. Brotman and N. Badler, *Generating Soft Shadows with a Depth Buffer Algorithm*, IEEE Computer Graphics and Applications **4** (1984), no. 10, 71–81.

CCC87. R. Cook, L. Carpenter, and E. Catmull, *The Reyes Image Rendering Architecture*, Computer Graphics (SIGGRAPH '87 Conference Proceedings), 1987, pp. 95–102.

CCF94. B. Cabral, N. Cam, and J. Foran, *Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware*, 1994 Symposium on Volume Visualization, October 1994, pp. 91–98.

DW85. M. Dippé and E. Wold, *Antialiasing through Stochastic Sampling*, Computer Graphics (SIGGRAPH '85 Conference Proceedings), 1985, pp. 69–78.

FC62. A. Frolov and N. Chentsov, *On the calculation of certain integrals dependent on a parameter by the Monte Carlo method*, Zh. Vychisl. Mat. Fiz. **2** (1962), no. 4, 714 – 717, (in Russian).

Gla95. A. Glassner, *Principles of Digital Image Synthesis*, Morgan Kaufmann, 1995.

GW99. M. Goss and K. Wu, *Study of Supersampling Methods for Computer Graphics Hardware Antialiasing*, Tech. report, HP Labs, 1999.

HA90. P. Haeberli and K. Akeley, *The Accumulation Buffer: Hardware Support for High-Quality Rendering*, Computer Graphics (SIGGRAPH '90 Conference Proceedings), 1990, pp. 309–318.

HDKS00. W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel, *Illuminating Micro Geometry Based on Precomputed Visibility*, Computer Graphics (SIGGRAPH '00 Conference Proceedings), 2000, pp. 455–464.

HSS97. W. Heidrich, Ph. Slusallek, and H.-P. Seidel, *An Image-Based Model for Realistic Lens Systems in Interactive Computer Graphics*, Graphics Interface '97, May 1997, pp. 68–75.

JC98. H. Jensen and P. Christensen, *Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps*, Computer Graphics (SIGGRAPH '98 Conference Proceedings), 1998, pp. 311–320.

Kel97. A. Keller, *Instant Radiosity*, Computer Graphics (SIGGRAPH '97 Conference Proceedings), 1997, pp. 49–56.

KHM95. C. Kolb, P. Hanrahan, and D. Mitchell, *A Realistic Camera Model for Computer Graphics*, Computer Graphics (SIGGRAPH '95 Conference Proceedings), 1995, pp. 317–324.

LV00. T. Lokovich and E. Veach, *Deep Shadow Maps*, Computer Graphics (SIGGRAPH '00 Conference Proceedings), 2000, pp. 385–392.

MF92. M. McCool and E. Fiume, *Hierarchical Poisson Disk Sampling Distributions*, Proceedings of Graphics Interface '92, May 1992, pp. 94–105.

PKK00. M. Pauly, T. Kollig, and A. Keller, *Metropolis Light Transport for Participating Media*, Rendering Techniques 2000 (Proc. 11th Eurographics Workshop on Rendering), Springer, 2000, pp. 11–22.

POAU00. M. Peercy, M. Olano, J. Airey, and J. Ungar, *Interactive Multi-Pass Programmable Shading*, Computer Graphics (SIGGRAPH '00 Conference Proceedings), 2000, pp. 425–432.

Sob62. I. Sobol, *The use of w^2-distribution for error estimation in the calculation of integrals by the Monte Carlo method*, U.S.S.R. Comput. Math. and Math. Phys. **2** (1962), 717 – 723.

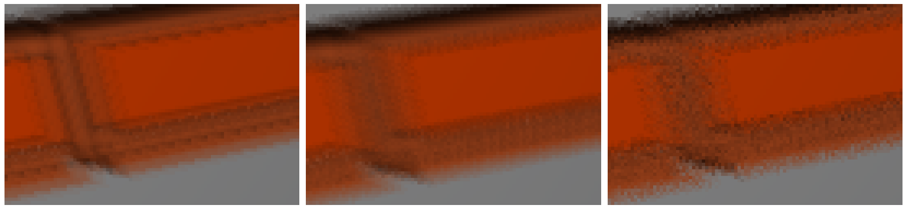

Fig. 4. Blending between regular and irregular sampling for the example of motion blur. In all images we use 16 samples per pixel. From left to right: method of dependent tests (Accumulation Buffer), interleaved sampling, and for illustration independent random samples.

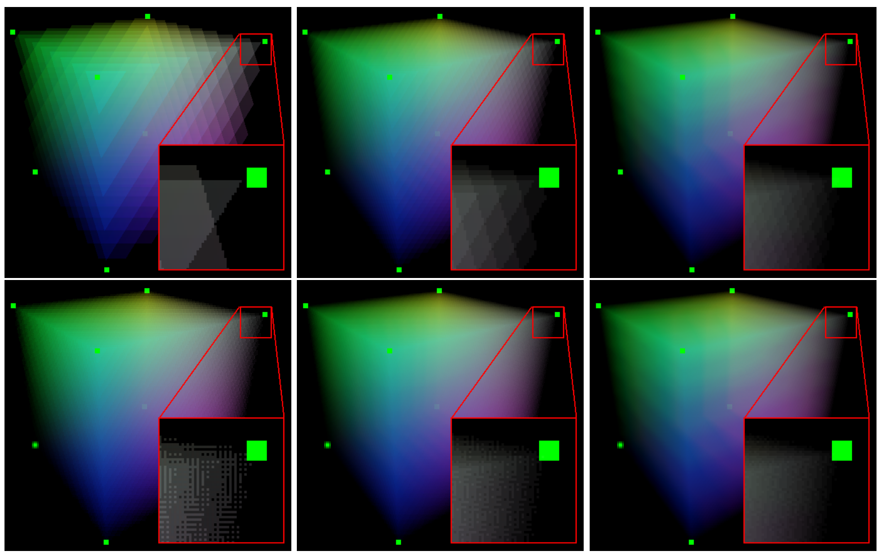

Fig. 5. A comparison of interleaved sampling for volume rendering (bottom) with the traditional texture-based approach described by Cabral et al. [CCF94] (top). From left to right: 20, 60, and 120 planes. Note that in the 120 plane example quantization artifacts due to the limited dynamic range of the framebuffer start to appear. To avoid these artifacts, smaller numbers of planes have to be used, in which case interleaved sampling produces much better results than the original algorithm.

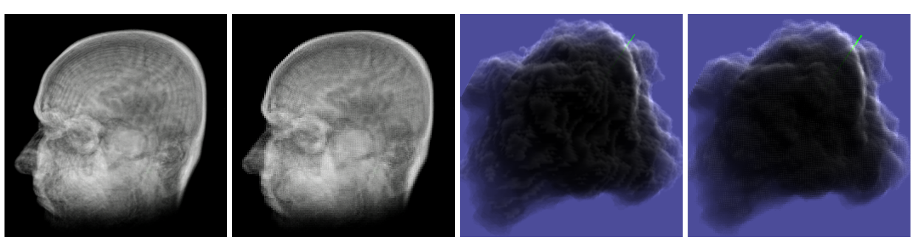

Fig. 6. Illustration of our volume rendering algorithm with real data sets. Left of pair: traditional texture based method exhibiting aliasing artifacts that show up as ringing structures. Right of pair: interleaved sampling with drastically reduced depth aliasing.