LA-UR- 09-08318

*Approved for public release;*
*distribution is unlimited.*

| | |
|---|---|
| *Title:* | Physically-Based Interactive Schlieren Flow Visualization |
| *Author(s):* | Carson S. Brownlee, University of Utah<br>Vinccent Pegoraro, University of Utah<br>Siddharth Shankar, University of Utah<br>Patrick S. McCormick, LANL, CCS-1<br>Charles D. Hansen, University of Utah |
| *Intended for:* | IEEE Pacific Visualization 2010<br>Taipei, Taiwan<br>March 2-5, 2010 |




Form 836 (7/06)

# Physically-Based Interactive Schlieren Flow Visualization

C. Brownlee, University of Utah
V. Pegoraro, University of Utah
S. Shankar, University of Utah
P. McCormick, Los Alamos National Labs
C. Hansen, University of Utah

**ABSTRACT**

Understanding fluid flow is a difficult problem and of increasing importance as computational fluid dynamics produces an abundance of simulation data. Experimental flow analysis has employed techniques such as shadowgraph and schlieren imaging for centuries which allow empirical observation of inhomogeneous flows. Shadowgraphs provide an intuitive way of looking at small changes in flow dynamics through caustic effects while schlieren cutoffs introduce an intensity gradation for observing large scale directional changes in the flow. The combination of these shading effects provides an informative global analysis of overall fluid flow. Computational solutions for these methods have proven too complex until recently due to the fundamental physical interaction of light refracting through the flow field. In this paper, we introduce a novel method to simulate the refraction of light to generate synthetic shadowgraphs and schlieren images of time-varying scalar fields derived from computational fluid dynamics (CFD) data. Our method computes physically accurate schlieren and shadowgraph images at interactive rates by utilizing a combination of GPGPU programming, acceleration methods, and data-dependent probabilistic schlieren cutoffs. Results comparing this method to previous schlieren approximations are presented.

**Index Terms:** [Scalar Field Data, GPUs and Multi-core Architectures, Flow Visualization.]: —

## 1 INTRODUCTION

Recent advances in CFD have produced a wealth of simulated flow data [9]. Understanding these flows is of great importance for applications ranging from aircraft design to combustion analysis [13]. A range of techniques have been developed for understanding these flows both computationally and experimentally [19]. Some of the common experimental methods include dye injection and photographic techniques such as schlieren photography that can provide insight into local and global flows respectively. Producing these images in the laboratory setup can be expensive and time consuming due to the complex optics.

Recreating these experimental techniques computationally with the simulated physical constraints presents scientists used to schlieren photography a familiar and intuitive visualization. Conversely, replicating these systems on the computer allows additional degrees of control in the visualization that would be difficult or impossible due to the physical configuration of experiments. This freedom allows for useful features such as displaying silhouettes around edges or selectively culling ranges in the data. While methods have been developed for approximating schlieren images without refracting light [20, 18], they are not well suited for all data sets, such as shock waves or mixed materials with large changes in refractive indices which results in light paths diverging from linear approximations.

In this paper, we present a novel technique for generating schlieren and shadowgraph images by tracing light paths through time-varying scalar fields of computed flows. Calculating light refracting through a flow presents a number of challenges. Light paths must be recomputed whenever the viewpoint changes thus an interactive method for determining them at each frame is presented. Graphics hardware is used to: trace refraction through inhomogeneous datasets, employ acceleration structures for adaptively sampling data, computationally replicate schlieren cutoffs, and filter out noise. By utilizing these techniques we can simulate realistic light transport through a flow at interactive rates. To our knowledge this is the first technique to computationally replicate schlieren images by generating refractive light paths at interactive rates.

After describing an overview of the experimental setup and the related work in section 2, we provide an overview of our method in sections 3, 4, 5, and 6. A description of the quality of the images and performance are then given in section 7. Finally, we end the paper with ideas for further extensions to our method in section 8.

## 2 RELATED WORK

We draw upon the great body of work in the fields of experimental schlieren and shadowgraph photography as the basis for our work. Our method improves upon previous work on interactive schlieren and shadowgraph visualization by tracing curved light paths rather than relying on line of sight approximations. In order to accomplish this task we build upon previous work in computer graphics literature.

Shadowgraph techniques have been used for centuries to look at flows that are not visible to the human eye such as heat dissipation or shock waves [13]. The idea is that small changes do not scatter light to a large degree but it was noticed that shining a bright light through them will produce a clear image of the flow by looking at the shadows formed from light refraction. In a shadowgraph system refracted light is imaged on a film plane. Figure 1 shows the optical setup of a typical shadowgraph system. A light source is filtered through a slit apparatus thus producing a small point light source. Nearly parallel rays are sent through the test area and focused onto a film plane. Light that was refracted in the test area will group together to produce bright areas in the film plane or disperse and create darker regions. Figure 3(a) shows light and dark regions surrounding a gunshot from an AK-47 as regions of less dense air refract light forming a bright fringe around features in the data. Shadowgraphs only look at changes in the second derivative and are a poor indicator of the amount or direction of refraction. If all rays were refracted the same amount in the same direction then the resulting image would be identical to a translated image of no refraction at all. Schlieren photographic techniques provide additional information by introducing a one dimensional cutoff that shifts intensity values based on the amount and direction of displacement at the focused cutoff region. In figure 2, light rays traverse the flow from a light source similarly to the shadowgraph setup. In the schlieren system the light source is then refocused in a small area and a cutoff is inserted to reduce light from the light source. In figure 2 a vertical knife-edge is inserted at the center of the re-focused light source. If no light is refracted then the knife-edge reduces the light source by half resulting in a gray image, whereas if light is refracted shifts in the focused image of the original light source cause more or less of the focused light to be blocked by the cutoff. If the focused image is shifted down the resulting region is darker and if shifted up then more of the original light gets through to the

film plane. A knife-edge cutoff thus provides information about the amount of light shifted along a single axis. Another common type of cutoff is a circular cutoff that shades the image based on the amount of displacement without the directional information of the knife-edge. Color filters can also be used as a cutoff to produce colors based on the direction of displacement. An illustration of a color filter is shown in figure 4. Whereas a knife-edge cutoff only gives information about the amount of displacement along one axis, color can give two dimensional information about the direction of displacement. Figure 3(b) demonstrates how a color filter cutoff emphasizes gradations in shock waves resulting from a gunshot compared to a similar shadowgraph image 3(a).

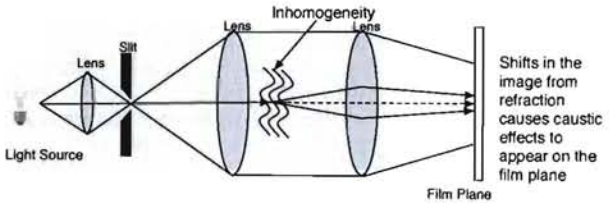



Figure 1: 2D illustration of the shadowgraph optical setup.

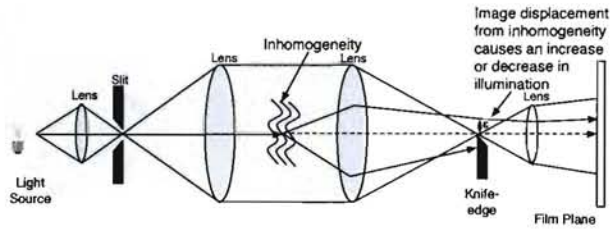



Figure 2: 2D illustration of the schlieren optical setup.

Computational schlieren images of three dimensional fluid flows have been computed non-interactively using a ray tracing method by Anyoji et al. [1, 16]. Such techniques produce an accurate image but are not ideal for data exploration. A non-photorealistic method for producing schlieren-like images using line of sight ray traversals for visualization was recently introduced [18] but without refracting light. In order to reproduce an accurate physically-based representation, tracing non-linear light trajectories is necessary for flows with large variations in refractive index such as shock waves or flows with multiple materials. Ray tracing also allows for the reproduction of the optics used in an experimental setup. The inverse of the problem was achieved by Atcheson et al. [2] by using schlieren photographs to compute a three-dimensional scalar field.

Algorithms for computing caustics have been developed over the past two decades in the computer graphics community. Photon maps were originally introduced as a method for computing caustic and global illumination effects offline [7]. Photon maps were later extended to volumetric photon mapping to compute scattering effects and caustics through inhomogeneous media [8, 5]. Although these offline methods are not directly applicable to our work, they present filtering techniques for reducing noise in regions of low photon density as well as equations for computing light paths. Tracing light refraction through volumes at interactive rates was introduced with Eikonal rendering which relied on pre-computing wavefront propagation through a grid [6]. Eikonal rendering relies on a long pre-computation step that isn't feasible for schlieren systems where the light source changes relative to the volume whenever the camera rotates. Sun et al. presented a technique [17] that calculated

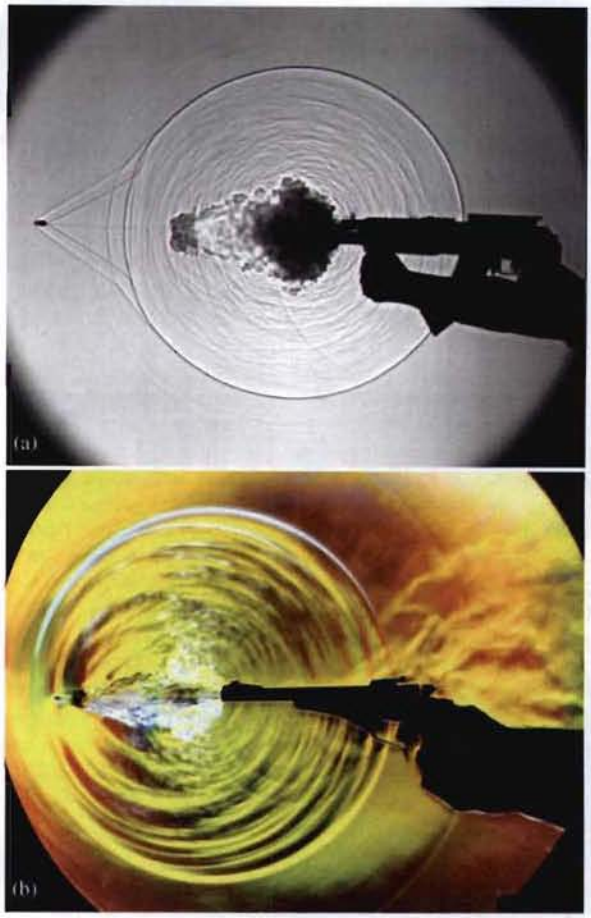



Figure 3: (a) A shadowgraph photograph of an AK-47 (Courtesy of G.S. Settles). (b) A schlieren photograph of a gunshot with a color filter applied (reproduced from [14] with permission).

single-scattering effects through a volume. Viewing rays were then computed as a separate pass for interactive light refraction. In a typical schlieren setup the film plane is directly facing the light source, so having a separate pass for computing light scattering and viewing rays is unnecessary. Scattering effects can play a role in some flows but we focus on purely refractive media such as air.

## 3 Computational Schlieren & Shadowgraph Imaging

Our method for computing schlieren images relies on a number of acceleration techniques for tracing photons through inhomogeneous media. The overall series of steps used by our rendering pipeline are presented in figure 5. The pre-computation steps utilize the CPU while the image generation and rendering stages are done on the GPU using CUDA [11] which gives us the flexibility to arbitrarily store array values (a scatter operation) without relying on the framebuffer. This is important for our technique as the final photon positions can not be predicted. The ray casting algorithm is ideally suited for the GPU since each ray can run concurrently in its own thread and data locality can be exploited from nearby rays. This coherency benefits from CUDA's single instruction multiple thread (SIMT) architecture as many threads operate on the same data. The parallel nature of the computation benefits from the GPU's parallel architecture as long as the data can be stored on chip. CUDA's OpenGL interoperability also allows us to filter the resulting image and display to the screen without copying it back to the host CPU.

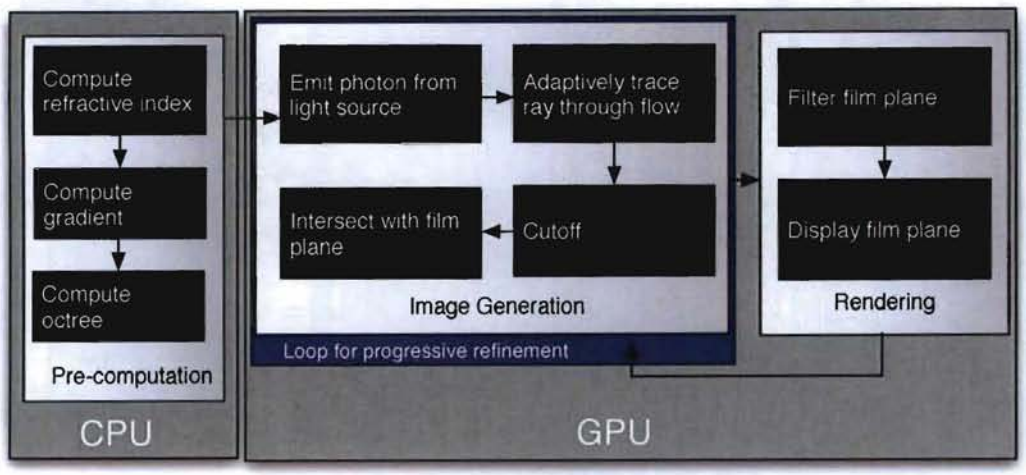



Figure 5: Illustration of the rendering pipeline.

Figure 4: A typical color filter used in schlieren optical setups.

## 4 Pre-computation

The pre-computation stages are required to compute the refractive indices and gradient from related fields as well as construct an octree to accelerate ray traversals. These stages allow accurate and fast computations of light refraction through the flow at later stages of the pipeline.

### 4.1 Computing the Refractive Index

In order to accurately simulate a schlieren photograph it is important to use correct indices of refraction. The indices of refraction in a medium can be computed from a combination of several other scalar fields such as temperature, pressure, and humidity using Ciddor's method [4]. Ciddor's method has been adopted by the International Association of Geodesy (IAG) as a standard method for computing index of refraction. It will not be reproduced here due to the complexity of the method but is explained in more detail in Appendix A. Figure 6(a) shows a heptane data set with indices of refraction computed from pressure and temperature fields. The resulting time-varying scalar fields of refractive indices, $f$, will later be used for computing light paths through the flow. The gradient of $f$, $\nabla f$ is also computed as a pre-processing step using finite differences.

The Gladstone-Dale relation provides a method for computing the refractive indices from density fields [10]. Because we are working with gasses we use the abbreviated form:

$$n - 1 = Kp. \tag{1}$$

$K$ defines the Gladstone-Dale constant, $p$ the density and $n$ is the refractive index we want to compute. For data with more than one material type, the Gladstone-dale constant will need to be interpolated between the different materials using a mixture fraction field. In a simple case we varied the constant for pure air to pure helium based on a provided mixture fraction. $K$ varies by temperature and wavelength but with the temperature around 290 Kelvin and assuming our light has a constant wavelength of $0.633\mu m$ we then know that $K_h$ for helium is approximately $0.196cm^3/g$ and $K_a$ is $0.226cm^3/g$ for air. If $m_h$ is the amount of helium in the mixture and $m_a$ is the amount of air then $n$ can be found by

$$n = (K_h * m_h + K_a * m_a) * p + 1. \tag{2}$$

### 4.2 Octree

Many flow datasets contain large regions of nearly homogeneous refractive indices but only changes in the refractive index are of interest to schlieren and shadowgraph imaging. A computational schlieren system can attain significant speedup by utilizing space-skipping techniques similar to empty space-skipping commonly employed in volume rendering as shown by Sun et al. [17]. Instead of skipping over empty-space in the data, however, we compute regions of nearly homogeneous refractive indices in the data which determine how big of a step through the data can be taken before reaching a significantly large change in refractive index.

The octree is computed as a min-max octree with a tolerance value $t$ that determines the level of each region and thus the size of the area that can be skipped over. Only the octree level values, $j$, are stored in the resulting 3D texture which has the same dimensions as $f$. The min-max octree structure is built to determine how large the homogeneous regions are but no intermediary nodes are stored in a texture so that lookups into the acceleration structure will not require a tree traversal. When traversing through the data, a lookup into the texture will return the octree level for a sample. For example, if a lookup returns a level $j = 2$ then the homogeneous region is of size $2^2$ times the texel size and this entire region can be skipped without encountering a refractive index value that is more than $t$ from the current voxel's refractive index. This behavior is illustrated in figure 7 which shows two different rays which lie in different levels of the octree.

## 5 Schlieren and Shadowgraph Image Generation

The image generation stage computes light paths from the light source to the film plane. This process starts with generating parallel rays from the light source which are then traversed through the refracting flow using a pre-computed gradient and the acceleration

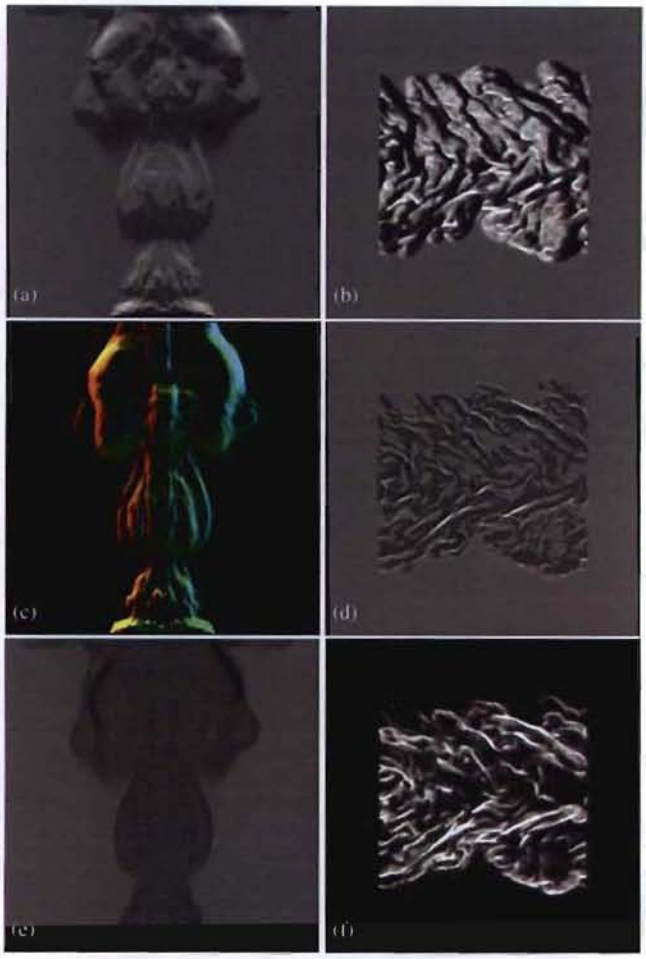



Figure 6: A heptane dataset rendered using refractive indices calculated from temperature and pressure with (a) a knife-edge cutoff, (e) a circular cutoff, and (c) color filter. (d) A simulated combustion dataset rendered as a shadowgraph image and (b) using a schlieren knife-edge cutoff to enhance the flow gradation and (f) using a complemented circular cutoff.

structure discussed in section 4.2. Finally, the rays are weighted by a cutoff for a schlieren image and projected to the film plane.

### 5.1 Emitting Photons from the Light Source

Photons are emitted along a grid to simulate the light source. Ideally the rays are parallel, however the behavior of any given optical setup can be replicated by making modifications to the ray tracer. The system relies on progressive rendering to show increased detail over time. Banding effects from the volume can be smoothed by using jittered sampling to alter the starting positions of rays. The cost of computing three random numbers for jittered sampling for each ray at each pass is prohibitive. Instead, an array of random floats are precomputed. This array can be any size, however for this paper an array that is three times the size of the image is used so that each thread can access three different random numbers. At the start of each rendering pass, only three random numbers are generated and passed to all threads. Each thread then adds these numbers to their thread ids to obtain a unique lookup into the pre-computed array of random floats. Thus, the system only needs to generate three random numbers at each pass instead of thousands or millions.

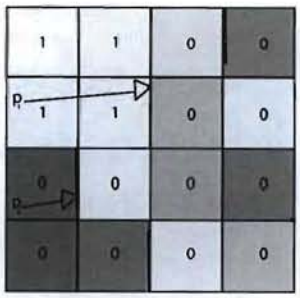



Figure 7: An illustration of a traversal through the octree. $P_1$ and $P_2$ are two rays traversing through the flow. $P_1$ is in a homogeneous region of the data and in a cell of the octree texture that will report a level number of 1 allowing $P_1$ to skip to the edge of that level. $P_2$, on the other hand, is at the lowest level of the acceleration structure and will only traverse to the next voxel.

### 5.2 Adaptively Tracing Rays through the Flow

Photons typically trace curved paths through a medium with spatially varying indices of refraction. Although the trajectory can be approximated using Snell's law, which is intended for refraction through discrete surfaces [12], it may produce undesirable artifacts when used to compute a ray moving through a compressible gas with no discernible surface. Snell's law also requires a significant amount of floating point arithmetic in three dimensions. In contrast, the ray equation of geometric optics based on Fermat's principle presents a very fast and accurate approximation of the ray curve $x(s)$ through inhomogeneous materials [3].

$$\frac{d}{ds}\left(f\frac{dx}{ds}\right) = \nabla f. \tag{3}$$

In order to simulate $x(s)$ this equation is discretized using piecewise linear approximations. The position $x_i$ is updated according to ray direction $v_i$, the refractive index f, and the step size $\Delta s$. The direction is updated according to the gradient of the scalar field of refractive indices $\nabla f$.

$$x_{i+1} = x_i + \frac{\Delta s}{f} v_i. \tag{4}$$

$$v_{i+1} = v_i + \Delta s \nabla f. \tag{5}$$

The step size, $\Delta s$, can vary to adapt to the homogeneity of the refractive indices by using the acceleration structure computed in the pre-computation step described in section 4.2. The step size is modified to be the maximum of the base step size and the largest homogeneous region that can be skipped over. The homogeneous region will be $2^j$ times the size of a voxel, where $j$ is the octree level stored at the current location $x_i$.

### 5.3 Reproducing the Cutoff

A typical setup may have a knife-edge in the center of the focal region to reduce any unaltered light by half. This allows both brighter and darker displaced regions to show up in the resulting image. This intensity value is accumulated from the number of photons that reach the film plane and a Monte Carlo Russian roulette style killing off of photons leads to a realistic simulation of this process. However, assigning an energy value instead which can be weighted by the probability of being killed significantly reduces noise and requires fewer photons to be traced. If $\vec{d}$ is the resulting ray direction at the cutoff region and $\vec{d_o}$ is the original ray direction from when the ray was first generated then the resulting displacement is:

$$\vec{e} = (\vec{d} - \vec{d}_o) \tag{6}$$

$$e_x = \vec{e} \cdot \vec{camera}_x \tag{7}$$

$$e_y = \vec{e} \cdot \vec{camera}_y \tag{8}$$

where $e_x$ and $e_y$ are displacements along the camera axis $\vec{camera}_x$ and $\vec{camera}_y$. If $\vec{e}$ is the displacement from the original direction relative to the camera angle then the resulting change in illumination $I$ from a vertical knife-edge cutoff is:

$$\frac{\delta I}{I} = \frac{Kc_2}{\vec{e}} \int_{\gamma 1}^{\gamma 2} \frac{\partial p}{\partial z} dy \tag{9}$$

where $c_2$ is the focal distance of the lens projecting light onto the cutoff, $K$ is the Gladstone-Dale constant, and the displacement is iterated over the focal region with the integral where $\gamma 1$ and $\gamma 2$ are the z coordinates of the ray entering and leaving the medium and p is the density [10, 15]. In the experimental setup the focal distance or the cutoff can be altered in order to intensify the change in illumination. In a computer simulation, however, the same effect can be achieved by replacing $c_2$, $K$, and the integration over the focal region by a scalar value, $k$. This value can be altered to correspond to an optical setup or modified to fit a desired range of intensities.

$$I = 0.5 - e_y * k \quad \text{(horizontal knife-edge).} \tag{10}$$

$$I = 0.5 - e_x * k \quad \text{(vertical knife-edge).} \tag{11}$$

$$I = 1 - |\vec{e}| * k \quad \text{(circular cutoff).} \tag{12}$$

$$I = HSV(cos(\vec{d}, \vec{d}_o) * k, 1, |\vec{e}| * k) \quad \text{(color filter).} \tag{13}$$

The value $k$ typically maps to the largest expected displacement as to yield normalized intensities without clamping [13]. The knife-edge can be flipped or rotated as desired and the circular cutoff can become a complement circular cutoff by complementing the equations. Where a circular cutoff will show regions with more displacement as darker, a complement cutoff shows regions with higher displacement as brighter. Once the intensities have been weighted according to the cutoff they are projected to the film plane and their values are accumulated. This leads to a potential race condition as different threads try to write to the same regions of the film plane at the same time. CUDA provides atomic operations that result in a slight speed decrease but overall we find that this occurrence is sufficiently rare enough to ignore without introducing noticeable error for most instances. However in cases where there is a great deal of refraction synchronization may be necessary to avoid artifacts. For such cases we wrote values and the window coordinates into shared memory space where each thread has it's own separate index into a shared memory buffer. At the end of a the CUDA kernel the threads synchronize and thread zero writes the values from shared memory out to the pixel buffer.

## 6 Filtering

Once a sufficient number of photons have been traced, the resulting image is filtered for noise and rendered to the screen. A simple Gaussian filter helps reduce noise while smoothing over gradations in luminance values. While a mean filter is better suited for reducing noise, it blurs out many of the small details.

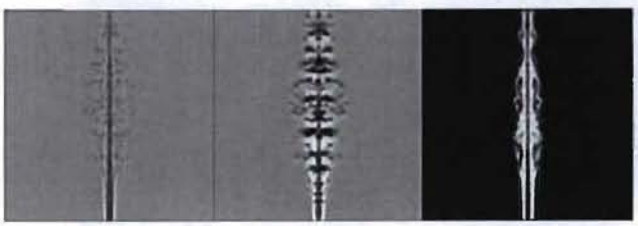
Photon mapping techniques have developed several methods for reducing noise on a surface with a limited number of photons. Jensen et al. [7] presented a cone filtering method weighting a given area by a sphere that encapsulates a set number of photons. Low density regions have a large filter width, while areas with high sample density have smaller filter width leading to a crisper image. This works well for caustics where large numbers of samples concentrate in a small area but may not always be the best approach to rendering high frequency schlieren images where dark crisp lines may be desirable.

In practice, only limited filtering is necessary as long as the photons are produced on a regular grid and the photons are given a weighted energy corresponding to the cutoff. The filter width should be decreased as more passes in the progressive rendering system are computed. This will lead to an initially blurry image but ultimately yield a better resolved image after sufficient passes of the renderer.

Shadowgraph | Knife-edge | Circular complement

| Cutoff: | Average fps: | Photons per second: |
|---|---|---|
| None | 13.34 | 34,969,318.61 |
| Knife-edge | 13.21 | 34,633,443.87 |
| Circular complement | 11.93 | 31,276,874.99 |

Figure 8: Results of a combustion dataset of dimensions 480x720x100 seen in figure 6 rendered with 10 iterations of progressive refinement per frame using cone filtering on a Geforce GTX 280 card at 512x512 resolution.



Shadowgraph | Knife-edge | Circular complement

| Cutoff: | Average fps: | Photons per second: |
|---|---|---|
| None | 25.71 | 33,698,933.02 |
| Knife-edge | 24.73 | 32,412,275.28 |
| Circular complement | 20.76 | 27,210,861.76 |

Figure 9: Results of a coal fire with 5 iterations of progressive refinement per frame on a Geforce GTX 280 card at 512x512 resolution.

## 7 Results

The method allows for high photon counts per second on approximately $256^3$ sized datasets, as shown in figures 8 and 9. An NVIDIA GeForce GTX 280 GPU with 1 GB VRAM was used for timings. 35 million photons allows for a nearly interactive 13 FPS on a 512x512 image with 10 samples per pixel (10 iterations of progressive refinement) on the combustion dataset, as demonstrated in figures 8 and 9. The frame rate varies based on the frequency of the data due to the adaptive step sizes through the volume and the size of the overall dataset. The frame rate is further influenced by the image size. This compares favorably as a visualization method to the images generated by Anyoji et al. [1], who reported rendering times of about 20 minutes. Figure 8 shows a moderate impact of using a cutoff with a shadowgraph performing slightly faster than a knife-edge cutoff and noticeably faster than the circular cutoff due to the

normalization required in equation 12. The progressive rendering system displays a blurry image while rotating but a very crisp image with fine details when the mouse is released which works very well in practice. The amount of time for the image to converge varies, but when generating the images and videos for this paper we found that typically after one second (at least 100 iterations) there was little discernible improvement in image quality with additional time for 512x512 images.

Figure 10(a) shows a helium plume rendered using a traditional volume rendering technique that uses a one dimensional transfer function over the density scalar field. This is compared against an approximation of schlieren imaging without computing refraction in 10(b), and our method shown in 10(c). The refractive indices were computed from density measurements using the Gladstone-Dale relation as shown in section 4.1 with a Gladstone-Dale constant of $0.233cm^3/g$ for air and helium due to a lack of mixture fractions. The volume rendered image using a transfer function provides a good indication of the shape of the flow by showing a discrete surface where the helium meets the surrounding air. The schlieren rendering in figure 10(b) gives no indication of depth but gives a detailed rendering of the underlying changes in the flow by shading the degree of change in the flow rather than a set density value. This is similar to a technique of shading a volume based on the magnitude of a gradient except that the shading conforms to a cutoff value and alters according the ray direction. The fringes of features are pulled out giving a silhouette to areas of the flow where large changes in the flow meet with orthogonal viewing rays. The technique also alleviates the need to tweak a transfer function as both large and small changes in the data are displayed and shaded according to their values, akin to an accumulative maximum intensity projection. A transfer function can still be used to pull out certain parts of the data using the technique, however the resulting image will no longer match the actual experimental schlieren image. Figure 10(c) gives further information and an accurate reproduction of what a real schlieren photograph would show by tracing refraction through the data. The bottom of the plume shows sharp features where the helium is emanating resulting in significant changes in refractive indices. The rays cluster or disperse around the incoming helium resulting in sharp areas in the flow instead of the area clamping to white as seen in 10(b) without refraction. This becomes less severe towards the top of the plume which shows that the helium is mixing with the air resulting in less light refraction. Edges of the flow are further enhanced as light in those areas bends around large changes in refractive indices.

Figure 11(e) shows an image from a simulation of the X38 aircraft on re-entry. The coloring over the density field shows distinct regions by showing differences in direction that a one-dimensional knife-edge cutoff might miss. Coloring a more detailed image such as the coal fire or heptane datasets as in figure 6(c) results in more information but users may prefer to see only intensity variations. Figure 11 shows a comparison of our method with Svakine et al.'s method [18] using the X38 dataset. Our method provides a clear image of the airflow around the body and bow of the plane, as well as vortices formed around the tail fins of the plane.

Filtering is very beneficial when rendering with a small photon count. To illustrate this, figure 12 shows an image of a combustion dataset rendered with and without filtering at different samples per pixel. In our experiments, we found a count of 10 samples per pixel to be sufficient to reveal coarse low-frequency features while finer details come through when the user stops interacting with the system and the renderer quickly reaches over 100 samples per pixel in less than a second. The rows in figure 12(a) clearly demonstrate the benefits of filtering at lower sample counts.

## 8 Discussion and Future Work

In this paper we have demonstrated that reproducing light paths for computing schlieren photographs is possible at nearly interactive frame rates by intelligently combining various acceleration techniques and exploring the computational resources of modern graphics hardware. The method provides scientists with an accurate tool simulating familiar visualization techniques in a computational environment which requires far less resources and time than an experimental setup with physical constraints and complicated optics. The method also opens the door for making a sufficiently accurate reproduction of real world photographs that can be used to validate simulation data.

Reproducing an exact replication of schlieren photographs' error presents several challenges. One such source of error comes from one of the many cutoffs used and the artifacts they may produce. It is not clear to what degree these artifacts contribute to the overall image but the various cutoffs used may present undesirable refraction themselves [13]. Reproducing the lenses may also be necessary for a mathematical representation of their respective focal lengths affecting the focusing around the cutoff. Additionally, the light source could be faithfully reproduced as well as the amount of luminance over the length of the exposure.

The system assumes a constant wavelength across photons. Visible light waves have wavelengths across the visible spectrum and will refract differently producing various effects such as chromatic aberration. Finally, only purely refractive flows have been investigated so far, but simulating scattering effects may also be necessary depending on the materials used in the simulation. Some materials, such as fire, may even need emissive calculations. Future work could explore all of the above issues for faithfully reproducing an experimental setup.


## Acknowledgements

The authors would like to thank Kelly Gaither for providing the x38 data and David Ebert for allowing us to reuse images from [18]. We would like to thank Gary Settles for images of shadowgraph and schlieren photographs.

Additional acknowledgements were removed for an anonymous submission.


## Appendix A.

Ciddor presents a method for computing accurate refractive indices from air in [4]. The method is composed of a 10 step process that calculates the densities and compressibility of air at certain conditions in order to compute the refractive index. While it is beyond the scope of this paper to reproduce the entire derivation here the method is largely governed by

$$n_{prop} - 1 = (p_a/p_{axs})(n_{axs} - 1) + (p_w/p_{ws})(n_{ws} - 1). \quad (14)$$

In Eq.14 $n_{prop}$ is the refractive index that is being calculated., $p_{axs}$ is the density of dry air at 15°C., and $p_{ws}$ is the density of pure water vapor at 20°C. The other variables, $p_a$ and $p_w$, are the densities of the dry air and water vapor components.

## References


[1] M. Anyoji and M. Sun. Computer analysis of the schlieren optical setup. In *Proc. of SPIE*, volume 6279, page 62790M, 2007.

[2] H. T. B. M. S. Atcheson, Irkhe. Time resolved 3d capture of non-stationary gas flows. *ACM Transaction on Graphics*, 25(5):132, December 2008.

[3] M. Born, E. Wolf, and A. B. Bhatia. *Principles of Optics (7th edition)*. Cambridge University Press, New York, 7 edition, 1999.

[4] P. E. Ciddor. Refractive index of air: New equations for the visible and near infrared. *Applied Optics*, 35:1566, 1996.

[5] D. Gutierrez, F. J. Seron, O. Anson, and A. Munoz. Chasing the green flash: A global illumination solution for inhomogeneous media. *Spring Conference on Computer Graphics 2004*, pages 97–105, 2004.

[6] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Trans. on Graphics (Siggraph'07)*, pages 59:1–9, Aug. 2007.

[7] H. W. Jensen. Global illumination using photon maps. In *proceedings of the seventh eurographics workshop on rendering*, pages 21–30, 1996.

[8] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. *Proceedings of ACM Siggraph 98*, pages 311–320, 1998.

[9] G.-S. Li. *Interactive Texture Based Flow Visualization*. PhD thesis, University of Utah, 2008.

[10] W. Merzkirch. *Flow Visualization*. Academic Press, 1987.

[11] NVIDIA. *CUDA Programming Guide*, 2009. `http://developer.nvidia.com/object/cuda.html`.

[12] V. Pegoraro, C. Brownlee, P. S. Shirley, and S. G. Parker. Towards interactive global illumination effects via sequential monte carlo adaptation. In *Proceedings of the 3rd IEEE Symposium on Interactive Ray Tracing*, pages 107–114, 2008.

[13] G. Settles. *Schlieren and Shadowgraph Techniques, Visualizing Phenomena in Transparent Media*. Springer, New York, 2001.

[14] G. Settles. High-speed imaging of shock waves, explosions and gunshots. *American Scientist*, 94(1):22–31, 2006.

[15] A. J. Smits and T. T. Lim. *Flow Visualization : Techniques and Examples*. Imperial College Press, London, 2000.

[16] M. Sun. Computer modeling of shadowgraph optical setup. In *Proc. of SPIE*, volume 6279, page 62790L, 2007.

[17] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo. Interactive relighting of dynamic refractive objects. *ACM Transaction on Graphics*, 27(3):35:1–9, 2008.

[18] N. A. Svakhine, Y. Jang, D. Ebert, and K. Gaither. Illustration and photography inspired visualization of flows and volumes. *IEEE Visualization 2005*, pages 687–694, 2005.

[19] L. A. Vasil'ev. *Schlieren Methods*. Keter Inc, New York, 1971.

[20] L. A. Yates. Images constructed from computed flowfields. *AIAA*, 31(10):1877–1884, 1993.

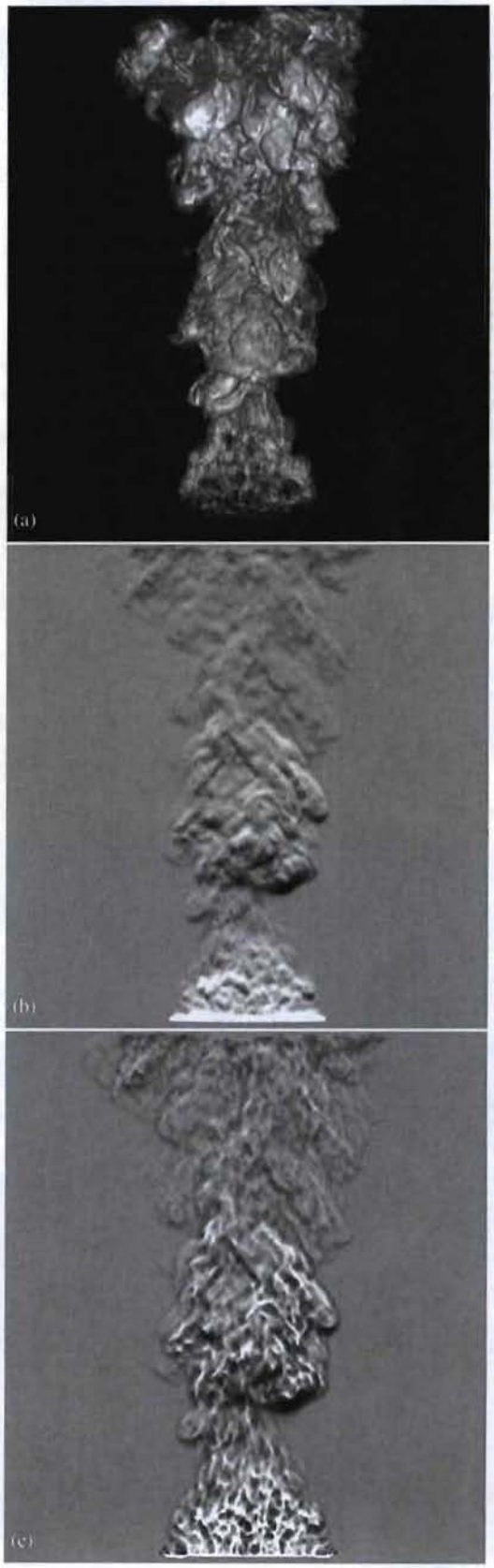



Figure 10: Comparison of volume rendering (a) with a line of sight schlieren approximation (b) with our method (c).

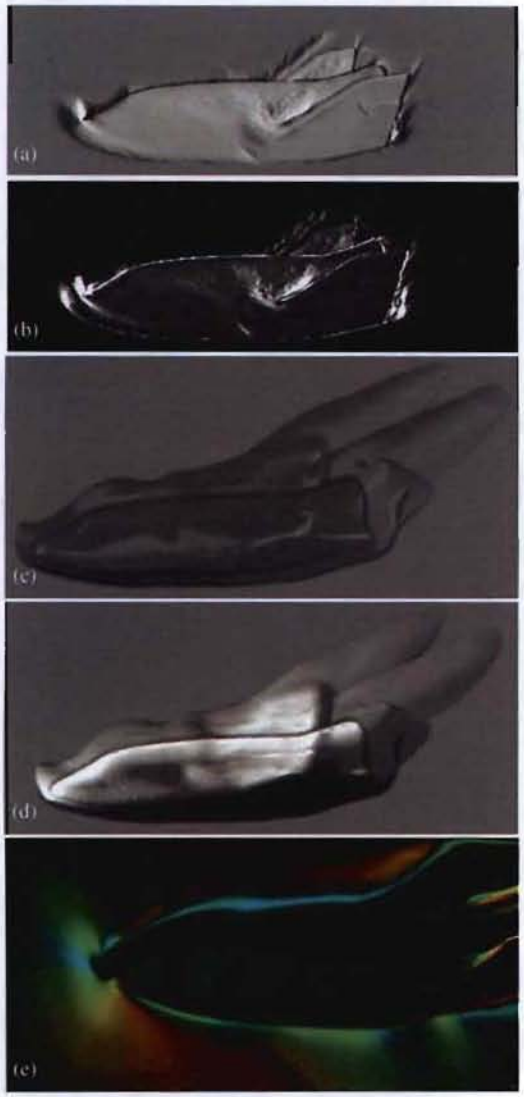



Figure 11: Comparison of the line of sight technique [18] (a,b, reproduced with permission) and our method using a shadowgraph (c) a knife-edge cutoff (d), and a color filter (e).

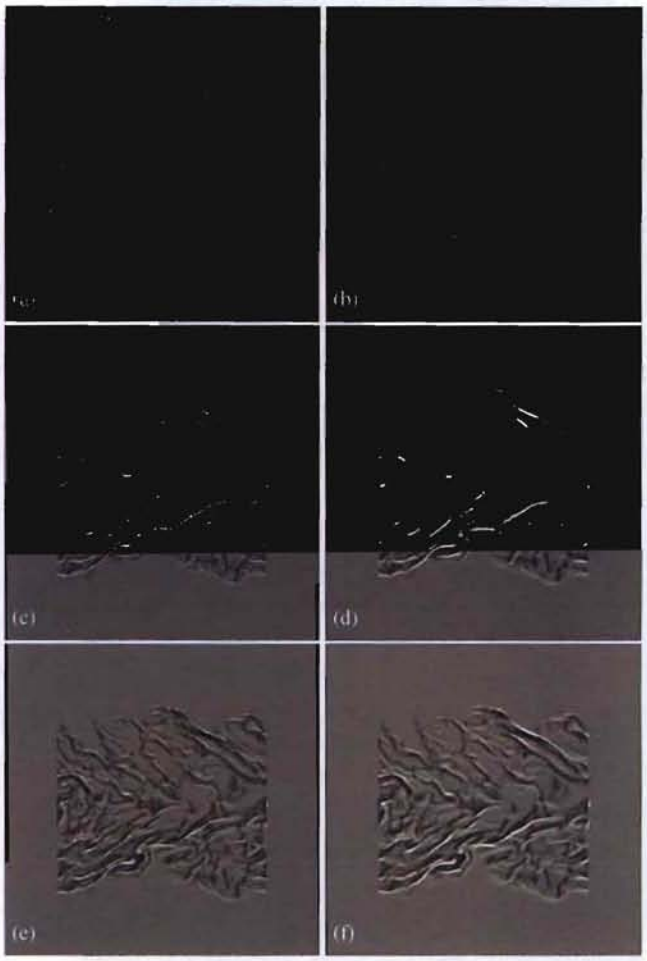



Figure 12: Comparison of unfiltered film plane with 1, 10, and 100 samples per pixel (a, c ,e) and the corresponding images of the film plane filtered with a cone filter of maximum width 6 in (b, d, f).