

# Physically-Based Interactive Flow Visualization Based on Schlieren and Interferometry Experimental Techniques

Carson Brownlee, Vincent Pegoraro, Siddharth Shankar,  
Patrick S. McCormick, *Member, IEEE*, and Charles D. Hansen, *Senior Member, IEEE*

**Abstract**—Understanding fluid flow is a difficult problem and of increasing importance as computational fluid dynamics (CFD) produces an abundance of simulation data. Experimental flow analysis has employed techniques such as shadowgraph, interferometry, 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. Interferometry tracks changes in phase-shift resulting in bands appearing. 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 shadowgraph, schlieren and interferometry images of time-varying scalar fields derived from computational fluid dynamics 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. Applications of our method to multifield data and custom application-dependent color filter creation are explored. Results comparing this method to previous schlieren approximations are finally presented.

**Index Terms**—Scalar field data, GPUs and multicore architectures, flow visualization.

## 1 INTRODUCTION

RECENT advances in computational fluid dynamics (CFD) have produced a wealth of simulated flow data [6], [8], [13], [14]. Understanding these flows is of great importance for applications ranging from aircraft design to combustion analysis [20]. A range of techniques have been developed for understanding these flows both computationally and experimentally [26]. 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 complicated optics involved.

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 [25], [28], 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 our previous paper, we presented a novel technique for generating schlieren and shadowgraph images by tracing light paths through time-varying scalar fields of computed flows [5]. 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 data sets, 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.

In this paper, we expand upon our previous work by introducing interferometry visualization as well as interactive color filter editing and an exploration of multifield data visualization. Interferometry allows a different view of data

- C. Brownlee and C.D. Hansen are with the Scientific Computing and Imaging Institute, University of Utah, 72 S Central Campus Drive, Room 3750, Salt Lake City, UT 84112. E-mail: {brownlee, hansen}@cs.utah.edu.
- V. Pegoraro is with the Universität des Saarlandes, Fachbereich 6.2 - Informatik, Lehrstuhl für Computergraphik, Campus E1 1, Zimmer E 16, 66123 Saarbrücken, Deutschland. E-mail: pegoraro@cs.uni-saarland.de.
- S. Shankar is with the TerraSim, Inc., One Gateway Center, Suite 2050, 420 Ft. Duquesne Blvd., Pittsburgh, PA 15222. E-mail: funkysidd@gmail.com.
- P.S. McCormick is with the Los Alamos National Labs, PO Box 1663, Los Alamos, NM 87545. E-mail: pat@lanl.gov.

Manuscript received 1 June 2010; revised 18 Aug. 2010; accepted 19 Oct. 2010; published online 7 Dec. 2010.

Recommended for acceptance by H.-W. Shen, J.J. van Wijk, and S. North.

For information on obtaining reprints of this article, please send e-mail to: tvccg@computer.org, and reference IEEECS Log Number TVCGSI-2010-06-0111.

Digital Object Identifier no. 10.1109/TVCG.2010.255.

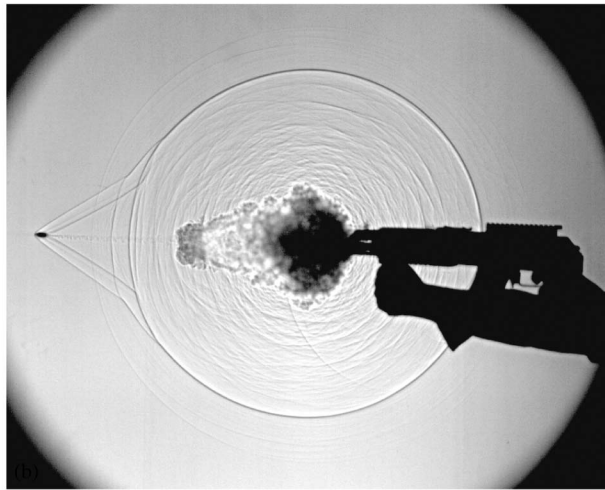
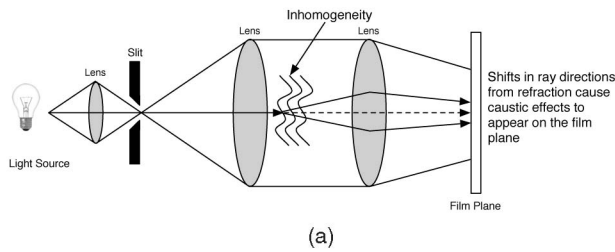


Fig. 1. (a) 2D illustration of the shadowgraph optical setup. (b) A shadowgraph photograph of an AK-47 (Courtesy of G.S. Settles).

by tracking phase-shift through the flow producing visual bands. Custom color filters allow for exploring specific regions in the flow. Multifield data present a difficult problem and an interesting exploration of schlieren visualization.

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. We further explore the use of our method with custom color filters in Section 7 and look at multifield data in Section 8. A description of the quality of the images and performance are then given in Section 9. Finally, we end the paper with ideas for further extensions to our method in Section 10.

## 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 [20]. 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. Fig. 1a shows the optical setup of a typical shadowgraph system. A light

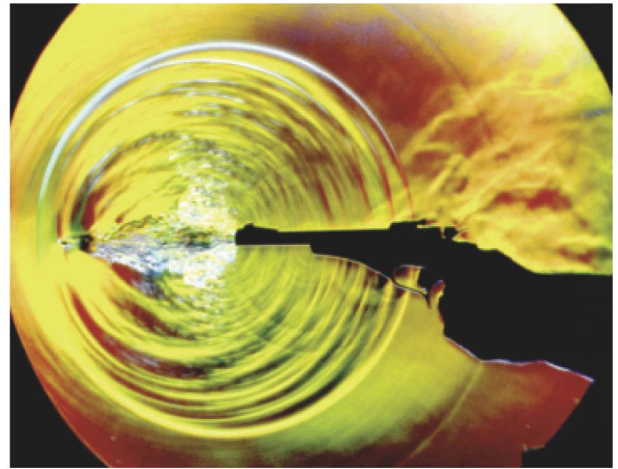
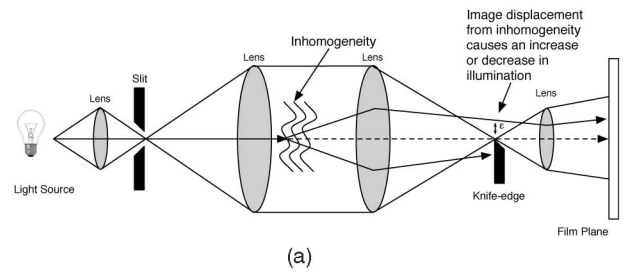


Fig. 2. (a) 2D illustration of the schlieren optical setup. (b) A schlieren photograph of a gunshot with a color filter applied (reproduced from [21] with permission).

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. Fig. 1b 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 1D cutoff that shifts intensity values based on the amount and direction of displacement at the focused cutoff region. In Fig. 2a, 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. A vertical knife-edge is then inserted at the center of the refocused light source. If no light is refracted then the knife-edge reduces the light source by half resulting in a gray image. Refracted light causes shifts in the focused image of the original light source resulting in more or less of the focused light being 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

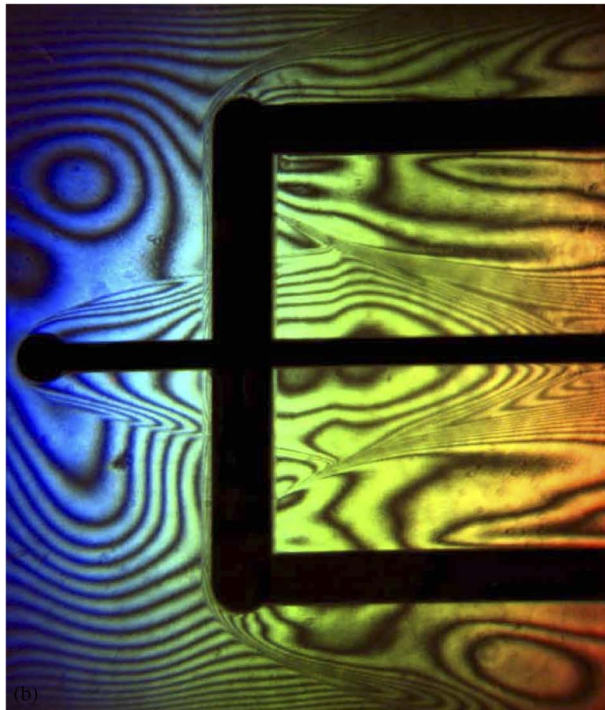
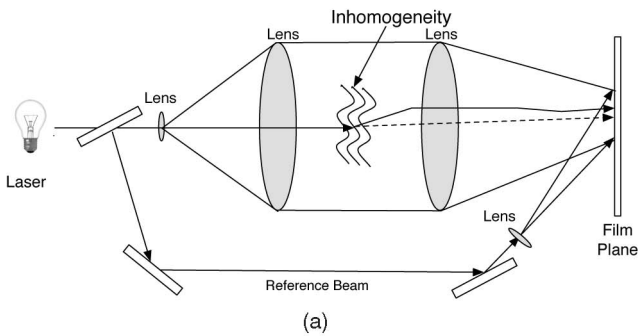


Fig. 3. (a) 2D illustration of the interferometer optical setup. (b) Example of a finite-fringe interferometry image from a physical experiment.

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 Fig. 4. Whereas a knife-edge cutoff only gives information about the amount of displacement along one axis, color can give 2D information about the direction of displacement. Fig. 2b demonstrates how a color filter cutoff emphasizes gradations in shock waves resulting from a gunshot compared to a similar shadowgraph as shown in Fig. 1b.

Interferometry differs from schlieren and shadowgraph images by looking at phase shift instead of refraction. When light travels through a disturbance and encounters a change in refractive index, the speed of that light changes resulting in a phase shift [22]. The idea behind interferometry is to directly measure this phase shift and display it providing a picture of changes in refractive index. On the experimental side, this method allows for the direct calculation of refractive indices instead of looking at changes in gradient

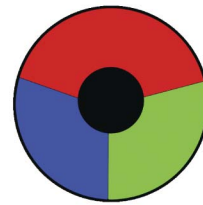


Fig. 4. A typical color filter used in schlieren optical setups.

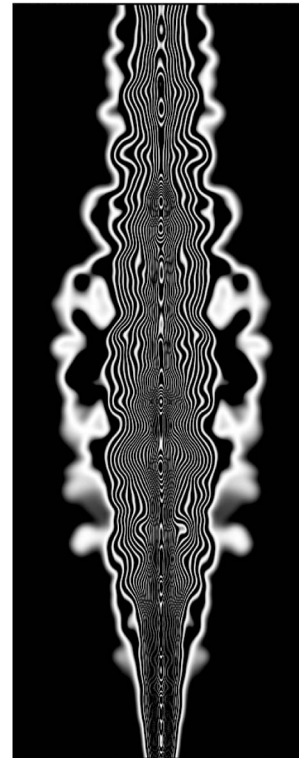


Fig. 5. Infinite-fringe interferometry image computed using our method.

values as in schlieren and shadowgraph images. The optical setup required is described in Fig. 3a. The setup starts with a beam of light typically generated by a laser because of the polarized parallel light at homogeneous frequencies. A reference beam is also created by splitting the light beam before hitting the inhomogeneity. This reference beam can be used to measure the phase shift of the main beam by comparison to the reference beam. The main light beam is sent through the inhomogeneity where differences in refraction will produce phase shifts. Where the phases line up, bright bands are created and where they conflict dark bands emerge. These banded fringe patterns produce a view of the underlying flow as depicted in Fig. 3b, which shows an experimental photograph of an interferometry setup demonstrating the flow around an expansion tube [16]. High frequency details in the image denote large changes in refractive indices. In Fig. 5, this is demonstrated by our computed image of a coal fire through the tight small bands in the center of the coal fire and larger dissipated fringes as the flame disperses.

In the perfect case, where the test beam and reference beam are perfectly aligned and no disturbances are intercepted then no fringes should appear. This is known as infinite-fringe interferometry. Finite-fringe interferometry



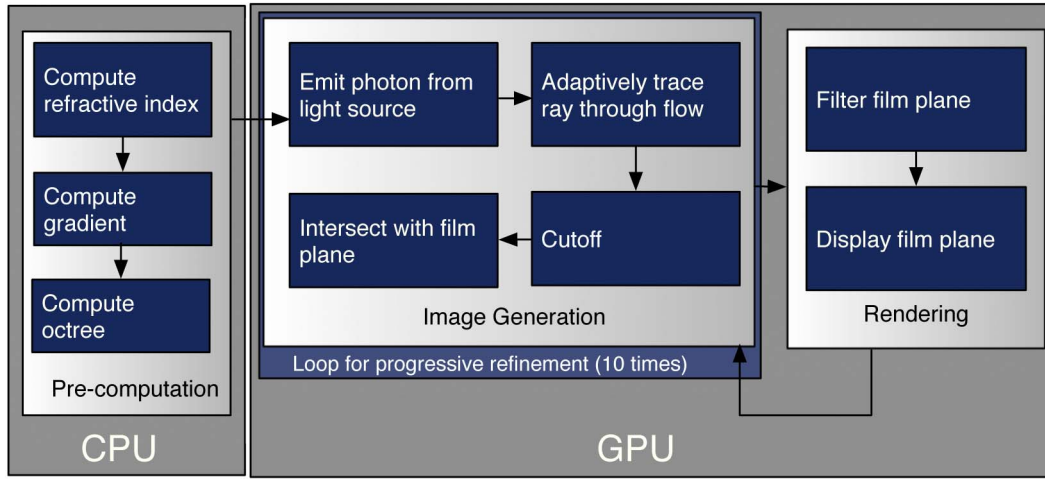


Fig. 6. Illustration of the rendering pipeline.

puts the test beam and reference beam at slight angles, producing fringes even when there is no difference in the phases of the two beams. This is a more commonly used technique as finite-fringe interferometry allows for the determination of the phase shift from the images produced on the experimental side. The spacing of these beams produced through finite-fringe interferometry can be calculated as a measurement of  $\chi$  by

$$\chi = \frac{\lambda}{2 \sin \frac{1}{2} \gamma}, \quad (1)$$

where  $\lambda$  is the wavelength of the light and  $\gamma$  the angle of beam intersection [22]. The phase shift,  $\phi$  can be calculated as a function over the refractive index field  $n$  as

$$\phi = \frac{2\pi}{\lambda} \int_{z_1}^{z_2} (n - n_0) dz, \quad (2)$$

where  $n_0$  is the reference refractive index, which is the refractive index that the reference beam is hitting [28]. In many cases, this is simply the refractive index of air. This equation assumes a line of sight traversal with no refraction, however, in the computed case the integration over the  $z$ -axis is easily adapted to the integration over a bending light path using a piecewise linear approximation [17]. A fringe is produced whenever a phase shift of  $2\pi$  or an optical distance of  $\lambda$  is encountered for the infinite-fringe case [27]. For the finite-fringe case, the phase shift and the angle between beams must be taken into account to calculate the resulting fringes.

Computational schlieren images of 3D fluid flows have been computed noninteractively using a ray tracing method by Anyoji and Sun [1], [23]. Such techniques produce an accurate image but are not ideal for data exploration. A nonphotorealistic method for producing schlieren-like images using line of sight ray traversals for visualization was recently introduced [25] but without refracting light. In order to reproduce an accurate physically based representation, tracing nonlinear 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 3D 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 [11]. Photon maps were later extended to volumetric photon mapping to compute scattering effects and caustics through inhomogeneous media [9], [12]. 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 precomputing wavefront propagation through a grid [10]. Eikonal rendering relies on a long precomputation 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 [24] that calculated 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 AND 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 Fig. 6. The precomputation steps utilize the CPU while the image generation and rendering stages are done on the GPU using CUDA [18], 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 cannot 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. Time-varying flow fields are rendered frame at a time with precomputation being computed before each frame.

## 4 PRECOMPUTATION

The precomputation stages are required to compute the refractive indices and gradient from related fields as well as construct an octree to accelerate ray traversals during runtime. These stages allow accurate and fast computations of light refraction through the flow at later stages of the pipeline. The precomputed data needed to be passed to the GPU for rendering consists of a 3D texture of refractive indices, a 3D texture of gradient values, and an array of randomly generated floats.

### 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 [7]. 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 in its entirety here due to the complexity of the method but we provide a brief overview. Ciddor presents a method for computing accurate refractive indices from air [7]. 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

$$n_{prop} - 1 = (p_a/p_{axs})(n_{axs} - 1) + (p_w/p_{ws})(n_{ws} - 1). \quad (3)$$

This equation calculates the refractive index by multiplying the proportion of dry air by the refractivity of dry air and adding in the portion of water vapor by the refractivity of pure water vapor. In (3),  $n_{prop}$  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.  $p_a$  and  $p_w$  are the densities of the dry air and water vapor components.  $n_{axs}$  and  $n_{ws}$  are the refractivity of dry air and pure water vapor.

Fig. 7a 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 preprocessing step using finite differences.

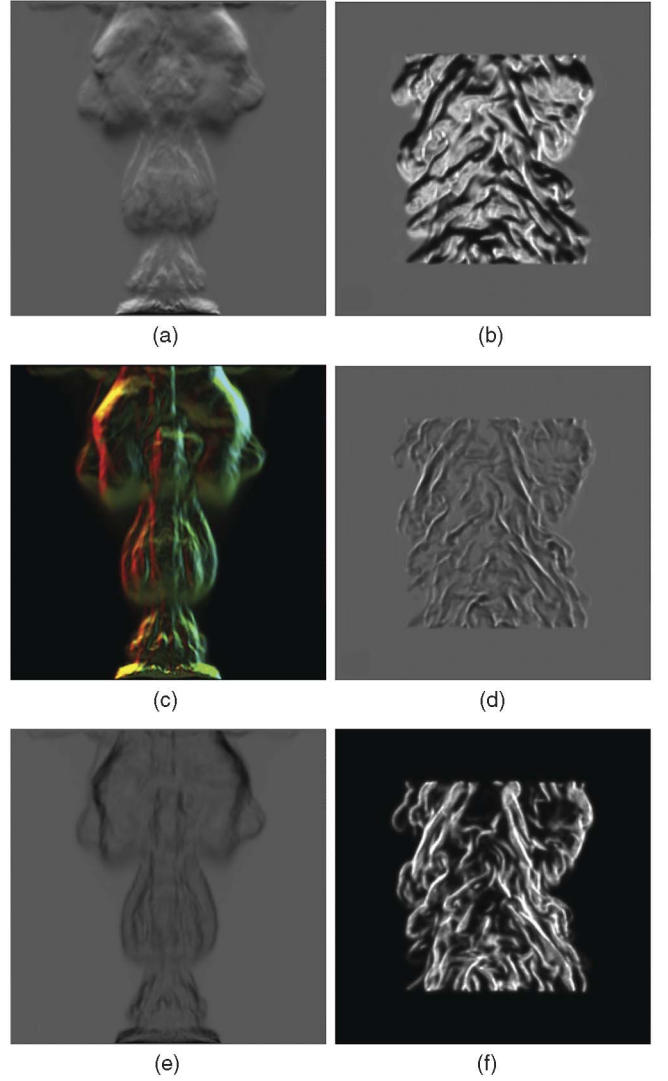


Fig. 7. A heptane data set rendered using refractive indices calculated from temperature and pressure with (a) a knife-edge cutoff, (c) color filter, and (e) a circular cutoff. (b) A simulated combustion data set rendered using a schlieren knife-edge cutoff to enhance the flow, (d) as a shadowgraph image, and (f) using a complemented circular cutoff.

Alternatively, the Gladstone-Dale relation provides a method for computing the refractive indices from density fields [17]. Because we are working with gasses, we use the abbreviated form:

$$n - 1 = Kp. \quad (4)$$

$K$  defines the Gladstone-Dale constant,  $p$  the sample 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. As an example, one can vary the value from 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  $0.633 \mu\text{m}$ , we then know that  $K_h$  for helium is approximately  $0.196 \text{ cm}^3/\text{g}$  and  $K_a$  is  $0.226 \text{ cm}^3/\text{g}$  for air. If  $m_h$  is the volume fraction of helium in the mixture and  $m_a$  is the fraction of air then  $n$  can be found by

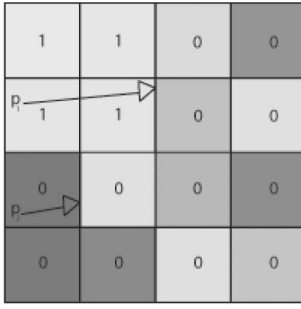


Fig. 8. An illustration of a traversal through the octree.  $P_1$  and  $P_2$  are two rays traversing through the flow.  $P_1$  is in a homogeneous region of the data and in a cell of the octree texture that will report a level number of 1 allowing  $P_1$  to skip to the edge of that level.  $P_2$ , on the other hand, is at the lowest level of the acceleration structure and will only traverse to the next voxel.

$$n = (K_h * m_h + K_a * m_a) * p + 1. \quad (5)$$

## 4.2 Octree

Many flow data sets 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. [24]. 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 Fig. 8, which shows two different rays that lie in different levels of the octree. The distance to the edge of the octree level is calculated to avoid overstepping homogeneous regions.

## 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 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, but 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 is 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 precomputed array of random floats. Thus, the system only needs to generate three random numbers at each pass instead of thousands or millions.

### 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 [19], 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].

$$\frac{d}{ds} \left( f \frac{dx}{ds} \right) = \nabla f. \quad (6)$$

In order to simulate  $x(s)$ , this equation is discretized using piecewise linear approximations. The position  $x_i$  is updated according to the 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  $\nabla f$ .

$$x_{i+1} = x_i + \frac{\Delta s}{f} v_i. \quad (7)$$

$$v_{i+1} = v_i + \Delta s \nabla f. \quad (8)$$

The step size,  $\Delta s$ , can vary to adapt to the homogeneity of the refractive indices by using the acceleration structure computed in the precomputation 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. A better solution is to assign an energy value instead, which can be weighted by the probability of being killed, significantly reducing noise, and requiring 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

$$\vec{e} = (\vec{d} - \vec{d}_o), \quad (9)$$

$$e_x = \vec{e} \cdot \text{cam}\vec{e}ra_x, \quad (10)$$

$$e_y = \vec{e} \cdot \text{cam}\vec{e}ra_y, \quad (11)$$

where  $e_x$  and  $e_y$  are displacements along the camera axis  $\text{cam}\vec{e}ra_x$  and  $\text{cam}\vec{e}ra_y$ . 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

$$\frac{\delta I}{I} = \frac{K c_2}{\vec{e}} \int_{\delta 1}^{\delta 2} \frac{\partial p}{\partial z} dy, \quad (12)$$

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  $\delta 1$  and  $\delta 2$  are the  $z$ -coordinates of the ray entering and leaving the medium and  $p$  is the density [17], [22]. 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.

In the case of computing interferometry, the phase shift must be computed at each step in a piecewise linear fashion in order to approximate (2). At each step, we compute  $n_{sum} = n_{sum} + (n - n_0) * \Delta s$ . At the end of the traversal the intensity is computed as the following:

$$I = 0.5 - e_y * k \quad (\text{horizontal knife-edge}), \quad (13)$$

$$I = 0.5 - e_x * k \quad (\text{vertical knife-edge}), \quad (14)$$

$$I = 1 - |\vec{e}| * k \quad (\text{circular cutoff}), \quad (15)$$

$$I = \sin\left(\frac{2\pi}{\lambda} n_{sum}\right) \quad (\text{infinite-fringe interferometry}), \quad (16)$$

$$I = HSV(\cos(\vec{d}, \vec{d}_o) * k, 1, |\vec{e}| * k) \quad (\text{color filter}). \quad (17)$$

The value  $k$  typically maps to the largest expected displacement as to yield normalized intensities without clamping [20]. 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. In cases, where there is a great deal of refraction, however, 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 its own separate index into a shared memory buffer. At the end of 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.

Several methods exist for smoothing images generated with a limited number of photons. Jensen [11] presented a cone filtering method weighting a given area by a sphere that encapsulates a set number of photons for use in photon mapping. 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.

## 7 INTERACTIVE CUTOFF CREATION

In a schlieren optical setup, if regions of the data need to be shaded in a certain way a custom cutoff can be created through a painstaking process involving a lot of trial and error. Fortunately, this becomes easy on the computer. The ability to paint directly into a color filter and see the results in real time becomes instrumental in pulling out regions of data. To eliminate the need to determine where to paint we have also enabled the user to paint on the color filter by clicking on a pixel of the resultant image. That pixel is then traced through the schlieren system to determine where it lies on the color filter and the appropriate region of the color filter is then colored in as demonstrated in Fig. 9. This helps remove the guess work of where to color the filter. Additional aid to the user can be given by displaying a histogram of where light is hitting the color filter, thus showing where high frequencies of photons are gathering in specific regions, which the user can then focus on and paint accordingly. These methods allow for high frequency informative color filters to be created very easily through interaction with the program.

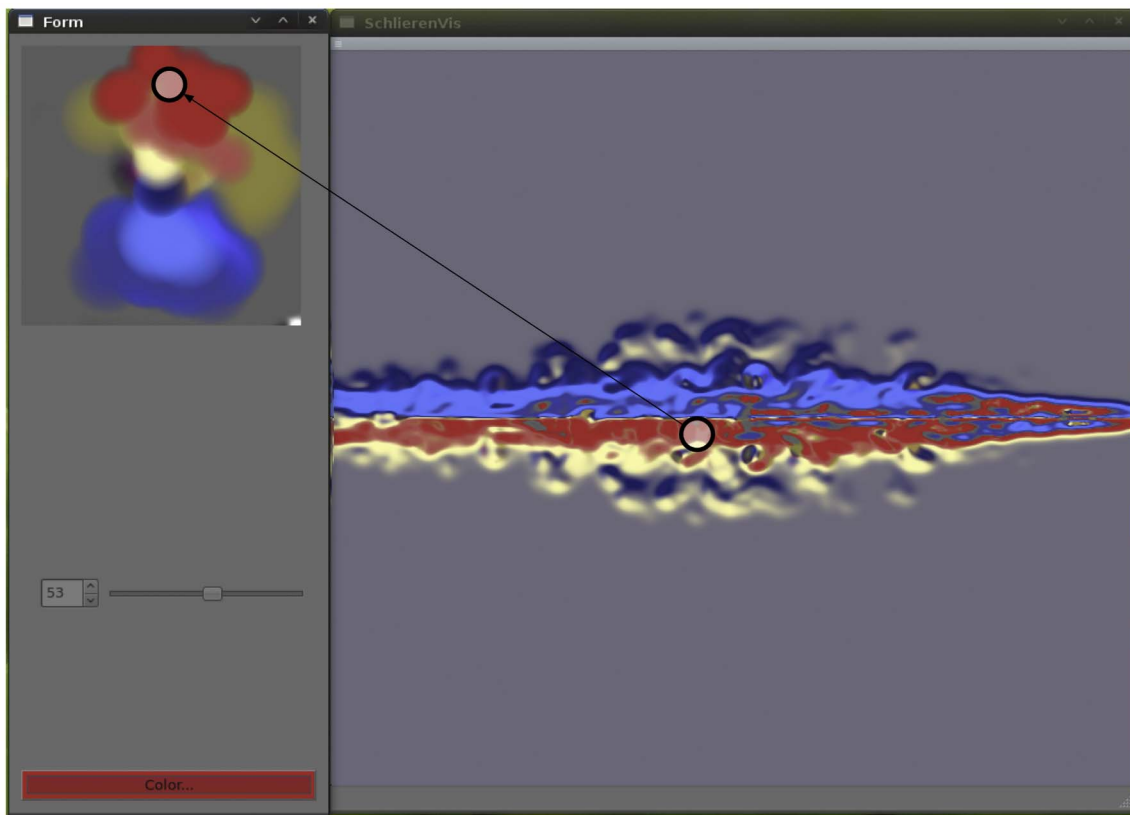


Fig. 9. Creating a custom color filter by painting on the schlieren image. The corresponding region on the color filter is looked up by calculating where that pixel lies on the color filter and coloring the filter red in this case.

## 8 MULTIFIELD FLOW ANALYSIS

The techniques presented in this paper are focused on accurate reproduction of physical methods, however, the visualization of data with no direct mapping to a real world experiment is possible. One such example is the visualization of multifield data. Fig. 10 demonstrates a schlieren image of a 5-field CFD combustion simulation that combines the scalar dissipation rate, Hydrogen Oxygen mass fraction, vorticity, heat release, and mixture fraction fields together [15]. This combination of data fields allows us to look at several possible features at once, such as the high frequency mixing inside the jet as well as the large features of the outer flame present surrounding the jet. Since there is no real world correlation from this combination of data fields, there is also no direct computation of refractive indices and so a mapping was created from the data values to refractive indices that could highlight the respective areas of interest in the data without drowning out other data fields.

The refractive index then becomes  $n = f(r_1, r_2, r_3, r_4, r_5)$  where  $f$  is a function combining each resultant partial refractive index  $r_i$ , where  $r_i = t_i(s_i)$  and  $s$  is the scalar field. Here  $t_i$  is a function that maps each scalar value from a data field into a partial refractive index. This mapping should put the scalar value within some limits depending on the results desired. For our simulation a  $t_i$  that mapped each data field with the refractive index of air was chosen. In order to bring out features, each  $t_i$  can be computed through a simple 1D transfer function specific to that field in order to bring out features. For our example, we used a simple average where  $n = r_1 + r_2 + r_3 + r_4 + r_5$  and each  $r_i = s_i/5$ . Instead of

specifying opacity values in a traditional transfer function used with volume rendering, the x-axis of our transfer functions specified the original normalized data values and the y-axis maps to the resultant altered refractive indices. As such an unaltered transfer function starts out with a simple diagonal line. Because a schlieren image is altered according to the derivative of the data, setting uninteresting regions to the lowest value in the transfer function might not have the desired result. Instead, straight lines result in no refraction and larger changes in the transfer function result in more pronounced regions in the resulting image.

As expected the final image is of much higher frequency than any of the individual fields as it is a convolution of multiple fields. One issue with this, however, is that it becomes hard to tell which feature corresponds to which data field. One possible way to ameliorate this would be to accumulate a color when sampling along the ray according to the data field being used, which is then weighted by the magnitude of the gradient such that the data field with the larger gradient will contribute more color. When the schlieren cutoff is applied the resulting image then attains both a global view of the flow through the schlieren cutoff as well as color information pertaining to each field. In practice, we found that the colors become mixed together and unhelpful when dealing with more than three data fields at once.

## 9 RESULTS

The method allows for high photon counts per second on approximately  $256^3$  sized data sets, as shown in Figs. 11



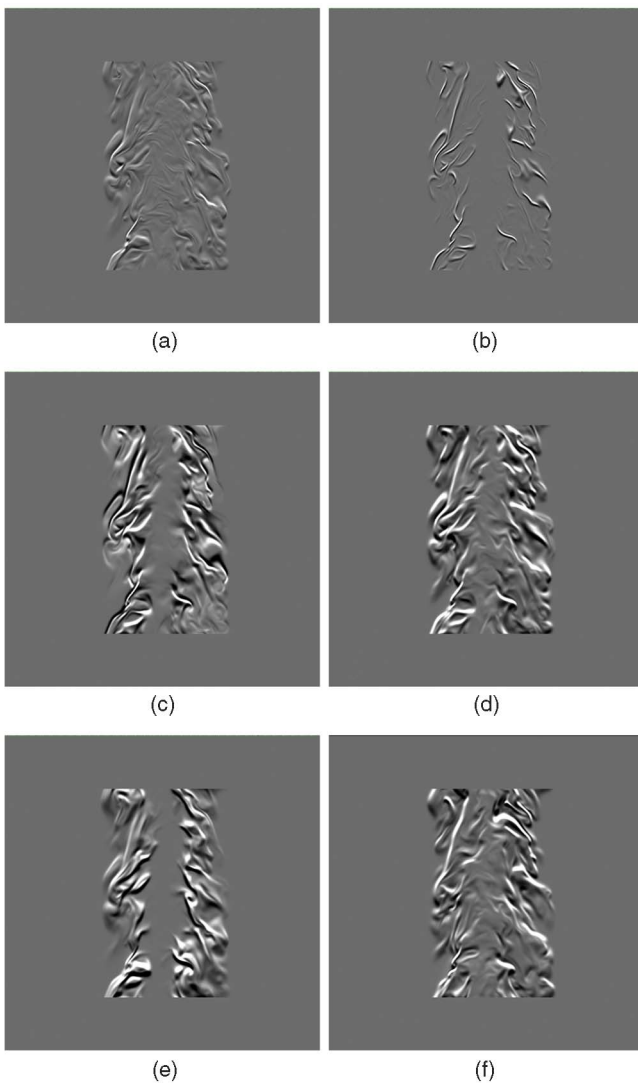


Fig. 10. Demonstration of multifield data rendered using a schlieren knife-edge cutoff: (a) shows a combination of five different data fields and (b), (c), (d), (e), and (f) show individual renderings of scalar dissipation rate, heat release, vorticity, Hydrogen Oxygen mass fraction, and mixture fractions, respectively.

and 12. An NVIDIA GeForce GTX 280 GPU with 1 GB VRAM was used for timings. Thirty-five million photons allow for a nearly interactive 13 FPS on a  $512 \times 512$  image with 10 samples per pixel (10 iterations of progressive refinement) on the combustion data set, as demonstrated in Figs. 11 and 12. 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 data set. The frame rate is further influenced by the image size. This compares favorably as a visualization method to the images generated by Anyoji and Sun [1], who reported rendering times of about 20 minutes. Fig. 11 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 (15).

Fig. 13a shows a helium plume rendered using a traditional volume rendering technique that uses a 1D transfer function over the density scalar field. This is compared against an approximation of schlieren imaging

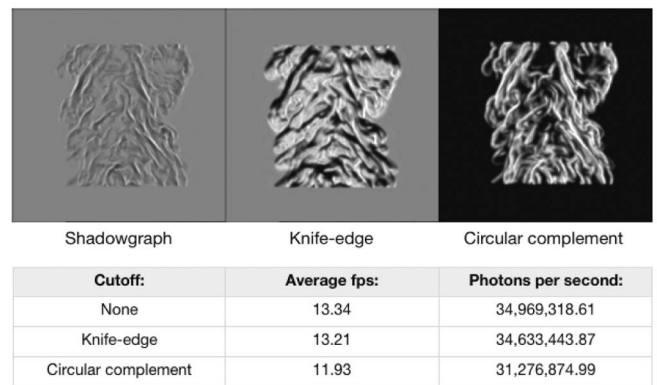


Fig. 11. Results of a combustion data set of dimensions  $480 \times 720 \times 100$  seen in Fig. 7 rendered with 10 iterations of progressive refinement per frame using cone filtering on a GeForce GTX 280 card at  $512 \times 512$  resolution.

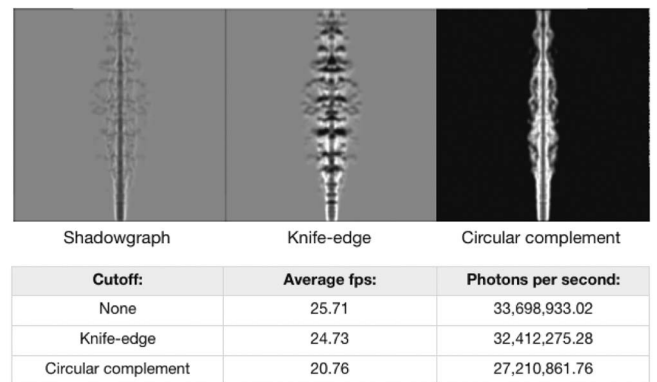


Fig. 12. Results of a coal fire with five iterations of progressive refinement per frame on a GeForce GTX 280 card at  $512 \times 512$  resolution.

without computing refraction in Fig. 13b, and our method shown in Fig. 13c. 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  $0.233 \text{ cm}^3/\text{g}$  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 Fig. 13b 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 to 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, though the resulting image will no longer match the actual experimental schlieren image. Fig. 13c 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

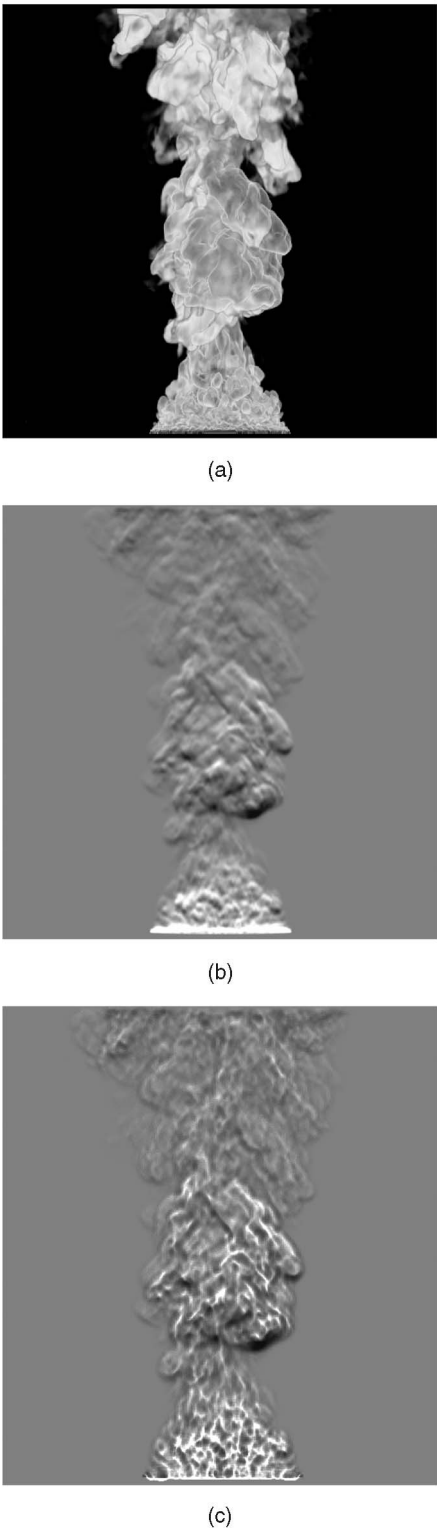


Fig. 13. Comparison of volume rendering (a) with a line of sight schlieren approximation (b) and with our method (c).

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 Fig. 13c without refraction. This becomes less severe toward 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

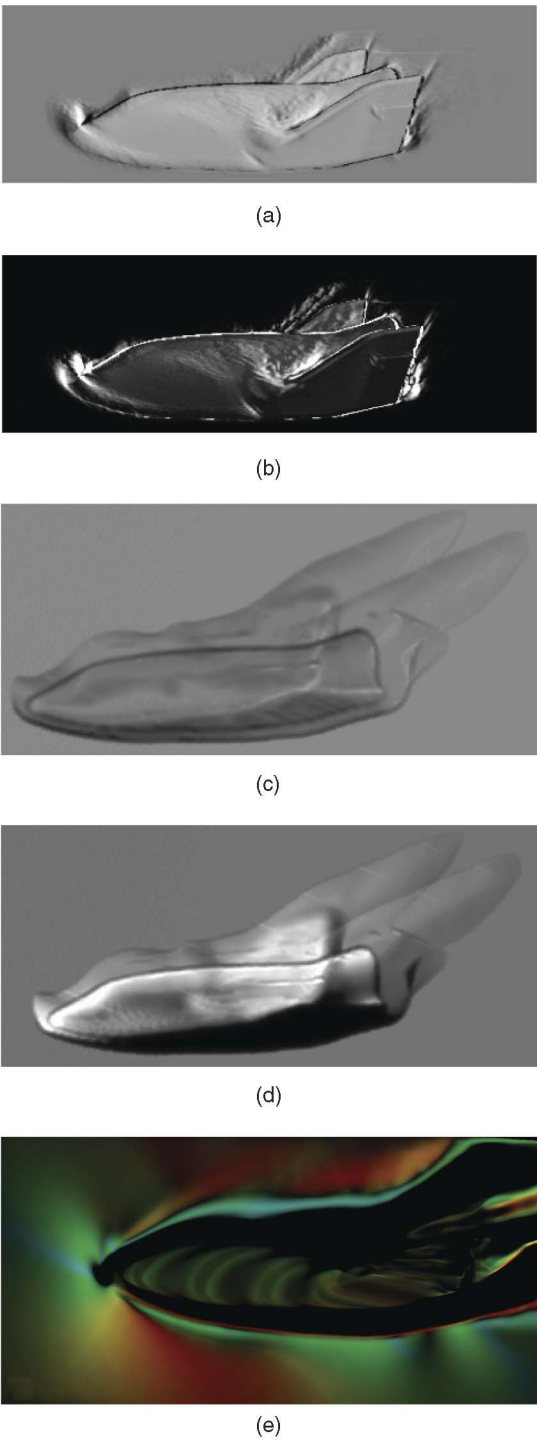


Fig. 14. Comparison of the line of sight technique [25] (a,b, reproduced with permission) and our method using a shadowgraph (c) a knife-edge cutoff (d), and a color filter (e).

enhanced as light in those areas bends around large changes in refractive indices.

Fig. 14e shows an image from a simulation of the X38 aircraft on reentry. The coloring over the density field shows distinct regions by showing differences in direction that a 1D knife-edge cutoff might miss. Coloring a more detailed image such as the coal fire or heptane data sets as in Fig. 7c results in more information but users may prefer to see only intensity variations. Fig. 14 shows a comparison

TABLE 1  
Video Memory Usage and Octree Construction Time  
for Various Data Sets

Dataset	Data Size (Megabytes)	Memory (Megabytes)	Octree Build (Seconds)
Combustion (480x720x100)	138.24	589.62	3.12
Heptane (293x293x293)	100.62	429.71	0.45
Helium (227x302x302)	82.81	354.05	0.40
Coal Fire (402x162x162)	42.20	181.45	0.59
X38 (256x256x256)	67.11	287.31	0.16

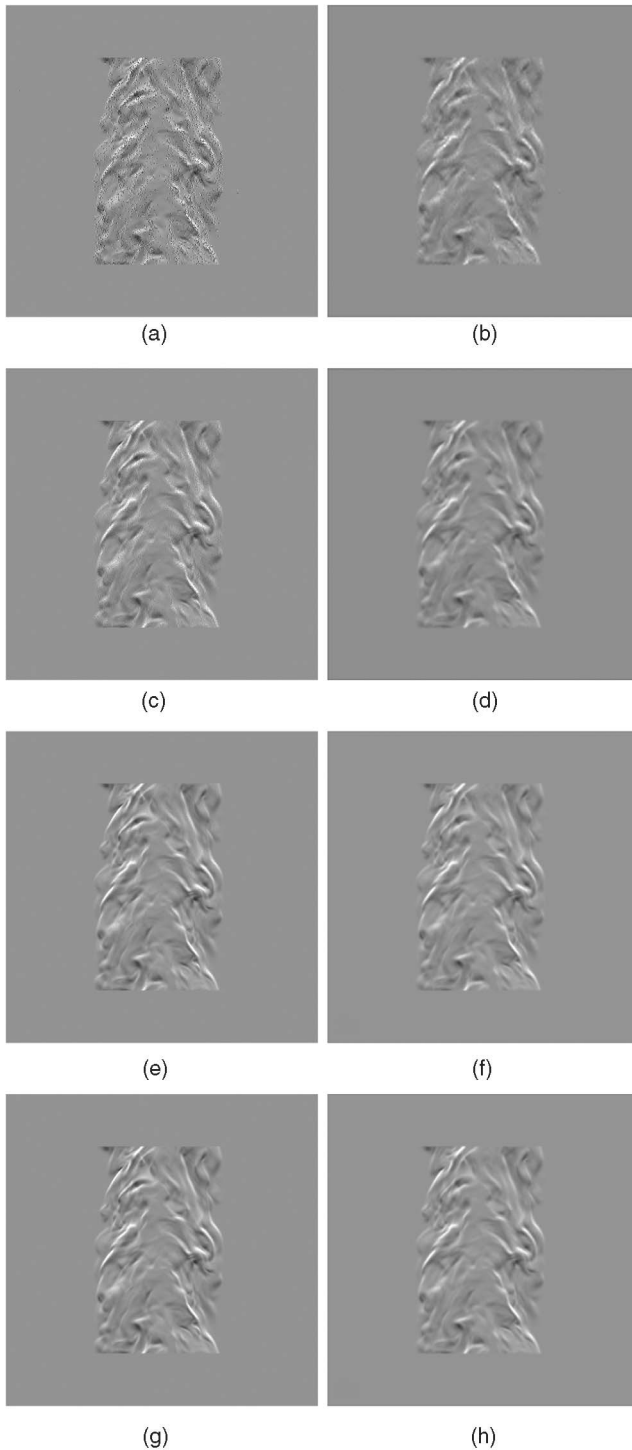


Fig. 15. Comparison of unfiltered film plane with 1, 10, 100, and 1,000 samples per pixel (a, c, e, and g) and the corresponding images of the film plane filtered with a cone filter in (b, d, f, and h).

of our method with Svakine et al.'s method [25] using the X38 data set. 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.

Fig. 15 shows an image of a combustion data set rendered with and without filtering at different samples-per-pixel (spp). The left column is without filtering while the right column is filtered using a cone filter described in Section 6. For the unfiltered images the root mean square

differences, RMS, between 1 and 10 spp, Figs. 15a and 15c, is 7.75 percent while the RMS between 10 and 100 spp, Figs. 15c and 15e, is 2.78 percent. The RMS between 100 and 1,000 spp, Figs. 15e and 15g is just 0.91 percent revealing a very small change beyond 100 spp. Filtering is very beneficial when rendering with a small photon count. To illustrate this, the RMS between 1 and 10 spp filtered, Figs. 15b and 15d, is 0.83 percent while the RMS between 10 and 100 spp filtered, Figs. 15d and 15f, is 2.55 percent. The RMS between 10 to 100 spp filtered, Figs. 15f and 15h is only 0.176 percent. Recall that the cone filter expands its width to encapsulate a set number of samples. Thus, as the sample rate increases, the blurring cone decreases. In effect, above 100 spp, there is little or no blurring taking place.

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 approximately one second (at least 100 iterations or samples-per-pixel) there was little discernible improvement in image quality with additional time for  $512 \times 512$  images as supported by the RMS terms.

Video Memory usage and octree construction time on the CPU are listed in Table 1. The memory requirements are made up of the pixel buffer for a  $512 \times 512$  image, the random number array, the gradient, and the refractive indices. Single variables passed to the GPU and locally declared variables are ignored in the memory requirements. When working with large data sets memory usage can be mitigated by computing gradient values on the GPU at runtime as the gradient makes up a large portion of the memory usage. The octree construction times are for single-threaded calculation and memory allocation on the CPU.

## 10 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 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 [20]. 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.

Combining multiple data fields into a single refractive index has been explored, however, there are other possible methods that might be used. One such method of exploring multifield data may be to accumulate a color value per field similarly to a volume renderer. Refraction could then be based on another field with a simple grayscale modifier such as a knife-edge cutoff applied such that one field produces a color value while the other produces a shadowgraph or schlieren effect and the resulting image is a convolution of the two techniques. Yet another method would be to have each data field traced independently with separate cutoffs applied and then the resulting images combined later on. These are merely some of the proposed methods for such a complex topic which might be usefully investigated.

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. This is especially important for interferometry where a light source with uniform wavelength should be chosen. Finally, only purely refractive flows have been investigated so far, but simulating scattering effects, emission, absorption, and polarization may also be necessary depending on the materials used in the simulation [4]. 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.

## ACKNOWLEDGMENTS

The authors would like to thank Kelly Gaither for providing the x38 data and David Ebert for allowing them to reuse images from [25]. The authors would like to thank Gary Settles for images of shadowgraph and schlieren photographs. The authors would also like to thank Jeremy Thornock and Diem Nguyen from the Center for the Simulation of Accidental Fires and Explosions (C-SAFE) for providing the helium data. The authors would also like to thank Jamal Mohd-Yusof for his help and ideas for their paper. Additional thanks go to Tim McIntyre for the use of his interferometry example image and Mathias Schott for his assistance generating a volume rendering of the helium

data set. This publication is based on work supported by: DOE: VACET, C-SAFE Alliance Center; KUS-C1-016-04 awarded by King Abdullah University of Science and Technology (KAUST); the US National Science Foundation (NSF): CNS-0615194, CNS-0551724, CCF-0541113, IIS-0513212; and the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under contract DE-AC52-06NA25396.

## REFERENCES

- [1] M. Anyoji and M. Sun, "Computer Analysis of the Schlieren Optical Setup," *Proc. SPIE*, vol. 6279, pp. 62790M, 2007.
- [2] B. Atcheson, I. Irkhe, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, "Time Resolved 3d Capture of Non-Stationary Gas Flows," *ACM Trans. Graphics*, vol. 25, no. 5, p. 132, Dec. 2008.
- [3] M. Born, E. Wolf, and A.B. Bhatia, *Principles of Optics*, seventh ed. Cambridge Univ. Press, 1999.
- [4] D.K. Brayford, "Rendering of Polarized Subsurface Light Scattering," PhD thesis, Univ. of Manchester, 2006.
- [5] C. Brownlee, V. Pegoraro, P.S. McCormick, S. Shankar, and C.D. Hansen, "Physically-Based Interactive Schlieren Flow Visualization," *Proc. IEEE Pacific Visualization Symp.*, pp. 145-152, 2010.
- [6] J.H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E.R. Hawkes, S. Klasky, W.K. Liao, K.L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C.S. Yoo, "Terascale Direct Numerical Simulations of Turbulent Combustion Using s3d," *Computational Science and Discovery*, vol. 2, pp. 1-3, 2009.
- [7] P.E. Ciddor, "Refractive Index of Air: New Equations for the Visible and near Infrared," *Applied Optics*, vol. 35, pp. 15-66, 1996.
- [8] K. Gaither, "Visualization's Role in Analyzing Computational Fluid Dynamics Data," *IEEE Computer Graphics*, vol. 24, no. 3, pp. 13-15, May/June 2004.
- [9] D. Gutierrez, F.J. Serron, O. Anson, and A. Munoz, "Chasing the Green Flash: A Global Illumination Solution for Inhomogeneous Media," *Proc. Spring Conf. Computer Graphics*, pp. 97-105, 2004.
- [10] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel, "Eikonal Rendering: Efficient Light Transport in Refractive Objects," *ACM Trans. Graphics (SIGGRAPH '07)*, vol. 26, no. 3, pp. 1-9, Aug. 2007.
- [11] H.W. Jensen, "Global Illumination Using Photon Maps," *Proc. Seventh Eurographics Workshop Rendering*, pp. 21-30, 1996.
- [12] H.W. Jensen and P.H. Christensen, "Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps," *Proc. ACM SIGGRAPH '98*, pp. 311-320, 1998.
- [13] C. Johnson, R. Ross, S. Ahern, J. Ahrens, W. Bethel, K.-L. Ma, M. Papka, J. van Rosendale, H.-W. Shen, and J. Thomas, "Visualization and Knowledge Discovery: Report from the DoE/Ascr," *Proc. Workshop Visual Analysis and Data Exploration at Extreme Scale*, Oct. 2007.
- [14] F.T. Johnson, E.N. Tinoco, and N.J. Yu, "Thirty Years of Development and Application of Cfd at Boeing Commercial Airplanes, Seattle," *Computers and Fluids*, vol. 34, pp. 1115-1117, 2005.
- [15] S. Liu, J.C. Hewson, J.H. Chen, and H. Pitsch, "Effects of Strain Rate on High-Pressure Nonpremixed N-Heptane Autoignition in Counterflow," *Combustion and Flame*, vol. 137, pp. 320-339, May 2004.
- [16] T.J. McIntyre, I. Lourel, T.N. Eichmann, R.G. Morgan, P.A. Jacobs, and A.I. Bishop, "An Experimental Expansion Tube Study of the Flow over a Toroidal Ballute," *J. Spacecraft and Rockets*, vol. 41, no. 5, pp. 716-725, 2004.
- [17] W. Merzkirch, *Flow Visualization*. Academic Press, 1987.
- [18] NVIDIA, *CUDA Programming Guide*, <http://developer.nvidia.com/object/cuda.html>, 2009.
- [19] V. Pegoraro and S.G. Parker, "Physically-Based Realistic Fire Rendering," *Proc. Second Eurographics Workshop Natural Phenomena*, pp. 51-59, 2006.
- [20] G. Settles, *Schlieren and Shadowgraph Techniques, Visualizing Phenomena in Transparent Media*. Springer, 2001.
- [21] G. Settles, "High-Speed Imaging of Shock Waves, Explosions and Gunshots," *Am. Scientist*, vol. 94, no. 1, pp. 22-31, 2006.



- [22] A.J. Smits and T.T. Lim, *Flow Visualization: Techniques and Examples*. Imperial College Press, 2000.
- [23] M. Sun, "Computer Modeling of Shadowgraph Optical Setup," *Proc. SPIE*, vol. 6279, pp. 62790L, 2007.
- [24] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo, "Interactive Relighting of Dynamic Refractive Objects," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 35:1-9, 2008.
- [25] N.A. Svakhine, Y. Jang, D. Ebert, and K. Gaither, "Illustration and Photography Inspired Visualization of Flows and Volumes," *Proc. IEEE Visualization*, pp. 687-694, 2005.
- [26] L.A. Vasil'ev, *Schlieren Methods*. Keter, Inc., 1971.
- [27] F.J. Weinberg, *Optics of Flames*. Butterworths, 1963.
- [28] L.A. Yates, "Images Constructed from Computed Flow Fields," *Am. Inst. of Aeronautics and Astronautics*, vol. 31, no. 10, pp. 1877-1884, 1993.



**Carson Brownlee** received the BS degree in computer science from Gonzaga University, in 2006. He is a doctoral student with the Scientific Computing and Imaging Institute (SCI) at the University of Utah. His research interests are focused on massively parallel rendering for scientific visualization, ray tracing, and computer graphics.



**Vincent Pegoraro** received the PhD degree in computer science from the University of Utah, in 2009. He is a postdoctoral researcher in the Computer Graphics Laboratory (CGUdS) within the Department of Computer Science at Saarland University, Germany. As part of the Multimodal Computing and Interaction (M2CI) Cluster of Excellence, his research interests broadly include predictive rendering, efficient global illumination, photorealistic image synthesis,

computer graphics, and scientific visualization. As a research assistant at the Scientific Computing and Imaging (SCI) Institute, his dissertation focused on efficiently simulating physically based light transport in participating media within the framework of the Center for the Simulation of Accidental Fires and Explosions (C-SAFE).



**Siddharth Shankar** received the BS degree in information technology from the Cochin University of Science and Technology and the MS degree in computing from the University of Utah. He is currently a software engineer at TerraSim, Inc. While at Utah, he worked as a research assistant at the Scientific Computing and Imaging Institute. His interests are in the field of computer graphics, computer vision, and large-scale scientific visualization.



member of the IEEE and IEEE Computer Society.

**Patrick S. McCormick** received the MS degree in computer science with distinction from the University of New Mexico, in 1996. He serves as a team leader for the Applied Computer Science group at Los Alamos National Laboratory (LANL). Over his 17 year career at LANL, his research has explored topics revolving around high-performance computing, data analysis and visualization, and programming models for emerging processor architectures. He is a



Computing Laboratory (ACL) located at Los Alamos National Laboratory, where he formed and directed the visualization efforts in the ACL. He was a Bourse de Chateaubriand postdoc fellow at INRIA, Rocquencourt, France, in 1987 and 1988. He is a senior member of the IEEE.

**Charles D. Hansen** received the BS degree in computer science from Memphis State University, in 1981, and the PhD degree in computer science from the University of Utah, in 1987. He is a professor of computer science and an associate director of the Scientific Computing and Imaging Institute at the University of Utah. He was a visiting scientist at INRIA-Rhone Alpes in the GRAVIR group in 2004-2005. From 1989 to 1997, he was a technical staff member in the Advanced

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**