

Radiative Transport Based Flame Volume Reconstruction from Videos

Liang Shen, Dengming Zhu, Saad Nadeem, Zhaoqi Wang, and Arie E. Kaufman, *Fellow, IEEE*

Abstract—We introduce a novel approach for flame volume reconstruction from videos using inexpensive charge-coupled device (CCD) consumer cameras. The approach includes an economical data capture technique using inexpensive CCD cameras. Leveraging the smear feature of the CCD chip, we present a technique for synchronizing CCD cameras while capturing flame videos from different views. Our reconstruction is based on the radiative transport equation which enables complex phenomena such as emission, extinction, and scattering to be used in the rendering process. Both the color intensity and temperature reconstructions are implemented using the CUDA parallel computing framework, which provides real-time performance and allows visualization of reconstruction results after every iteration. We present the results of our approach using real captured data and physically-based simulated data. Finally, we also compare our approach against the other state-of-the-art flame volume reconstruction methods and demonstrate the efficacy and efficiency of our approach in four different applications: (1) rendering of reconstructed flames in virtual environments, (2) rendering of reconstructed flames in augmented reality, (3) flame stylization, and (4) reconstruction of other semitransparent phenomena.

Index Terms—Flame Volume Reconstruction, Flame Rendering, Flame Videos, Radiative Transport Equation, CCD Camera Synchronization

1 INTRODUCTION

Capturing and reconstructing fluids has been the subject of considerable research in computer graphics. Several capture-based methods have been developed for the reconstruction of flames [1], [2], [3], [4], gases [5], water surface [6], and mixing fluids [7]. Information extracted from the reconstruction is valuable for direct re-rendering, developing data-driven models, and combining with physically-based simulation methods to obtain improved results for a variety of applications [3], [8], such as re-simulation and detail enhancement. More importantly, we can deepen the understanding of the principle underlying specific fluid behavior through the captured data and model this real-world phenomenon more accurately. In this paper, we focus on flame volume reconstruction from captured videos.

Compared to other fluids, flames are more turbulent and noisy, and hence, high-end professional cameras were traditionally required to capture this rapidly changing phenomenon. The cost of this high-end equipment has previously been the bottleneck in the accurate volume reconstruction of flame data. With the rapid development of couple-charged device (CCD) consumer cameras, it is now possible to capture flame data using these relatively inexpensive devices. However, it is challenging to synchronize these CCD consumer cameras for accurate flame volume reconstruction. In this work, we introduce a novel method using a stroboscope to synchronize the CCD cameras (Fig. 1(a)) and obtain a good reconstruction from our captured flame data.

- L. Shen is with the University of Chinese Academy of Sciences, China.
E-mail: shenliang@ict.ac.cn.
- S. Nadeem and A.E. Kaufman are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794.
E-mail: {sanadeem, ari}@cs.stonybrook.edu.
- L. Shen, D. Zhu and Z. Wang are with the Institute of Computing Technology, Chinese Academy of Sciences, China.
E-mail: {shenliang, mdzhu, zqwang}@ict.ac.cn.

Previous solutions for flame volume reconstruction have used simplified linear rendering models with parallel projection. The use of a linear rendering model ignores the complex phenomena such as extinction and scattering during the reconstruction process. Parallel projection, on the other hand, is not how our eyes or photography normally work, and hence, this crude approximation leads to considerable errors in the final reconstruction.

In contrast, we use perspective projection and a more accurate rendering formation model based on the radiative transport equation (RTE) [9], which directly incorporates emission, extinction, and scattering in our reconstruction process. More specifically, we use perspective projection and RTE (Fig. 1(f)–(h)) to render different views for our flame volume data. These rendered views are iteratively compared against the synchronized captured frames (Fig. 1(e)) and refined to adjust the volume data until a certain threshold is reached.

The final flame volume data incorporates both the color intensity and the black-body radiation based temperature information from the captured video data. In essence, the rendered views are used to assign color intensities to the voxels in the volume data. Simultaneously, we use the black-body radiation model to build the color-temperature mapping and, consequently, to reconstruct the temperature values from the color intensities (Fig. 1(i)). In this paper, we focus on providing an accurate visual rendering of our reconstruction in various applications, and thus, the reconstruction of the true temperature will be the focus of future work (as discussed in Section 7).

In the implementation, we accelerate our volume reconstruction method using the Compute Unified Device Architecture (CUDA) parallel computing framework to achieve real-time performance, even with the use of the complex rendering model. We evaluate our method with simulated and real captured data, and demonstrate the efficiency and efficacy of our flame volume reconstruction in four different applications.

The contributions of this paper are as follows:

- A novel flame volume reconstruction method based on the radiative transport equation which allows modeling of complex phenomena such as emission, extinction, and scattering. Moreover, our method is GPU-accelerated and provides real-time performance.
- A consumer CCD camera synchronization technique.
- Four applications of our work: (1) rendering reconstructed flames in virtual environments, (2) rendering reconstructed flames in augmented reality, (3) flame stylization, and (4) reconstruction of other phenomena.

The paper is organized as follows. We discuss related works in the next section, followed by the details of our algorithm in Section 3 and its implementation in Section 4. We evaluate our algorithm and show additional results in Section 5. Then, we demonstrate the efficacy of our approach in four different applications in Section 6. Finally, we conclude with the current limitations of the proposed method and the avenues for future work in Section 7.

2 RELATED WORK

The reconstruction and modeling of flames has drawn significant attention in the fields of computer graphics and visualization. Various techniques have been proposed to solve this problem.

2.1 Computed Tomography

Computed tomography is widely used in medical imaging and requires thousands of input views for classic methods, such as filtered back-projection and algebraic techniques [10], [11], [12], [13], [14]. However, for common reconstruction problems, it is difficult to obtain a large number of input views and set up strictly parallel positions for the cameras. For our flame reconstruction, we focus on the sparse-view tomography problem for semitransparent phenomena which uses limited views as input. Some reconstruction methods have been proposed from this perspective, including the wavelet graph model [15], time-resolved Schlieren tomography for gas flows [5], and stochastic tomography for mixing fluids [7].

Our work is most closely related to the stochastic tomography method for mixing fluids [7], which dynamically changes the samples in the reconstructed volume based on a random walk algorithm. However, unlike stochastic tomography, our method is based on the radiative transfer equation, which describes the physical phenomenon of energy transfer in the form of electromagnetic radiation. We also apply an adaptive sample adjustment method and CUDA-based acceleration to our reconstruction process, which provides much faster convergence than the random walk method.

2.2 Flame Color and Temperature Measurement

Most flame measurement methods for research involve temperature sensors [16], [17], lasers [18], and special optical systems [19], [20], [21], [22], [23]. However, these methods fail to provide 3D reconstruction at a resolution high enough for realistic rendering of flames.

In recent years, the physical nature of flame illumination has been utilized using the mathematical black-body radiation model to aid in flame temperature reconstruction [24], [25], [26]. Unfortunately, the black-body radiation model by itself cannot

provide realistic visual flame results, which limits its application in the computer graphics domain.

In terms of flame color intensity reconstruction, density-sheet decomposition [1], [27], color-temperature mapping [4], and appearance transfer [3] methods have been introduced. These methods convert the volume reconstruction to parallel slices reconstruction and require a planar configuration of the input views; if the in-plane assumption is removed, then these methods do not work. The grid optical tomography approach [2], [28] was proposed to overcome this in-plane limitation, though it is not applicable to a more realistic rendering model.

2.3 Camera Synchronization

Consumer cameras have been used to reconstruct time-varying natural phenomena [5], [7]. The key obstacle in the use of consumer cameras is the synchronization problem, due to the lack of synchronization hardware.

Previous works on the synchronization of multiple video sequences are based on feature tracking and geometric constraints [29], [30], [31]. A different method based on detecting flashes [32] provided frame-level synchronization. Recently, to solve the issue of rolling shutter shear and the synchronization problem of complementary metal-oxide-semiconductor (CMOS) consumer-grade camcorders, a method based on strobe illumination and the subframe warp method was proposed [33].

Unfortunately, there are no obvious features that can be tracked in the flame videos. Moreover, the flame changes rapidly and irregularly, and therefore the synchronization accuracy of the frame-level or the sub-frame warp [33] is unacceptable for the reconstruction of the flame.

3 ALGORITHM

In this paper, we present a method for flame volume reconstruction from videos. Fig. 1 shows the pipeline of our approach. We present a smear-based CCD camera synchronization method to capture flame videos using multiple cameras. For each video frame, we reconstruct color and temperature using an iterative approach.

For color reconstruction, the red, green, and blue (RGB) channels are reconstructed separately. In essence, the volume data is first initialized with zeros, and for each iteration, the RTE based rendering model is used to render the reconstructed images. After comparing the captured video frames with the reconstructed images, we adjust the volume data using the residuals of the pixel intensities. The iteration process ends when the error difference between two successive iterations falls below 0.01. The reconstruction is implemented using the CUDA parallel computing framework. The CUDA acceleration and the iterative approach provide a real-time visualization of the reconstruction results after every iteration.

For temperature reconstruction, the black-body radiation model is used to build the color-temperature mapping. Following the same steps in the color reconstruction, we achieve real-time visualization of the reconstruction results after every iteration by simply looking up the computed color-temperature mapping.

3.1 Smear-Based CCD Camera Synchronization

To solve the multi-view reconstruction problem, synchronized flame videos need to first be captured. Traditionally, high-end industrial cameras containing synchronization hardware were used

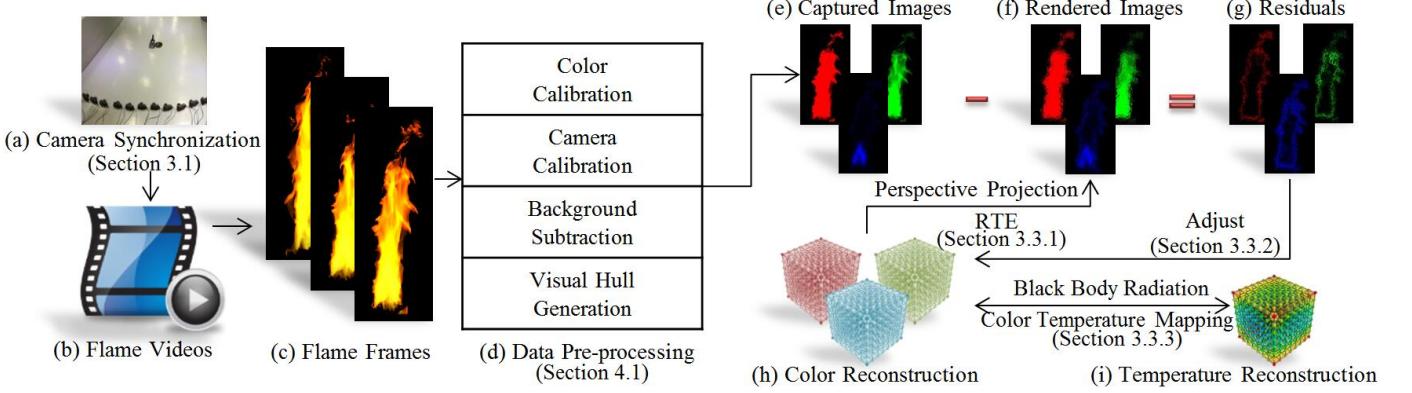


Fig. 1: The pipeline for volume reconstruction of flames from videos. With a smear-based CCD camera synchronization method (a), flame videos (b) are captured and then converted to frame sequences (c). For each frame, after data pre-processing (d), the captured images (e) from different views are compared to the rendered images (f), which are the rendering results of the color volume data (h) using perspective projection and RTE. The residuals between the captured and rendered images are used to adjust the volume data (h). The comparison and adjustment are carried out iteratively to finish the color reconstruction. For the temperature reconstruction (i), we use black-body radiation to build the mapping between the color and the temperature. Specifically, the temperature reconstruction is directly related to the green channel reconstruction, as the temperature values are mapped to the green values in our method.

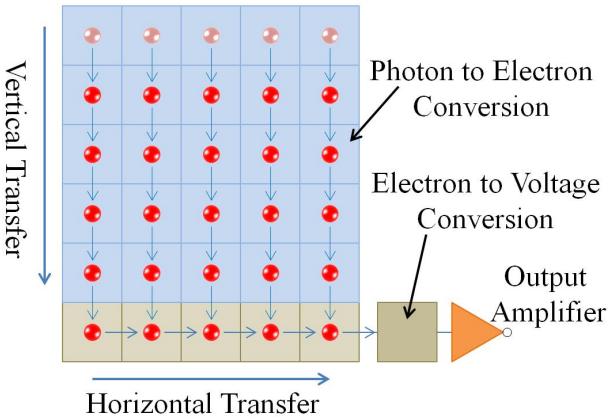


Fig. 2: CCD structure. The row-by-row vertical transfer leads to the generation of smear.

for this purpose. However, these cameras are expensive, costing at least 700 dollars per camera. In our work, we utilize cheap consumer CCD cameras (that can cost as low as 100 dollars per camera) to achieve state-of-the-art flame volume reconstruction results in real-time. These cameras, however, do not contain any synchronization hardware. Therefore, we present a method to synchronize these inexpensive consumer CCD cameras using the smear phenomenon in them [34]. More specifically, we achieve synchronization by resetting the camera shutter until a fixed accuracy is obtained.

3.1.1 CCD Smear

Common architectures of CCD image sensors include full-frame, frame-transfer, and interline architectures. All of these architectures follow similar imaging processes as shown in Fig. 2. More specifically, the cells of the CCD sensor convert the gathered photons to electrical charges. These charges are vertically transferred to the horizontal readout line row-by-row. For each row, after the horizontal transfer process, the charges are then converted to voltage information, and finally the digital data for the image is retrieved through the amplifier. The final image is generated by the same operations for all rows of the cells in the sensor.



Fig. 3: CCD smear. (a) The smear of a light source with constant lighting. (b) The smear of a strobe light source.

If an intense light source is imaged onto the CCD image sensor, undesired signals appear as a bright vertical (from top to bottom) stripe emanating from the light source location in the image. The undesired brighter section around the light source which is the result of the overflow of charge is called “blooming”, and the vertical brighter section is called “smear”. Smear is produced by the incident light during the vertical transfer process. Fig. 3 shows different smear patterns for different light sources. When the light source emits a light with constant lighting, the cells of the CCD chip continue to capture the intense light during the vertical transfer process, which leads to the generation of a vertical bright line along the light source position. Similarly, when a strobe light source is used, the smear line would change to smear dots, since the strobe would only shine intermittently.

3.1.2 CCD Camera Synchronization

A video camera uses two procedures, acquisition and read, to produce one image frame. In the acquisition phase, the cells of the CCD sensor convert the gathered photons to electrical charges. Then, in the read phase, the charges are vertically transferred to the horizontal readout line row-by-row. To describe the synchronization method, we focus on the time t_{image_read} taken to transfer the pixels for the final image. Assuming the resolution of the image is $m \times n$, we get:

$$t_{image_read} = nt_{per_row}. \quad (1)$$

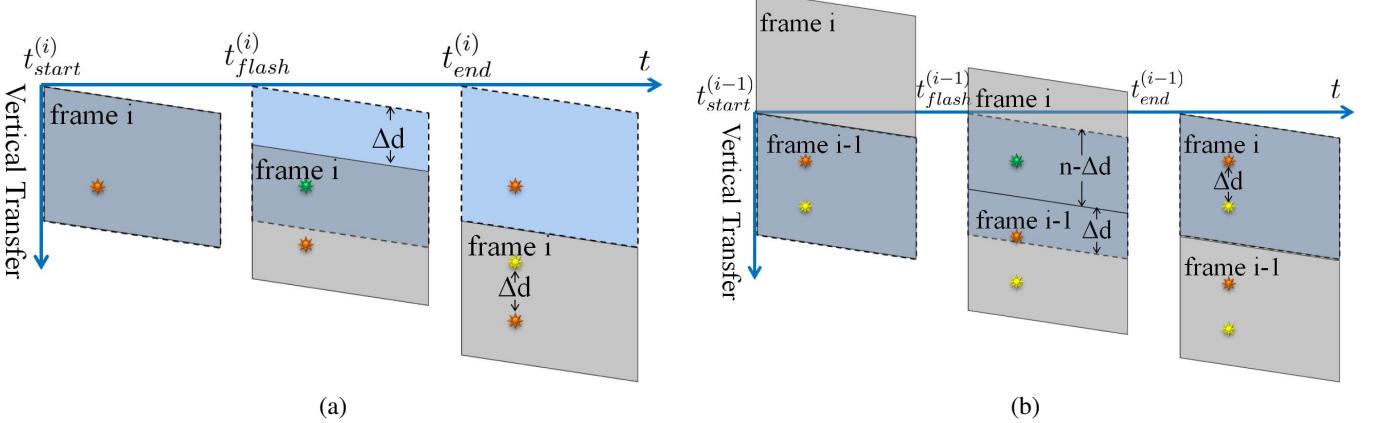


Fig. 4: For different flash times, the smear dots would appear either above or below the light source. Scenarios (a) and (b) present the process of these two situations. An orange dot denotes the position of the light source in the images, and the yellow one represents the smear. When the strobe turns on, the light source is shown as a green dot.

Here, t_{per_row} denotes the time needed to transfer one row of the image, and is used to measure the error of the synchronization later.

To implement our synchronization method, we use CCD cameras to capture images of a strobe, which has a controllable flash rate. When we set a much higher flash rate than the CCD frame rate, there will be several bright dots (smear) in one image, as shown in Fig. 3(b). We can use two adjacent dots to compute t_{per_row} as:

$$t_{per_row} = \frac{1}{\Delta d_{smear} f_{flash}}. \quad (2)$$

Here, Δd_{smear} denotes the distance in rows between two adjacent bright dots on the same side (above/below) of the light source in the image, and f_{flash} denotes the flash rate which is given by the strobe instrument.

When the flash rate is lowered, the number of bright dots in the image decreases. When the flash rate equals the video frame rate, the number of bright dots are either zero (the strobe turns on in the image acquisition phase) or one (the strobe turns on in the image read phase). In the single bright dot case, the dot could be either above or below the light source. If the smear dot is above the light source, the smear is generated during the read phase of the current frame, as shown in Fig. 4(a). The distance in rows, $\Delta d^{(i)}$, between the bright dot and the light source for the i th frame can be expressed as:

$$\Delta d^{(i)} = \frac{t_{flash}^{(i)} - t_{start}^{(i)}}{t_{per_row}}, \quad (3)$$

If the smear dot is below the light source, as shown in Fig. 4(b), the smear is generated during the read phase of the last frame:

$$\Delta d^{(i)} = n - \frac{t_{flash}^{(i-1)} - t_{start}^{(i-1)}}{t_{per_row}}. \quad (4)$$

Here, $t_{start}^{(i)}$ denotes the time needed for the i th frame to start the transfer, and $t_{flash}^{(i)}$ is the time needed for the strobe to turn on during the read phase of the i th frame, which results in the smear dot.

For the synchronization of multiple video cameras, we need $t_{start}^{(i)}$ for all cameras to be the same. From Equations 3 and 4, $t_{start}^{(i)}$ is determined by $\Delta d^{(i)}$, $t_{flash}^{(i)}$, and t_{per_row} . We make the

cameras capture the same strobe images, so that the $t_{flash}^{(i)}$ is the same. Inexpensive cameras of the same model still have good accuracy and stability with respect to the frame rate, so t_{per_row} stays consistent.

Therefore, synchronization can be achieved by the following settings:

- Set the flash rate of the strobe to the same value as the frame rate of the cameras.
- Keep the smear dot on the same side of the light source for all camera images.
- Adjust the smear dots to make them equi-distant from the light source.

We have provided a detailed animation of the synchronization process in the supplementary video.

3.1.3 Data Capture

To implement our method for flame reconstruction, we use eleven Canon PowerShot G12 cameras to capture RGB videos of flames with a resolution of 1280×720 , as shown in Fig. 5. In order to easily calibrate the cameras, the cameras are set up in an arc-like topology to ensure that all of the cameras see the calibration board simultaneously. Moreover, this topology prevents cameras from capturing images of each other. Unlike previous works [1], [4], [27], the camera positions are not restricted to be on a strict plane, which makes our approach more practical. The distance between the flame and each camera is approximately one meter. All of the cameras are set to the same configuration to obtain the same response to the incident light. During the experiments, we ignite paper and alcohol with different particles to generate different flames.

For the smear-based synchronization method, we use a Monarch Instrument Nova-Strobe dbx stroboscope as the light source. The dbx has flash rates ranging from 30 to 20000 flashes per minute that are adjustable in 0.1 step increments. We set the rate to 23.98 flashes per second, which is the same as the video frame rate.

The distance between the smear dot and the light source is determined by the start time of the shutter, so we adjust the start time of the shutters to make the distances for each camera almost the same. Consequently, we achieve the synchronization of

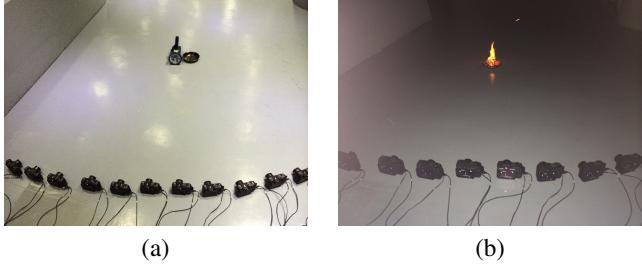


Fig. 5: Data capture setup. (a) The scene of camera synchronization using a stroboscope. (b) A scene of the flame videos capture.

the cameras as described in Section 3.1.2. For the G12 camera, the start time of the shutter can be adjusted using the button for switching between different resolutions in the video mode. Since this adjustment requires manual intervention, we do not expect to obtain the exact same distances for each camera. However, we can easily set a distance within a 100-pixel offset within 5 trials for each camera. The resultant time to transfer one row of the pixels is about $54 \mu\text{s}$ using Equation 2. Therefore, we can easily control the accuracy of our synchronization within $100 \times 54 \mu\text{s} = 5.4\text{ms}$, much less than the frame-level synchronization, $1\text{s} \div 23.98 = 41.7\text{ms}$. We can achieve more accurate synchronization if the 100-pixel offset distance for each camera is reduced even further with more trials.

3.2 Rendering Model

In this section, we provide a concise introduction to the rendering model used in this paper. Instead of the simplified linear optical model [1], [2], [4], [27], [28] or parallel projection [1], [4], [27], we use a radiative transport equation (RTE) based rendering model and perspective projection for the reconstruction.

The RTE is an integro-differential equation which describes the light transport in participating media, as follows:

$$(\vec{\omega} \cdot \nabla) L(\vec{x}, \vec{\omega}) = +\sigma_a(\vec{x}) L_e(\vec{x}, \vec{\omega}) - \sigma_t(\vec{x}) L(\vec{x}, \vec{\omega}) + \sigma_s(\vec{x}) \int_S L(\vec{x}, \vec{\omega}_i) \cdot p(\vec{\omega}, \vec{\omega}_i) d\omega_i. \quad (5)$$

Here, the left side of the equation denotes the variation of the spectral radiance in the direction of interest $\vec{\omega}$. L is the spectral radiance, L_e the emitted spectral radiance, σ_a the absorption coefficient, σ_s the scattering coefficient, $\sigma_t = \sigma_s + \sigma_a$ the extinction coefficient, and $p(\vec{\omega}, \vec{\omega}_i)$ the phase function, which specifies the spherical distribution of the scattering light at a position.

The optical model [35] for volume rendering is described as follows:

$$E(s_e) = \int_{s_b}^{s_e} T(s, s_e) L(s) ds + T(s_b, s_e) E(s_b). \quad (6)$$

Here, $E(s)$ denotes the intensity at position s , $T(s_1, s_2)$ is the transparency from s_1 to s_2 , s_b is the position of the background, s_e is the position of the eye point, and $L(s)$ is the radiative contribution at position s .

3.3 Volume Reconstruction

In terms of the goal of flame volume reconstruction from video frames, reconstructing the color intensity [1], [2], [27] is intuitive.

However, some works have also focused on temperature reconstruction [4], [25]. In this work, we reconstruct both color and temperature using our method. Instead of building a linear system [2] or designing a random walk framework [7], we dynamically adjust the volume data in an iterative way. To explain our method clearly, we provide the following definitions:

Key points: Vertices of voxels in the reconstructed volume. Each key point contains a value and all key points constitute the reconstructed volume. The goal of our method is to reconstruct the values of the key points.

Sample points: Points used to render the flames and to adjust the value of their eight neighboring key points during the adjustment phase.

3.3.1 Volume Rendering

Given the volume data, we use the rendering model described in Section 3.2 to render the reconstructed images. When traveling through the flame, traced rays would be curved [36], [37]. As we deal with small scale flames at present, we ignore this effect for simplification. Rays are cast into the volume, and sample points on each ray are used to determine the final intensity of the corresponding pixels. We calculate the radiance of the sample points, and then integrate them to obtain the pixel color. In order to integrate the radiance of sample points on one ray, we apply under blending [38] in front-to-back order using the following equations:

$$L_{dst} = T_{dst}(T_{src}L_{src}) + L_{dst}, \quad (7)$$

$$T_{dst} = (1 - T_{src})T_{dst}. \quad (8)$$

The radiance of sample points is subject to the emission, extinction, and scattering phenomena, corresponding to the three summands in the right side of Equation 5. The description of these phenomena is as follows.

Emission: During the process of combustion, some photons are emitted by the hot soot particles and the emission results in the light of flames [35]. We directly model the volume data as color intensities which are used for the emission phenomenon.

Extinction: While traveling through the flame, some photons are absorbed or scattered in different directions. This is called extinction. The extinction coefficient σ_t in Equation 5 is relative to the number of particles in a voxel [35]. The sophisticated optical models specify the extinction coefficient as a finite, continuously varying function of the volume data. Here, we assume σ_t to be a constant for simplification.

Scattering: The scattering phenomenon is defined as the process by which photons travel from the direction of interest to different directions (out-scattering), or from different directions to the direction of interest (in-scattering). The distribution of scattered light is specified by the phase function

$$p(\vec{\omega}, \vec{\omega}_i) = \frac{1}{4\pi} \cdot \frac{1 - g^2}{(1 + g^2 + 2g(\vec{\omega} \cdot \vec{\omega}_i))^{\frac{3}{2}}}. \quad (9)$$

Here, g is a value in the range $(-1, 1)$, and $g < 0$, $g = 0$, and $g > 0$ correspond to backward, isotropic, and forward scattering, respectively. Here, we use an isotropic phase function for simplification [39]. As described in [40], in a low-albedo medium such as the flames of interest, extinction usually dominates largely over scattering which contributes very little to the final image. Moreover, we opt for isotropic scattering because of the computational efficiency.

3.3.2 Volume Adjustment

Tomographic reconstruction approaches [1], [2], [4] minimize the residuals of intensities between the input and the reconstructed images to drive the reconstruction process. We follow a similar residual intensity minimization approach for our reconstruction.

After the reconstructed image is generated by the volume rendering process, a ray then travels through the reconstructed volume and several sample points are generated. Each sample point is used to adjust the value of its eight neighboring key points based on the sample point transparency, the distance between the sample point and the key point, and the residuals between the captured and the reconstructed images. The adjustment is done as follows:

$$A_{j,s} = \alpha_l \Delta L(x, y) T(s, s_e) / e^{\alpha_d D(j, s)}, \quad (10)$$

$$V_{j,k+1} = V_{j,k} + \sum_s G(s) A_{j,s}. \quad (11)$$

Here, $V_{j,k}$ denotes the volume data of the key point j after k iterations, and $A_{j,s}$ denotes the value of adjustment for the key point j influenced by the sample point s . As shown in Equation 11, for each iteration, we adjust the volume data of the key point j by adding each $A_{j,s}$ weighted by a random number $G(s)$, which is generated from a Gaussian distribution with mean equal to 1. For $A_{j,s}$, we consider three aspects, as shown in Equation 10:

- $\Delta L(x, y)$: the residual of color intensity between the source and the reconstructed image at pixel (x, y) .
- $T(s, s_e)$: the transparency from the sample s to the eye position. This is used to balance the adjustment of different regions in the reconstructed volume.
- $D(j, s)$: the distance between key point j and sample point s .

α_l and α_d are scale factors employed to describe the quantity of each part which contributes to the adaptive adjustment value.

We discuss each part in detail in the following sections.

Color Constraint: The ultimate goal of the reconstruction is to produce flame frames with the same color as the captured frames. Therefore, the differences of pixel intensity between the captured frames and the reconstructed frames are used to reduce the residuals between the input and reconstructed images. The residual is given as $\Delta L(x, y) = L_{src}(x, y) - L_{rec}(x, y)$, where $L_{rec}(x, y)$ denotes the intensity of pixel (x, y) in the reconstructed image, and $L_{src}(x, y)$ the intensity of pixel (x, y) in the captured source frame.

Transparency: Given that we use the under operator to blend the samples in front-to-back order, assuming the transparency for all samples to be a constant $\tau \in (0, 1)$ and k is the order of the sampling for a ray, the blending transparency of the k th sample is computed as follows:

$$T(s, s_e) = (1 - \tau)^{k-1}, \quad (12)$$

which is in the range $(0, 1)$ and decreases exponentially with the sample order k . The longer the distance between the sample and the eye, the less contribution the sample makes towards the final intensity. Therefore, $T(s, s_e)$ is used to scale the adjustment value.

Distance Constraint: It is apparent that the samples closer to the key points would cause greater changes to the volume data than the values of the key points. We use the Euclidean distance between key point j and sample point s to evaluate the adjustment value for the distance constraint part. The transformation $1/e^{D(j, s)}$ constrains this part in the range $(0, 1]$.

3.3.3 Temperature Reconstruction

In order to reconstruct the black-body radiation based flame temperature, we first build the color-temperature mapping [41], and then follow the same steps as the color reconstruction for the green channel, as described above. Meanwhile, to provide the visualization results after every iteration, we look up the color-temperature mapping on the fly for rendering.

We select the green channel for temperature reconstruction because of the following reasons:

- When judging the relative luminance of different colors in well-lit situations, humans tend to perceive light within the green parts of the spectrum as brighter than the red or blue lights of equal power [42].
- Based on the color mapping methods [41], [42], we found that the intensity of the green channel grows with the temperature within a certain range.

Therefore, the relationship between the green channel and temperature is confirmed based on the observation and the computational results. The color-temperature mapping is created as follows. In the local thermodynamic equilibrium (LTE) system, the black-body radiation law provides that the electromagnetic spectral radiance emitted by the medium (black-body) is directly indicated by the temperature of the medium of a given distribution of wavelength [43], which fits the situation in most fires. Ideally, we assume these kinds of fire act as a black-body and compute the emitted spectral radiance using Planck's formula:

$$L(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1}. \quad (13)$$

Here, T denotes the temperature, k is the Boltzmann constant, h is the Plank constant, and c is the speed of light in the medium. This spectrum can then be converted to RGB [4], [40], [41], building the color-temperature mapping.

4 IMPLEMENTATION

In this section, we present the details of the data processing and acceleration techniques used in our algorithm.

4.1 Data Pre-processing

The flame videos are captured in a dark environment. For the data pre-processing, we split the videos into individual frames. Pixels with intensities greater than 30 are set as flame pixels, and the non-flame pixels are assigned the color black. After the color calibration [44], the processed frames are used as input to our algorithm. To map the camera coordinates to the object (flame) coordinates, the standard techniques for camera calibration [45] are used to compute the intrinsic and extrinsic parameters of cameras.

4.2 Sampling Step and Volume Dimension

The 3D volume consists of voxels and an appropriate sampling step needs to be set to access all these voxels. Since we use perspective projection in the reconstruction, only a few rays vertically intersect with the reconstructed volume cube. Besides, we try to deal with small scale flames, hence few rays intersect with the volume with large angles. If the sampling step is large, then some volume data will be missed along the ray. In contrast, if the sampling step is small, then duplicate data will be accessed.

Therefore, we set the sampling step equal to the voxel edge length, so that samples on the same ray would evenly access one voxel ideally.

To determine the volume dimensions, we first measure the position and size of the flames in the world coordinate system, which is determined by a calibration board [45]. Then, given the camera parameters, we calculate the projection pixels of the eight vertices of the volume. Ideally, we obtain twelve projected lines l_I corresponding to the twelve edges of the volume cube for each input image I . The lines l_I are then projected to the image uv coordinate, and each line $l_{(I,uv)}$ will occupy $N_{(I,I,u)}, N_{(I,I,v)}$ of pixels along the u and v dimensions, respectively. Let $n_{(max,d)}$ denote the maximum number of $N_{(I,I,u)}$ and $N_{(I,I,v)}$, and l_d , ($d \in \{x,y,z\}$) the length of the volume in dimension d . We set $\frac{n_{(max,d)}}{\alpha}$ as the number of voxels in dimension d . Then, the edge length of each voxel is $\frac{\alpha l_d}{n_{(max,d)}}$, and the dimensions of the other two directions can similarly be calculated. A large dimension leads to a waste of memory and some voxels might not be accessed by any ray. In contrast, a small dimension leads to low resolution of the reconstructed results. In experiments, we use $\alpha = 1.5$ to access all the voxels in our flame volume data.

4.3 CUDA-Based Acceleration

In order to accelerate our algorithm, we implemented the visual hull computation, volume rendering, and adjustment using CUDA.

4.3.1 Visual Hull

To reduce the number of volume values to be reconstructed, we use a visual hull [46] in our reconstruction method. We exploit the bitwise operations in the CUDA kernel to achieve good performance and obtain the visual hull within 200 milliseconds for ten input views and 128^3 voxels. As shown in Algorithm 1, we first initialize the visual hull tags of the volume with 0. Then, we assign the *imageIndex* from 0 to the number of input views (*cameraNumber* – 1), and the visual hull tags are updated with the fast bitwise operation of *imageIndex*. Only the rays cast from the flame pixels are used in this process. Finally, if the last *cameraNumber* bits of the voxel tag are all equal to 1, the voxel is inside the visual hull. Our reconstruction method only reconstructs the volume data inside the visual hull. We simply assign a small negative number to the key points outside the visual hull, which will not visually affect the final reconstructed results.

Algorithm 1: Visual Hull Computation on GPU

```

1: Initialize all visual hull tags of the volume with 0
2: for each input image do
3:   for flame rays in parallel do
4:     for each visualHullTag around the ray do
5:       visualHullTag |= (1 << imageIndex)
6:     end for
7:   end for
8: end for

```

4.3.2 Rendering and Adjustment

To visualize the reconstruction results after every iteration in real time, we apply GPU acceleration with CUDA in rendering and adjustment processes, as shown in Algorithm 2. The computation task for rendering is independent for each ray. Also, the process of adjusting the temperature field is highly parallel, except for

writing the adjusted value to the same key point from some rays, which could be solved using atomic operations.

In the implementation, the volume data is loaded in the 3D texture memory of the GPU, and the visual hull tags and the source frame data are loaded in the global memory of the GPU. In terms of the temperature reconstruction, the color-temperature mapping is loaded in the 1D texture memory of the GPU for fast access. In the rendering process, every ray cast from the camera optical center is computed by a CUDA kernel. Every kernel computes the positions of sample points, queries the sample values from the texture memory, and calculates the final intensity of the corresponding pixels. In the adjustment process, residuals of intensity, the transparency, and the distance between the key point and the sample point are applied to adjust the volume data using atomic operations.

Algorithm 2: Volume Data Rendering and Adjustment on GPU

```

1: // Rendering
2: totalrays ← image size
3: for ray < totalrays in parallel do
4:   for each sample point on the ray do
5:     calculate values from texture memory
6:   end for
7:   integrate the values
8: end for
9: // Adjustment
10: for flames ray in parallel do
11:   for each sample point on the ray do
12:     compute the adjustment value A of 8 key points
13:     add the As to the key points using atomic operation
14:   end for
15: end for

```

5 RESULTS

To evaluate our reconstruction algorithm, we analyze the performance of our approach on simulated and real captured data for both the color intensity and temperature reconstruction. Experiments are performed on a laptop with an Intel Core i7-4710MQ CPU 2.50GHz, with 8GB of memory, and an NVIDIA GeForce GTX 850M graphics card.

5.1 Color Intensity Reconstruction

We perform a quantitative evaluation of our method using simulated data, where the fire is modelled by a physically-based method [47]. Using the simulated data, we render multi-view flame images with the method, discussed in Section 3.2, and use the images as input to our reconstruction method.

We use ten input images from different views with intervals of 36° in a plane. In reality, the cameras can not strictly be constrained to a horizontal plane, (i.e., some cameras might look up at the flame and some might look down) and thus we set up the virtual cameras in such a manner. We render the flame images with a resolution of 640×480 . The simulated volume data is assumed to have a $20\text{cm} \times 20\text{cm} \times 20\text{cm}$ volume and the center of the volume is the origin. The cameras are set 30cm away from the origin with a field of view angle of 60° . In the reconstruction process, the coefficients in Equation 10 are set experimentally as $\alpha_l = 0.001$ and $\alpha_d = 1$. The value of τ in Equation 12 is chosen as 0.05. For

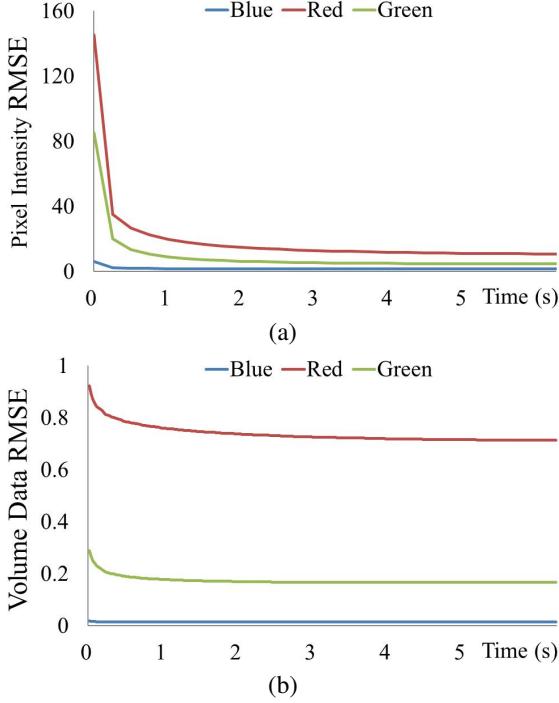


Fig. 6: (a) Relation between pixel intensity and the runtime of our method. (b) Relation between the volume data error and the runtime. Pixel intensity error and volume data error are efficiently reduced within two seconds.

flames of different sizes, we change the value of α_d proportionally with the length of the volume edge.

The difference between the input images and the reconstructed results is computed as pixel intensity error for the red, green, and blue channels, within the range [0, 255]. To compute the pixel intensity errors, we first specify a rectangle in the flame image as a bounding box (which contains the whole flame), and then we compute the root-mean-square error (RMSE) of all pixels in the bounding box. The difference in terms of the volume data between the ground truth and the reconstructed one is computed as the volume data error for the red, green, and blue channels. As we focus on the visual effect of the reconstruction, only the data in the visual hull is compared during the RMSE computation, and regions outside the visual hull are ignored.

We analyze the pixel intensity RMSE and the volume data RMSE with time as shown in Fig. 6. For each color channel, we visualize the results after each iteration with an average frame rate of 40.2 frames per second (fps). Our method can obtain acceptable reconstruction results within two seconds. Fig. 7 shows the visual results with the three channels synthesized after a series of iterations.

To perform extra evaluation, we use nine views (input views) of the rendered images from the ten total views as input to reconstruct the flame. After obtaining the reconstructed volume data, the image for the left view (test view) is rendered. We compare the results generated by our method and the algebraic tomography method of Ihrke and Magnor [2]. Fig. 8 and Table 1 show that our method achieves acceptable results for the input views, and for the test view, our approach achieves much better visual quality. The scattering improves the smoothness of the reconstructed results. Through the comparison, the superiority of the RTE model over the simplified linear optical imaging model is

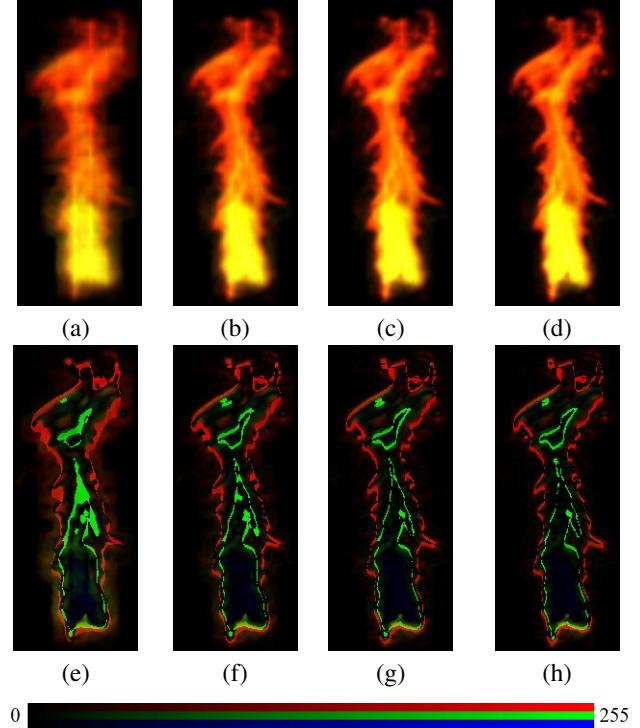


Fig. 7: Results of synthesizing the red, green, and blue channels after (a) 10, (b) 30, (c) 50, and (d) 70 iterations, and the corresponding errors (e), (f), (g) and (h). The errors for all channels are shown in a single image (such as (e), (f), (g) or (h)); the red, green and blue intensities represent the corresponding errors for the red, green and blue channels.

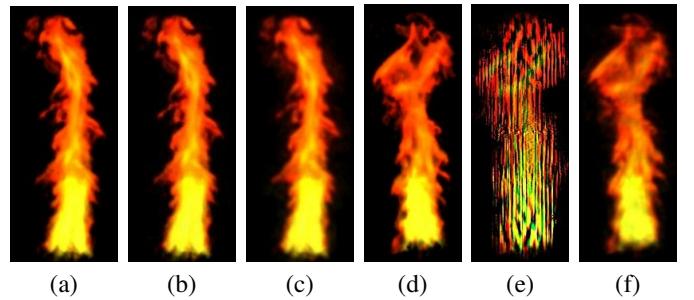


Fig. 8: Extra evaluation. (a) One of the input images, (b) and (c), reconstructed images by the algebraic tomography method and our method, respectively. (d) the ground truth of the test view, (e) and (f), results of the algebraic and our method, respectively.

TABLE 1: Reconstruction errors of extra evaluation

	Input View			Test View		
	Red	Green	Blue	Red	Green	Blue
Algebraic Tomography	4.2	2.3	3.0	79.3	47.9	28.9
Our Method	12.1	5.2	2.5	17.6	8.0	3.0

justified.

With the data captured from the setup described in Section 4.1, we use different fuels to generate different flames. We reconstructed these flames in a 152mm×152mm×152mm volume. Fig. 9 illustrates the reconstructed results for the captured flames.

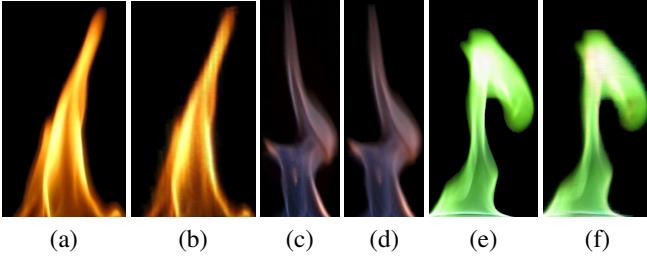


Fig. 9: (a), (c) and (e): input images; (b), (d) and (f): corresponding reconstructed images. (a) is generated by lighting paper fuel. (c) and (e) are generated by alcohol mixed with potassium chloride and boric acid, respectively.

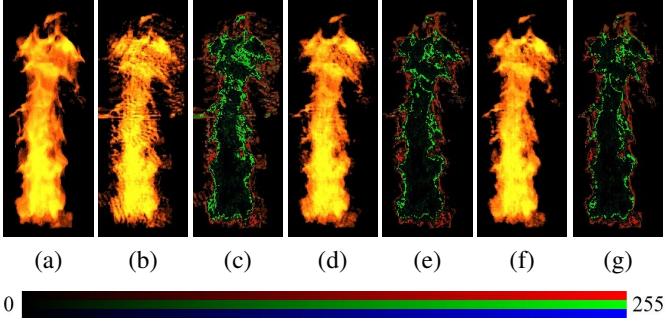


Fig. 10: Reconstructed results for simulated data using different numbers of input cameras. (a) The simulated ground truth image, the reconstructed results with (b) 4, (d) 16, (f) 28 input cameras, and the corresponding errors when using (c) 4, (e) 16, (g) 28 input cameras.

5.2 Temperature Reconstruction

Since the temperature of radiators with observable color (minimum visible red) is more than 800K [48], and the highest temperature achieved by burning our current fuels is limited, the temperature is assumed to be in the range (1000K, 2300K). Using simulated data, we analyze the reconstruction results achieved from different numbers of input cameras. The difference between the input image and the reconstructed image is computed as the pixel intensity error for the red, green, and blue channels, as shown in Figs. 10 and 11(a). The difference in terms of the temperature field between the ground truth and the reconstructed result is computed as temperature field errors, as shown in Fig. 11(b). As the number of input cameras increases, the errors decrease. The increase in the number of cameras leads to a tighter visual hull, which means that the region to be reconstructed becomes smaller.

With sixteen input views, the result of one cross section of the volume data is shown in Fig. 12. Without any constraints from fluid dynamics, the temperature values in the cross section seem coarse. However, we can still obtain visually plausible flame reconstruction results, as shown in Fig. 10(d).

For the captured flames, the black-body radiation based temperature reconstruction can only be applied on the flames with yellow colors. The reconstruction results are illustrated in Fig. 13. As shown in Table 2, the pixel intensity errors for the captured data are higher than the errors for the simulated data. The reasons for this include the accuracy of camera calibration, the limitation of the black-body radiation based color-temperature mapping, and the limited number of input cameras.

We also compare our method with the approach of Wu et al.

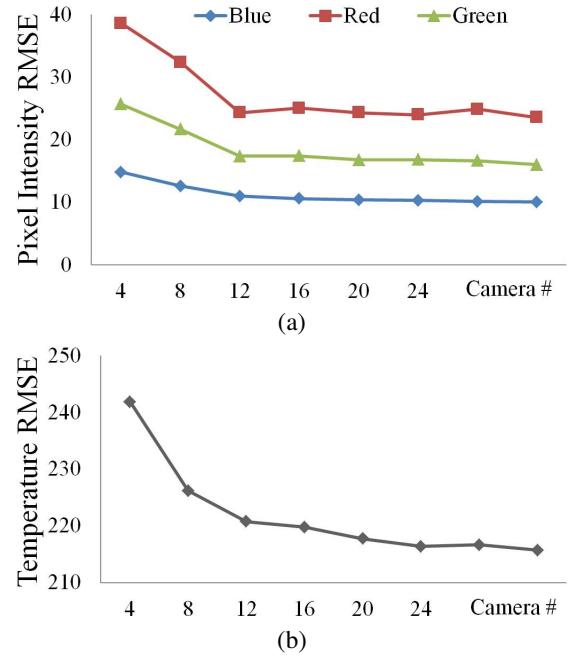


Fig. 11: Reconstruction errors with different numbers of cameras. (a) The intensity errors between Fig. 10(a) and reconstructed images with different numbers of input views. (b) The temperature errors between the ground truth and reconstructed result.

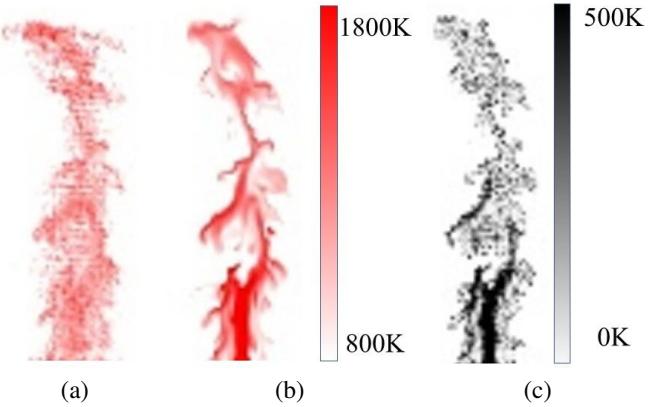


Fig. 12: Cross section of reconstructed result. (a) The temperature values of one cross section plane in the reconstructed volume, (b) the ground truth, and (c) the error plot between them.

[4]. We used four input images with a resolution of 200×310 and viewing angles of 0° , 45° , 90° , and 135° . As shown in Fig. 14, the results from the Wu et al. approach for the 0° angle view are rather poor. For the 45° angle view, since their method uses the image from this view to generate the visual hull, and no color information is exploited for the reconstruction, the reconstructed result is more noisy. Even though we exploit both color and temperature information, we are still able to achieve better time efficiency than their method, as shown in Table 3.

5.3 Performance Analysis

Table 3 shows an overview of different methods that have been proposed for flame reconstruction. Our method provides an economical way to capture flame data, and unlike other methods does not suffer from the in-plane constraint for the cameras. Moreover, we exploit a more accurate rendering model in our

TABLE 2: Color intensity errors of captured data.

Data Type \ Channel	Red	Green	Blue
	Simulated	25.2	15.2
Captured	25.8	18.6	23.9

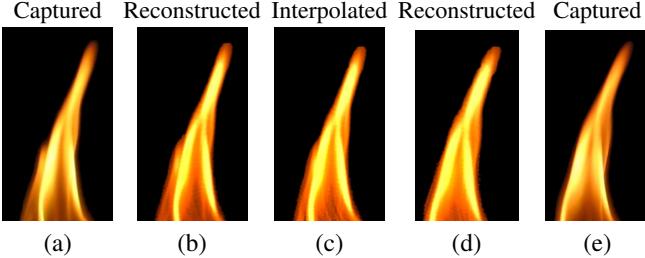


Fig. 13: Reconstructed results from captured images. (a) and (e) are the captured frames, and (b) and (d) are the corresponding reconstructed images. (c) is the intermediate reconstructed view between (b) and (d).

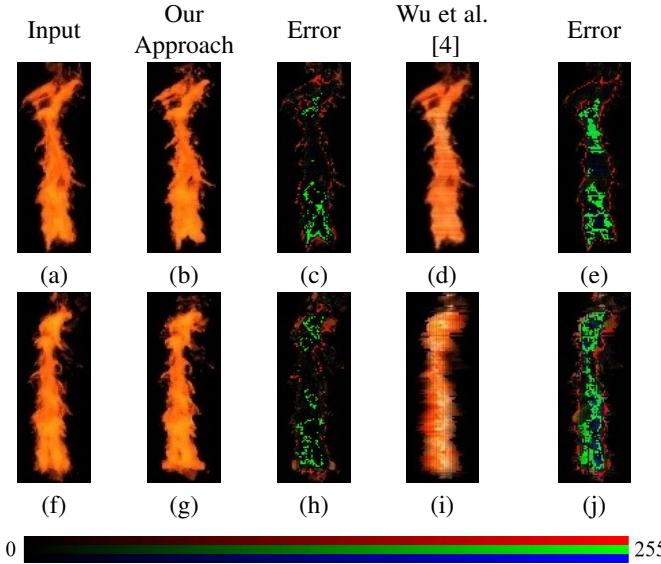


Fig. 14: Comparison with Wu et al. [4]. Similar to their method, we use four images as input, corresponding to 0° , 45° , 90° , and 135° angle views. Here we show the (a) 0° and (f) 45° angle input views. The corresponding reconstruction results for their method are shown in (d) and (i), and the respective errors are shown in (e) and (j). The results for our method are shown in (b) and (g) with the respective errors shown in (c) and (h).

reconstruction process. Finally, though our approach visualizes the reconstruction results after each iteration, our method can still achieve real-time performance during the reconstruction process (approximately 40fps on our laptop).

6 APPLICATIONS

Our proposed flame reconstruction method can be used in a variety of applications.

6.1 Reconstructed flames in virtual environments

Flame effects are widely used in movies and video games. Physically-based flame simulations are time-consuming due to the

use of computationally expensive solvers for the Navier Stokes equations. Our approach provides an effective way for modeling a 3D flame from several input images, and the flame volume data can then be reused in virtual environments. Since we capture real flame videos, the flames modeled by our method are more realistic than the simulations.

Given a virtual environment, reconstructed flames can be added into the scene using our method. Fig. 15(a) shows a frame from a bonfire scene using reconstructed data from simulated flames. Fig. 15(b) shows a frames from a fireplace scene using reconstructed data from a captured flame. Multiple flames can be combined as a flame group and set into the same scene as shown in Fig. 15(c). A complete animation of the reconstructed flames in virtual environment settings can be seen in the accompanying supplementary video.

6.2 Reconstructed flames in augmented reality

Augmented reality (AR) platforms have become prevalent in the last few years for training and educational purposes. More recently, with the interest generated by games such as Pokémon Go [49], the future of AR games seems bright. To cater for AR games and displays in the future, we have also demonstrated our flame reconstruction results in augmented reality settings. Given an augmented reality setting, we can render our reconstructed flames in the scene, as shown in Fig. 16. A complete animation of the reconstructed flames in an AR setting can be seen in the accompanying supplementary video.

6.3 Flame Stylization

In some flame effects, special objects may gradually come out from the flames, which could be treated as a case of volume stylization [39]. Color intensity reconstruction methods cannot achieve these effects since the objects will maintain their original color. However, with the temperature reconstruction workflow of our method, we can easily capture the flame stylization effects.

For all pixels of the object image, we first set the green channel values equal to the gray intensity, and then combine this “green” image with one of the input flame images. Since our method runs iteratively and visualizes the results after every iteration, the users can visualize the complete process of the object appearance in a flame style in the volume. Fig. 17 shows some stylization results. Since our method achieves real-time performance, the artists can observe the volume stylization effects immediately after changing the appearance of the input images. Moreover, given images from different views of an object, the flame stylization effects would make the object seem to be overheated or burned, as shown in Fig. 18.

6.4 Reconstruction of Other Phenomena

Using the color intensity reconstruction workflow, our approach can also be applied to reconstruct other semitransparent phenomena, such as smoke. We use a smoke dataset with 47 images from different views, with 3 views as input. The errors for the 3 input views and all the 47 views using our reconstruction approach and the sheet decomposition method [1] are shown in Fig. 19 and Table 4.

TABLE 3: Comparison of different methods for flame reconstruction.

Method	Camera Type	In-plane Constraint	Projection Model	Rendering Model	Iterative	GPU Accelerated	Time(s)***
Flame-sheet 2003 [27], 2007 [1]	Industrial CCD	Yes	Parallel	Linear	No	No	>60
Algebraic Tomography 2004 [2]	Industrial CCD	No	Perspective	Linear	Yes	No	>60
Stochastic Tomography 2012 [7]	Consumer CMOS*	No	Perspective	Linear & Splatting	Yes	No	>60
Color Temperature 2014 [4]	Industrial CCD	Yes	Parallel	Linear	No	No	0.91
Two-Color Pyrometric 2015 [24]	Industrial CCD	No	Perspective	Linear	No	Yes	>10
Refractive Reconstruction 2015 [25]	Industrial HDR CMOS	No	Perspective	Refractive	Yes	No	>60
Appearance Transfer 2015 [3]	Consumer CMOS**	Yes	Parallel	Linear	Yes	No	>60
Our Method	Consumer CCD	No	Perspective	Radiative Transfer	Yes	Yes	0.56

*The consumer CMOS cameras were used to capture mixing fluids [7], which does not require very precise synchronization.

**Only two cameras are needed in this method, so the synchronization can be achieved by trial and error. The response time of the same trigger signal may be different for each consumer camera. The purpose of this method is to create a fluid animation using the volume sequence modeled from a sparse set of videos as a reference. The precise reconstruction of the volume is not required.

***The time for reconstructing one frame of the video.

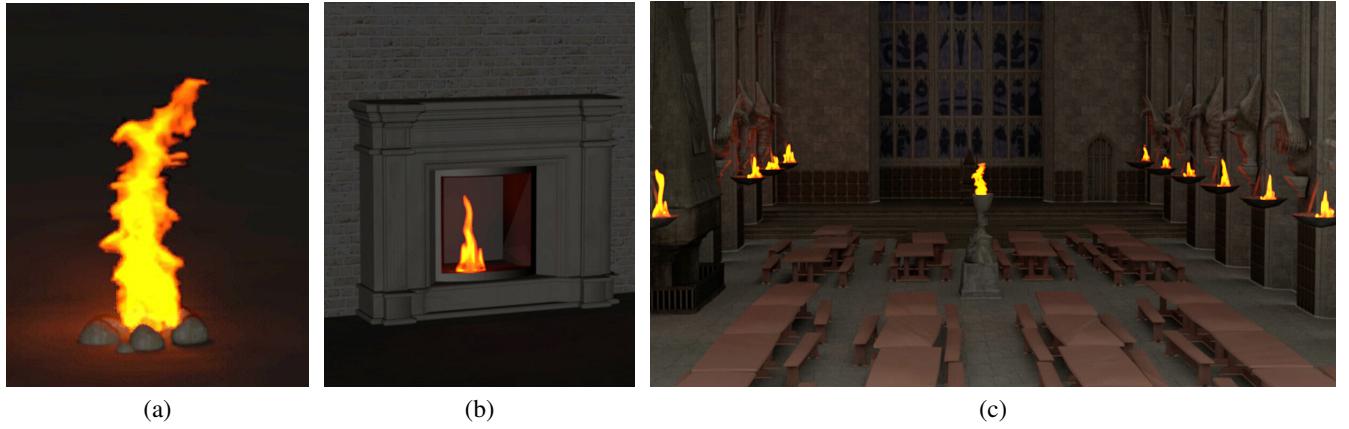


Fig. 15: Reconstructed flames in a virtual environment.

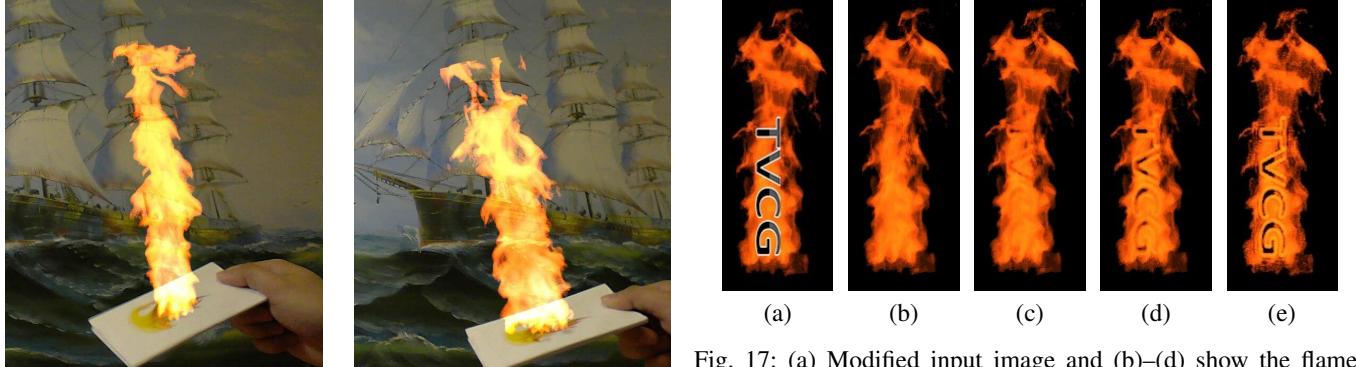


Fig. 16: Reconstructed flames in augmented reality.

7 CONCLUSION AND FUTURE WORK

In this paper, we have presented an algorithm for flame volume reconstruction. We have devised a novel camera synchronization

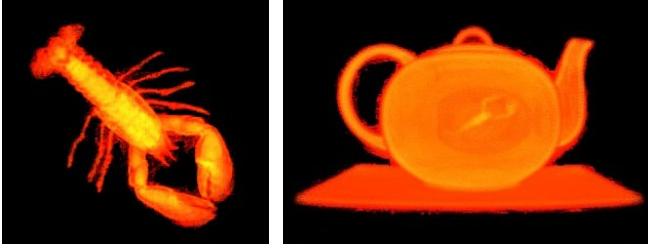


Fig. 18: The flame stylization of a lobster (left), and the flame stylization of a teapot with a small lobster inside (right).

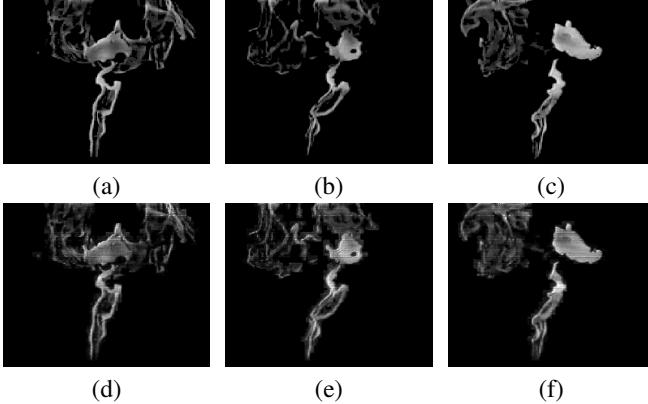


Fig. 19: Smoke reconstruction input images (top row) and the corresponding reconstructed images (bottom row).

method to capture flame data using inexpensive consumer CCD cameras. We also presented a novel reconstruction method that enables complex flame rendering models to be used in the reconstruction process. Moreover, we accelerated our reconstruction method using the GPU, which provides real-time performance for each iteration. Finally, we evaluated our method on both simulated and real captured data and demonstrated a variety of different applications for our approach.

There are two limitations of our current approach. The method currently can only deal with small-scale flames, and flames with complex internal structures cannot be accurately reconstructed. Since we currently deal with small scale flames, refraction is ignored when the rays travel through the volume and constant transparency for the volume data is used for simplicity. In the future, we will remove these constraints and create a more comprehensive picture of the flame volume data. We will also extend our approach to deal with reconstruction of large-scale fire and flames with complex internal structures.

In the case of temperature reconstruction, we assume the flame particles act as an ideal black-body, and little is considered in terms of the effects of different chemicals on rendering flame images, such as pink flames affected by lithium ion and green flames affected by copper ion. Thus, the temperature reconstruction cannot be applied on all kinds of flames. In the future, we will deal with this issue using a physically-based fire rendering approach [40].

We will also work on creating a user interface for artists and designers to edit special flame effects after every iteration of the reconstruction process. Since our approach can achieve real-time performance, allows visualization after every iteration, and provides visually plausible results, the artists/designers can immediately receive feedback after editing the input images and

TABLE 4: RMSE of smoke reconstruction with 3 input views.

Reconstruction method	RMSE	
	Input	All
Flame sheet decomposition	10.1	18.4
Our method	13.9	16.7

adjust the results accordingly.

ACKNOWLEDGMENTS

This research is supported by National High Technology Research and Development Program of China (863 Program) (Grant No. 2015AA016401), National Natural Science Foundation of China under Grant Nos. 61379085, 61173067 and 61532002, and US National Science Foundation grants CNS0959979, IIP1069147, CNS1302246, NRT1633299, and CNS1650499. The authors would like to thank the team of Prof. W. Wu at Beihang University for providing their code and helping us evaluate our method. The bonsai, teapot, and lobster volumetric datasets are courtesy IAPR-TC18 Digital Geometry repository.

REFERENCES

- [1] S. W. Hasinoff, K. N. Kutulakos, Photo-consistent reconstruction of semitransparent scenes by density-sheet decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (5) (2007) 870–885.
- [2] I. Ihrke, M. Magnor, Image-based tomographic reconstruction of flames, *Proceedings SIGGRAPH/Eurographics Symposium on Computer Animation* (2004) 365–373.
- [3] M. Okabe, Y. Dobashi, K. Anjyo, R. Onai, Fluid volume modeling from sparse multi-view images by appearance transfer, *ACM Transactions on Graphics* 34 (4) (2015) 93.
- [4] Z. Wu, Z. Zhou, D. Tian, W. Wu, Reconstruction of three-dimensional flame with color temperature, *The Visual Computer* (2014) 1–13.
- [5] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, H.-P. Seidel, Time-resolved 3D capture of non-stationary gas flows, *ACM Transactions on Graphics* 27 (5) (2008) 132–140.
- [6] C. Li, D. Pickup, T. Saunders, D. Cosker, D. Marshall, P. Hall, P. Willis, Water surface modeling from a single viewpoint video, *IEEE Transactions on Visualization and Computer Graphics* 19 (7) (2013) 1242–1251.
- [7] J. Gregson, M. Krimerman, M. B. Hullin, W. Heidrich, Stochastic tomography and its applications in 3D imaging of mixing fluids, *ACM Transactions on Graphics* 31 (4) (2012) 52–61.
- [8] J. Gregson, I. Ihrke, N. Thuerey, W. Heidrich, From capture to simulation-connecting forward and inverse problems in fluids, *ACM Transactions on Graphics* 33 (4) (2014) 139–149.
- [9] R. Siegel, *Thermal radiation heat transfer*, CRC press, 2001.
- [10] A. Andersen, A. Kak, Simultaneous algebraic reconstruction technique (SART): A superior implementation of the art algorithm, *Ultrasonic Imaging* 6 (1) (1984) 81–94.
- [11] M. Ehlke, H. Ramm, H. Lamecker, H.-C. Hege, S. Zachow, Fast generation of virtual x-ray images for reconstruction of 3D anatomy, *IEEE Transactions on Visualization and Computer Graphics* 19 (12) (2013) 2673–2682.
- [12] M. Fiddy, The radon transform and some of its applications, *Journal of Modern Optics* 32 (1) (1985) 3–4.
- [13] F. Natterer, F. Natterer, *The Mathematics of Computerized Tomography*, Springer, 1986.
- [14] A. A. G. Ramm, *The Radon Transformation and Local Tomography*, CRC press, 1996.
- [15] T. Frese, C. A. Bouman, K. Sauer, Adaptive wavelet graph model for bayesian tomographic reconstruction, *IEEE Transactions on Image Processing* 11 (7) (2002) 756–770.
- [16] R. T. Baum, K. B. McGrattan, M. R. Nyden, An examination of the applicability of computed tomography for the measurement of component concentrations in fire-generated plumes, *Combustion and Flame* 113 (3) (1998) 358–372.
- [17] D. Correia, P. Ferrao, A. Caldeira-Pires, Advanced 3D emission tomography flame temperature sensor, *Combustion Science and Technology* 163 (1) (2001) 1–24.

[18] W. Xue, W. Donglou, P. Gongpei, Use of moire tomography to measure the temperature field of the flame of a pyrotechnical composition from its infrared radiation, *Combustion, Explosion and Shock Waves* 37 (4) (2001) 440–442.

[19] B. W. Albers, A. K. Agrawal, Schlieren analysis of an oscillating gas-jet diffusion flame, *Combustion and Flame* 119 (1) (1999) 84–94.

[20] M. M. Hossain, G. Lu, Y. Yan, Tomographic imaging based measurement of three-dimensional geometric parameters of a burner flame, *Proceedings of Instrumentation and Measurement Technology Conference* (2014) 1111–1114.

[21] M. Kang, X. Li, L. Ma, Three-dimensional flame measurements using fiber-based endoscopes, *Proceedings of the Combustion Institute* 35 (3) (2015) 3821–3828.

[22] A. Schwarz, Multi-tomographic flame analysis with a schlieren apparatus, *Measurement Science and Technology* 7 (3) (1996) 406–413.

[23] C. Toro N, L. Arias P, S. Torres, D. Sbarbaro, Flame spectra-temperature estimation based on a color imaging camera and a spectral reconstruction technique, *Applied Optics* 53 (28) (2014) 6351–6361.

[24] Z. Zhou, D. Tian, Z. Wu, Z. Bian, W. Wu, 3D reconstruction of flame temperature distribution using tomographic and two-color pyrometric techniques, *IEEE Transactions on Instrumentation and Measurement* 64 (11) (2015) 3075–3084.

[25] X. Wang, Z. Wang, H. Cheng, Image based temperature field reconstruction for combustion flame, *Optik-International Journal for Light and Electron Optics* 126 (11) (2015) 1072–1080.

[26] Z. Luo, H.-C. Zhou, A combustion-monitoring system with 3-D temperature reconstruction based on flame-image processing technique, *IEEE Transactions on Instrumentation and Measurement* 56 (5) (2007) 1877–1882.

[27] S. W. Hasinoff, K. N. Kutulakos, Photo-consistent 3D fire by flame-sheet decomposition, *Proceedings of Ninth IEEE International Conference on Computer Vision* (2003) 1184–1191.

[28] I. Ihrke, M. Magnor, Adaptive grid optical tomography, *Graphical Models* 68 (5) (2006) 484–495.

[29] R. L. Caceroni, F. L. Pádua, G. A. Santos, K. N. Kutulakos, Linear sequence-to-sequence alignment, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2004) 746–753.

[30] C. Dai, Y. Zheng, X. Li, Subframe video synchronization via 3D phase correlation, *IEEE International Conference on Image Processing* (2006) 501–504.

[31] C. Lei, Y.-H. Yang, Tri-focal tensor-based multiple video synchronization with subframe optimization, *IEEE Transactions on Image Processing* 15 (9) (2006) 2473–2480.

[32] P. Shrestha, H. Weda, M. Barbieri, D. Sekulovski, Synchronization of multiple video recordings based on still camera flashes, *Proceedings of the 14th Annual ACM International Conference on Multimedia* (2006) 137–140.

[33] D. Bradley, B. Atcheson, I. Ihrke, W. Heidrich, Synchronization and rolling shutter compensation for consumer video camera arrays, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2009) 1–8.

[34] N. Teranishi, Y. Ishihara, Smear reduction in the interline ccd image sensor, *IEEE Transactions on Electron Devices* 34 (5) (1987) 1052–1056.

[35] N. Max, Optical models for direct volume rendering, *IEEE Transactions on Visualization and Computer Graphics* 1 (2) (1995) 99–108.

[36] F. J. Serón, D. Gutierrez, G. Gutierrez, E. Cerezo, Implementation of a method of curved ray tracing for inhomogeneous atmospheres, *Computers & Graphics* 29 (1) (2005) 95–108.

[37] Y. Zhao, Y. Han, Z. Fan, F. Qiu, Y.-C. Kuo, A. E. Kaufman, K. Mueller, Visual simulation of heat shimmering and mirage, *IEEE Transactions on Visualization and Computer Graphics* 13 (1) (2007) 179–189.

[38] L. Bavoil, K. Myers, Order independent transparency with dual depth peeling, *NVIDIA OpenGL SDK* (2008) 1–12.

[39] O. Klehm, I. Ihrke, H.-P. Seidel, E. Eisemann, Property and lighting manipulations for static volume stylization using a painting metaphor, *IEEE Transactions on Visualization and Computer Graphics* 20 (7) (2014) 983–995.

[40] V. Pegoraro, S. G. Parker, Physically-based realistic fire rendering, *Proceedings of the Second Eurographics Conference on Natural Phenomena* (2006) 51–59.

[41] D. Q. Nguyen, R. Fedkiw, H. W. Jensen, Physically based modeling and animation of fire, *ACM Transactions on Graphics* 21 (3) (2002) 721–728.

[42] Wiki CIE 1931 color space, https://en.wikipedia.org/wiki/CIE_1931_color_space.

[43] H. Rushmeier, A. Hamins, M. Y. Choi, Volume rendering of pool fire data, *IEEE Computer Graphics and Applications* 15 (4) (1995) 62–67.

[44] I. Ihrke, Reconstruction and rendering of time-varying natural phenomena, PhD thesis.

[45] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, *Proceedings of the Seventh IEEE International Conference on Computer Vision* 1 (1999) 666–673.

[46] A. Laurentini, The visual hull concept for silhouette-based image understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (2) (1994) 150–162.

[47] Y. Hong, D. Zhu, X. Qiu, Z. Wang, Geometry-based control of fire simulation, *The Visual Computer* 26 (9) (2010) 1217–1228.

[48] P. J. Pardo, E. M. Cordero, M. I. Suero, Á. L. Pérez, Influence of the correlated color temperature of a light source on the color discrimination capacity of the observer, *Journal of the Optical Society of America* 29 (2) (2012) A209–A215.

[49] Pokémon Go, <http://www.pokemongo.com/>.



Liang Shen is a PhD candidate at the Institute of Computing Technology, Chinese Academy of Sciences. He did his BSc in Computer Science from China University of Geosciences, Wuhan, China. His main research interests are in fluid reconstruction and computer vision.



Dengming Zhu received his PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2009, where he is also working as an Associate Professor. His main scientific interests are in the areas of fluid simulation, parallel visualization, and virtual reality.



Saad Nadeem is a PhD candidate in the Computer Science department, Stony Brook University. He did his BSc Honors in Computer Science and Mathematics from School of Science and Engineering, Lahore University of Management Sciences, Pakistan. His research interests include computer vision, computer graphics, and visualization.



Zhaoqi Wang is a Researcher and a Director of PhD students at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include virtual reality and intelligent human computer interaction. He is a senior member of the China Computer Federation.



Arie E. Kaufman is a Distinguished Professor and Chairman of the Computer Science Department, Director of the Center of Visual Computing (CVC), and Chief Scientist of the Center of Excellence in Wireless and Information Technology (CEWIT) at Stony Brook University. He has conducted research for over 40 years in visualization and graphics and their applications, has published more than 300 refereed papers, has presented more than 20 invited keynote talks, has been awarded/filed more than 40 patents, and has been PI/co-PI on more than 100 grants. He was the founding Editor-in-Chief of *IEEE Transaction on Visualization and Computer Graphics* (TVCN), 1995–1998. He is a Fellow of IEEE, a Fellow of ACM, the recipient of the IEEE Visualization Career Award (2005), and was inducted into the Long Island Technology Hall of Fame (2013). He received his PhD in Computer Science from Ben-Gurion University, Israel, in 1977.