Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2004)
R. Boulic, D. K. Pai (Editors)

Image-Based Tomographic Reconstruction of Flames

Ivo Ihrke and Marcus Magnor

Max-Planck-Institut für Informatik, Saarbrücken, Germany

Abstract

Non-invasively determining the three-dimensional structure of real flames is a challenging task. We present a tomographic method for reconstructing a volumetric model from multiple images of fire. The method is similar to sparse-view computerized tomography and is applicable to static camera setups observing dynamic flames. Using an algebraic reconstruction method, we can restrict the solution space such that a high quality model is obtained from only a small number of camera images. An additional advantage is fast processing of multi-video sequences to generate time-varying models for animation purposes. The resulting three-dimensional fire model is useful for realistic rendering of fire animations, as well as for analyzing gasdynamics of fires.

1. Introduction

Generating computer animated fire is a difficult and computationally expensive problem. Since fire is a chaotic process it is very difficult to generate convincing animations by means of simulation. A number of methods have been proposed using different kinds of primitives. The main techniques are particle systems [Ree83, TTC97] and simulation of fire propagation [BPP01, NFJ02, ZWF*03]. The most convincing animations have been obtained by physics-based simulation of the combustion process, i.e by solving the Navier-Stokes equations of fluid dynamics [NFJ02]. A good overview of fire modeling techniques can be found in [Has02].

In combustion science, methods for non-intrusive measurements of physical properties of flames have been developed. The main approaches are tomographic methods such as Schlieren tomography and electrical capacitance tomography [Sch96, WHWG99]. These methods, however, require several measurement passes or special equipment and are therefore not suitable to model general flames for computer graphics purposes.

In the presented work, we are not concerned with measuring or simulating physical properties of fire, but with generating visually accurate animations that can be rendered from arbitrary positions. Our work is motivated by the observation that the fire's chaotic behaviour is not well captured by current methods. We therefore use an image based approach to model the three-dimensional emission distribution within real flames. To this end we apply a three-dimensional sparse view tomographic method. Although the method is intended for computer graphics applications it might be possible to use it for analytical purposes as well, e.g. to verify fire simulations and to perform temperature measurements using color pyrometry.

In the following section we give some background information and discuss previous work. Sect. 3 introduces the image formation model and the basic equations and concepts of our method. In Sect. 4 we discuss the details of the reconstruction process followed by Sect. 5 which deals with the rendering of the resulting model. Sect. 6 describes the experiments that were performed to validate our method. Finally, in Sect. 7, we discuss advantages and limitions of our method and give directions for future work.

2. Background and Overview

Image-based modeling of transparent phenomena has not received wide attention in computer graphics. There have been some approaches trying to extend surface reconstruction by taking transparency into account [BV99]. Computerized tomographic methods have been applied to rigid body reconstructions [GW99]. Transparent, volumetric phenomena are treated by Hasinoff et. al. [HK03, Has02]. The authors investigate the ill-posedness of the reconstruction problem. While the similarity of the reconstruction problem to computerized tomography (CT) is pointed out, the CT method is deemed not applicable in the sparse view case. Instead, a small set of irregularly distributed basis functions is fit to the image data. The reconstruction is performed in 2D by using epipolar slices to make the problem tractable. However, the results suffer from overfitting. In Ref. [HK03] the *flame sheet decomposition* algorithm is developed, which reconstructs a surface (the flame sheet) with varying transparency and color. The results are reported to be applicable to noncomplex flames when only a few images are available. Nevertheless the approach treats the volumetric flame as a 2D phenomenon. It therefore doesn't generalize well and fails to capture more complex flame structures.

Our work, in contrast, is a real three-dimensional approach, being essentially a computerized tomography method. We take perspective projection into account. Our approach differs from existing computerized tomography algorithms in that we restrict the solution by using additional information about the visual hull to discard variables (voxel emissivities) that need not be computed. This is not possible with the Radon transform method [Rad17], which is usually used to solve the problem efficiently. Therefore, we choose to employ an algebraic representation of the inversion formula, solving it via an iterative method.

3. Image formation model and basic equations

Hasinoff et. al [HK03] present a simplified image formation model for fire. The fire is modeled as a 3D density field ϕ of fire reaction products i.e. soot particles. Image intensity is related to the density of luminous particles in the fire. The model has the form

$$I_p = \int_c \phi \, ds. \tag{1}$$

Here I_p is pixel p's intensity, c a curve through 3D space and ϕ is the density field of the soot particles. Curve c is the backprojected ray of pixel p in our case. This model assumes infinitely many pixels. We approximate every pixel by one ray through the density field. Additional assumptions of the model are

- Negligible absorption/scattering - this assumption is valid for fire not substantially obscured by smoke, and
- Proportional self-emission - the brightness depends on the density of the soot particles only

In order to invert (1) we have to make an assumption on the structure of ϕ. We do this by assuming that ϕ can be represented as a linear combination of basis functions ϕ_i:

$$I_p = \int_c \left(\sum_i a_i \phi_i \right) ds \tag{2}$$

The sum and the coefficients a_i can be moved out of the integral and we get

$$I_p = \sum_i a_i \left(\int_c \phi_i \, ds \right). \tag{3}$$

Eq. (3) describes a linear system of equations,

$$\mathbf{p} = \mathbf{S}\mathbf{a} \tag{4}$$

The rows represent the equations for one pixel and the columns contain the integrals of the pixel's backprojected rays over the basis function ϕ_i. The choice of the basis functions ϕ_i is essential for the tractability of the problem. Eq. (4) has to be solved for the coefficients $\mathbf{a}_i$ contained in vector $\mathbf{a}$ to reconstruct the density field of the flame. Unfortunately, matrix $\mathbf{S}$ is not well behaved. It has, in general, a large condition number (the quotient between the largest and the smallest singular value), making the inversion of (4) an ill-conditioned problem. This inversion is exactly the computerized tomography (CT) problem. The CT problem is usually solved using the Radon transform [Rad17]. The Radon transform uses the Fourier slice theorem to obtain the reconstruction by applying an inverse Fourier transform. This method has the drawbacks that a camera setup is required where all cameras principal axes meet in one point, and the basis functions must be of the type

$$\phi_i^{Fourier}(x,y,z) = e^{i(\alpha^i x + \beta^i y + \gamma^i z)}. \tag{5}$$

These basis functions have infinite support und thus give rise to a full matrix $\mathbf{S}$ (i.e. every pixel is influenced by every basis function). To be able to solve the linear system in (4) we would like to have a sparse matrix $\mathbf{S}$. This is obtained by choosing basis functions with local support. The simplest of these basis functions is the box function:

$$\phi_i^{Box}(x,y,z) = \begin{cases} 1 & \begin{array}{l} x^i_{min} < x \leq x^i_{max} \\ y^i_{min} < y \leq y^i_{max} \\ z^i_{min} < z \leq z^i_{max} \end{array} \\ 0 & \text{else} \end{cases} \tag{6}$$

Most algebraic reconstruction techniques (ART [KS01] chapter 7) use this basis function and approximate (3) in some way.

The advantage of algebraic reconstruction techniques that use basis functions with local support is the ability to constrain the problem and restrict the solution space by keeping basis function coefficents from being estimated that are known to be zero. We will use this fact to perform a sparse-view tomographic reconstruction of good quality using images of fire.

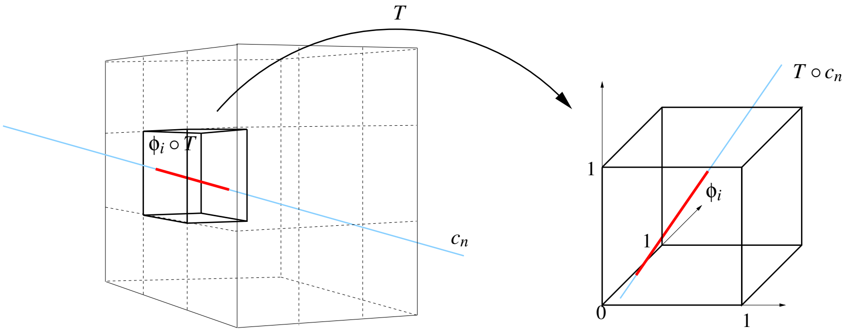


Figure 1: *Relationship between basis function ϕ_i, defined on the unit cube, and curve c defined in world coordinates.*

4. Implementation

The following section presents implementation details, how to efficiently set up and solve the linear system (3). We split the process into two steps and describe them separately. The matrix generation process determines the entries of **S**, regardless of additional knowledge about the solution. This knowledge is used in the process of solving the linear system. Regarding the reconstruction problem as two separate parts allows for the efficient processing of whole sequences of video data.

4.1. Setting up the linear system

As can be seen in (3), the entries of matrix **S** consist of integrals over the basis functions ϕ_i. Since these are chosen to have local support, they are zero over a wide range of the volume. Therefore the integral is zero for a large number of entries of matrix **S**. Determining the entries s_{ni} amounts to intersecting the backprojected rays of all pixels with the support of all basis functions. This is essentially a volume raytracing process. To simplify matters, we choose voxel-aligned basis functions. This choice decreases the amount of computation needed for the intersections from $O(n^3)$ (intersect all basis functions) to $O(3n)$ (intersect $3n$ planes and perform a suitable lookup).

We now consider specific types of basis functions and the resulting structure of matrix **S** of (4). We present a unified approach to the integration problem for different kinds of basis functions. It is not the most efficient implementation for the box basis function but serves as an example for more complicated cases. We define the basis function on the unit cube, transform the curve c of (3) from world coordinates to the unit cube, perform the integration and adjust the result so it is valid in world coordinates (see Fig. 1). We need to compute

$$s_{ni} = \int_{c_n} \phi_i \circ T \; ds \tag{7}$$

where c_n is the ray backprojected from pixel n, $n = 1 \ldots n_{max}$, and n_{max} is the number of pixels that are influenced by any of the basis functions in all camera images. $A \circ B(x) = A(B(x))$ denotes the concatenation of functions. We want to perform the integration in unit coordinates. Therefore, the integral has to be transformed in the following way:

$$\int_{c_n} \phi_i \circ T \; ds = \frac{||r_n||}{||Tr_n||} \int_{T \circ c_n} \phi_i \; dt \tag{8}$$

The factor $\frac{||r_n||}{||Tr_n||}$ relates the integral in unit coordinates to the integral in world coordinates, r_n is the direction vector of the backprojected ray c_n. This factor is only valid for a linear curve c and a linear transform T. The proof is given in the Appendix.

4.1.1. Box basis function

In case of the box basis function ϕ_i^{Box}, the whole computation simplifies considerably. $T \circ \phi_i^{Box}$ is unity in exactly one voxel i and zero elsewhere. x_1 and x_2 are the points of intersection of ray c_n with voxel i. The integral on the unit cube can then be transformed in the following way: Let us consider the curve c_n in world coordinates as

$$c_n(t) = (1-t)x_1 + tx_2 \tag{9}$$

and denote the transformed curve $T \circ c_n$ as

$$F(t) = T \circ c_n(t) = T\big((1-t)x_1 + tx_2\big). \tag{10}$$

Applying (18) to compute the integral in unit coordinates yields

$$s_{ni} = \frac{||x_2 - x_1||}{||T(x_2 - x_1)||} \int_0^1 \phi_i(F(t))\ ||T(x_2 - x_1)||\ dt \quad = (11)$$

$$||x_2 - x_1|| \int_0^1 \phi_i(F(t))\ dt. \qquad (12)$$

Because $F(t)$,$t \in [0,1]$ is completely contained in the support of ϕ_i and $\phi_i = const. = 1$ we arrive at

$$s_{ni} = ||x_2 - x_1||. \qquad (13)$$

This is simply the distance that the backprojected ray travels in the voxel corresponding to basis function ϕ_i.

4.1.2. Trilinear basis function

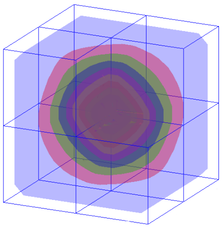


Figure 2: *Visualization of the trilinear basis funtion. This basis function has a support of 8 voxels (1 removed for better visibility). The values of the function are shown as transparent iso-surfaces. At the meeting point of all voxels, the function is one, on the borders it falls off to zero.*

The trilinear basis function has a support of 8 voxels that are arranged in a cube. A visualization is shown in Fig. 2. In the center of this cube the function is one and on its borders it falls off to zero. The values in between are trilinearly interpolated. This results in a cubic polynomial in three dimensions for each voxel. The intersection of the backprojected ray $T \circ c_n$ in unit coordinates and the basis function ϕ_i is a cubic polynomial in every voxel as well, and can thus be integrated analytically. The coefficients of the polynomial can be found by computing an approximation which will be exact because polynomials approximate polynomials of the same degree perfectly. Another option is to compute it using a computer software like Maple. The polynomial that has to be integrated is given in the Appendix.

The trilinear basis functions are arranged on the voxel grid such that there is a one-voxel overlap in every dimension. This ensures smooth blending when rendering the fire. Furthermore, it is well suited for visualization using graphics hardware (see Sect. 5).

4.2. Solution

After having set up the linear system (4) we face a number of difficulties:

- the matrix $\mathbf{S}$ is large
- the linear system is ill-conditioned
- we want to obtain a physically plausible, i.e. non-negative density field ϕ

We wish to compute a least squares solution to (4):

$$\mathbf{a} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{p} \qquad (14)$$

The size of the matrix $\mathbf{S}^T\mathbf{S}$ that is to be inverted is the overall number of basis functions squared. For a reasonably resolved voxel model i.e. more than 64^3 voxels, this is a large system which can only be solved using iterative methods.

4.3. Conjugate gradients for a regularized solution

Fortunately there exist iterative solution methods for linear systems of equations with regularizing properties which is especially useful for our application. The conjugate gradient method (CG, e.g. [Å96, Han98]), developed to solve large symmetric positive definite (SPD) matrices, is suitable for our task. The normal equations (14) are by construction symmetric and positive definite.

Hansen [Han98] discusses the regularizing properties of the CG method in detail. Despite the incomplete theoretical understanding of the convergence properties it is experimentally and partially theoretically shown that the CG method behaves quite similar to the truncated singular value decomposition. The singular values are captured in their natural order starting with the largest [Han98].

This leaves us with finding a non-negative solution. We choose basis functions that are non-negative everywhere. This ensures that non-negative coefficients $\mathbf{a}$ will yield a non-negative density field. Therefore we have to find a non-negative solution vector to (14). We do this by projecting the current solution $\mathbf{a}_k$ to the subspace of non-negative solutions in every iteration k of the CG method. This is done by setting the non-negative entries of $\mathbf{a}_k$ to zero. We apply the CGLS variant [Han98] of conjugate gradient methods to our problem. This variant was developed for solving the normal equations without explicitly computing $\mathbf{S}^T\mathbf{S}$. This saves memory because the explicit representation of the product is usually dense.

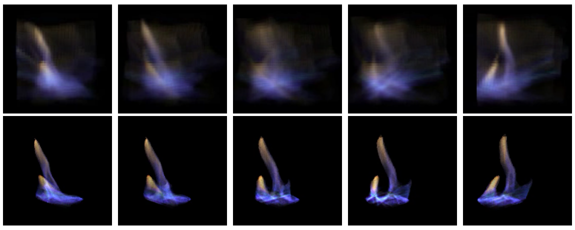


Figure 3: *Synthesized views from the reconstructed volume based on the full equation system (4) (upper row) and the visual hull restricted system (lower row). The left and rightmost images correspond to views near original input views, whereas the middle views are in between views. Ghosting artifacts are clearly visible in the full system case (upper row), the density field suffers from overfitting. These problems are resolved in the visual hull-restricted solution (lower row). The images cover approximately* 90° *and are equally spaced.*

```
Initialization:

    $\mathbf{a}_0 = 0, \quad r_0 = \mathbf{p} - \mathbf{S}\mathbf{a}_0, \quad d_0 = \mathbf{S}^T r_0, \quad k = 0$

Iteration:

    $\alpha_k = ||\mathbf{S}^T r_{k-1}||_2^2 / ||\mathbf{S} d_{k-1}||_2^2$
    $\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha_k d_{k-1}$
    $r_k = r_{k-1} - \alpha_k \mathbf{S} d_{k-1}$
    $\beta_k = ||\mathbf{S}^T r_k||_2^2 / ||\mathbf{S}^T r_{k-1}||_2^2$
    $d_k = \mathbf{S}^T r_k + \beta_k d_{k-1}$

Termination: L-curve criterion
```

As the termination criterion we adopt a variant of the L-curve criterion [Å96, Han98]. The quotient $\frac{||x_k||_2}{||\mathbf{S}x_k - \mathbf{p}||_2}$ of the norm of the solution at step k over the norm of the residual is plotted and analyzed. The point of highest curvature on this curve is the best trade-off between a smooth solution and accuracy in the fit [Han98]. The number of iterations of the CG method plays the role of the regularization parameter in our case.

The results of applying this methodology (using the box basis function) to an input multi-video sequence of 8 camera streams is shown in the upper row of Fig. 3. As can be expected, the number of views is not sufficient to restrict the solution to the real density field and ghosting artifacts are clearly visible.

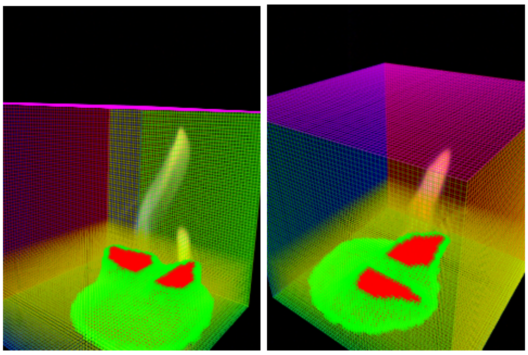


Figure 4: *The image shows the basis functions inside the visual hull in red, partially inside and partially outside green and outside yellow. The box depicts the area of discretization seen from two of the recording camera's viewpoints.*

4.4. Visual hull restricted solution

We circumvent this problem by exploiting additional information. We know that each basis function whose support projects outside the silhouette of the fire in any source image must have a coefficient of zero because the density field vanishes outside the visual hull. The visual hull [Lau94] is a conservative approximation to the geometry of a rigid object that is found by intersecting the backprojected silhouettes from all source images. The actual flames are guaranteed to lie inside the visual hull. We can use this information to determine the basis functions with possibly nonzero coefficients. Fig. 4 shows the area of discretization and the basis functions whose support is inside/outside the visual hull in case of the step basis function, and a discretization of 64^3 basis functions. As can be seen from the figure, most of the 64^3 basis functions are situated in 'empty' space. Only about one tenth of them contribute to the density field that is to be reconstructed.

Since the contribution of each basis function to each pixel in the source images is stored in the columns of the linear system (4) it is simple to adjust it such that the coefficients of those basis functions whose support lies outside the visual hull are not estimated: we simply have to remove the corresponding columns from the linear system before solving the normal equations.

4.5. Animated fire

The restriction of the linear system at solving time is also an efficient way to deal with multi-video sequences, i.e. the reconstruction of a time-varying volumetric model of the flame. Given a static camera setup, the only things that change from frame to frame are the affected basis functions and the right hand side of the linear system (4). Since setting up the linear system is much more expensive, computationally, than solving it, it is advisable to compute the full linear system first and restrict the problem while solving for every frame of the multi-video sequence.

Some results when using this procedure are shown in the lower row of Fig. 3. The images are taken from the same virtual viewpoints as in the row above. By constraining the reconstruction to the volume of the visual hull, photo-realistic rendering results can be obtained.

5. Rendering

Given the coefficients $\mathbf{a}_i$ of the basis functions ϕ_i, rendering corresponds to the direct application of the forward projection formula (3). Therefore, it is sufficient to create the matrix $\mathbf{S}$ for the new view and perform the matrix multiplication (4) to obtain the pixel values. Matrix $\mathbf{S}$ is a precomputed direct volume rendering [Max95]. The similarity between volume rendering and computerized tomography is also pointed out in Ref. [Mal93], where the Fourier Slice Theorem is used to speed up direct volume rendering. An animation from a static viewpoint can be obtained very efficiently since it amounts to just one matrix-vector multiplication (4) per frame.

We reproduce the color information by computing the reconstruction for each color channel separately. The rendering is performed using three different voxel models that are rendered to the three color channels, respectively.

In case of the trilinear basis function it is more convenient (and faster) to perform a volume slice rendering approach using modern graphics hardware [CCF94]. Modern graphics cards perform the trilinear interpolation automatically, so the coefficents $\mathbf{a}_i$ can be used as a volume texture to perform the rendering. Hardware accelerated rendering allow for interactive frame rates.

6. Experimental validation

We recorded a multi video sequence with 8 cameras at 640x480 pixels and 15 frames per second. An approximately circular camera setup was used to acquire the images. The recording was performed in a darkened room with the fire being the only source of light. It was therefore possible to circumvent the step of background subtraction which is nontrivial for transparent phenomena. However, we had to use very high gain settings, introducing noise in the images, notable in the blue channel, Fig. 6(a).

6.1. Experiments

We experimentally analyze the dependency of our method on the discretization resolution. Since we aim at creating photo-realistic images from arbitrary viewpoints using the reconstructed volumetric model, we perform a reconstruction, followed by a rendering of the model. We reconstuct a volumetric model using all views except one, which in turn is used to validate the rendered image of the model. Seven out of the eight views are used to reconstruct a sequence of 100 frames. The reconstructed model is then projected into the eighth view and the difference is computed for each color channel. The average difference in intensity is shown in Fig. 5 for all 100 frames and different levels of discretization. The pixel values range from zero to 255.

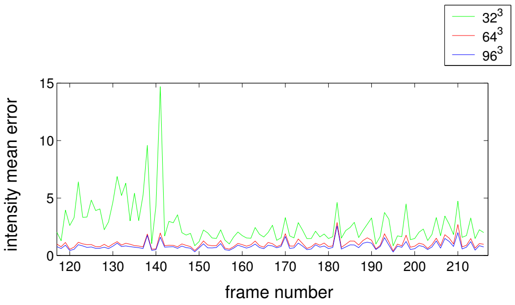
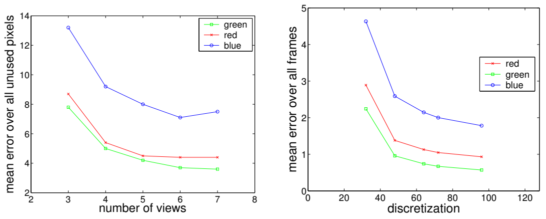


Figure 5: *Plot of the mean reconstruction error for all frames of a sequence. The different curves show results for a discretization of space into 32^3 (green), 64^3 (red) and 96^3(blue) voxels.*

Fig. 5 shows that approximation quality becomes better with higher level of discretization. We also performed experiments with a discretization level of 48^3 and 72^3 voxels. The results of these are not shown for clarity, but are included in Fig. 6(b). These experiments show the tendency to converge towards the correct unused view, suggesting that the 3D structure of the density field is indeed captured accurately.



(a) error vs. number of views

(b) error vs. discretization

Figure 6: *Mean error plotted against the number of views for reconstruction (left) and the mean error over the whole sequence of Fig. 5 against the level of discretization. The three curves denote the three color channels red, green and blue.*

Another experiment was performed to evaluate the dependency on the number of views that are used for the reconstruction. We reconstruct one frame of the sequence using

3, 4, 5, 6 and 7 views of the flame. The comparison is performed against the left-out views. The results are shown in Fig. 6(a). Here as well, the convergence of the solution can clearly be observed. Visually acceptable reconstruction results, especially when used in animated renderings, are obtainable with as few as 4 to 5 views. 8 views are sufficient for photo-realistic rendering.

6.2. Discussion

Potential sources of error of our method are

- camera calibration errors,
- color calibration errors,
- 2D image processing (rescaling), and
- discretization (number of views and spatial discretization).

Calibration errors are inevitable and tend to create too small visual hulls. This means that the actual silhouette is a bit larger than the silhouette of the reconstructed model. Since this does not happen with synthetic test images, such errors can be attributed to camera calibration inaccuracies. We exclude these pixels from the error measurement to prevent biasing the error measure.

We recorded the sequences with the same settings for all cameras. Since this produces visually quite similar images, we didn't concentrate on color calibration.

Rescaling the images is necessary to fit the matrix in memory (2 GB RAM). It should be noted that 2D image processing might influence the result of the reconstruction because it can introduce effects not described by our model. Projecting the result of the reconstruction back to the original views and referring to (14), the reconstruction/rendering loop $\hat{\mathbf{p}} = \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{p}$ can be interpreted as a filter on the 2D pixel data $\mathbf{p}$ where $\hat{\mathbf{p}}$ is the solution projected to the original views. If the pixel data $\mathbf{p}$ is filtered in 2D prior to the reconstruction procedure, it might not correspond to a filter in the 3D domain and therefore introduce artifacts. Finally, the discretization of the density field and the assumption on its special structure, i.e. its composition from basis functions, introduce errors of their own.

While one has to be aware of these error sources, our validation experiments demonstrate that it is possible to compute density fields that are able to generate photo-realistic images from arbitrary viewpoints.

7. Conclusions and future work

We have presented a method that is capable of reconstructing dynamic, volumetric models of fire for animation purposes. Our approach is applicable in case of

- negligible scattering (fire not obscured by smoke)
- no sensor saturation in the input images
- no opaque objects inside the flame
- no part of the flame is seen by less than 2 cameras

We obtain photo-realistic results. Validation shows that our approach reconstructs the actual 3D distribution of flame intensity. Our results are obtained using 8 cameras, demonstrating that the method is applicable in a sparse view scenario. The reconstruction is optimized for processing of multi-video-sequences, making it a suitable tool to model fire for animation purposes.

Future work includes adding an exponential decay term for modeling absorption, incorporating occluding bodies inside the flame, exploiting temporal coherence, background subtraction, color calibration and higher order basis functions, e.g. for hierarchical representations.

Appendix

Transformation of curve integrals

Here we give the proof for Eq. (8).

$$\begin{aligned}
& \int_{c_n} \phi_i \circ T \; dt && (15)\\
= \quad & \int_0^1 \phi_i \circ T \circ c_n(t) \; \left|\left|\tfrac{dc_n(t)}{dt}\right|\right| \; dt && (16)\\
\overset{(*)}{=} \quad & \tfrac{||r_n||}{||Tr_n||} \int_0^1 \phi_i \circ T \circ c_n(t) ||Tr_n|| \; dt && (17)\\
\overset{(**)}{=} \quad & \tfrac{||r_n||}{||Tr_n||} \int_0^1 \phi_i \circ T \circ c_n(t) \left|\left|\tfrac{d(T\circ c_n)}{dt}\right|\right| \; dt && (18)\\
= \quad & \tfrac{||r_n||}{||Tr_n||} \int_{T\circ c_n} \phi_i \; dt && (19)\\
(*) \quad & \tfrac{dc_n(t)}{dt} \overset{c_n(t) linear}{=} r_n, const. && (20)\\
(**) \quad & \tfrac{d(T\circ c_n)(t)}{dt} = \tfrac{dT}{dc_n}\tfrac{dc_n}{dt} \overset{\overset{T linear}{c_n linear}}{=} Tr_n && (21)
\end{aligned}$$

Note that this proof is only valid for a linear curve c_n and linear transformation T.

Polynomial to be integrated for the trilinear basis function

In the following we assume the curve c_n to be in the form $T \circ c_n(s) = p + sr$. The direction vector r is normalized. The cubic polynomial that is to be integrated for the trilinear basis function is given by:

$$\begin{aligned}
s^3* \quad & (a_1 r_x r_y r_z)+ \\
s^2* \quad & (a_1(r_x r_y p_z + p_x r_y r_z + p_y r_x r_z)+ \\
& a_2 r_x r_y + a_3 r_x r_z + a_4 r_y r_z)+ \\
s\ * \quad & (a_1(p_z p_x r_y + p_z p_y r_x + p_x p_y r_z)+ \\
& a_2(p_x r_z + p_y r_x) + a_3(p_x r_z + p_z r_x)+ \\
& a_4(p_y r_z + p_z r_y) + a_5 r_x + a_6 r_y + a_7 r_z)+ \\
1\ * \quad & (a_1 p_x p_y p_z + a_2 p_x p_y + a_3 p_x p_y + a_4 p_y p_z+ \\
& a_5 p_x + a_6 p_y + a_7 p_z + a_8)
\end{aligned} \qquad (22)$$

The constants $a_1 \dots a_8$ are defined as

$$
\begin{aligned}
a_1 &= -p_{000} - p_{001} + p_{010} + p_{011} \\
&\quad - p_{100} - p_{101} + p_{110} + p_{111} \\
a_2 &= -p_{000} + p_{001} + p_{010} - p_{011} \\
a_3 &= -p_{000} + p_{001} - p_{100} + p_{101} \\
a_4 &= -p_{000} + p_{010} - p_{100} + p_{110} \\
a_5 &= -p_{000} - p_{001} \\
a_6 &= p_{000} - p_{010} \\
a_7 &= p_{000} + p_{100} \\
a_8 &= -p_{000}
\end{aligned}
\qquad (23)
$$

$p_{000} \dots p_{111}$ are the values at the corner points of the unit cube in the order of p_{zyx}, respectively. The integration has to be performed from s_1 to s_2 which are the intersections of c_n with the voxel in question in unit coordinates.

References

[Å96] Åke Björck: *Numerical Methods for Least Squares Problems*. Society of Industrial and Applied Mathematics, 1996. 4

[BPP01] Beaudoin P., Paquet S., Poulin P.: Realistic and Controllable Fire Simulation. In *Graphics Interface 2001* (June 2001), pp. 159–166. 1

[BV99] Bonet J. S. D., Viola P. A.: Roxels: Responsibility Weighted 3D Volume Reconstruction. In *Proc. International Conference on Computer Vision (ICCV '99)* (1999), pp. 418 – 425. 1

[CCF94] Cabral B., Cam N., Foran J.: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *Proc. Symposium on Volume Visualization (VolVis'94)* (1994), 91–98. 6

[GW99] Gering D. T., Wells III W. M.: Object Modeling using Tomography and Photography. In *Proc. of IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes* (June 1999), pp. 11–18. 1

[Han98] Hansen P. C.: *Rank-Deficient and Discrete Ill-Posed Problems*. Society of Industrial and Applied Mathematics, 1998. 4, 5

[Has02] Hasinoff S. W.: Three-Dimensional Reconstruction of Fire from Images. MSc Thesis, University of Toronto, Department of Computer Science, 2002. 1

[HK03] Hasinoff S. W., Kutulakos K. N.: Photo-Consistent 3D Fire by Flame-Sheet Decomposition. In *In Proc. 9th IEEE International Conference on Computer Vision (ICCV '03)* (2003), pp. 1184 – 1191. 1, 2

[KS01] Kak A. C., Slanley M.: *Principles of Computerized Tomographic Imaging*. Society of Industrial and Applied Mathematics, 2001. 2

[Lau94] Laurentini A.: The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Recognition 16*, 2 (Feb. 1994), 150–162. 5

[Mal93] Malzbender T.: Fourier Volume Rendering. *ACM Transaction on Graphics 12*, 3 (July 1993), 233–250. 5

[Max95] Max N.: Optical Models for Direct VOLUME Rendering. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (June 1995), 99–108. 5

[NFJ02] Nguyen D. Q., Fedkiw R., Jensen H. W.: Physically Based Modelling and Animation of Fire. *ACM Transactions on Graphics 21*, 3 (July 2002), 721–728. 1

[Rad17] Radon J.: Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. Ber. Ver. Sachs. Akad. Wiss. Leipzig, MathPhys. Kl., 69:262 277, April 1917. 2

[Ree83] Reeves W. T.: PARTICLE Systems - A Technique for Modeling a Class of Fuzzy Objects. In *ACM Transactions on Graphics* (April 1983), vol. 2, pp. 91–108. 1

[Sch96] Schwarz A.: Multi-tomographic flame analysis with a schlieren apparatus. *Meas. Sci. Technol. 7* (1996), 406–413. 1

[TTC97] Takahashi J., Takahashi H., Chiba N.: Image Synthesis of Flickering Scenes Including Simulated Flames. *IEICE Transactions on Information Systems E80-D*, 11 (November 1997), 1102–1108. 1

[WHWG99] Waterfall R. C., He R., Wolanski P., Gut Z.: Monitoring Flame Position and Stability in Combustion Cans using ECT. In *Proc. 1st World Congress on Industrial Process Tomography* (April 1999), pp. 35–38. 1

[ZWF*03] Zhao Y., Wei X., Fan Z., Kaufman A., Qin H.: Voxels on Fire. In *Proc. 14th IEEE Visualization Conference (VIS'03)* (October 2003), pp. 271–278. 1

:

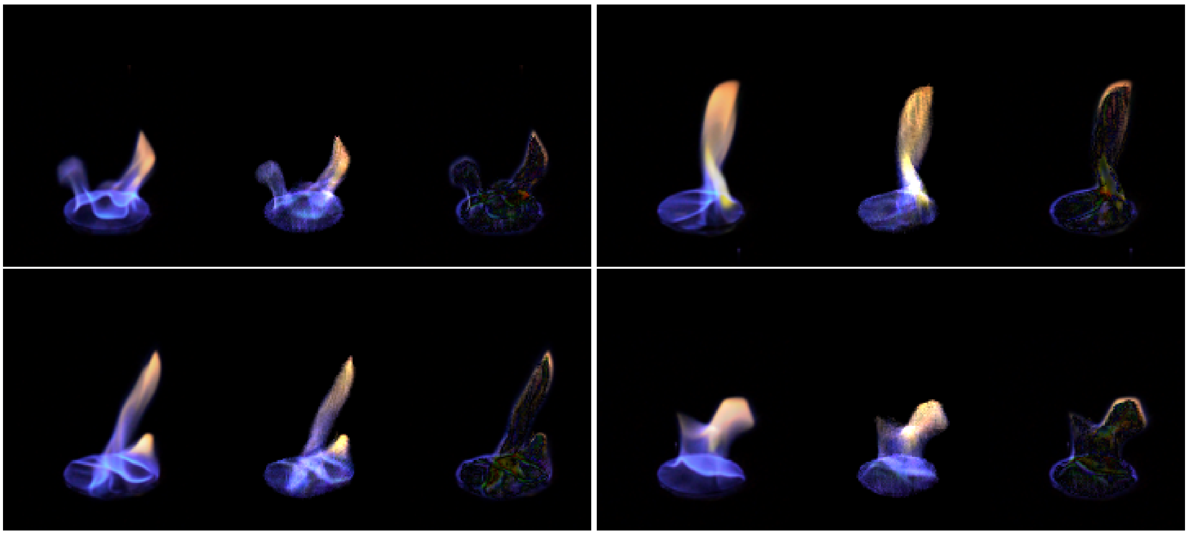


Figure 7: *Difference between original, unused views and reconstruction rendered from the same viewpoint. Left: Original, Middle: Reconstruction, Right: Difference.*

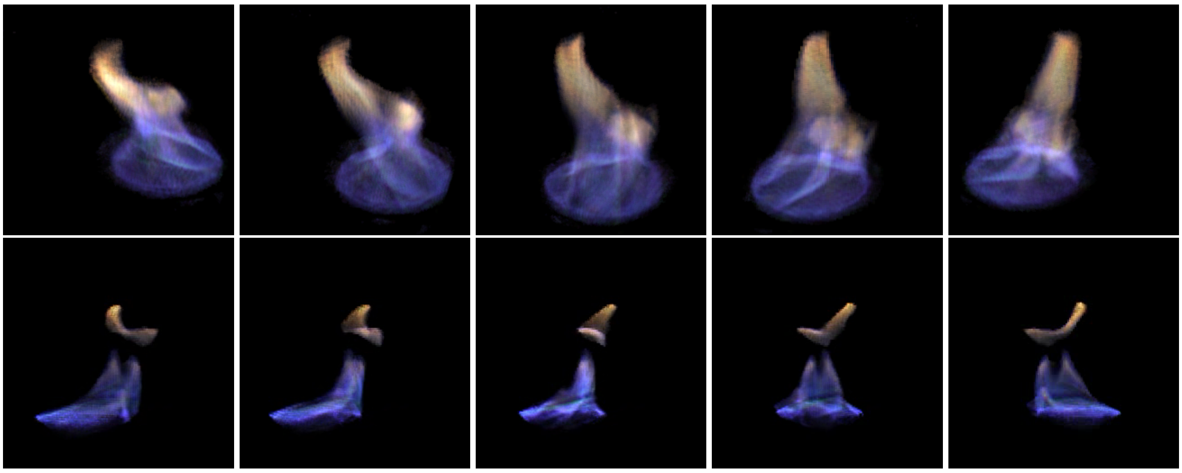


Figure 8: *Synthesized views of two different flames with black background, the 5 images of each row cover approx.* 120°.

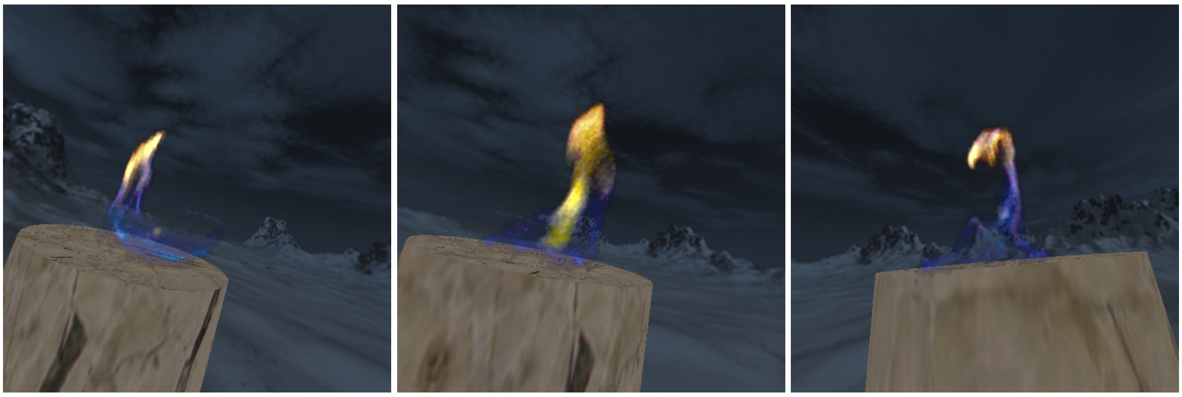


Figure 9: *Synthesized views of animated fire in a virtual environment.*