Learning Flames

Timo Stich, Marcus Magnor

MPI Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: {tstich, magnor}@mpi-inf.mpg.de

Abstract

In this work we propose a novel approach for realistic fire animation and manipulation. We apply a statistical learning method to an image sequence of a real-world flame to jointly capture flame motion and appearance characteristics. A low-dimensional generic flame model is then robustly matched to the video images. The model parameter values are used as input to drive an Expectation-Maximization algorithm to learn an *auto regressive process* with respect to flame dynamics. The generic flame model and the trained motion model enable us to synthesize new, unique flame sequences of arbitrary length in real-time.

1 Introduction

The motion of flames are governed by hydrodynamical forces. The interaction between laminar and turbulent flow within flames can cause not quite periodic, yet at the same time also not-yet truly chaotic flame motion. These are the basic properties of auto regressive processes(ARP). In this paper we will use learning methods to train such an ARP that captures the characteristic flame dynamics. We are further proposing a generic flame model which we robustly fit to recorded video sequences of flames.

The result will be a method to synthesize arbitrary long and unique flame sequences from our trained model in real-time.

2 Related Work

Recent research into realistic flame modeling and animation in computer graphics has been concentrating on physical models and on solving the Navier-Stokes equations [4, 5, 6]. By careful model parameter adjustment, the obtained results can be very convincing [3]. However, fine-tuning the model parameters towards some desired flame output is tedious as the parameters' physical interpretation does not directly relate to their impact on flame animation and appearance. Another drawback of physically based approaches is the computational power needed to solve the underlying differential equations which necessitates trading off animation realism for real-time performance on standard hardware.

To overcome having to solve the partial differential equations of the underlying physics, Schoedl et al. [7] propose an image-based approach to flame animation. Their idea is to segment a real-world input video sequence into smaller time periods that can be re-arranged to create new and possibly longer sequences. However, the inevitable recurrence of identical sub-sequences limits the sense of realism of any newly generated longer sequence. Also, for more turbulent flames, dividing the video image sequence into concatenatable sub-sequences appears difficult.

Our approach aims at overcoming both problems by learning a model of the characteristic motion and appearance of a flame. To the video images of a real-world flame, we adapt a generic flame model. From the evolution of model parameter values, an auto regressive process (ARP) is trained. The learned ARP is able to drive the synthetic flame animation in real-time, creating arbitrary long and unique sequences.

The rest of the paper is structured as follows. In Section 3 we describe our generic flame model that can be matched automatically to the flame in the video images. Section 4 introduces the learning framework that we employ to derive a statistical model of flame motion and appearance. We present real-world results from our experiments in Section 5 before we conclude in Section 6.

3 Flame Model

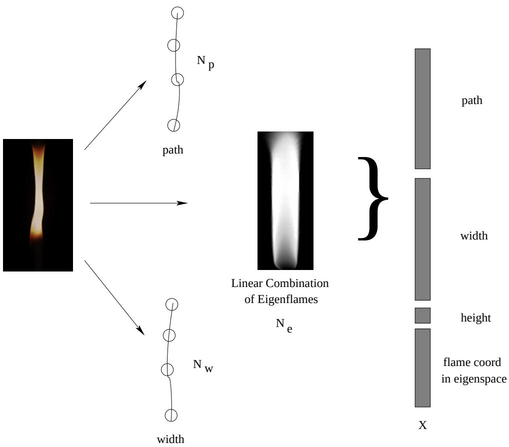

Figure 1: Overview of the generic flame model. One flame is represented by the state vector x.

As input to our method, we record a real-world flame over a period of time with a conventional video camera, Fig. 5. Our sequences are recorded at 1000×1000-pixel resolution, 12-bit linear dynamic range per pixel, and 40 frames per second. To be able to make use of statistical learning methods to capture the characteristics of flame motion and intensity variations, the flame images must be transformed to a low-dimensional, yet visually faithful representation. Reducing the dimensionality is a common pre-processing step in computer vision and machine learning to concentrate on the relevant features of the problem [2].An overview of the generic flame model is depicted in Figure 1. We propose a generic flame model that can be matched automatically to the video images and from which realistic flames can be rendered, Fig. 3.

Our flame model is based on prior knowledge of general flame composition and consists of two parts. The first part captures the deformation of the flame by a set of shape parameter values. The second part represents the visual appearance of the flame as a linear combination of *eigenflames* which are obtained from the input images, Section 3.2. The model parameter values are derived without manual intervention directly from the input video images.

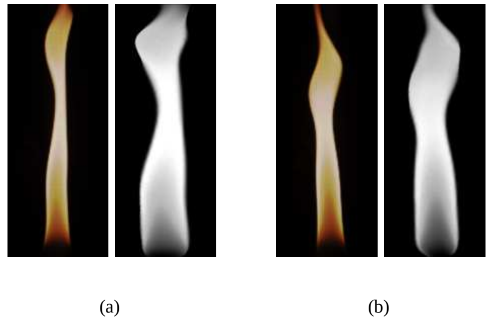

Figure 2: Comparison between input images (a, b left) and rendered model images (a,b right). The flames are faithfully reproduced by the 26-parameter model.

3.1 Deformation Parameters

As shape parameters, we rely on the geometric image moments M_{pq} as defined by

$$M_{pq} = \int\int x^p y^q I(x,y)dxdy, \qquad (1)$$

where I is the gray valued image of the flame and x and y are the coordinates. The order of the moment is defined as $p+q$. An introduction to geometric moments can be found here [8]. The parameters of the deformation in our approach are estimated from the normalization transformation.

We use moments up to the order 3 to normalize the flame images. After the normalization process the following properties hold

- all normalized images exhibit the average intensity.
- the main axes of the gray-value distribution are aligned with the image axes.

Making the (valid) assumption that the main axis of the flame distribution truthfully represents the overall orientation of the flame, we are able to estimate the height of the flame from the second-order moment. The total height of the flame is subdivided into several slices. Each height slice is normalized using the moments of this slice. This way, the deformation parameters path eccentricity and width of flame is determined for each slice The estimated height-slice parameters are used to center and normalize each slice of the flame image. The final step in parameter estimation consists of sampling

N_p path and N_w width parameters at equidistant heights. As the result, flame shape is described by $N_p + N_w + 1$ (flame height) parameters.

During rendering, the normalization procedure is reversed. To get a smooth deformation of the synthesized normalized flames the sparsly sampled parameters are interpolated with a cubic spline. The resulting deformation process is then the inverse normalization transformation defined by the current model parameters.

3.2 Eigenflames

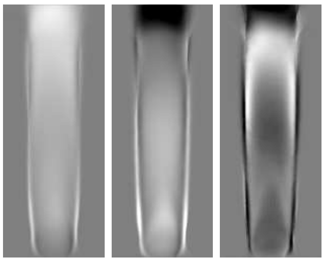

Figure 3: The first three eigenflames of the recorded sequence.

After the flame shape has been normalized as described, we can now analyze flame appearance over the entire video sequence. One crucial observation is that after normalization, the resulting flame images are very similar. This is a necessary pre-condition for a successful eigen-image analysis [10]. As the result of our principal-component analysis, we obtain a set of "eigenflames" that form the basis of our "flame-space", Fig. 3. Reducing the dimensionality of our flame representation in a guaranteed least-squares error sense consists of simply selecting those N_e eigenflames with the largest eigenvalues as the basis for the sub flame-space of our generic flame model.

During rendering, we obtain each new flame by a linear combination of the selected eigenflames weighted with the sampled parameters from the current model parameters.

4 Learning the ARP

Having matched our generic flame model to the recorded flame images, we are now in a position to model how shape and appearance parameters evolve over time. In this paper we model the underlying temporal process of the observed parameters as an auto regressive process.

In the following, we refer to the model parameters derived from the input video sequence as $\mathbf{x_t}$. Thus, $\mathbf{x_t}$ is a vector of length $N_p + N_w + N_e + 1$. Note that at an time instant t, the flame is completely represented by $\mathbf{x_t}$.

We assume the temporal Markov property which renders the model independent of any previous state x_{t-k} for $k > K$. This leads to the definition of the ARP like follows:

$$x_t = \sum_{k=1}^{K} A_k x_{t-k} + d + B w_t \qquad (2)$$

x_t is the vector describing the state of the model at time t. A_k and d are deterministic parameters of the process and matrix B models the weighting of the stochastic vector w where each component w_i is an independent random $N(0,1)$ variable. An ARP is now sufficiently described by the set of parameters $\lambda = \{A, d, B\}$ with $A = (A_1 A_2 ... A_K)$. The learning process is then to estimate a parameter set λ^* that faithfully represents the flame dynamics.

Further, we assume that the state of the model x_t is not observed directly but is the result of a noisy observation. This assumption reduces the influence of errors in the estimation of the parameters in the previous step.

In addition, we assume that the state of the model x_t is not observed directly but is the result of a noisy observation. This assumption reduces the influence of errors in the estimation of the parameters in the previous step. Since the state of the model is hidden, using a Maximum-Likelihood approach to estimate the parameters is not directly possible [11]. Instead, we use an iterative Expectation-Maximization algorithm for learning the stochastic dynamic process [9].

4.1 EM with Condensation

To learn the ARP, we rely on the condensation algorithm [12]. The condensation algorithm is, in essence, a particle filter approach which approximates a multivariate distribution. In the context of learning an ARP, the condensation algorithm was first applied in the Expectation step of the EM algorithm [13, 1]. We give a short outline of the learning process here.

The EM algorithm is divided into two steps. The first step consists of estimating the expected values of moments and auto correlations given an ARP λ_{i-1}. In the second step, the expected values are used to compute a Maximum-Likelihood Estimation of a new ARP λ_i. Both steps are repeated until a terminating criterion is met.

As mentioned above, the condensation algorithm is used to estimate the expected values in the Expectation step. The number of particles used in the computation is set to N_c. Because the approximation quality of the true multivariate distribution increases with N_c, one has to find a reasonable trade-off between computational complexity and goodness-of-fit. In our experiments we found that a value of N_c in the range of 50 to 300 yields good results, Section 5.

4.2 Initialisation

Since the EM algorithm converges to a local optimum, the learning results depend on a good initial ARP λ_0. We therefore relax the assumption of a noisy observation process and restrict the initial influence of the parameters to the time axis, which means the A_k^0 are diagonal matrices. Further with the assumption that $d^0 = 0$, we can estimate A_k^0 directly from the input data. B^0 is initialized as a diagonal matrix with the standard deviations of the observed parameters on the diagonal. Because very large or small ranges of parameter values can cause numerical instabilities during the learning, we scale the input data to a standard deviation of one. After the learning process, the parameters λ are rescaled to reproduce values in the original range of the variables.

4.3 Synthesizing new flames

Having learned the ARP that represents the characteristic dynamics of our flame, we can now use it to generate new parameter values for our generic flame model. This is done by repeatedly evaluating (2) to create new $\mathbf{x_t}$. The obtained parameter values are then used to synthesize new flame renditions from the flame model introduced in Section 3. Evaluating (2) can be repeated over arbitrary lengths of time to create new, unique flame sequences. Figure 4 gives an overview of the process.

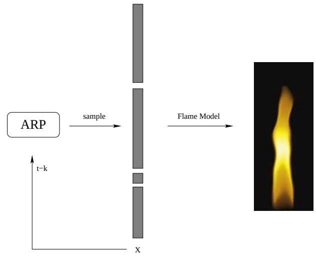

Figure 4: To generate a sequence of new flames, simple sampling of the ARP is done. The obtained x are used as input to our flame model. This yields images for sampled flame parameters.

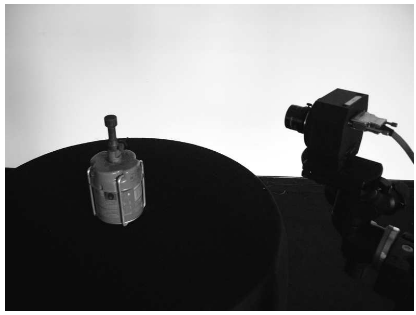

Figure 5: Acquisition setup

5 Results

The results presented here are based on a sequence of flame images recorded with a 1 Megapixel camera at 40 fps, Fig. 5. Recording the video sequence in the dark avoids background subtraction of the transparent flame images. From the recorded video sequence, images of flames consisting of two or more disjoint plumes are discarded, because our generic flame model, Section 3, cannot represent such (rather rare) flame appearances. From the recorded video sequence, a sub-sequence of 100 flame images is later selected that shows a single flame without topology changes, i.e., without plumes. Fig. 6 depicts three images from the input sequence. The flame model consists of $N_p = N_w = 10$ and $N_e = 5$ parameters, Section 3, yielding in a total 26 parameters. As starting ARP, the heuristic proposed in section 4.1 and $K = 2$ is used. $N_c = 200$ particles for the condensation algorithm are used. The iterative optimization routine is ter-

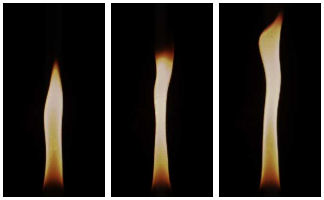

Figure 6: Images from the input sequence.

minated if the likelihood does not increase during three consecutive iterations. The current Matlab implementation of the algorithm takes about 30 minutes on a 3 GHz Pentium4.

5.1 Synthesized flame sequences

Arbitrary new, unique flame sequences can be synthesized from the learned model, Section 4.3. Movies of the resulting synthetic flames can be found on the project homepage[1]. Fig. 7 depicts a sample of synthetic flames. Note, however, that the true quality of the proposed algorithm can be assessed only from the animated flame. While a synthetic flame with the same flickering and movement characteristics as the original flame can be obtained directly without any additional scaling or transformation of the parameters, our model additionally gives control over flame appearance. For example, the general path of the flame can be adjusted over time by adding an offset to the path parameters, resulting in a bent flame. This can be useful to introduce the effects of external forces such as wind. The same direct manipulations can be applied to the flame height, width, and scale.

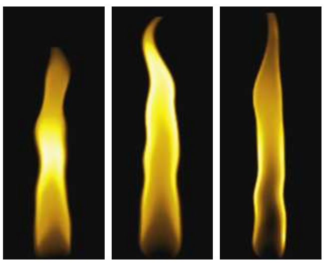

Figure 7: Novel flames created from the learned model.

[1] http://mpi-inf.mpg.de/~tstich/learning_flames/

The current model relies on gray-valued flame images indicating emissivity. Due to the nature of flames we can introduce a transfer function to map flame intensity to color. It is also easily possible to extend the model to include the color of the input images.

Objectively validating the accuracy of our results is not easily possible. The degree of obtained realism might be best evaluated by simply looking at the synthesized images and movies.

6 Discussion and Future Work

In this paper we have introduced a novel approach for the animation and modeling of flames. Unlike previous physically-based or image-based approaches, our method relies on a statistical learning approach. We propose a general flame model which can be robustly matched to recorded video sequences. By learning the temporal characteristics of the flame shape and appearance, we are able to synthesize arbitrarily long, unique sequences in real-time.

6.1 Future Work

Our compact representation of flame characteristics can be extended to offer additional degrees of freedom in manipulating flame animations For example, by learning different flames, continuous interpolation between laminar-flow flames and turbulent flames becomes possible, yielding intuitive controls over flame animation. Since the applied learning approach is capable of learning multi-mode dynamic systems, it can be used directly to model the different dynamics, e.g., also depending on the influence of external forces on the flame.

In our current implementation we assume a single flame without topology changes, i.e., plumes. Using a separate ARP, the dynamics of plumes can also be modeled. We are also planning to expand the generic flame model to directly support color flames by preserving the color of the eigenflames and blending them into the synthesized image. Finally, we can apply our approach not only to 2D video images but also directly to time-varying 3D volume models [14]. This will allow integrating learned flames into 3D virtual environments.

References

[1] Ben North, Andrew Blake, "Learning and Classification of Complex Dynamics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1016-1034, 2000.

[2] Christoph Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 1997.

[3] Arnauld Lamorlette, Nick Foster, "Structural Modeling of Flames for a Production Environment", *ACM Transitions on Graphics*, pp. 729-735, 2002.

[4] Philippe Beaudoin et al., "Realistic and Controllable Fire Simulation", *Proceedings of Graphics Interface*, pp. 159-166, 2001.

[5] Jos Stam, Eugene Fiume, "Depiction of Fire and Other Gaseous Phenomena Using Diffusion Processes", *Proceedings of SIGGRAPH 1995*, pp. 129-136, 1995.

[6] Duc Quang Nguyen et al., "Physically Based Modeling and Animation of Fire", *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 721 - 728, 2002.

[7] Arno Schödl et al., "Video Textures", *Proceedings of SIGGRAPH 2000*, pp. 489-498, 2000.

[8] R Mukundan, K R Ramakrishnan, *Moment Functions in Image Analysis*, World Scientific, 1998.

[9] Arthur Dempster et al., "Maximum likelihood from incomplete data via the em algorithm", *Journal of the Royal Statistical Society*, vol. 39, pp, 1-38, 1977.

[10] Mathew Turk, Alex Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991.

[11] Arthur Gelb ed., *Applied optimal Estimation*, MIT Press, 1974.

[12] Michael Isard, Andrew Blake, "Contour tracking by stochastic propagation of conditional density", *Proc. European Conf. on Computer Vision*, vol. 1, pp. 343-356, 1996.

[13] Andrew Blake et al., "Learning Multi-Class Dynamics", *Advances in Neural Information Processing Systems*, M.S. Cearns, S. Solla, and D. Cohn, eds., vol. 11, 1999.

[14] Ivo Ihrke, Marcus Magnor, "Image-based tomographic reconstruction of flames", *Proceedings of the Eurographics symposium on Computer animation*, pp. 365-373, 2004.