



HAL
open science

Influence of Fog on Computer Vision Algorithms

Sule Kahraman, Raoul de Charette

► **To cite this version:**

Sule Kahraman, Raoul de Charette. Influence of Fog on Computer Vision Algorithms. [Research Report] Inria Paris. 2017, pp.1-3. hal-01620602

HAL Id: hal-01620602

<https://inria.hal.science/hal-01620602>

Submitted on 20 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Influence of Fog on Computer Vision Algorithms

Sule Kahraman, Raoul de Charette*

Abstract—This technical report describes a new preliminary approach to simulate fog in images using accurate physical and photometric models to study the influence of small particles on computer vision algorithms.

Index Terms—Fog, Bad weather, Vision in bad weather, Computer Vision, Computer Graphics

I. INTRODUCTION

The Earth's atmosphere is significantly non transparent in bad weather conditions. The small particles in the atmosphere create a scattering medium like fog that attenuates the light passing through and result in failure of computer vision algorithms such as feature detection, object recognition, tracking.

Particles with size between $1\mu\text{m}$ and $10\mu\text{m}$ are considered to be small particles, too small to be individually detected by a camera. Hence, the great amount of droplets within the pixel's solid angle change the intensity produced at a pixel. Volumetric scattering models such as attenuation and airlight [1], [2] are used to describe the effect of these small particles on the image.

In the last two decades, various methods [3] have been developed for photo-realistic rendering of fog in computer graphics using scattering models and several studies [4], [5] have been conducted to investigate the effect of bad weather on computer vision. Realistic fog simulation could allow evaluating the effects of foggy weather on computer vision algorithms under otherwise identical conditions (i.e. environment and camera setup).

In this document, we propose a method for the simulation of fog using realistic physical and photometric model for small particles and we investigate the influence of fog on computer vision algorithms. In the following we provide the details of our method for physical simulation of fog. We use the atmospheric scattering model to simulate fog and later show the advantage of heterogeneity for more realism. Finally, common features detectors are ran on images with simulated fog to evaluate the effects of fog.

II. SIMULATION OF FOG

In this section, we explain our method for rendering foggy images. As explained in [4], the small particles in the atmosphere creates a scattering medium for light. Light passing through a scattering medium is attenuated and distributed to other directions.

Authors are with the Robotics and Intelligent Transportation Systems (RITS) Team, INRIA Paris, 2 rue Simone Iff, 75012 Paris, France {raoul.de-charette, sule.kahraman}@inria.fr

A. Atmospheric Scattering Model

Atmospheric scattering model describes how suspended small particles alters the intensity and color of light due to attenuation and airlight.

First, we explain the effect of attenuation that fog creates. Due to attenuation, the radiance of a scene point decreases as its depth from the observer increases. Allards law [6] which is derived from the Bouguer's exponential law of attenuation describes this relation as follows:

$$E(d, \lambda) = \frac{I_o(\lambda)e^{-\beta(\lambda)d}}{d^2} \quad (1)$$

where d is the distance from the light source λ is the wavelength of light, I_o is the radiant intensity of the point source, and $\beta(\lambda)$ is the total scattering coefficient of the medium.

Second effect of fog in the atmosphere is airlight. The latter causes the atmosphere to behave like a source of light due to the scattering of environmental illumination (e.i. sunlight, skylight, etc.) by particles in the atmosphere. Apparent brightness of a scene point increases with depth which means that airlight increases with path length, whereas attenuation causes scene radiance to decrease with path length. Total radiance at distance d from the camera, $L(d, \lambda)$ is

$$L(d, \lambda) = L_\infty(1 - e^{-\beta(\lambda)d}) \quad (2)$$

where L_∞ is the horizon radiance where the radiance of airlight is maximum when object at infinity.

The total scattering coefficient is written as a function of λ in these equations because β depends on the ratio of droplet radius to wavelength and refractive index of water (which depends on temperature). However, for fog and dense haze, the influence of wavelength can be neglected [4]. This is because $\gamma \approx 0$ in the Rayleighs law [1], [2], which describes the relationship between the scattering coefficient β , and the wavelength λ :

$$\beta(\lambda) = \frac{\text{Constant}}{\lambda^\gamma} \quad (3)$$

Hence, we assume the total scattering coefficient $\beta(\lambda)$ to be constant over because the spectral bandwidth of the camera is limited (visible light range for a gray-scale camera, and even narrower spectral bands when the camera is color) [4].

In order to physically simulate fog on images, we use the atmospheric scattering model which is the sum of the attenuation (eq. 1) and airlight models (eq. 2). Hence, the intensity of light received by the observer $I(d, \lambda)$ is the sum of directly transmitted light $D(d, \lambda)$ and airlight $A(d, \lambda)$ as described in [4]. Because the total scattering coefficient is

independent of wavelength in the case of fog, we can describe the atmospheric scattering model as:

$$\begin{aligned} I(d) &= D(d) + A(d) \\ D(d) &= I_o e^{-\beta d} \\ A(d) &= L_\infty (1 - e^{-\beta d}) \end{aligned} \quad (4)$$

When rendering fog on images, $I(d)$ is the foggy image intensity as an RGB vector and $D(d)$ is the product of transmission map $t(d)$ and the clear scene image I_o . Hence, $I(d)$ is explicitly:

$$I(d) = I_o e^{-\beta d} + L_\infty (1 - e^{-\beta d}) \quad (5)$$

Note that the spatially aligned depth map d is required. In virtual environment this is directly obtained from the rendering engine, whereas for real images we used MonoDepth [7] to predict depth from single RGB images.

Fig. 1 shows the original image along with a sample simulation of thick fog ($\beta = 40$). Additional visual results are shown in fig. 3. As the total scattering coefficient β increases, fog’s density increases and the visibility decreases.

Fig. 1. Original image and sample homogeneous thick fog



B. Heterogeneous Fog

Using atmospheric scattering model in our renderings results smooth, evenly distributed homogeneous fog simulation, yet fog in the atmosphere is not homogeneous. Hence, images with homogeneous fog are not realistic. Common practice in the literature is to add 2D Perlin noise [8] to create natural looking turbulence texture for rendering heterogeneous fog [9], [10]. However, 2D noise isn’t realistic because fog itself is in a 3D space. 3D noise would better simulate spatial variations and potentially accurate spatio-temporal variations (assuming displacement is known). In our renderings we use Simplex noise simply because Simplex noise outperforms the more popular Perlin noise. Simplex noise, also developed by Ken Perlin, is designed to overcome the limitations of his classic Perlin noise algorithm. As S. Gustavson explains in [11], on the contrary to Perlin noise, Simplex noise has a gradient well-defined and continuous everywhere and it has a lower computational complexity. Although difficult to visualize, fig. 2 shows a binary representation of the simplex noise space.

We implemented the Simplex Noise [11] and substitute β in eq. 5, with β' the noised scattering coefficient such that: $\beta' = N(\beta)$. Where $N(\beta)$ is the noise function, randomly scaling β in the interval $[\beta/2, 2\beta]$.

The pinhole camera model [12] is used to retrieve the (X, Y, Z) camera coordinates from all image coordinates (x, y) and there associated depth (d). Camera coordinates are then mapped in the noise space to retrieve the ad-hoc noise value. Using this noise value to perturb the scattering coefficient

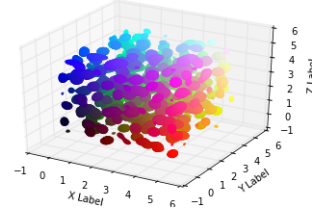


Fig. 2. Binary visualization of the Simplex noise demonstration in 3D space

would lead to acceptable simulation but non-realistic. Indeed it is often considered that the light attenuation is constant, but in fact all rays go through different volumes of atmosphere that exhibits varying scattering coefficients.

To make our model more precise, we average the scattering coefficients along the ray in each pixel. For each pixel, we calculate the noised scattering coefficient β' for 10 discrete steps in $[0, d]$ using the volume of 3D Simplex noise and average these scattering coefficients along the ray. This method is closer to the real life we are not assuming that the scattering coefficient is constant through the ray.

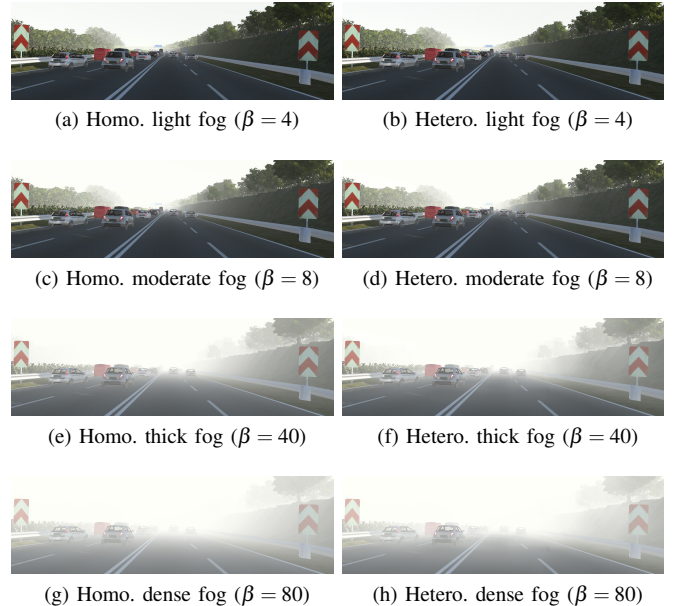


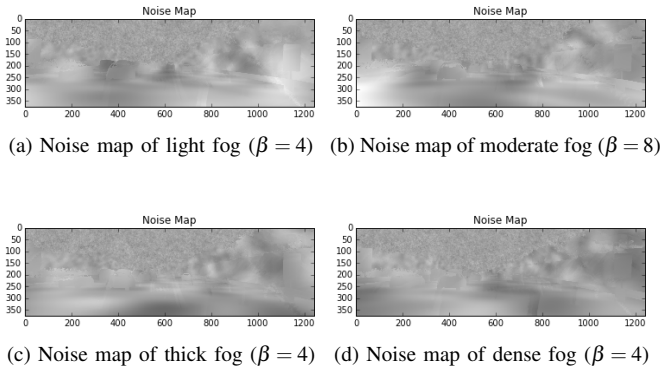
Fig. 3. Sample images showing homogeneous (left) vs. heterogeneous (right) fog, for different scattering coefficient.

In fig. 3, we render fog images with homogeneous (left) and heterogeneous (right) using the two aforementioned methods. As can be observed in fig. 3, fog with simplex noise is unevenly distributed and has different densities at different depths and points which produce more realistic images.

Figure 4 also illustrates the noise map of the heterogeneous fog shown above. Observing the noise maps, we see that the noise close to the camera is smooth and low frequency just like fog, yet in the sky area the frequency of fog is very high. This might be due to an error in our code or an unknown characteristic of the Simplex noise. Nevertheless, the high frequency noise in the sky should further be studied. In practice, note also that we map the camera space to the noise

space with a scaling to ensure keeping a low frequency noise that resembles the density of fog.

Fig. 4. Noise map for the generation of heterogeneous fog



III. BENCHMARKING

In this section we present and explain our results on the performance of some detection algorithms of Python’s OpenCV library with the foggy conditions. After we rendered heterogeneous fog on images, we ran features detectors (Harris Corner, SIFT and SURF) algorithms with OpenCV and ran these algorithms on the foggy images. For Harris corner we compared the number of corners detected in the original clear scene image to the number of corners detected in foggy images with different scattering coefficient values ranging between 1 and 100. For SIFT and SURF, we compare the number of keypoints detected in the original clear scene image to the number of keypoints detected in the foggy images. As shown in Figure 5, performance of all these three detection algorithms decay very quickly with the influence of fog.

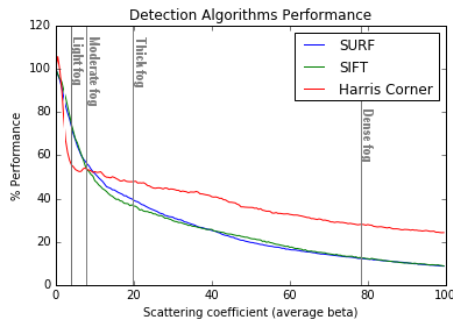


Fig. 5. Impact of fog on detection algorithms

IV. CONCLUSIONS

With this work we have introduced a new method to simulate fog in images from RGB and depth data, accounting for physical and photometric models of small particles. Results are still very preliminary but acknowledge that fog has a great influence on the performance of basic features detectors which let foresee a probable influence on more complex computer vision algorithms. Future work include testing with object detection algorithms and investigating the presence of high frequency noise in distant areas (e.g. sky).

ACKNOWLEDGMENT

This work has been conducted during an internship at Inria Paris under the supervision of Raoul de Charette.

REFERENCES

- [1] W. E. K. Middleton, *Vision through the Atmosphere*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1957, pp. 254–287.
- [2] E. J. McCartney, *Optics of the atmosphere: Scattering by molecules and particles*, 1976.
- [3] J. Shi, Z. Liu, Y. Li, and H. Yuan, “Virtual scene simulation and implementation based on directx11,” in *2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, July 2013, pp. 1620–1623.
- [4] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233–254, Jul 2002. [Online]. Available: <https://doi.org/10.1023/A:1016328200723>
- [5] K. Garg and S. K. Nayar, “Vision and rain,” *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 3–27, Oct. 2007. [Online]. Available: <http://dx.doi.org/10.1007/s11263-006-0028-6>
- [6] E. . Allard, *Memoire sur lintensite et la portee des phares*. Paris: Dunod, 1876.
- [7] A. Saxena, S. H. Chung, and A. Y. Ng, “3-d depth reconstruction from a single still image,” *International journal of computer vision*, vol. 76, no. 1, pp. 53–69, 2008.
- [8] K. Perlin, “Improving noise,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 681–682, Jul. 2002. [Online]. Available: <http://doi.acm.org/10.1145/566654.566636>
- [9] D. Zdrojewska, “Real time rendering of heterogenous fog based on the graphics hardware acceleration,” 2004.
- [10] J.-P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, “Vision enhancement in homogeneous and heterogeneous fog,” *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 2, pp. 6–20, Summer 2012.
- [11] S. Gustavson, “Simplex noise demystified,” 2005.
- [12] K. Ikeuchi, *Computer vision: A reference guide*. Springer Publishing Company, Incorporated, 2014.