Fog Simulation and Refocusing from Stereo Images

Yifei Wang
Department of Electrical Engineering
Stanford University
yfeiwang@stanford.edu

Noam Weinberger
Department of Electrical Engineering
Stanford University
noamw@stanford.edu

Abstract

In this project, we use stereo images to estimate depth and use the depth information to refocus and simulate fog or smoke in the image. This type of simulation is useful for realistic scene rendering and animation. We use common techniques for depth estimation from stereo images and a novel extension of a known technique for fog simulation.

I. Introduction

A key component of computer vision in a 3D world is depth estimation, the ability to determine the distance of various objects in a scene. This is necessary for augmented reality scenes to be rendered, for automated navigation, or any other task where a computer interacts with the real world. In addition to the practical purposes, depth estimation also allows for artistic applications, such as refocusing images and other similar effects.

II. Related Work

Adding fog to images and 3D scenes has been implemented using various techniques. Sun, et al., [1] introduce a model intended to capture the physical properties of fog. They describe fog as water droplets in the air which scatter the ambient light. The more distant an object in the image, the more fog there is for the light from that object to travel through, and therefore the more the light is scattered. They model this with a combination of two images: the original, undistorted image and an entirely white (or gray) image representing the ambient light. Each pixel in the resulting image is a weighted combination of these, with greater weight given to the white image for pixels that are more distant.

Guo, et al., [2] also use the atmospheric scattering model, but change the weights to be less homogenous, even across pixels at the same depth. They add Perlin noise to the fog, in order to add heterogeneity.

Depth estimation is also implemented using many different techniques. A common technique, used for example by Scharstein and Szeliski, [3] is to use the disparity between stereo images. Two images are taken of the same scene, with slight displacement of the camera between capturing each image. The disparity of an object's position in each of the resulting images will be greater for objects closer to the camera.

III. Pipeline

Our pipeline performs two tasks: depth estimation and fog simulation.

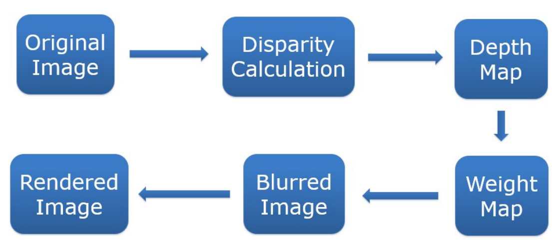

Fig. 1 — Pipeline, consisting of six steps

The first step is getting the stereo images, for which we mainly used the Middlebury Computer Vision Stereo Dataset [4]. From the stereo images, we calculate the disparity for each pixel, which is defined as the displacement of the pixel of the same object. After applying ADMM with a TV prior, we have the depth map.

The weight map is calculated based on a physical model, which will be discussed in Section V. Blending blurred image and the original image based on weight map gives us the rendered image.

IV. Depth Estimation

The principle behind our depth mapping is dual vision. If the same object is observed by two cameras displaced slightly, the object in both images will also have a displacement in the same direction as the cameras (Fig. 2).

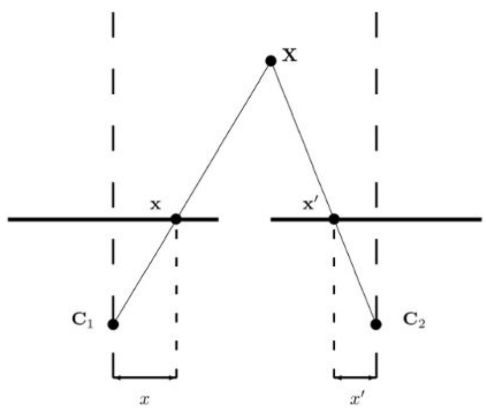

Fig. 2 — Illustration of disparity, which results from different angle of observation [5]

Based on the stereo images (Fig. 3A/B, 4A/B), we used a disparity function in Matlab as the basis of disparity map, which takes into account only the displacement in horizontal direction. In addition to considering displacement in vertical direction, we calculated disparity in different color channels as well as gradient of pixels in both directions and take their average. (Fig 3C, 4C)

Because the disparity map is very noisy, and contains a lot of void pixels, whose disparity is not reliable. We applied alternating direction method of multipliers (ADMM) with a total variation (TV) prior, which uses the augmented Lagrangian:

$$L(x,z,u) = \frac{1}{2}V \cdot \left\|x - x_0\right\|^2 + \lambda\|z\|_1 + \rho\left[u^T(Dx - z) + \frac{1}{2}\|Dx - z\|_2^2\right] \quad (4.1)$$

In this formula, x is the calculated depth map, x_0 is the disparity map given. Dx is the gradient of x in both directions. [6] λ and ρ are coefficients in Lagrangian, in our experiment, they are set respectively as 10 and 0.05. V is a map of the degree of reliability for the pixels, for example, if the disparity of a pixel is invalid, then the corresponding element in V is 0. V is calculated as such:

$$V = exp\left(-min\left\{\left|x - x_{neighbor}\right|^2\right\}/2\right) \quad (4.2)$$

The disparity for invalid pixels are set as a very large negative value by default, which will result in a reliability of 0.

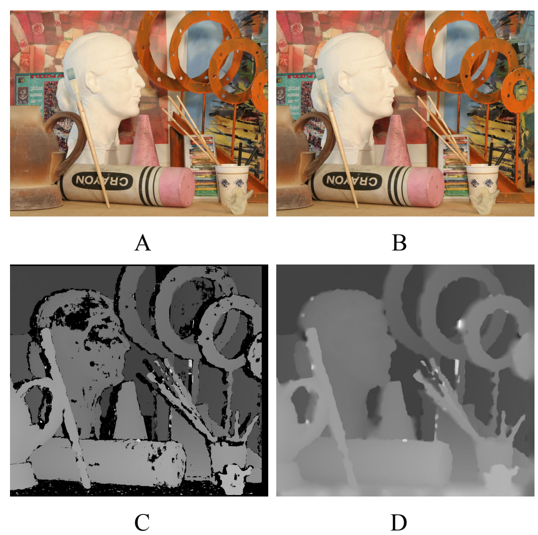

Fig. 3 — Stereo images with a mannequin head. A/B: original stereo images; C: disparity map; D: depth map

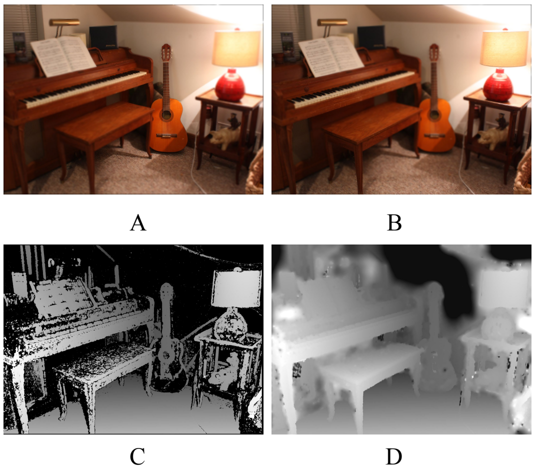

Fig. 4 — Stereo images with an indoor piano. A/B: original stereo images; C: disparity map; D: depth map

The depth map (Fig. 3D, 4D) generated is much more uniform than the disparity map, while a few artificial points still exists. But it is generally a good result for the next step.

V. Fog Simulation and Refocusing

In order to ensure that our fog simulation was working correctly, we applied it to the depth maps we produced, as well as ground-truth depth maps from the Middlebury dataset.

For fog simulation, we used the atmospheric scattering model used by Sun, et al. [1], however we extended it further by adding a blurring factor. This extended model allows for refocusing and fog simulation to be implemented together or independently. It also can produce more realistic fog simulation, since atmospheric scattering also causes blur.

Our model is expressed by the formula:

$$J(x) = I(x)\cdot[1-W(x)] + [\alpha\cdot B(x)+(1-\alpha)\cdot C(x)]\cdot W(x) \tag{5.1}$$

$I(x)$ is the original, undistorted image. $B(x)$ is the image after being blurred with a Gaussian kernel. $C(x)$ is an entirely white (or gray) image representing the ambient light. A weighted combination of these three images creates $J(x)$, the resulting image with simulated fog. The weighting of the images depends on $W(x)$, which is described below, and α, a parameter controlling how much the fog adds light versus blur. If $\alpha = 1$, there is only a refocus or localized blurring effect, and no ambient light. If $\alpha = 0$, the fog has no blurring effect, and this formula becomes more similar to that used by Sun, et al.

The weight matrix $W(x)$ determines how much each pixel is drawn from the distorted image, rather than the original one. The weight matrix can be constructed in two ways: flat fog or local source. When producing a flat fog, $W(x)$ depends only on the depth map. We used a similar formula to that used by Sun, et al.:

$$W(x) = e^{-2.3\times D(x)} \tag{5.2}$$

In this formula, $D(x)$ is the depth value at that pixel. This weighting allows for greater fog intensity at more distant pixels.

When producing a local-source fog, this depth-based weighting is used and is multiplied with a weighting based on the position of the pixel. The formula in this case is:

$$W(x) = e^{-2.3\times D(x)} e^{-\frac{(x-px)}{\sigma x}} e^{-\frac{(y-py)}{\sigma y}} \tag{5.3}$$

In this formula, px and py are the chosen fog source location and σx and σy control how quickly the fog fades in each direction. Using this local-source fog allows for simulating smoke due to fire in a specific location, or some other kind of localization.

To implement refocusing (bokeh effect), the rendered image and the weight matrix W(x) is calculated differently. First α in Eq (5.1) is set to 1. Then the formula used is:

$$W(x) = 1-\left(1-\frac{|D(x)-D_f|}{max\{|D(x)-D_f|\}}\right)^t \tag{5.4}$$

In this formula, D_f is the reference depth, which is set to the depth where an object should be in focus. Adjusting D_f will produce images with different focal distance. Adjusting parameter t, the power coefficient (set to 6 in our code), will create the effect of a camera with different depth of field. In short, this equation makes objects that are farther away from the focus have greater weight from the blurred image. It also ensures that the weight changes smoothly around the focus, like a real depth of field effect.

Some results are shown in the images below:

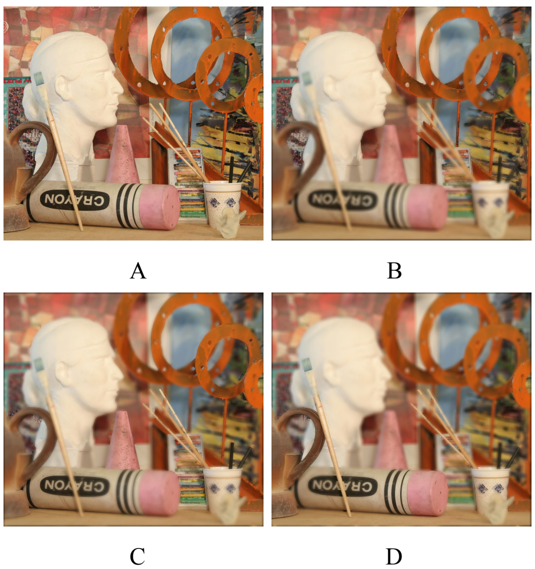

Fig. 5 — Mannequin head. A: Original Image; B: Focused on background; C: Focus on mid-ground; D: Focused on foreground

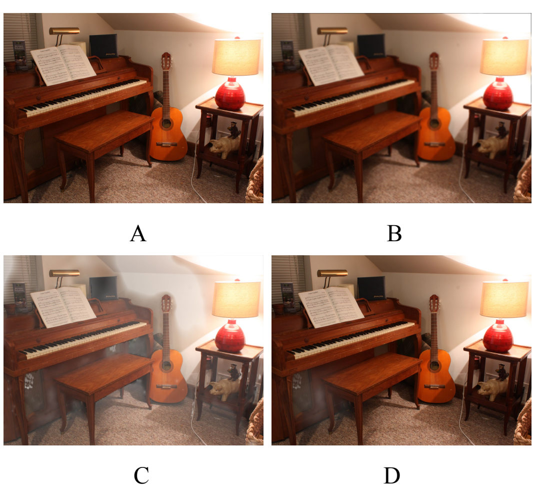

Fig. 6 — Indoor piano. A: Original Image; B: Focused on the closest corner of the chair; C: Image with general smoke effect; D: Image with smoke effect localized near bottom left corner

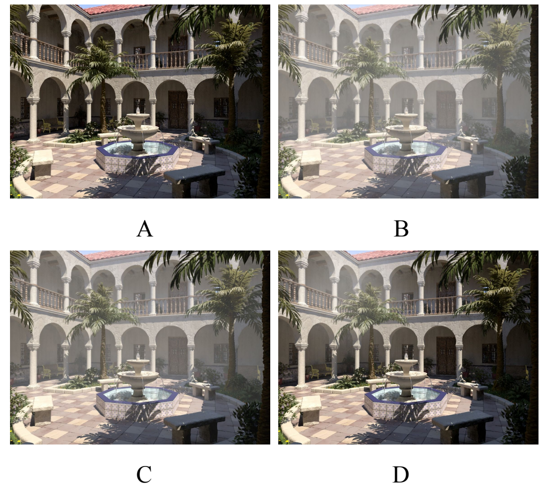

Fig. 7 — Outdoor patio. A: Original Image; B: Image with general fog effect; C: Image with a low intensity fog; D: Image with fog effect localized near upper left corner

VI. Discussion

In conclusion, we built a pipeline for fog simulation and refocusing based on stereo images taken by two cameras displaced by a small distance. First, the disparity map was calculated based on the displacement of objects in both images and optimized by applying ADMM with TV prior. Then, based on the depth map, we blended the ambient light constant, blurred image, and original image together to produce a fog effect or bokeh effect.

The physical model of fog we use is very comprehensive, by tuning the parameters we can simulate different fog simulation or control the range of fog for local source simulation.

Refocusing effect is also realistic, but the blurring is only tuned by changing the ratio of original image and the blurred image. An improvement can be using a depth-based Gaussian filter.

We also tried to build our own dual camera and used the image it obtained to implement pipeline. However, the images were very badly calibrated. Since the objects were too far away, they deformed differently in the images, which made the disparity map very difficult to calculate.

VII. References

[1] Sun, C., Kong, B., He, L., and Tian, Q. *An Algorithm of imaging simulation of fog with different visibility*. 2015 IEEE International Conference on Information and Automation, 2015.

[2] Guo, F., Tang, J., and Xiao X. *Foggy Scene Rendering Based on Transmission Map Estimate.* International Journal of Computer Games Technology, 2014.

[3] Scharstein, D. and Szeliski R. *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.* International Journal of Computer Vision, 2002.

[4] http://vision.middlebury.edu/stereo/data/

[5] http://www.jarnoralli.fi/joomla/research/topics/disparity

[6] Wahlberg, Bo, et al. "An ADMM algorithm for a class of total variation regularized estimation problems." *IFAC Proceedings Volumes* 45.16 (2012): 83-88.

VIII. Acknowledgements

We would like to thank Professor Gordon Wetzstein for his instruction in lectures this quarter. We would also like to thank Felix Heide for his advice and mentorship on this project.