Computational Geometry for Bokeh Effect Generation

Fundamentals of Computer Vision Project Presentation, Fall 2023

Abolfazl Eshagh, Mohammad Yousef Najafi, Sina Imani, Mohammad Bagher Soltani, Alireza Alipanah

Prof. Kasaei
Supervisor: Vida Ramezanian

Table of Contents

What is Bokeh Effect?

- Bokeh is the shallow-depth of field effect which blurs the background of portrait photos (typically) to bring emphasis to the subject in the foreground.
- Used in photography to capture images where the closer objects look sharp and everything else stays out-of-focus.
- Most of the modern cameras can take bokeh images with the help of a good auto-focus hardware.
- For cameras without a good auto-focus hardware, we have to rely on software to generate bokeh images.

Bokeh Effect - Example

Literature Review

Literature Review

- Reviewed 5 papers.
- There are multiple works using Deep Learning methods to generate Bokeh Effect.
- Only few mentionable works use Computational Geometry methods and get actual good results.
- Our work is based on Paper 4(explained in slide 11).
- We will give a brief summary of each in this presentation in order to manage time. For more details refer to project-report.

Depth-aware Blending of Smoothed Images for Bokeh Effect Generation

- Authored by Saikat Duttaa from the Indian Institute of Technology Madras, Chennai, PIN-600036, India.
- The original image and different versions of smoothed images are blended to generate the bokeh effect with the help of a monocular depth estimation network.
- The encoder part of this network consists of a series of convolutional modules (which is a variant of inception module) and downsampling.
- In the decoder part, there is a series of convolutional modules and upsampling with skip connections that add back features from higher resolution in between.

Depth-aware Blending of Smoothed Images for Bokeh Effect Generation - Cont.

The depth-of-field image can be a weighted sum of the original image and differently smoothed versions of the original image:

$$\hat{I}_{bokeh} = W_0 \odot I_{org} + \sum_{i=1}^{n} W_i \odot B(I_{org}, k_i)$$

I_org is the original image, B(I_org, k_i) is image I_org smoothed by blur kernel of size k_i ×k_i, such that for each pixel position (x, y):

$$\sum_{i=0}^{n} W_i[x, y] = 1$$

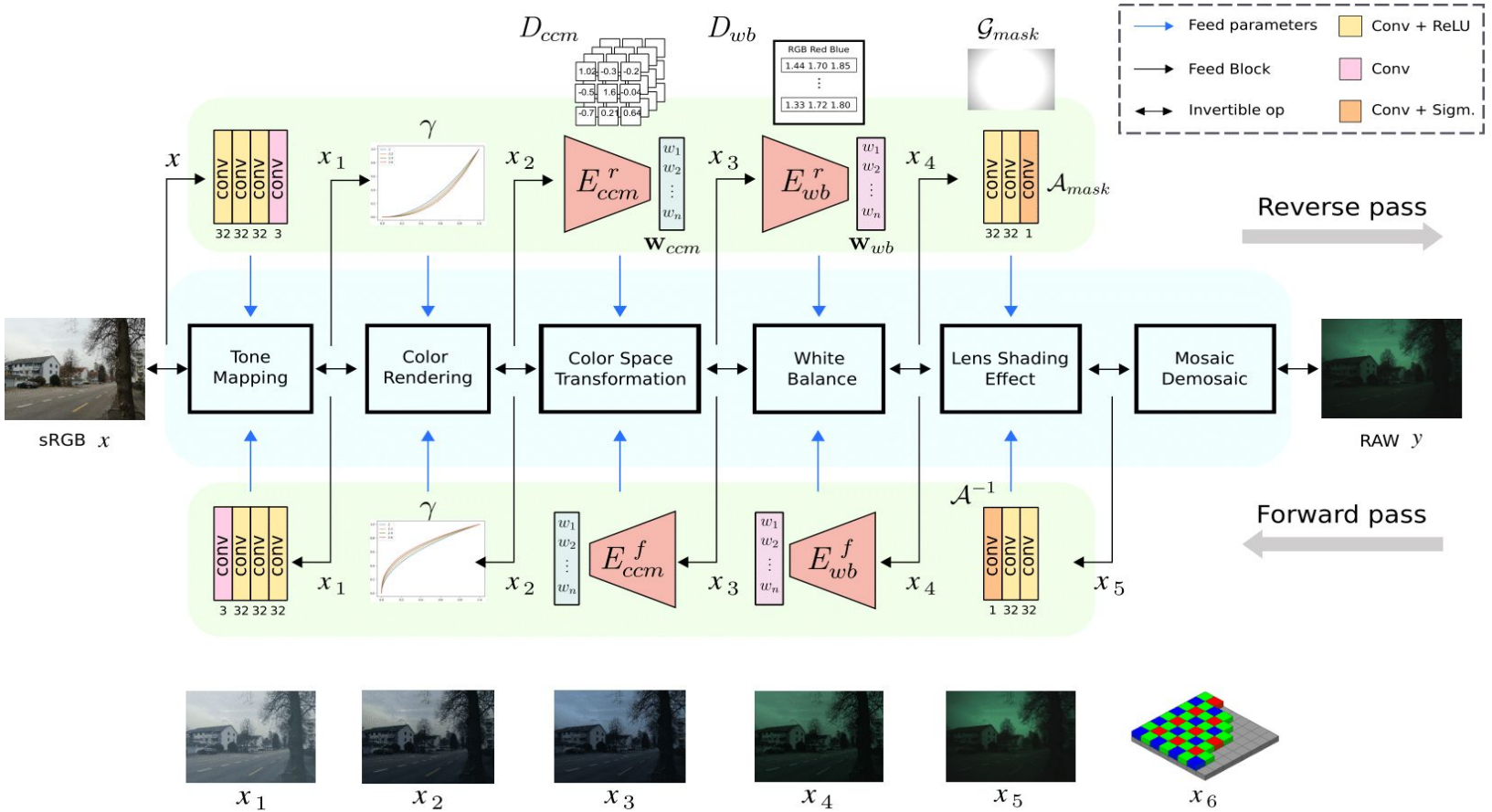
The weights W_i are predicted with the help of a neural network.

Efficient Multi-Lens Bokeh Effect Rendering and Transformation

They design their network EBokehNet for Bokeh effect rendering and transformation considering the following desired features:

(i) the network should allow control of the strength and style of the Bokeh effect. This is fundamental to tackle the novel Bokeh Effect Transformation task.
(ii) The model must be efficient in order to be usable. To achieve this, they adopt the already efficient NAFNet architecture and simplify it further by reducing the number of encoder-decoder blocks.
(iii) The model should be able to render or convert the Bokeh effect of one lens to the effect of another lens without modifying the sharp foreground regions in the image.

Steps of EBokehNet method



BGGAN: Bokeh-Glass Generative Adversarial Network for Rendering Realistic Bokeh

- Achieves state-of-the-art performance on AIM 2020 Bokeh Challenge.
- The first GAN-based method for synthetic bokeh effect rendering
- Proposes a novel generator called Glass-Net, which generates bokeh images not relying on complex hardware.
- The proposed BGGAN uses multiple PatchGAN discriminators operating on different patch sizes as a multi-scale discriminator to adversarially train the GlassNet generator to improve results across both local details and global image coherence.
- It uses a weighted combination of pixel-level losses like L1 and SSIM along with perceptual losses from VGGNet and adversarial losses from the discriminator to optimize the GlassNet generator.

Using Depth Mapping to realize Bokeh effect with a single camera Android device

- uses operators supported by tflite framework to reimplement IN to make sure all of the model operations compute on smartphone GPU.

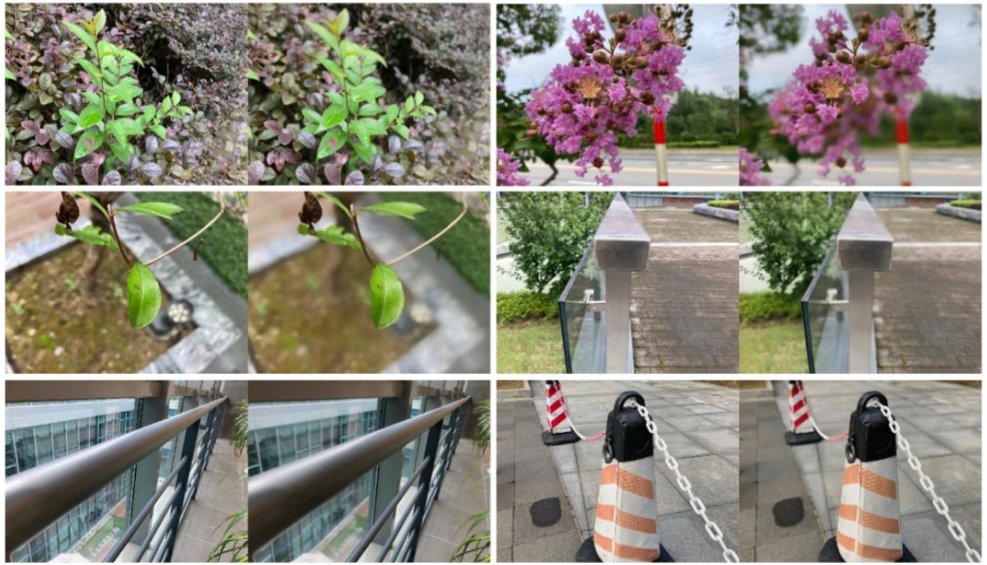


Fig. 8. Images taken by iPhone SE2 are on the left, and bokeh images processed by BGGAN are on the right.

A Closed Form Solution to Natural Image Matting

- Authored by Anat Levin, Dani Lischinski, and Yair Weiss from the School of Computer Science and Engineering, Hebrew University.
- Interactive digital matting is the process of extracting a foreground object from an image based on limited user input.
- We must estimate the foreground and the background colors, as well as the foreground opacity (“alpha matte”) from a single color measurement.

A Closed Form Solution to Natural Image Matting - Cont.

The color of the i-th pixel is a linear combination of the corresponding foreground and background colors:

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

α_i is the pixel's foreground opacity, all quantities on the right hand side of the equation are unknown.

Goal is to find α, a and b minimizing the cost function:

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2 \right)$$

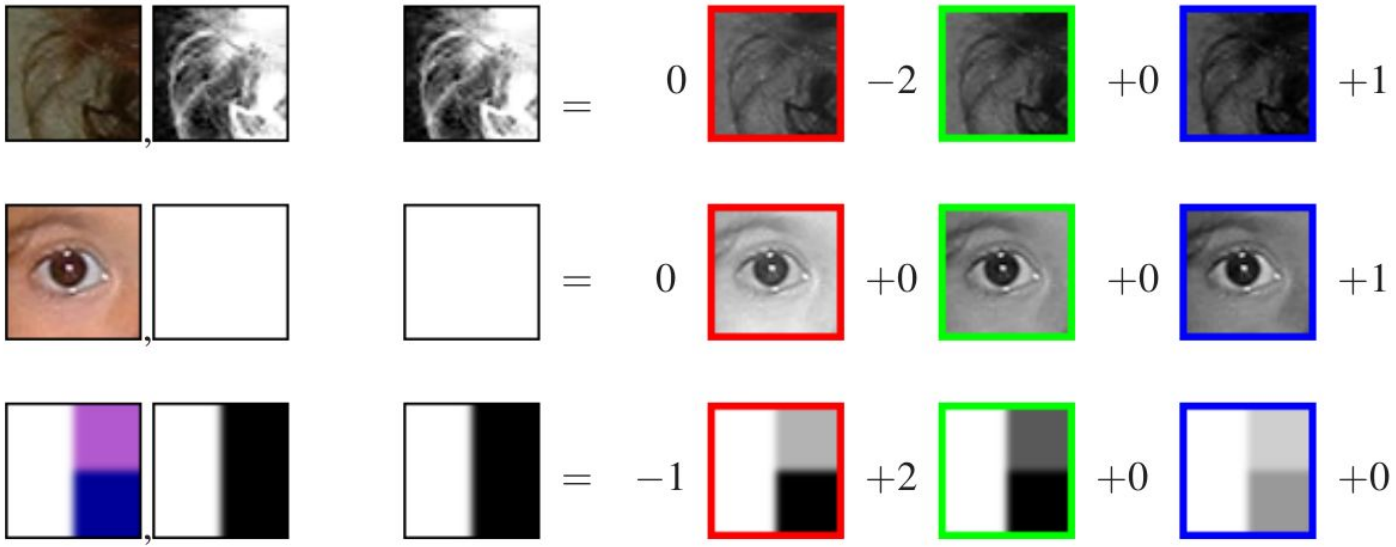
A Closed Form Solution to Natural Image Matting - Cont.

The matting Laplacian matrix L is a symmetric positive definite matrix. This matrix may also be written as L = D−W, where D is a diagonal matrix D(i,i) = ∑jW(i, j) and W is a symmetric matrix, whose off-diagonal entries are defined by:

$$\sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + (I_i - \mu_k)(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k) \right) \right)$$

Alpha Matte as A Local Linear Combination of the Image

The underlying idea proposed by Levin et al is to represent the alpha values as locally a linear combination of image intensity



Defocus as A Local Linear Combination of the Image

Alpha	<->	Defocus
Confident Pixels	<->	Edge Pixels
Prior Values	<->	Sparse Defocus Map

$$E(d) = d^T L d + \lambda (d - \hat{d})^T D (d - \hat{d})$$

Implementation

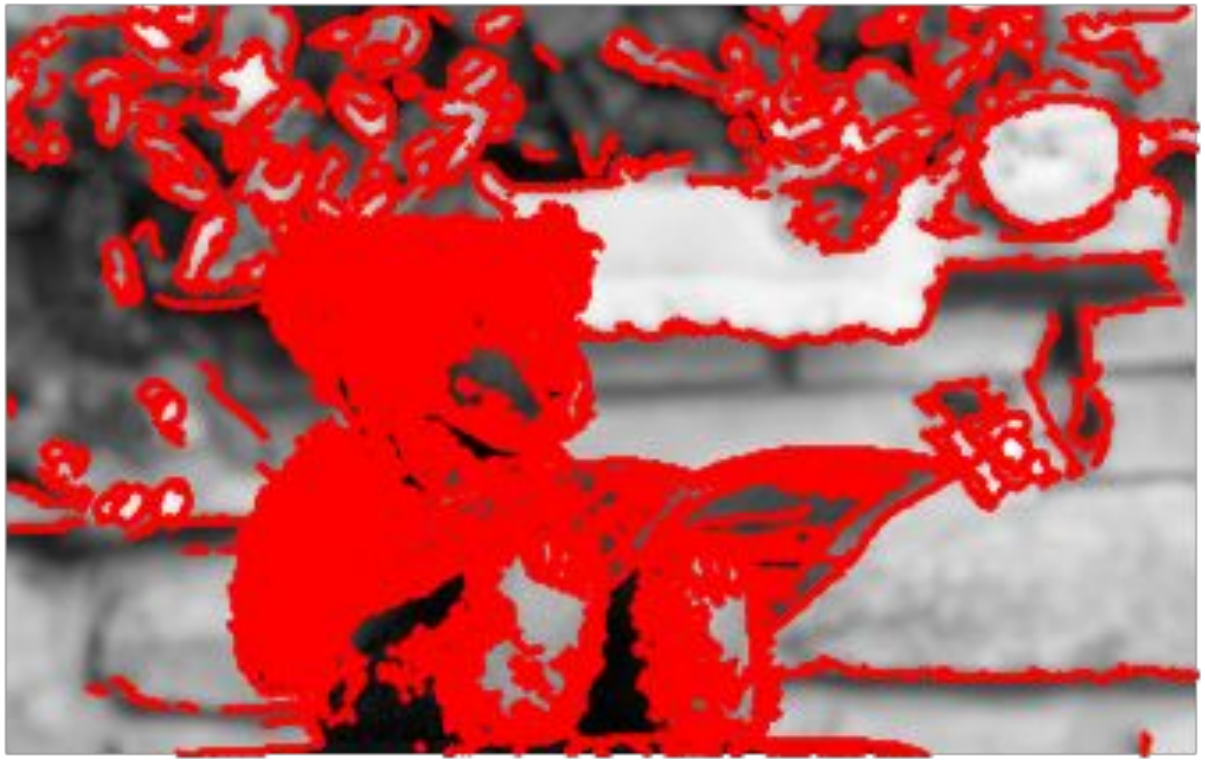
Implementation - Idea

- Based on paper 4.
- We use Computational Geometry methods to generate Bokeh Effect.
- The main idea is to use Depth Mapping to generate Bokeh Effect.

Implementation - Procedure

Gradient Ratio	Sparse Defocus Map Generation	Sparse Defocus Map Propagation	Full Defocus Map with Matting Laplacian	Bokeh Effect
Calculate the gradients in vertical and horizontal directions and compute the magnitude of the gradient in the first (original in method 1) and the second re-blurred images. Then calculate the ratio of the second gradient by the first gradient. Use median blur to reduce noise.	Perform canny edge detection on the original image. Then compute the sparse defocus by calculating the estimated original blur sigma at edge pixel.	Many pixels in the sparse defocus map are NaN. Use NaN in-paint interpolation to fill these values.	Apply matting laplacian to estimate the transparency mask α (matte).	Use a threshold to compute the foreground and background masks from the matte. Then, apply gaussian blurring on the background and merge the two sections.

Sigma Estimation Candidates



Foreground and Background Masks

Foreground Mask

Background Mask

Implementation - Results

Original Image

Defocus Blur Map with first method

Defocus Blur Map with second method

Original Image

Defocus Blur Map with first method

Defocus Blur Map with second method

Implementation - Results

Original Image

Bokeh Effect Applied

Implementation - Results

Implementation: Libraries Used

- OpenCV
- NumPy
- SciPy
- Scikit-Image
- Matplotlib

References

- Depth-aware Blending of Smoothed Images for Bokeh Effect Generation
- Efficient Multi-Lens Bokeh Effect Rendering and Transformation
- BGGAN: Bokeh-Glass Generative Adversarial Network for Rendering Realistic Bokeh
- Using Depth Mapping to realize Bokeh effect with a single camera Android device
- A Closed Form Solution to Natural Image Matting