Brandon Johnson and Charles Mason
Bjohnson1@rollins.edu
Cmason@rollins.edu

Miniature: Computation Photography Project 2

The focus of this project was to create an algorithm that alters an image to match the aesthetics of miniature faking also known as the diorama effect. Diorama processing creates the illusion of scaling an image down through imitating a shallow depth of field. This is accomplished by creating a focal point traditionally within the middle section of the image and gradually obscuring the contents of the image through increasing the blur as you deviate farther from the center. In order to produce a believable effect, photos that were taking at above angle were used, giving the illusion that someone is standing over a miniature model.

Photo A and Photo B represent an aerial image of midtown New York City. Photo C and Photo D represent an image of a busy street in Chandni Chowk, Delhi. Photo A and C artistically differs from the aesthetics conveyed in Photo B and C by presenting a uniformly focused image. Photos B and D are generated by using Photos A and C, respectively, as input in our miniature function. The stark contrast between both sets of photos is attributed the variance in intensity of distortion and blur as the image veers from the center. More specifically, in Photo B, the bottom portion of the image is marked by buildings with indefinitive characteristics. As the image approaches the center, the lines of the individual buildings become more distinct thus creating a sense of closeness to the subject by virtually distorting the depth of field. When the image reaches the upper most region, the image transitions into a green landscape, but also gradually become blurrier. This technique was accomplished by using a 3x3 kernel with a normalized weight of 1/9.

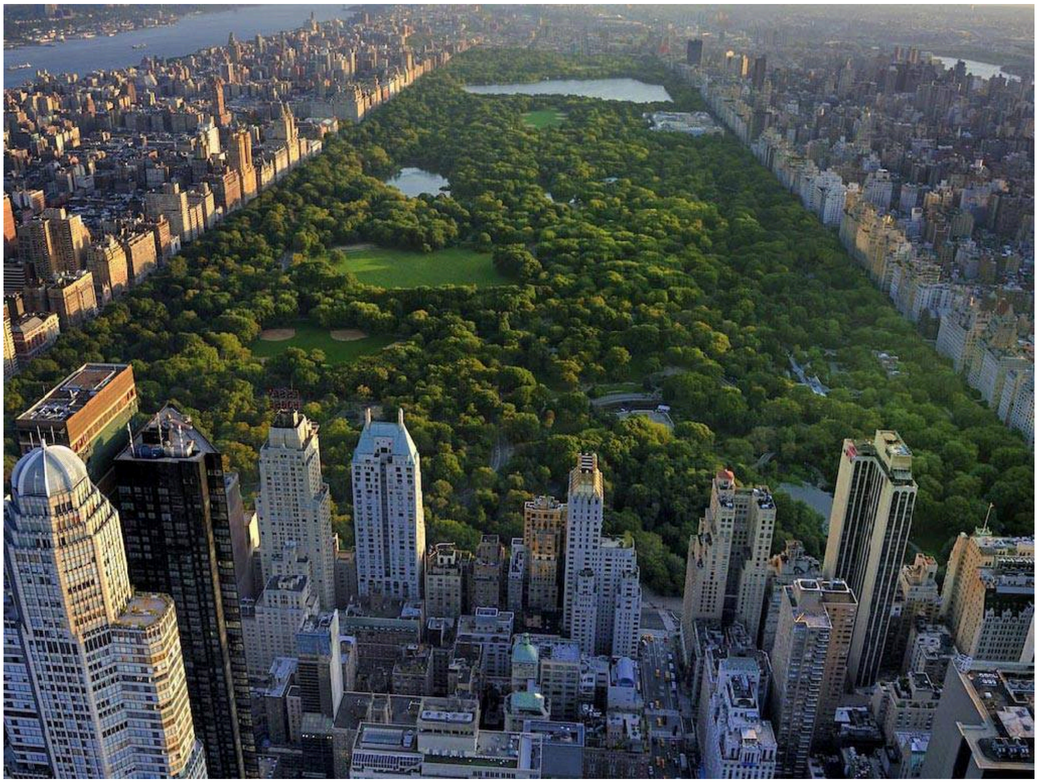


Photo A: Uniformly focused image of
midtown Manhattan

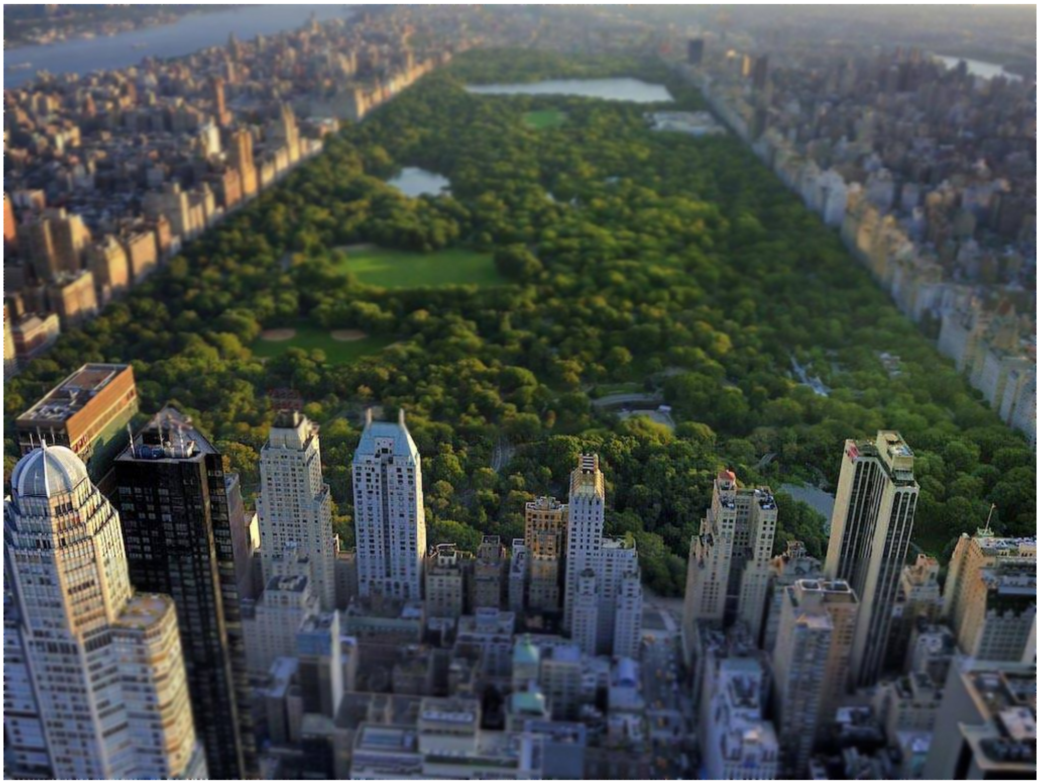


Photo B: Image of midtown Manhattan with
unevenly distributed intensities of blur

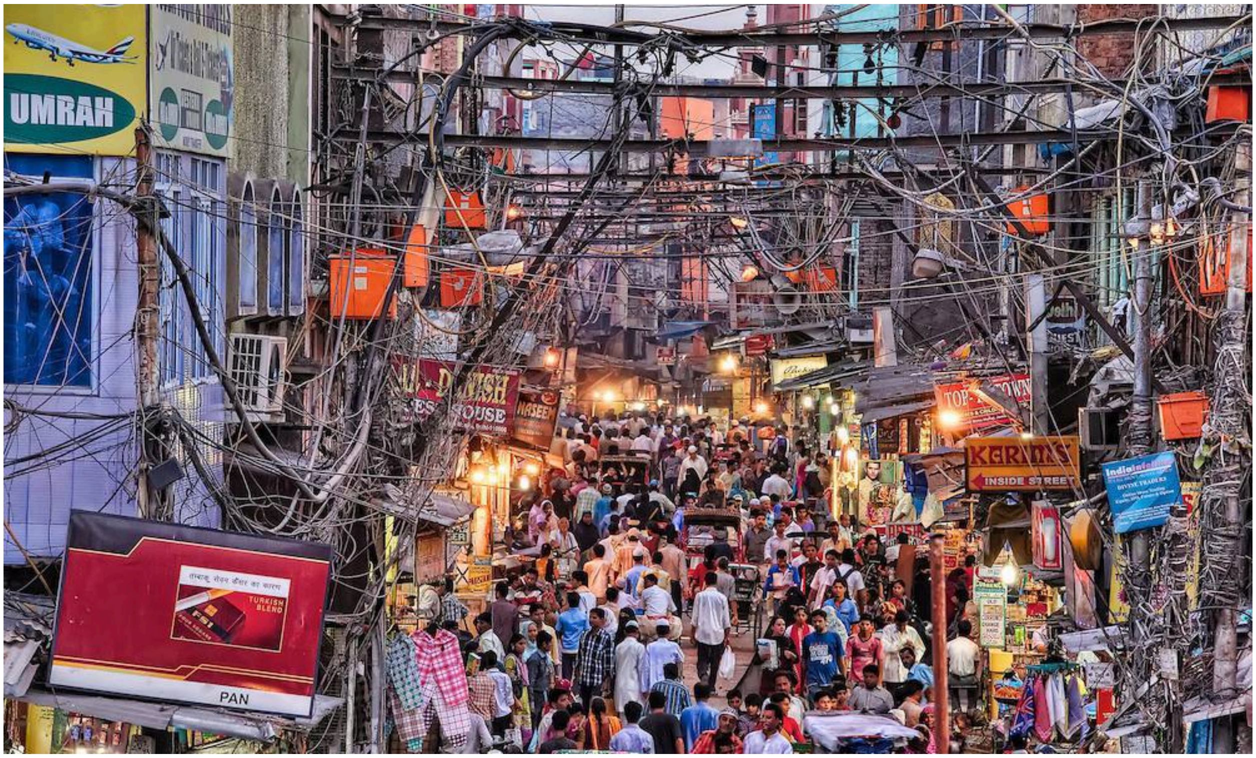


Photo C: Uniformly focused image of
Chandni Chowk, Delhi

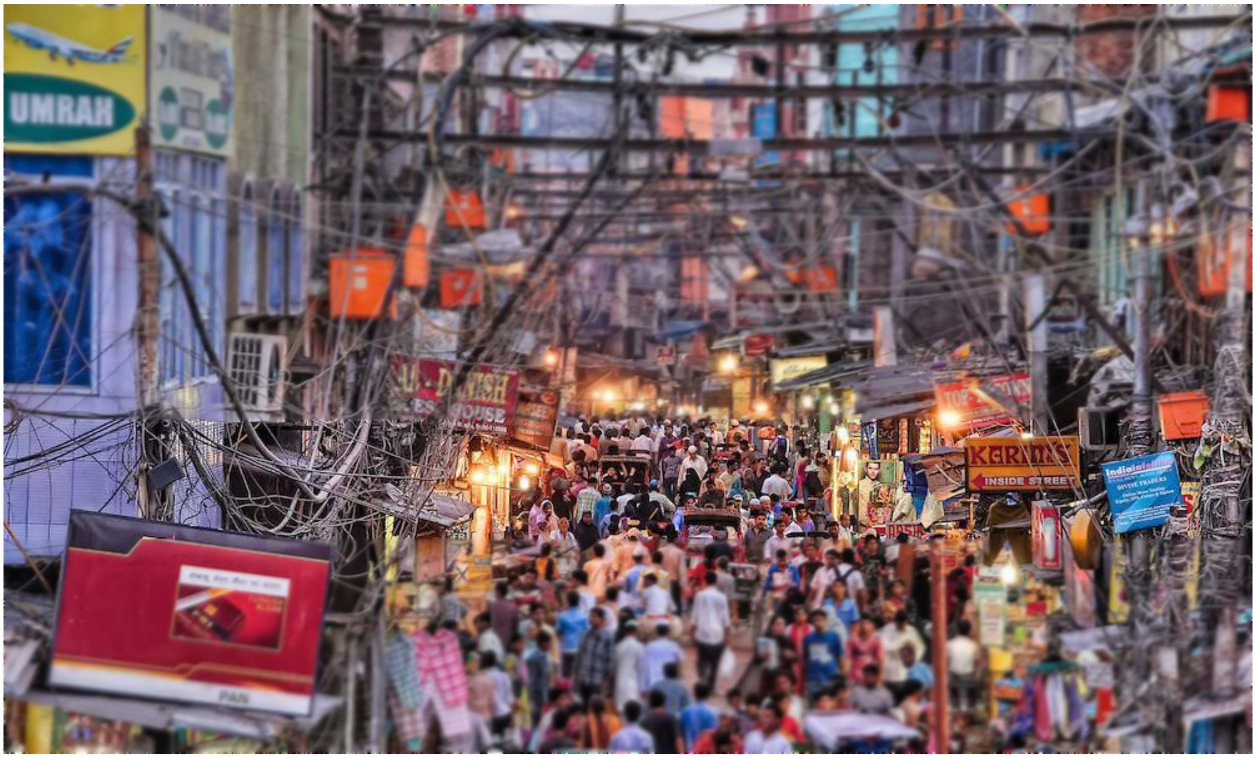
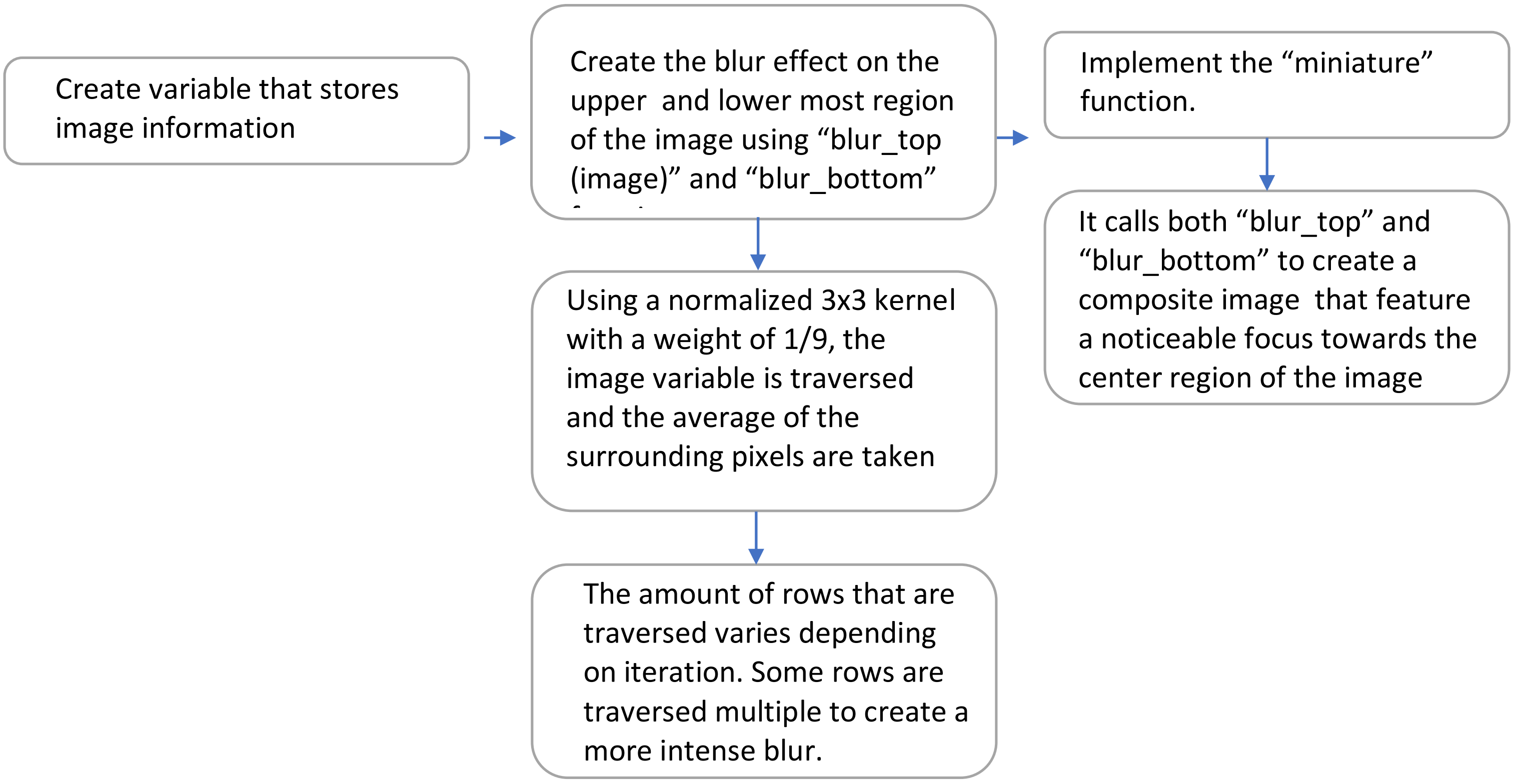


Photo D: Image of Chandi, Delhi with
unevenly distributed intensities of blur



Create variable that stores image information

Create the blur effect on the upper and lower most region of the image using “blur_top (image)” and “blur_bottom”

Using a normalized 3x3 kernel with a weight of 1/9, the image variable is traversed and the average of the surrounding pixels are taken

The amount of rows that are traversed varies depending on iteration. Some rows are traversed multiple to create a more intense blur.

Implement the “miniature” function.

It calls both “blur_top” and “blur_bottom” to create a composite image that feature a noticeable focus towards the center region of the image

The above infographic details the necessary steps required to implement our miniature algorithm. Initially, the function “blur_top” is created which takes in the parameter of “image.” This function instantiates a 3x3 kernel with a weight of 1/9 to traverse the image creating an average box filter. It is similar to the native cv2 function “filter2d”; however, we created an outer for loop that iterates the traversal 3 times as a means of processing the uppermost rows more to get a blurrier effect. As the image transitions towards the center region of its surface, the blur effect dilutes. This is because within the “blur_top” function, the amount of rows that are processed increases with each iteration. The same process is executed when implementing the “blur_bottom” function, but instead of decreasing the end position of the rows throughout each iteration, we increase the starting position of each row during every iteration to increase the blur as the image goes towards the bottom. Finally, the “miniature” function is implemented. It calls both the “blur_bottom” and “blur_top” to form a composite image that includes both top and bottom blur creating a “fake” focus in the center.