```
import numpy as np
import cv2 as cv
import math

def gaussian(m,n,sigma):
    gauss = np.zeros((m,n))
    pi=3.1416
    m //=2
    n //=2
    for x in range(-m,m+1):
        for y in range(-n,n+1):
            r  = (x**2 + y**2)/(2*sigma**2)
            gauss[x+m][y+n] = math.exp(-r)/(2*pi*sigma**2)
    return gauss

def bilateral(img,filt,sigma):
    w1 = filt.shape[0]
    h2 = filt.shape[1]
    w = w1//2
    h = h2//2
    img = img/255
    m = img.shape[0]
    n = img.shape[1]
    out = np.zeros((m,n),np.float32)
    pi = 3.1416

    for i in range(m):
        for j in range(n):
            rs = 0.0
            factor = 0.0
            range_domain = np.zeros((w1,h2),np.float32)
            for x in range(-w,w+1):
                for y in range(-h,h+1):
                    if i-x>=0 and i-x<m and j-y>=0 and j-y<n:
                        v = img[i][j] - img[i-x][j-y]
                        v = (math.exp(-(v*v)/(2*sigma*sigma))/((math.sqrt(2*pi))*sigma))
                        range_domain[x+w][y+h] = filt[x+w][y+h] * v
                        factor += v
            for x in range(-w,w+1):
                for y in range(-h,h+1):
                    if i-x>=0 and i-x<m and j-y>=0 and j-y<n:
                        rs += (range_domain[x+w][y+h] * img[i-x][j-y])
            rs = (rs/factor)
            out[i][j] = rs

    out = out*255
    out = cv.normalize(out, None, 0, 1.0,cv.NORM_MINMAX, dtype=cv.CV_32F)
    return out

img = cv.imread("Lena.jpg",cv.IMREAD_GRAYSCALE)
cv.imshow("original",img)

g_s = int(input("Enter value of gaussian sigma:\n"))
p = (2*g_s) + 1
gauss_filt = gaussian(p,p,g_s)

w = int(input("Enter kernel width:\n"))
h = int(input("Enter kernel height:\n"))
g_r = int(input("Enter value of sigma:\n"))

avg_filt = np.ones((w,h))/(w*h)
cv.imshow("final_guass_bilateral",bilateral(img,gauss_filt,g_r))
cv.imshow("final_average_bilateral",bilateral(img,avg_filt,g_s))


cv.waitKey()

cv.destroyAllWindows()
```