In [2]:

```
import cv2 # tested with the 3.1.0 version
import numpy as np
import matplotlib.pyplot as plt
import os
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
```

1. Basic Image Operations

1.1 Image Read/Write ¶

In [3]:

```
# a) read image.
def my_imread(imagepath):
    bgr_img = cv2.imread(imagepath,1) # Read the image from imagepath.
    return bgr_img

IMG_NAME = '37073.jpg' # Choose an image filename from "data/img/"
IMG_PATH = os.path.join('data/img/', IMG_NAME)
BGR_image = my_imread(IMG_PATH)
```

In [4]:

```
# Optional: uncomment those lines to display the image using cv2.

cv2.imshow('image', BGR_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [15]:

```
# b) Display the correct color in RGB
B,G,R = cv2.split(BGR_image)
RGB_image = cv2.merge([R,G,B])
plt.imshow(RGB_image)
```

Out[15]:

```
<matplotlib.image.AxesImage at 0x2799977bb38>
```

In [17]:

```
# c) Write the same image in the correct color and Grayscale and save as
# "problem1_rgb.jpg" and "problem1_gray.jpg".
RGB_FILENAME = "problem1_color.jpg"
GRAY_FILENAME = "problem1_gray.jpg"
cv2.imwrite(RGB_FILENAME,BGR_image) # save the image in the correct color forma
t.
cv2.imwrite(GRAY_FILENAME,cv2.cvtColor(BGR_image, cv2.COLOR_BGR2GRAY)) # save th
e image in grayscale format.
# Once the two images have been generated in the current directory,
# run the next block to show your results.
```

Out[17]:

```
True
```

Display saved color image from current folder.

Display saved grayscale image from current folder.

1.2 Image Smoothing

In [36]:

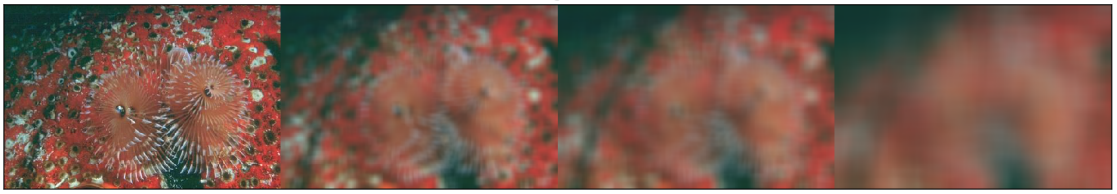
```
# Optional: plots an image
def plot_helper(rgb_img, title):
    plt.figure(figsize=(15, 15)) # the figure size can be adjusted to make
 sure the images are clear for grading.
    plt.subplot(111)
    plt.imshow(rgb_img)
    plt.title(title)
    plt.xticks([])
    plt.yticks([])
    plt.show()
```

In [38]:

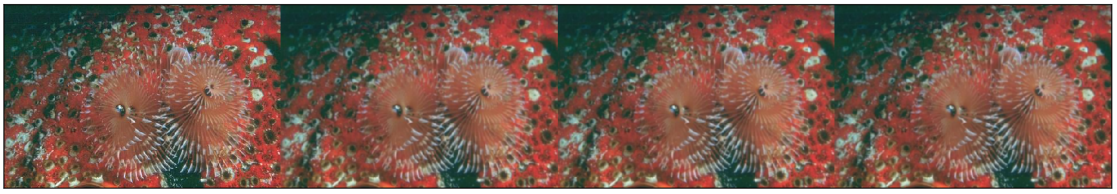
```
# IMAGES = ['12084.jpg', '21077.jpg', '24077.jpg'] # choose 3 images from "data/
img"
KERNEL_SIZES = [15,25,55] # choose at least 3 different filter sizes
print ("kernel sizes are: " + str(KERNEL_SIZES))
for i in range(len(IMAGES)):
    imagepath = os.path.join('data/img/', IMAGES[i])
    image = my_imread(imagepath)
    B,G,R = cv2.split(image)
    image = cv2.merge([R,G,B])
    avg_blurs = [image]
    gaus_blurs = [image]
    for j in range(len(KERNEL_SIZES)):
        kernel_size = KERNEL_SIZES[j]
        # a) average smoothing
        avg_blur = cv2.blur(image,(kernel_size,kernel_size))
        avg_blurs.append(avg_blur)
        # b) gaussian smoothing
        gaus_blur = cv2.GaussianBlur(image,(kernel_size,kernel_size),1)
        gaus_blurs.append(gaus_blur)
    avg_stack = np.hstack(avg_blurs)
    gaus_stack = np.hstack(gaus_blurs)
    # Optional: you could choose to use plot_helper to plot avg_stack and gaus_s
tack.
    # Make sure they are in RGB before using plot_helper.
    plot_helper(avg_stack , "Average Blur")
    plot_helper(gaus_stack , "Gaussian Blur")
```

```
kernel sizes are: [15, 25, 55]
```

Average Blur



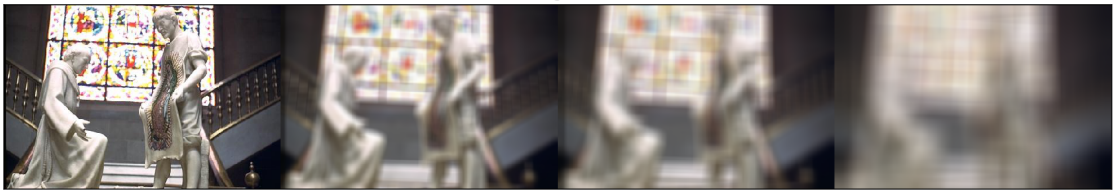
Gaussian Blur



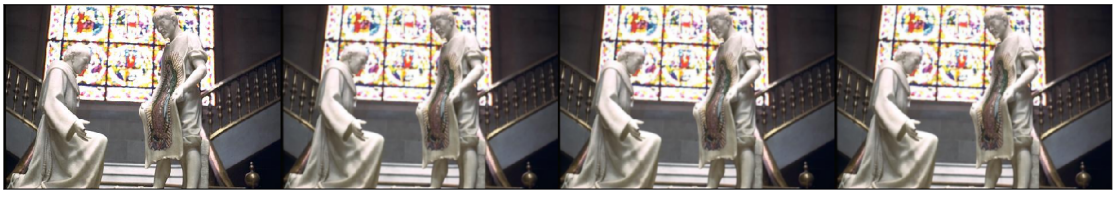
Average Blur

Gaussian Blur

Average Blur



Gaussian Blur



c) What do you observe? (Please briefly answer this question in this block)

For Average Blur, larger size of filter will make picture blurrer.

For Gaussian Blur, the size of filter does not has any obvious influence on pictures.

1.3 Denoising

In [45]:

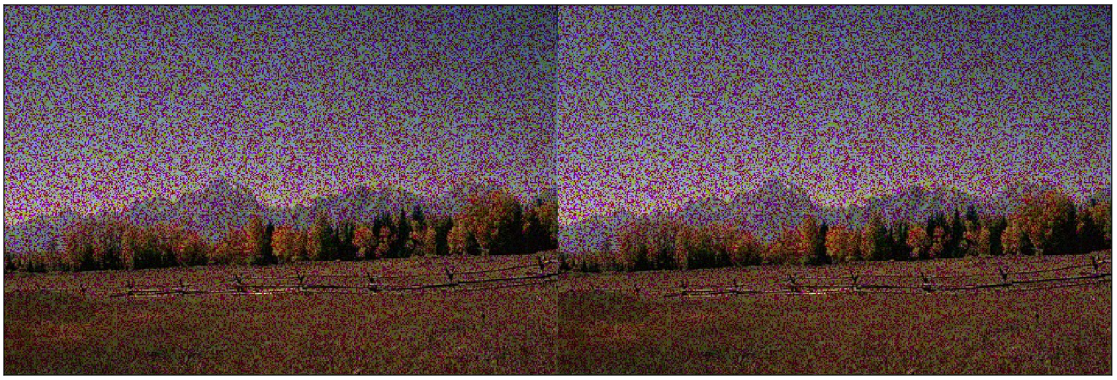
```
import glob
imagepaths = glob.glob('data/snp/*.jpg')
```

In [46]:

```
MEDIAN_FILTERS = [1, 3, 5, 7, 9] # Filter sizes for each image.
for i, imagepath in enumerate(imagepaths):
    image = my_imread(imagepath)
    B,G,R = cv2.split(image)
    image = cv2.merge([R,G,B])
    median = cv2.medianBlur(image,MEDIAN_FILTERS[i]) # Apply the median filter o
n this image.
    print ("Filter size "+ str(MEDIAN_FILTERS[i]))
    stack = np.hstack((image, median))
    # Optional: you could choose to use plot_helper to plot stacked image.
    # Make sure it is in RGB before using plot_helper.
    plot_helper(stack, 'Median Blur: %dx%d'%(MEDIAN_FILTERS[i], MEDIAN_FILTERS[i
]))
```

Filter size 1

Median Blur: 1x1



Filter size 3

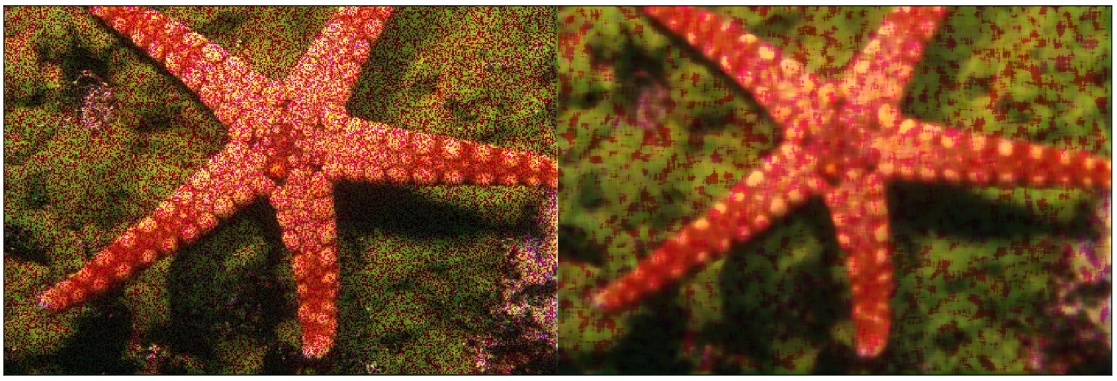
Median Blur: 3x3

Filter size 5

Median Blur: 5x5

Filter size 7

Median Blur: 7x7



Filter size 9

Median Blur: 9x9