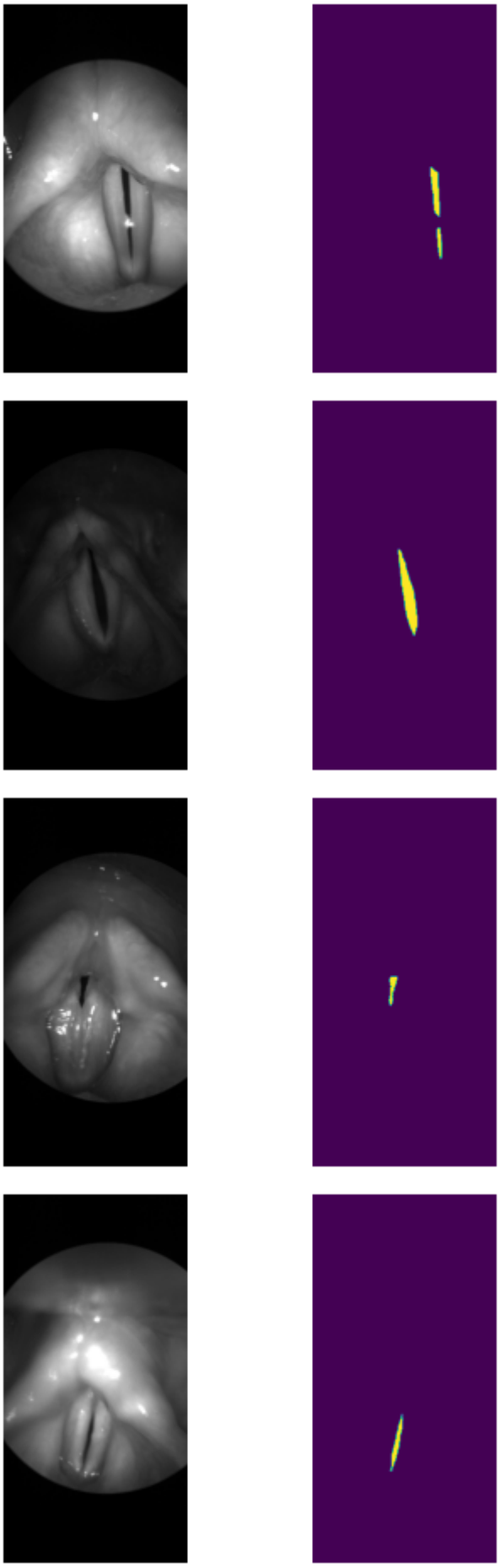
In [2]:
```
import os
import json
import numpy as np
import os.path
import matplotlib.pyplot as plt
```

In [3]:
```
path = "Mini_BAGLS_dataset"
files = os.listdir(path)
```

In [4]:
```
from pathlib import Path
path_pathlib = Path(path)
path_pathlib
```

Out[4]:
```
WindowsPath('Mini_BAGLS_dataset')
```

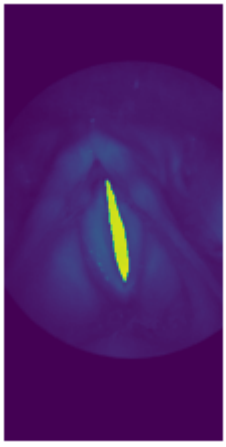
In [31]:
```
#Task 1/4
#Load four arbitrary images and their corresponding segmentation masks
and metadatafrom PIL import Image
import random
from PIL import Image

a = [];b = []
filenames = [f for f in files if '.meta' in f]
for i in range(4):
    l = (random.choice(filenames).split(".")[0])
    img_path = os.path.join(path, l + ".png")
    seg_path = os.path.join(path, l + "_seg.png")

    img = Image.open(img_path)
    seg = Image.open(seg_path)
    a.append(img);b.append(seg)
    fig, (ax1, ax2) = plt.subplots(1, 2)

    ax1.imshow(img);ax1.axis("off");
    ax2.imshow(seg);ax2.axis("off");
```

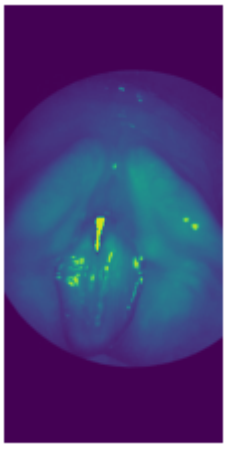


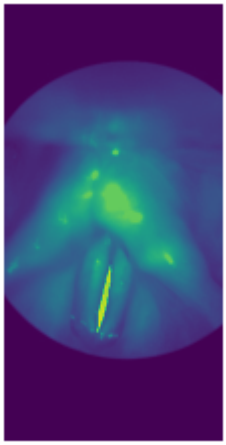
In [32]:

```
#Task 2/4
#Plot the images with the segmentation masks overlaid in a picture.
#To show all four resulting figures, please use the subplots() method
```

```
from matplotlib.pyplot.
#Each subplot should have the “Subject disorder status” as the title
(contained in the.meta file).

for i in range(4):
    img = a[i].convert('L')
    seg = b[i].convert('L')
    new_img = Image.blend(img, seg, 0.5)
    fig, (ax1) = plt.subplots(1)
    ax1.imshow(new_img);ax1.axis("off");
```



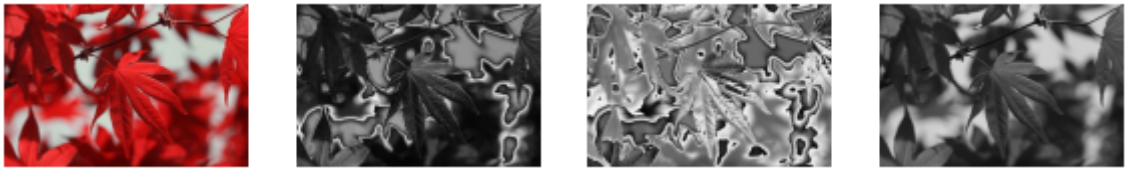




In [33]:

```
#Tasks 3/4
#Load the "leaves.jpg" that we have provided for you as RGB image.
#Implement the following three variations to convert an image from RGB
to Grayscale.
#Again, use the subplots() function to show all three variants side by
side

from PIL import Image
import matplotlib.image as mp

fig = plt.figure(figsize=(10, 10))
rows = 1
columns = 4

leaves_path = os.path.join(path + "\leaves.jpg")
leaves_image = mp.imread(leaves_path)

R = leaves_image[:,:,0]
G = leaves_image[:,:,1]
B = leaves_image[:,:,2]

img_light = np.divide((np.maximum(R,G,B)+np.minimum(R,G,B)),2)
img_avg = np.divide((R+G+B),3)
img_lum = 0.2989 * R + 0.5870 * G + 0.1140 * B

fig.add_subplot(rows, columns, 1)
plt.imshow(leaves_image);plt.axis("off");

fig.add_subplot(rows, columns, 2)
plt.imshow(img_light, cmap='gray');plt.axis("off");
```

```
fig.add_subplot(rows, columns, 3)
plt.imshow(img_avg, cmap='gray');plt.axis("off");

fig.add_subplot(rows, columns, 4)
plt.imshow(img_lum, cmap='gray');plt.axis("off");
```



In [35]:

```
#Tasks 4/4
#Answer the following question: which method for RGB to grayscale
conversion is the preferred one?
#State in 1-2 sentences why you think this.

Answer:
The Luminosity Method is the preferred method for grayscale conversion.

The amount of red, green and blue colours are properly calculated, and
they blend perfectly to produce a proper grayscale luminious picture.
This is due to the fact that when compared with other grayscale
conversions, the image created using the luminosity method is having
more clarity.
```