

In [2]:

```
import PIL
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from skimage.data import imread
```

In [3]:

```
# df = pd.read_csv("../train_ship_segmentations_v2.csv", index_col=0).dropna()
# display(df.head())
# img_masks = [df['EncodedPixels'] ['000155de5.jpg']]
# print(img_masks)
```

In [4]:

```

masks = pd.read_csv("../train_ship_segmentations_v2.csv").dropna().drop_duplicates('ImageId')
display(masks.head())
print(masks.shape[0])
image_list = []
for index in masks.index:
    image_list.append(masks.loc[index, 'ImageId'])
print(image_list)
# img_masks = masks.loc[masks['ImageId'] == '000155de5.jpg', 'EncodedPixels'].tolist()
# print(img_masks)

```

	Imageld	EncodedPixels
2	000155de5.jpg	264661 17 265429 33 266197 33 266965 33 267733...
3	000194a2d.jpg	360486 1 361252 4 362019 5 362785 8 363552 10 ...
9	00021ddc3.jpg	108287 1 109054 3 109821 4 110588 5 111356 5 1...
18	0002756f7.jpg	255784 2 256552 4 257319 7 258087 9 258854 12 ...
22	00031f145.jpg	340363 1 341129 4 341896 6 342663 7 343429 10 ...

```
42556
['000155de5.jpg', '000194a2d.jpg', '00021ddc3.jpg', '0002756f7.jpg', '00031f145.jpg', '000532683.jpg', '00053c6ba.jpg', '0005d01c8.jpg', '0005d6d95.jpg', '0006c52e8.jpg', '000d42241.jpg', '000e6378b.jpg', '000fd9827.jpg', '00113a75c.jpg', '001234638.jpg', '001566f7c.jpg', '0017c19d6.jpg', '0018e5d6c.jpg', '0019fc4d8.jpg', '001aee007.jpg', '001bcf222.jpg', '001e418bc.jpg', '001eb2794.jpg', '001ed8b9b.jpg', '001eda01f.jpg', '0022f64e5.jpg', '00269a792.jpg', '0027854cc.jpg', '002868a5c.jp
```

In [6]:

```
#### reference:https://www.kaggle.com/paulorzp/run-length-encode-and-decode
#### https://www.kaggle.com/inversion/run-length-decoding-quick-start

def rle_encode(img):
    '''
    img: numpy array, 1 - mask, 0 - background
    Returns run length as string formatted
    '''
    pixels = img.flatten()
    pixels = np.concatenate([[0], pixels, [0]])
    runs = np.where(pixels[1:] != pixels[:-1])[0] + 1
    runs[1::2] -= runs[:2]
    return ' '.join(str(x) for x in runs)

def rle_decode(mask_rle, shape=(768, 768)):
    '''
    mask_rle: run-length as string formatted (start length)
    shape: (height,width) of array to return
    Returns numpy array, 1 - mask, 0 - background

    '''
    s = mask_rle.split()
    starts, lengths = [np.asarray(x, dtype=int) for x in (s[0:][::2], s[1:][::2])]
    starts -= 1
    ends = starts + lengths
    img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
    for lo, hi in zip(starts, ends):
        img[lo:hi] = 1
    return img.reshape(shape).T
```

In [7]:

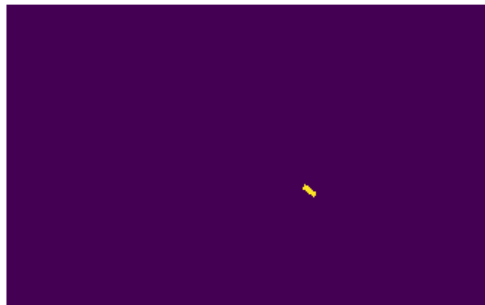
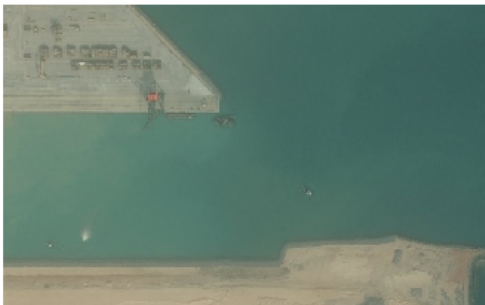
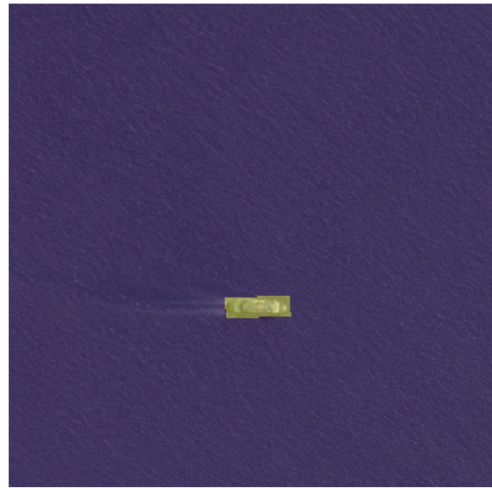
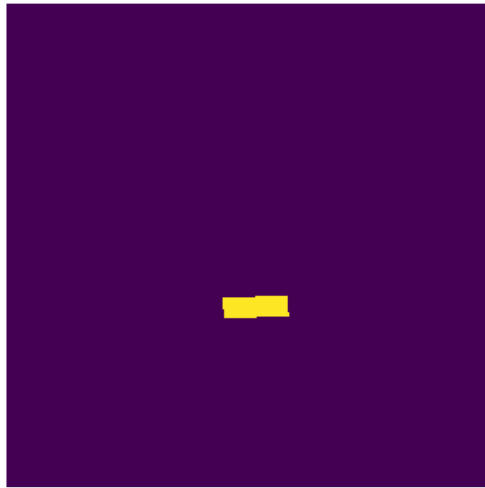
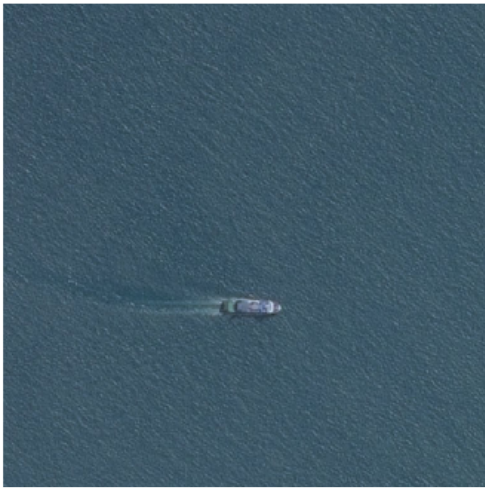
```
# img = imread('./train_v2/000155de5.jpg')
# all_masks = np.zeros((768, 768))
# for mask in img_masks:
#     all_masks += rle_decode(mask)

# fig, axarr = plt.subplots(1, 3, figsize=(15, 40))
# axarr[0].axis('off')
# axarr[1].axis('off')
# axarr[2].axis('off')
# axarr[0].imshow(img)
# axarr[1].imshow(all_masks)
# axarr[2].imshow(img)
# axarr[2].imshow(all_masks, alpha=0.4)
# plt.tight_layout(h_pad=0.1, w_pad=0.1)
# plt.show()
```

In [8]:

```
image_list = image_list[:10]
for imageID in image_list:
    img_masks = masks.loc[masks['ImageId'] == imageID, 'EncodedPixels'].tolist()
    img = imread('./train_v2/'+imageID)
    all_masks = np.zeros((768, 768))
    for mask in img_masks:
        all_masks += rle_decode(mask)

fig, axarr = plt.subplots(1, 3, figsize=(15, 40))
axarr[0].axis('off')
axarr[1].axis('off')
axarr[2].axis('off')
axarr[0].imshow(img)
axarr[1].imshow(all_masks)
axarr[2].imshow(img)
axarr[2].imshow(all_masks, alpha=0.4)
plt.tight_layout(h_pad=0.1, w_pad=0.1)
plt.show()
```



In []: