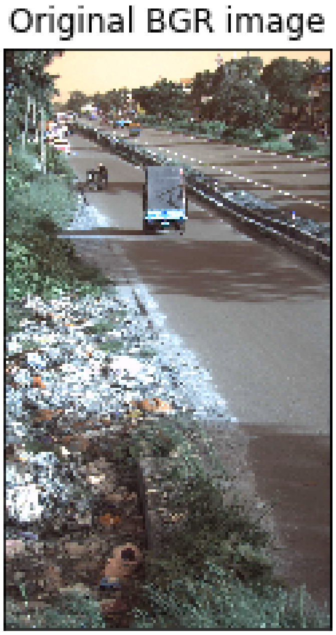
hsv110919

September 12, 2019

[153]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
# Load the image
img = cv2.imread("1.jpg")
print(img.shape)
plt.title('Original BGR image')
plt.imshow(img)
plt.xticks([]), plt.yticks([])
plt.show()
```

```
(1920, 1080, 3)
```



[154]:

```
background= cv2.imread("back.jpg")
print(background.shape)
```

```
plt.title('Background image')
plt.imshow(background)
plt.xticks([]), plt.yticks([])
plt.show()
```

```
(1920, 1080, 3)
```

[155]:

```
#cv2.imshow('original_img',img)
per=50
width= int( (img.shape[1]*per) / 100)
height= int ((img.shape[0]*per) /100 )
dim=(width, height)
img = cv2.resize(img,dim, interpolation = cv2.INTER_AREA)
print(img.shape)
image=img
#cv2.imshow('Resized_img',img)
# Apply log transform
#c= 255/ (np.log(1+np.max(img)))
```

```
(960, 540, 3)
```

[156]:

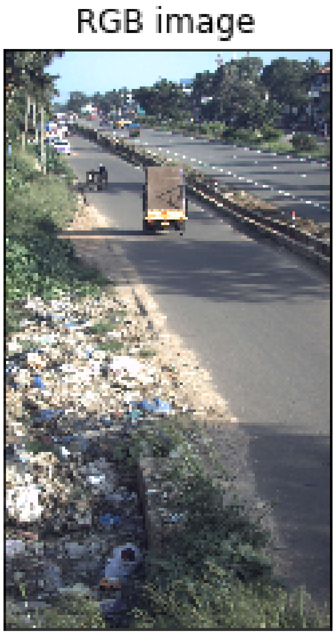
```
#cv2.imshow('original_img',img)
per1=50
width1= int( (background.shape[1]*per) / 100)
height1= int ((background.shape[0]*per) /100 )
```

```
dim1=(width1, height1)
background = cv2.resize(background,dim1, interpolation = cv2.INTER_AREA)
print(background.shape)

#cv2.imshow('Resized_img',img)
# Apply log transform
#c= 255/ (np.log(1+np.max(img)))
```

```
(960, 540, 3)
```

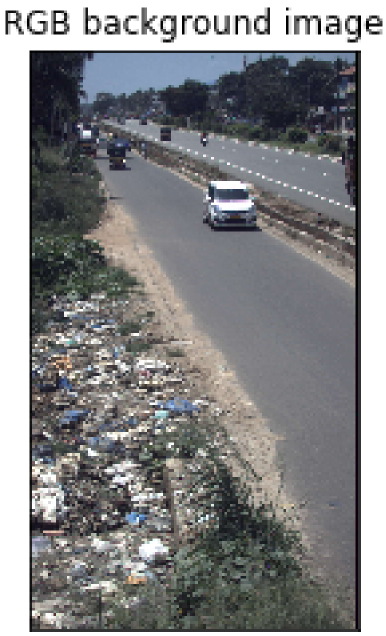
[157]:

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.title('RGB image')
plt.imshow(img, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
image=img
```



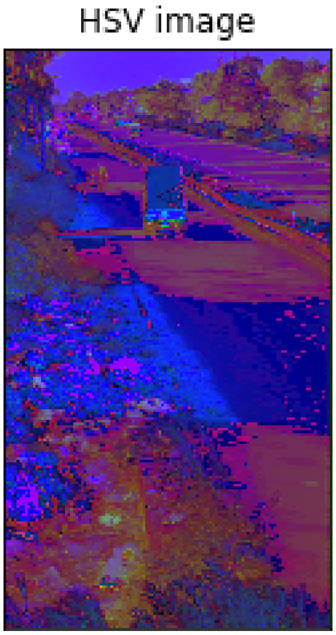
[158]:

```
background = cv2.cvtColor(background, cv2.COLOR_BGR2RGB)
plt.title('RGB background image')
plt.imshow(background, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
```

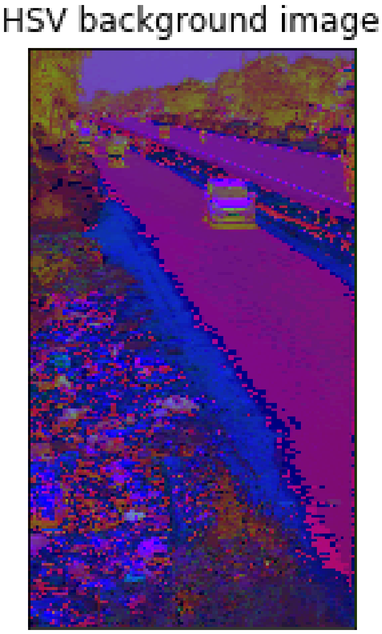


[159]:

```
img = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
plt.title('HSV image')
plt.imshow(img, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
```

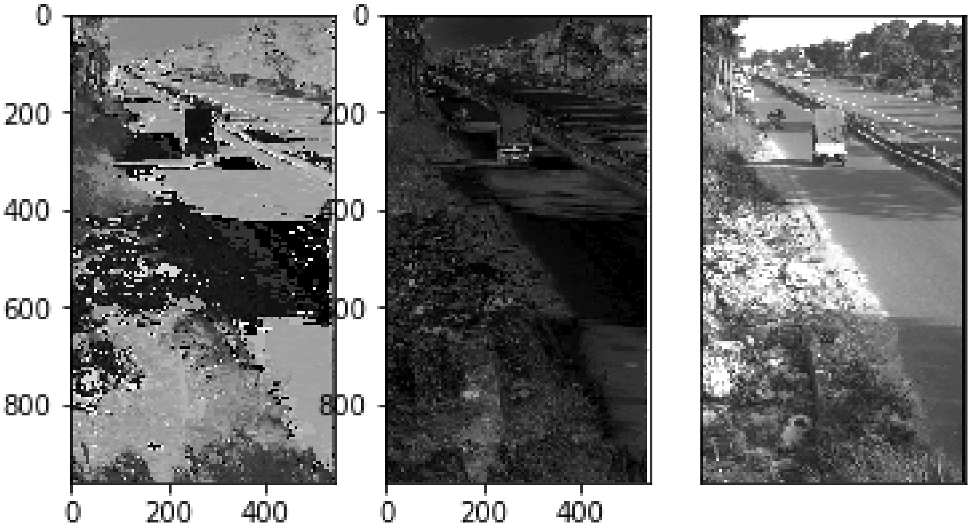


[160]:
```
background= cv2.cvtColor(background, cv2.COLOR_RGB2HSV)
plt.title('HSV background image')
plt.imshow(background, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
```



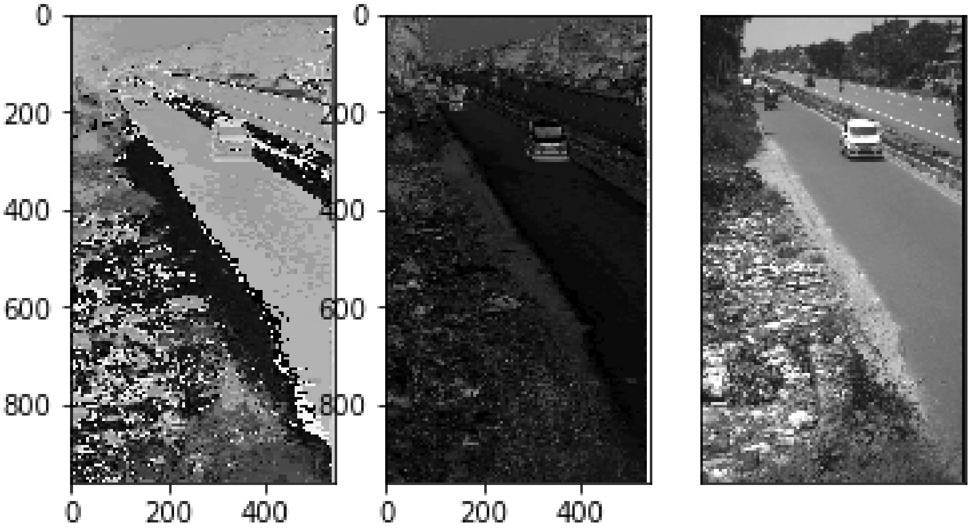
[161]:
```
H, S, V = cv2.split(img)

BH,BS,BV= cv2.split(background)
```

[162]:
```
plt.title('HSV components image')
plt.subplot(131)
plt.imshow(H, cmap='gray')
plt.subplot(132)
plt.imshow(S, cmap='gray')
plt.subplot(133)
plt.imshow(V, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
```



[163]:

```
plt.title('HSV components background image')
plt.subplot(131)
plt.imshow(BH, cmap='gray')
plt.subplot(132)
plt.imshow(BS, cmap='gray')
plt.subplot(133)
plt.imshow(BV, cmap='gray')
plt.xticks([]), plt.yticks([])
plt.show()
```



[164]:

```
alpha1 = 0.2 ; beta1 = 0.8 ; thetas= 1; thetah= 1
s=V
print(s.shape)
```

```
(960, 540)
```

[165]:

```
import sys
V = np.float32(V)/255
BV= np.float32(BV)/255
S = np.float32(S)/255
BS= np.float32(BS)/255
H = np.float32(H)/255
BH= np.float32(BH)/255
s= V.copy()
```

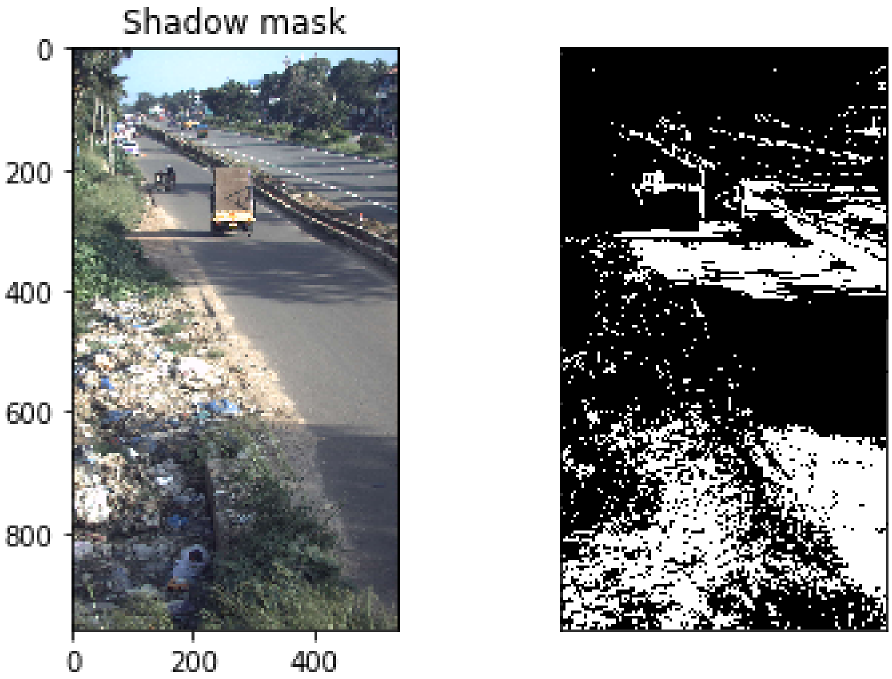
[166]:

```
for i in range(H.shape[0]):
    for j in range(H.shape[1]):
        try:
            x= V[i,j]/ BV[i,j]
        except ZeroDivisionError:
            x = 0

        if ( (alpha1 <= x <= beta1) and (abs(S[i,j]-BS[i,j])<= thetas) and␣
 ↪(abs(H[i,j]-BH[i,j])<=thetah)) :
            s[i,j]=1
        else:
            s[i,j]=0

s=s*255
```

[169]:

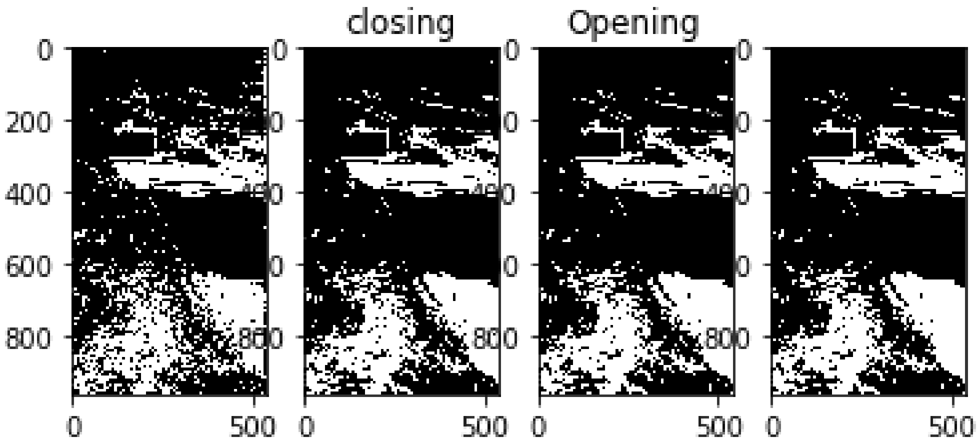
```
print(s.shape)
s = np.array(s,dtype=np.uint8)

plt.subplot(121)
plt.imshow(image, cmap = 'gray')
plt.title('Shadow mask')
plt.subplot(122)
plt.imshow(s, cmap= 'gray')


plt.xticks([]), plt.yticks([])
plt.show()
```

```
(960, 540)
```



[184]:
```
plt.subplot(141)
plt.imshow(s, cmap= 'gray')

median = cv2.medianBlur(s, 5)
plt.subplot(142)
plt.imshow(median, cmap = 'gray')
# Closing - Good for removing noise
kernel = np.ones((3,3), np.uint8)
closing = cv2.morphologyEx(median, cv2.MORPH_CLOSE, kernel)
plt.title('closing')
plt.subplot(143)
plt.imshow(closing, cmap = 'gray')


# Opening - Good for removing noise
opening = cv2.morphologyEx(closing, cv2.MORPH_OPEN, kernel)
plt.title('Opening')
plt.subplot(144)
plt.imshow(opening, cmap = 'gray')
```

[184]:
```
<matplotlib.image.AxesImage at 0x7f7a500df9e8>
```

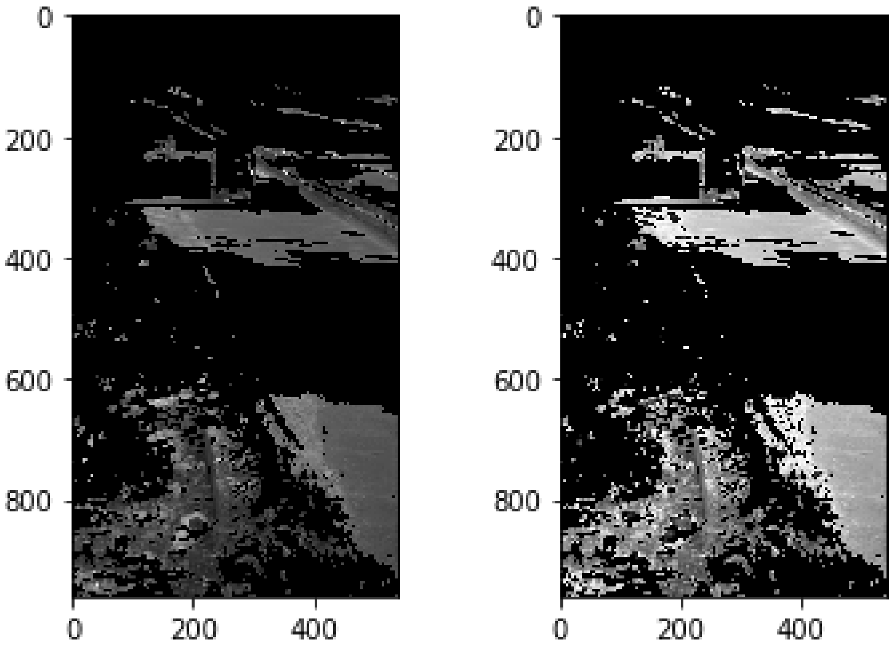


[205]:

```
#V=V*255
#V = np.array(V,dtype=np.uint8)
roi = cv2.bitwise_and(V,V, mask=opening)
plt.title('ROI')
plt.subplot(121)
plt.imshow(roi, cmap = 'gray')
roi=roi*2
plt.subplot(122)
plt.imshow(roi, cmap = 'gray')
```

[205]:

```
<matplotlib.image.AxesImage at 0x7f7a49a3aa90>
```



[ ]:

```
```