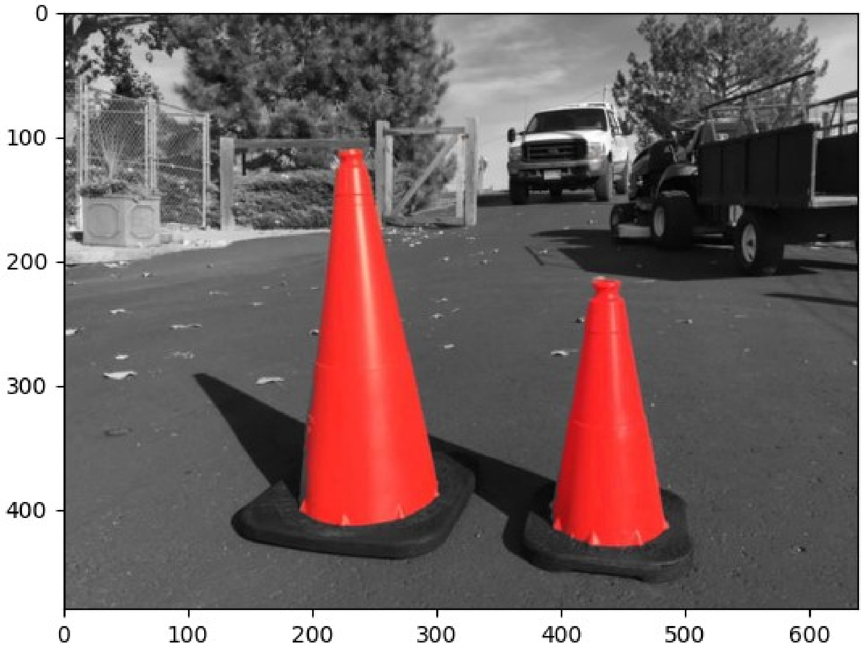
HOMEWORK SET 3 – Víctor Mira Ramírez

Exercise 1

In the solving of this exercise I firstly tried rgb filtiering, but it wasn't very good as it was hard not to get the orange from the truck behind. I used the histograms to make this filters (They are commented on the code itself).

Then I tried HSV filtering, which made the work easier. However I used three masks as red is divided into two in the Hue scale, the lowest and the highest. Moreover, some highlights of the cones were difficult to isolate so I made a mask for them. I inverted the mask and passed it through a grayscale img projected into a vector of 3 components of the same gray value ((12,12,12) (123,123,123) instead of (12) (123) etc.).

The final filters (in HSV) are:
(0,70,150), (8,255,255) + (9,60,240), (12,255,255) + (175,30,140), (180,255,255))



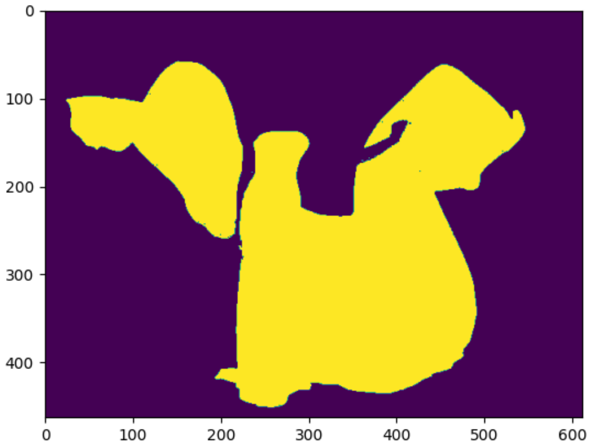
Code:

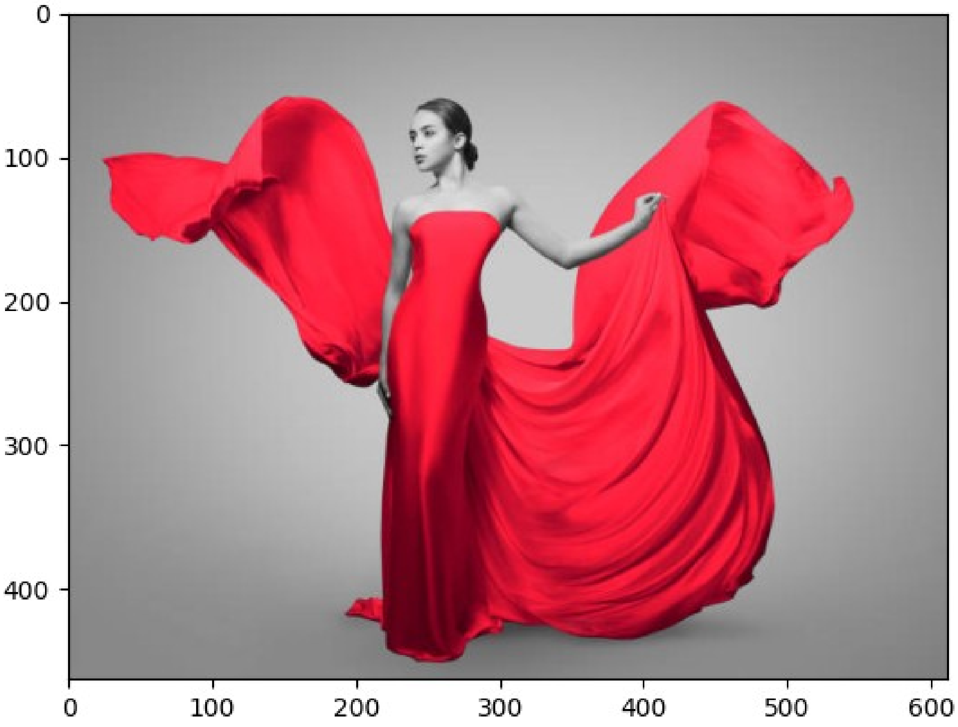
```
##############

# EXERCISE 1 #
##############

"""
How do you separate the traffic cone from the background?
The image “traffic_cone.jpg” can be download in the attachment.
Use the “Color Threshold” apps include in the MATLAB,
choose the appropriate color space, show all the threshold conditions
and your result.
"""
path=r'/home/victor/fisicaua/tercero/SIUE/robotic_vision/entregas/hw3/traffic_cone.jpg'

img = cv2.imread(path, cv2.IMREAD_COLOR)
RGB_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # translate to rgb

gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # gray image
backtorgb = cv2.cvtColor(gray_img,cv2.COLOR_GRAY2RGB) # gray image with vectorsize = 3

#______________________________________________________________

# # HISTOGRAMS

# plt.figure()
# plt.hist(RGB_img.ravel(),256,[0,256]) #grayscale

# plt.figure()
# plt.title('blue histogram')
# plt.plot(cv2.calcHist(RGB_img, [0], None, [256], [0,256])) #blue

# plt.figure()
# plt.title('green histogram')
# plt.plot(cv2.calcHist(RGB_img, [1], None, [256], [0,256])) #green

# plt.figure()
# plt.title('red histogram')
# plt.plot(cv2.calcHist(RGB_img, [2], None, [256], [0,256])) #red
```

```
#____________________________________________________________

# MASKING
HSV_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV) # translate to hsv

# Trying RGB masking
# lower_red = np.array([170, 15, 15])
# higher_red = np.array([255, 150, 170])

mask1 = cv2.inRange(HSV_img, (0,70,150), (8,255,255))
mask2 = cv2.inRange(HSV_img, (9,60,240), (12,255,255))
mask3 = cv2.inRange(HSV_img, (175,30,140), (180,255,255))

mask = mask1 + mask2 + mask3
inv_mask = cv2.bitwise_not(mask) # inverse mask to use in grayscale img

# Calculates the masked rgb image
masked_img = cv2.bitwise_and(HSV_img, HSV_img, mask = mask)
rgb_masked_img = cv2.cvtColor(masked_img,cv2.COLOR_HSV2RGB)
# Calculates the masked grayscale image
masked_gry = cv2.bitwise_and(backtorgb, backtorgb, mask = inv_mask)

# Adds the two masked images
result = cv2.addWeighted(rgb_masked_img, 1, masked_gry, 1, 0)

plt.figure()
plt.imshow(result)
```

Exercise 2

In this exercise I made a function that made the stipulated behaviour and finetuned the filter to get exactly what was asked without any extra or lacking pixel. The filter used was (168,120,0), (180,255,255). The result photos are:





Code:

```
##############
# EXERCISE 2 #
##############

"""
At the end of the lecture slides, we show an example of how to use HSV filter
to highlight the girls red dress. The link in problem 1 also provide an
example of very powerful MATLAB tool – createMask. Write your own creteMask
function that can separate the woman in the red dress
(shown below, ‘red_dress.jpg’ file can be download from the attachment)
from the background using HSV filter, include all the code and plot the
original image, mask and filtered image.
"""

def createMask(img):
HSV_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV) # translate to hsv

gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # gray image
backtorgb = cv2.cvtColor(gray_img,cv2.COLOR_GRAY2RGB) # gray image with vectorsize = 3

mask = cv2.inRange(HSV_img, (168,120,0), (180,255,255))

inv_mask = cv2.bitwise_not(mask) # inverse mask to use in grayscale img

masked_img = cv2.bitwise_and(HSV_img, HSV_img, mask = mask)
rgb_masked_img = cv2.cvtColor(masked_img,cv2.COLOR_HSV2RGB)
# Calculates the masked grayscale image
masked_gry = cv2.bitwise_and(backtorgb, backtorgb, mask = inv_mask)

# Adds the two masked images
result = cv2.addWeighted(rgb_masked_img, 1, masked_gry, 1, 0)

plt.figure()
plt.imshow(result)

plt.figure()
plt.imshow(mask)
```

```
path=r'/home/victor/fisicaua/tercero/SIUE/robotic_vision/entregas/hw3/red_dress.jpg'
createMask(cv2.imread(path, cv2.IMREAD_COLOR))

plt.figure()
plt.imshow(cv2.cvtColor(cv2.imread(path, cv2.IMREAD_COLOR),cv2.COLOR_BGR2RGB))

plt.show()
```