HSV color conversion

April 26, 2018

1 HSV colorspace

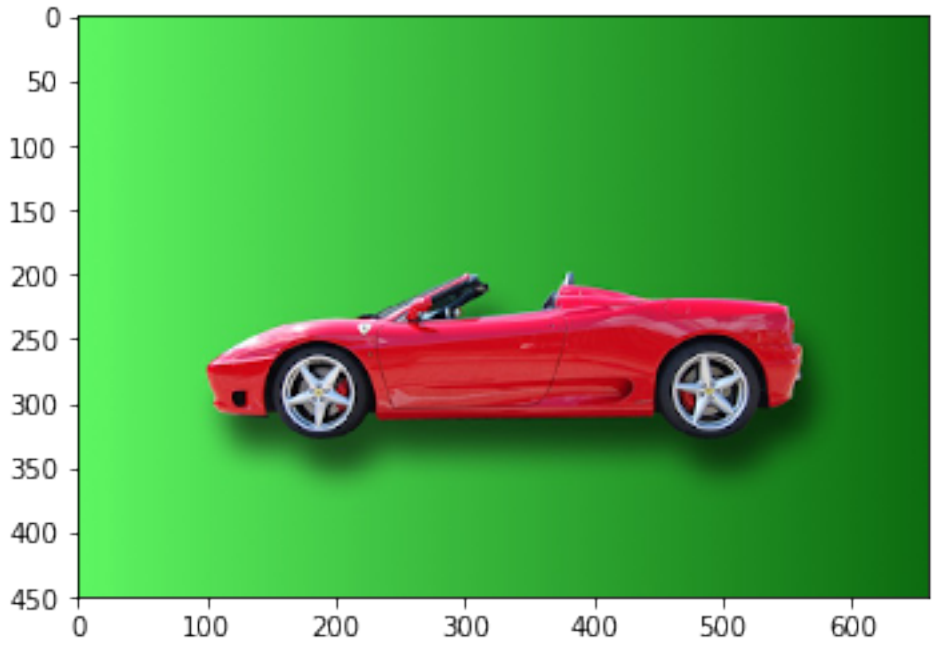
1.0.1 Import resources

In [1]:
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

import numpy as np
import cv2

%matplotlib inline
```

1.0.2 Read in RGB image

In [2]:
```
# Read in the image
image = mpimg.imread('images/car_green_screen2.jpg')

plt.imshow(image)
```

Out[2]:
```
<matplotlib.image.AxesImage at 0x7f75843cb2e8>
```

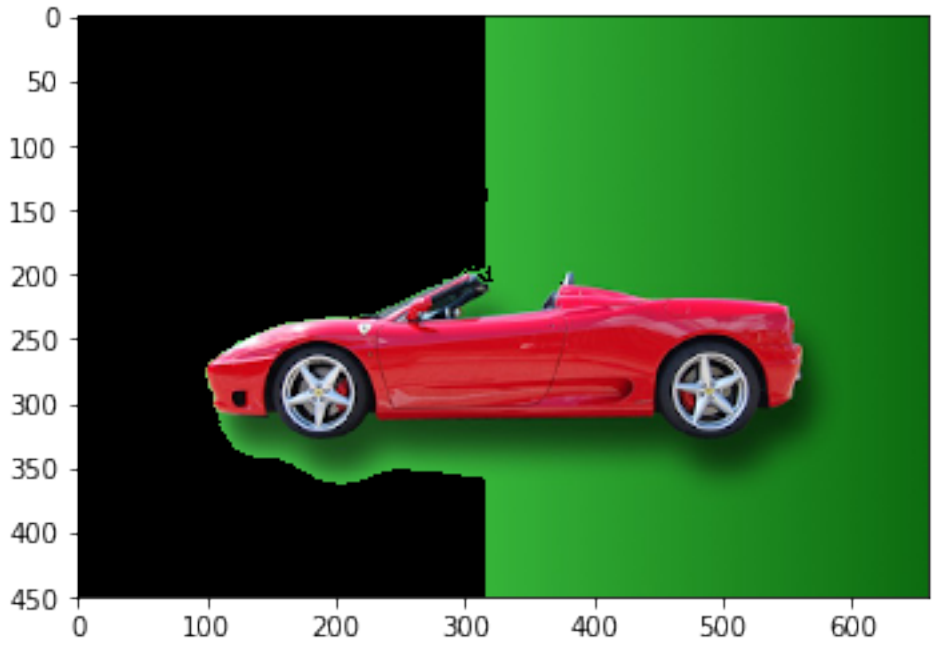


1.0.3 RGB threshold

Visualize the green threshold you defined in the previous, consistent green color case.

In [3]:
```
# Define our color selection boundaries in RGB values
lower_green = np.array([0,180,0])
upper_green = np.array([100,255,100])

# Define the masked area
mask = cv2.inRange(image, lower_green, upper_green)

# Mask the image to let the car show through
masked_image = np.copy(image)

masked_image[mask != 0] = [0, 0, 0]

# Display it!
plt.imshow(masked_image)
```

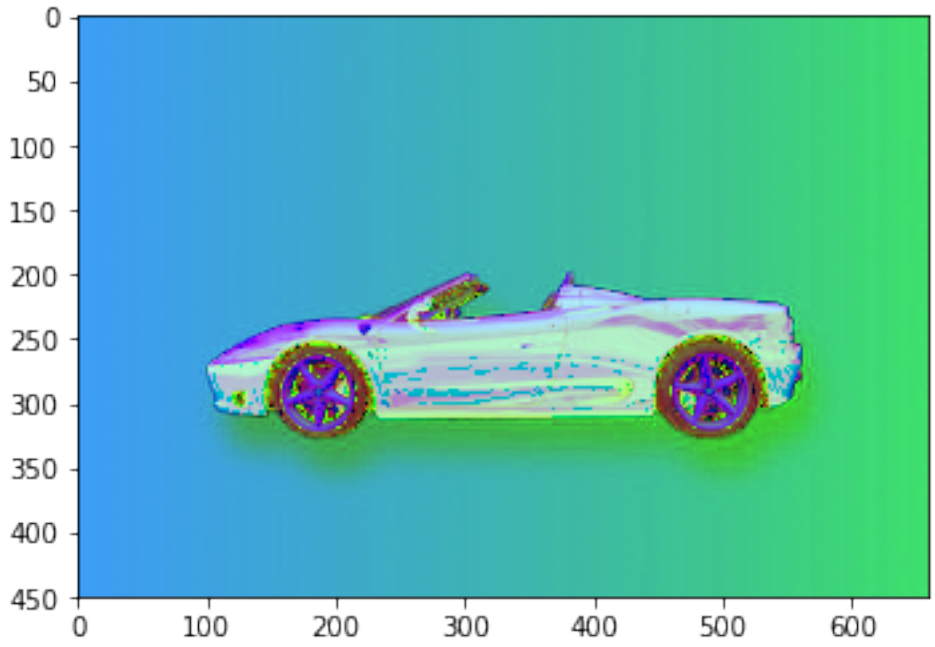
Out[3]:
```
<matplotlib.image.AxesImage at 0x7f75842f32e8>
```

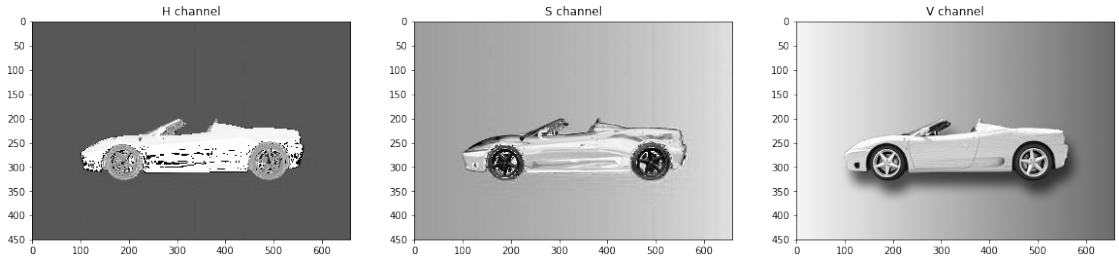


1.0.4 Convert to HSV

In [20]:
```
# Convert to HSV
hsv = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
plt.imshow(hsv)

# HSV channels
h = hsv[:,:,0]
s = hsv[:,:,1]
v = hsv[:,:,2]

# Visualize the individual color channels
f, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(20,10))
ax1.set_title('H channel')
ax1.imshow(h, cmap='gray')
ax2.set_title('S channel')
ax2.imshow(s, cmap='gray')
ax3.set_title('V channel')
ax3.imshow(v, cmap='gray')
```

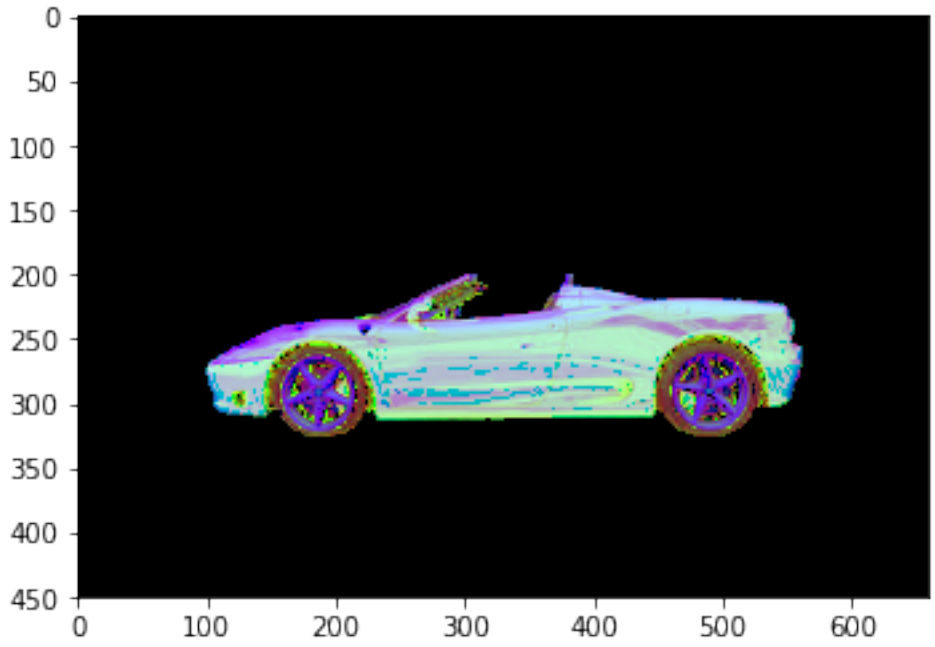
Out[20]:
```
<matplotlib.image.AxesImage at 0x7f7583ff7710>
```





1.0.5 TODO: Mask the green area using HSV color space

```
In [25]: ## TODO: Define the color selection boundaries in HSV values
         lower_color_value, higher_color_value = np.array([50, 0, 0]), np.array([75, 255, 255])
         mask = cv2.inRange(hsv, lower_color_value, higher_color_value)
         ## TODO: Define the masked area and mask the image
         # Don't forget to make a copy of the original image to manipulate
         masked_image = np.copy(hsv)
         masked_image[mask != 0] = [0, 0, 0]
         plt.imshow(masked_image)

Out[25]: <matplotlib.image.AxesImage at 0x7f7582513278>
```


0
50
100
150
200
250
300
350
400
450
0
100
200
300
400
500
600