Práctico Visión: Hoja de Trucos.

Mensaje tipo Image:

- sensor_msgs/Image
  - std_msgs/Header header
  - uint32 height
  - uint32 width
  - string encoding
  - uint8[] data

Nodos Necesarios

```
roslaunch usb_cam usb_cam-test.launch
ó rosrun usb_cam usb_cam_node
rosrun rqt_image_view rqt_image_view  #visualización
```

Comandos Útiles

Transformación entre ROS msg e imagen opencv

```
bridge = CvBridge()   #objeto tipo CvBridge necesario para la conversión
image = bridge.imgmsg_to_cv2(msg, "bgr8") #conversión de msg a imagen cv2
msg =   bridge.cv2_to_imgmsg(image, "bgr8") ##conversión de imagen cv2 tipo msg
```

Filtrar píxeles entre 2 límites

```
mask = cv2.inRange(image, lower_limit, upper_limit)
```

Cambiar entre espacios de colores

```
image_out = cv2.cvtColor(image, color_space)
```

```
color_space :  cv2.COLOR_BGR2HSV ; cv2.CV_BGR2GRAY
```

Aplicar Máscara sobre Imagen

```
image_out = cv2.bitwise_and(image, image, mask= mask)
```

Transformación morfológicas

```
kernel = np.ones((5,5),np.uint8)   # Matriz de 1's de 5x5 ocupada en las transfo
img_out = cv2.erode(img, kernel, iterations = 1)
img_out = cv2.dilate(img, kernel, iterations = 1)
```

Buscar contornos de blobs

```
contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

Obtener y Dibujar rectángulo

```
x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(img, (x1,y1), (x2,y2), (0,0,0), 2)
```