Crash Course Paparazzi 2016
Computer vision

lesson 4. . . Making your drone see something

Roland Meertens, Christophe de Wagter, Guido de Croon

Februari 2016

Introduction

Welcome to the most important part of this course: understanding how vision works. When using image processing right your drone can perform the most awesome tricks. However: when you do it wrong your drone will crash.

Goals of this exercise

- Finding the camera data
- The format of the image
- Your first project

Finding the camera data

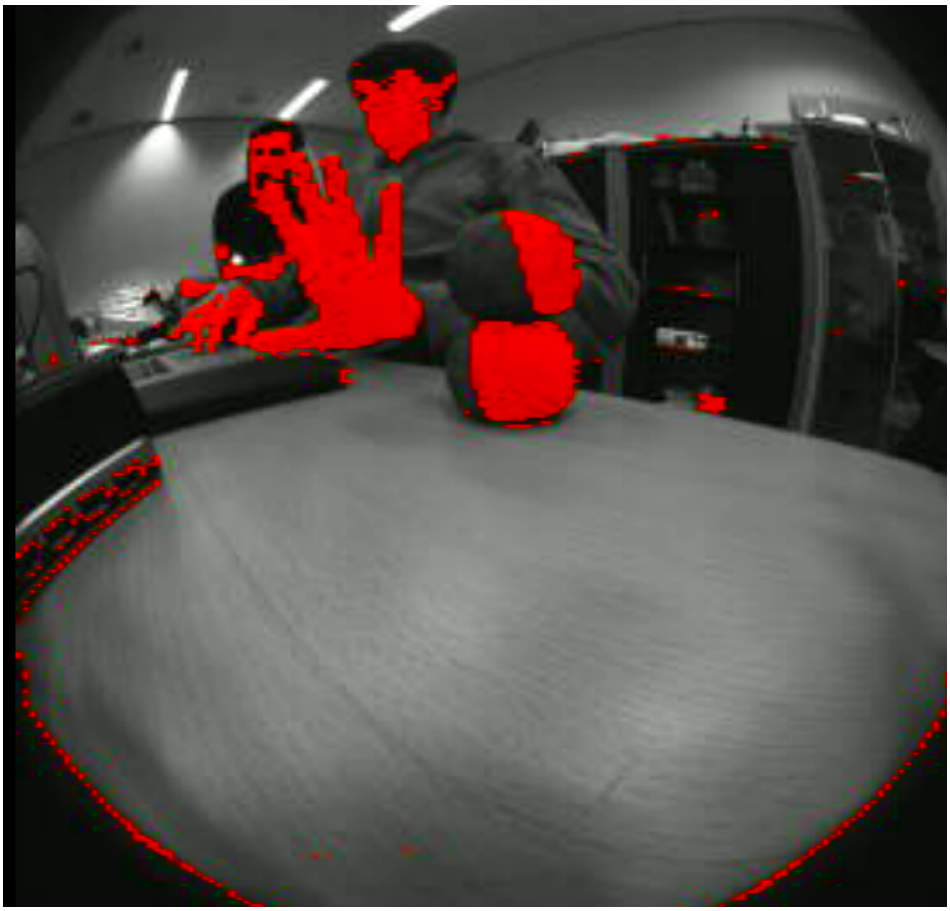
To get data out of the camera in paparazzi you have to include the video_thread.xml module in your airframe file, and define VIDEO_THREAD_CAMERA to front_camera. After you added this you can create a new module that does something with the data of the front camera. A nice example for such a module is the colorfilter module. If you want to see what the image looks like after it comes out of the camera or the color filter you have to can also include the video_rtp_stream module. The modules part of your airframe file might now look like this:

```
<modules main_freq="512">
  <module name="geo_mag" />
  <module name="air_data" />
  <module name="send_imu_mag_current" />
  <module name="logger_file">
    <define name="FILE_LOGGER_PATH" value="/data/ftp/internal_000" />
  </module>
  <!-- Video/Camera modules -->
  <module name="video_thread" />
  <module name="cv_ae_awb">
    <define name="CV_AE_AWB_AV" value="1" />
    <define name="CV_AE_AWB_VERBOSE" value="0" />
  </module>
  <module name="cv_colorfilter">
    <define name="COLORFILTER_CAMERA" value="front_camera" />
  </module>
```

```
  <module name="orange_avoider" />
  <module name="video_rtp_stream">
    <configure name="VIEWVIDEO_HOST" value="192.168.42.65" />
    <configure name="VIEWVIDEO_BROADCAST" value="FALSE" />
    <define name="VIEWVIDEO_CAMERA" value="front_camera" />
    <define name="VIEWVIDEO_SHOT_PATH" value="/data/ftp/internal_000" />
    <define name="VIEWVIDEO_FPS" value="18" />
    <define name="VIEWVIDEO_WRITE_VIDEO" value="FALSE" />
    <define name="VIEWVIDEO_VIDEO_FILE" value="orangeAvoider" />
    <define name="VIEWVIDEO_STREAM_VIDEO" value="TRUE" />
  </module>
</modules>
```

Upload this to your bebop. To view the video you open the .sdp (stream description protocol) file from the bebop in vlc: vlc ftp://192.168.42.1/internal_000/stream.sdp or use the Bebop Video Stream tool from within paparazzi.
If you open your ground control station and go to settings–ColorFilter you can change what is filtered by the colorfilter. Look at the YUV colorspace on Wikipedia (https://en.wikipedia.org/wiki/YUV) and try to find settings that filter the color orange. If you open the RTP stream you can see what pixels are selected, they become red.
In the lab we tried filtering the color red, as found on our hands and little red and green balls. We used the following values, and obtained the following result:

```
y_min = 4
y_max = 91
u_min = 0
u_max = 124
v_min = 127
v_max = 255
```



Note that streaming images takes a large amount of bandwith. This could cause your joystick of optitrack position commands to be delayed (causing instability) or not even arrive.

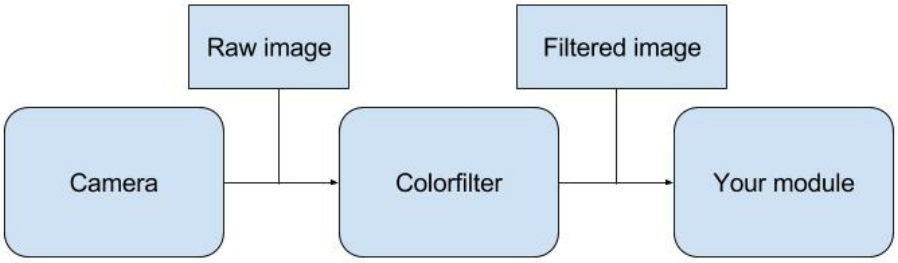
Take some time to check what the colorfilter does in the file colorfilter.c.
In the colorfilter_init function the function cv_add_to_device is called with as argument the colorfilter_func function and the front camera:

```
void colorfilter_init(void)
{
  listener = cv_add_to_device(&COLORFILTER_CAMERA, colorfilter_func);
}
```

Now each time the video thread gets an image from the camera it calls the colorfilter_func function with the new image as an argument:

```
bool_t colorfilter_func(struct image_t* img)
{
  // Filter
  color_count = image_yuv422_colorfilt(img,img,
      color_lum_min,color_lum_max,
      color_cb_min,color_cb_max,
      color_cr_min,color_cr_max
      );
```

```
  return FALSE;
}
```



Note that the order in which you call cv_add is important. The first function to get the image might edit the image. If you add modules that do something with the image, the first module you added in your airframe file will get the image first. The reason that the image might be changed is because the functions is passed a pointer to the image. The argument the function gets is a pointer to an image it is possible for a module to change the actual image. If you take a look at the colorfilter function you see this module does this. Colorfilter sets UV to zero (makes the nonfiltered image grayscale) and enhances the color at the filtered colors. A second module added after the colorfilter will see a gray image with some enhanced colours.

The format of the image

The image you receive is yuv422, or UYVY. Per two pixels four bytes of data are recorded. The U and V together define the color and both pixels get the same color (but different intensity). Image.c has several functions that can be used with this format. Please take a look at this file and try to understand what these functions are doing, how you can use them, and what other functions would be useful.

Your first project

Now it is time to choose an exercise to learn how to program in Paparazzi using your flightplan skills, code skills and computer vision skills. The two options are:

Choice 1: Read a sign

Fly to a position in the center of the arena. If you see a blue sign fly to a waypoint to the left, hover here for five seconds and go back to the center. If you see a red sign perform the same behaviour but fly to the right.

Choice 2: Follow me!

Create a real "selfie-drone" in which the drone follows an orange blob at a certain distance. Start by hovering at a waypoint and determining where the orange blob is (more to the left or more to the right?) and adjust your heading based on this. To start following you, you can determine the size of the blob. Is it too big? Fall back! Otherwise: fly forwards by slowly moving your waypoint.