Overview of image enhancement techniques for UAV image velocimetry

Abstract

This report is a deliverable of the **COST Action CA16219** - "Harmonization of UAS techniques for agricultural and natural ecosystems monitoring" (**www.costharmonious.eu**), and describes image enhancement workflow for image velocimetry (removal of visual noise, accentuation of tracer particles), image format handling, the use of different image colorspaces, etc.

The aim of the investigation is to explore which image enhancement methods are most viable for application in UAV image velocimetry workflow, especially when natural or artificial tracer particles are used. Throughout a series of Jupyter notebooks, most important aspects of image quality, formats, colorspace models, filtering methods, and the combinations of the mentioned, will be explored and described.

The code in the notebooks can be used by anyone to improve the accuracy of their image velocimetry. Finally, a "no-code" workflow will be described with the use of the SSIMS tool.

Contents

Disclaimer: The code presented in this and connected notebooks is free to use, adapt, and distribute. The author of the notebooks offers no guarantees that the code will work correctly on every device and hold no liability for any damages that might occur due to its use.

1 Examples

Code in this report is written in Python programming language and prepared for presentation in a series of Jupyter notebooks, although static PDF versions are also available in the `./PDFs` folder. Some of the figures in the notebooks are interactive, and can be manipulated using the icons below or next to them (such as zoom and pan). Image handling will be done using OpenCV library which is by far the most popular C++/Python library for programmatic interaction with image data, including image processing and computer vision.

Libraries necessary for the presentation in this notebook can be imported as:

In [1]:
```
# Necessary libraries
import matplotlib.pyplot as plt
import cv2

# Use [%matplotlib widget] inside JupyterLab,
# and [%matplotlib notebook] for Jupyter Notebook
%matplotlib widget
```

The report will explore various image enhancement procedures using five images taken during various field campaigns. These images (individual video frames) are located in the `./1080p` and `./4K` folders in two different resolutions (1920-by-1080 px and 3840-by-2160 px). The notebooks use 1080p images,

but you can change the folders to `./4K` if you wish to obtain more detailed view. Images used can be examined using the following code:

In [2]:
```
fig, ax = plt.subplots(nrows=3, ncols=2, figsize=(9.8, 9))

for i in range(5):
    # Using 1080p images for higher loading speed
    img_path = './1080p/{}.jpg'.format(i+1)

    # OpenCV library defaults to Blue-Green-Red (BGR) channel order,
    # which is the order of information in storage/memory (0x00bbggrr)
    img_bgr = cv2.imread(img_path)
    # However, Matplotlib library defaults to a more common
    # Red-Green-Blue (RGB) channel order
    img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

    ax[i//2][i%2].imshow(img_rgb)
    ax[i//2][i%2].set_title('Image {}'.format(i+1))


ax[2][1].set_visible(False)
[a.axis('off') for a in ax.reshape(-1)]


plt.tight_layout()
plt.show()
```

Figure

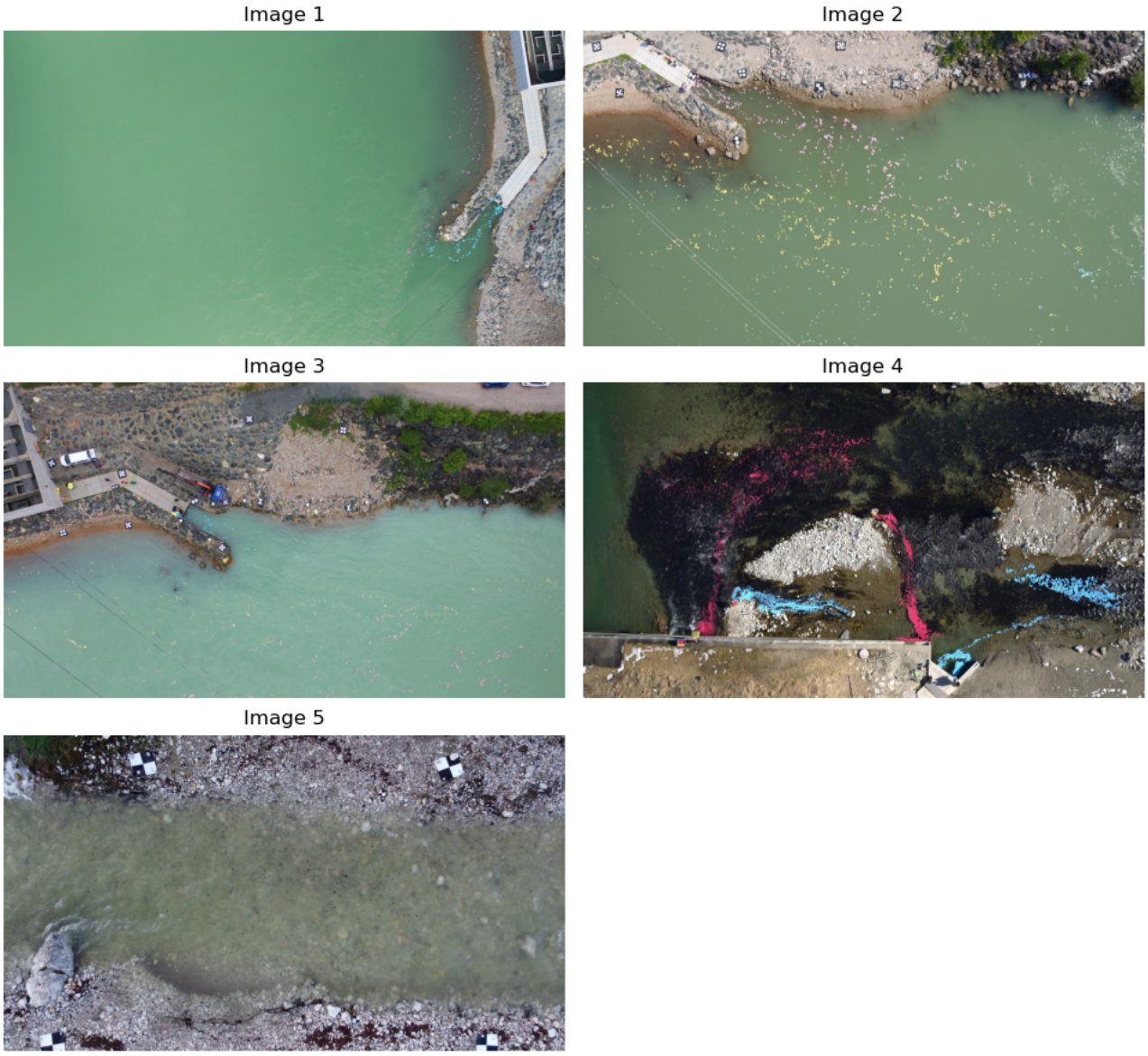

Image 1 presents a case with pink- and cyan-colored artificial tracer particles at the right-hand image side and close to the riverbank. There are some minor light reflections and waves on the water surface as well as some barely visible riverbed features close to the riverbank.

Image 2 presents a case with pink-, yellow-, cyan-, and white-colored tracer particles scattered across a uniform water surface with no light reflections and/or surface waves.

Image 3 is similar to Image 2, except with surface waves that disturb the perception of tracer particles. This image was also taken with a UAV at a higher flight altitude.

Images 1 through 3 are taken at different times on the same site.

Image 4 presents a drastically more challenging case of shallow, clear water, with water surface perturbations caused by shallow flow, and islands of deposited sediment (gravel) in the river itself. Magenta- and cyan-colored artificial tracer particles were used with high seeding density.

Image 5 presents another challenging case with significant surface perturbations and light reflections. This is also a case of a shallow, supercritical flow with relatively clear water where riverbed features are

quite prominent. Dark-colored dried leaves were used as low-density seeding (tracer) particles.

2 Image velocimetry workflow

In order to understand the role of the image enhancement for image velocimetry, one should have a solid understanding of the entire workflow from video/image acquisition of the water surface flow to the estimation of surface velocities. The general approach is often involves the following operations, performed in given order:

1. Video/image acquisition,
2. Removal of geometric distortions of the camera,
3. Stabilization of the video,
4. Orthorectification,
5. **Image enhancement**,
6. Velocimetry analyses.

For explanation about steps 1 through 4, the reader can consult the HARMONIOUS video series on the **project's YouTube channel** or the relevant publications listed on the **project's website**. The guiding idea of those steps is to accurately provide a stable frame of reference for the entire video, and to estimate the camera position relative to the real-world with minimal image distortions.

While in some cases velocimetry analyses can begin immediately after step 4, certain surface flow and ground conditions may prove to be unfavorable for a direct application of image velocimetry algorithms - there exist some reflections of light off the water surface, tracer particles are not well-defined relative to the water surface, riverbed is visible and interferes with tracer particle detection, etc. Such conditions call for an application of **image enhancement** algorithms, which allow for a targeted accentuation of tracer particles (features), with the goal of improving the accuracy (i.e., reducing the measurement uncertainty) of velocity estimation.

Alternatively, one could advocate for **image enhancement** to precede steps 2-4 in order to facilitate the remainder of the workflow. However, as it will be shown later on, many of the image enhancement algorithms accentuate certain image features at the expense of deteriorating other (for velocimetry less important) features. Many of such non-tracer features can be used in various steps of the video stabilization and orthorectification, and their deterioration could reduce the reliability and accuracy of said procedures. Additionally, some stabilization and orthorectification methods use certain image enhancement algorithms in their respective implementations, so premature image enhancement can even be detrimental to their overall goals.

Continue to next chapter: Image formats >>