ENV-408 Ex01: Image Measurements, Coordinates and Distorsion Model

Objectives

> 1. Learn how to perform manual image feature measurements on an image and become familiar with common image coordinate systems.

> 2. Understand what an image distorsion model is and what impact it has on the coordinates of measured distored/undistored image features.

Methodology:

- Given an image with indicated objects (see the close-up screenshots `Manual_GCP_1*.jpg`), perform *manual measurements of their image coordinates* by clicking on the central pixel of objects viewed in the image using a professional photogrammetric software.

> **Note**: These coordinates will be used in following exercises so try to be as accurate as possible when clicking points.

- Convert the obtained coordinates from the *top-left* image coordinate system to the *perspective-centered* image coordinates.
- Implement a *Contrady-Brown* distorsion model (with a given set of parameters) and use it to undistort the N-image measurements (via a numerical *solver*).
- *Visualize the impact of the distorsion model* on the image level by undistorting a grid of pixels covering the entire image. *Test different values of distortion coefficients* to understand their impact on the image coordinates.

Input data

Data for the exercise is provided in the following folder on moodle (click "**Download folder**"):

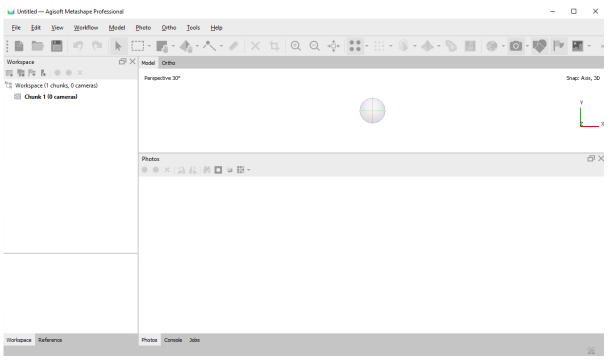
```
main
|   Lab01_Exercise2.py
|   Lab01_Description.pdf
|
└─── raw_data
|    |   1092311568.jpg
|    |   1092311568_marked.jpg
|    |   1092311568_assessement.jpg
|    |
|    └─── GCPs_1092311568
|         |   GCPs_100_1092311568.jpg
|         |   ...
|
└─── measurements
```

Part 1 : Introduction to Agisoft, manual measurement on images

Agisoft Metashape is a professional photogrammetric software. You can run it on EPFL computers located in the exercise room (GR-building) or remotely through the EPFL VDI machines accessible here with your gaspar credentials.

> Note: To run the VDI in a browser choose `VMware Horizon HTML Access`, alternatively download and install the VMware Horizon Client and connect to `https://vdi.epfl.ch/portal/webclient/index.html`

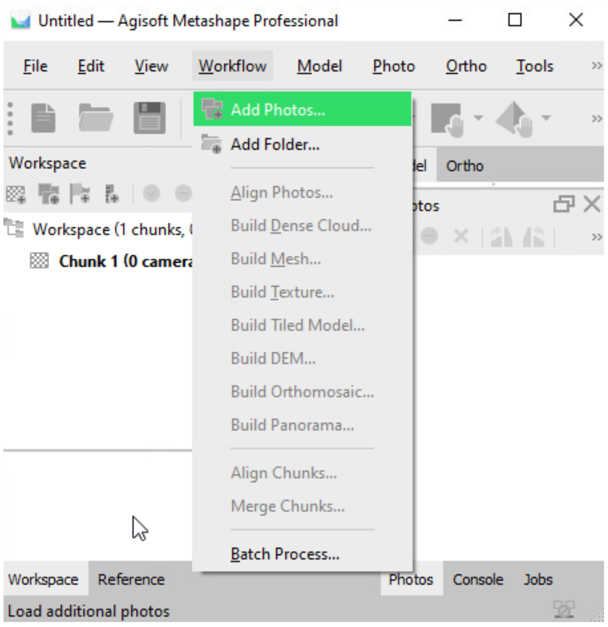
Access a vdi machine on the ENAC-SSIE-WIN pool and launch Agisoft from the Applications folder on your Desktop `Applications\APP-SSIE-TOPO\Agisoft Metashape Professional`. Agisoft main page will open


Untitled — Agisoft Metashape Professional
File Edit View Workflow Model Photo Ortho Tools Help
Workspace
Workspace (1 chunks, 0 cameras)
Chunk 1 (0 cameras)
Model Ortho
Perspective 30°
Snap: Axis, 3D
Photos
Workspace Reference
Photos Console Jobs

1.1 Load images :

> Note: For executing this step you may want to first copy the image `raw_data/1092311568_marked.jpg` to a personal network folder, to which the VDI machine has access.

To import images into your project, go to the `Workflow` tab, `Import photos`.



Import the photo `raw_data/1092311568_marked.jpg` into your project.
Once imported, you should see one image in your so called "chunk". The image can be

displayed by double clinking its name or thumbnail.

1.2 Create markers :

You can see that 12 control points have been approximately marked on the image.
We need to extract accurately the coordinates (in pixels) of those control points. For each of the 12 points, in the `raw_data/GCPs_1092311568/GCP_"X"_1092311568.jpg`, you will find close range screeshots over each GCP to help you accurately pin the point on the image.

Accurately create a pin a marker for each control point in the image. To do so :

1. Right click on the selected pixel and select `Add marker`.
2. Refine the position of the marker if needed by holding left mouse button.
3. Rename the marker with the corresponding control point id : Under `Chunk -> Marker` , right click on your marker and rename it.

1.3 Export markers

To export the image coordinates of the markers, go to `File -> Export -> Export Markers` and save your markers as an Agisoft xml file.

Save these measurements to the `measurements` folder, they will be needed in the next exercise section.

Part 2 : Image coordinates, distortion model

Objectives

> 1. Learn how to correct lens distorsions on image measurements..

> 2. Complete a python code skeleton provided: *`Lab01_Exercise2.py`* where the necessary functions and function declarations are defined.

2.1 : Coordinate projection

Read camera IO coefficients from 'cam_param.txt' file and assign the following variables:

- Camera height: h
- Width [px]: w,
- Focal length [px]:c
- Principal point offset [px]: cx,cy
- Radial distorision coefficients: k1,k2
- Tangential distortion coefficients: p1,p2

2.1.1 Load Agisoft manual measurements

Use the provided `parse_xml` function to load and parse your data in a np.array with ids and a second np.array with uv coordinates (see code provided).

2.1.2 Definition of conversion between image coordinate systems

Up to now, measurements were expressed in "Top-Left" image coordinates, expressed in pixels:

- The origin is the top-left corner of the image
- U is oriented horizontally →
- V is oriented downward ↓

In the next part of this lab, it is necessary to express measurements in perspective centered coordinates, expressed with unitless coordinates :

- The origin is the perspective center of the image. Simply put, this is the center of the pixel array shifted along the x and y axis by the principal point offset in pixels c_x, c_y
- X is oriented horizontally →
- Y is oriented downward ↓

Given :

- (u, v) : coordinates in top-left image coordinate [px]
- (x, y) : coordinates in perspective-centered coordinates [-]
- c_x, c_y : perspective center offset [px]
- w, h : image width, height [px]
- c : focal length [px]

$$x = \frac{u - (\frac{w-1}{2} + c_x)}{c} \tag{1}$$

$$y = \frac{v - (\frac{h-1}{2} + c_y)}{c} \tag{2}$$

Note:

- to convert to perspective sensor coordinates, one aspect of the camera intrinsic orientation is considered : the center offset.
- This offset is generally due to the imperfect mounting of the lens with respect to the sensor, causing the center of the lens to be slightly shifted with respect to the center of the sensor.

(a) Define function `topleft2perspective` to project pixel coordinates from sensor topleft to perspective centered frame using equations 1 and 2 above.

(b) Define function `perspective2topleft` to project pixel coordinates from perspective to sensor top-left frame by inverting equations 1 and 2 above.

2.1.3 Project and save measurements

- Use the functions above to convert uv to perspective centered coordinate (keep the np.array dim. similar to uv)
- Save your measurements along with ids using provided code. Appart from the 'id_xy.txt' and 'id_uv.txt' measurements file.

2.2 : Undistort manual measurements

> **Objectives:** Cameras are not perfect and image distortion due to imperfect mounting and lens distortion must be corrected to accurately map each pixel.

In this section you will implement a simple distortion model and visualize its impact on your measurements.

2.2.1 Set up

- First, load the raw measurements you generated in Part 1 as **'id_xy'**.

2.2.2 Define camera distortion model

We will use the simplified Contrady-Brown distortion model. This model relates distorded image coordinates $x^{'}, y^{'}$ (the points you just measured), to the undistorded ones x, y through radial coefs (k_1, k_2) and tangential coefs (p_1, p_2) :

$$x(1 + k_1 \cdot r^2 + k_2 \cdot r^4) + p_1(r^2 + 2 \cdot x^2) + 2 \cdot p_2 \cdot x \cdot y - x^{'} = 0 \tag{3}$$

$$y(1 + k_1 \cdot r^2 + k_2 \cdot r^4) + p_2(r^2 + 2 \cdot y^2) + 2 \cdot p_1 \cdot x \cdot y - y^{'} = 0 \tag{4}$$

with $r^2 = x^2 + y^2$

Tasks

(a) Define the model

We will use `scipy` python package non-linear solver to find for any measurment on the distorded image $(x^{'}, y^{'})$ the corresponding undistorded measurement (x, y).

1. Define the **undisort** function, which returns the two symbolic equations (3) and (4), given the model's parameters, and undistorded measured image coordinates.

(b) Solve the system of equations

2. Call the `fsolve` method on your undistort function to solve for x and y at each image measurement.

3. Convert `xy_corrected` list to Nx2 np.array

> Have a look at scipy.optimize.fsolve documentation to understand how to format the input.

> An initial guess can be made with the raw measurements. Hence your raw measurements will be called twice, once packed in **var** tupple, as initial guess for the fsolve to start

> optimizing, and once in x_d and y_d variables, representing the raw coordinate in the **undistort** function.

(c) Visualize the effect of the distortion on your measurements and auto evaluate your implementation

The `plot_measurements` function provided below will do this for you (no need to modify it).

- The image provided contains indications for you to evaluate your measurements and their distortion. The `plot_measurement` function will display a zoomed view around each GCP.
- On each image, the red square represents the area where your raw measurement should land. The red cross represents the actual position of your raw manual measurement. If your measurement does not fall into the square, GCP manual measurements or coordinate conversion might be incorrect.
- The green square represents the area where your undistorted measurement should land. The green cross represents the actual position of your undistorted measurement. If your measurement does not fall into the square, you mast likely have an issue with the distortion part.

Tasks

4. Project the undistorted measurements from perspective to top left coordinates
5. Concatenate measurements ids, u and v into a Nx3 np.array
6. Save np.array to a txt file, same format as part 1

2.3 : Visualize the camera distortion model

> **Objectives:** In this section, we aim at understanding the effect of the different distortion coefficient on the image correction.

The function `grid_points` generates a regular square grid on the image with spacing of **k** pixels.

Tasks

1. Use `grid_points` to generate a grid every 150 pixels up to the image size and undistort those points using the same function as in **Part 2**.
2. Define arbitrary distortion coefficients **(k1_dummy,k2_dummy,p1_dummy,p2_dummy)** and apply them to the distortion model to observe their influence on the total distortion.
3. Visualize the result of the undistorded coordinates estimation by projecting both grids (raw and corrected) to sensor centered coordinates and visualize the results.