Computer Vision

Jupyter Notebooks and Google Colab for Exercise 2

Autonomous Vision Group
University of Tübingen / MPI-IS

Environment Setup

You have two choices to complete our exercises:

1. Setup jupyter notebook locally on your machine
2. Use google colab in your browser

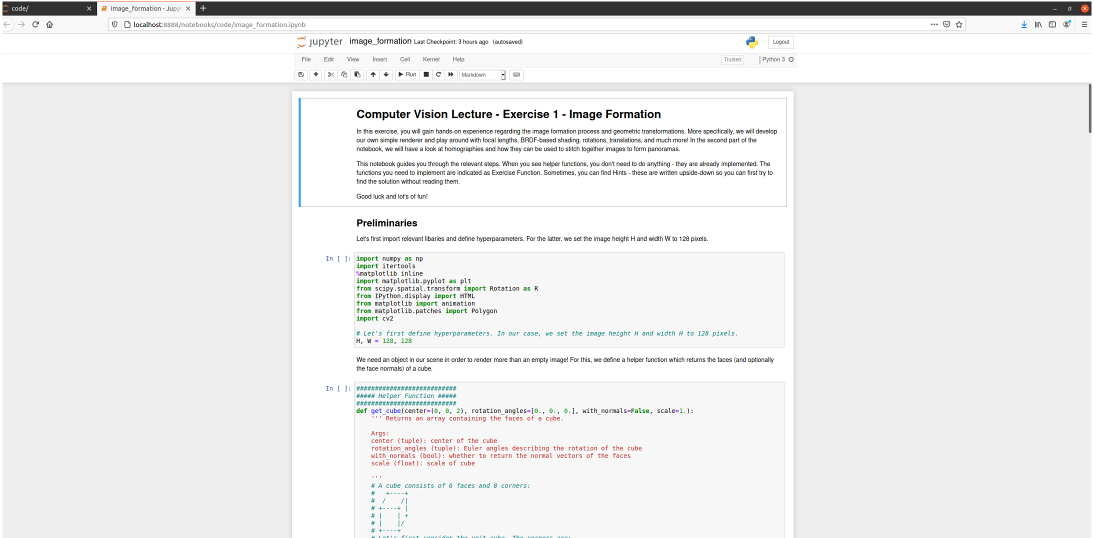
1. Local Environment Setup

1. Local Environment Setup

- Follow the instructions for your OS to install the Python package manager `conda`:
  `https://docs.conda.io/projects/conda/en/latest/user-guide/install/`
- Download the archive for exercise 1 and open a terminal in the `code` directory
- Create the new environment `lecturecv` with required packages (numpy, etc.):
  ```
  conda env create -f environment.yml
  ```
- Before launching your notebook you need to activate the environment:
  ```
  conda activate lecturecv-ex02
  ```
- Run this command from the directory where the jupyter notebooks are located:
  ```
  jupyter-notebook
  ```

1. Local Environment Setup

You can then navigate to the respective notebook and edit it in the browser


code/
image_formation - Jupyt
localhost:8888/notebooks/code/image_formation.ipynb
jupyter
image_formation Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Run
Markdown

Computer Vision Lecture - Exercise 1 - Image Formation

In this exercise, you will gain hands-on experience regarding the image formation process and geometric transformations. More specifically, we will develop our own simple renderer and play around with focal lengths, BRDF-based shading, rotations, translations, and much more! In the second part of the notebook, we will have a look at homographies and how they can be used to stitch together images to form panoramas.

This notebook guides you through the relevant steps. When you see helper functions, you don't need to do anything - they are already implemented. The functions you need to implement are indicated as Exercise Function. Sometimes, you can find Hints - these are written upside-down so you can first try to find the solution without reading them.

Good luck and lot's of fun!

Preliminaries

Let's first import relevant libaries and define hyperparameters. For the latter, we set the image height H and width W to 128 pixels.

In [ ]:

```
import numpy as np
import itertools
%matplotlib inline
import matplotlib.pyplot as plt
from scipy.spatial.transform import Rotation as R
from IPython.display import HTML
from matplotlib import animation
from matplotlib.patches import Polygon
import cv2

# Let's first define hyperparameters. In our case, we set the image height H and width H to 128 pixels.
H, W = 128, 128
```

We need an object in our scene in order to render more than an empty image! For this, we define a helper function which returns the faces (and optionally the face normals) of a cube.

In [ ]:

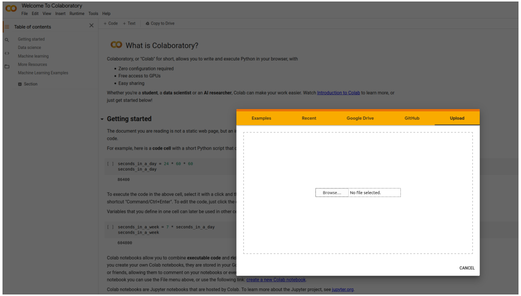
```
###########################
##### Helper Function #####
###########################
def get_cube(center=(0, 0, 2), rotation_angles=[0., 0., 0.], with_normals=False, scale=1.):
    ''' Returns an array containing the faces of a cube.

    Args:
    center (tuple): center of the cube
    rotation_angles (tuple): Euler angles describing the rotation of the cube
    with_normals (bool): whether to return the normal vectors of the faces
    scale (float): scale of cube

    '''
    # A cube consists of 6 faces and 8 corners:
    #   +----+
    #  /    /|
    # +----+ |
    # |    | +
    # |    |/
    # +----+
    # Let's first consider the unit cube. The corners are:
```

2. Online Environment Setup: Google Colab

2. Online Environment Setup: Google Colab

Alternatively, you can use Google Colab online

- Create a free Google account at: `https://google.com`
- Navigate to `https://colab.research.google.com/` in your browser
- Click on File → Upload notebook and upload the respective notebook



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an i code.

For example, here is a **code cell** with a short Python script that c

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day

    86400
```

To execute the code in the above cell, select it with a click and t shortcut "Command/Ctrl+Enter". To edit the code, just click the c

Variables that you define in one cell can later be used in other c

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week

    604800
```

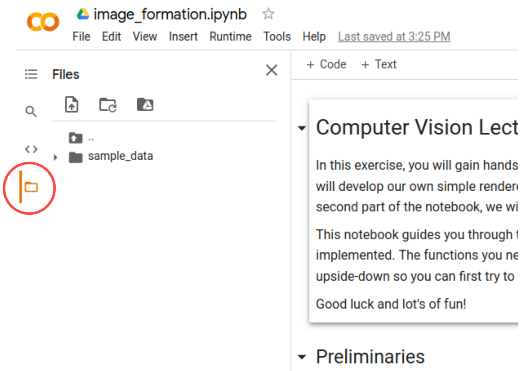
Colab notebooks allow you to combine **executable code** and **ric** you create your own Colab notebooks, they are stored in your G or friends, allowing them to comment on your notebooks or eve notebook you can use the File menu above, or use the following link: create a new Colab notebook.

Colab notebooks are Jupyter notebooks that are hosted by Colab. To learn more about the Jupyter project, see jupyter.org.

2. Online Environment Setup: Google Colab

Alternatively, you can use Google Colab online

- Create a free Google account at: `https://google.com`
- Navigate to `https://colab.research.google.com/` in your browser
- Click on File → Upload notebook and upload the respective notebook
- You can also upload additional files by clicking on the folder symbol on the left:



Computer Vision Lect

In this exercise, you will gain hands
will develop our own simple render
second part of the notebook, we wi

This notebook guides you through t
implemented. The functions you ne
upside-down so you can first try to

Good luck and lot's of fun!

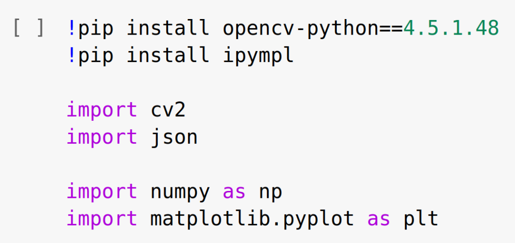
Preliminaries

Coding Exercises - Structure-From-Motion

3. Setup for the Structure-From-Motion Exercise

Running the structure-from-motion notebook in Colab requires some setup

- Colab comes with an old version of `opencv-python` that does not have SIFT
- We can a install newer version directly from the notebook running in collab
- Additionally we need to install `ipympl`
- To do that, simply uncomment the first lines of the notebook



```
!pip install opencv-python==4.5.1.48
!pip install ipympl

import cv2
import json

import numpy as np
import matplotlib.pyplot as plt
```

3. Setup for the Structure-From-Motion Exercise

Besides the dependencies we also need a few files...

- `code/sfm/img1.png`
- `code/sfm/img2.png`
- `code/sfm/cameras.npz`

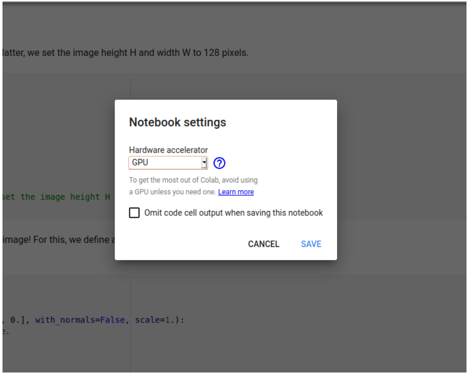
You can simply upload these from the colab environment (see prev. slides)

Coding Exercises - Stereo

3. Setup for the Stereo Exercise

If you don't have a GPU, we recommend running this notebook in Colab

- Click on Runtime → Change runtime type and select "GPU" and click "Save":



latter, we set the image height H and width W to 128 pixels.

set the image height H

image! For this, we define a

```
, 0.], with_normals=False, scale=1.):
e.
```

Notebook settings

Hardware accelerator

GPU

To get the most out of Colab, avoid using a GPU unless you need one. Learn more

□ Omit code cell output when saving this notebook

3. Setup for the Structure-From-Motion Exercise

For this we need a few files aswell:

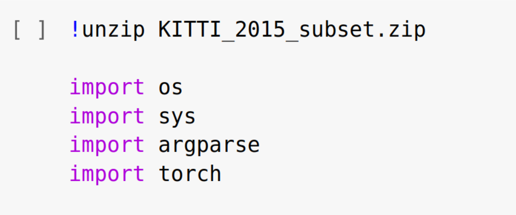
- `code/stereo/stereo_batch_provider.py`
- `code/stereo/KITTI_2015_subset`

You can simply upload these from the colab environment (see prev. slides)

3. Setup for the Structure-From-Motion Exercise

Can't upload folders - but there is a workaround

- Create a .zip archive of `code/stereo/KITTI_2015_subset`
- Upload the .zip (might take a bit)
- Uncomment the first line of the notebook to unzip the archive



```
[ ]  !unzip KITTI_2015_subset.zip

     import os
     import sys
     import argparse
     import torch
```

Questions?