Formatting and Parameters

Labelling format is based on the JavaScript Object Notation (JSON). In this document, two JSON files are explained: The first one is `cocolabel.json`, which is the format of the information used in our dataset. The second one is `cocolabel_NYU.json` which is compatible with NYU format and is also included in the dataset for compatibility purposes. At the end, camera and IMU parameters are presented.

1. Dataset Labelling

The JSON file `(cocolabel.json)` includes several fields as described below.

1.1 Information

The field of general information includes six subfields, such as the year and the name of the contribution, as described below:

Info	
contributor	Contributor of the sequence
date_created	When the sequence is created
description	Description of the sequence
url	URL for further information
version	Version
year	Year of the contribution

1.2 License

The licensing information is presented with three subfields as follows:

Licenses	
id	Licensing id
name	Name of the licensing plan
url	url for further information

1.3 Images

Information for each RGB image is described with the following fields:

Images	
id	Image id
license	Relevant license id
floorplan_ID	Associated to layout id
height	Image height
width	Image width
shutter_speed	Camera shutter speed
timestamp	Timestamp for current image
date_captured	When the image is created
file_name	Path and file name of current image
trajectoryID	Associated trajectory id
sequenceIndex	Associated to trajectory type
coco_url	Same as filename, reserved

Example:

"images":[{"coco_url":"./rgb/0.png","date_captured":"2017-10-26 19:56","file_name":"./rgb/0.png","floorplan_ID":"3FO4JD8RVAW4","id":0,"width":640,"height":480,"trajectoryID":0,"sequenceIndex":0,"timestamp":37500000,"shutter_speed":5000000,"license":1}]

1.4 Per-pixel Labelling Mask Image

For each image, labeling information is given in a 16-bit image (.png). It provides per-pixel index to encoded object names. Such object names are then associated to specific categories, explained in Sec. 1.5. The labelling fields are explained below:

Maskimages	
id	Label image id associated to image id
mask_image_filename	Path and file name of current label image
url	Same as filename, reserved

Example:

"maskimages":[{"id":0,"mask_image_filename":"./mesh/0.png","url":"./mesh/0.png"}]

1.5 Category of Objects

The field of 'categories', include three subfields: id, name and supercatergory, as described below:

categories	
Id	category id
Name	Category name, same as category id
supercategory	Parent category

Example:

"categories":[{"id":101,"name":"101","supercategory":"650"}]

1.6 2D Bounding Box Annotation

The field of 'annotations' provide the 2D bounding box information as follows:

annotations	
area	Area of bounding box
bbox	2d bounding box, defined ad left-top corner and right bottom corner
category_id	Associated to category id
image_id	Associated image id
iscrowd	Reserved
id	Annotation id
label_id	Associated label id for
mask_image_id	Associated label image id

Here an example is given for the 2D bounding box for the chair shown in Fig. 1.

{
"area": 45288,
"bbox": [
    218,
    276,
    439,

```
    479
  ],
"category_id": 179,
"id": 13,
"image_id": 0,
"iscrowd": 0,
"label_id": 36,
"mask_image_id": 0
}
```


The section above presents the 14th (id:13) annotation with label id:36. As shown in Fig.1, the chair inside the red bounding box associated to per-pixel index 36.

Figure 1: 2D bounding box for chair

2. NYU Compatible Labelling

File `cocolabel_NYUv2.json` provides association to the predefined NYU40 categories. In NYU format, the format of the '`information`', '`license`', '`images`' and '`maskimages`' are the same as the ones explained in the previous section. The '`categories`' and '`annotations`' are slightly different described in the next two subsections.

2.1 Category of Objects

This field is similar to the field shown in `cocolabel.json` file. Notice that the `supercategory` field is null for now:

Categories	
Id	Category id for NYU40
Name	Category name in NYU40
Supercategory	Parent Category, null for now

Example:

```
"categories":[{"id":1,"name":"wall","supercategory":"null"}]
```

2.2 2D Bounding Box Annotation

This field is also similar to the field shown in cocolabel.json file, as follows:

Annotations	
id	Annotation id
image_id	Associated image id
mask_image_id	Associated label image id
label_id	Associated label id for NYU40
category_id	Associated to NYU category id
Bbox	2D bounding box, defined ad left-top corner and right bottom corner
Area	Area of bounding box
Iscrowd	Reserved

3. Camera and pose Information

All camera intrinsic parameters are included in file called `cam0.info`. These parameters include image resolution (width and height), focal length in pixels (Fx, Fy), principal point, distortion coefficients, and the coordinate system. The system of coordinates is based on the right hand system, with Y axis pointing up. An example is shown here:

```
#Image Res.(w,h):
640 480
#focus lens:
600 600
#principal pt:
320 240
#Undist. coeff. :
0 0 0
#Coordinate Sys.:
RH_Y_UP
```

All intrinsic and extrinsic camera parameters are included in a separate file called `cam0.ccam`. The header of the file includes the following information:

Parameter	**Description**
`Version`	ViSim camera format version
`camera no`	Number of the camera poses
`f`	Focal length in pixels unit
`cx`	Principal point along x in pixels unit
`cy`	Principal point along y in pixels unit
`dist.coeff[0]`	Distortion coefficients in OpenCV format
`dist.coeff[0]`	Distortion coefficients in OpenCV format
`dist.coeff[0]`	Distortion coefficients in OpenCV format
`dist.coeff[0]`	Distortion coefficients in OpenCV format
`orientation, w`	Orientation in quaternions
`orientation, x`	Orientation in quaternions
`orientation, y`	Orientation in quaternions
`orientation, z`	Orientation in quaternions
`position, x`	Position
`position, y`	Position
`position, z`	Position
`width`	image resolution: width in pixels unit
`height`	image resolution: height in pixels unit

Example:

```
#ViSim camera format version:
2
#Camera No.
1000
#<list of cameras>
#<camera info: f, cx, cy, dist.coeff[0],dist.coeff[1],dist.coeff[2]>
<orientation: w,x,y,z> <position: x,y,z> <image resolution: width,
height>
600 320 240 0 0 0 0.586408615 0.58603406 -0.395277888 -0.395530581 -
1.67119229 -1.93938518 1.15760112 640 480
```

This example describes 1000 camera poses, only the first line is shown. Each image is 640x480 with camera focal length of 600 pixels.

Additionally, all extrinsic (pose and orientation) with their timestamps are included in a separate file: `cam0_gt.visim`

4. Renderer Parameters

The renderer parameters are included in the `cam0.renderer`. These parameters are '`eye`' position, '`lookat`' position, '`up`' position, '`framerate`', and '`shutter_speed`'. An example is shown below, where information for the first pose is included:

```
#first three elements, eye and next three, lookAt and the last there,
up direction
#frame rate per second: 25
#shutter_speed (ns): 5000000
0000000000037500000 -1.67119229 -1.93938518 1.15760112 2.30785275 -
0.331670374 1.15485859 -1.67059982 -1.9391458 2.15760088
```

Abovementioned information is ground truth, and in another file, called cam0_shutter.renderer, the shutter-open pose and shutter-close pose are included following the format used for cam0.renderer.

Additionally, list of all images are included in cam0.timestamp, where each line has only two parameters: timestamp (ns) and the name of the image. For example:

```
#timestamp [ns], filename of cam0
000000000037500000,000000000037500000.png
```

5. IMU measurements

Timestamped measurements from a simulated inertial measurement unit (IMU) is output in CSV format. Each row of the file includes timestamp, linear acceleration and angular velocity. The first line of the `imu.csv` file is given below. The rest of the lines are measurements.

```
#timestamp[ns],
w_RS_S_x[rad s^-1],
w_RS_S_y[rad s^-1],
w_RS_S_z[rad s^-1],
a_RS_S_x[m s^-2],
a_RS_S_y[m s^-2],
a_RS_S_z[m s^-2]
```

The simulated IMU parameters, with some example values, are given below:

Parameter	**Description**
`a_max = 176`	max. acceleration [m/s^2]
`g_max = 7.8`	max. rotation speed [rad/s]
`sigma_g_c = 0.0012`	gyro noise density [rad/(s*sqrt(Hz))]
`sigma_a_c = 0.008`	accelerometer noise density [m/(s^2*sqrt(Hz))]
`sigma_gw_c = 4e-06`	gyro bias noise density [rad/(s^2*sqrt(Hz))]
`tau_g = 3600`	gyro bias autocorrelation time [s]
`tau_a = 3600`	accelerometer bias autocorrelation time [s]
`g = 9.81`	Earth's acceleration due to gravity [m/s^2]
`rate = 800`	IMU sample rate [Hz]