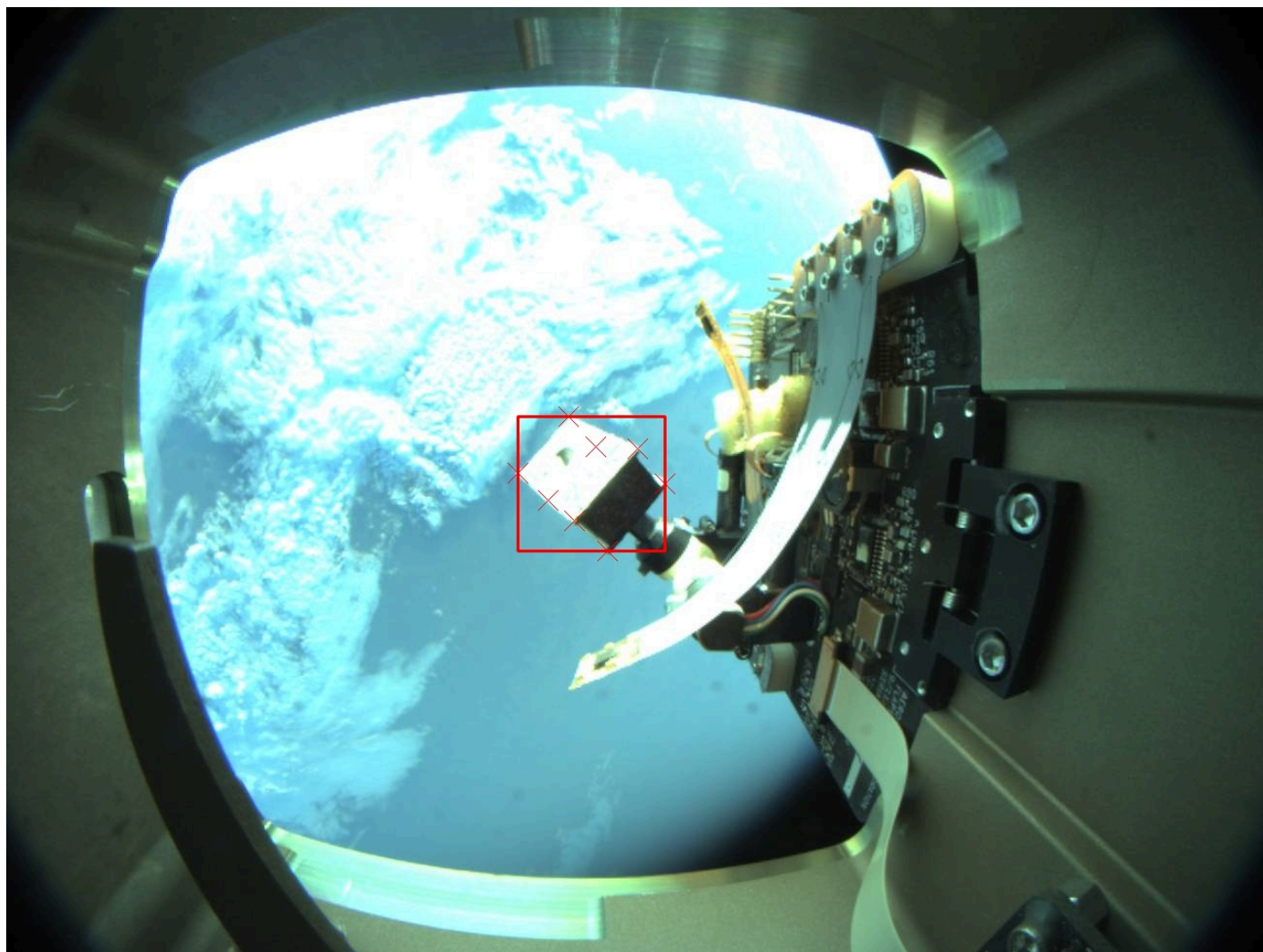


Data Description

- [What is in this Dataset?](#)
- [What is ExoRomper?](#)
- [Data Structure](#)
- [Ground Truth JSON](#)
- [Frames](#)

What is in this Dataset?



This dataset, the ExoRomper Satpose Dataset V3, contains images from the ExoRomper payload's flight unit aboard the Slingshot-1 LEO space vehicle, supplemented by imagery from the ground unit in Aerospace labs. This dataset is intended for public release and is intended for use in the development of:

- Machine Learning-based algorithms for processing visible-light (RGB) camera data in satellite pose estimation applications, which are relevant for autonomous rendezvous, proximity, docking, and undocking operations.

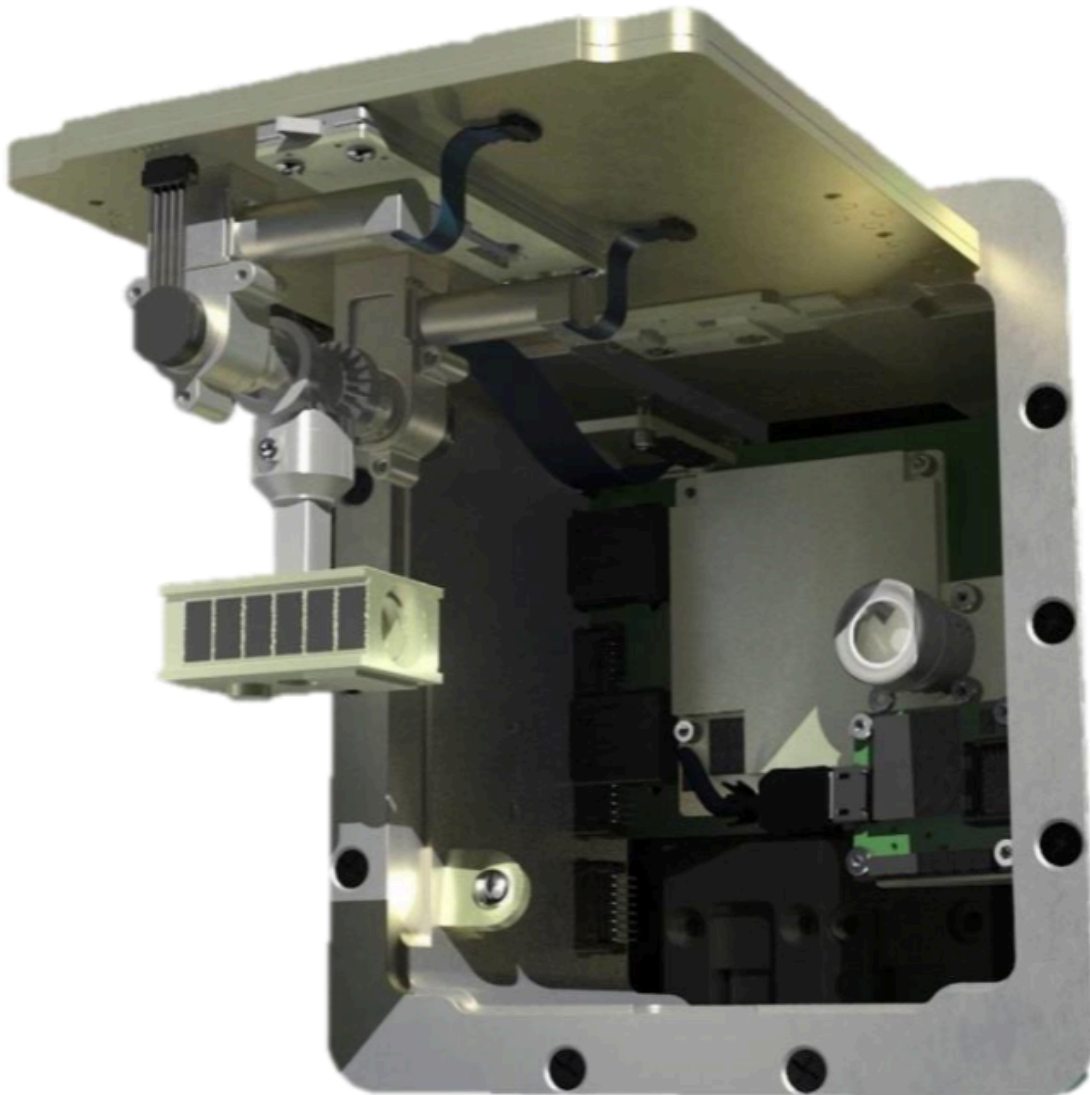
- Trusted AI/ML for space vehicles and space systems, e.g. confidence and uncertainty metrics for AI/ML outputs, out-of-distribution detection, etc.

For each image file, there is an associated label file containing information related to the image, e.g. the ExoRomper mini satellite pose (position and orientation). The contents of the label files are described in more detail later in this document.

What is ExoRomper?

ExoRomper is a machine vision platform onboard Aerospace's Slingshot-1 space vehicle. The platform is testing satellite pose estimation algorithms in a LEO operational environment.

The payload, rendered below, features a re-positionable miniature spacecraft model on a two-axis gimbal. The mini spacecraft and gimbal assembly sit on a wing deployed off the side of the Slingshot-1 space vehicle, exposed to the lighting conditions of the LEO space environment. A visible light camera points at the mini spacecraft and captures imagery that is passed to the payload's onboard Exo-CoralReef AI processor, where the images are processed to produce estimates of the mini satellite's pose with respect to the camera frame (pose := position and orientation).



The goal is to improve satellite pose estimation accuracy in operational lighting conditions by obtaining data directly from the operational environment.

Data Structure

The dataset contains the following structure:

```

data/
  seed/ # RAW DATA input to preprocessing stage
    run_1/ # every run is a separate data collect
      _camera_settings.json # camera calib for this run
      _object_settings.json # obj settings for this run
      images/
        xxx.jpg # RAW images: .jpg, .png, .tif
      labels/ # manually labeled keypoints
        xxx.json
    run_2/
      _camera_settings.json
      _object_settings.json
      images/
        xxx.jpg
      labels/
        xxx.json
    ...
    run_n/ # append data collects
      _camera_settings.json
      _object_settings.json
      images/
        xxx.jpg
      labels/
        xxx.json
  raw/ # output from preprocessing stage
    images/
      xxx.jpg # RAW images: .jpg, .png, .tif
    labels/
      xxx.txt # YOLO format labels
    gt_json/
      xxx.json # image labels w/ all details
  rectified/ # output from preprocessing stage
    images/
      xxx.jpg # RECTIFIED images: .jpg, .png, .tif
    labels/
      xxx.txt # YOLO format labels
    gt_json/
      xxx.json # image labels w/ all details
  yolo_dataset.yaml
  kpt_dataset.yaml

```

- The `seed/` folder contains the raw data, camera calibration info, and manually labeled keypoints from each individual data collection run. The data in `seed/` is used to populate the data in `raw/` and `rectified/`. **Users should interact with the data in `raw/` and `rectified/` rather than the data in `seed/`.**
- Images are one of the following formats: `.jpg`, `.png`, `.tif`. The `raw/images/` and `rectified/images/` folders contain the raw (unrectified) and rectified images, respectively. The labels for these images are in the adjacent

labels/ and gt_json/ folders.

- The labels in raw/labels and rect/labels follow the YOLO format, defined [here](#)
- The json files in gt_json (ground truth json) are defined in [Ground Truth JSON Template](#).
 - There is one label per image with the same exact title as the image, just with the .json file ending instead of the image's file ending.
 - The camera_matrix and dist_coeffs for each image are in its ground truth json file.
 - The rectified data is generated by rectifying the raw image and label data. Therefore, bounding boxes, keypoints and camera intrinsic values are slightly different in the rectified data due to the adjustment for rectification of the image.

Ground Truth JSON

Here is an example json file meant to accompany an image:

```

{
  "objects": [
    {
      "class": "ExoRomper_Sat",
      "instance_id": 0,
      "location": [
        10.641682144175379,
        -12.1142061396896,
        90.5900254169129
      ],
      "quaternion_xyzw": [
        0.13755139387069099,
        0.9689906619448471,
        -0.19976046077673698,
        0.04724901499696178
      ],
      "dcm_sat2cam": [
        [
          -0.9576942892520939,
          0.2854490024104745,
          0.036613048996112674
        ],
        [
          0.24769506238231437,
          0.8823507447089918,
          -0.40013037798061457
        ],
        [
          -0.14652236827647933,
          -0.3741337064932174,
          -0.9157265777841652
        ]
      ],
      "pose_transform": [
        [
          -0.9576942892520939,
          0.2854490024104745,
          0.036613048996112674,
          10.641682144175379
        ],
        [
          0.24769506238231437,
          0.8823507447089918,
          -0.40013037798061457,
          -12.1142061396896
        ],
        [

```

```

        -0.14652236827647933,
        -0.3741337064932174,
        -0.9157265777841652,
        90.5900254169129
    ],
    [
        0.0,
        0.0,
        0.0,
        1.0
    ]
],
"cuboid_centroid": [
    10.641682144175379,
    -12.1142061396896,
    90.5900254169129
],
"projected_cuboid_centroid": [
    343.53404693223246,
    214.9507466363741
],
"bounding_box": {
    "top_left": [
        318.80785770492025,
        180.60842506316493
    ],
    "bottom_right": [
        375.02820980354664,
        244.71722539878405
    ]
},
"cuboid": [
    [
        1.9940749258747967,
        -19.69639552904419,
        79.68857163945943
    ],
    [
        17.319098942486804,
        -23.66001191728598,
        82.03322257661965
    ],
    [
        20.219260806977225,
        -14.695328351042622,
        78.23202411864857
    ],
    [

```

```
        4.894236790365218,  
        -10.73171196280083,  
        75.88737318148834  
    ],  
    [  
        1.0641034813735342,  
        -9.533083928336577,  
        102.94802671517722  
    ],  
    [  
        16.38912749798554,  
        -13.496700316578371,  
        105.29267765233745  
    ],  
    [  
        19.289289362475962,  
        -4.532016750335013,  
        101.49147919436636  
    ],  
    [  
        3.9642653458639554,  
        -0.568400362093219,  
        99.14682825720614  
    ]  
],  
"projected_cuboid": [  
    [  
        322.1434863351526,  
        189.01601029416966  
    ],  
    [  
        363.9781269514589,  
        180.60842506316493  
    ],  
    [  
        375.02820980354664,  
        203.150644989655  
    ],  
    [  
        331.3286950280463,  
        213.0973871357615  
    ],  
    [  
        318.80785770492025,  
        224.38742178917437  
    ],  
    [  
        352.2694501832233,
```



```
        216.31006307692778
      ],
      [
        360.23066033543773,
        235.69287140188734
      ],
      [
        325.7125028453065,
        244.71722539878405
      ]
    ],
    "projected_cuboid_man_label": [
      [
        321.0,
        189.0
      ],
      [
        364.0,
        180.0
      ],
      [
        375.0,
        205.0
      ],
      [
        332.0,
        212.0
      ],
      [
        318.0,
        225.0
      ],
      [
        -1.0,
        -1.0
      ],
      [
        -1.0,
        -1.0
      ],
      [
        327.0,
        244.0
      ]
    ],
    "avg_label_error": 1.229342987447524,
    "label_inliers": [
      0,
```

```

        1,
        2,
        3,
        4,
        7
    ],
    "camera_matrix": [
        [
            232.84493735225536,
            0.0,
            316.40612299842843
        ],
        [
            0.0,
            234.6340808707978,
            246.06154875369714
        ],
        [
            0.0,
            0.0,
            1.0
        ]
    ],
    "dist_coeffs": [
        -0.2609812247001386,
        0.06865645970173294,
        0.0007012444183362578,
        0.0003518821317049685,
        -0.007806651855746771
    ],
    "camera_name": "ximea_ground",
    "distortion_model": "plumb_bob",
    "image_height": 486,
    "image_width": 648
}
]
}

```

These ground truth json files are auto-created from the manually-labeled keypoints in the `dataset/seed` directory. They contain the following elements:

- `class` : class name. For this dataset, there is only one class, "ExoRomper_Sat".
- `instance_id` : instance of this class in the image (zero-indexed; there is only one satellite ever in the image so the value should always be zero)
- `location` : XYZ position of the object's frame w.r.t. the camera frame (in mm)
- `quaternion_xyzw` : A scalar-last quaternion that maps the rotation of points from the ExoRomper target satellite frame to the camera frame. This is an active quaternion.

- `dcm_sat2cam` : The Direction Cosine Matrix (DCM) that passively maps the rotation of points from the ExoRomper target satellite frame to the camera frame.
- `pose_transform` : 4x4 homogeneous transformation matrix that passively maps the rotation of points from the ExoRomper target satellite frame to the camera frame (redundant but provided for convenience, can be assembled from `location` and `dcm_sat2cam`).
- `cuboid_centroid` : 3D position of the ExoRomper target frame origin w.r.t. the camera frame (in mm).
- `projected_cuboid_centroid` : 2D projection of the previous onto the image (in pixels).
- `bounding_box` : 2D bounding box of the ExoRomper target in the image (in pixels)
- `cuboid` : 3D coordinates of the vertices of the 3D bounding cuboid, keypoints 0-7 inclusive, in mm.
- `projected_cuboid` : 2D coordinates of the projection of the above (in pixels).
- `projected_cuboid_man_label` : manually labeled keypoints (in pixels). Only available for the data in `raw/`, not in `rectified/`. A value of `-1` means that the keypoint was not labeled, likely because it was not visible.
- `avg_label_error` : this is the average reprojection error between the manually labeled keypoints in `projected_cuboid_man_label` and their PnP-calculated counterparts in `projected_cuboid` (omitting keypoints that were not manually labeled from the calculation of this average error value).
- `label_inliers` : indices for the manually labeled keypoints that were returned as inliers by [solvePnP Ransac](#), which was used to calculate the pose given the manually labeled 2D keypoint positions.
- `camera_matrix` : the principal point and focal lengths of the camera formed as the `cameraMatrix` passed to [solvePnP Ransac](#) and detailed further [here](#)
- `dist_coeffs` : the camera's distortion coefficients listed as `(k_1, k_2, p_1, p_2[, k_3[, k_4, k_5, k_6]])`. `k_1`, `k_2`, `k_3`, `k_4`, `k_5`, and `k_6` are radial distortion coefficients. `p_1` and `p_2` are tangential distortion coefficients. These coefficients are meant to be passed into to OpenCV's [solvePnP Ransac](#) and detailed further [here](#)
- `camera_name` : the camera name. Indicates which camera this was taken from - the ground unit's "ximea_ground" or the flight unit's "ximea_flight"
- `distortion_model` : the camera distortion model
- `image_height` : the image height (pixels)
- `image_width` : the image width (pixels)

Each element of `projected_cuboid_centroid`, `bounding_box`, `projected_cuboid`, and `projected_cuboid_man_label` is a list whose elements are the pixel coordinates of image keypoints. The values are w.r.t. the following OpenCV image coordinate frame used for points in the 2D image frame, where (0,0) is located at the top-left of the image:

```

0-----> +U
|
|
|
|
v
+V

```

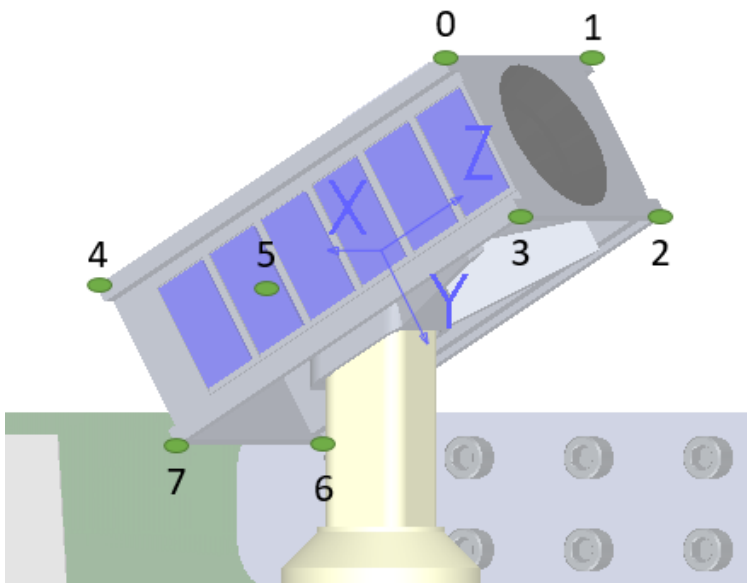
And coordinates are noted as

```

"projected_cuboid_centroid": [ u, v],
"bounding_box":
{
  "top_left": [ u, v],
  "bottom_right": [ u, v]
},
"projected_cuboid": [
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v]
],
"projected_cuboid_man_label": [
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v],
  [u, v]
],

```

The keypoints are numbered and indexed as follows:



And their values are expressed with respect to the ExoRomper target satellite frame illustrated above at the target satellite's 3D centroid.

Frames

The ExoRomper target satellite frame has already been illustrated above.

The camera frame follows OpenCV conventions for the pinhole camera model where the $+Z_c$ is the optical axis, the $+X_c$ axis points to the right (inline with the $+U$ axis of the 2D image frame), and the $+Y_c$ axis points down (inline with the $+V$ axis of the 2D image frame).

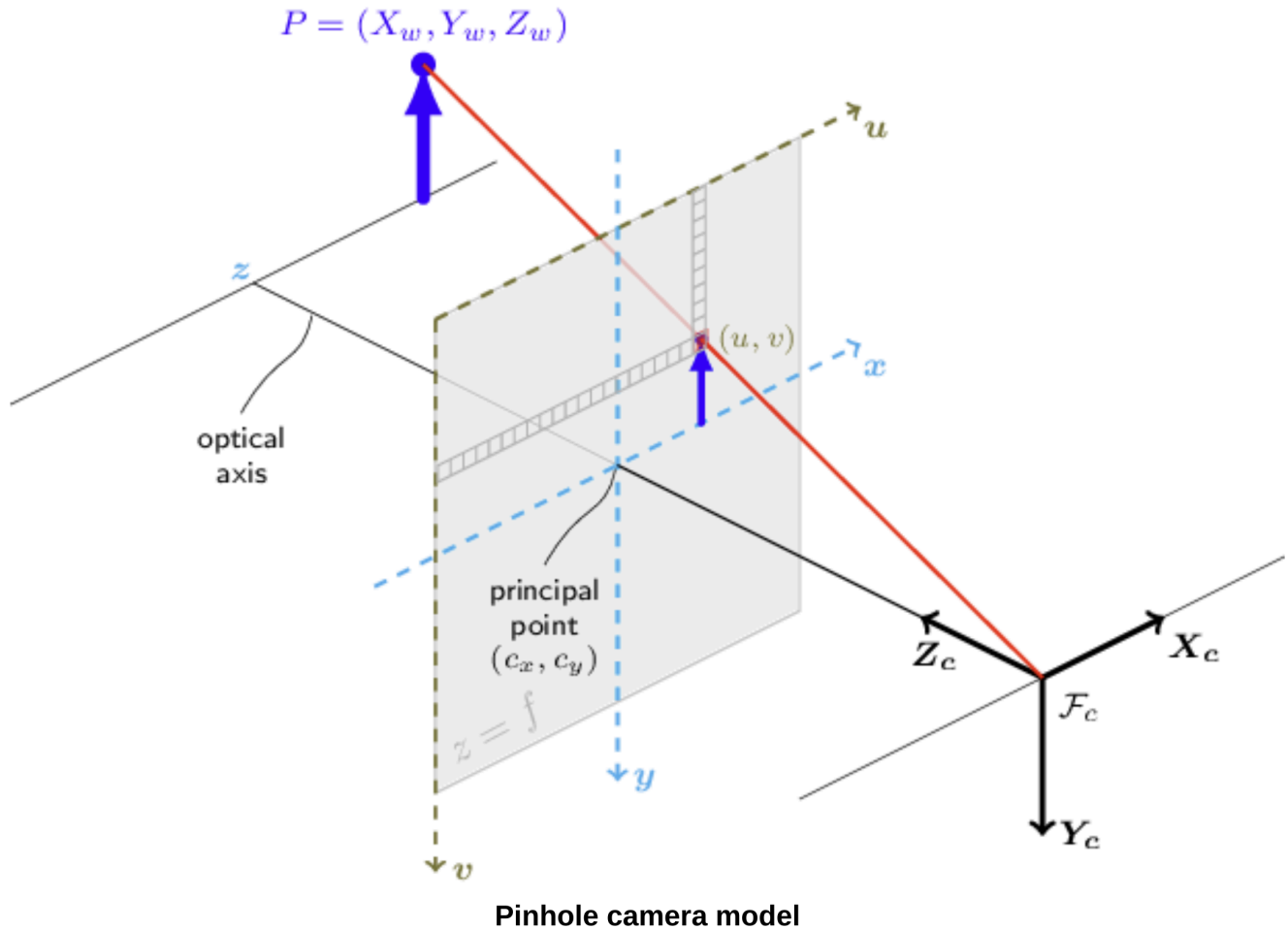


Image source: [OpenCV Camera Calibration and 3D Reconstruction](#)