# Jump Stitch Metadata & Depth Maps
# Version 1.2

jump-help@google.com

## Contents



## 1. Introduction

This document defines the stitch metadata and depth map file formats produced by the Jump Assembler. It also describes the equations by which depth map values map to 3D points in the scene and how 3D scene points project into the source cameras and the panorama.

## 2. Stitch Metadata File Format

For every shot, the Jump Assembler exports a JSON formatted metadata file. This file contains information about the rig including calibration parameters for each of the cameras, the vertical field of view of the shot, and the interpupillary distance. See Appendix A for an example.

In this section we describe each field found in the metadata file. Some field names are followed by the mathematical symbol(s) that represent their value in equations in this document. The number in brackets after each field gives the first version of the metadata spec in which it appeared. All fields are included in each metadata file even if they are not directly relevant, for example depth related fields are included even in a stitching session which contains no depth stitches.

`major_version_number` (v1.0): Major version of the metadata file. This value will be incremented when changes are made to the file format which are not backward compatible.

`minor_version_number` (v1.0): Minor version of the metadata file. Incrementing this value indicates that one or more new fields have been added to the format. Any other type of change will result in an update to the `major_version_number` instead.

`interpupillary_distance_in_meters` (v1.0): The interpupillary distance (IPD) used for rendering the stitch, measured in meters.

`rig_type` (v1.0): A string defining the type of rig used for the capture. Currently the only valid values are `Odyssey` and `Halo`.

`rig_id` (v1.0): A string that gives a globally unique rig id. Two shots with the same `rig_id` must have been captured using the same physical rig, although calibration may still change between the shots as individual components move or are replaced.

`missing_bottom_coverage_in_degrees` (v1.0): The elevation, in degrees from the bottom of the stitch, where content is missing for the color stitch.

`missing_top_coverage_in_degrees` (v1.0): The elevation, in degrees from the top of the stitch, where content is missing for the color stitch.

`missing_bottom_depth_coverage_in_degrees` (v1.1): The elevation, in degrees from the bottom of the stitch, where content is missing for the depth stitch.

`missing_top_depth_coverage_in_degrees` (v1.1): The elevation, in degrees from the top of the stitch, where content is missing for the depth stitch.

`mono_top_coverage_in_degrees` (v1.1): The elevation range, in degrees, for which the top of the stitch is monoscopic. See section 3 for more details.

`mono_bottom_coverage_in_degrees` (v1.1): The elevation range, in degrees, for which the bottom of the stitch is monoscopic. See section 3 for more details.

`partial_stereo_top_coverage_in_degrees` (v1.1): The elevation range, in degrees, over which the top of the stitch transitions from mono to stereo. See section 3 for more details.

`partial_stereo_bottom_coverage_in_degrees` (v1.1): The elevation range, in degrees, over which the bottom of the stitch transitions from mono to stereo. See section 3 for more details.

`minimum_encodable_depth_in_meters` (v1.0): Depth stitches are encoded as inverse depth and this value is the minimum depth which can be encoded. See section 7 for more details.

`distortion_delta_in_meters` (v1.0): $\delta$ - Jump stitches use an omnidirectional stereo (ODS) projection that has been distorted to better fit the physical camera geometry. This value is the distance by which ODS viewing ray origins are shifted to produce the distorted ODS projection. See Section 6 for more details.

`frames_per_second` (v1.0): The framerate of the stitch.

`stitching_algorithm` (v1.2): The algorithm used to produce the stitch. Can either be `OPTICAL_FLOW` or `MULTI_VIEW_STEREO` which correspond to the 'Standard' and 'High' stitching quality options in Jump Manager respectively.

`cameras` (v1.0): An array of *cameras* which provide the intrinsics and extrinsics of each camera in the rig as described below.

Each camera in the `cameras` array contains the following fields:

`name` (v1.0): The name of the camera. By convention this is the name of the video file corresponding to that camera without the file extension.

`position` (v1.0): $[c_x, c_y, c_z]$ - The position of the camera in world coordinates. Given in meters.

`orientation` (v1.0): $[a_x, a_y, a_z]$ - The orientation of the camera in angle-axis format, where angle in radians is equal to the magnitude of the vector.

`projection_type` (v1.0): The projection model used for the camera. Either `perspective` or `fisheye`. For the `Odyssey` and `Halo` rigs this value will always be `fisheye`.

`image_size` (v1.0): $[w, h]$ - The width and height of the camera's image in pixels.

Table 1: Metadata field names and the mathematical symbols corresponding to them.

| JSON Field | Symbol |
|---|---|
| `distortion_delta_in_meters` | $\delta$ |
| `camera::position` | $[c_x, c_y, c_z]$ |
| `camera::orientation` | $[a_x, a_y, a_z]$ |
| `camera::image_size` | $[w, h]$ |
| `camera::principal_point` | $[u_0, v_0]$ |
| `camera::radial_distortion` | $[r_0, r_1]$ |
| `camera::focal_length` | $f$ |

`principal_point` (v1.0): $[u_0, v_0]$ - The horizontal and vertical position of the camera's principal point in pixels.

`radial_distortion` (v1.0): $[r_0, r_1]$ - First and second order radial distortion parameters for the camera.

`focal_length` (v1.0): $f$ - Focal length of the camera in pixels.

## 3. Coverage Near the Poles

The ODS projection does not accurately model stereo near the poles. One way to avoid the distortion this introduces is to fade to a monoscopic stitch near the poles. This is done by reducing the IPD of the stitch near the poles. Depending on the rig used for capture it is also possible that content is not available at one or both of the poles. This section defines the parameters used to describe the stitch near the poles.

At the top and bottom pole there are three distinct regions which don't exhibit fully stereo content. The layout of the regions is shown below and the corresponding ranges in a sample stitch are shown in figure 1.

| |
|---|
| `missing_top_coverage_in_degrees` |
| `mono_top_coverage_in_degrees` |
| `partial_stereo_top_coverage_in_degrees` |
| Main region of stitch (fully stereo) |
| `partial_stereo_bottom_coverage_in_degrees` |
| `mono_bottom_coverage_in_degrees` |
| `missing_bottom_coverage_in_degrees` |

We will describe these regions for the top of the stitch, the ones at the bottom are exactly analagous.

> The first region contains no valid image data, color is diffused in from the boundary of the next region. This region spans an elevation range from zero to `missing_top_coverage_in_degrees`.
>
> The second region contains a monoscopic stitch (corresponding to an IPD of zero). This region spans an elevation range from `missing_top_coverage_in_degrees` to (`missing_top_coverage_in_degrees` + `mono_top_coverage_in_degrees`).
>
> The third region contains a stereo stitch with varying IPD. The exact value of the IPD in this region is not specified but it is guaranteed to be monotonically increasing from zero at the beginning of the region to `interpupillary_distance_in_meters` at the end of the region. This region spans an elevation range from (`missing_top_coverage_in_degrees` + `mono_top_coverage_in_degrees`) to (`missing_top_coverage_in_degrees` + `mono_top_coverage_in_degrees` + `partial_stereo_top_coverage_in_degrees`).

Figure 1 shows the values described above for one eye from a stitch captured with a Halo rig. Section 6 defines exactly how these values interact with IPD.

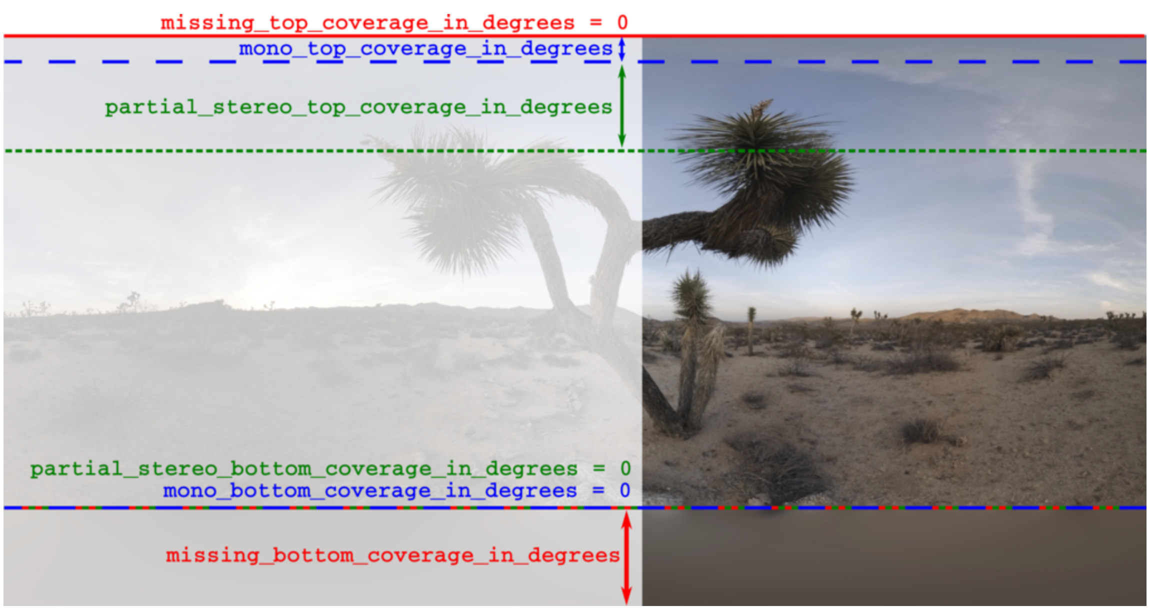


Figure 1: Sample color frame for a single eye from a Halo rig with coverage values marked.

For depth stitches we only provide depth information in the region for which we have full stereo. This means that only one value is required for the top pole and one for the bottom pole. The region at the top of the stitch for which we have no depth data spans an elevation range of zero to `missing_top_depth_coverage_in_degrees`. Within this region depth is diffused from the region boundary. This is show in figure 2.

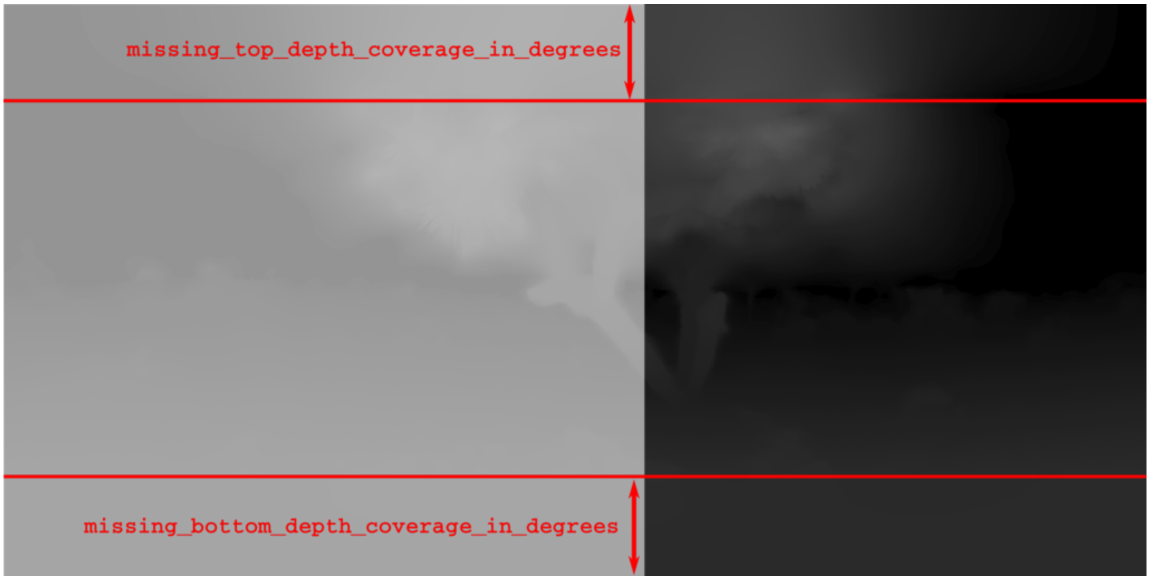


Figure 2: Sample depth frame for a single eye from a Halo rig with coverage values marked.

## 4. Coordinate Systems

In image space the top left corner of the top left pixel has coordinates $[0, 0]$. The center of the top left pixel lies at $[0.5, 0.5]$ and for an image size of $w \times h$ the center of the bottom right pixel is at $[w - 0.5, h - 0.5]$.

We use a right handed coordinate system. A camera at the origin which has not been rotated looks down the positive $z$ axis, with the positive $x$ axis pointing to the right and the positive $y$ axis pointing down.

The output stitches are over-under equirectangular in which the left eye is on top. The positive $y$ axis points down the image. The positive $z$ axis corresponds to the left edge of the stitch.

## 5. Camera Model

In this section, we describe how a point in world space with homogeneous coordinates $[p_x, p_y, p_z, p_w]$ is projected into the image space of one the cameras in the rig with coordinates $[u, v]$. The process can be broken down into two steps as follows:

1. **Transform Into Camera Space**

   For a camera with `position` $[c_x, c_y, c_z]$ and `orientation` $[a_x, a_y, a_z]$ we transform a point in homogeneous coordinate in world space $[p_x, p_y, p_z, p_w]$ to a point in camera space $[x, y, z]$ using the following equation,

   $$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} p_x - p_w \times c_x \\ p_y - p_w \times c_y \\ p_z - p_w \times c_z \end{bmatrix}. \tag{1}$$

   Here, the $3 \times 3$ rotation matrix $R$ can be computed from from the angle axis representation of `orientation` $[a_x, a_y a_z]$ as follows:

   $$\theta = \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2}, \tag{2}$$

   $$\begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} = \frac{1}{\theta} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3}$$

   $$R = \begin{bmatrix} \hat{a}_x\hat{a}_x(1-\cos\theta)+\cos\theta & \hat{a}_x\hat{a}_y(1-\cos\theta)-\hat{a}_z\sin\theta & \hat{a}_x\hat{a}_z(1-\cos\theta)+\hat{a}_y\sin\theta \\ \hat{a}_x\hat{a}_y(1-\cos\theta)+\hat{a}_z\sin\theta & \hat{a}_y\hat{a}_y(1-\cos\theta)+\cos\theta & \hat{a}_y\hat{a}_z(1-\cos\theta)-\hat{a}_x\sin\theta \\ \hat{a}_x\hat{a}_z(1-\cos\theta)-\hat{a}_y\sin\theta & \hat{a}_y\hat{a}_z(1-\cos\theta)+\hat{a}_x\sin\theta & \hat{a}_z\hat{a}_z(1-\cos\theta)+\cos\theta \end{bmatrix} \tag{4}$$

   Here, $\theta$ is the angle of rotation around the unit vector $[\hat{a}_x, \hat{a}_y, \hat{a}_z]$.

2. **Project Into Image Space**

   Given `focal_length` $f$, `principal_point` $[u_0, v_0]$ and `radial_distortion` $[r_0, r_1]$, we can now, depending on the `projection_type`, project a point $[x, y, z]$ in camera space into the image as follows.

- `fisheye`

$$\alpha = \tan^{-1}\frac{\sqrt{x^2+y^2}}{z} \tag{5}$$

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \frac{\alpha}{\sqrt{x^2+y^2}}\begin{bmatrix}x\\ y\end{bmatrix}, \tag{6}$$

$$d = 1 + r_0\alpha^2 + r_1\alpha^4 \tag{7}$$

$$\begin{bmatrix}u\\ v\end{bmatrix} = \begin{bmatrix}fd\hat{x}+u_0\\ fd\hat{y}+v_0\end{bmatrix}. \tag{8}$$

  Here, $\alpha$ is the the angle between the optical axis and the ray to the point $[x, y, z]$, and $d$ is the radial distortion factor.

- `perspective`

$$\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} = \frac{1}{z}\begin{bmatrix}x\\ y\end{bmatrix} \tag{9}$$

$$d = 1 + r_0\left(\hat{x}^2+\hat{y}^2\right) + r_1\left(\hat{x}^2+\hat{y}^2\right)^2 \tag{10}$$

$$\begin{bmatrix}u\\ v\end{bmatrix} = \begin{bmatrix}fd\hat{x}+u_0\\ fd\hat{y}+v_0\end{bmatrix} \tag{11}$$

  Here, $d$ is the radial distortion factor.

## 6. Panorama projection model

In this section we describe the mapping from a point in the stitched panorama to a 3D ray and the mapping from 3D points back into the panorama.

Jump videos use the omnidirectional stereo (ODS) projection. A summary of the ODS model and how it can be used to render imagery can be found in Rendering Omni-directional Stereo Content[1]. That document describes the *ideal* ODS projection, where viewing rays originate from a circle of diameter equal to the interpupillary distance (IPD). In practice though, the diameter of a Jump camera is significantly larger than IPD. To avoid holes in the stitch, we use a *distorted ODS projection* as shown in Figure 3. In this model the rays originate from the circle on which the cameras lie. Visually this results in subjects that are close to the camera being vertically stretched.

[1]Currently Rendering Omni-directional Stereo Content and this document differ in their coordinate system conventions. We are working on fixing this discrepancy.

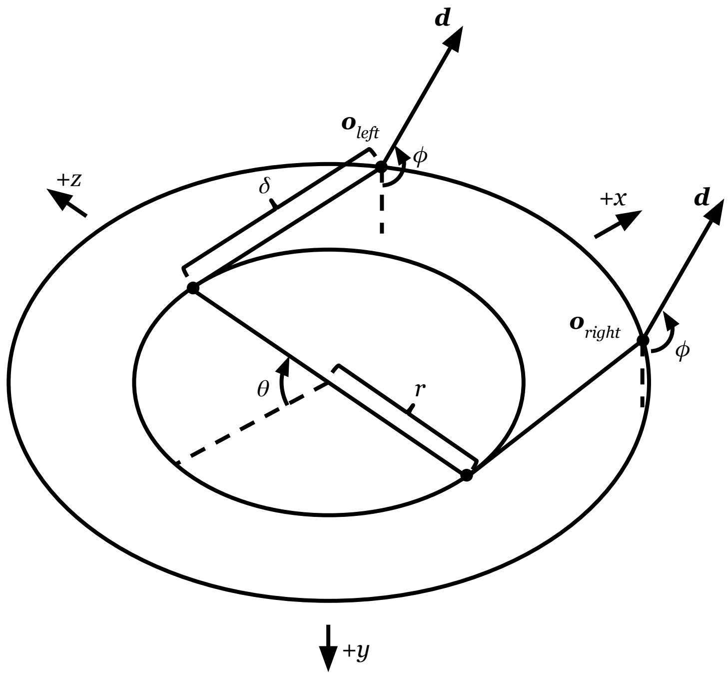


Figure 3: This figure shows the distorted ODS model where ray origins are shifted horizontally by the distance $\delta$ to $\boldsymbol{o}_{\text{left}}$ and $\boldsymbol{o}_{\text{right}}$ .

## 6.1. Projecting from the Panorama to a 3D Ray

Consider a pixel with position $[\theta, \phi]$ in an equirectangular panorama, where $\theta \in [0, 2\pi]$ is the latitude and $\phi \in [0, \pi]$ is the longitude and $[0, 0]$ corresponds to the bottom left of the panorama. Then for the two eyes, the respective rays have the same direction

$$\boldsymbol{d}(\theta, \phi) = \begin{bmatrix} \sin\theta\sin\phi \\ \cos\phi \\ \cos\theta\sin\phi \end{bmatrix}, \tag{12}$$

but different origins:

$$\boldsymbol{o}_{\text{left}}(\theta, \phi) = k \begin{bmatrix} \delta\sin\theta - r\cos\theta \\ 0 \\ \delta\cos\theta + r\sin\theta \end{bmatrix}, \quad \boldsymbol{o}_{\text{right}}(\theta, \phi) = k \begin{bmatrix} \delta\sin\theta + r\cos\theta \\ 0 \\ \delta\cos\theta - r\sin\theta \end{bmatrix}. \tag{13}$$

where $r =$ `interpupillary_distance_in_meters`$/2$.

For the majority of the stitch $k$ is 1 but near the poles it is reduced to avoid distortion (by reducing the effective IPD and distortion delta of the stitch):

$$k = \begin{cases} 0, & \text{if } \phi_d < a \\ \text{undefined}, & \text{if } a <= \phi_d < b \\ 1, & \text{if } b <= \phi_d < c \\ \text{undefined}, & \text{if } c <= \phi_d < d \\ 0, & \text{if } d <= \phi_d \end{cases} \tag{14}$$

where

$$\phi_d = \phi * 180/\pi \tag{15}$$

$$a = \texttt{missing\_bottom\_coverage\_in\_degrees} + \texttt{mono\_bottom\_coverage\_in\_degrees} \tag{16}$$

$$b = a + \texttt{partial\_stereo\_bottom\_coverage\_in\_degrees} \tag{17}$$

$$c = 180 - (\texttt{missing\_top\_coverage\_in\_degrees} + \texttt{mono\_top\_coverage\_in\_degrees}) \tag{18}$$

$$d = c - \texttt{partial\_stereo\_bottom\_coverage\_in\_degrees}. \tag{19}$$

In the regions $a <= \phi < b$ and $c <= \phi < d$ the exact value of $k$ is undefined. We only guarantee that in the region $a <= \phi < b$ it monotonically increases from 0 to 1 with increasing $\phi$ and in the region $c <= \phi < d$ it monotonically decreases from 1 to 0 with increasing $\phi$.

### 6.2. Projecting from a 3D Point to the Panorama

To project a point $[x, y, z]$ into the panorama use the following equations:

$$\theta_{\text{left}} = \operatorname{atan2}(x, z) + \sin^{-1}\left(\frac{kr}{\sqrt{x^2 + z^2}}\right) \tag{20}$$

$$\theta_{\text{right}} = \operatorname{atan2}(x, z) - \sin^{-1}\left(\frac{kr}{\sqrt{x^2 + z^2}}\right) \tag{21}$$

$$\phi = \frac{\pi}{2} + \tan^{-1}\left(\frac{y}{\sqrt{x^2 + z^2 - k^2 r^2} - k\delta}\right) \tag{22}$$

where $k$ is defined in equation 15. When projecting an arbitrary 3D point we don't know what the correct value of $k$ should be (since it depends on $\phi$). This means that we must solve for $k$ at the same time as $\phi$. One way to do this is to initially assume a value of $k = 1$ and if the resulting value of $\phi$ is not consistent with equation 15 then iteratively update $k$.

### 6.3. Disparity in the Panorama

It is possible to estimate disparity in the panorama (the horizontal distance in pixels between corresponding points in the left and right eye panoramas) directly from the depth value for a pixel:

$$\text{disparity} = \frac{W}{2\pi}\left(\pi - 2\tan^{-1}\left(\frac{k\delta + d\sin(\phi)}{kr}\right)\right) \tag{23}$$

where $d$ is the depth associated with the pixel in meters, $W$ is the width of the panorama in pixels and $k$ is defined in equation 15.

## 7. Depth Maps

Figure 4: Example color and depth frames output from Jump Assembler.

The depth maps produced by the Jump Assembler are in the same format as the stitches, over-under equirectangular in which the left eye is on top. The depth stitches use an inverse depth encoding to minimize quantization artifacts and the metric depth along a ray can be found from the greyscale values (in the range 0 to 1) as follows:

$$\texttt{depth\_in\_meters} = \frac{\texttt{minimum\_encodable\_depth\_in\_meters}}{\texttt{value\_in\_depth\_stitch}} \tag{24}$$

The value of `minimum_encodable_depth_in_meters` can be found in the stitch metadata.

Using the projection described in Section 6 `depth_in_meters` can then be used to generate a 3D point for each pixel in the stitch.

## A. Example Stitch Metadata

An abbreviated example of a sample stitch metadata file is shown below


```
{
  "major_version_number": 1,
  "minor_version_number": 1,
  "rig_type": "Halo",
  "rig_id": "ATSaZ63HTME",
  "interpupillary_distance_in_meters": 0.05,
  "missing_bottom_coverage_in_degrees": 32.5,
  "missing_top_coverage_in_degrees": 0,
  "minimum_encodable_depth_in_meters": 0.3,
  "distortion_delta_in_meters": 0.1479019945774904,
  "frames_per_second": 59.940059940059939,
  "mono_bottom_coverage_in_degrees": 0,
  "partial_stereo_bottom_coverage_in_degrees": 0,
  "missing_bottom_depth_coverage_in_degrees": 32.5,
  "mono_top_coverage_in_degrees": 5,
  "partial_stereo_top_coverage_in_degrees": 25,
  "missing_top_depth_coverage_in_degrees": 30,
  "cameras": [ {
    "name": "camera01",
    "position": [ 0, -0, 0.14999999999999999 ],
    "orientation": [ 0.0067773935363294744, 0.02128360697196437, 1.5790957390209466 ],
    "projection_type": "fisheye",
    "image_size": [ 1920, 1440 ],
    "principal_point": [ 968.30373943599193, 704.98857987122119 ],
    "radial_distortion": [ 0.096251462704779911, -0.016483790420092587 ],
    "focal_length": 834.50935697536784
  }, {
    "name": "camera02",
    "position": [ 0.057402514854763463, -0, 0.13858192987669302 ],
    "orientation": [ 0.31680907223631799, -0.28573457838845956, 1.5570910398280071 ],
    "projection_type": "fisheye",
    "image_size": [ 1920, 1440 ],
    "principal_point": [ 959.49326826870185, 698.67798012008973 ],
    "radial_distortion": [ 0.098776848879089912, -0.017666832605844663 ],
    "focal_length": 834.11484482529715
  }, {
...
  } ]
}
```

## B. Version History

- Version 1.0 September 14, 2016.
- Version 1.1 November 8, 2017.
  - Added parameters which define a transition to a monoscopic stitch at the poles.
  - Added details on projecting 3D points into the panorama and on estimating disparity in the panorama.
- Version 1.2 April 9, 2018.
  - Added `stitching_algorithm` field.