TECHNICOLOR LIGHT-FIELD DATASET
USER GUIDE FOR UNDISTORTED CONTENTS
GETTING STARTED WITH CALIBRATION PARAMETERS

Contact:
neus.sabater@technicolor.com
guillaume.boisson@technicolor.com

technicolor
FEEL THE WONDER

CITING THE TECHNICOLOR LIGHT-FIELD DATASET

Dataset and Pipeline for Multi-View Light-Field Video. N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, T. Langlois, R. Gendrot, O. Bureller, A. Schubert, V. Allie. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

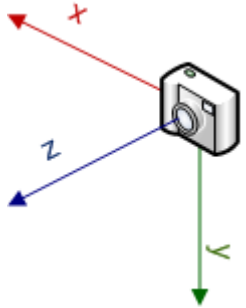
```
@InProceedings{Sabater_2017_CVPR_Workshops,
Title = {Dataset and Pipeline for Multi-View Light-Field Video},
Author = {Sabater, Neus and Boisson, Guillaume and Vandame, Benoit
and Kerbiriou, Paul and Babon, Frederic and Hog, Matthieu and
Langlois, Tristan and Gendrot, Remy and Bureller, Olivier and
Schubert, Arno, and Allie, Valerie},
booktitle = {The IEEE Conference on Computer Vision and Pattern
Recognition (CVPR) Workshops},
year = {2017}
}
```


technicolor
FEEL THE WONDER

Conventions and notations

Camera Coordinate System (CS):

- x rightward
- y downward
- $z > 0$ in the scene



Homogeneous vectors and perspective projection:

$$\begin{pmatrix} s \\ t \\ 1 \end{pmatrix} \equiv \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \Leftrightarrow \quad \begin{cases} s = x/z \\ t = y/z \end{cases}$$

technicolor
FEEL THE WONDER

Calibration parameters for undistorted contents

File `cameras_parameters.txt`

- 12 parameters for original contents
  - 5 intrinsic parameters
  - 7 extrinsic parameters (quaternion + translation)

- Example

```
# f cu cv ar sk  qw qx qy qz  tx ty tz
2354.05 1020.15 486.68 1. 0.  0.987 0.152 0.007 -0.028  0.1075 0.5194 1.7274
…
```


technicolor
FEEL THE WONDER

Intrinsic parameters

```
# f cu cv ar sk  qw qx qy qz  tx ty tz
2354.05 1020.15 486.68 1. 0.  0.987 0.152 0.007 -0.028  0.1075 0.5194 1.7274
```

- f: focal length (in pixels)
- (c_u, c_v): principal point coordinates
- ar: aspect ratio
- sk: skew coefficient

Intrinsic matrix

$$\mathbf{K} = \begin{pmatrix} f & sk & c_u \\ 0 & ar.f & c_v \\ 0 & 0 & 1 \end{pmatrix}$$

technicolor
FEEL THE WONDER

Extrinsic parameters

```
# f cu cv ar sk  qw qx qy qz  tx ty tz
2354.05 1020.15 486.68 1. 0.  0.987 0.152 0.007 -0.028  0.1075 0.5194 1.7274
```

- Let $\mathbf{R} \in \mathbb{R}^{3\times3}$ be the rotation matrix that derives from the unit quaternion

$$\boldsymbol{q} = q_w + i.q_x + j.q_y + k.q_z$$

  - See appendix

- Let $\mathbf{T}$ denote the translation $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$

Extrinsic matrix

$$\mathbf{Q} = (\mathbf{R} \quad \mathbf{T}) \in \mathbb{R}^{3\times4}$$

technicolor
FEEL THE WONDER

Extrinsic parameters & Pose matrix

```
# f cu cv ar sk  qw qx qy qz  tx ty tz
2354.05 1020.15 486.68 1. 0.  0.987 0.152 0.007 -0.028  0.1075 0.5194 1.7274
```

- Let $\mathbf{R} \in \mathbb{R}^{3\times3}$ be the rotation matrix that derives from the unit quaternion

$$\boldsymbol{q} = q_w + i.q_x + j.q_y + k.q_z$$

  - See appendix

- Let $\mathbf{T}$ denote the translation $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$

Pose matrix

$$\mathbf{P} = (\mathbf{R}^t \quad -\mathbf{R}^t.\mathbf{T}) \in \mathbb{R}^{3\times4}$$

technicolor
FEEL THE WONDER

Usage examples

technicolor
FEEL THE WONDER

From a coordinate system to another

Pose and extrinsic matrices are transition matrices

Let $\mathbf{X}_{world}$ and $\mathbf{X}_{cam}$ be the coordinates of a 3D point respectively in the World CS and in the CS of the camera:

$$\mathbf{X}_{cam} = \mathbf{Q}.\begin{pmatrix}\mathbf{X}_{world}\\1\end{pmatrix} \quad \text{and} \quad \mathbf{X}_{world} = \mathbf{P}.\begin{pmatrix}\mathbf{X}_{cam}\\1\end{pmatrix}$$

technicolor
FEEL THE WONDER

Projection equation

From World coordinate system to image (pixel) coordinates:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \equiv \mathbf{K}.\mathbf{Q}.\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$

technicolor
FEEL THE WONDER

3D reconstruction

From a pixel $\begin{pmatrix} u \\ v \end{pmatrix}$ to World CS, assuming a depth z in the CS of the camera:

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \mathbf{P}.\begin{pmatrix} z.\mathbf{K}^{-1}.\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \\ 1 \end{pmatrix}$$

technicolor
FEEL THE WONDER

Appendix

technicolor
FEEL THE WONDER

Deriving a rotation matrix from a quaternion

See e.g. https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation#Quaternion-derived_rotation_matrix

Resources

- Eigen C++ library: method `toRotationMatrix()` http://eigen.tuxfamily.org/dox-3.2/classEigen_1_1QuaternionBase.html
- MATLAB: function `quat2rotm()` https://fr.mathworks.com/help/robotics/ref/quat2rotm.html

technicolor
FEEL THE WONDER