FTV360: a Multiview 360°Video Dataset with Calibration Parameters
Thomas Maugey, Laurent Guillo, Cédric Le Cam

To cite this version:

HAL Id: hal-02398005
https://inria.hal.science/hal-02398005
Submitted on 6 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
FTV360: a Multiview 360° Video Dataset with Calibration Parameters

Thomas Maugé
Inria Rennes Bretagne Atlantique
Rennes, France
thomas.maugey@inria.fr

Laurent Guillo
CNRS, IRISA
Rennes, France
laurent.guillo@irisa.fr

Cédric Le Cam
Inria Rennes Bretagne Atlantique
Rennes, France

ABSTRACT

In this paper, we present a new dataset in order to support researches in Free Viewpoint Television (FTV) and 6 degrees-of-freedom (6DoF) immersive communication. This dataset relies on a novel acquisition procedure consisting in a synchronized capture of a scene by 40 omnidirectional cameras. We have also developed a calibration solution that estimates the position and orientation of each camera with respect to a same reference. This solution relies on a regular calibration of each individual camera, and on a graph-based synchronization of all these parameters. These videos and the calibration solution are made publicly available.

CCS CONCEPTS
- • Information systems → Multimedia databases; • Computing methodologies → Camera calibration.

KEYWORDS
Datasets, 6DoF, Free Viewpoint Television, 360°, Calibration

1 INTRODUCTION

Free Viewpoint Television (FTV) is an emerging application in which a multimedia content is transmitted to users such as they are enabled to choose and change the viewing angle in real time [11]. In other words, users have the possibility to observe the scene from the viewpoint they want: either by choosing between a predefined set of views (Fig. 1(a)), or by freely navigating in the scene (Fig. 1(b)) and thus experiencing 6 degrees of freedom (6DoF).

The target applications are numerous: from the transmission of sportive and cultural events for television, to more immersive communication with augmented reality applications as the "industry 4.0", education and health.

Ultimately, FTV with 6DoF is a challenging scenario that faces multiple research issues, dealing with the whole processing chain.

The data representation task consists in combining the signal captured by multiple heterogeneous devices (e.g., light field, 360, stereo cameras) and finding a proper description well adapted to the transmission constraints. Different compact representations have been investigated such as LDI [6, 10], GBR [7], mesh [1], point-clouds [3]. Then a dedicated compression scheme has to be designed. Contrary to "traditional" compression in which all the data is sent to the user, the specificity of FTV is that the transmission system has to transmit only what is required by the user. Indeed, it is not necessary to transmit all the scene when the user is only looking at one specific subpart of it. However, since online encoding is not conceivable, the main problem is to design a compression strategy that encodes all the data a priori such that only a subset could be extracted after the user request [4, 9]. At the user’s side, efficient rendering algorithms are needed to make the navigation as smooth as possible, by artificially increasing the number of views.

Although developing competitive representation, compression and rendering solutions is already highly challenging, making the user experiencing a real 6 degrees of freedom (6DoF) navigation in the scene is nowadays impossible mostly because no proper acquisition system exists. Indeed, the acquisition for FTV presents several important issues. In order to enable a high quality navigation, it is required that the user has access to a number of viewpoints sufficiently large to cover a complete wide 3D scene. Even though some of them might be virtually synthesized, the initial capture must be sufficiently dense to enable good synthesis quality. This makes the acquisition process very costly in terms of hardware. Another issue is that the calibration algorithms used for small captured systems have to be revisited since the cameras may be more distant in a FTV context. Nowadays, datasets are either totally synthetic or only enable 3DoF navigation (i.e., only the rotation of the head and a small 3D sensation).

In this paper, we propose a novel dataset made of 8 sequences (each of them contains 40 synchronized and calibrated 360° videos). This enables a 3DoF navigation at many different places in the scene, which comes down to a "discretized 6DoF" solution. We also
User is able to make discrete translation in the scene
At every discrete camera position, the user is able to watch every direction

Figure 2: Proposed acquisition architecture for Free Viewpoint Acquisition.

Figure 3: The used 360° cameras capture two hemispherical images leading to an image at a resolution of 3840 × 1920.

2 ACQUISITION PROCEDURE
Ultimate Free Viewpoint Navigation enables a user to freely change the position, \( t = [x, y, z] \in \mathbb{R}^3 \), and the angle, \( r = [\alpha, \beta, \gamma] \in [-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi] \) of his viewpoint. Naturally, it is impossible in practice to sample at every position the light rays coming from every direction. The challenge for an acquisition system is yet to make this sampling as dense as possible. For that purpose, perspective cameras have shown their limitation since each of them captures the light rays at one given position coming from one fixed subset of directions. Recently, omnidirectional (or 360°) cameras have been introduced in the public market. Their strength is that they are able to record the light rays at one given position coming from every direction, at a price of a possibly decreased angular resolution since the field of view is increased.

This has motivated the following proposed acquisition procedure: 40 omnidirectional cameras have been spread inside a scene and synchronously film its content. If a user is navigating through the recorded video, he/she is able to discretely translates in the scene, i.e., \( t \in \{\delta_i\} \), and at each translation position, he/she is able to visualize the angle he/she desires, i.e., \( \forall i, r(\delta_i) \in [-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi] \) (see Fig. 2). Such acquisition system has never been studied and seems promising since it gets closer to full 6DoF navigation. As such, it is considered to constitute an interesting starting point for further research.

In order to keep the data representation as close as possible to what is recorded by the camera (for calibration purposes), we keep the raw data format. In our solution, the videos are shot with Samsung Gear 360 cameras. They are made of two fisheye lenses spanning a bit more than 180° and placed at a distance of 4.5 cm. The raw footage consists of the image captured by the two lenses, without any geometrical correction, written side by side on the same frame. The resolution of the two hemispherical images put side by side is 3840 × 1920 at 30 fps, as shown by the example in Fig. 3. All the videos are synchronized manually at the frame precision. The calibration procedure that was used to estimate the relative position of each camera is explained in the next section.

3 CALIBRATION
3.1 Internal parameters estimation
The specificity of omnidirectional cameras is that their projection rules strongly depend on their architecture. As an example, catadioptric cameras (using a curved mirror to make the vision omnidirectional) do not project a light ray on its sensor as a fisheye lens does. In [2, 8], a general projection model has been proposed: the so-called unified Spherical Model. It is illustrated in Fig. 4. Let \( P = [X, Y, Z]^\top \) be a point in the 3D world expressed with respect to a world coordinate system centered in \( O \). The unified spherical
As mentioned in [8], the unified spherical model is sometimes
implemented in the OpenCV library. In Table 1, we show that the
unified spherical model gives good results for our fisheye cameras.

\begin{align}
P_s &= \frac{p}{\|p\|}.
\end{align}

Let us now define a point $O_s$, that is the translation of $O$ along the $Z$
axis. The distance between $O$ and its translation $O_s$ is denoted by $\xi$.
The unified spherical model then performs a second projection from
the sphere to a plane that is parallel to the $X$ and $Y$ axis, and placed
at a distance $f$ from $O_s$. This second projection is a perspective
projection, as in the traditional pinhole cameras. The projection
$p = [x, y]$ of point $P_s$ onto the sensor array reads:
\begin{align}
\begin{cases}
x = f \frac{X}{Z + \xi} \\
y = f \frac{Y}{Z + \xi}
\end{cases}
\end{align}

As a result, the pixel coordinates in the image plane can be written:
\begin{align}
\begin{cases}
u_x = k_x \left( \frac{fX}{Z + \xi \sqrt{X^2 + Y^2 + Z^2}} \right) + k_x x_0 \\
u_y = k_y \left( \frac{fY}{Z + \xi \sqrt{X^2 + Y^2 + Z^2}} \right) + k_y y_0
\end{cases}
\end{align}

As mentioned in [8], the unified spherical model is sometimes
completed with a radial and tangent distortions terms parameterized
with 5 parameters ($k_1, \ldots, k_5$). Here, we neglect these parameters
(i.e., $k_1 = \ldots = k_5 = 0$) since they are not necessary to model the
projection properly in our system. A calibration algorithm aims at
estimating the projection parameters $k_x, k_y, f, \xi, x_0, y_0$. For
this paper, we have used the algorithm proposed in [8] and that is
implemented in the OpenCV library. In Table 1, we show that the
unified spherical model gives good results for our fisheye cameras.
We also show that the same parameters can be used for every lens
of all cameras. We precise that, in the following, the two lenses are
treated separately, and thus considered as two different cameras
(each one have its own set of parameters).

### 3.2 Individual external parameters estimation

In addition to the intrinsic values, the output of the calibration
algorithm mentioned in the previous section contains the extrinsic
parameters: a rotation and a position vector, respectively denoted
by $t$ and $r$. They describe the position of the camera with respect
to the pattern that is recorded. We thus record long calibration
sequences where a pattern is synchronously filmed by all cameras.
The pattern used in our experiments is a chessboard. To run the
calibration of each camera we select the frames in which the pattern
is visible. The output of such joint calibrations are:
- A time instant set $I$ giving the frame timestamps during
  which the pattern is visible in at least one camera.
- For each, camera $i$, a time instant set $I_i \subset I$ corresponding to
  the time instants during which the pattern is visible in camera $i$.
- A set of positions $\{t_{n_i}\}$ and rotations $\{r_{n_i}\}$, where $n \in [1, |I|]$ and
each $t_{n_i} \in \mathbb{R}^3$ and each $r_{n_i} \in \mathbb{R}^3$ are sorted in the time
instant order.

The rotation angles $r$ are given in the Rodrigues format [5].
The vector $r$ gives the rotation axis, and $\|r\|$ is rotation angle. The
remaining step consists in gathering all these asynchronous relative
positioning in order to deduce the position of all cameras in a fixed
world coordinate system. For that aim, we propose an algorithm
that is described in the next section.

### 3.3 Global external parameters estimation

The challenge of this part is to gather all the parameters, relative
to the moving chessboard pattern, and to deduce camera parameters.
We first compute a correspondence matrix counting the co-
ocurrence of cameras in the pattern detection. Concretely, for two
cameras $i$ and $j$, the matrix value is equal to:
\begin{align}
G(i, j) = |I_i \cap I_j|.
\end{align}

The matrix element $G(i, j)$ indicates how much the joint calibration
of two cameras $i$ and $j$ is reliable. Indeed, the calibration gives better
results if more points are used in the optimization. We then choose
the reference camera $i_{\text{ref}}$, from which all the other camera positions

### Table 1: Example of MSE computed between the pixel position measured and the ones estimated thanks to the unified spherical model parameters, for a camera specific parameters (cam) and averaged camera parameters (av.). This has been tested on all our cameras.

<table>
<thead>
<tr>
<th>Cam. 1 front</th>
<th>Cam. 1 rear</th>
<th>Cam. 2 front</th>
<th>Cam. 2 rear</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cam)</td>
<td>0.85</td>
<td>1.11</td>
<td>1.36</td>
</tr>
<tr>
<td>(av.)</td>
<td>1.37</td>
<td>1.45</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Unified spherical model adopted for the calibration of the camera used during our acquisition.
will be set. We pick the most reliable one:

\[ i_{\text{ref}} = \arg\max_i \sum_j G(i, j). \]  

(5)

The camera \( i_{\text{ref}} \) is thus the camera having the highest number of simultaneous calibration parameters with the other cameras.

The next step consists in setting: \( t^{\text{ref}} = 0 \) and \( t^i = 0 \). Let us now denote by \( N(t_{\text{ref}}) \) the neighborhood of camera \( i_{\text{ref}} \), i.e. the set of cameras that have joint calibration parameters. We denote a coordinate change as:

\[
\phi_{i \rightarrow j} : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3 \\
\phi_{i \rightarrow j}(r^i, t^i, r^j, t^j) \rightarrow r^{ij}, t^{ij}
\]  

(6)

where \( r^{ij} \) and \( t^{ij} \) are the coordinates of camera \( i \) expressed in \( j \) coordinate system. We now do the following operation

\[
\forall i \in N(t_{\text{ref}}), (r^{i | \text{loc}}, t^{i | \text{loc}}) = \\
\sum_{n \in I \cap I_{\text{ref}}} \phi_{i \rightarrow i_{\text{ref}}}(r^n, t^n, r^i_{\text{ref}}, t^i_{\text{ref}})
\]  

(7)

In other words, we average all the joint estimated positions converted to the \( i_{\text{ref}} \) coordinate system. Then the algorithm repeats this for the most reliable camera in this neighborhood. It seeks its own neighborhood and estimate this position, converted to the \( i_{\text{ref}} \) coordinate system. These operations are repeated until one has an estimate of all the positions with respect to a single camera \( i_{\text{ref}} \).

### 3.4 Calibration validation

In this section, we show that the proposed calibration strategy gives meaningful distance and orientation estimation for the cameras. In a first quantitative test, we position two cameras at different distances (measured with a laser), and we compare the results of the calibration algorithm. In Table 2, we show the distance evaluation remains close to the true one, even when the baseline between cameras is large (5 m).

<table>
<thead>
<tr>
<th>( \Delta x ) real (m)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std. Dev. of estimated ( \Delta x ) (m)</td>
<td>0.02</td>
<td>0.03</td>
<td>0.07</td>
<td>0.03</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 2: Standard deviation between \( \Delta x \) estimated as a function of the true \( \Delta x \) between the two cameras.

Then, we present qualitative results in which, we show the estimated cameras positions, when 40 of them are positioned in a scene. The distance between the cameras are typically between 2 and 3 m. Fig. 5 illustrates the estimated camera arrangement that was used during a capture. The algorithm properly retrieves the organization of the capture system and the relative distance and orientation of the cameras.

![Figure 5: Example of estimated camera position.](image)

### 4 DESCRIPTION OF THE DATASET

Based on these developed tools, we have built a complete dataset that we share on the following website:

https://project.inria.fr/ftv360.

The dataset is made of several **Captures** including the following steps. (i) We position 40 omnidirectional cameras in a scene. Their distance with the neighboring ones lies between 1m and 3m. (ii) We record one or several calibration sequences, in which a chessboard pattern is moving in the scene. The recorded videos are then used to estimate the calibration parameters with our proposed algorithm. (iii) We record several **Sequences** with the same camera arrangement (and thus the same calibration parameters). In each sequence, a scene (1 min to 4 min) is acquired by all the synchronized cameras. Our dataset is made of three different captures, with, in total 8 different sequences (each of them having 39 or 40 synchronized videos). The structure of the shared dataset is depicted in Fig. 6. We precise that the camera parameters include extrinsic and intrinsic parameters. The extrinsic parameters are given both in the Euler and Rodrigues formats. A complete webpage of the site is dedicated to an explication of what these parameters mean. We also share, for each of the capture, a map of the camera positioning in the scene. Finally, the calibration toolkit is also made available on the project website.

### 5 CONCLUSION AND FURTHER RESEARCHES

In this paper, we propose a new dataset for supporting researches on FTV. For that purpose, a new acquisition system and a complete calibration solution have been developed. The 8 videos sequences along with the calibration toolkit are made publicly available on [https://project.inria.fr/ftv360](https://project.inria.fr/ftv360).

These data can serve for the development of new tools for FTV. In particular, virtual view synthesis algorithms are the ultimate step before real 6DoF, where a user could smoothly navigate in the scene. In the same spirit researches on depth estimation, super resolution and inpainting could benefit from this innovative dataset especially because of the original use of multiple omnidirectional cameras. Finally, researchers working on interactive video compression will, for the first time, be able to test their new solutions on a meaningful dataset, i.e., a high number of viewpoints spread in a large scene.

### REFERENCES


---

1[https://project.inria.fr/ftv360/informations/calibration-parameters/](https://project.inria.fr/ftv360/informations/calibration-parameters/)
FTV360: a Multiview 360° Video Dataset with Calibration Parameters

Figure 6: Structure of the FTV360 dataset.


