Real-Time Distance-Dependent Mapping for a Hybrid ToF Multi-Camera Rig

Frederic Garcia*, *Member*, *IEEE*, Djamila Aouada*, *Member*, *IEEE*, Bruno Mirbach‡, and Björn Ottersten*, *Fellow*, *IEEE*

*Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
{frederic.garcia, djamila.aouada, bjorn.ottersten}@uni.lu
‡Advanced Engineering Department, IEE S.A.
bruno.mirbach@iee.lu

***Abstract*—We propose a real-time mapping procedure for data matching to deal with hybrid ToF multi-camera rig data fusion. Our approach takes advantage of the depth information provided by the ToF camera to calculate the distance-dependent disparity between the two cameras that constitute the system. As a consequence, the not co-centric binocular system behaves as a co-centric system with co-linear optical axes between their sensors. The association between mapped and non-mapped image coordinates can be described by a set of look-up tables. This, in turn, reduces the complexity of the whole process to a simple indexing step, and thus, performs in real-time. The experimental results show that in addition to being straightforward and easy to compute, our proposed data matching approach is highly accurate which facilitates further fusion operations.**

***Index Terms*—Time of flight, sensor fusion, 3D data fusion, multimodal sensors, multi-sensor systems, data matching, mapping.**

I. Introduction

TIME-OF-FLIGHT cameras are relatively new 3-D sensors that promise to be an alternative to other 3-D sensing systems such as stereo vision systems, laser scanners or structured light. They present several advantages such as simultaneously providing intensity and depth information for every pixel at a high frame rate. Moreover, the recent advances in industrializing and producing economic, compact, robust to light and illumination changes ToF cameras is starting to have an impact on commercial applications [1], [2]. However, ToF cameras and especially the industrial ones, cannot yet attain the resolution and precision of alternative 3-D sensing systems. Indeed, two main drawbacks are currently restricting the use of ToF cameras in a wide range of computer vision and robotics applications; namely, the noise within depth measurements and the low resolution of the given depth maps. The low resolution problem is even more prominent in industrial ToF cameras as a compromise for their higher robustness to ambient conditions, *i.e.*, larger working temperature range and higher reliability under sun lighting. We note that in contrast to 3-D sensing devices intended for gaming applications or research purposes such as Microsoft's Kinect camera [3], ToF cameras that are used for automotive applications or applications in industrial automation have resolutions lower than (64×64) pixels. Therefore, in applications where the limited resolution of a ToF camera is critical, a very promising strategy is sensor fusion [4], [5], [6], [7], [8], [9], [10], [11], *i.e.*, combining ToF data with data provided by other sensors, usually 2-D colour cameras [12]. Indeed, current research efforts in ToF and 2-D data fusion have proven to deliver dense depth maps at near real-time frame rates, outperforming, in some cases, alternative 3-D sensing systems [4], [6]. We talk about a low-level data fusion in contrast to higher fusion levels in which the fusion deals with post-processed data (feature or decision level fusions) [13]. While it is true that these low-level fusion algorithms can now perform in real-time, they all assume a perfect alignment of the data to be fused, which is far from a trivial task for most real-world data and scenarios. In fact, forward warping techniques [14], [15], that match the distance measurements from the ToF camera onto the colour camera are straightforward procedures which lead to the assignment of a colour value to each of the (low-resolution) ToF pixels. However, we herein propose to tackle just the opposite case, *i.e.*, backward warping. Our objective is to assign to each high-resolution 2-D pixel an accurate distance value. This requires mapping the 2-D image onto the ToF image, which is not straightforward if one has to take into account the distance dependency of the disparity. Furthermore, such dependency on the distance requires to recompute the whole mapping procedure for each recorded frame and thus, it makes the real-time implementation quite challenging.

In this paper, we propose an original framework to align the data recorded by each of the cameras that constitute the hybrid ToF multi-camera rig. We note that our method is not only intended to map the data from low-resolution ToF cameras but conceptually applies also to other 3-D sensing modalities such as the recently emerging ToF laser scanners, *i.e.*, the ibeo LUX [16] or the Eco Scan FX8 [17] whose resolutions are also far below the resolutions of standard 2-D cameras. In contrast to stereo vision approaches, we use the distance information provided by the 3-D sensor to calculate the distance-dependent disparity that relates each of the devices that constitute the hybrid ToF multi-camera rig. Our mapping procedure yields to an accurate data matching that facilitates further data fusion techniques to overcome the above mentioned ToF drawbacks. Furthermore, we represent the relationship between non-mapped and mapped image co-

ordinates by an associative array, *i.e.*, look-up table, reducing the complexity of the whole mapping procedure to a simple indexation step and thus, enabling for a real-time performance.

The outline of this paper is as follows: In Section II, we introduce the distance-dependent disparity that is related to the field of view (FOV) effect in hybrid ToF multi-camera rig. In Section III, we describe our technique to map the image coordinates relative to each camera to a unified coordinate frame and propose, in Section IV, the data matching procedure. In Section V, we illustrate and quantify the results of our proposed technique. Finally, we give our conclusions and perspectives in Section VI.

II. Distance-dependent disparity

A hybrid ToF multi-camera rig provides multi-modal data related to the camera reference frame from which the data has been recorded. In our case, our hybrid ToF multi-camera rig is composed by a conventional 2-D camera, with a reference frame $\mathcal{A}$, and an industrialized ToF camera with a reference frame $\mathcal{B}$. In general, the two reference frames $\mathcal{A}$ and $\mathcal{B}$ are not co-centric, *i.e.*, the two cameras are displaced with respect to each other by a distance between the centres of projection, $\mathbf{O}_{\mathcal{A}}$ and $\mathbf{O}_{\mathcal{B}}$, respectively. This distance is known as the *baseline* b of the hybrid ToF multi-camera rig. The distance Z at which a point $\mathbf{P}$ is located with respect to the baseline b is obtained from the similar triangles ($\mathbf{p}_{\mathcal{A}}$,$\mathbf{P}$,$\mathbf{p}_{\mathcal{B}}$) and ($\mathbf{O}_{\mathcal{A}}$,$\mathbf{P}$,$\mathbf{O}_{\mathcal{B}}$) such that

$$\frac{b + x_{\mathcal{A}} - x_{\mathcal{B}}}{Z - f} = \frac{b}{Z}, \tag{1}$$

where $x_{\mathcal{A}}$ and $x_{\mathcal{B}}$ are the coordinates of the projections $\mathbf{p}_{\mathcal{A}}$ and $\mathbf{p}_{\mathcal{B}}$ with respect to the principal points $c_{\mathcal{A}}$ and $c_{\mathcal{B}}$, and f is the common focal length. Solving (1) for Z, we obtain

$$Z = f\frac{b}{\rho}, \tag{2}$$

where $\rho = x_{\mathcal{A}} - x_{\mathcal{B}}$, the *binocular disparity*, measures the difference in retinal position between the corresponding points in the two images. In stereo systems, the disparity leads to the estimation of the distance Z. However, this requires the detection of the projections $\mathbf{p}_{\mathcal{A}}$ and $\mathbf{p}_{\mathcal{B}}$ which relates to the well-known *correspondence problem* [18], which is typically performed by feature matching or correlation analysis and thus, numerically demanding and suffering from shadow effects or texture patterns. In contrast, we tackle the opposite case by the use of the ToF camera as it provides the distance with respect to its own reference frame $\mathcal{B}$, *i.e.*, $Z_{\mathcal{B}}$, at which each point is located within the given depth maps. This allows us to estimate the disparity $\rho(Z_{\mathcal{B}})$ for each of the ToF camera pixels, which simplifies the mapping by avoiding demanding operations such as feature matching and image correlation (see Section III).

We note that the relationship between the $Z_{\mathcal{B}}$ measurements and the disparity $\rho(Z_{\mathcal{B}})$ causes a dependency on the scene. Therefore, it has to be recalculated whenever the scene changes, which is typically the case for every frame of data acquisition, and for each ToF camera pixel as it is not constant for all pixel locations. By differentiating disparity $\rho(Z_{\mathcal{B}})$ in (2) with respect to the distance $Z_{\mathcal{B}}$, we define the absolute disparity variation $\Delta\rho(Z_{\mathcal{B}})$ as a function of the absolute depth variation $\Delta Z_{\mathcal{B}}$, and obtain

$$\Delta\rho(Z_{\mathcal{B}}) = f_{\mathcal{B}} b \frac{\Delta Z_{\mathcal{B}}}{Z_{\mathcal{B}}^2}, \tag{3}$$

where $f_{\mathcal{B}}$ is the focal length of the ToF camera. We note that only in situations where the depth variation of the object in the scene $\Delta Z_{\mathcal{B}}$ is small enough compared to the squared distance $Z_{\mathcal{B}}^2$ from the object to the system, the disparity $\rho(Z_{\mathcal{B}})$ can be assumed as constant and thus, included in a simple projective transformation for all recorded frames. Actually, this scenario is commonly used in research efforts that integrate non-industrial ToF cameras such as the SwissRanger™ ToF camera, in their ToF multi-camera rig [4], [19], [20]. In this case, the rather small FOV provided by the SwissRanger™ camera, *i.e.*, $47.5° \times 39.6°$, forces such systems to be installed at a relatively large distance from the object. As a consequence, these systems can still function while neglecting the distance-dependent disparity, which is not the case for the majority of ToF cameras, which require the variation of disparity to be taken into account. In what follows we propose to solve this problem by defining a new matching procedure that exploits the distance-dependent disparity. As a result, any ToF camera available on the market can be integrated in a hybrid ToF multi-camera rig intended for low-level data fusion regardless of its specifications.

III. Unified reference frame for a hybrid ToF multi-camera rig

We denote the intensity images given by the 2-D camera as $\mathbf{I}$ with image coordinates $(u^{\mathbf{I}}_{\mathcal{A}}, v^{\mathbf{I}}_{\mathcal{A}})$. Similarly, we denote the low-resolution depth maps given by the ToF camera as $\mathbf{D}$ with image coordinates $(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}})$. In the following we present the transformation or mapping of these image coordinates to a unified reference frame $\mathcal{C}$, which is the basis for the data matching (or warping) described in Section IV.

A. Background: image coordinate transformation

Let $\mathbf{P} = [X, Y, Z]^{\mathrm{T}}$ be a point related to the camera reference frame. Its projection on the camera image frame results in a new point $\mathbf{p} = [x, y, z]^{\mathrm{T}}$ defined as follows:

$$\mathbf{p} = \mathbf{K} \cdot \mathbf{P} \Leftrightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{4}$$

with $\mathbf{K}$ the matrix of intrinsic camera parameters defined as follows:

$$\begin{aligned} \mathbf{K} &= \mathbf{K_s}\mathbf{K_f} \\ &= \begin{bmatrix} \delta_x^{-1} & 0 & c_x \\ 0 & \delta_y^{-1} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \delta_x^{-1} f & 0 & c_x \\ 0 & \delta_x^{-1} f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{5}$$

where f is the focal length, δ_x and δ_y are the effective horizontal, and respectively vertical pixel size, and (c_x, c_y) is

the position of the optical axis or principal point in the image (all units are in millimetres). The image coordinates in pixels, *i.e.*, $\mathbf{p}' = [u, v, 1]^{\mathrm{T}}$ are given by:

$$\begin{cases} u = x/z \quad \Rightarrow \quad u = x/Z, \\ v = y/z \quad \Rightarrow \quad v = y/Z, \end{cases} \tag{6}$$

as $z = Z$ from (4). We note that $\mathbf{p} = Z \cdot \mathbf{p}'$. From (4) and (6), and considering (5), the image coordinates (u, v) are defined as

$$\begin{cases} u = \delta_x^{-1} f \frac{X}{Z} + c_x, \\ v = \delta_y^{-1} f \frac{Y}{Z} + c_y. \end{cases} \tag{7}$$

In order to determine the coordinates of the point $\mathbf{P}$ with respect to a different camera reference frame $\mathcal{C}$, *i.e.*, $\mathbf{P}_{\mathcal{C}}$, an additional coordinate transformation must be considered. In general, this coordinate transformation is given by the extrinsic camera parameters, being a (3×3) rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t} = (t_x, t_y, t_z)^{\mathrm{T}}$, *i.e.*,

$$\mathbf{P}_{\mathcal{C}} = \mathbf{R}\Big[\mathbf{P} - \mathbf{t}\Big]. \tag{8}$$

Hence, from (4) and (8), the image coordinates $\mathbf{p}'_{\mathcal{C}} = [u_{\mathcal{C}}, v_{\mathcal{C}}, 1]^{\mathrm{T}}$ with respect to the reference frame $\mathcal{C}$ result from

$$\mathbf{p}'_{\mathcal{C}} = \frac{Z}{Z_{\mathcal{C}}} \mathbf{K}_{\mathcal{C}} \mathbf{R} \mathbf{K}^{-1} \Big[\mathbf{p}' - \frac{\mathbf{K}}{Z} \mathbf{t}\Big]. \tag{9}$$

B. Distortion correction

A necessary step to be completed before starting the image coordinates transformation is the correction of the distortion due to the camera lens. This is a classical step in system calibration that consists in correcting the raw distorted images $\mathbf{I}$ and $\mathbf{D}$ according to the intrinsic and extrinsic camera parameters that are to be determined. The research on ToF camera calibration is not yet extensive, but new insights have been proposed in [21], [22], [23]. We may resort to classical calibration tools such as Bouguet's toolbox for Matlab [24] or image processing tools such as those included in Intel's computer vision library *OpenCV* [25]. Once the camera parameters are known, we correct the distortion for the 2-D image coordinates and the ToF image coordinates and proceed with the resulting undistorted image coordinates $(u^{\mathbf{I}}_{\mathcal{A}}, v^{\mathbf{I}}_{\mathcal{A}})$ and $(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}})$, respectively.

C. Choice of the unified reference frame

The image coordinates $\mathbf{p'}_{\mathcal{A}} = [u^{\mathbf{I}}_{\mathcal{A}}, v^{\mathbf{I}}_{\mathcal{A}}, 1]^{\mathrm{T}}$ of a point $\mathbf{P}_{\mathcal{A}} = [X_{\mathcal{A}}, Y_{\mathcal{A}}, Z_{\mathcal{A}}]^{\mathrm{T}}$ related to the 2-D camera reference frame $\mathcal{A}$ are transformed to the unified reference frame $\mathcal{C}$ using (9), *i.e.*,

$$\mathbf{p}'_{\mathcal{C}} = \frac{Z_{\mathcal{A}}}{Z_{\mathcal{C}}} \mathbf{K}_{\mathcal{C}} \mathbf{R}_{\mathcal{A}\mathcal{C}} \mathbf{K}^{-1}_{\mathcal{A}} \Big[\mathbf{p}'_{\mathcal{A}} - \frac{\mathbf{K}_{\mathcal{A}}}{Z_{\mathcal{A}}} \mathbf{t}_{\mathcal{A}\mathcal{C}}\Big], \tag{10}$$

with $\mathbf{R}_{\mathcal{A}\mathcal{C}}$ and $\mathbf{t}_{\mathcal{A}\mathcal{C}}$ the rotation matrix and translation vector from the reference frame $\mathcal{A}$ to the reference frame $\mathcal{C}$, respectively. Since the image transformation in (10) requires the knowledge of the coordinate $Z_{\mathcal{A}}$, we choose the unified reference frame $\mathcal{C}$ to be co-centric to the 2-D camera reference frame $\mathcal{A}$, *i.e.*, $\mathbf{t}_{\mathcal{A}\mathcal{C}} = [0, 0, 0]^{\mathrm{T}}$. Hence, (10) amounts to

$$\mathbf{p}'_{\mathcal{C}} = \mathbf{K}_{\mathcal{C}} \mathbf{R}_{\mathcal{A}\mathcal{C}} \mathbf{K}^{-1}_{\mathcal{A}} \mathbf{p}'_{\mathcal{A}} =: \mathbf{H}_{\mathcal{A}\mathcal{C}} \cdot \mathbf{p}'_{\mathcal{A}}, \tag{11}$$

with $\mathbf{H}_{\mathcal{A}\mathcal{C}}$ a plane-to-plane transformation or projective transformation from reference frame $\mathcal{A}$ to reference frame $\mathcal{C}$. Similarly, the transformation of the image coordinates of a point $\mathbf{p}'_{\mathcal{B}}$, related to the ToF camera reference frame $\mathcal{B}$, to the unified reference frame $\mathcal{C}$ is analogous. Thus, using (9), we find

$$\begin{aligned} \mathbf{p}'_{\mathcal{C}} &= \frac{Z_{\mathcal{B}}}{Z_{\mathcal{C}}} \mathbf{K}_{\mathcal{C}} \mathbf{R}_{\mathcal{B}\mathcal{C}} \mathbf{K}^{-1}_{\mathcal{B}} \Big[\mathbf{p}'_{\mathcal{B}} - \frac{\mathbf{K}_{\mathcal{B}}}{Z_{\mathcal{B}}} \mathbf{t}_{\mathcal{B}\mathcal{C}}\Big] \\ &= \frac{Z_{\mathcal{B}}}{Z_{\mathcal{C}}} \mathbf{H}_{\mathcal{B}\mathcal{C}} \Big[\mathbf{p}'_{\mathcal{B}} - \frac{\mathbf{K}_{\mathcal{B}}}{Z_{\mathcal{B}}} \mathbf{t}_{\mathcal{B}\mathcal{C}}\Big], \end{aligned} \tag{12}$$

where $\mathbf{R}_{\mathcal{B}\mathcal{C}}$ and $\mathbf{t}_{\mathcal{B}\mathcal{C}}$ are the rotation matrix and the translation vector from the reference frame $\mathcal{B}$ to the reference frame $\mathcal{C}$, respectively. $\mathbf{H}_{\mathcal{B}\mathcal{C}}$ is the projective transformation from reference frame $\mathcal{B}$ to reference frame $\mathcal{C}$. We note that in this case the distance $Z_{\mathcal{B}}$ is known as it results from

$$Z_{\mathcal{B}} = \mathbf{D}(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}}) \cdot \frac{f_{\mathcal{B}}}{d(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}})}, \tag{13}$$

with

$$d(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}}) = \sqrt{f^2_{\mathcal{B}} + \big(\delta_{x,\mathcal{B}}(u^{\mathbf{D}}_{\mathcal{B}} - c_{x,\mathcal{B}})\big)^2 + \big(\delta_{y,\mathcal{B}}(v^{\mathbf{D}}_{\mathcal{B}} - c_{y,\mathcal{B}})\big)^2}, \tag{14}$$

and $\mathbf{D}$ being a radial measurement as acquired by the ToF camera. The conversion in (13) is therefore necessary to obtain the distance $Z_{\mathcal{B}}$ that relates to each pixel $(u^{\mathbf{D}}_{\mathcal{B}}, v^{\mathbf{D}}_{\mathcal{B}})$ in $\mathbf{D}$. This in turn allows the transformation of the image coordinates from the reference frame $\mathcal{B}$ to the reference frame $\mathcal{C}$.

D. Distance-dependent disparity shift

The transformation of the image coordinates in (12) consists of two steps. The first step concerns the binocular disparity shift, *i.e.*,

$$\mathbf{p}''_{\mathcal{B}} = \frac{Z_{\mathcal{B}}}{Z_{\mathcal{B}} + t_z} \Big[\mathbf{p}'_{\mathcal{B}} - \frac{\mathbf{K}_{\mathcal{B}}}{Z_{\mathcal{B}}} \mathbf{t}_{\mathcal{B}\mathcal{C}}\Big], \tag{15}$$

followed by the the projective transformation $\mathbf{p}'_{\mathcal{C}} = Z_{\mathcal{B}}/Z_{\mathcal{C}} \cdot \mathbf{H}_{\mathcal{B}\mathcal{C}} \mathbf{p}''_{\mathcal{B}}$. The factor $Z_{\mathcal{B}}/(Z_{\mathcal{B}} + t_z)$ in (15), where t_z is the third component of the vector $\mathbf{t}_{\mathcal{B}\mathcal{C}} = [t_x, t_y, t_z]^{\mathrm{T}}$, leads to writing $\mathbf{p}''_{\mathcal{B}}$ in homogeneous coordinates, *i.e.*, $\mathbf{p}''_{\mathcal{B}} = [u'^{\mathbf{D}}_{\mathcal{B}}, v'^{\mathbf{D}}_{\mathcal{B}}, 1]^{\mathrm{T}}$. For our setup, we may neglect t_z, *i.e.*, $t_z \approx 0$, since the two cameras in the hybrid ToF multi-camera rig are chosen to be co-planar, *i.e.*, the rotation matrix $\mathbf{R}_{\mathcal{B}\mathcal{C}}$ is a rotation in two dimensions and $Z_{\mathcal{B}} = Z_{\mathcal{C}}$. As a result, $\mathbf{H}_{\mathcal{B}\mathcal{C}}$ can be approximated by an affine transformation, and (15) simplifies

to

$$\begin{aligned}\mathbf{p}''_{\mathcal{B}} &= \begin{bmatrix} u'^{\mathbf{D}}_{\mathcal{B}} \\ v'^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{B}} \\ v^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} - \frac{1}{Z_{\mathcal{B}}} \begin{bmatrix} \delta^{-1}_{x,\mathcal{B}} f_{\mathcal{B}} & 0 & c_{x,\mathcal{B}} \\ 0 & \delta^{-1}_{y,\mathcal{B}} f_{\mathcal{B}} & c_{y,\mathcal{B}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{B}} \\ v^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} - \begin{bmatrix} \delta^{-1}_{x,\mathcal{B}} f_{\mathcal{B}} \cdot t_x / Z_{\mathcal{B}} \\ \delta^{-1}_{y,\mathcal{B}} f_{\mathcal{B}} \cdot t_y / Z_{\mathcal{B}} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{B}} \\ v^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} - \frac{f_{\mathcal{B}}}{Z_{\mathcal{B}}} \begin{bmatrix} t_x / \delta_{x,\mathcal{B}} \\ t_y / \delta_{y,\mathcal{B}} \\ 0 \end{bmatrix} \\ &=: \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{B}} \\ v^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} - \begin{bmatrix} \rho_x(Z_{\mathcal{B}}) \\ \rho_y(Z_{\mathcal{B}}) \\ 0 \end{bmatrix} = \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{B}} \\ v^{\mathbf{D}}_{\mathcal{B}} \\ 1 \end{bmatrix} - \frac{f_{\mathcal{B}}}{Z_{\mathcal{B}}} \begin{bmatrix} b_{x,\mathcal{B}} \\ b_{y,\mathcal{B}} \\ 0 \end{bmatrix}, \end{aligned} \tag{16}$$

which corresponds to $\mathbf{p}'_{\mathcal{B}}$ minus the binocular disparity introduced in (2). The possible error when determining the focal length $f_{\mathcal{B}}$ of the ToF camera can be neglected as $f_{\mathcal{B}} << Z_{\mathcal{B}}$ when correcting the disparity in (16). The binocular disparity in (16) is decomposed into two components as $\rho(Z_{\mathcal{B}}) = \rho_x(Z_{\mathcal{B}}) \cdot \vec{e}_x + \rho_y(Z_{\mathcal{B}}) \cdot \vec{e}_y$, where $\vec{e}_x$ and $\vec{e}_y$ are respectively the unit vectors along the x and y axes of the ToF reference frame $\mathcal{B}$.

We note that the order of the two previous steps can be exchanged by multiplying in (12) the transformation $\mathbf{H}_{\mathcal{BC}}$ inside the disparity shift, *i.e.*,

$$\mathbf{p}'_{\mathcal{C}} = \mathbf{H}_{\mathcal{BC}} \mathbf{p}'_{\mathcal{B}} - \frac{\mathbf{K}_{\mathcal{C}} \mathbf{R}_{\mathcal{BC}}}{Z_{\mathcal{C}}} \mathbf{t}_{\mathcal{BC}} =: \mathbf{p}''_{\mathcal{C}} - \frac{\mathbf{K}_{\mathcal{C}}}{Z_{\mathcal{C}}} \mathbf{t}'_{\mathcal{BC}}, \tag{17}$$

with the baseline $\mathbf{t}'_{\mathcal{BC}} = [t'_x, t'_y, t'_z]^{\mathrm{T}}$ measured from the reference frame $\mathcal{C}$ and $\mathbf{p}'_{\mathcal{B}}$ being transformed to $\mathbf{p}''_{\mathcal{C}}$ by $\mathbf{H}_{\mathcal{BC}}$. Analogously to (15), (17) simplifies to

$$\begin{aligned}\mathbf{p}'_{\mathcal{C}} &= \begin{bmatrix} u^{\mathbf{D}}_{\mathcal{C}} \\ v^{\mathbf{D}}_{\mathcal{C}} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} u'^{\mathbf{D}}_{\mathcal{C}} \\ v'^{\mathbf{D}}_{\mathcal{C}} \\ 1 \end{bmatrix} - \frac{1}{Z_{\mathcal{B}}} \begin{bmatrix} \delta^{-1}_{x,\mathcal{C}} f_{\mathcal{C}} & 0 & c_{x,\mathcal{C}} \\ 0 & \delta^{-1}_{y,\mathcal{C}} f_{\mathcal{C}} & c_{y,\mathcal{C}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t'_x \\ t'_y \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} u'^{\mathbf{D}}_{\mathcal{C}} \\ v'^{\mathbf{D}}_{\mathcal{C}} \\ 1 \end{bmatrix} - \begin{bmatrix} \delta^{-1}_{x,\mathcal{C}} f_{\mathcal{C}} \cdot t'_x / Z_{\mathcal{B}} \\ \delta^{-1}_{y,\mathcal{C}} f_{\mathcal{C}} \cdot t'_y / Z_{\mathcal{B}} \\ 0 \end{bmatrix} \\ &=: \begin{bmatrix} u'^{\mathbf{D}}_{\mathcal{C}} \\ v'^{\mathbf{D}}_{\mathcal{C}} \\ 1 \end{bmatrix} - \begin{bmatrix} \rho_x(Z_{\mathcal{B}}) \\ \rho_y(Z_{\mathcal{B}}) \\ 0 \end{bmatrix} = \begin{bmatrix} u'^{\mathbf{D}}_{\mathcal{C}} \\ v'^{\mathbf{D}}_{\mathcal{C}} \\ 1 \end{bmatrix} - \frac{f_{\mathcal{C}}}{Z_{\mathcal{B}}} \begin{bmatrix} b_{x,\mathcal{C}} \\ b_{y,\mathcal{C}} \\ 0 \end{bmatrix}. \end{aligned} \tag{18}$$

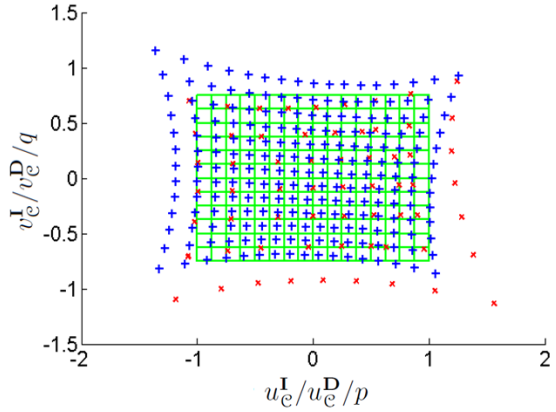
We see from (18) that image coordinates $\mathbf{p}'_{\mathcal{B}}$ are first transformed to $\mathbf{p}''_{\mathcal{C}}$ and then the disparity is computed using the intrinsic parameters in $\mathcal{C}$ and the distance $Z_{\mathcal{B}}$ given by the ToF camera. The values of the depth map $\mathbf{D}$ are, however, not invariant under this disparity shift, but may be recomputed according to (see equations (13) and (14)):

$$\mathbf{D}'(u'^{\mathbf{D}}_{\mathcal{B}}, v'^{\mathbf{D}}_{\mathcal{B}}) = Z_{\mathcal{B}} \cdot \frac{d(u'^{\mathbf{D}}_{\mathcal{B}}, v'^{\mathbf{D}}_{\mathcal{B}})}{f_{\mathcal{B}}}, \tag{19}$$

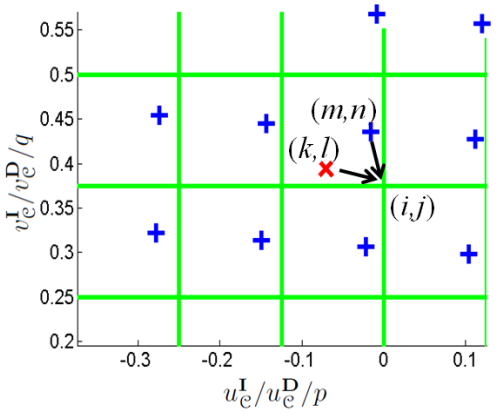
where $(u'^{\mathbf{D}}_{\mathcal{B}}, v'^{\mathbf{D}}_{\mathcal{B}})$ are the image coordinates shifted by the disparity, according to (16).

IV. Data matching

Data matching results from mapping the images $\mathbf{I}$ and $\mathbf{D}'$ on a common grid of pixels related to the reference frame $\mathcal{C}$, where the mapped images will be pixel aligned. Let us consider $\mathbf{I}$ to be the 2-D image of $(M \times N)$ pixels with image coordinates $\{(u^{\mathbf{I}}_{\mathcal{A},mn}, v^{\mathbf{I}}_{\mathcal{A},mn}),\ m = 1, \ldots, M;\ n = 1, \ldots, N\}$. Similarly, we consider $\mathbf{D}'$ to be the disparity shifted depth map of $(K \times L)$ pixels with image coordinates $\{(u'^{\mathbf{D}}_{\mathcal{B},kl}, v'^{\mathbf{D}}_{\mathcal{B},kl}),\ k = 1, \ldots, K;\ l = 1, \ldots, L\}$. Due to the transformation to the common grid, these image coordinates become $\{(u^{\mathbf{I}}_{\mathcal{C},mn}, v^{\mathbf{I}}_{\mathcal{C},mn}),\ m = 1, \ldots, M;\ n = 1, \ldots, N\}$ and $\{(u^{\mathbf{D}}_{\mathcal{C},kl}, v^{\mathbf{D}}_{\mathcal{C},kl}),\ k = 1, \ldots, K;\ l = 1, \ldots, L\}$, respectively. We define such a common mesh grid as $\mathbf{\Psi} = \{(p_{ij}, q_{ij}),\ i = 1, \ldots, M;\ j = 1, \ldots, N\}$, where the pair (p_{ij}, q_{ij}) represents the location of the image pixel corresponding to the row index i and the column index j. We set the grid $\mathbf{\Psi}$ to be of the same resolution $(M \times N)$ as the 2-D camera. There is, however, no restriction regarding the resolution of the resulting mapped images. Our choice of M and N in this paper is motivated by the low-level data fusion, which is intended for enhancing the ToF depth map up to the same 2-D camera resolution. In general, state-of-the-art approaches that deal with the mapping of images to a common grid intended for data matching are based on forward warping [14], [15]. Thus, each mapped image coordinate from $\mathbf{I}$ and $\mathbf{D}$ are assigned to the nearest pixel of the common grid. However, in most of the cases, the resolution of the depth map $\mathbf{D}$ is far below the resolution of the 2-D image $\mathbf{I}$, *i.e.*, $K << M$ and $L << M$, as illustrated in Fig. 1a. As a result, the warping of such a depth map $\mathbf{D}$ onto the common grid presents a large number of missing depth pixels. In other words, forward warping generates a sparse number of warped depth pixels, as shown in Fig. 2a. In contrast, we propose a backward warping approach in which we determine for each pixel (p_{ij}, q_{ij}) on the common grid, the nearest pixel $(u^{\mathbf{I}}_{\mathcal{C},mn}, v^{\mathbf{I}}_{\mathcal{C},mn})$ on the image $\mathbf{I}$ after being transformed onto $\mathcal{C}$, as illustrated in Fig.1b. Similarly, we determine for each pixel (p_{ij}, q_{ij}) the nearest pixel $(u^{\mathbf{D}}_{\mathcal{C},kl}, v^{\mathbf{D}}_{\mathcal{C},kl})$. As a result, our mapped images, $\mathbf{I}_{\mathcal{C}}$ and $\mathbf{D}_{\mathcal{C}}$ are perfectly aligned with a major advantage of $\mathbf{D}_{\mathcal{C}}$ being a dense depth map. Indeed, we show in Fig. 2b a comparison of the deth maps obtained using a forward mapping and our proposed counterpart; that could be referred to as backward warping. The two techniques are overall equivalent. Our proposed approach has however one clear advantage. It provides a dense depth map while the forward warping provides a very sparse depth map. As a result if there is a requirement for depth map downsampling, which is common for a real-time implementation, the downsampled sparse depth map becomes unusable. We claim therefore that our proposed backward warping is more appropriate for real-time applications.

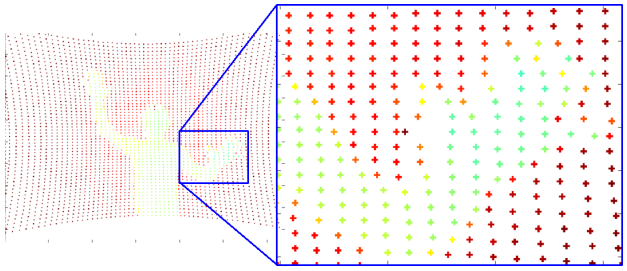


(a)

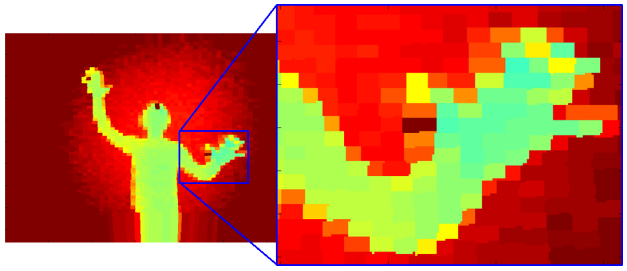


(b)

Fig. 1: Image coordinate transformation. (a) Shown are the transformed 2-D image coordinates $(u_{\mathcal{C}}^{\mathbf{I}}, v_{\mathcal{C}}^{\mathbf{I}})$ depicted as '+', the transformed ToF image coordinates $(u'^{\mathbf{D}}_{\mathcal{C}}, v'^{\mathbf{D}}_{\mathcal{C}})$ depicted as '$\times$', and the mesh grid Ψ coordinates (p, q). (b) Detail of the mapping procedure. It is apparent that a certain ToF pixel (k, l) will be mapped to several mesh grid pixels (i, j). Reference frames $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are depicted in blue, red, and green, respectively.



(a) Forward warping



(b) Proposed backward warping

Fig. 2: Comparison of the dense depth map obtained using our method, *i.e.*, backward warping (b) and the sparse depth map points obtained by forward warping (b). We refer the reader to the electronic version of the paper in order to better appreciate the differences between the forward and backward warping result.

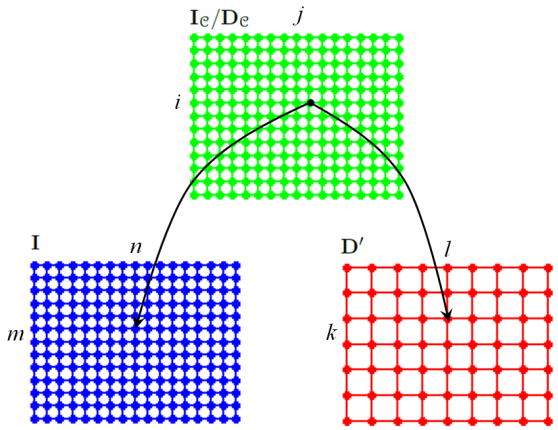


Fig. 3: The look-up tables $\mathbf{L}_{\mathcal{AC}}$ and $\mathbf{L}_{\mathcal{BC}}$ associate each pixel coordinates in $\mathcal{C}$ to a unique pixel in $\mathcal{A}$ and $\mathcal{B}$, respectively.

A. 2-D camera LUT

The relationship between the raw images and the mapped ones can be represented by an array that associates each pixel coordinates in the unified reference frame $\mathcal{C}$ to a unique pixel in $\mathcal{A}$ and $\mathcal{B}$, as illustrated in Fig.3. This associative array or look-up table (LUT) can be computed off-line in order to reduce the complexity of the mapping procedure to a single indexing operation and leading to real-time implementation.

We define the mapping $(i, j) \mapsto (m, n) = \mathbf{L}_{\mathcal{AC}}(i, j)$, as $\mathbf{L}_{\mathcal{AC}}(i, j) = \arg\min_{(m,n)} \|(p_{ij}, q_{ij}) - (u^{\mathbf{I}}_{\mathcal{C},mn}, v^{\mathbf{I}}_{\mathcal{C},mn})\|_2$. The stored LUT $\mathbf{L}_{\mathcal{AC}}$ allows to generate the new mapped image as follows $\mathbf{I}_{\mathcal{C}}(i, j) = \mathbf{I}\big(\mathbf{L}_{\mathcal{AC}}(i, j)\big)$, for all i, j.

B. ToF camera LUT

The same procedure as the one presented for determining the 2-D camera LUT applies for the ToF camera LUT that we refer to as $\mathbf{L}_{\mathcal{BC}}$. Thus, we place the same mesh grid $\mathbf{\Psi}$ onto the disparity corrected and transformed image coordinates $(u^{\mathbf{D}}_{\mathcal{C}}, v^{\mathbf{D}}_{\mathcal{C}})$ and we perform a nearest neighbour search to determine the pixel (k, l) from $\mathbf{D}'$ with the position $(u^{\mathbf{D}}_{\mathcal{C}}, v^{\mathbf{D}}_{\mathcal{C}})$ nearest to (p_{ij}, q_{ij}). The mapped depth map $\mathbf{D}_{\mathcal{C}}$ results from $\mathbf{D}_{\mathcal{C}}(i, j) = \mathbf{D}\big(\mathbf{L}_{\mathcal{BC}}(i, j)\big)$, for all (i, j). We note that the mapping described by this mesh grid also upsamples the mapped image coordinates to the 2-D camera resolution $(M \times N)$. We did not consider other interpolation techniques such as linear or bilinear interpolation because they may generate unwanted artefacts when applied on ToF data due to their characteristics such as incorrect measurements at large distances. These pixel values must not be considered in an interpolation, but require a special treatment. Also, real distances within the edges in the scene should not be

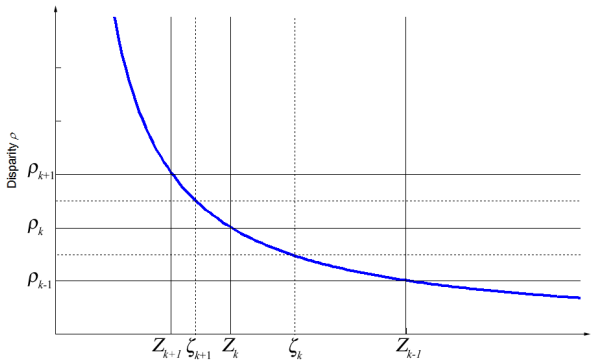


Fig. 4: Z range of the ToF camera divided into K intervals $[Z_{k+1}, Z_k]$ defined by equidistant disparity values $\rho_k(Z_{\mathcal{B}}) = k \times \rho$. Within each interval, the disparity ρ varies less than 1 pixel size δ.

interpolated. At the end of the mapping process, both resulting images $\mathbf{I}_{\mathcal{C}}$ and $\mathbf{D}_{\mathcal{C}}$ generated from their respective $\mathbf{L}_{\mathcal{AC}}$ and $\mathbf{L}_{\mathcal{BC}}$ LUTs are pixel aligned. Nevertheless, $\mathbf{L}_{\mathcal{BC}}$ that generates $\mathbf{D}_{\mathcal{C}}$ is distance-dependent. Due to the disparity shift presented in Section III-D, the resulting $\mathbf{L}_{\mathcal{BC}}$ LUT depends on the depth map information and thus on the scene configuration. The easiest way to deal with this dependence would be computing the $\mathbf{L}_{\mathcal{BC}}$ LUT for each recorded ToF frame; however, this implies a high computational time, and consequently, it will not be viable if real-time performance is required. Indeed, the off-line computation of a single $\mathbf{L}_{\mathcal{BC}}$ is close to 15 minutes using Matlab for Windows on the system we have used to run our experimental results.

In order to achieve real-time performance on dynamic scenes, we propose to consider an array $\{\mathbf{L}_{\mathcal{BC},k}\}$, $k = 0, \ldots, K-1$, of LUTs where each LUT $\mathbf{L}_{\mathcal{BC},k}$ tackles a different disparity $\rho_k(Z_{\mathcal{B}})$, corresponding to a plane at a fixed distance $Z_k = f_{\mathcal{C}} \cdot \frac{|\mathbf{b}|}{k}$ to the system. We choose the discrete disparities as multiples of the pixel size in the mapped depth map $\mathbf{D}_{\mathcal{C}}$, *i.e.*, $\rho_k = s_b k, k = 0, \ldots, K-1$ where $s_b = \mathbf{b}/|\mathbf{b}|$ is the unit vector of the baseline shift. Dividing the Z range of the ToF camera into K intervals $[\zeta_{k+1}, \zeta_k]$ around Z_k with

$$\begin{aligned} \zeta_0 &= \infty \\ \zeta_k &= f \cdot \frac{|\mathbf{b}|}{(k - \frac{1}{2})}, \; k = 1, \ldots, K, \end{aligned} \tag{20}$$

one finds that for each pixel of the ToF camera with a Z value in the interval $[\zeta_{k+1}, \zeta_k]$, the disparity equals $\rho_k(Z_{\mathcal{B}})$, with an error less than $\delta/2$, *i.e.*, half the size of a pixel in the mapped depth map $\mathbf{D}_{\mathcal{C}}$, as shown in Fig. 4. The maximum binocular disparity is given by the minimum $Z-$measurement range of the ToF camera, Z_{min} (the minimum Z value in the setup). The number K of different disparities to be considered is given by $K \geq f \cdot \frac{|\mathbf{b}|}{Z_{min}} + \frac{1}{2}$. The mapping is then performed by the iterative Algorithm 1, where $\mathbf{Z}$ denotes the image of $Z_{\mathcal{B}}$ values calculated from the depth map $\mathbf{D}$ using (13). This mapping procedure allows the low-resolution depth map $\mathbf{D}$ to be mapped in real-time to a depth map $\mathbf{D}_{\mathcal{C}}$, where each pixel matches a pixel in the already mapped $\mathbf{I}_{\mathcal{C}}$ image. In the occlusion handling block, we check if the condition $Z \in [\zeta_{k+1}, \zeta_k]$ is fulfilled. If not, the selected pixel is labelled as occluded.

Algorithm 1 Mapping algorithm

```
for i = 1 to N do
  for j = 1 to M do
    k = K
    {Search Z_k interval}
    while (k > 0) and (Z(L_BC,k(i,j)) > ζ_k) do
      k ← k − 1
    end while
    {Occlusion handling}
    if (k < K) and (Z(L_BC,k(i,j)) < ζ_{k+1}) then
      k ← k + 1
    end if
    {Mapping}
    D_C(i,j) = D'(L_BC,k(i,j))
  end for
end for
```

Algorithm 2 Optimized mapping algorithm

```
for i = 1 to N do
  for j = 1 to M do
    k = K
    {Search Z_k interval}
    while (k > 0) and (Z'(L_BC,0(i, j − sk)) > ζ_k)
    do
      k ← k − 1
    end while
    {Occlusion handling}
    if (k < K) and (Z'(L_BC,0(i, j − sk)) < ζ_{k+1})
    then
      k ← k + 1
    end if
    {Mapping}
    Z_C(i,j) = Z'(L_BC,0(i, j − sk))
  end for
end for
```

Although we achieve a high performance within the mapping procedure, the memory required to store the K LUTs is considerable, being a problem to deal with in case of real embedded applications. To that end, we propose a procedure to reduce the memory requirements which is applicable to hybrid ToF multi-camera systems with co-planar cameras. In this case, we proceed by considering the transformation $\mathbf{H}_{\mathcal{BC}}$ inside the disparity shift, as discussed in Section III-D (see (18)). In our case, the x axes of the camera reference frames are chosen to be parallel to the baseline between the cameras, *i.e.*, $\mathbf{b} = [b_x, 0, 0]^{\mathrm{T}}$, and thus the disparity shift extends in the x direction of the image frame. The disparity differs by exactly one pixel in x direction when calculated at two different distances Z_k and Z_{k+1}. The corresponding two LUTs are then related via $L_{\mathcal{BC},k+1}(i,j) = L_{\mathcal{BC},k}(i, j-s)$ with $s = sign(b) = \pm 1$ being the sign of the baseline shift

with respect to the x axis, *i.e.*, indicating on which side of the ToF-camera the 2-D camera is positioned with respect to the x axis of the unified reference frame. Consequently, it is sufficient to store a single LUT $L_{\mathcal{BC},0}$ calculated on an extended mesh grid $\mathbf{\Psi}$ of size $M \times (N+k)$, which defines all K LUTs via $L_{\mathcal{BC},k}(i,j) = L_{\mathcal{BC},0}(i, j-sk)$ with $i = 1,\ldots,M$, $j = 1\ldots,N$, and $k = 0,\ldots,K-1$. Unlike the range image $\mathbf{D}$, the $\mathbf{Z}$ image needs to be recalculated by the same projective transformation resulting in a new $\mathbf{Z}'$ image (see (13)). We proceed by using the Algorithm 2, where $\mathbf{Z}_{\mathcal{C}}$ is the resulting matrix of $Z_{\mathcal{B}}$ coordinates on the common coordinate grid in the unified reference frame. The latter allows to calculate a radial distance image $\mathbf{D}_{\mathcal{C}}$ using (19) for the coordinates of the common coordinate grid.

V. Experimental results

We herein evaluate the proposed real-time mapping procedure for hybrid ToF multi-camera rig data matching. We have performed our experiments based on various scenes including our own recorded sequences as well as scenes from the Middlebury stereo dataset[1]. The Middelbury dataset provides ground truth disparity maps in addition to the corresponding 2-D RGB images from different views. For our own recordings, we have used a hybrid ToF multi-camera rig composed of a 3D MLI Sensor™ from IEE S.A. [26], and a Flea®2 video camera from Point Grey Research, Inc. [27] (see Fig. 5). The 3D MLI Sensor™ is an industrialized ToF camera that has a resolution of (56×61) pixels with a pixel size δ of 68 μm × 49 μm and covers a measurement range up to 7500 mm. The Flea®2 camera provides a resolution of (648×488) pixels with a pixel size of 7.7 μm × 7.7 μm. The two cameras were coupled for a narrow baseline of 30 mm, and they were frame-synchronized with each other at the ToF camera frame rate. Regarding the implementation, we programmed the mapping presented in Section IV in C language, and we ran the experiments on a Pentium IV, 2.66 GHz with 1 GB of RAM.

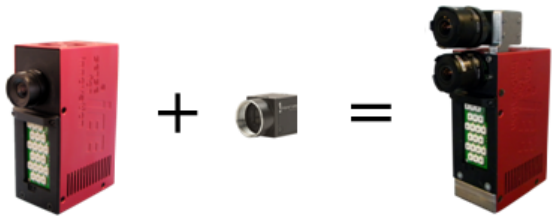
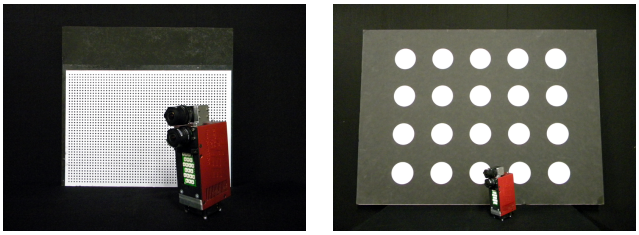


Fig. 5: Frome left to right: 3D MLI Sensor™ from IEE S.A., Flea®2 video camera from Point Grey Research, Inc and hybrid ToF multi-camera rig prototype.

A. Hybrid ToF multi-camera rig calibration

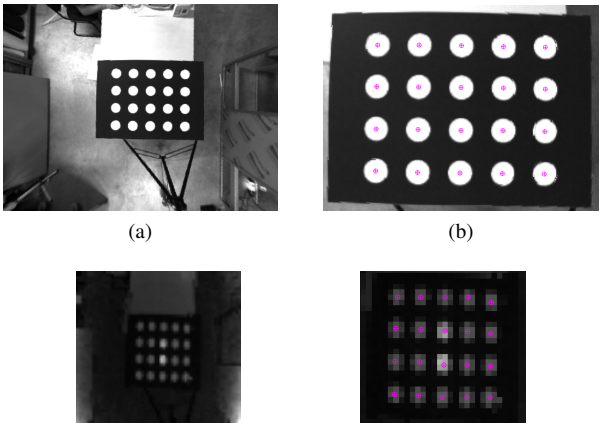
We determine the calibration parameters of our test rig by following a standard calibration method as discussed in Section III with the insights proposed by Fuchs et al. in [21].

[1]Middlebury Stereo Dataset, http://vision.middlebury.edu/stereo



(a) 2-D camera calibration pattern. (b) ToF camera calibration pattern.

Fig. 6: Calibration patterns used to estimate the intrinsic and relative extrinsic camera parameters.



(a) (b) (c) (d)

Fig. 7: ToF camera pattern images recorded by the 2-D and the ToF camera, (a) and (c) respectively, to estimate the relative extrinsic parameters. The centroid operator detects with sub-pixel accuracy the centroid of each target, shown in red in (b) and (d).

Although the calibration procedure does not require special tools, the calibration pattern must contain circular targets large enough to be distinguished in the low-resolution ToF amplitude image $\mathbf{A}$. Amplitude images $\mathbf{A}$ result from the intensity reflected by the active illumination emitted by the ToF camera. Therefore, we have designed different calibration targets in order to estimate each camera's intrinsic parameters, as shown in Fig. 6. From [28], the determination of the relative extrinsic parameters, *i.e.*, the parameters of an affine transformation that relates each camera reference frame to the unified reference frame, requires four correspondence points with no three points collinear on either plane. Thus, the same ToF calibration pattern (see Fig. 6b) can be used as it allows to estimate up to 20 control points. The control points correspond to the centroid of each dot in the image, which are determined with sub-pixel accuracy, only limited by the image resolution (see Fig. 7b, 7d). In the case where a control point appears as only one pixel, the centroid will be the image coordinates of this pixel and therefore will induce a discretization error in the interval $[-\delta/2, \delta/2]$. Assuming that the discretization error is statistically equally distributed over that interval, one can easily calculate the Root Mean Square Error (RMSE) to be $\Delta = \delta/\sqrt{12}$. When a dot appears as a blob of N pixels,

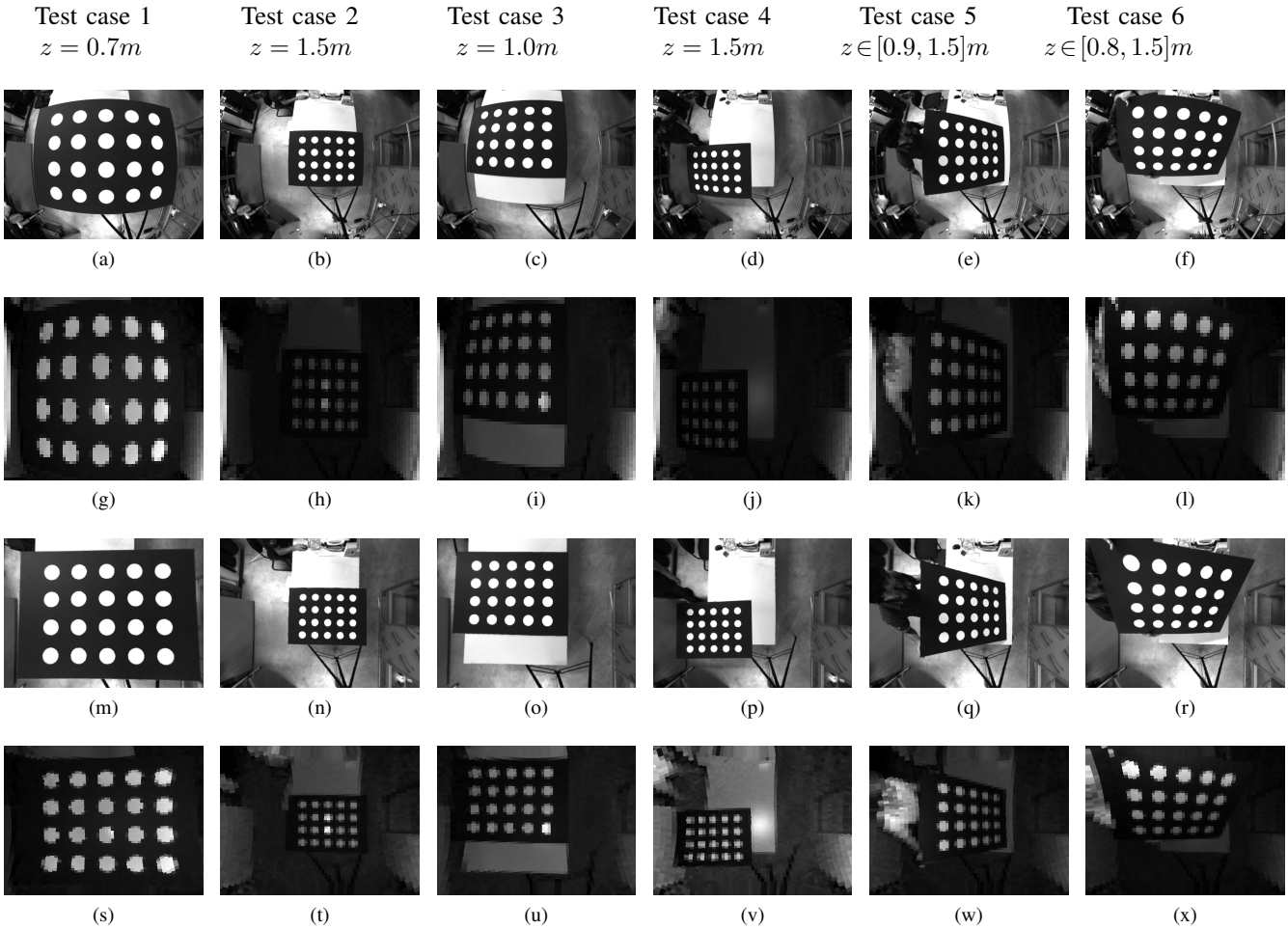


Fig. 8: Test cases for data matching. 1^{st} row: 2-D acquisitions. 2^{nd} row: ToF acquisitions. 3^{rd} row: 2-D mapped. 4^{th} row: ToF mapped.

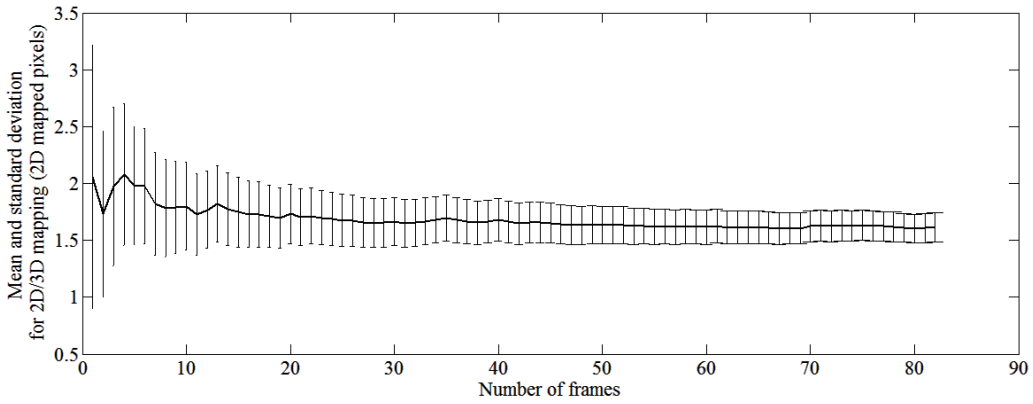
one obtains a RMSE of

$$\Delta = \frac{1}{\sqrt{12N}}\delta, \tag{21}$$

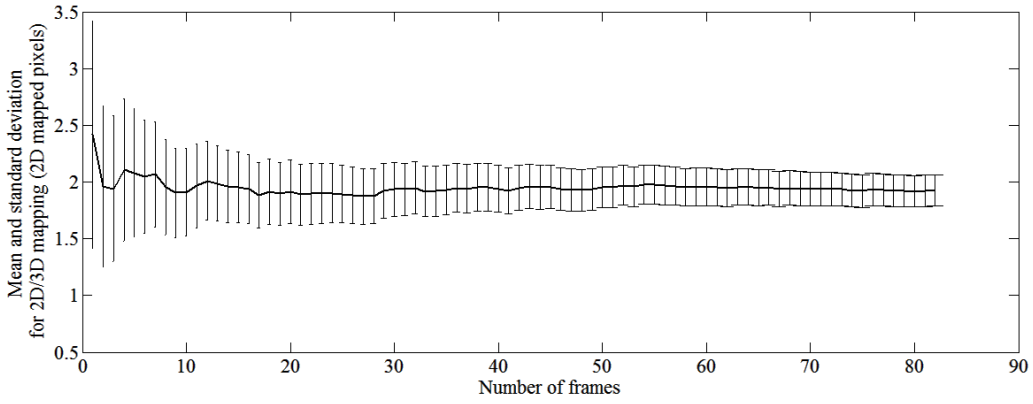
which is then more accurate than when using edges, *i.e.*, $\Delta = \delta/2$. The relative calibration pattern is located at a distance of 1530 mm from the sensing system and roughly positioned in the centre of the FOV (see Figure 7c). In addition, we consider the 20 control points in order to obtain a maximum accuracy. Thereby, we take as reference the positions detected in the 2-D image **I**. In the ToF amplitude image **A**, the average size of the detected dots is 7.7 pixels, yielding, according to (21), a sub-pixel accuracy of the centroid of $\Delta_x = 7.1$ μm and $\Delta_y = 5.1$ μm. We note that the pixel size of the ToF camera is $\delta_x = 68$ μm and $\delta_y = 49$ μm. The RMSE of the centroid coordinates after relating the centroid coordinates in **A** with the centroid coordinates in **I** is 5.4 μm in the x direction and 7.9 μm in the y direction. We confirm that our centroid operator achieves an accuracy of the same order as the one given by (21), which is clearly better than the low resolution of the ToF camera and close to 1 pixel of the 2-D camera resolution, which is $(7.4 \times 7.4)\mu m^2$.

B. Data matching

In order to analyse the data mapping step, we have considered six representative test cases in which we recorded the calibration pattern displaced around the FOV of the sensing system, and at different depths and orientations (see Fig. 8). We first quantitatively compare our proposed approach to a common mapping using a simple projective transformation, *i.e.*, a plane-to-plane transformation or a 2-D homography. To that end, we focus on the four first test cases where the recorded pattern is always located parallel to the sensing system. In Table I, the two first rows report the RMSE of the centroids of the mapped control points using a 2-D homography. As expected, the use of a 2-D homography performs better if the distance at which it has been computed coincides with the distance at which the control points are located (see the first four test cases in the second row of Table I). However, if we use a unique homography for these test cases, the matching error increases as soon as we vary the depth at which the pattern is located (see the first four test cases in the first row of Table I). In general cases where the pattern is arbitrarily located and oriented in front of the sensing system (see test cases 5 and 6 in Fig. 8 and the last two columns of Table I), the use of a plane-to-plane transformation reports an



(a)



(b)

Fig. 9: Quantitative evaluation of the mapping accuracy and precision over a video sequence of 82 frames: (a) error in x direction, (b) error in y direction.

error much bigger than using the proposed approach. Indeed, the proposed data mapping approach presents an accuracy up to one 2-D pixel, which is caused by the approximation, given in (20), of Z_k by the interval $[\zeta_{k+1}, \zeta_k]$. We note that the errors reported in Table I also contain the error of the centroid operator introduced in Section V-A. Thus, the evaluation results for our mapping method show a consistent error of about 2 mapped image pixels, or less if we take into account the error due to the centroid operator. This observation is confirmed by the quantitative evaluation carried out on a video sequence of 82 frames provided as supplementary material, and the accuracy results are summarized in Fig. 9. The recorded video contains frames of a randomly displaced calibration pattern within the FOV of the sensing system. We automatically determine the coordinates of the centroids of each control point for each 2-D frame, using the Hough transform for circles, and each ToF mapped frame. Fig. 9 plots the mean error between the coordinates of the 2-D and the ToF centroids over an increasing number of frames along with the corresponding standard deviation at each point. We do this evaluation for both x direction (Fig. 9a) and y direction (Fig. 9b). In both graphs, we start by considering the centroids of a single and arbitrary frame within the video sequence. We can observe that both mean errors on x and y directions are consistently between 1.5 and 2 2-D pixels, and their standard deviations converge, starting from 20 frames, to a value of less than one half of a mapped 2-D pixel. These quantitative results confirm that our proposed mapping method is accurate and has a subpixel precision. The last row of Table I reports the error when considering the most common or general approach for data mapping, *i.e.*, using a full 3-D projection (with no approximations). By using this

TABLE I: Data matching error for the six representative test cases. The table compares the RMSE (in pixels) over 20 control points, separately computed for x and y pixel coordinates, between our mapping procedure and the mapping using first a simple projective transformation (two first rows) and a 3-D transformation without approximations (last row).

Test cases		1	2	3	4	5	6
RMSE using a unique proj. transf. ($z = 1.5\ m$)	x	7.52	1.67	3.66	1.33	2.59	3.75
	y	1.45	1.26	1.23	1.88	1.42	1.57
RMSE using a computed proj. transf. for each test case	x	1.29	1.31	1.87	1.33	3.52	3.90
	y	1.48	1.26	1.22	1.88	1.42	1.69
RMSE using the proposed mapping procedure	x	2.14	1.45	1.69	1.56	1.47	2.04
	y	1.40	1.27	1.37	1.84	1.43	1.72
RMSE using a 3-D projection	x	1.58	1.37	1.51	1.42	1.48	2.00
	y	1.43	1.25	1.21	1.79	1.35	1.76

3-D projection, the mapped centroids are matching with more accuracy than by using the proposed approach. However, the loss in accuracy is worth a significant gain in speed. We note that the mean in seconds for computing this 3-D projection for a single frame is 782.67 seconds (brute force and non-optimized Matlab implementation), while using the proposed approach only takes 0.54 seconds for the whole mapping procedure (also in Matlab).

C. Application to low-level data fusion

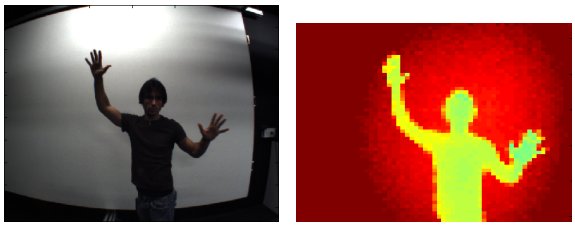
We further evaluate our mapping procedure by performing a low-level data fusion technique that considers our mapped data. To that end we use the pixel weighted average strategy (PWAS) for depth sensor data fusion that we presented in [6]. The PWAS filter copes well with inaccurate edge values within the low-resolution depth map. In contrast to alternative depth enhancement methods proposed in the literature, the PWAS filter contains an additional factor $Q(\cdot)$, named credibility map, and defined at a pixel position $\mathbf{q}$ as a weighted function of the gradient of the low-resolution depth map $\nabla\mathbf{R}$ such that $Q(\mathbf{q}) = f_{\mathbf{Q}}(|\nabla\mathbf{R}(\mathbf{q})|)$. The weighting function is chosen to be a Gaussian function with standard deviation $\sigma_{\mathbf{Q}}$. The credibility map assigns a reliability weight to each depth map value as a function of the scene's geometry. By so doing, depth measurements that are considered to be unreliable are replaced by reliable values in their neighbourhood and adjusted to the 2-D guidance image. Considering $Q_{\mathcal{C}}$, the credibility map relative to the unified reference frame $\mathcal{C}$, the PWAS filter takes the following form:

$$\mathbf{J}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in N(\mathbf{p})} f_{\mathbf{S}}(\mathbf{p},\mathbf{q}) f_{\mathbf{I}}\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big) Q_{\mathcal{C}}(\mathbf{q}) \mathbf{R}_{\mathcal{C}}(\mathbf{q})}{\sum_{\mathbf{q}\in N(\mathbf{p})} f_{\mathbf{S}}(\mathbf{p},\mathbf{q}) f_{\mathbf{I}}\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big) Q_{\mathcal{C}}(\mathbf{q})}, \tag{22}$$

where the weighting functions $f_{\mathbf{S}}(\cdot)$ and $f_{\mathbf{I}}(\cdot)$ are also chosen to be Gaussian functions with standard deviations $\sigma_{\mathbf{S}}$ and $\sigma_{\mathbf{I}}$, respectively. The resulting filtered image $\mathbf{J}$ is an enhanced version of $\mathbf{R}$, that presents less discontinuities and a significantly reduced noise level. The reduction of the global noise is due to the nature of the bilateral filter on which the PWAS filter is based.

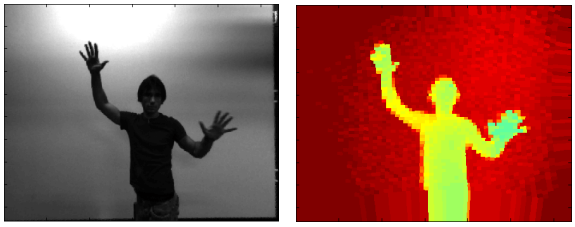
Fig. 10 shows a visual example where the enhanced depth map $\mathbf{J}$ results from the PWAS filter applied to the mapped data recorded by our test rig. Note that $\mathbf{J}$ has the same resolution as the guidance image $\mathbf{I}$ with accurate depth measurements along depth edges. Moreover, the global noise has been significantly reduced.

Finally, we quantify our mapping using the cluttered scenes: *Art*, *Books*, and *Moebius* from the Middlebury dataset (Fig. 11). To that end, we compute the depth maps related to each of the views, *i.e.*, *view 1* and *view 5*, for each scene considering the provided disparity maps and the given system parameters. Then, we downsample the computed depth map at *view 5* for each scene and we map them to *view 1*. We find a global RMSE of less than 0.15% between the mapped depth map and the one originally computed from the given system parameters. Notice that this measurement has been computed without considering the occlusion areas (see the third column of Fig. 11). This result consolidates the above mapping experiments.



(a) 2-D guidance image $\mathbf{I}$, (640×480) pixels

(b) Low-resolution depth map $\mathbf{R}$, (61×56) pixels



(c) $\mathbf{I}_{\mathcal{C}}$ resulting from mapping using $L_{\mathcal{AC}}$, (640×480) pixels

(d) $\mathbf{R}_{\mathcal{C}}$ resulting from mapping $\mathbf{R}$ using $L_{\mathcal{BC}}$, (640×480) pixels

(e) Enhanced depth map $\mathbf{J}$, (640×480) pixels

Fig. 10: Low-level data fusion example using the PWAS filter presented in [6], $\sigma_{\mathbf{S}} = 10$, $\sigma_{\mathbf{I}} = 10$, and $\sigma_{\mathbf{Q}} = 50$. Distance measurements are done in mm.

VI. Conclusion

In this paper, we presented a dedicated mapping procedure intended for hybrid ToF multi-camera rig data matching. The mapping procedure projects the image coordinates from each camera reference frame to a unified reference frame where the projected data is pixel aligned. We showed that this proposed mapping is suitable for all kinds of ToF cameras even with large fields of view and low resolutions. This was achieved by accounting for disparity variations in the mapping model. Disparity correction becomes then possible by using the depth information acquired by the ToF camera. Furthermore, we presented a real-time implementation of this procedure thanks to a pixel association described in a set of look-up tables that solve the binocular disparity. Indeed, whereas the computation of a single look-up table to map a pair of raw data recordings to the unified reference frame was close to 15 minutes using Matlab, by using the optimal implementation discussed in Section IV-B, the time for data matching of the same recordings

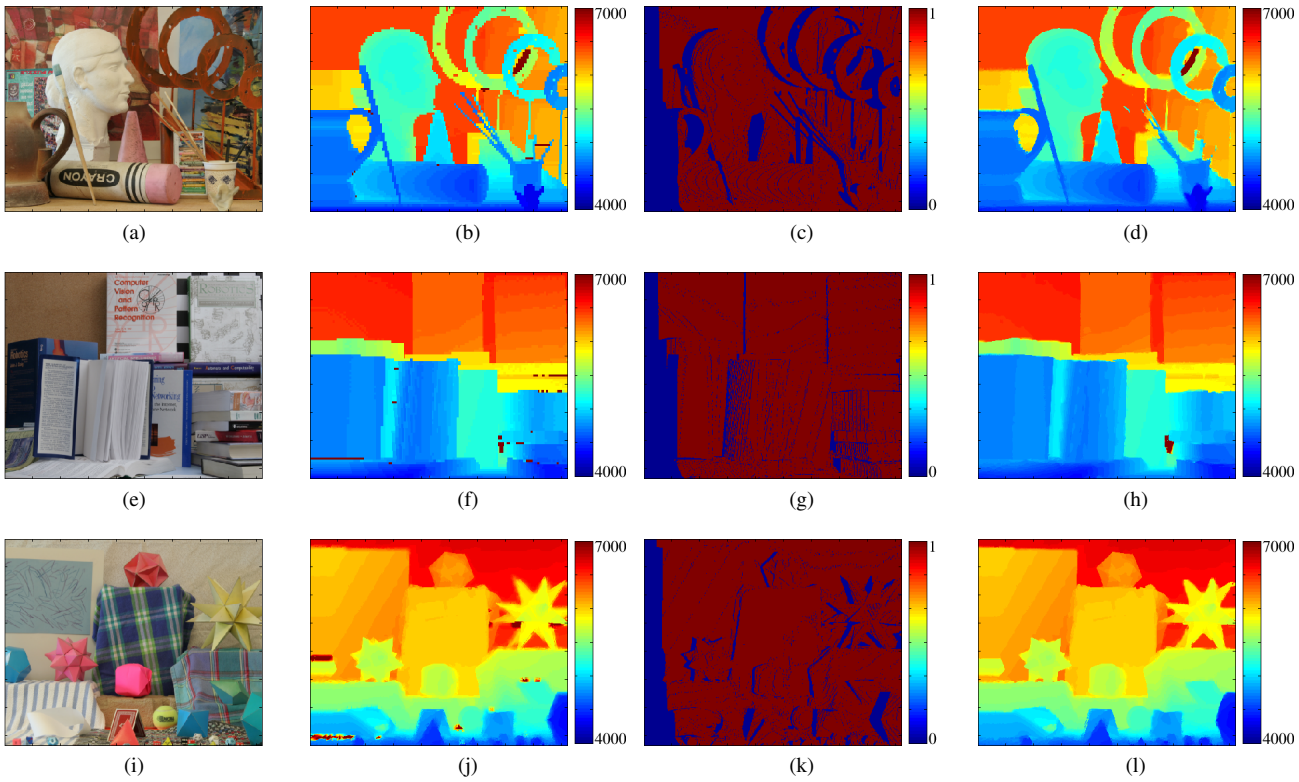


Fig. 11: Data fusion on the *Art*, *Books* and *Moebius* scenes from the Middlebury dataset, 1^{st}, 2^{nd}, and 3^{rd} rows respectively. 1^{st} col.: 2-D guidance image $\mathbf{I}$. 2^{nd} col.: Low-resolution depth map $\mathbf{R}$, (downsampled by a factor of 4). 3^{rd} col.: Occlusion map. 4^{th} col.: Enhanced depth map $\mathbf{J}$ using the PWAS filter. (These examples were presented first in [29]).

was reduced to 2 seconds, *i.e.*, reduced by a factor of 450. In order to verify that the proposed method can run in real-time applications, we have implemented it in C. We achieved a computation time of only 2 milliseconds per frame. Our final experimental results show an accurate pixel alignment that assists fusion techniques enhancing the initial low resolution depth maps, and at the same time, reducing their noise level. The results from fusion techniques based on the calibration and mapping methods developed herein show promise. More elaborate fusion techniques such as the multi-lateral filter for real-time depth enhancement presented in [30] can certainly be considered. However, real-time in depth enhancement has been possible by considering the recent fast implementation techniques for bilateral filtering [31], [32], [33] in which the data to be filtered must be downsampled. This is the main reason why we propose a backward warping approach that generates a dense depth map instead of the sparse depth map generated using forward warping.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their thorough review of the present paper and for their valuable comments and suggestions. They also thank Thomas Solignac for his great support in implementing the proposed method in C++. This work was supported by the National Research Fund (FNR), Luxembourg, under the AFR PhD grant TR-PHD BFR08-120 and under the CORE project C11/BM/1204105/FAVE/Ottersten.

REFERENCES

[1] S. Foix, G. Aleny, and C. Torras, "Exploitation of time-of-flight (ToF) cameras." Institut de Robtica i Informtica Industrial, CSIC-UPC, Tech. Rep. IRI-TR-10-07, 2010.

[2] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Sensors in Computer Graphics," in Eurographics - State of the Art Reports, March 2009, pp. 119–134.

[3] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo, "Kinect identity: Technology and experience," Computer, vol. 44, no. 4, pp. 94–96, April 2011.

[4] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCVW), 2008.

[5] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in NIPS. MIT Press, 2005, pp. 291–298.

[6] F. Garcia, B. Mirbach, B. Ottersten, F. Grandidier, and A. Cuesta, "Pixel Weighted Average Strategy for Depth Sensor Data Fusion," in International Conference on Image Processing (ICIP), September 2010, pp. 2805–2808.

[7] S. Gloud, P. Baumstarck, M. Quigley, Y. N. Andrew, and K. Daphne, "Integrating visual and range data for robotic object detection," in Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCVW), 2008.

[8] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in SIGGRAPH '07: ACM SIGGRAPH 2007 papers. New York, NY, USA: ACM, 2007, p. 96.

[9] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.

[10] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.

[11] J. Zhu, L. Wang, R. Yang, J. Davis, and Z. Pan, "Reliability Fusion of Time-of-Flight Depth and Stereo for High Quality Depth Maps," IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 99, p. 1, 2010.

[12] A. Frick, B. Bartczack, and R. Koch, "3D-TV LDV content generation with a hybrid ToF-multicamera RIG," in 3DTV-CON, June 2010, pp. 1–4.

[13] K. Natroshvili, M. Schmid, M. Stephan, A. Stiegler, and T. Schamm, "Real time pedestrian detection by fusing pmd and cmos cameras," in Intelligent Vehicles Symposium, 2008, pp. 925–929.

[14] M. Do, Q. Nguyen, H. Nguyen, D. Kubacki, and S. Patel, "Immersive Visual Communication," IEEE Transactions on Signal Processing Magazine, IEEE, vol. 28, no. 1, pp. 58–66, January 2011.

[15] E.-K. Lee and Y.-S. Ho, "Generation of multi-view video using a fusion camera system for 3D displays," IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2797–2805, November 2010.

[16] "Ibeo atutomotive systems," http://www.ibeo-as.com, August 2011.

[17] "The Nipon Signal Co., LTD," http://www.signal.co.jp, August 2011.

[18] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," International Journal of Computer Vision, vol. 47, no. 1–3, pp. 7–42, April–June 2002.

[19] Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun, "Design and calibration of a multi-view TOF sensor fusion system," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2008, pp. 1–7.

[20] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, "Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction," in IEEE Workshop on 3-D Digital Imaging and Modeling, 3DIM, 2009.

[21] S. Fuchs and G. Hirzinger, "Extrinsic and Depth Calibration of ToF-cameras," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 2008.

[22] T. Kahlmann and H. Ingensand, "Increased Accuracy of 3D Range Imaging Camera by Means of Calibration," in Optical 3-D Measurement Techniques VIII (Eds.: Grn, Kahmen), 2007, pp. 101–108.

[23] L. Marvin, K. Andreas, and H. Klaus, "Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images," in International Symposium on Signals, Circuits and Systems. ISSCS, vol. 1, 2007, pp. 1–4.

[24] J.-Y. Bouguet, "Camera calibration toolbox for matlab," http://vision.caltech.edu/bouguetj/calib, November 2009.

[25] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 1st ed. O'Reilly Media, 2008.

[26] "IEE S.A., 3D MLI Sensor™," http://www.iee.lu, October 2010.

[27] "Point Grey Research, Inc., Flea®2," http://www.ptgrey.com/products/flea2/index.asp, October 2010.

[28] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy, "Canonical frames for planar object recognition," in In Proceedings of European Conference on Computer Vision (ECCV). Springer Berlin / Heidelberg, 1992, pp. 757–772.

[29] F. Garcia, D. Aouada, B. Mirbach, T. Solignac, and B. Ottersten, "Real-time Hybrid ToF multi-camera Rig Fusion System for Depth Map Enhancement," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2011, pp. 1–8.

[30] ——, "A New Multi-lateral Filter for Real-Time Depth Enhancement," in Advanced Video and Signal-Based Surveillance (AVSS), 2011.

[31] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in International Journal of Computer Vision, vol. 81. Kluwer Academic Publishers, 2009, pp. 24–52.

[32] F. Porikli, "Constant time o(1) bilateral filtering," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 2008, pp. 1–8.

[33] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time O(1) bilateral filtering," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 2009, pp. 557–564.

Frederic Garcia Becerro (M'12) was born in Girona, Spain, on October 2, 1982. He received the B.Sc. degree in Technical Computer Systems Engineering and the M.Sc. in Computer Science from the University of Girona, Spain, in 2003 and 2006, respectively. He received the European M.Sc. in Computer Vision and Robotics (VIBOT) from Heriot-Watt University, Scotland, the Université de Borgogne, France, and the University of Girona, Spain, in 2008, and the Ph.D. degree in Computer Science in March 2012 from the University of Luxembourg, Luxembourg. Dr. Garcia is currently a Research Associate at the Interdisciplinary Centre for Security, Reliability and Trust (SnT) in Luxembourg. His research interests span areas of image processing, computer vision, and robotics; focusing on sensor fusion techniques for real people sensing applications such as identification, gesture recognition, and counting. Dr. Garcia is member of the IEEE Signal Processing Society. Dr. Garcia received a paper awarded Best Student Paper Award at IEEE 7th International Symposium on Image and Signal Processing and Analysis (ISPA'11).

Djamila Aouada (S'05, M'09) was born in Blida, Algeria, on November 10, 1982. She received the State Engineering degree (Ingéniorat d'État) in electronics in June 2005, from the École Nationale Polytechnique (ENP), Algiers, Algeria, and the Ph.D. degree in electrical engineering in May 2009, from North Carolina State University (NCSU), Raleigh, NC. From June to August 2007, she participated in the data sciences summer school at Los Alamos National Laboratory (LANL), Los Alamos, NM, as part of the Geometric Measure Theory group (GMT). From July to September 2008, Dr. Aouada worked as a consultant for Alcatel-Lucent Bell Laboratories, Murray Hill, NJ. Since November 2009, Dr. Aouada has been performing and supervising research as a Research Associate at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg . Her research interests span the areas of signal and image processing, computer vision, pattern recognition and data modeling. Dr. Aouada is member of the IEEE Signal Processing Society and the Eta Kappa Nu honor society (HKN). She is a co-author of a paper awarded Best Student Paper Award at IEEE 7th International Symposium on Image and Signal Processing and Analysis (ISPA'11).

Bruno Mirbach received the Ph.D. degree in theoretical physics from the University of Kaiserslautern, Kaiserslautern, Germany, in 1996. He was a Postdoctoral Researcher with the Center for Nonlinear and Complex Systems, Como, Italy, with the Max-Planck-Institute of the Physics of Complex Systems, Dresden, Germany, and with the University of Ulm, Ulm, Germany. His research interests were focussing on non-linear dynamics and quantum chaos. After that he joined automotive industry, working on the research and development of intelligent optical systems for safety applications. He is currently Algorithm Group Leader at IEE S. A., Contern, Luxembourg. In the Advanced Engineering Department he is responsible for the development of algorithms for 3D camera based machine vision applications in automotive safety, advanced building security, and automation.

Björn Ottersten was born in Stockholm, Sweden, 1961. He received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986. In 1989 he received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA. Dr. Ottersten has held research positions at the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, the Katholieke Universiteit Leuven, Leuven, and the University of Luxembourg. During 96/97 Dr. Ottersten was Director of Research at ArrayComm Inc, a start-up in San Jose, California based on Ottersten's patented technology. He has co-authored journal papers that received the IEEE Signal Processing Society Best Paper Award in 1993, 2001, and 2006 and 3 IEEE conference papers receiving Best Paper Awards. In 1991 he was appointed Professor of Signal Processing at the Royal Institute of Technology (KTH), Stockholm. From 1992 to 2004 he was head of the department for Signals, Sensors, and Systems at KTH and from 2004 to 2008 he was dean of the School of Electrical Engineering at KTH. Currently, Dr. Ottersten is Director for the Interdisciplinary Centre for Security, Reliability and Trust at the University of Luxembourg. Dr. Ottersten has served as Associate Editor for the IEEE Transactions on Signal Processing and on the editorial board of IEEE Signal Processing Magazine. He is currently editor in chief of EURASIP Signal Processing Journal and a member of the editorial board of EURASIP Journal of Applied Signal Processing. Dr. Ottersten is a Fellow of the IEEE and EURASIP. In 2011 he received the IEEE Signal Processing Society Technical Achievement Award. He is a first recipient of the European Research Council advanced research grant. His research interests include security and trust, reliable wireless communications, and statistical signal processing.