Real-time Hybrid ToF Multi-Camera Rig Fusion System for Depth Map Enhancement

Frederic Garcia[12*] Djamila Aouada[1] Bruno Mirbach[2] Thomas Solignac[2]
Björn Ottersten[1]

[2]Advanced Engineering Department, IEE S.A.
{frederic.garcia, bruno.mirbach, thomas.solignac}@iee.lu

[1]Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
{djamila.aouada, bjorn.ottersten}@uni.lu

Abstract

We present a full real-time implementation of a multilateral filtering system for depth sensor data fusion with 2-D data. For such a system to perform in real-time, it is necessary to have a real-time implementation of the filter, but also a real-time alignment of the data to be fused. To achieve an automatic data mapping, we express disparity as a function of the distance between the scene and the cameras, and simplify the matching procedure to a simple indexation procedure. Our experiments show that this implementation ensures the fusion of 3-D data and 2-D data in real-time and with high accuracy.

1. Introduction

Time-of-Flight (ToF) cameras are becoming very competitive compared to other 3-D sensing modalities. Indeed, nowadays ToF systems are as affordable and compact as stereo vision systems. Their major advantage is their ability to provide an entire depth map at a high frame rate and independently of scene illumination. There are however two main drawbacks that are today restraining the potential of ToF technology; namely, low resolution of ToF data, and their high noise level. In applications where the limited resolution of a ToF camera is critical, research efforts are directed towards sensor fusion [6, 7, 8, 9, 10, 14, 15, 20, 21], where ToF data is combined with data provided by a conventional 2-D video camera. A successful hybrid ToF multi-camera rig, deployed in real-world applications, needs not only to provide good quality depth maps, but also to capture this data in real-time. For this reason the first proposed fusion efforts were not suitable, as they were based on Markov Random Fields, which are known to be computationally demanding [7, 10]. Another approach based on the bilateral filter was proposed by Kopf et al. [15]. It is referred to as the joint bilateral upsampling (JBU) filter. Until recently, the bilateral filter and its extensions were computationally expensive; their conventional brute-force implementation has a computational complexity of the order of $O(n^2)$ per output pixel, where n is the filter radius. Recent proposals for the bilateral filter implementation have made it now suitable for real-time applications [11, 16, 17, 19].

This improvement applies to all extensions of the JBU, that are used in fusing ToF data with 2-D data [6, 9]. However, while it is true that these 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, mapping the distance measurements from the ToF camera onto the colour camera is a straightforward procedure which would result 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. We want to assign to each of the high-resolution 2-D pixels 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 go beyond the alignment assumption, and propose a real-time implementation of a full system for hybrid ToF multi-camera rig data fusion. In other words, we propose an original technique for real-time alignment of

*F. Garcia was supported by the AFR Grant Scheme (Aides à la Formation-Recherche), managed by the National Research Fund of Luxembourg (FNR). AFR Grant: TR-PHD BFR08-120.

the captured data by the ToF camera and the 2-D camera, followed by a real-time fusion of these data. For the data fusion step, we consider the pixel weighted average strategy (PWAS) filter [9] as an improved version of the JBU algorithm. To that end, we have adapted the method proposed by Yang et al. [19] to make the PWAS filter perform in real-time.

The organization of the paper is as follows: In Section 2, we introduce the PWAS filter and its new formulation for an implementation in real-time. In Section 3, we explain the problem of the dependence of disparity on the distance between the scene and the camera. Solving this problem starts by mapping the image coordinates relative to each camera to a unified reference frame as presented in Section 4. We give our proposed algorithm for the data matching procedure in Section 5. In Section 6, we present our experimental results. Finally, in Section 7, we give our conclusions and perspectives.

2. Pixel Weighted Average Strategy (PWAS)

The JBU is a multilateral filter that enhances depth maps using a spatial weighing term $f_s(\cdot)$ applied on the pixel position, and a range weighing term $f_r(\cdot)$ applied on the pixel value. Both weighing terms are generally chosen to be Gaussian functions. Thus, this filter adjusts the edges in the depth map $\mathbf{R}$ to the edges in the 2-D image $\mathbf{I}$. The filtering can only be applied once the two images are aligned and mapped to the same reference frame, that we refer to as $\mathcal{C}$. Applying the JBU on the resulting images $\mathbf{I}_{\mathcal{C}}$ and $\mathbf{R}_{\mathcal{C}}$ may be achieved as follows:

$$\mathbf{J}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot f_r\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big)\cdot \mathbf{R}_{\mathcal{C}}(\mathbf{q})}{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot f_r\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big)}, \tag{1}$$

where $N(\mathbf{p})$ is the neighbourhood at the pixel indexed by the vector $\mathbf{p} = (i,j)^T$, where i corresponds to rows and j corresponds to columns. The resulting filtered image $\mathbf{J}$ is an enhanced version of $\mathbf{R}$, that presents less discontinuities, and a significantly reduced noise level. Nevertheless, the direct application of the JBU filter for depth enhancement may introduce unwanted artefacts such as texture copying and edge blurring. In order to deal with these artefacts, improved versions of JBU have recently been proposed by Chan et al. in 2008 [6] and Garcia et al. in 2010 [9]. While both approaches are good solutions, the filter in [6] requires some parameter tuning. Moreover, the PWAS filter in [9] copes well with inaccurate edges as it includes a new factor $Q(\cdot)$, named credibility map, to the kernels in (1). This factor explicitly accounts for the unreliability of distance measurements along the edges. We opted for this most recent JBU extension as it outperforms the alternative depth fusion filters [9]. The PWAS filter is thus expressed as:

$$\mathbf{J}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot f_r\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big)\cdot Q_{\mathcal{C}}(\mathbf{q})\cdot \mathbf{R}_{\mathcal{C}}(\mathbf{q})}{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot f_r\big(\mathbf{I}_{\mathcal{C}}(\mathbf{p}),\mathbf{I}_{\mathcal{C}}(\mathbf{q})\big)\cdot Q_{\mathcal{C}}(\mathbf{q})}, \tag{2}$$

where $Q_{\mathcal{C}}$ is the mapped credibility map that results from weighing the gradient of the original depth map, *i.e.,* $Q(\mathbf{q}) = f_t(|\nabla\mathbf{R}(\mathbf{q})|)$. Similarly to [9], we choose $f_s(\cdot)$, $f_r(\cdot)$, and $f_t(\cdot)$ to be Gaussian functions with variances σ_s, σ_r, and σ_t, respectively. The value of σ_s is chosen equal or larger than the scale factor between the low-resolution depth map and the 2-D image. σ_r is the mean of the gradient of the 2-D image, and σ_t is the mean noise in the ToF camera measurements.

For a real-time implementation of the PWAS filter, we propose to adapt the method in [19], as it has been shown to outperform state-of-the-art methods for accuracy, speed and memory consumption. We thus proceed by defining two mappings $G^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}(\cdot)$ and $H^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}(\cdot)$ for a fixed value of the 2-D image $\mathbf{I}_{\mathcal{C}}$ at the pixel $\mathbf{p}$, such that:

$$\begin{aligned} G^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}: &\quad \mathbf{q} \longmapsto f_R\big(\mathbf{I}_{\mathcal{C}}(\mathbf{q}),\mathbf{I}_{\mathcal{C}}(\mathbf{p})\big)\cdot Q_{\mathcal{C}}(\mathbf{q})\cdot \mathbf{R}_{\mathcal{C}}(\mathbf{q}),\\ H^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}: &\quad \mathbf{q} \longmapsto f_R\big(\mathbf{I}_{\mathcal{C}}(\mathbf{q}),\mathbf{I}_{\mathcal{C}}(\mathbf{p})\big)\cdot Q_{\mathcal{C}}(\mathbf{q}). \end{aligned} \tag{3}$$

We then may rewrite (2) as follows:

$$\mathbf{J}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot G^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}(\mathbf{q})}{\sum_{\mathbf{q}\in N(\mathbf{p})} f_s(\mathbf{p},\mathbf{q})\cdot H^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}(\mathbf{q})}. \tag{4}$$

We note that $f_S(\mathbf{p},\mathbf{q})$ is a function of the difference $(\mathbf{p}-\mathbf{q})$. We hence write (4) as:

$$\mathbf{J}(\mathbf{p}) = \frac{\Big(f_s * G^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}\Big)(\mathbf{p})}{\Big(f_S * H^{\mathbf{I}_{\mathcal{C}}(\mathbf{p})}\Big)(\mathbf{p})}, \tag{5}$$

where $*$ denotes the convolution between functions. We note that the downsampling of the data to be filtered does not introduce significant errors [16]. It actually ensures good memory and speed performances. Moreover, it was shown in [16] that using a sampling rate proportional to the Gaussian bandwidth provides a consistent approximation. The reformulation of the filter in (5) using two convolutions and their application on downsampled data enables its implementation to perform in real-time. However, we recall that this filtering operation requires perfectly aligned input images. We explain the necessity of this alignment in Section 3, and dedicate the rest of the paper to present our proposed automatic matching procedure.

3. Distance-dependent disparity

In general, the two reference frames of each individual camera constituting a hybrid ToF multi-camera rig are not

co-centric, *i.e.,* the centres of projection of each camera are displaced by a distance b, known as the *baseline* of the ToF multi-camera rig. Thus, the projections of a point P in space onto each camera image plane and with respect to each camera's principal point, differs by a distance called *binocular disparity*:

$$\rho = f\frac{b}{Z}. \tag{6}$$

In stereo systems, the disparity leads to the estimation of the distance Z at which the point P is located in the scene. However, this requires to solve the correspondence problem, *i.e.,* to find the feature-correspondence pairs. In contrast, in our case the problem is reversed due to the ToF camera, and Z is used to estimate the disparity ρ for each of the ToF camera pixels, which simplifies the mapping by avoiding demanding operations such as feature matching and image correlation.

The relationship between the Z measurements and the feature-correspondence pairs 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. By differentiating disparity ρ in (6) with respect to depth Z, we define the absolute disparity variation $\Delta\rho$ as a function of the absolute depth variation ΔZ, and obtain:

$$\Delta\rho = fb\frac{\Delta Z}{Z^2}. \tag{7}$$

We note that only in situations where the depth variation of the object in the scene ΔZ is small enough compared to the squared distance Z^2 from the object to the system, the disparity ρ 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 a SwissRanger™ ToF camera in their ToF multi-camera rig [6, 12, 13]. The rather small field of view (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.

4. Unified reference frame

A hybrid ToF multi-camera rig delivers the two raw images $\mathbf{I}$ and $\mathbf{R}$ related to each camera's reference frame, that we refer to as $\mathcal{A}$ for the 2-D camera and $\mathcal{B}$ for the ToF camera. To achieve the low-level data matching required for data fusion, we proceed by transforming the image coordinates related to each camera reference frame to the unified reference frame $\mathcal{C}$. This transformation allows us to establish a mapping of the data from both sensors on a common coordinate grid on $\mathcal{C}$. As a consequence of this mapping process, the indexed data at the end of the mapping matches pixel to pixel, *i.e.,* the mapped images are pixel aligned, and ready to be fused. Our mapping procedure results from the following steps:

1. Distortion correction. This is a classical first step in system calibration. It consists in correcting the raw distorted images $\mathbf{I}$ and $\mathbf{R}$ according to the extrinsic and intrinsic camera parameters that are to be determined. To that end, we can resort to classical calibration tools such as Bouguet's toolbox for Matlab [5] or image processing tools such as those included in Intel's computer vision library *OpenCV* [1]. Once those parameters are known, we correct the distortion for the 2-D image coordinates (u_d, v_d) and ToF image coordinates (x_d, y_d), and proceed with the resulting undistorted image coordinates (u_u, v_u) and (x_u, y_u), respectively.

2. Binocular disparity suppression. Our main contribution deals with the variation of disparity in the case where it cannot be modelled as a constant value as previously presented in Section 3. Our approach consists of suppressing the disparity by means of ToF camera measurements.

We transform the value of the depth map $\mathbf{R}$ at the pixel location (x_u, y_u), noted as $\mathbf{R}(x_u, y_u)$, into the Z coordinate of the corresponding pixel using the undistorted image coordinates (x_u, y_u), as follows:

$$Z = \mathbf{R}(x_u, y_u) \cdot \frac{f}{d(x_u, y_u)}, \tag{8}$$

where f is the ToF camera focal length, and

$$d(x_u, y_u) = \sqrt{f^2 + {x_u}^2 + {y_u}^2} \tag{9}$$

is the radial distance of the pixel coordinates to the projection centre in 3-D.

The Z coordinate of the pixel at location (x_u, y_u) allows to compute the disparity shift of this pixel using (6), *i.e.,*

$$\begin{pmatrix} x'_u \\ y'_u \end{pmatrix} = \begin{pmatrix} x_u - \rho_x \\ y_u - \rho_y \end{pmatrix} = \begin{pmatrix} x_u \\ y_u \end{pmatrix} - \frac{f}{Z}\begin{pmatrix} b_x \\ b_y \end{pmatrix}. \tag{10}$$

Thereby we have considered the paired sensors to be installed on the same plane and separated along the x and y axes, *i.e.,* $b = b_x \cdot \vec{e}_x + b_y \cdot \vec{e}_y + 0 \cdot \vec{e}_z$, where $\vec{e}_x$, $\vec{e}_y$ and $\vec{e}_z$ are respectively the unit vectors along the x, y and, z axes of the ToF reference frame $\mathcal{B}$. Consequently, the binocular disparity in (6) is decomposed into two components as $\rho = \rho_x \cdot \vec{e}_x + \rho_y \cdot \vec{e}_y$. It is important to note that ρ is a function of the distance Z, and thus not constant for all pixel locations. As a result of this image coordinates displacement, the system behaves as a monocular system where the two

camera reference frames of the individual cameras are co-centric. Consequently, the binocular disparity ρ gets suppressed. The values of the depth map $\mathbf{R}$ are, however, not invariant under this disparity shift, but may be recomputed according to (see equations (8) and (9)):

$$\mathbf{R}'(x'_u, y'_u) \quad = \quad Z \cdot \frac{d(x'_u, y'_u)}{f} \,. \tag{11}$$

3. Projective transformation. The last step concerns the transformation of the resulting image coordinates from the previous steps to the unified reference frame $\mathcal{C}$. This operation describes a mapping from plane to plane which can be solved by a projective transformation [2, 18] when both reference systems have the same centres. This latter condition has been fulfilled by the disparity correction in the previous transformation step.
Although we are considering a test rig with a baseline shift of b along the x and y directions, our approach may be easily generalised to the case of a shift in any direction between the sensors.

5. Data matching

The proposed data matching results from mapping the image coordinates from each individual camera to the unified reference frame $\mathcal{C}$. The relationship between the raw images and the mapped ones can be represented by an array that associates each pixel coordinates from the recorded image to the corresponding new location after the mapping. This associative array or look-up table (LUT) can be computed offline in order to reduce the complexity of the mapping procedure to a single indexing operation and leading to real-time implementation. In what follows we propose an efficient mapping technique that tackles the binocular disparity for any setup and scenario.

5.1. 2-D camera LUT

In order to determine the LUT $\mathbf{L}_{\mathcal{AC}}$ that associates the recorded 2-D images $\mathbf{I}$, relative to $\mathcal{A}$, to the unified reference frame $\mathcal{C}$, we perform the mapping procedure presented in Section 4 on the 2-D image coordinates (u_d, v_d). We start by defining a mesh grid $\mathbf{\Psi} = \{(p_{ij}, q_{ij}),\, i = 1, \ldots, M;\, j = 1, \ldots, N\}$, 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 fusion, which is intended for enhancing the ToF depth map up to the same 2-D camera resolution. We proceed by placing the mesh grid onto the mapped image coordinates and perform a nearest neighbour search on which the 2-D image $\mathbf{I}$ will be mapped, resulting in the rectified 2-D image $\mathbf{I}_{\mathcal{C}}$ used in (2) and (4). In other words, for each pixel (i, j) in $\mathbf{I}_{\mathcal{C}}$, the corresponding pixel (m, n) in $\mathbf{I}$ is determined by the closest coordinates (u_{mn}, v_{mn}) to the pixel coordinates (p_{ij}, q_{ij}) defined by the mesh grid $\mathbf{\Psi}$. We thus may define the mapping $(i, j) \mapsto (m, n) = \mathbf{L}_{\mathcal{AC}}(i, j)$, as $\mathbf{L}_{\mathcal{AC}}(i, j) = \arg\min_{(m,n)} \|(p_{ij}, q_{ij})^T - (u_{mn}, v_{mn})^T\|_2$. The stored LUT $\mathbf{L}_{\mathcal{AC}}$ allows to generate the new mapped image as follows $\mathbf{I}_{\mathcal{C}}(i, j) = \mathbf{I}(\mathbf{L}_{\mathcal{AC}}(i, j))$, for all i, j.

We recall that the mesh grid $\mathbf{\Psi}$ corresponds to the coordinates on the unified reference frame $\mathcal{C}$. We will thus use the same mesh grid for the second part of the mapping as presented in Section 5.2.

5.2. 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 $\mathbf{L}_{\mathcal{BC}}$. We use $\mathbf{\Psi}$ to define a new depth map $\mathbf{R}_{\mathcal{C}}$ from $\mathbf{R}'$, the recorded image relative to $\mathcal{B}$ corrected in disparity. 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 interpolated. At the end of the mapping process, both resulting images $\mathbf{I}_{\mathcal{C}}$ and $\mathbf{R}_{\mathcal{C}}$ generated from their respective $\mathbf{L}_{\mathcal{AC}}$ and $\mathbf{L}_{\mathcal{BC}}$ LUTs are pixel aligned. Nevertheless, $\mathbf{L}_{\mathcal{BC}}$ that generates $\mathbf{R}_{\mathcal{C}}$ is distance dependent. Due to the disparity suppression realised in the second step of our mapping procedure (see Section 4), the resulting 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 at each ToF frame acquisition; however, that implies a high computational time, and consequently, it will not be viable if a real-time performance is required. Indeed, the offline 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. Therefore, in order to achieve a 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 ρ_k, corresponding to a plane at a fixed distance $Z_k = f \cdot \frac{|b|}{|\rho_k|}$ to the system. We choose the discrete disparities as multiples of the pixel size δ in the mapped image $\mathbf{R}_{\mathcal{C}}$, *i.e.,* $\rho_k = \delta s_b k, k = 0, \ldots, K-1$ where $s_b = b/|b|$ is the unit vector of the baseline shift. Dividing the Z range of the

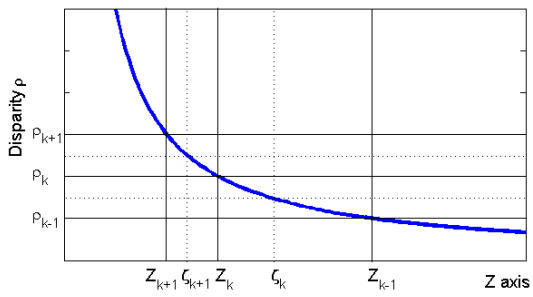


Figure 1. Z range of the ToF camera divided into K intervals $[Z_{k+1}, Z_k]$ defined by equidistant disparity values $\rho_k = k \times \rho$. Within each interval, the disparity ρ varies less than 1 pixel size δ.

ToF camera into K intervals $[\zeta_{k+1}, \zeta_k]$ around Z_k with

$$\begin{aligned} \zeta_0 &= \infty \\ \zeta_k &= f \cdot \frac{|b|}{(k-\frac{1}{2})\delta},\ k = 1, \ldots, K, \end{aligned} \tag{12}$$

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 ρ_k, with an error less than $\delta/2$, *i.e.,* half the size of a pixel in the high-resolution 3-D image $\mathbf{R}_\mathcal{C}$ (Figure 1). 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 $K \geq f \cdot \frac{|b|}{Z_{min}\delta} + \frac{1}{2}$. The mapping is performed as follows:

for $i = 1$ to N **do**
  for $j = 1$ to M **do**
    $k = K$
    while $(k > 0)$ **and**
      $(Z(\mathbf{L}_{\mathcal{BC},k}(i,j)) > \zeta_k)$ **do** $\{Z_k$ interval$\}$
      $k \leftarrow k - 1$
    end while
    if $(k < K)$ **and**
      $(Z(\mathbf{L}_{\mathcal{BC},k}(i,j)) < \zeta_{k+1})$ **then** {Occlusion handling}
      $k \leftarrow k + 1$
    end if
    $\mathbf{R}_\mathcal{C}(i,j) = \mathbf{R}'(\mathbf{L}_{\mathcal{BC},k}(i,j))$ {Mapping}
  end for
end for

where Z denotes the image of Z values calculated from the depth map $\mathbf{R}$. This mapping procedure allows the low-resolution depth map $\mathbf{R}'$ to be mapped in real-time to a depth map $\mathbf{R}_\mathcal{C}$, where each pixel matches a pixel in the already mapped $\mathbf{I}_\mathcal{C}$ image. Moreover, for a faster overall performance, we avoid downsampling each ToF image $\mathbf{R}_\mathcal{C}$ before filtering with (2) as suggested at the end of Section 2. Instead, we automate its downsampling by mapping $\mathbf{R}'$ using a downsampled LUT $\mathbf{L}_{\mathcal{BC}}$. We note that the downsampling of the ToF camera LUT does not apply to the 2-D LUT $\mathbf{L}_{\mathcal{AC}}$ as the high-resolution 2-D images are required for the final interpolation step.

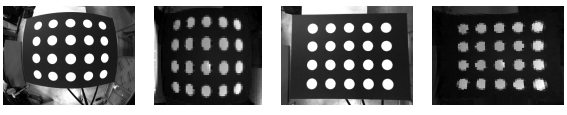
6. Experimental results

We herein evaluate the three main aspects related to our full real-time implementation of a hybrid ToF multi-camera rig for depth enhancement; namely, data matching, data fusion and computation time. 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. For our own recordings, we have used a hybrid ToF multi-camera rig composed of a 3D MLI Sensor™ from IEE S.A. [3], and a Flea®2 video camera from Point Grey Research, Inc. [4]. The resolution of the 3D MLI Sensor™ is (56×61) pixels, with a measurement range of 7.5 m and a frame rate of 11 frames per second (fps). The Flea®2 camera provides a resolution of (648×488) pixels, and a frame rate of 80 fps. 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 whole fusion setup in C, and we ran the experiments on a Pentium IV, 2.66 GHz with 1 GB of RAM.

6.1. Data matching

In order to analyse the data matching step, we have considered three different test cases in which we recorded the calibration pattern displaced around the FOV of the sensing system, and at different depths. In the first test case, the pattern is roughly centred in the system FOV and parallel to the system at a distance of 683 mm, as shown in Figure 2. In the second test case, the pattern is shifted to the corner of the FOV and moved away from the system to a distance of 1530 mm. In the third test case, the pattern is tilted in order to cover a depth range from 900 mm to 1530 mm. We quantify the accuracy of the mapping procedure by computing the root mean square error (RMSE) between the computed centroid coordinates related to each dot of each

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



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

Figure 2. First test case for data matching. (a) 2-D acquisition. (b) ToF acquisition. (c) 2-D mapped. (d) ToF mapped.

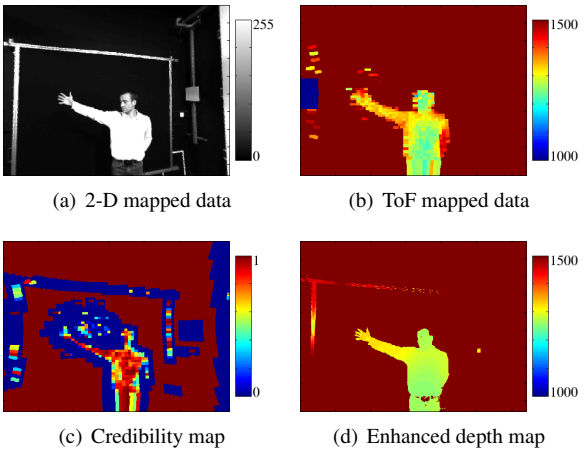
Table 1. Data matching error for the three test cases. The table compares the RMSE (in pixels) over 20 control points, separately computed for x and y pixel coordinates, between our mapping procedure and the mapping using a projective transformation.

Test cases		1	2	3
RMSE with a unique proj. transf. at 1.5m	x	7.52	1.33	2.59
	y	1.45	1.88	1.42
RMSE with a different proj. transf. for each test case	x	1.29	1.33	3.52
	y	1.48	1.88	1.42
RMSE with the proposed mapping procedure	x	2.14	1.56	1.47
	y	1.40	1.84	1.43

pair of 2-D and ToF mapped amplitude images [2]. Furthermore, we compare our proposed mapping method with a classical mapping using a simple projective transformation. To that end, we first calculate the projective transformation at the distance in which the system was calibrated, *i.e.,* 1.5 m. We then calculate a different projective transformation for each test case. Table 1 reports the RMSE results for all the tests. We find that the error when mapping using a projective transformation is small if it is computed at the same distance of the target and if the pattern is parallel to the sensing system. As soon as the distances between the target and the projective transformation are not coincident, the error increases. The error increases again when the target is tilted. The evaluation results for our mapping method show a consistent error of about 2 mapped image pixels, or less. This observation confirms that the proposed method accurately adapts to the distance-dependent disparity explained in Section 3. However, the error slightly increases to 2.14 pixels in the specific case where the target is planar. This is caused by the approximation, given in (12), of Z_k by the interval $[\zeta_{k+1}, \zeta_k]$. We note that all RMSE values also include small inaccuracies introduced by the centroid estimation and the calibration step. We further evaluate our mapping procedure on the scenes *Art*, *Books*, and *Moebius*, provided by the Middlebury dataset (Figure 5). The Middlebury dataset also provides the required information to generate ground truth depth maps from the provided disparity maps, such as the baseline of the system they used or the minimum disparity value, 200 pixels. We therefore compare the mapped depth maps from *view 5* with the ground truths at *view 1* relative to the maximum distance in the scene determined by the minimum disparity value. We find a global RMSE of less than 0.15%. Notice that this measurement has been computed without considering the occlusion areas (see the third column of Figure 5). This result consolidates the above mapping experiments using a calibration pattern. Figure 5 presents the filter output that includes the proposed mapping procedure for data matching.

[2] Amplitude images result from the intensity reflected by the active illumination emitted by the ToF camera.

6.2. Data fusion



(a) 2-D mapped data (b) ToF mapped data

(c) Credibility map (d) Enhanced depth map

Figure 3. PWAS filtering of frame 31.

To illustrate the effectiveness of the real-time data fusion, we record a video sequence[3] of 138 frames using our test rig described above. The video sequence contains a person moving his arms around the FOV of the system. Figure 3 presents the PWAS filtering applied on frame 31. The enhanced depth map, Figure 3(d), has the same resolution as the 2-D image, Figure 3(a), with the depth edges accurately aligned according to the 2-D edges in regions of low credibility (see the contour of the hand in Figure 3(d)). When there is no contrast between foreground and background in the 2-D image, the filter adjusts the depth measurements within the credibility map boundaries, yielding, in some cases, to wrong distance measurements. This phenomenon occurs on the hair measurements of the person. Due to the low contrast between the background colour and the hair colour, the depth measurements get adjusted to the background value.

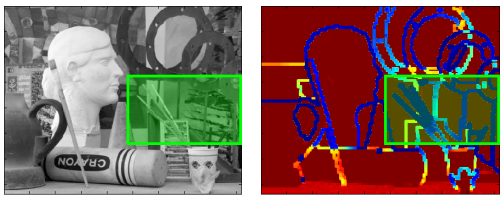
We chose the same *Art*, *Books*, and *Moebius*, scenes from the Middelbury data in order to compare the JBU performance against its extended version, the PWAS filter. To that end, we donwsampled the depth maps at *view 5* by a factor of 2, 4, and 8. We then mapped the downsampled depth maps to *view 1* and finally fused them with the provided corresponding high-resolution 2-D images at *view 1*. Table 2 reports the RMSE measure, between the JBU and the PWAS outputs, and the given ground truth. The RMSE measure has been computed without considering occlusion areas and unknown disparity pixels (3^{rd} column of Figure 5). Figure 5 shows the visual results for the case of a downsampling rate of 4. We notice that PWAS outperforms JBU in almost all cases. It is slightly worse when filtering such small objects that are completely assigned low credibility weights. In such cases, these pixels are adjusted to

[3] Video sequence provided as supplementary material.

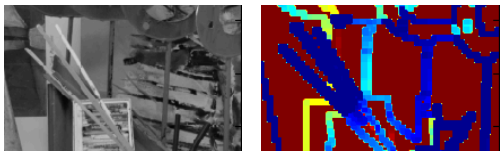
the distance value of the closest object with a similar colour value, as occurs in Figure 4. This assumption handles perfectly outliers or regions with unknown disparity.

Table 2. Comparison between the JBU and the PWAS filter outputs applied on the Middlebury scenes in Figure 5. s_r corresponds to the sampling rate. σ_s, σ_r, and, σ_t are the used variances for each weighing term in the JBU and PWAS expressions. The last two columns report the RMSE of the JBU and PWAS filters output against the depth map ground truth.

Scene	s_r	σ_s	σ_r	σ_t	RMSE JBU	RMSE PWAS
Art	2x	3.00	0.03	117.25	0.016	0.017
	4x	5.00	0.03	136.78	0.019	0.022
	8x	9.00	0.03	170.15	0.023	0.025
Books	2x	3.00	0.03	69.27	0.011	0.010
	4x	5.00	0.03	86.23	0.016	0.010
	8x	9.00	0.03	85.77	0.019	0.014
Moebius	2x	3.00	0.03	76.74	0.010	0.009
	4x	5.00	0.03	78.85	0.010	0.006
	8x	9.00	0.03	93.94	0.014	0.009



(a) 2-D intensity image to be fused (b) Credibility map



(c) Zoom of green area in (a) (d) Zoom of green area in (b)

Figure 4. Case in which PWAS fails.

6.3. Computation time

Data matching for data captured by the 2-D camera is achieved by a single indexation step. For the ToF camera, the matching requires using the k^{th}-LUT, *i.e.,* $\mathbf{L}_{\mathcal{BC},k}$, that corresponds to the input pixel distance. In the worst case, *i.e.,* a depth map with all distance measurements equal to the maximum allowed distance, it would be necessary to go through all k steps and then, perform the indexation. However, in a real-world scenario, this is not always the case. The fast PWAS filter implementation requires 3 milliseconds per image on the system introduced above. If we process the current video sequence from a file stored beforehand on memory, the whole fusion system performs at 20 Hz. However, if we process the data online, the fusion is limited by the ToF camera frame rate, which is in this case 11 fps.

7. Conclusion and perspectives

We have presented a full real-time hybrid ToF multi-camera rig fusion system to enhance the low-resolution depth maps provided by ToF cameras. The whole computation time of our system is smaller than the data acquisition time of the ToF camera. We hence showed that we can provide enhanced 3-D videos up to the ToF camera frame rate, 11 fps in our case. This was achieved thanks to the good performance of our automatic mapping procedure combined with the real-time implementation of PWAS, the considered multi-lateral fusion filter. This low-level fusion performed by our test rig provided enhanced depth measurements, well adjusted to the 2-D guidance image, and with a considerably reduced noise level.

As a future research direction, we plan to improve the accuracy of the PWAS filter. Our experiments pointed out instances where the PWAS filter fails in avoiding some artefacts. For example, in the case where there is no contrast between foreground and background in the 2-D guidance image, as happened with the dark hair of the person in the video sequence used in Section 6, we realized that it would be more suitable to consider the real depth edges given by the ToF camera instead of those given by the guidance image. Another undesirable artefact was often due to using grayscale images rather than colour images, which caused to map distinct colours to the same values, and hence resulted in lost edges. It will therefore be more accurate to consider the three red, green, and blue colour components, while always keeping in mind the necessity of a real-time implementation.

References

[1] *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media; 1st edition (September 24, 2008). 3

[2] *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003. 4

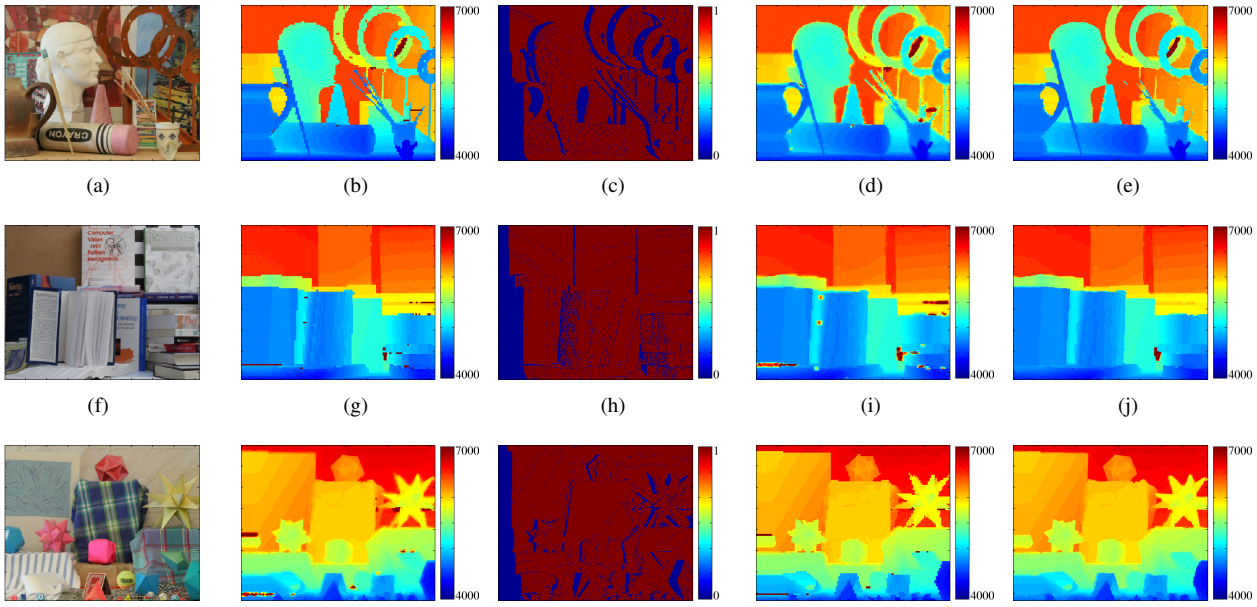
[3] IEE S.A., 3D MLI Sensor™. http://www.iee.lu, November 2010. 5

[4] Point Grey™, Flea®2. http://www.ptgrey.com/products/flea2/index.asp, November 2010. 5

[5] J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, November 2009. 3

[6] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A Noise-Aware Filter for Real-Time Depth Upsampling. In *Workshop on M2SFA2, ECCV*, 2008. 1, 2, 3

[7] J. Diebel and S. Thrun. An Application of Markov Random Fields to Range Sensing. In *NIPS*, pages 291–298. MIT Press, 2005. 1



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

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Figure 5. 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 input image. 2^{nd} col.: Depth map input (4x downsampled). 3^{rd} col.: Occlusion map. 4^{rd} col.: JBU filtering. 5^{th} col.: PWAS filtering.

[8] S. Foix, G. Alenya, and C. Torras. Lock-in Time-of-Flight (ToF) Cameras: A Survey . *Sensors Journal, IEEE* , 2011. 1

[9] 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, 2010. ICIP 2010. IEEE Conference on*, pages 2805 –2808, 2010. 1, 2

[10] S. Gloud, P. Baumstarck, M. Quigley, Y. N. Andrew, and K. Daphne. Integrating Visual and Range Data for Robotic Object Detection. In *Workshop on M2SFA2, ECCV*, 2008. 1

[11] M. Igarashi, M. Ikebe, S. Shimoyama, K. Yamano, and J. Motohisa. O(1) bilateral filtering with low memory usage. In *International Conference on Image Processing, 2010. ICIP 2010. IEEE Conference on*, pages 3301 –3304, 2010. 1

[12] 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'08*, pages 1–7, 2008. 3

[13] 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 *3DIM*, 2009. 3

[14] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-Flight Sensors in Computer Graphics. In *Eurographics 2009 - State of the Art Reports*, pages 119–134, 2009. 1

[15] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele. Joint Bilateral Upsampling. In *SIGGRAPH*, page 96, New York, NY, USA, 2007. ACM. 1

[16] S. Paris and F. Durand. A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach. In *International Journal of Computer Vision*, volume 81, pages 24–52. Kluwer Academic Publishers, 2009. 1, 2

[17] F. Porikli. Constant time o(1) bilateral filtering. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, 2008. 1

[18] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Canonical frames for planar object recognition. In *European Conference in Computer Vision, ECCV*, pages 757 –772. Springer Berlin / Heidelberg, 1992. 4

[19] Q. Yang, T. Kar-Han, and N. Ahuja. Real-time o(1) bilateral filtering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 557 –564, 2009. 1, 2

[20] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-Depth Super Resolution for Range Images. In *CVPR*, pages 1–8, 2007. 1

[21] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, 2008. 1