Machine Vision and Applications
DOI 10.1007/s00138-012-0428-2

ORIGINAL PAPER

Real-time free viewpoint video from a range sensor and color cameras

Stéphane Pelletier · Jeremy R. Cooperstock

Received: 20 August 2011 / Revised: 28 February 2012 / Accepted: 11 April 2012

Abstract We propose a method for computing a depth map at interactive rates from a set of closely spaced calibrated video cameras and a Time-of-Flight (ToF) camera. The objective is to synthesize free viewpoint videos in real-time. All computations are performed on the graphics processing unit, leaving the CPU available for other tasks. Depth information is computed from color camera data in textured regions and from ToF data in textureless ones. The trade-off between these two sources is determined locally based on the reliability of the depth estimates obtained from the color images. For this purpose, a confidence measure taking into account the shape of the photo-consistency score as a function of depth is used. The final depth map is computed by minimizing a cost function. This approach offers a significant time savings relative to other methods that apply denoising to the photo-consistency score maps, obtained at every depth, and importantly, still obtains acceptable quality of the rendered image.

Keywords Image-based rendering · Stereo · Range data · Sensor fusion · Graphics processors

1 Introduction

Camera arrays are increasingly employed in digital imaging and computer vision applications to capture images of a scene from different perspectives. Scene depth information can often be retrieved from this data and exploited in conjunction with the original color images in order to build partial textured models of scene objects. Such models have the potential to support a wide range of activities in entertainment, surveillance, and education. For example, they can be employed to produce input data for three-dimensional television (3DTV) or synthesize arbitrary views of a scene in free viewpoint television (FTV).

Many techniques have been proposed for obtaining scene depth information from color images. This information often takes the form of a *depth map*, which provides, for each pixel of a reference camera, the distance from the camera to the corresponding point in the scene. Since such methods only rely on visual cues to infer scene geometry, they often yield depth maps of poor quality in textureless or repetitive pattern regions.

To overcome these difficulties, one can employ active range sensing devices such as Time-of-Flight (ToF) cameras. These cameras produce a depth map corresponding to their field of view in real-time by measuring the time delay between the emission of an infrared light signal and its reception at every pixel. Although most ToF cameras suffer from high noise levels and offer a lower resolution than color cameras, they can produce good depth estimates in textureless regions. Recently, a few methods have been proposed for combining depth estimates obtained from a set of calibrated color cameras and a ToF camera. The emphasis of these methods is on exploiting the complementarity of the two data sources in order to produce better quality depth maps. However, we demonstrate that in many view interpolation scenarios, ToF data are actually not necessary to obtain increased visual quality in textureless regions. Indeed, regularization combined with proper weighting of the depth values computed using photo-consistency constraints can often produce results of comparable quality when the interpolated viewpoint is reasonably close to the position of the cameras.

S. Pelletier (✉) · J. R. Cooperstock
Department of Electrical and Computer Engineering,
McGill University, Montreal, QC, Canada
e-mail: stephane.pelletier@mail.mcgill.ca

J. R. Cooperstock
e-mail: jer@cim.mcgill.ca

Published online: 05 May 2012

Our work focuses on interactive applications such as remote medical training sessions, in which the students may view a cadaveric dissection from arbitrary viewpoints located above the surgical table. In this case, real-time performance is also of paramount importance.

The contribution of this article is a technique for computing a depth map at interactive rates from several calibrated video cameras and a ToF camera. In particular, we present a method for computing a confidence value for each depth estimate that can exploit the capabilities of modern graphics processing units (GPUs) available on the consumer market. Although a ToF camera does not always increase visual quality, we show that it can be exploited to accelerate the computation of scene depth maps, which proves useful in applications where real-time performance is important.

2 Literature review

Image-based rendering (IBR) synthesizes the image seen by a virtual camera placed in the vicinity of the recording cameras. When scene geometry is known accurately, high-quality images can be synthesized provided that each scene point is visible by at least one camera [6]. Geometric models can be computed by employing stereo methods [20] or multiview stereo algorithms [22]. Although light field rendering methods [4] can synthesize scene views without reliance on geometric information, they require a large number of cameras that must be placed very close to one another, which limits the possibility of reconstructing the scene from significantly different viewpoints.

For IBR methods employing geometry, depth information may be recovered from color images using plane sweep algorithms [5], in which photo-consistency between the camera images is measured on planes in front of the virtual camera. Several techniques are available, ranging from the accurate but slow [25,31] to fast but less accurate [24]. Unfortunately, depth from stereo measurements obtained exclusively from color images are less accurate where texture is lacking.

New techniques, exploiting a ToF camera to obtain more accurate depth estimates in textureless regions, overcome this problem. Some techniques attempt to increase the resolution of ToF depth maps using one or two reference color images. In particular, Kuhnert and Stommel [17] describe a simple method in which depth values obtained from a photonic mixer device (PMD) sensor [16] are interpolated and then averaged with those obtained from stereo images. Diebel and Thrun [7] exploit the fact that depth discontinuities tend to align with color differences, using this knowledge to upsample the resolution of a ToF depth map. Yang et al. [28] employ bilateral filtering to improve an upsampled depth map acquired with a ToF camera, based on one or two high-resolution reference color images.

Other approaches use depth information obtained from a ToF device to improve depth from stereo methods. Gudmundsson et al. [12] propose a method in which a disparity estimate is derived from ToF depth measurements and then employed to initialize and constrain a stereo-matching algorithm applied to a pair of color images. Zhu et al. [30] optimize a cost function defined from depth measurements obtained from a ToF camera and a stereo algorithm. Two global weighting factors are employed to balance the contributions of these two data sources. As a subsequent enhancement, these authors extended their work to include temporal coherence in the cost function [29], thus gaining improvements in the computation of depth maps of dynamic scenes.

In all previous methods, local reliability of depth information obtained from the different sources is not taken into account. In particular, depth values in textureless regions are weighted similarly to those obtained in textured areas, which is suboptimal. To improve upon these methods, Bartczak and Koch [2] employ a photo-consistency cost that penalizes depth estimates in textureless regions. They compute a sparse depth map for every camera pair and combine these using a consensus measure to generate the final depth map. However, computation time required by the method is not discussed. Yang et al. [27] proposed a technique in which photo-consistency costs are weighted based on locally computed confidence values, and a separate confidence map is generated for the ToF sensor data based on the strength of the signal reflected by the objects. The resulting system is reported to achieve performance of 8 fps using only three-color cameras. However, their method does not take into account the shape of the photo-consistency cost function along the depth axis when computing confidence values. As we demonstrate later in Fig. 9, exploiting this information can yield reconstructed views of improved visual quality.

The previous approaches, although too slow for our medical application, nevertheless demonstrate the value of combining complementary data sources to obtain improved quality of image-based rendering from disparate viewpoints. However, when a sufficient number of color cameras are employed and when the desired view point is located close to these cameras, we shall demonstrate that ToF data are often unnecessary to increase this visual quality, even in textureless regions. Unlike other methods, the technique described below exploits such data in order to accelerate the computation of the interpolated views. It is of course desirable to obtain the most accurate reconstruction possible of arbitrary viewpoints, as attempted by the more computationally expensive techniques described above. However, that is not the purpose of the work described here. Our objective, rather, is to obtain a *reasonably* accurate reconstruction within tightly bounded time constraints, so as to facilitate viewer interaction with the rendered viewpoint.

3 Algorithm

Given a ToF camera and N regular color cameras positioned at different locations in a scene, we want to generate the depth map corresponding to one of the color cameras, henceforth called the reference camera. For our configuration, cameras will typically be organized as shown in Fig. 4. For this purpose, we denote the color images by $\{I_i\}_{i=1}^{N}$ and the ToF depth map by Z. In the following, we assume that the values in Z correspond to real scene distances obtained by processing the raw values returned by the range sensor. The nature of these computations, which may include scaling, shifting and other corrections, depends on the particular hardware employed and is beyond the scope of this paper. Image and scene coordinates are represented by two- and three-dimensional vectors, respectively. The RGB color value of I_i at point $\mathbf{q} \in \mathbb{R}^2$ is $I_i(\mathbf{q}) \in \mathbb{R}^3$ and the depth value of Z at $\mathbf{q}$ is $Z(\mathbf{q}) \in \mathbb{R}$. $P_i : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a linear operator that projects a point in the scene onto the image plane of the ith camera based on its extrinsic and intrinsic calibration parameters. Consequently, the color perceived by the ith camera for a visible point $\mathbf{p} \in \mathbb{R}^3$ in the scene is denoted by $I_i(P_i(\mathbf{p}))$.

The general approach of our method is as follows. First, we compute a depth map for the reference camera using a simple photo-consistency test between the color camera images. As we shall see, this initial depth map is generally noisy and does not yield interpolated images of good quality. For this reason, we also compute a *confidence map* that measures the reliability of the depth map at every point. We then combine the initial depth map, the confidence map and a warped version of the ToF depth map in order to synthesize the final depth map of the reference camera. The result is then employed to synthesize different views of the scene.

3.1 Initial depth map estimation

A point in the scene is considered to be photo-consistent when its perceived color is similar in all camera images in which the point is visible [15]. Assuming a point $\mathbf{p} \in \mathbb{R}^3$ in the scene is located in the *view frustum* of all cameras, we measure its photo-consistency with respect to a reference camera r as follows:

$$\Theta(\mathbf{p}) = \frac{1}{N} \sum_{i=1}^{N} \left\| I_i\Big(P_i(\mathbf{p})\Big) - I_r\Big(P_r(\mathbf{p})\Big) \right\|_1 . \qquad (1)$$

A low value of $\Theta(\mathbf{p})$ means that the colors perceived by the different cameras at point $\mathbf{p}$ are similar to that observed by the reference camera, which might be an indication that $\mathbf{p}$ belongs to a real surface in the scene. On the other hand, one has to be aware that high values of $\Theta(\mathbf{p})$ can be obtained for real surfaces that are occluded or non-Lambertian.

To determine scene depth at a particular pixel of the reference camera, a popular method consists of measuring photo-consistency at several locations along a line that passes through the camera center and this pixel on the image plane. Using 3D camera geometry [13], we select the set of points $\mathcal{S}(\mathbf{q})$ in the scene that project to pixel $\mathbf{q} \in \mathbb{R}^2$ in I_r, i.e.,

$$\mathcal{S}(\mathbf{q}) \triangleq \{\mathbf{p} \in \mathbb{R}^3 : P_r(\mathbf{p}) = \mathbf{q}\}. \qquad (2)$$

For every pixel $\mathbf{q}$ of the reference camera, we determine the point $U(\mathbf{q})$ in $\mathcal{S}(\mathbf{q})$ that maximizes photo-consistency, i.e.,

$$U(\mathbf{q}) \triangleq \underset{\mathbf{p} \in \mathcal{S}(\mathbf{q})}{\operatorname{argmin}}\, \Theta(\mathbf{p}). \qquad (3)$$

This point represents an initial guess for the surface location at $\mathbf{q}$. The associated depth $D(\mathbf{q})$ is obtained by computing the distance between $U(\mathbf{q})$ and the reference camera center $\mathbf{c}$, as follows:

$$D(\mathbf{q}) = \|U(\mathbf{q}) - \mathbf{c}\|_2 . \qquad (4)$$

Pixel depths can be computed efficiently in parallel, e.g., using plane-sweep algorithms, which are well-suited for GPU implementation. Such algorithms first define a set of parallel planes in front of the reference camera. Photo-consistency is then evaluated by projecting all camera images onto a given plane, repeating this process for all planes. For this purpose, the GPU's optimized texture-filtering capabilities can be employed to resample images onto the plane, and the z-buffer can be used to determine the depth at each pixel that maximizes the photo-consistency measure.

In general, the surface computed using Eq. (3) exhibits a significant amount of noise, which, in turn, degrades the quality of interpolated views. To illustrate this, Fig. 1 shows an example of view interpolation using a noisy depth map. Note that the depth map is only reconstructed for scene regions that are visible by all cameras. One can see that errors in the depth map yield noticeable artefacts in the reconstructed view, especially around the heart cavity.

To ameliorate this problem, the photo-consistency measure given by Eq. (1) can be modified to include points in the neighborhood of $\mathbf{p}$ that are located at the same depth, i.e., on the same plane. In practice, this is achieved by first evaluating photo-consistency over each plane and then averaging the values locally using a box or bilateral filter [24,26,27]. To deal with partial occlusions, some methods measure photo-consistency scores between the reference camera and each of the other cameras, and retain the best score obtained [2,27]. In this case, $N - 1$ photo-consistency maps must be filtered at each plane in order to remove noise. Since the computation of these maps and the denoising operation need to be performed sequentially, the rendering target of the GPU must be switched at least twice per plane. Even on modern GPUs, which offer multiple rendering targets (MRT) capabilities, there is a significant cost associated with such operations.

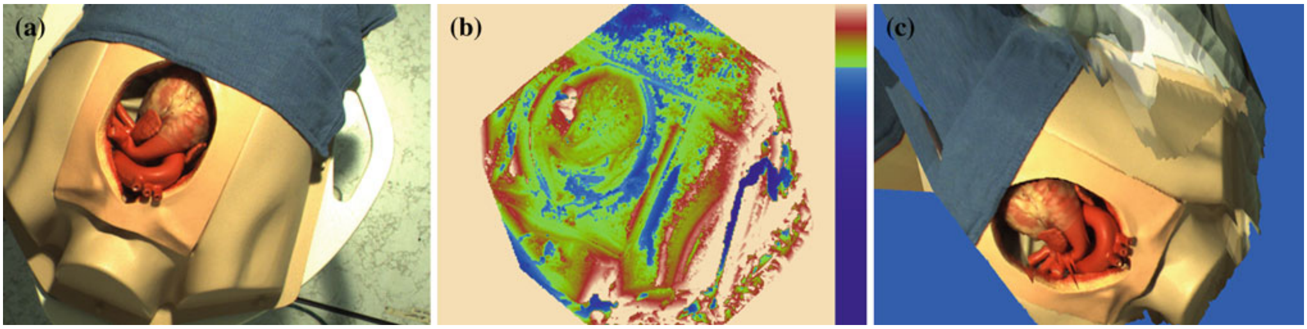

Fig. 1 View interpolation from a noisy depth map using the camera setup described in Sect. 4. **a** Image observed by the reference camera, i.e., camera 0 in Fig. 4. **b** Depth map computed by maximizing photo-consistency, as described in Sect. 3.1. **c** Image obtained by rendering the depth map shown in **b** from the perspective of camera 8 while projecting images from cameras 0 to 7 onto this map, as described in Sect. 3.4

Consequently, this solution can significantly increase the execution time of the plane-sweep algorithm when the number of planes is high. For this reason, our method computes the photo-consistency map using Eq. (1) only, i.e., without averaging, and then regularizes the resulting noisy depth map by employing the confidence map described in the next section.

3.2 Confidence map

We now describe a method for computing a confidence map to assess the reliability of depth estimates at every pixel. To satisfy the real-time requirements of our application, the computation of this map must exploit the parallel architecture of modern graphics cards. Although advanced approaches exist for assessing the accuracy of depth estimates [3], the confidence measure we propose is specifically designed to minimize the number of rendering target switches on the GPU, while discriminating between several shapes of the photo-consistency function $\Theta(\mathbf{p})$ along projection lines. To clarify this last point, Fig. 2 shows hypothetical values of $\Theta(\mathbf{p})$ along the projection lines associated with four pixels of the reference camera.In these plots, axis d represents the depth of $\mathbf{p}$, i.e., the distance between this point and the camera center. Our objective is to develop a confidence measure yielding decreasing values as we go from Fig. 2a to Fig. 2d. This requirement is based on the intuition that a narrow peak in the photo-consistency function provides a more reliable depth estimate than a broader peak, which, in turn, is more valuable than several narrow peaks located far apart from one another. This preference results from the fact that for performance reasons, we are limited to choosing a single depth candidate per pixel.

To define our confidence measure, let $\mathcal{T}_m(\mathbf{q}), m \in [1 \ldots M]$, be the set of points in $\mathcal{S}(\mathbf{q})$ whose distance from the initial surface point estimate $U(\mathbf{q})$ is at most s_m, i.e.,

$$\mathcal{T}_m(\mathbf{q}) \triangleq \{\mathbf{p} \in \mathcal{S}(\mathbf{q}) : \|\mathbf{p} - U(\mathbf{q})\|_2 < s_m\}. \tag{5}$$

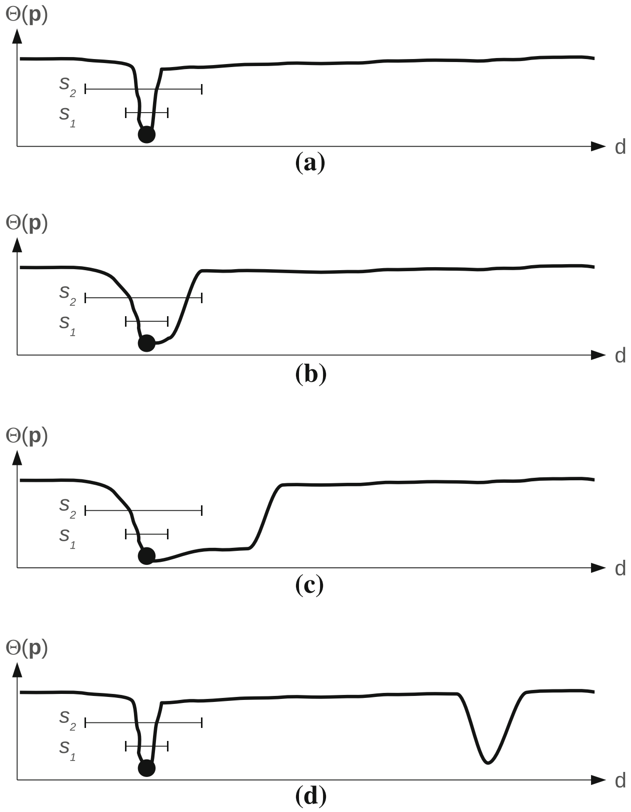

Fig. 2 Hypothetical values of $\Theta(\mathbf{p})$ along the projection lines associated with four pixels of a reference camera. Axis d represents the depth of $\mathbf{p}$. In each plot, the *black circle* indicates the position of the minimum of $\Theta(\mathbf{p})$ and the *horizontal segments* represent confidence intervals obtained from Eq. (5) using two hypothetical values of s_m, namely s_1 and s_2, where $s_1 < s_2$.

Also, let $\overline{\Theta}(\mathbf{q})$, $\underline{\Theta}(\mathbf{q})$ and $\Gamma_m(\mathbf{q})$ be the quantities defined as follows:

$$\overline{\Theta}(\mathbf{q}) \triangleq \max_{\mathbf{p} \in \mathcal{S}(\mathbf{q})} \Theta(\mathbf{p}), \tag{6}$$

$$\underline{\Theta}(\mathbf{q}) \triangleq \min_{\mathbf{p} \in \mathcal{S}(\mathbf{q})} \Theta(\mathbf{p}), \tag{7}$$

$$\Gamma_m(\mathbf{q}) \triangleq \min_{\mathbf{p}\in\mathcal{S}(\mathbf{q})\setminus\mathcal{T}_m(\mathbf{q})} \Theta(\mathbf{p}). \tag{8}$$

The *interval map* E_m is then defined as:

$$E_m(\mathbf{q}) \triangleq \begin{cases} \frac{|\Gamma_m(\mathbf{q})-\underline{\Theta}(\mathbf{q})|}{\overline{\Theta}(\mathbf{q})-\underline{\Theta}(\mathbf{q})}, & \overline{\Theta}(\mathbf{q}) - \underline{\Theta}(\mathbf{q}) > \beta \\ 0, & otherwise \end{cases}, \tag{9}$$

where β is a threshold employed to prevent textureless regions from artificially increasing interval map values. In practice, we found that setting this value to 0.05 yields good results. To clarify the meaning of interval maps, Fig. 3 shows three such maps computed for the depth map of Fig. 1b using $s_1 = 0.02$, $s_2 = 0.07$ and $s_3 = 0.12$. A value of $E_m(\mathbf{q})$ close to zero (black) means that the photo-consistency test yields at least one good match outside $\mathcal{T}_m(\mathbf{q})$. On the other hand, a value close to one (white) is obtained when no good matches exist outside this set. Intuitively, a high value of $E_m(\mathbf{q})$ is a good indication that the true surface point at $\mathbf{q}$ is located in $\mathcal{T}_m(\mathbf{q})$. In such cases, one can conclude that the initial surface estimate $U(\mathbf{q})$ and the associated depth estimate $D(\mathbf{q})$ are reliable, provided the associated interval s_m is relatively small.

Indeed, the value of s_m employed in the computation of E_m significantly impacts the meaningfulness of the latter. To illustrate this, two intervals, namely s_1 and s_2, are depicted around each global minimum in Fig. 2. Choosing s_1 to compute the interval map will result in a high value for the pixel of Fig. 2a and low values for the three other pixels. Consequently, depth values for these three last pixels will be considered equally unreliable. On the other hand, using s_2 will yield high interval map values for the pixels of Fig. 2a, b, and a low value for those of Fig. 2c, d. In this case, the depth estimates of Fig. 2a, b will be assumed to have similar reliability.

From these examples, one can see that high values obtained using large intervals do not provide a precise location of the peak, and thus, are less reliable than those obtained using small intervals. To take this observation into account, we first compute a set of M interval maps as described in Eq. (9). The confidence value $C(\mathbf{q})$ on the initial depth estimate $D(\mathbf{q})$ is then calculated as follows:

$$C(\mathbf{q}) \triangleq \frac{1}{a}\sum_{m=1}^{M}\frac{E_m(\mathbf{q})}{s_m}, \tag{10}$$

where $a \triangleq \sum_{m=1}^{M} s_m^{-1}$ is a normalizing constant. The values of E_m in Eq. (10) are scaled based on the size of the associated interval s_m, so that values obtained with large intervals have less impact on the confidence measurement. Consequently, the four pixels of Fig. 2 have distinct confidence values, as desired. The first image of Fig. 6 represents the confidence map resulting from a combination of the three interval maps of Fig. 3. As is evident, textureless regions have lower confidence.

The computation of the interval and confidence maps is performed using a plane-sweep strategy similar to that employed in the computation of the depth map in Sect. 3.1. Furthermore, since no switching of the rendering target needs to be performed at every depth plane, these maps can be computed very quickly on modern GPUs, even when many planes are used.

Values of M and s_m should be chosen carefully, taking into account the depth sampling density in the plane-sweep algorithm. In particular, small values of s_m and large values of M will not improve the reliability of the confidence map when the number of planes is low. Although increasing M may improve the distinction between different shapes of $\Theta(\mathbf{p})$, as illustrated in Fig. 2, this also impacts the computational burden associated with the algorithm. These issues are discussed further in Sect. 4.

As a final note, it is worth mentioning that other methods exist for measuring the reliability of the depth estimates obtained by maximizing photo-consistency. For example, Yang et al. [27] proposed a technique in which the confidence measurement at each pixel is based on the difference between the best photo-consistency score and all other scores, regardless of the depth at which each score is obtained. Similarly, Mühlmann et al. [18] measure the difference between the best and the nth best photo-consistency score at the pixel. Unlike our method, these approaches would not distinguish between the scenarios of Fig. 2b, d, which might impact the quality of the depth map, at least in theory. Furthermore, the reliability measurements obtained by the method of Mühlmann et al. are strongly affected by the density of points at which photo-consistency is measured, i.e., the number of planes used in the plane-sweep algorithm. Indeed, by increasing this density, many high photo-consistency scores will likely be obtained at depths in the vicinity of the global minimum, thereby reducing the reliability measurement. Although the value of n could be increased to overcome this problem, we argue that our interval-based method is simpler to use, since it is not affected by this sampling density.

3.3 Depth map regularization

The final depth map X is obtained by minimizing a cost function constructed from different penalty terms, each of which defines desirable properties of the solution. To encourage consistency of X at pixel $\mathbf{q}$ with the depth estimate $D(\mathbf{q})$ resulting from the photo-consistency test described in Sect. 3.1, we employ the following penalty term:

$$\Phi_1(X, \mathbf{q}) \triangleq \frac{C(\mathbf{q})}{2}\left[X(\mathbf{q}) - D(\mathbf{q})\right]^2. \tag{11}$$

The difference measured by this term is weighted according to the confidence map value $C(\mathbf{q})$ in order to adjust the strength of the consistency requirement at pixel $\mathbf{q}$ based on

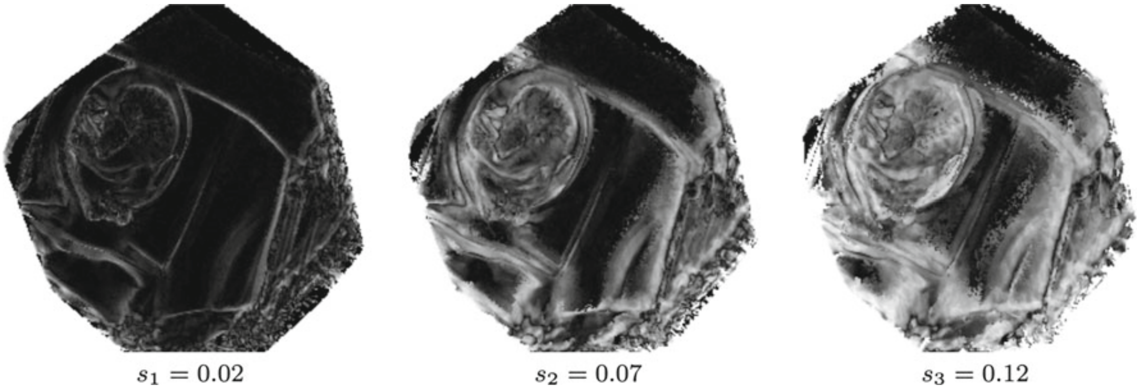

Fig. 3 Interval maps computed for the depth map of Fig. 1b using different interval values. *Black* and *white regions* correspond to values that are close to zero and one, respectively

the reliability of $D(\mathbf{q})$. We also want to encourage consistency with the ToF data, especially in regions where the confidence measurement values are low. For this purpose, we compute a warped depth map Z_W representing the ToF depth map Z viewed from the perspective of the reference camera. This is achieved by first rendering Z as a 3D surface in front of the ToF camera, and then calculating the values Z_W as follows:

$$Z_W(\mathbf{q}) = \|\mathbf{p} - \mathbf{c}\|_2 , \tag{12}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the point on the rendered surface that is visible at pixel $\mathbf{q}$ of the reference camera and $\mathbf{c}$ is the reference camera center, as before. From this warped depth map, we define the following penalty term:

$$\Phi_2(X, \mathbf{q}) \triangleq \frac{[1 - C(\mathbf{q})]}{2}\big[X(\mathbf{q}) - Z_W(\mathbf{q})\big]^2. \tag{13}$$

With our approach, priority is thus given to depth measurements obtained from photo-consistency constraints. In regions where the confidence map values are low, more weight is given to the range sensor data. However, no attempt is made to evaluate the reliability of the depth values obtained with the range sensor.

For our approach, we assume that the ToF camera is placed close to the reference camera. Otherwise, one should reduce the weight of the values in Z_W as a function of the orientation of the ToF camera with respect to the surface, reaching a minimum when the pixel rays of the camera are parallel to the surface [2].

Since depth values obtained from photo-consistency measurements are not always perfect and ToF data is generally very noisy, we also employ the following regularization term:

$$\Phi_3(X, \mathbf{q}) \triangleq \sum_{i=1}^{4} \phi\Big(R_i(X, \mathbf{q})\Big), \tag{14}$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a *potential function* and $\{R_i(X, \mathbf{q})\}_{i=1}^4$ are regularization filters such as first- or second-order derivatives of X at point $\mathbf{q}$ in four directions [21]. To preserve edges in the depth map, we employ the Huber function [14] as the potential function, which is defined as follows:

$$\phi(t) = \begin{cases} \frac{t^2}{2}, & |t| \leq \alpha \\ \alpha|t| - \frac{\alpha^2}{2}, & |t| > \alpha \end{cases} . \tag{15}$$

This function quadratically penalizes small discontinuities in the depth map, which are often associated with noise, whereas large discontinuities (actual edges) are only linearly penalized. The transition point between the quadratic and linear regions is determined by the parameter α. The higher this value, the more the Huber function becomes equivalent to a simple quadratic function.

The final depth map is computed by minimizing the cost function Φ obtained by combining the three penalty terms described previously:

$$\Phi(X) \triangleq \sum_{\mathbf{q}} \Phi_1(X, \mathbf{q}) + \Phi_2(X, \mathbf{q}) + \lambda\Phi_3(X, \mathbf{q}). \tag{16}$$

The first two terms favor solutions that are consistent with the depth maps D and Z_W, whereas the last term encourages piecewise smooth depth maps. The tradeoff between data consistency and smoothness is controlled by the value of λ. Note that this cost function does not explicitly take photo-consistency into account. Instead, this information is incorporated into the reconstruction process through depth map D and the associated confidence map C. The optimal solution will thus satisfy photo-consistency constraints in regions where the confidence level is high, whereas the regularization term will encourage depth continuity in other regions.

Since the cost function (16) is convex, it can be minimized by employing the multiplicative or additive half-quadratic reformulations introduced Geman and Reynolds [10] and Geman and Yang [11], respectively. In these reformulations, one only needs to minimize a series of least-square problems, which can be done using iterative techniques such as the method of conjugate gradients (CG) [23]. Since this method essentially involves two-dimensional convolutions and dot products between images, it can be implemented on modern graphics cards. In particular, dot products can be performed efficiently using the mipmapping hardware on GPUs.

Real-time performance requires that the number of CG iterations at each frame be limited. However, a sufficient number of iterations must be performed to achieve convergence; this number increases with the value of the regularization parameter λ. Significant savings are possible by re-using the depth map computed for the previous frame as the initial solution for the current frame. When the scene content does not change significantly from one frame to the next, fewer iterations are necessary to achieve convergence. Furthermore, since visual perception is less sensitive to details in areas of motion [19], potential visual artefacts resulting from incomplete convergence in such regions are less noticeable.

3.4 View synthesis

To synthesize a video frame, we render the regularized depth map as a 3D surface in front of the reference camera, while placing the viewing camera at the desired viewpoint. The rendering is performed using a computer graphics technique known as *projective texture mapping* [8], which projects camera images onto the surface as if by slide projectors. The parameters of such projections are determined by the calibration parameters of the cameras. More formally, the color of pixel $\mathbf{q}$ in the synthesized view I_s is calculated as follows:

$$I_s(\mathbf{q}) = \frac{\sum_{i=1}^{N} w_i \cdot I_i\Big(P_i(\mathbf{p})\Big)}{\sum_{i=1}^{N} w_i}, \tag{17}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the point on the rendered surface that is visible at $\mathbf{q}$. The contribution w_i of each camera is given by:

$$w_i = \exp(-a_i^2), \tag{18}$$

where a_i is the angular difference between the desired virtual camera position and that of camera i with respect to $\mathbf{p}$.

A few reasons justify the choice of this function for synthesizing the interpolated view. First, visual artefacts resulting from the projection of a single camera image onto a surface containing errors are less important when the viewpoint is located close to this camera. By increasing the contribution of cameras that are close to the viewpoint in Eq. (17), such artefacts become less visible. Also, camera images projected onto an erroneous surface do not align perfectly on this surface. In such cases, moving the viewpoint across the scene would result in sharp transition effects if the image from the closest camera alone was employed.

4 Experiments

We conducted a number of experiments to validate our algorithm, using a set of eight Basler acA1300-30gc color cameras, mounted on two concentric rings attached to the ceiling, as shown in Fig. 4. Although the cameras provide output of resolution 1,296×966 pixels, we operated over a subregion of 800×600 pixels for reasons related to bandwidth capacity of our processing architecture. A Baumer TZG01 ToF camera with a resolution of 176×144 pixels, placed in the middle of the rings, acquired low-resolution depth data. The optical axes of all cameras were roughly oriented towards the same point in the scene, namely the center of a small table top located 1.5 m below the cameras. This particular configuration was chosen for its suitability for our medical application scenario.

To determine the intrinsic and extrinsic parameters of the color cameras, and consequently, matrices $\{P_i\}_{i=1}^{N}$, we employed the ProCamCalib calibration software [1]. This takes as input a set of images of a square board containing several fiducial markers, which allows for high precision calibration [9]. Since our algorithm is quite sensitive to errors in the calibration parameters, we employed a 60 × 60 cm flat panel of wood onto which markers were directly printed, allowing us to achieve an average backprojection error of 0.2 pixels per calibration point. No color calibration was necessary. Since all of the color cameras are of the same model, we found that it was sufficient to set them to identical parameters (gain, exposure, aperture, etc.). Given the relatively low resolution of the ToF camera, its calibration required a different board with a large checkerboard pattern. To determine the relationship between the raw depth values returned by

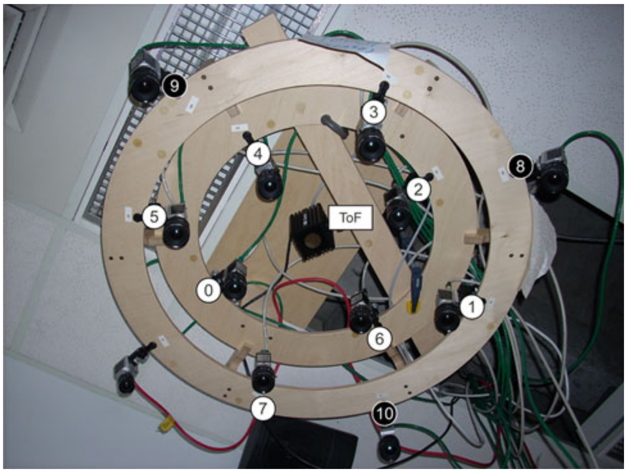

Fig. 4 Camera setup employed in the experiments. Color cameras (Basler acA1300-30gc) are numbered from 0 to 10. The square device in the middle is the ToF camera (Baumer TZG01). Cameras 0 to 7 are employed by the reconstruction algorithm, whereas cameras 8 to 10 are used exclusively to provide ground-truth images

the ToF camera and actual scene distances, we captured a depth image of a scene featuring objects at different depths. We then rendered this depth map as a 3D surface and projected the images from all color cameras onto this surface, as explained in Sect. 3.4. The depth map values were then scaled and shifted as needed to align the color images on the rendered surface properly.

To evaluate our approach, we performed reconstruction experiments using five different objects, the results of which are presented in Fig. 5. Our algorithm was implemented using the OpenGL Shading Language (GLSL) and all experiments were run on a machine equipped with a GeForce GTX465 graphics card. Images from the eight input cameras are uploaded from CPU to GPU memory each time a frame is generated. Three interval maps are employed when computing confidence maps. For the setup described above, our algorithm can compute depth maps with a resolution of 320 × 240 pixels at 30 fps using more than 80 depth layers. However, the synthesized views have the same resolution as the input images, namely 800 × 600 pixels. We found this depth map resolution to be sufficient for producing interpolated views of good quality.

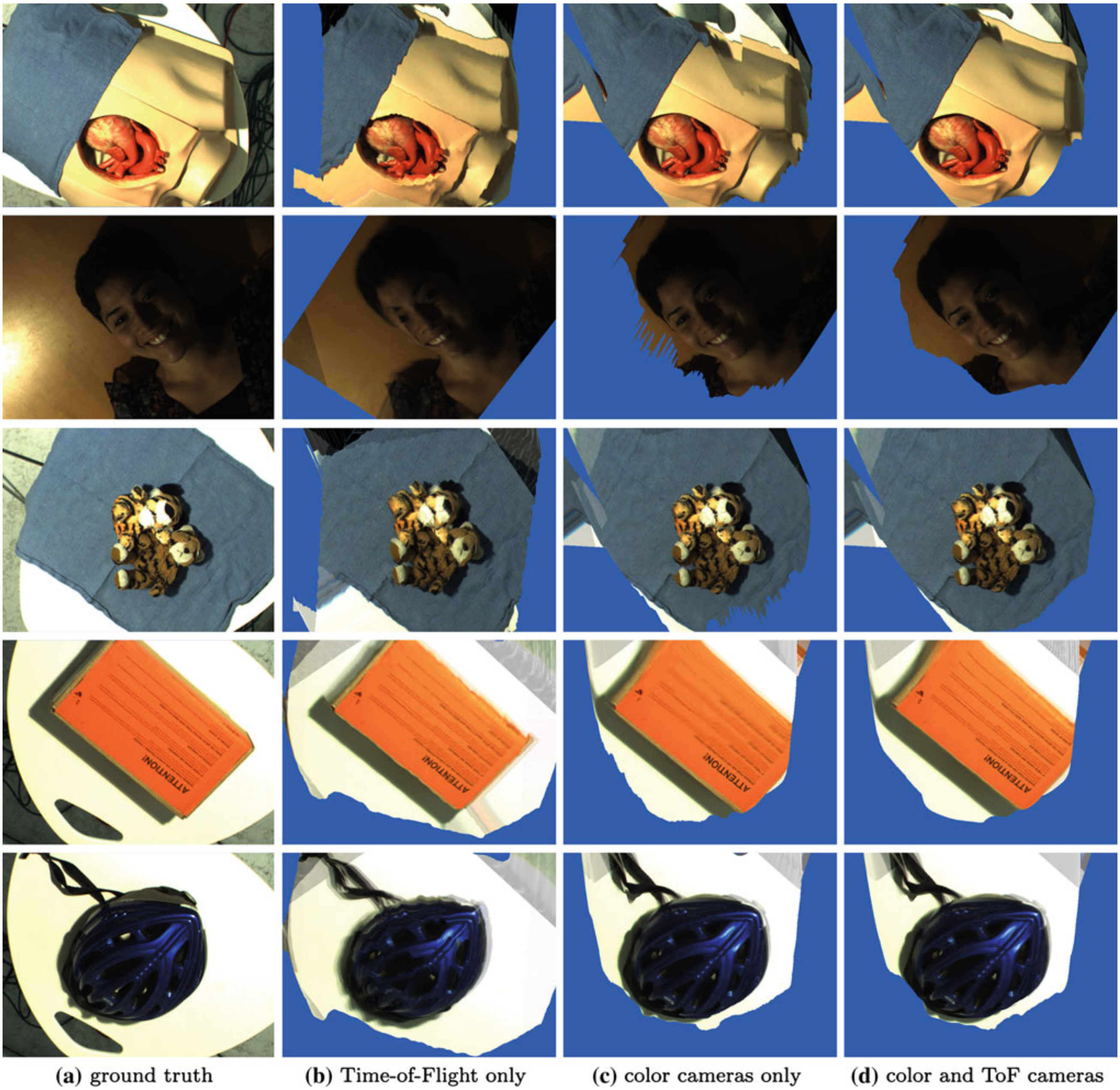

Fig. 5 Ground truth and view interpolation results using depth maps generated by different methods, based on the type of camera data used

In all experiments, camera 0 was used as the reference camera. The synthesized views were generated at a position on the outer ring of the camera array, shown in Fig. 4, and at least 20 cm away from any camera whose input was provided to the algorithm. A physical camera was placed at the same position to provide a ground-truth comparison, shown in the first column of Fig. 5. Obviously, this camera was not used to provide input to the reconstruction algorithm. Images captured by cameras 0 to 7, which are used by the proposed algorithm, are similar to these images and are not included here for reasons of space.

The second column shows results obtained by projecting color camera images onto the depth map obtained by regularizing data captured with the ToF camera. This was done by employing only the second and third terms of Eq. (16), without the confidence map factor. The third column presents view interpolation results obtained by exploiting only the information from the color cameras. This was achieved by using only the first and third terms of Eq. (16) in the construction of the reference camera depth map. Finally, views synthesized by exploiting both the ToF and color camera data, as described by the full Eq. (16), are shown in the last column. Note that our implementation of the proposed algorithm only reconstructs scene regions that are visible by all input cameras, i.e., cameras 0–7. Furthermore, we adjusted regularization parameters λ and α manually in each scenario in order to maximize the visual quality of the reconstructed images.

Comparing the images in Fig. 5, we observe that using only the ToF depth map yields interpolated views of very poor quality, which is essentially due to the low resolution and noisy nature of the ToF data.

The reconstructions obtained from depth maps computed exclusively from the photo-consistency measure of Eq. (1) yield reasonable quality interpolated views. In particular, it is interesting to observe the quality of the flat area under the two tiger teddy bears (third row), despite the repetitive pattern on this surface. The top of the helmet (bottom row) is also well reconstructed, despite the strong specular reflections. However, we note that the text on the orange box (fourth row) is blurred, which indicates that there are errors in the depth map in this region.

The resulting quality can sometimes be improved by combining the information from the color cameras with that of the ToF depth map. In particular, the printed text on the box is more legible in Fig. 5d than in Fig. 5c. However, in most cases presented here, incorporating the ToF data does not yield a significant improvement in visual quality over the pure photo-consistency-based method. One reason is that projecting input images onto an imperfect depth map does not always yield visible artefacts, as discussed in Sect. 3.4. As long as the viewpoint is not too far from the color cameras, only large errors in the depth map will generate visual defects in textureless regions. In our medical application, the interpolation viewpoint is constrained to be located in the neighborhood of the color cameras. Consequently, the quality of images generated without exploiting ToF data is generally acceptable, and often indistinguishable from a result that does include the range sensor data.

Nevertheless, the ToF camera can be highly useful to accelerate the computation of the depth map in textureless regions. This can be observed in Fig. 7, which illustrates depth map convergence as a function of frame number for two of the scenes of Fig. 5. Comparing this figure with Fig. 6, one can see that regions of low confidence take more time to converge when the photo-consistency measure alone is employed, due to the lower weight of the penalty term $\Phi_1(X, \mathbf{q})$ in such regions. The only way to reconstruct the

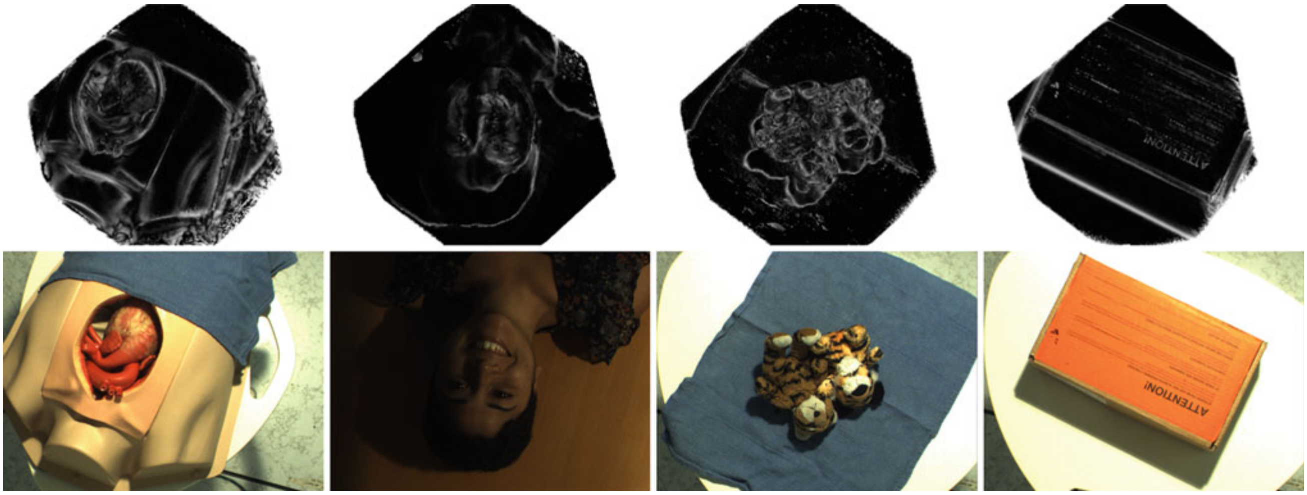

Fig. 6 Confidence maps computed for the scenes shown in rows 1 to 4 in Fig. 5. The second row shows the corresponding viewpoints of camera 0, which is used as the reference camera

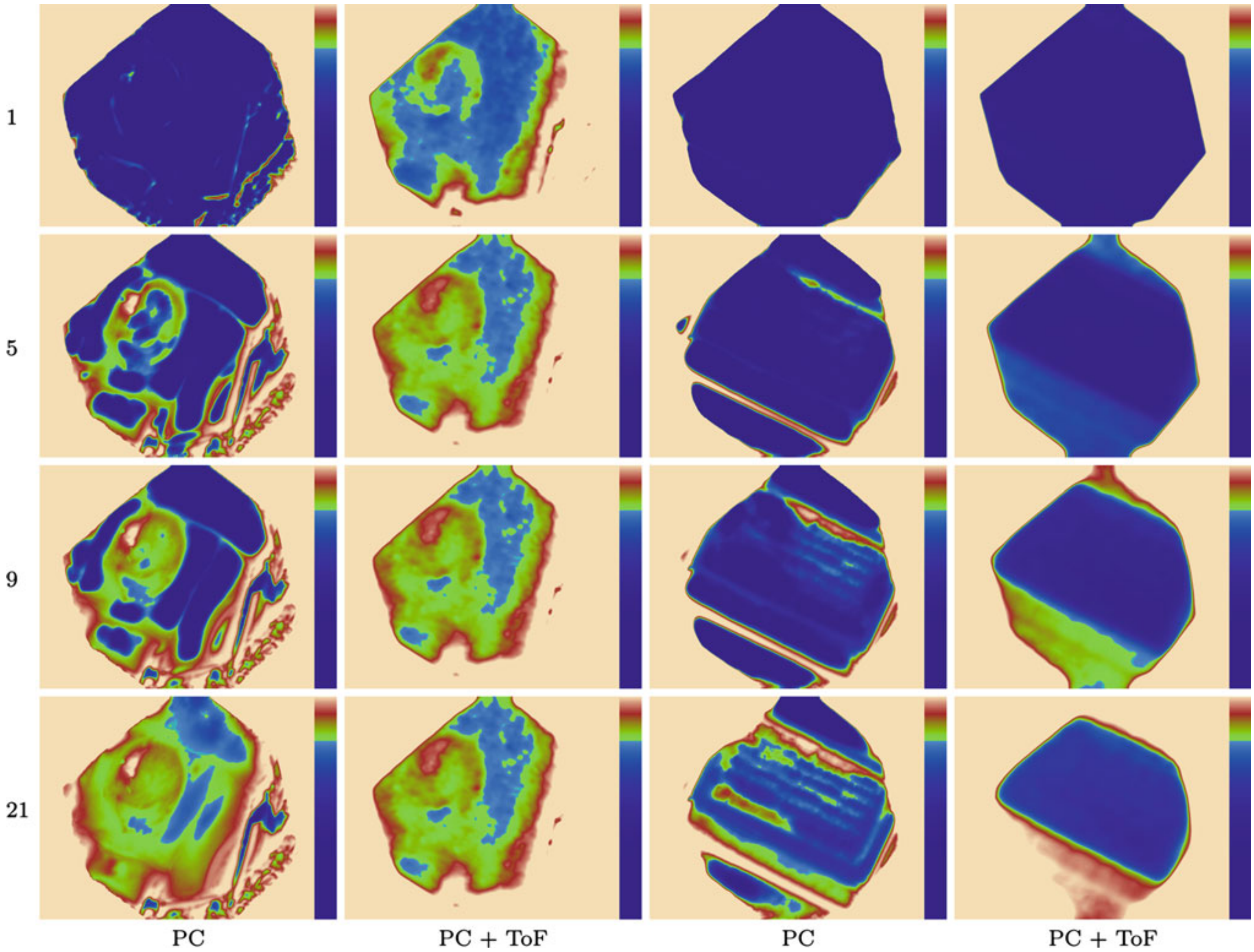

Fig. 7 Depth map obtained using photo-consistency (PC) only and PC with ToF data for two scenes of Fig. 5, as a function of frame number (indicated on the *left*)

depth map in these regions is to propagate the depth values from high-confidence regions to low-confidence ones in a manner similar to inpainting algorithms, which is accomplished by the regularization term $\Phi_3(X, \mathbf{q})$ at every CG iteration. Note that, in these experiments, the original depth values were set to zero at frame 0. In practice, since we use the previous depth map as the initial solution when computing the current one, actual performance is significantly improved in static regions, as explained in Sect. 3.3.

On the other hand, the use of ToF data accelerates convergence in regions of low confidence. Since such data provide a good approximation of the solution in these regions, only a few iterations are necessary to regularize the solution. Figure 8 illustrates the impact of the above on the interpolated views, comparing the views corresponding to the depth maps obtained after 9 frames from the two methods. The impact of large errors in the depth maps is readily visible in the views synthesized with the photo-consistency measurement alone.

In Fig. 9, we analyze the impact of the number M of interval maps and of the confidence interval size s_m on the visual quality of the views reconstructed by the proposed method for the same view, varying both values. The same regularization parameters were employed in all experiments, namely $\lambda = 2.7$ and $\alpha = 0.002$. Although visual artifacts are visible in Fig. 9a, b, both reconstructed using a single confidence interval, they are more prominent in the former, which uses the larger confidence interval s_1 of 0.30. As explained in Sect. 3.2, peaks of different widths in the photoconsistency curve $\Theta(\mathbf{p})$ should yield different confidence values for the associated depth estimates. However, peaks that are narrower than the confidence interval employed are all considered equally reliable by our method, and as such, use of a larger confidence interval makes it harder to select the "best" peak. For example, in Fig. 9a, the proposed algorithm cannot distinguish between two hypothetical peak widths of 0.05 and 0.15, whereas it can in Fig. 9b, which likely explains the better visual quality obtained in the latter case. Employing

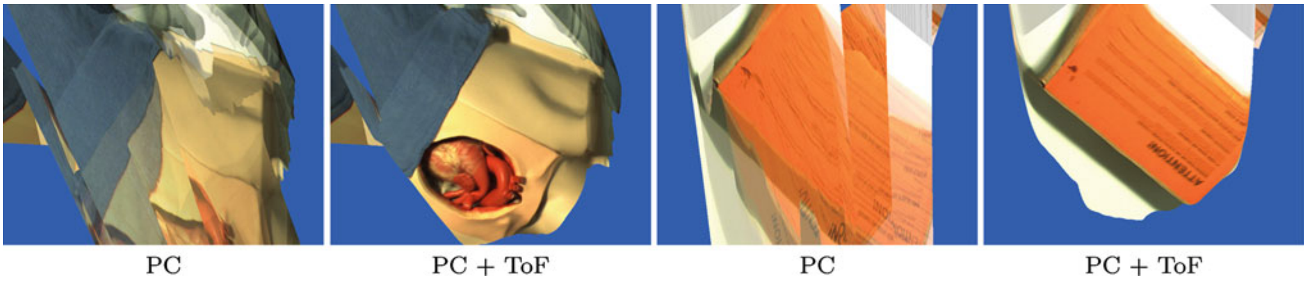
PC PC + ToF PC PC + ToF

Fig. 8 Interpolated images obtained after 9 frames using photo-consistency (PC) only and PC with ToF data. These images were synthesized from the depth maps shown in the third row of Fig. 7 using the technique described in Sect. 3.4

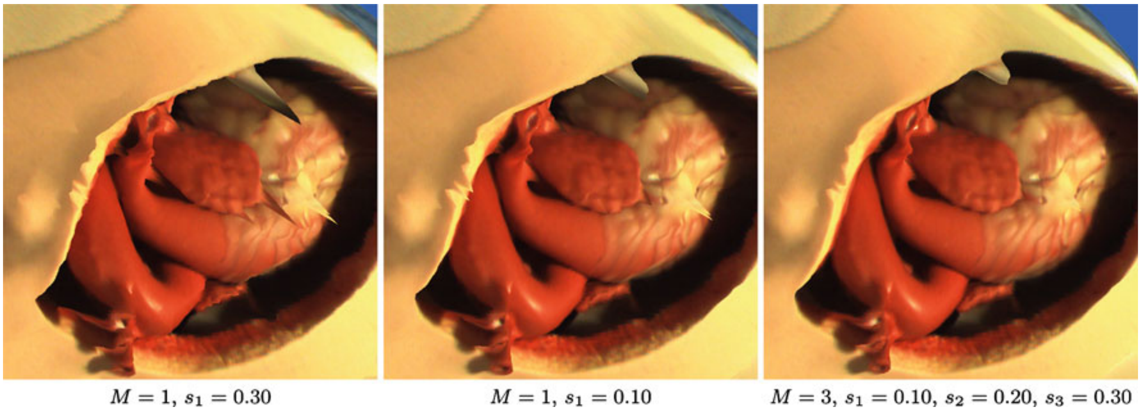
$M = 1,\ s_1 = 0.30$ $M = 1,\ s_1 = 0.10$ $M = 3,\ s_1 = 0.10,\ s_2 = 0.20,\ s_3 = 0.30$

Fig. 9 Effect of the number of confidence planes used on the interpolated images using only photo-consistency (PC). A ground truth view is provided in Fig. 1a

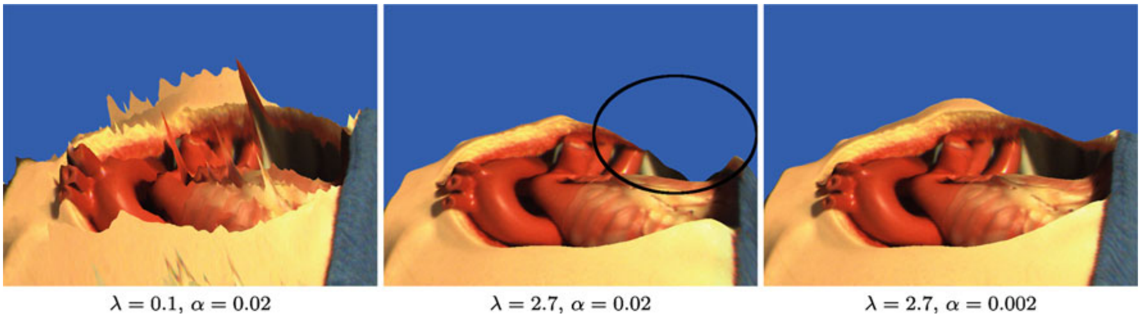
$\lambda = 0.1,\ \alpha = 0.02$ $\lambda = 2.7,\ \alpha = 0.02$ $\lambda = 2.7,\ \alpha = 0.002$

Fig. 10 Effect of the regularization parameters λ and α on a reconstructed view using only photo-consistency (PC)

three confidence intervals, as shown in Fig. 9c, increases the ability of our algorithm to discriminate between peak widths at different scales, which can further improve the visual quality (cf. Fig. 1a). These results demonstrate the improved visual quality of the synthesized views that can be achieved by our proposed method for computing the confidence map.

In Fig. 10, we illustrate the impact of the regularization parameters on the reconstructed view, deliberately choosing an off-axis view with respect to the input cameras in order to better show the impact of these parameters on the result. For this purpose, we reconstructed the same view using three different sets of values for λ and α. All other parameters of the algorithm were held constant across experiments. The effect of λ is evident upon comparing Fig. 10a and Fig. 10b, with an increased value yielding a smoother reconstructed surface. On the other hand, this results in the loss of some details in the structure of the scene, such as the edge around the heart cavity (see marked region). By reducing the value of the edge threshold parameter α, one can sometimes recover some of those details, as shown in Fig. 10c.

As is the case in the image restoration field, the optimal choice for these regularization parameters is a complex topic in itself, beyond the scope of this paper. However, we found that once these parameters are manually adjusted for a given camera setup, they can provide good reconstruction results for a wide variety of objects, provided the amount of light in the scene remains constant.

5 Conclusions

A method for performing view interpolation at interactive rates from a set of calibrated video cameras and a ToF camera was presented. Provided that the desired viewpoint is located in the vicinity of the physical cameras, we showed that a ToF camera is often unnecessary to obtain better visual quality. Nonetheless, we observed that a ToF camera is useful to accelerate the convergence speed of the algorithm.

Since IBR methods obtain improved depth resolution by increasing the number of planes, the cost of these algorithms is highly dependent on the efficiency of operations performed on each plane. In this regard, the denoising operation, typically applied during photo-consistency maximization, incurs costly switches of the rendering target on the GPU. However, we demonstrated that employing a confidence map to regularize a noisy depth map, obtained solely by maximizing photo-consistency scores, can yield views of impressive visual quality. This avoids the costly denoising step, and in turn, offers significant performance advantages.

Acknowledgments The authors would like to thank Canada's Advanced Research and Innovation Network (CANARIE) for funding of this work, conducted under a Network Enabled Platforms (NEP) program research contract, as part of the Health Services Virtual Organization project. Members of the Shared Reality Environment (SRE) lab at McGill University provided valuable assistance. Special thanks are due to Cristina Darolti, who conducted much of the early prototype development and testing, and Haijian Sun, who developed the software necessary to communicate with the cameras.

References

1. Audet, S., Okutomi, M.: A user-friendly method to geometrically calibrate projector-camera systems. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)—Workshops (Procams), pp. 47–54 (2009)
2. Bartczak, B., Koch, R.: Dense depth maps from low resolution time-of-flight depth and high resolution color views. In: Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II, ISVC '09, pp. 228–239. Springer-Verlag, Berlin, Heidelberg (2009)
3. Campbell, N.D.F., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: European Conference on Computer Vision, pp. 766–779 (2008)
4. Chai, J., Chan, S., Shum, H., Tong, X.: Plenoptic sampling. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 307–318 (2000)
5. Collins, R.: A space-sweep approach to true multi-image matching. In: 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, pp. 358–363 (1996)
6. Debevec, P., Yu, Y., Borshukov, G.: Efficient view-dependent image-based rendering with projective texture-mapping. In: Proceedings of Eurographics Rendering Workshop (1998)
7. Diebel, J., Thrun, S.: An application of markov random fields to range sensing. In: Proceedings of Conference on Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2005)
8. Fernando, R., Kilgard, M.J.: The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics. Addison-Wesley, Boston (2003)
9. Fiala, M., Shu, C.: Self-identifying patterns for plane-based camera calibration. Mach. Vis. Appl. **19**, 209–216 (2008)
10. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 367–383 (1992)
11. Geman, D., Yang, C.: Nonlinear image recovery with half-quadratic regularization. IEEE Trans. Image Proces. **4**(7), 932–946 (1995)
12. Gudmundsson, S.A., Aanaes, H., Larsen, R.: Fusion of stereo vision and Time-of-Flight imaging for improved 3D estimation. Int. J. Intell. Syst. Technol. Appl. **5**, 425–433 (2008)
13. Hartley, R.I, Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN: 0521540518
14. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
15. Jianfeng, Y., Cooperstock, J.R.: A new photo consistency test for voxel coloring. In: Proceedings of the Second Canadian Conference on Computer and Robot Vision (CRV). Victoria (2005)
16. Kraft, H., Frey, J., Moeller, T., Albrecht, M., Grothof, M., Schink, B., Hess, H., Buxbaum, B.: 3D-camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD (photonic mixer device)-technologies. In: OPTO (2004)
17. Kuhnert, K.D., Stommel, M.: Fusion of stereo-camera and PMD-camera data for real-time suited precise 3D environment reconstruction. In: In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4780–4785. Edinburgh, UK (2006)
18. Mühlmann, K., Maier, D., Hesser, J., Männer, R.: Dense disparity maps from color stereo images, an efficient implementation. Int. J. Comput. Vis. **47**, 79–88 (2002)
19. Reddy, M.: Perceptually optimized 3D graphics. IEEE Comput. Graphics Appl. **21**, 68–75 (2001)
20. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**(1), 7–42 (2002)
21. Schultz, R.R., Stevenson, R.L.: A bayesian approach to image expansion for improved definition. IEEE Trans. Image Process. **3**(3), 233–242 (1994)
22. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: International Conference on Computer Vision and Pattern Recognition, pp. 519–528 (2006)
23. Shewchuk, J.R.: An introduction to the conjugate gradient method without the agonizing pain. Tech. rep. Carnegie Mellon University, Pittsburgh (1994)
24. Taguchi, Y., Takahashi, K., Naemura, T.: Real-time all-in-focus video-based rendering using a network camera array. In: Proc. 3DTV-Conference, pp. 241–244 (2008)
25. Taguchi, Y., Wilburn, B., Zitnick, C.: Stereo reconstruction with mixed pixels using adaptive over-segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1–8 (2008)

26. Takahashi, K., Naemura, T.: Layered light-field rendering with focus measurement. Signal Process. Image Commun. **21**(6), 519–530 (2006)
27. Yang Q, Tan KH, Culbertson B, Apostolopoulos J (2010) Fusion of active and passive sensors for fast 3D capture. In: MMSP
28. Yang Q, Yang R, Davis J, Nistér D (2007) Spatial-depth super resolution for range images. In: CVPR
29. Zhu, J., Wang, L., Gao, J., Yang, R.: fusion for high accuracy depth maps using dynamic MRFs. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 899–909 (2010)
30. Zhu, J., Wang, L., Yang, R., Davis, J.: Fusion of Time-of-Flight depth and stereo for high accuracy depth maps. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
31. Zitnick, C., Kang, S.: Stereo for image-based rendering using image over-segmentation. Int. J. Comput. Vis. **75**(1), 49–65 (2007)

Author Biographies

Stéphane Pelletier received a Ph.D. from the Department of Electrical and Computer Engineering at McGill University in 2010. Since then, he has been involved in the development of a real-time view interpolation software for training students at the McGill Medical Simulation Centre. He also collaborated with a company in the development of an autostereoscopic display. His work interests are in the area of image processing and photorealistic 3D model reconstruction from images.

Jeremy R. Cooperstock directs McGill University's Shared Reality Lab, which focuses on computer mediation to facilitate high-fidelity human communication and the synthesis of perceptually engaging, multimodal, immersive environments. His accomplishments include the world's first Internet streaming demonstrations of Dolby Digital 5.1, uncompressed 12-channel 96kHz/24bit, multiple simultaneous streams of uncompressed high-definition video, and a simulation environment that renders graphic, audio, and vibrotactile effects in response to footsteps. His work has been recognized by an award for Most Innovative Use of New Technology from ACM/ IEEE Supercomputing and a Distinction Award from the Audio Engineering Society.