HAL
open science

Relative Pose Based Redundancy Removal: Collaborative RGB-D Data Transmission in Mobile Visual Sensor Networks

Xiaoqin Wang, Ahmet Sekercioglu, Tom Drummond, Vincent Frémont, Enrico Natalizio, Isabelle Fantoni

To cite this version:

Xiaoqin Wang, Ahmet Sekercioglu, Tom Drummond, Vincent Frémont, Enrico Natalizio, et al.. Relative Pose Based Redundancy Removal: Collaborative RGB-D Data Transmission in Mobile Visual Sensor Networks. Sensors, 2018, 18 (8), pp.2430. 10.3390/s18082430 . hal-01867323

HAL Id: hal-01867323

https://hal.science/hal-01867323

Submitted on 4 Sep 2018

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

sensors

MDPI

Article

Relative Pose Based Redundancy Removal: Collaborative RGB-D Data Transmission in Mobile Visual Sensor Networks

Xiaoqin Wang [1], Y. Ahmet Şekercioğlu [2,*], Tom Drummond [1], Vincent Frémont [2], Enrico Natalizio [2] and Isabelle Fantoni [3]

1 ARC Centre of Excellence for Robotic Vision, Monash University, Victoria 3800, Australia; xiaoqin.wang@monash.edu (X.W.); tom.drummond@monash.edu (T.D.)
2 Université de Technologie de Compiègne, Sorbonne Universités, CNRS, UMR 7253 Heudiasyc-CS 60 319, 60203 Compiègne, France; vincent.fremont@utc.fr (V.F.); enrico.natalizio@utc.fr (E.N.)
3 Ecole Centrale de Nantes, CNRS, UMR 6004 LS2N, 44300 Nantes, France; isabelle.fantoni@ls2n.fr
* Correspondence: asekerci@ieee.org

Received: 13 June 2018; Accepted: 20 July 2018; Published: 26 July 2018

Abstract: In this paper, the *Relative Pose based Redundancy Removal* (RPRR) scheme is presented, which has been designed for mobile RGB-D sensor networks operating under bandwidth-constrained operational scenarios. The scheme considers a multiview scenario in which pairs of sensors observe the same scene from different viewpoints, and detect the redundant visual and depth information to prevent their transmission leading to a significant improvement in wireless channel usage efficiency and power savings. We envisage applications in which the environment is static, and rapid 3D mapping of an enclosed area of interest is required, such as disaster recovery and support operations after earthquakes or industrial accidents. Experimental results show that wireless channel utilization is improved by 250% and battery consumption is halved when the RPRR scheme is used instead of sending the sensor images independently.

Keywords: RGB-D sensors; 3D mapping; visual sensors; robotic vision; collaborative coding; relative pose estimation

1. Introduction

Visual sensor networks (VSNs) allow the capture, processing, and transmission of per-pixel color information from a variety of viewpoints. The inclusion of low-cost compact RGB-D sensors, such as Microsoft Kinect [1], Asus Xtion [2] and Intel RealSense ZR300 [3], makes VSNs able to collect depth data as well.

RGB-D sensor-equipped VSNs can significantly enhance the performance of conventional applications such as immersive telepresence or mapping [4–7], environment surveillance [8,9], or object recognition and tracking [10–12] as well as opening the possibilities for new and innovative applications like hand gesture recognition [13], indoor positioning systems [14] and indoor relocalization [15]. The value of VSN applications becomes even more important, especially in places inaccessible to humans, such as supporting search and rescue operations after earthquakes, industrial or nuclear accidents. Indeed, examples of mapping (especially indoors) with networked mobile RGB-D sensors have started to appear in the research literature (Figure 1).

RGB-D sensors generate visual and depth data inevitably in huge quantities. The data volume will be even larger when multiple camera sensors observe the same scene from different viewpoints and exchange/gather their measurements to better understand the environment. As the sensors will most likely be communicating in ad hoc networking configurations, communication bandwidth

will be at a premium, and will be error-prone and not suitable for continuous data delivery in large quantities. Moreover, wireless transceivers consume a significant portion of the available battery power [16], and capacity limitation of on-board power sources should also be considered. Consequently, transmission of visual and depth information in resource-constrained VSN nodes must be carefully controlled and minimized as much as possible.

As the same scenery may be observed by multiple sensors (like the example shown in Figure 1), collected images will inevitably contain a significant amount of correlated information, and transmission load will be unnecessarily high if all the captured data are sent. In this paper, we focus on this issue and present a novel approach to the development of a comprehensive solution for minimizing the transmission of redundant RGB-D data in VSNs. Our framework, called *Relative Pose based Redundancy Removal* (RPRR), efficiently removes the redundant information captured by each sensor before transmission. We designed the RPRR framework particularly for RGB-D sensor-equipped VSNs, which eventually will need to work in situations with severely limited communication bandwidth. The scheme operates fully on board.

Figure 1. An example of 3D indoor mapping with two simultaneously operating mobile RGB-D sensor platforms [4].

In the RPRR framework, the characteristics of depth images, captured simultaneously with color data, are used to achieve the desired efficiency. Instead of using a centralized image registration technique [17,18], which requires one node to have full knowledge of the images captured by the others to determine the correlations, we propose a new approach based on relative pose estimation between pairs of RGB-D sensors and the 3D image warping technique [19]. The method we propose locally determines the color and depth information, which can only be seen by one sensor but not the others. Consequently, each sensor is required to transmit only the uncorrelated information to the remote station. In order to further reduce the amount of information before transmission, we apply a conventional coding scheme based on the discrete wavelet transform [20] with progressive coding features for color images, and a novel lossless differential entropy coding scheme for depth images (this algorithm was published in an earlier paper [21]). In addition, at the remote monitoring station, to deal with the artifacts that could occur in the reconstructed images due to the undersampling problem [22], we use our post-processing algorithms.

Early results of this work were presented in [23], and in this paper we

1. Add detailed theoretical refinements, practical implementation and experimental performance evaluation of the cooperative relative pose estimation algorithm [24] (Section 3.2),
2. Extend the theoretical development and practical implementation of the RPRR scheme for minimizing the transmission of redundant RGB-D data collected over multiple sensors with large pose differences (Section 3.3),
3. Describe the lightweight crack and ghost artifacts removal algorithms as a solution to the undersampling problem (Section 3.5), and
4. Include detailed experimental evaluation of wireless channel capacity utilization and energy consumption (Section 4.2).

In the following sections of the paper, after a discussion of the related work, we present the details of the RPRR framework in Section 3, and experimental results and their analysis can be found in Section 4, followed by our concluding remarks.

2. Related Work

A number of solutions exist in the research literature that intend to remove or minimize the correlated data for transmission in VSNs. They can be broadly classified into three groups:

- Optimal camera selection,
- Collaborative compression and transmission, and
- Distributed source coding.

The optimal camera selection algorithms [25–29] attempt to group the camera sensors with overlapping fields-of-view (FoVs) into clusters and only activate the sensor that can capture the image with the highest number of feature points. The pioneering work presented in [29] demonstrated that a correlation-based algorithm can be designed for selecting a suitable group of cameras communicating toward a sink so that the amount of information from the selected cameras can be maximized. Based on this work, in [28], the concept of "common sensed area" was proposed between two views to measure the efficiency of multiview video coding techniques and reduce the amount of information transmitted in VSNs. These algorithms operate under the assumption that the images captured by a small number of camera sensors in one cluster are good enough to represent the information of the scene/object. In these approaches, the location and orientation of the camera sensors are used to establish clusters, and a variety of existing feature detection algorithms [30,31] or place recognition approaches [15,32] are used to determine the similarity between captured images in each cluster. However, the occlusions in FoVs may cause significant differences between the images captured by cameras with very similar sensing directions. Therefore, the assumption is not realistic and this kind of approach is not applicable in many situations.

The collaborative compression and transmission methods [33–37] jointly encode the captured multi-view images. The spatial correlation is explored and removed at encoders by image registration algorithms. Only the uncorrelated visual content is delivered in the network after being jointly encoded by some recent coding techniques (e.g., Multiview Video Coding (MVC) [38,39]) and compressive sensing approaches [40,41]. However, at least one node in the network is required to have the full set of images captured by the other sensors in order to perform image registration. This means that the redundant information cannot be removed completely and still needs to be transmitted at least once. Moreover, as color images do not contain a full 3D representation of a scene, these methods introduce distortions and errors when the relative poses (location and orientation) between sensors are not pure rotation or translation, or the scenes have complex geometrical structures and occlusions.

The distributed source coding (DSC) algorithms [42–46] are other promising approaches that can be used to reduce the redundant data in multiview VSN scenarios. Each DSC encoder operates independently, but, at the same time, relies on joint decoding operations at the sink (remote monitoring station). The advantage of these approaches is that the camera sensors do not need to directly communicate the captured visual information with others in the network. Furthermore, these algorithms shift the computational complexity from the sensor nodes to the remote monitoring station, which fits the needs of VSNs well. However, the side information must be predicted as accurately as possible and the correlation structure should be able to be identified at the decoder side (remote monitoring station), without an accurate knowledge of the network topology and the poses of the sensors. These are the main disadvantages that prevent DSC algorithms from being widely implemented. A detailed discussion on multi-view image compression and transmission schemes in VSNs is presented in [47].

The algorithms mentioned above focus only on color (RGB) data. Just a few studies have been reported [4,48,49] that use RGB-D sensors in VSNs, as their use in networked robotics scenarios has

not yet become widespread. Consequently, our extensive review of the research literature has not identified any earlier studies that attempt to develop an efficient coding system that aims to maximize the bandwidth usage and minimize the energy consumption for RGB-D equipped VSNs.

3. Relative Pose Based Redundancy Removal (RPRR) Framework

3.1. Overview

In a mobile VSN tasked with mapping a region using RGB-D sensors, it is highly possible that multiple sensors will observe the same scene from different viewpoints. Because of this, scenery captured by the sensors with overlapping FoVs will have a significant level of correlated and redundant information. Here, our goal is to efficiently extract and encode the uncorrelated RGB-D information, and avoid transmitting the same surface geometry and color information repeatedly.

Consider the two sensors, a and b, of this VSN with overlapping FoVs. Let $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images returned by these sensors, and $\mathbf{C}_a$ and $\mathbf{C}_b$ are the corresponding color images. In the encoding procedure, we first estimate the location and orientation of one sensor relative to the other. Then, correlated and redundant information in color and depth images are identified to minimize unnecessary data transmissions to the central monitoring station. To achieve this, by using the relative pose information, sensor a computes a prediction of $\mathbf{Z}_b$ to determine the depth and color information that exists only in $\mathbf{Z}_b$ but not in $\mathbf{Z}_a$. Then, it informs sensor b to send only the uncorrelated depth and corresponding color information in $\mathbf{Z}_b$ and $\mathbf{C}_b$. To further improve the wireless channel capacity usage, depth image data is compressed with our own *Differential Huffman Coding with Multiple Lookup Tables (DHC-M)* method [21], and color images are compressed with Progressive Graphics File (PGF) scheme [50] prior to their transmission.

At the remote monitoring station, to improve the image quality, we apply algorithms for removal of the visual artifacts that may be introduced during the image reconstruction process.

A high-level view of the operation of the system is shown in Figure 2. A detailed explanation of each step is provided in the following sections.

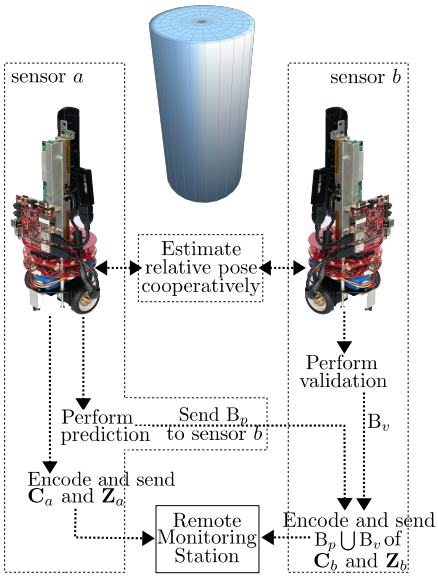


Figure 2. Operational overview of the RPRR framework. The sensors first cooperatively estimate their relative poses by using the algorithm shown in Figure 4, then, after identifying the non-overlapping image blocks, send only the non-redundant visual information to the remote monitoring station. Here, $\mathbf{C}_a$, $\mathbf{C}_b$ and $\mathbf{Z}_a$, $\mathbf{Z}_b$ are color and depth images obtained by sensors a and b, B_p is the set of image block coordinates that can only be observed by sensor b, and B_v is the set of image block coordinates covering the regions that sensor a incorrectly estimates as visible by sensor b.

3.2. Relative Pose Estimation

As an RGB-D sensor can provide a continuous measurement of the 3D structure of the environment, the relative pose between two RGB-D sensors can be estimated through explicit matching of surface geometries in the overlapping regions within their FoVs. A variety of algorithms have been proposed to determine whether multiple cameras are looking at the same scene, such as vision-based [27,51] or geometry-based [29,52] methods. Here, we assume that the sensors use one of these approaches to detect whether they are observing the same scene. Afterwards, as explained below, with our relative pose estimation algorithm, the sensors accurately estimate their relative position and orientation (relative pose).

The relative pose between the RGB-D sensors a and b can be represented by a transformation matrix:

$$\mathbf{M}_{ab} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

in SE(3) [53], where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. The transformation matrix $\mathbf{M_{ab}}$ represents the six degrees of freedom (6DoF) motion model, which not only describes the relative pose between two sensors and also the transformation of the structure between the depth images captured by both sensors.

The transformation matrix $\mathbf{M}_{ab}$ can be estimated by matching the surface geometries captured by two sensors. Taking advantage of the depth image characteristics, the depth pixels in a frame captured by sensor b can be mapped to a frame captured by sensor a. Consider the vector $\mathbf{p}_e = [x\ y\ z\ 1]^T$ which represents a real world point in Euclidean space by using homogeneous coordinates. Given the following intrinsic parameters of an RGB-D sensor:

- principal point coordinates (i_c, j_c) and
- focal length of the camera (f_x, f_y),

$\mathbf{p}_e$ can be estimated from the corresponding pixel in a depth image by using the pinhole camera model as

$$\frac{1}{z}\mathbf{p}_e = \frac{1}{z}\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T = \begin{bmatrix} \frac{i-i_c}{f_x} & \frac{j-j_c}{f_y} & 1 & \frac{1}{z} \end{bmatrix}^T,$$

where (i, j) denotes the pixel coordinates of the projection of this real world point in the depth image, and z is the corresponding depth value reported by the camera.

In the discussion that follows, we assume that $\mathbf{p_e}$ can be observed by both mobile RGB-D sensors a and b, and the projections of $\mathbf{p}_e$ are located at pixel coordinates (i_a, j_a) and (i_b, j_b) on the depth images $\mathbf{Z}_a$ and $\mathbf{Z}_b$, respectively. Under the assumption that the world coordinate system is equal to the mobile sensor coordinate system, and the intrinsic parameters of both sensors are identical, the depth pixel (projection) at (i_a, j_a) in $\mathbf{Z}_a$ can establish a relationship between the depth pixel at (i_b, j_b) in $\mathbf{Z}_b$ as

$$\begin{bmatrix} \frac{i_b-i_c}{f_x} & \frac{j_b-j_c}{f_y} & 1 & \frac{1}{z_b} \end{bmatrix}^T = \mathbf{M}_{ab}\begin{bmatrix} \frac{i_a-i_c}{f_x} & \frac{j_a-j_c}{f_y} & 1 & \frac{1}{z_a} \end{bmatrix}^T \tag{1}$$

and, to simplify the equation, by doing some rudimentary algebraic substitutions, we obtain

$$\begin{bmatrix} u_b & v_b & 1 & q_b \end{bmatrix}^T = \mathbf{M}_{ab}\begin{bmatrix} u_a & v_a & 1 & q_a \end{bmatrix}^T$$

in inverse depth coordinates.

We now need to estimate $\mathbf{M}_{ab}$. To accomplish an accurate estimate of $\mathbf{M}_{ab}$, we have developed an Iterative Closest Point (ICP) algorithm which operates in a distributed fashion by using the explicit registration of surface geometries extracted from the depth frames captured by two sensors [24]. It delivers robust results especially in circumstances with heavy occlusion. In our distributed algorithm, the registration problem is approached by iteratively minimizing a cost function whose error metric

is defined based on the bidirectional point-to-plane geometrical relationship as explained in the following paragraphs.

Let $\mathrm{P}_a = \{\mathbf{p}_{l,a}, l = 1, 2, \ldots, N_a\}$ and $\mathrm{P}_b = \{\mathbf{p}_{k,b}, k = 1, 2, \ldots, N_b\}$ denote two sets of measurements sampled from $\mathbf{Z}_a$ and $\mathbf{Z}_b$. Let us assume that the correspondences for $N = N_a + N_b$ pairs (A typical depth image may have hundreds of thousands of points, therefore running algorithms on the full point cloud is computationally expensive. In order to alleviate this problem, a commonly used method is to subsample the data for speeding up the operation with the cost of reduced accuracy. This is a fundamental trade-off of ICP performance: registration by using dense point clouds yields a more accurate alignment, however it needs longer processing time to complete. On the other hand, a subsampled point cloud results in lower accuracy, but requires a significantly shorter processing time. Thus, for the best ICP (and its variants) performance, striking a balance between accuracy and processing time is required by considering the timing requirements for obtaining results and the computational resources available. Considering these, after conducting a series of experiments on our sensor platforms, we have chosen $N_a = N_b = 250.$) of points $(\mathbf{p}_{l,a} \leftrightarrow \mathbf{p}^*_{l,b})$ and $(\mathbf{p}_{k,b} \leftrightarrow \mathbf{p}^*_{k,a})$ are established to form the sets P^*_a and P^*_b. Here, $\mathbf{p}^*_{l,b} \in \mathrm{P}^*_a$ (where $\mathrm{P}^*_a \subset \mathbf{Z}_b$) is the corresponding point of $\mathbf{p}_{l,a}$, and $\mathbf{p}^*_{k,a} \in \mathrm{P}^*_b$ (where $\mathrm{P}^*_b \subset \mathbf{Z}_a$) is the corresponding point of $\mathbf{p}_{k,b}$ (see Figure 3). Then, the transformation matrix $\mathbf{M}_{ab}$ can be estimated by minimizing the bidirectional point-to-plane error metric $\mathcal{C}$, expressed in normal least squares form as

$$\mathcal{C} = \sum_{l=1}^{N_a} \left[w_{l,a} (\mathbf{M}_{ab} \mathbf{p}_{l,a} - \mathbf{p}^*_{l,b})^T \vec{\mathbf{n}}^*_{l,b} \right]^2 + \sum_{k=1}^{N_b} \left[w_{k,b} (\mathbf{M}_{ab}^{-1} \mathbf{p}^*_{k,a} - \mathbf{p}_{k,b})^T \vec{\mathbf{n}}_{k,b} \right]^2, \tag{2}$$

where $w_{l,a}$ and $w_{k,b}$ are the weight parameters for the correspondences established in opposite directions between the pairs,

$$\vec{\mathbf{n}}^*_{l,b} = \begin{bmatrix} \beta^*_{l,b} & \gamma^*_{l,b} & \delta^*_{l,b} & 0 \end{bmatrix}^T \tag{3}$$

and

$$\vec{\mathbf{n}}_{k,b} = \begin{bmatrix} \beta_{k,b} & \gamma_{k,b} & \delta_{k,b} & 0 \end{bmatrix}^T, \tag{4}$$

are the surface normals at the points $\mathbf{p}^*_{l,b}$ and $\mathbf{p}_{k,b}$. The cost function presented in Equation (2) consists of two parts:

- the sum of squared distances from $\mathbf{Z}_a$ to $\mathbf{Z}_b$, and
- the sum of squared distances from $\mathbf{Z}_b$ to $\mathbf{Z}_a$.

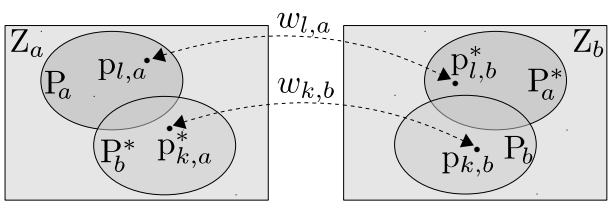


Figure 3. Two sets of points ($\mathrm{P}_a \subset \mathbf{Z}_a$ and $\mathrm{P}_b \subset \mathbf{Z}_b$) sampled from the depth images $\mathbf{Z}_a$ and $\mathbf{Z}_b$, and their corresponding point sets P^*_a and P^*_b. P_a and P_b have N_a and N_b number of elements, respectively. For finding the point sets, the project-and-walk method is used with a neighborhood size of 7×7 pixels based on the nearest neighbor criteria as proposed in [54].

The estimation of $\mathbf{M}_{ab}$ can be done by iteratively re-weighting the least squares operation in an ICP framework. Based on this principle, we have created the distributed algorithm which has two complementary components running concurrently on sensors a and b as shown in Figure 4.

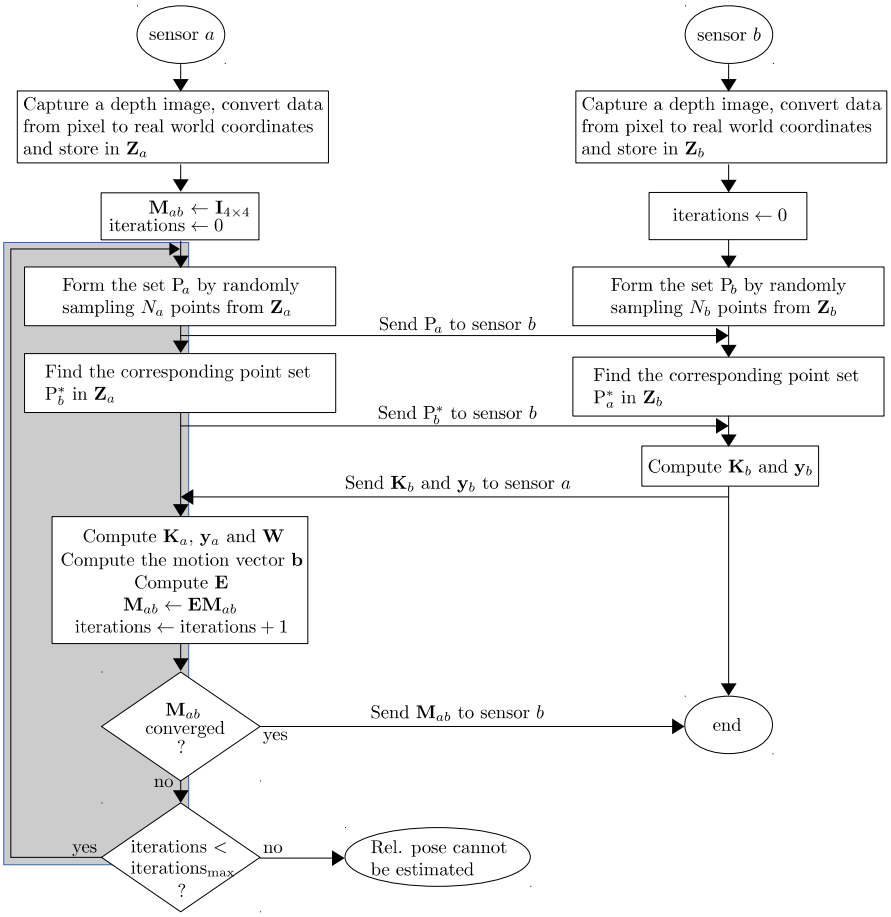


Figure 4. Operation of the cooperative relative pose estimation algorithm. The algorithm is distributed over two sensors, and operates iteratively (denoted in gray) until it converges or maximum number of iterations is reached. We have used the convergence criterion presented in [55], and iterations$_{\max}$ is set as 50 (see Section 4.1).

On sensor a, in the first iteration, $\mathbf{M}_{ab}$ is initialized as the identity matrix. Afterwards, in this coarse-to-fine algorithm, by using the information sent by sensor b, each iteration generates an update $\mathbf{E}$ to the sensor's pose, which modifies the transformation matrix $\mathbf{M}_{ab}$. $\mathbf{E}$ takes the same form as $\mathbf{M}_{ab}$ and can be parameterized by a six-dimensional motion vector having the elements $\alpha_1, \alpha_2, \ldots, \alpha_6$ via the exponential map and their corresponding group generator matrices $\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_6$ as

$$\mathbf{E} = \exp\left(\sum_{j=1}^{6} \alpha_j \mathbf{G}_j\right), \tag{5}$$

where

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here, $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{G}_3$ are the generators of translations in x, y and z directions, while $\mathbf{G}_4$, $\mathbf{G}_5$ and $\mathbf{G}_6$ are rotations about x, y and z axes, respectively. For details, please refer to [56,57]. The task then becomes finding the elements of the six-dimensional motion vector

$$\mathbf{b} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \end{bmatrix}^T$$

that describe the relative pose. By determining the partial derivatives of u_b, v_b and q_b with respect to the unknown elements of $\mathbf{b}$, the Jacobian matrix for each established corresponding point pair can be obtained as

$$\mathbf{J} = \begin{bmatrix} q_a & 0 & -u_a q_a & -u_a v_a & 1+u_a^2 & -v_a \\ 0 & q_a & -v_a q_a & -1-v_a^2 & v_a u_a & u_a \\ 0 & 0 & -q_a^2 & -v_a q_a & u_a q_a & 0 \end{bmatrix}. \tag{6}$$

The six-dimensional motion vector $\mathbf{b}$, which minimizes Equation (2), is then determined iteratively by the least squares solution

$$\mathbf{b} = (\mathbf{K}^T \mathbf{W} \mathbf{K})^{-1} \mathbf{K}^T \mathbf{W} \mathbf{y}, \tag{7}$$

in which

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_b \\ \mathbf{K}_a \end{bmatrix}, \tag{8}$$

where

$$\mathbf{K}_b = \begin{bmatrix} (\vec{\mathbf{n}}_{1,b}^{*\prime})^T \mathbf{J}_{1,a} \\ \vdots \quad \vdots \\ (\vec{\mathbf{n}}_{l,b}^{*\prime})^T \mathbf{J}_{l,a} \\ \vdots \quad \vdots \\ (\vec{\mathbf{n}}_{N_a,b}^{*\prime})^T \mathbf{J}_{N_a,a} \end{bmatrix} \qquad \mathbf{K}_a = \begin{bmatrix} (\vec{\mathbf{n}}_{1,b}^{\prime})^T \mathbf{J}_{1,a}^{*} \\ \vdots \quad \vdots \\ (\vec{\mathbf{n}}_{k,b}^{\prime})^T \mathbf{J}_{k,a}^{*} \\ \vdots \quad \vdots \\ (\vec{\mathbf{n}}_{N_b,b}^{\prime})^T \mathbf{J}_{N_b,a}^{*} \end{bmatrix}, \tag{9}$$

$\vec{\mathbf{n}}_{l,b}^{*\prime} = \begin{bmatrix} \beta_{l,b} & \gamma_{l,b} & \delta_{l,b} \end{bmatrix}^T$, $\vec{\mathbf{n}}_{k,b}^{\prime} = \begin{bmatrix} \beta_{k,b} & \gamma_{k,b} & \delta_{k,b} \end{bmatrix}^T$ are the surface normals at the points $\mathbf{p}_{l,b}^{*} \in \mathrm{P}_a^{*}$ and $\mathbf{p}_{k,b} \in \mathrm{P}_b$ expressed in a slightly different form than in Equations (3) and (4), and

$$\mathbf{J}_{l,a} = \begin{bmatrix} q_{l,a} & 0 & -u_{l,a} q_{l,a} & -u_{l,a} v_{l,a} & 1+u_{l,a}^2 & -v_{l,a} \\ 0 & q_{l,a} & -v_{l,a} q_{l,a} & -1-v_{l,a}^2 & v_{l,a} u_{l,a} & u_{l,a} \\ 0 & 0 & -q_{l,a}^2 & -v_{l,a} q_{l,a} & u_{l,a} q_{l,a} & 0 \end{bmatrix} \tag{10}$$

is the associated Jacobian matrix calculated over the corresponding point $\mathbf{p}_{l,a} \in \mathrm{P}_a$ (see Figure 3). Similarly,

$$\mathbf{J}_{k,a}^{*} = \begin{bmatrix} q_{k,a}^{*} & 0 & -u_{k,a}^{*} q_{k,a}^{*} & -u_{k,a}^{*} v_{k,a}^{*} & 1+u_{k,a}^{*2} & -v_{k,a}^{*} \\ 0 & q_{k,a}^{*} & -v_{k,a}^{*} q_{k,a}^{*} & -1-v_{k,a}^{*2} & v_{k,a}^{*} u_{k,a}^{*} & u_{k,a}^{*} \\ 0 & 0 & -q_{k,a}^{*2} & -v_{k,a}^{*} q_{k,a}^{*} & u_{k,a}^{*} q_{k,a}^{*} & 0 \end{bmatrix} \tag{11}$$

is the associated Jacobian matrix calculated over the corresponding point $\mathbf{p}_{k,a}^{*} \in \mathrm{P}_b^{*}$, and

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_b \\ \mathbf{y}_a \end{bmatrix}, \tag{12}$$

where

$$\mathbf{y}_b = \begin{bmatrix} -(\mathbf{p}_{1,a} - \mathbf{p}^*_{1,b})^T \vec{\mathbf{n}}^*_{1,b} \\ \vdots \\ -(\mathbf{p}_{N_a,a} - \mathbf{p}^*_{N_a,b})^T \vec{\mathbf{n}}^*_{N_a,b} \end{bmatrix} \qquad \mathbf{y}_a = \begin{bmatrix} -(\mathbf{p}^*_{1,a} - \mathbf{p}_{1,b})^T \vec{\mathbf{n}}_{1,b} \\ \vdots \\ -(\mathbf{p}^*_{N_b,a} - \mathbf{p}_{N_b,b})^T \vec{\mathbf{n}}_{N_b,b} \end{bmatrix}. \tag{13}$$

In addition,

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{N,N} \end{bmatrix} \tag{14}$$

contains the weightings for the bidirectional point-to-plane correspondences. As reported in [58], different weighting functions lead to various probability distributions. Based on our experiments, we have found that the asymmetric weighting function

$$w_{a,b} = \begin{cases} c/[c + (z_a - z_b)], & \text{if } z_b \leq z_a \,, \\ c/[c + (z_a - z_b)^2], & \text{otherwise} \end{cases} \tag{15}$$

yields satisfactory results. Here, z_a and z_b are the depth values of corresponding points in two depth images, and c is the mean of differences between the depth values of all corresponding points. An extended discussion of the weighting function can be found in our paper [24].

To detect the convergence of our algorithm, we use the thresholds for the ICP framework presented in [55]. Once the algorithm converges, the registration is considered completed and $\mathbf{M}_{ab}$ is used for the elimination of the redundant data in transmissions as explained in Section 3.3.

The sensors exchange very small amounts of information by using this algorithm making the process very bandwidth-efficient fitting the requirements of VSNs. We present an in-depth analysis of message exchange complexity in Section 4.1.

3.3. Identification of Redundant Regions in Images

3.3.1. Prediction

Sensor a, by using the relative pose information $\mathbf{M}_{ab}$, can now apply Equation (1) on each pixel in $\mathbf{Z}_a$ to create a predicted a depth image $\mathbf{Z}^*_b$, which is virtually captured from sensor b's viewpoint. In this process, though, it could happen that two or more different depth pixels are warped into the same pixel coordinates in $\mathbf{Z}^*_b$. This over-sampling issue could occur because some 3D world points are occluded by the other ones at the new viewpoint. In order to solve this problem, we always compare the depth values of the pixels warped to the same coordinates, and the pixel with the closest range information to the camera always overwrites the other pixels. As the depth image is registered to the color image, the color pixels in $\mathbf{C}_a$ can also be mapped along with the depth pixels to generate a virtual color image $\mathbf{C}^*_b$ as well.

Then, the captured images $\mathbf{Z}_a$, $\mathbf{C}_a$, and virtual images $\mathbf{Z}^*_b$, $\mathbf{C}^*_b$ are decomposed into blocks of 8×8 pixels. In $\mathbf{Z}^*_b$, some blocks have no depth information due to the fact that none of the pixels in $\mathbf{Z}_a$ can be warped into these regions. This indicates that the blocks with the same coordinates in $\mathbf{Z}_b$ and $\mathbf{C}_b$ contain the information which can only be observed by sensor b. Sensor a collects these block coordinates in the set B_p and transmits them to sensor b.

An illustration of this process is shown in Figure 5. In this example, the regions in which the depth information can only be observed by $\mathbf{Z}_b$ are outlined in yellow.

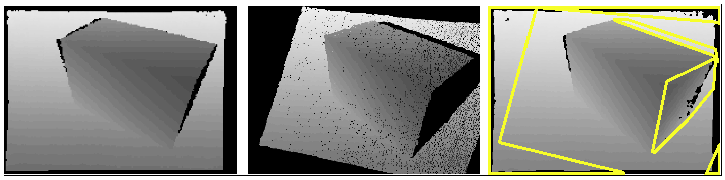


Figure 5. An intuitive example of the prediction process. The depth image $\mathbf{Z}_b^*$ is synthetically generated from $\mathbf{Z}_a$ as the image captured by sensor b virtually. The uncorrelated information in $\mathbf{Z}_b$ is outlined with yellow lines.

3.3.2. Validation

Although in most circumstances the prediction process can detect the uncorrelated information in the images captured by the other sensor, it may fail to operate correctly in situations when some points are occluded by the objects that can be seen by sensor b, but not by sensor a. A typical scenario is shown in Figure 6. In this example, the cylinder is outside the FoV of sensor a, and, because of this, it falsely treats some parts of the background (the dashed rectangular area) as the surface that is observable by sensor b. However, since the surface of the cylinder is included in $\mathbf{Z}_b$, it occludes the background from the viewpoint of sensor b. As a result, the prediction process cannot accurately determine the uncorrelated depth and color information in this case.

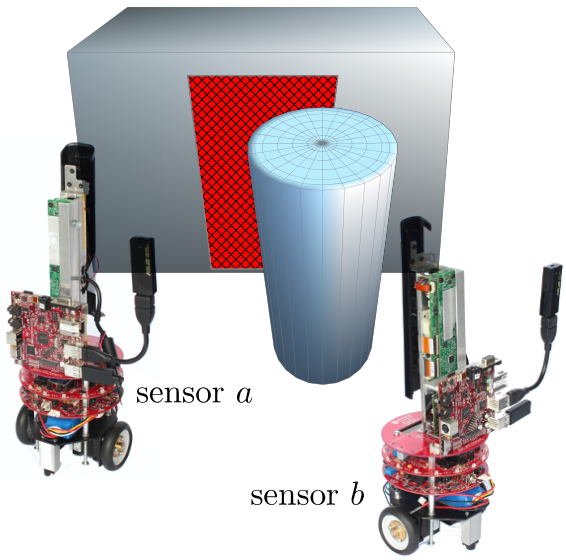


Figure 6. The rectangular surface area at the background is within the field of view of sensor b, but occluded by the cylinder at the foreground.

In order to solve this problem, we include a validation mechanism into the overall process. First, similar to the image warping process from sensor a to b, sensor b generates the synthetic image $\mathbf{Z}_a^*$ as virtually captured from sensor a's viewpoint by mapping the pixels (i_b, j_b) in $\mathbf{Z}_b$ to (i_a, j_a) in $\mathbf{Z}_a^*$ by applying

$$\begin{bmatrix} \frac{i_a - i_c}{f_x} & \frac{j_a - j_c}{f_y} & 1 & \frac{1}{z_a} \end{bmatrix}^T = \mathbf{M}_{ab}^{-1} \begin{bmatrix} \frac{i_b - i_c}{f_x} & \frac{j_b - j_c}{f_y} & 1 & \frac{1}{z_b} \end{bmatrix}^T.$$

In this process, the pixels representing the range information of the surface of the cylinder move out of the image coordinate range and are not shown in $\mathbf{Z}_a^*$. Sensor b identifies the image blocks containing these pixels, and records their coordinates in the set B_v. Then, sensor b transmits only the image blocks in $\mathbf{Z}_b$ and $\mathbf{C}_b$ that their coordinates are included in the union of the sets B_p and B_v.

3.4. Image Coding

After the elimination of the redundant image blocks, the remaining uncorrelated depth and color information is compressed to further improve the communication channel usage.

For depth images, we use our own design *Differential Huffman Coding with Multiple Lookup Tables (DHC-M)* lossless compression scheme [21]. It is very fast and capable of compressing the depth images without introducing any artificial refinements.

Among the many options for compressing color images, JPEG 2000 [59] and H.264 [60] intra mode can be mentioned as the leading schemes. As the wireless channels are impacted by noise and being error prone, coding schemes that provide progressive coding are considered to be more suitable for sensor networks. Moreover, since a sensor node of a VSN has limited computational capability, a lightweight image coding scheme is required in sensor network applications. Progressive Graphics File (PGF) scheme [50], which is based on discrete wavelet transform with progressive coding features, has high coding efficiency and low complexity. It has compression efficiency comparable to JPEG 2000, and is ten times faster. Moreover, PGF has a small, open source and easy to use C++ codec [61] without any dependencies. These properties make PGF suitable for onboard image compression.

3.5. Post-Processing on the Decoder Side

On the decoder side (remote monitoring station), first, the received bitstream is decompressed. Then, the color and depth images captured by sensor a are used to predict the color and depth images captured by sensor b.

The 3D image warping process represented by Equation (1) may introduce some visual artifacts in the synthesized view, such as disocclusions (Disocclusions are areas occluded in the reference viewpoint and which become visible in the virtual viewpoint, due to the parallax effect.), cracks (Cracks are small disocclusions, and mostly occur due to undersampling.), or ghosts (Ghosts are artifacts due to the projection of pixels that have background depth and mixed foreground/background color.). Various methods have been proposed in the literature for their prevention or removal [62,63].

In our framework, as the information that can only be observed by sensor b is transmitted, disocclusions can be eliminated by filling the areas affected by disocclusions in the synthesized image with the color and depth information transmitted by sensor b. Then, the main artifacts we need to deal with remain as cracks (Figure 7) and ghosts (Figure 8).

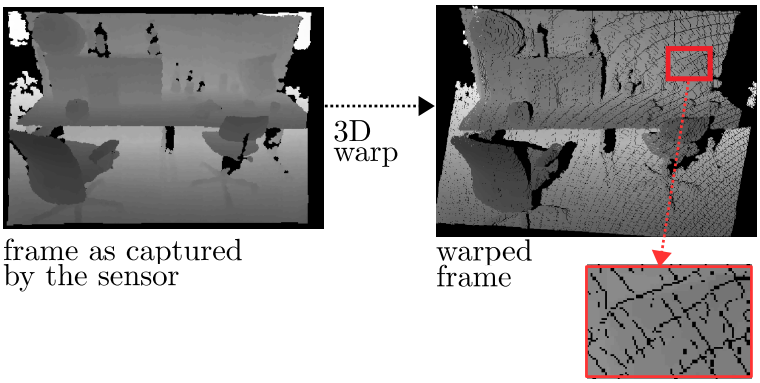


Figure 7. Crack artifacts: holes can be introduced during the image warping process due to the undersampling problem.

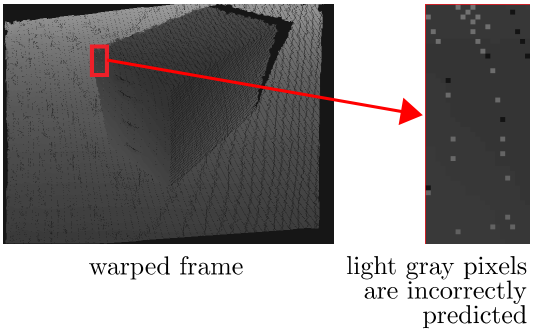


Figure 8. Ghost artifacts: the light gray pixels actually belong to the background surface and falsely warped onto the surface at the foreground.

3.5.1. Removal of Crack Artifacts

The missing color information in cracks is frequently avoided by operating a backward projection [64], which works in two steps:

- The cracks in the synthetic depth image are filled by a median filter, and then a bilateral filter is applied to smoothen the depth map while preserving the edges.
- The filtered depth image is warped back into the reference viewpoint to find the color of the synthetic view.

This approach exhibits good performance on filling the cracks, but at the same time it smoothens the complete image and introduces noise in regions with correct depth values, especially on the object boundaries. In order to avoid this adverse effect, we have modified it by using an adaptive median filter. The filter is applied only on the pixels with invalid depth values instead of the whole image. Instead of warping back the complete image to find the color information, we have adopted the work presented in [65], which warps back only the filled pixels in cracks, because the color information of the other pixels that are not in cracks can be directly estimated in the warping process.

3.5.2. Removal of Ghost Artifacts

As illustrated in Figure 8, some background surfaces are incorrectly shown on the foreground obstacle's surface. This is because the pixels representing the foreground surface become scattered after the warping process, and the background surface can be seen through the interspaces between these pixels. In order to remove this noise, we need to first identify the location of the incorrectly predicted pixels and then fill them with the correct values. As the value of the incorrectly predicted pixel is significantly different from its neighboring pixels, this kind of impulse noise can also be revised by using an adaptive median filter. We propose a windowing scheme with a 3×3 pixels size to determine whether or not a depth pixel contains incorrect values. If more than half of the neighboring pixels are out of a certain range, which is either much larger or much smaller than the center pixel in the window, the center pixel is estimated as an incorrectly predicted pixel. Then, it is replaced with the median value of its neighboring pixels, which are not out of the range. The corresponding color information can be found by backward warping, which is similar to the solution for crack artifacts presented in Section 3.5.1.

4. Experimental Results and Performance Evaluation

In this section, we first evaluate the performance of the relative pose estimation algorithm. Then, we analyze the overall performance of the RPRR framework through the experiments conducted on our mobile VSN platform.

4.1. Performance Evaluation of the Relative Pose Estimation

In order to quantitatively evaluate the performance of the relative pose estimation algorithm, we used two groups of datasets with varying degrees of occlusions. We first generated our own datasets by using a turntable setup to obtain the imagery viewed from accurately measured angular positions. A number of objects were placed on the center of the turntable, and the images were captured with a tripod mounted Kinect sensor [1]. In the experiments with the first dataset group, the ground truth is known exactly at every precisely controlled 5° interval. We used this setup to compare our algorithm (ICP-BD) [24] with the standard ICP [66] and ICP in inverse depth coordinates (ICP-IVD) [55]. The performance of the algorithms was evaluated based on the rotational and translational Root Mean Square (RMS) errors. The results show that

1. When the angular interval becomes greater than 15°, an increasing amount of occlusion occurs between two sensors' views. Under such circumstances, ICP-BD outperforms other variants as it reports much lower translational and rotational RMS error.
2. Standard ICP has the poorest performance across the experiments. ICP-IVD can provide similar accuracy in pose estimation before it diverges. However, as the scene becomes more occluded as the turntable is being rotated, ICP-IVD fails to converge sooner than ICP-BD.

In summary, ICP-BD estimation accuracy is much better than that of ICP and ICP-IVD. In addition, its estimation is very robust even under large pose differences. Details of the experiment methodology and results can be found in [24].

We also evaluated the number of iterations required for the ICP-BD algorithm's convergence. Our experiments show that, as one can expect, the number of iterations increases as the angular difference between two views increases. Two representative results are plotted in Figure 9.

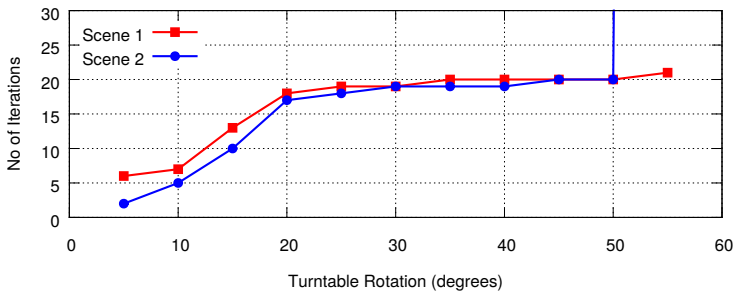


Figure 9. Number of iterations required by the ICP-BD algorithm in two scenes shown in Figure 5 of [24].

In order to gain further insight into the number of iterations required by our algorithm in densely cluttered scenes, we used a second group of datasets which were selected from the Technical University of Munich Computer Vision Group's RGB-D SLAM dataset and benchmark collection [67,68]. Each dataset is a sequence of Kinect video frames capturing one scene from different angles of view. For emulating the situations including varying amounts of occlusion between two sensor views, we created four new sequences from each dataset by extracting one frame out of every 5, 10, 20, and 30 frames. For each trial, we treated two consecutive frames in these 28 new sequences as the depth images captured by two separate sensors with varying relative poses. We recorded the number of iterations required by the ICP-BD algorithm that converged successfully. We normalized the results of 4005 trials to plot them as discrete probability distributions as shown in Figure 10. The results show that the average number of iterations is 5.1 and the maximum value is smaller than 20. Based on these numbers, we can say that the message exchange complexity of the relative pose estimation algorithm is near-constant. At each iteration, the depth information and image coordinates of 250 sampled points need to be transmitted, which lead to 1.09 kB of bandwidth consumption approximately (excluding the

protocol overheads). Therefore, on average, 5.6 kB of data are sent in each message when the relative localization algorithm (Figure 4) distributed over sensors *a* and *b* is in operation.

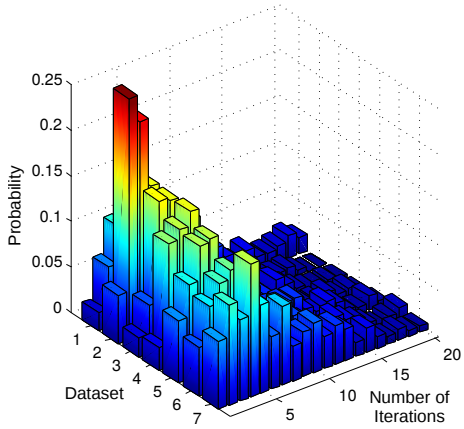


Figure 10. Distributions of the number of iterations required for the convergence of the algorithm. We experimentally obtained them over 4005 trials by using the 28 sequences we constructed by extracting image pairs that are 5, 10, 20 and 30 frames apart from the following seven RGB-D SLAM datasets [67,68]: 1. freiburg1_plant (1139 frames), 2. freiburg2_dishes (3005 frames), 3. freiburg3_cabinet (1121 frames), 4. freiburg3_large_cabinet (993 frames), 5. freiburg3_structure_texture_far (914 frames), 6. freiburg3_long_office_household (2509 frames), and 7. freiburg1_xyz_cabinet (800 frames). The plot shows that, for example, the algorithm converged after three iterations in 23% of the trials in the four sequences extracted from the dataset 2.

4.2. Performance Evaluation of the RPRR Framework

In this set of experiments, we evaluated the performance of the RPRR framework by using two mobile RGB-D sensors (Figure 11) of our VSN platform. The platform consists of multiple mobile RGB-D sensors named "eyeBug" (Figure 11). EyeBugs were designed for computer vision and mobile robotics experiments, such as multi-robot SLAM and scene reconstruction. We selected the Microsoft Kinect as the RGB-D sensor due to its low cost and wide availability. We mounted a Kinect vertically at the center of the top board of each eyeBug.

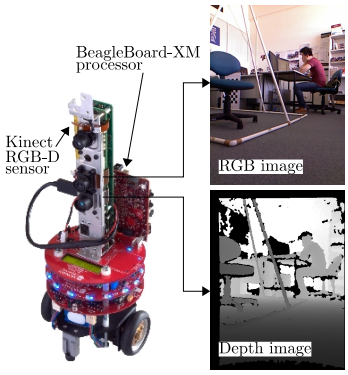


Figure 11. eyeBug [69,70], the mobile RGB-D sensor we used in our experiments. The color and depth data generated by the Kinect sensor is processed on a BeagleBoard-xM [71] computer running the GNU/Linux operating system.

A Kinect is capable of producing color and disparity-based depth images at a rate of 30 frames/second. A BeagleBoard-xM single-board computer [71] was used for image processing tasks. Each BeagleBoard-xM has a 1 GHz ARM Cortex-A8 processor, a USB hub, and an HDMI video output port. A USB WiFi adapter was connected to the BeagleBoard to provide communication between robots. We ran an

ARM-processor-optimized Linux kernel. OpenKinect [72], OpenCV [73] and libCVD [74] libraries were installed to capture and process image information. The default RGB video stream provided by the Kinect uses eight bits for each color at VGA resolution (640 × 480 pixels, 24 bits/pixel). The monochrome depth video stream is also in VGA resolution. The value of each depth pixel represents the distance information in millimeters. Invalid depth pixel values are recorded as zero, indicating that the RGB-D sensor is not able to estimate the depth information of that point in the 3D world.

Color and depth images were captured in six different scenes, as shown in Figure 12. In this set-up, sensor *a* transmits entire captured color and depth images to the central monitoring station. Then, sensor *b* is required to transmit only the uncorrelated color and depth information that cannot be observed by sensor *a*. At the central monitoring station, the color and depth images captured by sensor *b* are reconstructed by using the information transmitted by two sensors.

Figure 12. A demonstration of the RPRR framework over six scenes. (**a**) images captured and transmitted to the remote monitoring station by sensor *a*; (**b**) images captured by sensor *b*. Note that these images are not transmitted to the remote monitoring station; (**c**) image blocks transmitted by sensor *b* (black regions denote the parts of an image that are identified as redundant by our scheme and consequently not transmitted); (**d**) reconstructed images of sensor *b*'s point of view. They are produced at the remote monitoring station using the partial images transmitted by sensor *b* shown in row (**c**), and ideally should be identical to the corresponding images in row (**b**).

As the color and depth images captured by sensor *a* are compressed and transmitted to the receiver in their entirety, we only needed to evaluate the reconstruction quality of the images captured by sensor *b*. The depth images are usually complementary to the color images in many applications, and in our framework the color images are reconstructed according to depth image warping. Thus, if the color images can be accurately reconstructed, so the reconstructed depth images as well. Therefore, in this set of experiments, we focused on evaluating the quality of the reconstructed color images.

4.2.1. Subjective Evaluation

The image blocks transmitted by sensor *b* are shown in the third row of Figure 12. In the fourth row of the figure, reconstructed images can be seen. They were obtained by stitching the blocks extracted from the warped sensor *a* images into the black regions of the corresponding sensor *b* images. In the reconstructed images of scenes 2 and 4, we observe significant color changes on the stitching boundary. This is caused by the illumination variations within the scene, and auto-iris response of the sensors to different levels of scene brightness.

Generally, it is clear that the reconstructed images preserve the structural information of the original images accurately.

4.2.2. Objective Evaluation

Even though many approaches have been proposed to compress multi-view images [17,34,36,42,75–77], they cannot be applied in our system. These approaches either require the transmitter to have the knowledge of the full set of images or only work on cameras with very small motion differences. In contrast, in our case, each sensor only has its own captured image, and the motion difference between two visual sensors is very large. To the best of our knowledge, our proposal is the first distributed framework that efficiently codes and transmits images captured by multiple RGB-D sensors with large pose differences, and so we do not have any work to compare ours against. For this reason, we can only compare the performance of our framework with the approaches that compress and transmit images independently.

As the color information is coded using the PGF [50] lossy mode, we can vary the compression ratio, and, consequently, coding performance. The performance was evaluated according to two aspects: reconstruction quality and bits per pixel (bpp). We measured the Peak-Signal-to-Noise-Ratio (PSNR) between the reconstructed and original images captured by sensor *b* with different bpp. The results are shown in Figure 13.

Figure 13 shows that the RPRR framework can achieve much lower bpp than the independent transmission scheme. However, the PSNR upper bounds achieved by the RPRR framework are limited. It is because the reconstruction quality depends on the depth image accuracy and correlations between the color images. Since the depth images generated by a Kinect sensor are not accurate enough, the displacement distortion of depth images, especially the misalignment around the object edges, introduces noise into the reconstruction process. Another reason is the inconsistent illumination between the color images captured by two sensors. Even if the prediction and validation processes establish the correct correspondences between two color pixels according to the transformation between depth images, the values of these two color pixels can be very different due to the various brightness levels in two images. These characteristics lead to low PSNR upper bounds of the reconstructed color images. Several methods [78,79] have been proposed to overcome this drawback; however, the time-complexity of these methods prevents them from being implemented on sensor systems with constrained computational resources. We can see that the reconstructed color image in Scene 6 has the highest PSNR. This is because the relative pose between two sensors is small, which leads to small differences in the structure of the captured scenes and the brightness of their captured images. Therefore, more information captured by sensor *b* can be reconstructed by information observed by sensor *a*. For that reason, according to Figure 12(a-vi), only a small number of blocks in images captured by sensor *b* need to be transmitted. We also observe that Scenes 2 and 4 have

the lowest reconstruction qualities. This is because the brightness level is quite different in the color images captured by two sensors (see image pairs shown in Figure 12(a-ii,b-ii) and Figure 12(a-iv,b-iv)). Although the structures of the scenes are preserved nicely in the reconstructed color images, distinct color changes over the stitching boundaries are shown in Figure 12(d-ii,d-iv). Consequently, we can say that the RPRR framework is suitable for implementation of the VSN applications with very limited bandwidth requiring very high compression ratios. This is because when the bpp or the compression ratio increases, the quality of the color image reconstructed by RPRR decreases more gradually than the quality of the image compressed by the independent transmission scheme.

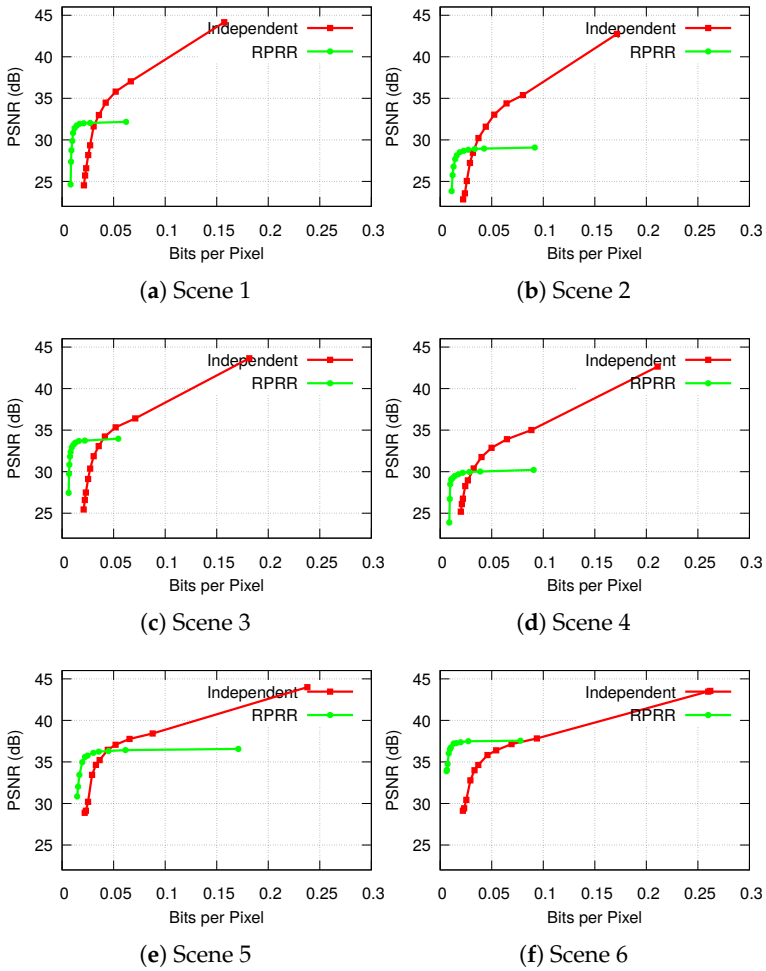


Figure 13. Comparisons of PSNR (dB) achieved by compressing the images at various levels by using the RPRR framework against transmitting them independently.

4.2.3. Energy Consumption

The limited battery capacity of mobile sensors places limits on their performance. Therefore, a data transmission scheme, while attempting to reduce the transmission load, must not have a significant negative impact on the overall energy consumption. In this section, we present our experimental measurements and evaluation regarding the overall energy consumption and amount of transmitted data of the RPRR framework collected on our eyeBug mobile visual sensors to demonstrate this aspect.

The overall energy consumption of the RPRR framework can be measured by

$$\begin{aligned} E^{R}_{\text{overall}} &= E_{\text{processing}} + E_{\text{encoding}} + E_{\text{sending}} \\ &= V_o I_p t_p + V_o I_e t_e + V_o I_s t_s \end{aligned} \tag{16}$$

in which V_o denotes the sensor's operating voltage, and I_p, I_e, and I_s represent the current drawn from the battery during processing, encoding, and sending operations. t_p, t_e, and t_s are the corresponding operation times required for these procedures.

The overall energy consumption when images are transmitted independently can be measured as

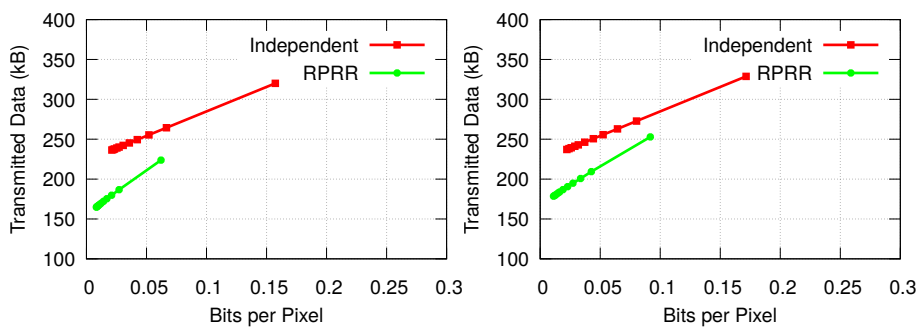
$$\begin{aligned} E^{I}_{\text{overall}} &= E_{\text{encoding}} + E_{\text{sending}} \\ &= V_o I_e t_e + V_o I_s t_s. \end{aligned} \tag{17}$$

Note that the operation times t_e and t_s are different in the two transmission schemes as the image sizes change after removing the redundant information.

Our sensor operates at 15 V, and the current levels remain fairly constant during each operation. We measured them as follows: $I_p = 0.06$ A, $I_e = 0.06$ A, and $I_s = 0.12$ A. Our experiments show that, in the RPRR framework, due to different compression ratios, the transmission time varies between 32 and 42 ms, and the operational time for processing and encoding remains between 509 and 553 ms. The overall energy consumption of the RPRR scheme changes between 480 and 520 mJ, depending on the compression ratio. The corresponding values for the independent scheme are between 918 and 920 mJ. The data clearly show that the RPRR framework leads to the consumption of much lower battery capacity than the independent transmission scheme. It cuts the overall energy consumption of the sensor nearly by half. In the RPRR framework, the energy consumption for two sensors are asymmetric, and if sensor a always transmits complete images, its energy will be quickly drained. A simple method to prolong the network lifetime is for the two sensors to transmit complete images alternately. The current consumed by an eyeBug in idle state is 650 mA. According to the experimental results above, the theoretical operational time of RPRR on a pair of eyeBugs with 2500 mA h 3-cell (11.1 V) LiPo batteries is around 5.2 h. In this period, around 3.24×10^4 color and depth image pairs can be transmitted to the remote monitoring station.

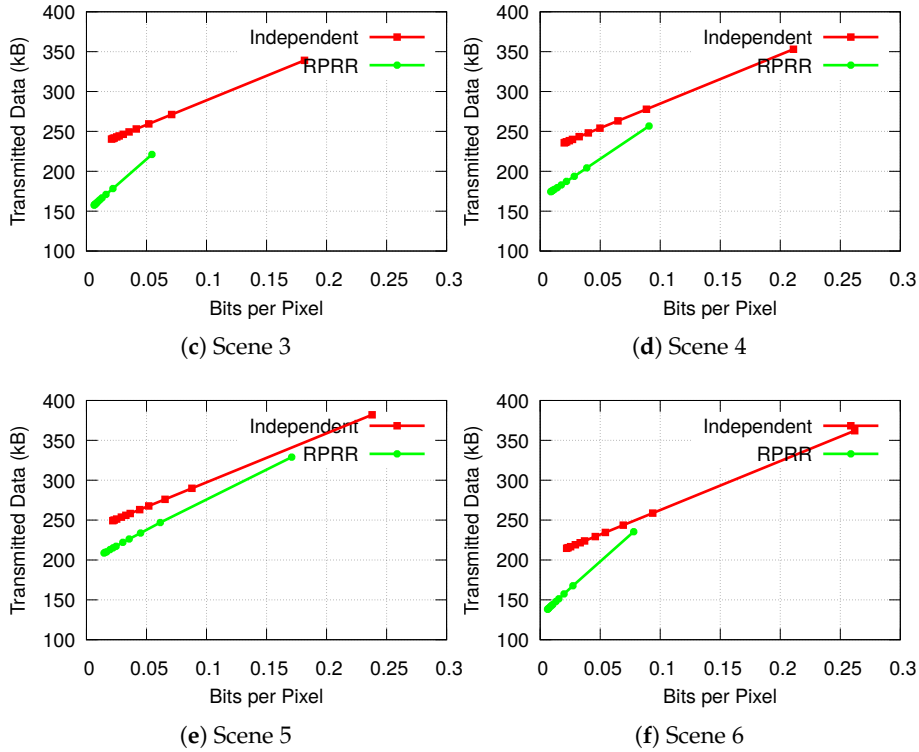
4.2.4. Transmitted Data Volume

Finally, we compare the amount of transmitted data for two pairs of color and depth images required by the RPRR and the independent transmission schemes. The results are shown in Figure 14. We can see that bits per pixel achieved by the independent transmission approach is much higher than bit per pixel achieved by the RPRR framework. It is also noticeable that even if the bits per pixel required by a color image is the same in both approaches, the RPRR framework transmits fewer number of bytes. This is because only parts of the color and depth images need to be transmitted in RPRR. In contrast, a complete depth image has to be sent in the independent transmission scheme. The data clearly show that the RPRR framework leads to more efficient use of the wireless channel capacity than the independent transmission scheme.



(**a**) Scene 1 (**b**) Scene 2

Figure 14. *Cont.*



(**c**) Scene 3 (**d**) Scene 4

(**e**) Scene 5 (**f**) Scene 6

Figure 14. Comparisons of the transmitted data for color images at various compression levels by using the RPRR framework against transmitting them independently.

5. Conclusions

We presented a novel collaborative transmission framework for mobile VSNs that efficiently removes the redundant visual information captured by RGB-D sensors. The scheme, called *Relative Posed based Redundancy Removal* (RPRR), considers a multiview scenario in which pairs of sensors observe the same scene from different viewpoints. Taking advantage of the unique characteristics of depth images, our framework explores the correlation between the images captured by these sensors using solely the relative pose information. Then, only the uncorrelated information is transmitted. This significantly reduces the amount of information transmitted compared with sending two individual images independently. The scheme's computational resource requirements are quite modest, and it can run on battery-operated sensor nodes. Experimental results show that the compression ratio achieved by the RPRR framework is 2.5 times better than the independent transmission scheme, and it yields this result while nearly halving the energy consumption of the independent transmission scheme on average.

The RPRR framework is the first attempt to remove the redundancy in the color and depth information observed by VSNs equipped with RGB-D sensors, and so there is room for further improvements. For example, our scheme only operates on pairs of mobile sensors at this stage. A simple extension of the RPRR framework for networks with a large number of RGB-D sensors is to choose one sensor as the reference which transmits complete images (like sensor a in Figure 2) while the other sensors transmit only the uncorrelated information (like sensor b in Figure 2). However, a certain amount of redundancy still exists in this approach and further refinements are possible.

Our future research efforts will concentrate on developing a more sophisticated extension which uses feature matching algorithms to assign sensors with overlapping FoVs to the same subgroups and applies RPRR on sensors in the same subgroup to remove redundancies in networks with a large number of RGB-D sensors.

Author Contributions: Conceptualization, X.W., Y.A.Ş. and T.D.; Formal analysis, T.D., V.F., E.N. and I.F.; Funding acquisition, Y.A.Ş., T.D. and V. F.; Investigation, X.W.; Methodology, X.W., Y.A. Ş., T.D. and V.F.; Writing—original draft, X.W. and Y.A.Ş.; Writing—review & editing, T.D., V.F., E.N. and I.F.

Funding: This work was supported by the Australian Research Council Centre of Excellence for Robotic Vision (Project Number CE140100016). This work was carried out in the framework of the Labex MS2T and DIVINA challenge team, which were funded by the French Government, through the program *Investments for the Future* managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kinect RGB-D Sensor. Available online: https://developer.microsoft.com/en-us/windows/kinect (accessed on 24 July 2018).
2. Xtion RGB-D Sensor. Available online: https://www.asus.com/fr/3D-Sensor/Xtion_PRO/ (accessed on 24 July 2018).
3. RealSense RGB-D Sensor. Available online: https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html (accessed on 24 July 2018).
4. Mohanarajah, G.; Usenko, V.; Singh, M.; D'Andrea, R.; Waibel, M. Cloud-Based Collaborative 3D Mapping in Real-Time With Low-Cost Robots. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 423–431. [CrossRef]
5. Beck, S.; Kunert, A.; Kulik, A.; Froehlich, B. Immersive Group-to-Group Telepresence. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 616–625. [CrossRef] [PubMed]
6. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663. [CrossRef]
7. Lemkens, W.; Kaur, P.; Buys, K.; Slaets, P.; Tuytelaars, T.; Schutter, J.D. Multi RGB-D Camera Setup for Generating Large 3D Point Clouds. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1092–1099.
8. Choi, W.; Pantofaru, C.; Savarese, S. Detecting and Tracking People Using an RGB-D Camera via Multiple Detector Fusion. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops, Barcelona, Spain, 6–13 November 2011; pp. 1076–1083.
9. Liu, W.; Xia, T.; Wan, J.; Zhang, Y.; Li, J. RGB-D Based Multi-attribute People Search in Intelligent Visual Surveillance. In *Advances in Multimedia Modeling*; Springer: Berlin, Germany, 2012; Volume 7131, pp. 750–760.
10. Almazan, E.; Jones, G. A Depth-Based Polar Coordinate System for People Segmentation and Tracking with Multiple RGB-D Sensors. In Proceedings of the IEEE ISMAR 2014 Workshop on Tracking Methods and Applications, Munich, Germany, 10–12 September 2014.
11. Alexiadis, D.; Zarpalas, D.; Daras, P. Real-Time, Full 3D Reconstruction of Moving Foreground Objects from Multiple Consumer Depth Cameras. *IEEE Trans. Multimed.* **2013**, *15*, 339–358. [CrossRef]
12. Tong, J.; Zhou, J.; Liu, L.; Pan, Z.; Yan, H. Scanning 3D Full Human Bodies Using Kinects. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 643–650. [CrossRef] [PubMed]
13. Wang, C.; Liu, Z.; Chan, S.C. Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera. *IEEE Trans. Multimed.* **2015**, *17*, 29–39. [CrossRef]
14. Duque Domingo, J.; Cerrada, C.; Valero, E.; Cerrada, J.A. An Improved Indoor Positioning System Using RGB-D Cameras and Wireless Networks for Use in Complex Environments. *Sensors* **2017**, *17*, 2391. [CrossRef] [PubMed]
15. Li, R.; Liu, Q.; Gui, J.; Gu, D.; Hu, H. Indoor Relocalization in Challenging Environments With Dual-Stream Convolutional Neural Networks. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 651–662. [CrossRef]
16. Aziz, A.A.; Şekercioğlu, Y.A.; Fitzpatrick, P.; Ivanovich, M. A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 121–144. [CrossRef]
17. Lu, J.; Cai, H.; Lou, J.G.; Li, J. An Epipolar Geometry-Based Fast Disparity Estimation Algorithm for Multiview Image and Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 737–750. [CrossRef]
18. Merkle, P.; Smolic, A.; Muller, K.; Wiegand, T. Efficient Prediction Structures for Multiview Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1461–1473. [CrossRef]

19. Fehn, C. Depth-Image-Based Rendering (DIBR), Compression, and Transmission for a New Approach on 3D-TV. In *Stereoscopic Displays and Virtual Reality Systems XI*; International Society for Optics and Photonics: Bellingham, WA, USA, 2004; Volume 5291, pp. 93–104.
20. Akansu, A.N.; Haddad, R.A. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*; Academic Press, Inc.: Orlando, FL, USA, 1992.
21. Wang, X.; Şekercioğlu, Y.A.; Drummond, T.; Natalizio, E.; Fantoni, I.; Fremont, V. Fast Depth Video Compression for Mobile RGB-D Sensors. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 673–686. [CrossRef]
22. Mark, W.R. Post-Rendering 3D Image Warping: Visibility, Reconstruction and Performance for Depth-Image Warping. Ph.D. Thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1999.
23. Wang, X.; Şekercioğlu, Y.A.; Drummond, T.; Natalizio, E.; Fantoni, I.; Frémont, V. Collaborative Multi-Sensor Image Transmission and Data Fusion in Mobile Visual Sensor Networks Equipped with RGB-D Cameras. In Proceedings of the 2016 IEEE International Conference on Multisen Fusion and Integration for Intelligent Systems (MFI 2016), Baden-Baden, Germany, 19–21 September 2016.
24. Wang, X.; Şekercioğlu, Y.A.; Drummond, T. A Real-Time Distributed Relative Pose Estimation Algorithm for RGB-D Camera Equipped Visual Sensor Networks. In Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013), Palm Springs, CA, USA, 29 October–1 November 2013.
25. Chow, K.Y.; Lui, K.S.; Lam, E. Efficient Selective Image Transmission in Visual Sensor Networks. In Proceedings of the IEEE 65th Vehicular Technology Conference (VTC2007-Spring), Dublin, Ireland, 22–25 April 2007; pp. 1–5.
26. Bai, Y.; Qi, H. Redundancy Removal Through Semantic Neighbor Selection in Visual Sensor Networks. In Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), Como, Italy, 30 August 30–2 September 2009; pp. 1–8.
27. Bai, Y.; Qi, H. Feature-Based Image Comparison for Semantic Neighbor Selection in Resource-Constrained Visual Sensor Networks. *J. Image Video Process.* **2010**, *2010*, 469563. [CrossRef]
28. Colonnese, S.; Cuomo, F.; Melodia, T. An Empirical Model of Multiview Video Coding Efficiency for Wireless Multimedia Sensor Networks. *IEEE Trans. Multimed.* **2013**, *15*, 1800–1814. [CrossRef]
29. Dai, R.; Akyildiz, I. A Spatial Correlation Model for Visual Information in Wireless Multimedia Sensor Networks. *IEEE Trans. Multimed.* **2009**, *11*, 1148–1159. [CrossRef]
30. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
31. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In *European Conference on Computer Vision*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 430–443.
32. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J. Visual Place Recognition: A Survey. *IEEE Trans. Robot.* **2016**, *32*, 1–19. [CrossRef]
33. Chia, W.C.; Ang, L.M.; Seng, K.P. Multiview Image Compression for Wireless Multimedia Sensor Network Using Image Stitching and SPIHT Coding with EZW Tree Structure. In Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2009), Hangzhou, China, 26–27 August 2009; Volume 2, pp. 298–301.
34. Chia, W.C.; Chew, L.W.; Ang, L.M.; Seng, K.P. Low Memory Image Stitching and Compression for WMSN Using Strip-based Processing. *Int. J. Sens. Netw.* **2012**, *11*, 22–32. [CrossRef]
35. Cen, N.; Guan, Z.; Melodia, T. Interview Motion Compensated Joint Decoding for Compressively Sampled Multiview Video Streams. *IEEE Trans. Multimed.* **2017**, *19*, 1117–1126. [CrossRef]
36. Wu, M.; Chen, C.W. Collaborative Image Coding and Transmission over Wireless Sensor Networks. *J. Adv. Signal Process.* **2007**, *2007*, 223. [CrossRef]
37. Shaheen, S.; Javed, M.Y.; Mufti, M.; Khalid, S.; Khanum, A.; Khan, S.A.; Akram, M.U. A Novel Compression Technique for Multi-Camera Nodes through Directional Correlation. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 539838. [CrossRef]
38. Vetro, A.; Wiegand, T.; Sullivan, G. Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard. *Proc. IEEE* **2011**, *99*, 626–642. [CrossRef]

39. Redondi, A.E.; Baroffio, L.; Cesana, M.; Tagliasacchi, M. Multi-view Coding and Routing of Local Features in Visual Sensor Networks. In Proceedings of the the 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016), San Francisco, CA, USA, 10–15 April 2016.
40. Ebrahim, M.; Chia, W.C. Multiview Image Block Compressive Sensing with Joint Multiphase Decoding for Visual Sensor Network. *ACM Trans. Multimed. Comput. Commun. Appl.* **2015**, *12*, 30. [CrossRef]
41. Zhang, J.; Xiang, Q.; Yin, Y.; Chen, C.; Luo, X. Adaptive Compressed Sensing for Wireless Image Sensor Networks. *Multimed. Tools Appl.* **2017**, *76*, 4227–4242. [CrossRef]
42. Deligiannis, N.; Verbist, F.; Iossifides, A.C.; Slowack, J.; de Walle, R.V.; Schelkens, R.; Muntenau, A. Wyner-Ziv Video Coding for Wireless Lightweight Multimedia Applications. *J. Wirel. Commun. Netw.* **2012**, *2012*, 1–20. [CrossRef]
43. Yeo, C.; Ramchandran, K. Robust Distributed Multiview Video Compression for Wireless Camera Networks. *IEEE Trans. Image Process.* **2010**, *19*, 995–1008. [PubMed]
44. Heng, S.; So-In, C.; Nguyen, T. Distributed Image Compression Architecture over Wireless Multimedia Sensor Networks. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 5471721. [CrossRef]
45. Hanca, J.; Deligiannis, N.; Munteanu, A. Real-time Distributed Video Coding for 1K-Pixel Visual Sensor Networks. *J. Electron. Imag.* **2016**, *25*, 1–20. [CrossRef]
46. Luong, H.V.; Deligiannis, N.; Forchhammer, S.; Kaup, A. Distributed Coding of Multiview Sparse Sources with Joint Recovery. In Proceedings of the 2016 Picture Coding Symposium (PCS), Nuremberg, Germany, 4–7 December 2016; pp. 1–5.
47. Wang, X.; Şekercioğlu, Y.A.; Drummond, T. Multiview Image Compression and Transmission Techniques in Wireless Multimedia Sensor Networks: A Survey. In Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013), Palm Springs, CA, USA, 39 October–1 November 2013.
48. Kadkhodamohammadi, A.; Gangi, A.; de Mathelin, M.; Padoy, N. A Multi-view RGB-D Approach for Human Pose Estimation in Operating Rooms. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017.
49. Shen, J.; Su, P.C.; Cheung, S.; Zhao, J. Virtual Mirror Rendering With Stationary RGB-D Cameras and Stored 3D Background. *IEEE Trans. Image Process.* **2013**, *22*, 3433–3448. [CrossRef] [PubMed]
50. Stamm, C. A New Progressive File Format for Lossy and Lossless Image Compression. In Proceedings of the International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic, 4–8 February 2002; pp. 30–33.
51. San, X.; Cai, H.; Lou, J.G.; Li, J. Multiview Image Coding Based on Geometric Prediction. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1536–1548.
52. Ma, H.; Liu, Y. Correlation Based Video Processing in Video Sensor Networks. In Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing, Maui, HI, USA, 13–16 June 2005; Volume 2, pp. 987–992.
53. Corke, P. *Robotics, Vision and Control*; Chapter 2—Representing Position and Orientation; Springer: Berlin, Germany, 2011.
54. Benjemaa, R.; Schmitt, F. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. *Image Vis. Comput.* **1999**, *17*, 113–123. [CrossRef]
55. Lui, W.; Tang, T.; Drummond, T.; Li, W.H. Robust Egomotion Estimation Using ICP in Inverse Depth Coordinates. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2012), St. Paul, MN, USA, 14–18 May 2012; pp. 1671–1678.
56. Drummond, T.; Cipolla, R. Real-Time Visual Tracking of Complex Structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 932–946. [CrossRef]
57. Drummond, T. *Lie Groups, Lie Algebras, Projective Geometry and Optimization for 3D Geometry, Engineering and Computer Vision*. Available online: https://twd20g.blogspot.com/2014/02/updated-notes-on-lie-groups.html (accessed on 24 July 2018).
58. Holland, P.W.; Welsch, R.E. Robust Regression Using Iteratively Reweighted Least-Squares. *Commun. Stat. Theory Methods* **1977**, *6*, 813–827. [CrossRef]
59. Skodras, A.; Christopoulos, C.; Touradj, E. The JPEG 2000 Still Image Compression Standard. *IEEE Signal Process. Mag.* **2001**, *18*, 36–58. [CrossRef]

60. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H. 264/AVC Video Coding Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [CrossRef]
61. libPGF Project. Available online: http://www.libpgf.org/ (accessed on 24 July 2018).
62. Xi, M.; Xue, J.; Wang, L.; Li, D.; Zhang, M. A Novel Method of Multi-view Virtual Image Synthesis for Auto-stereoscopic Display. In *Advanced Technology in Teaching*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 163, pp. 865–873.
63. Fickel, G.P.; Jung, C.R.; Lee, B. Multiview Image and Video Interpolation Using Weighted Vector Median Filters. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5387–5391.
64. Mori, Y.; Fukushima, N.; Fujii, T.; Tanimoto, M. View Generation with 3D Warping Using Depth Information for FTV. *Signal Process. Image Commun.* **2009**, *24*, 65–72. [CrossRef]
65. Do, L.; Zinger, S.; Morvan, Y.; de With, P. Quality Improving Techniques in DIBR for Free-Viewpoint Video. In Proceedings of the 2009 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video, Potsdam, Germany, 4–6 May 2009; pp. 1–4.
66. Besl, P.; McKay, N.D. A Method for Registration of 3D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
67. Technical University of Munich Computer Vision Group RGB-D SLAM Dataset and Benchmark. Available online: http://vision.in.tum.de/data/datasets/rgbd-dataset (accessed on 24 July 2018).
68. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the International Conference on Intelligent Robot Systems (IROS 2012), Vilamoura, Algarve, 7–12 October 2012; pp. 573–580.
69. D'Ademo, N.; Lui, W.L.D.; Li, W.H.; Şekercioğlu, Y.A.; Drummond, T. eBug: An Open Robotics Platform for Teaching and Research. In Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2011), Melbourne, Australia, 7–9 December 2011.
70. EyeBug—A Simple, Modular and Cheap Open-Source Robot. 2011. Available online: http://www.robaid.com/robotics/eyebug-a-simple-and-modular-cheap-open-source-robot.htm (accessed on 24 July 2018).
71. Beagleboard-xM Single Board Computer. Available online: http://beagleboard.org/beagleboard-xm (accessed on 24 July 2018).
72. OpenKinect Library. Available online: http://openkinect.org (accessed on 24 July 2018).
73. OpenCV: Open Source Computer Vision Library. Available online: http://opencv.org (accessed on 25 July 2018).
74. LibCVD—Computer Vision Library. Available online: http://www.edwardrosten.com/cvd/ (accessed on 25 July 2018).
75. Colonnese, S.; Cuomo, F.; Melodia, T. Leveraging Multiview Video Coding in Clustered Multimedia Sensor Networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 475–480.
76. Gehrig, N.; Dragotti, P.L. Distributed Compression of Multi-View Images Using a Geometrical Coding Approach. In Proceedings of the IEEE International Conference on Image Processing (ICIP), San Antonio, TX, USA, 16–19 September 2007; Volume 6.
77. Merkle, P.; Morvan, Y.; Smolic, A.; Farin, D.; Mueller, K.; de With, P.H.N.; Wiegand, T. The Effects of Multiview Depth Video Compression on Multiview Rendering. *Signal Process. Image Commun.* **2009**, *24*, 73–88. [CrossRef]
78. Vijayanagar, K.; Loghman, M.; Kim, J. Refinement of Depth Maps Generated by Low-Cost Depth Sensors. In Proceedings of the 2012 International SoC Design Conference (ISOCC), Jeju Island, Korea, 4–7 November 2012; pp. 355–358.
79. Vijayanagar, K.; Loghman, M.; Kim, J. Real-Time Refinement of Kinect Depth Maps using Multi-Resolution Anisotropic Diffusion. *Mob. Netw. Appl.* **2014**, *19*, 414–425. [CrossRef]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).