A Novel Kinect V2 Registration Method Using Color and Deep Geometry Descriptors

Conference Paper · September 2018

DOI: 10.23919/EUSIPCO.2018.8553327

Yuan Gao
Christian-Albrechts-Universität zu Kiel
8 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Reinhard Koch
Christian-Albrechts-Universität zu Kiel
246 PUBLICATIONS **6,319** CITATIONS
SEE PROFILE

A Novel Kinect V2 Registration Method Using Color and Deep Geometry Descriptors

Yuan Gao, Tim Michels and Reinhard Koch
Christian-Albrechts-University of Kiel, 24118 Kiel, Germany
{yga, tmi, rk}@informatik.uni-kiel.de

***Abstract*—The novel view synthesis for traditional sparse light field camera arrays generally relies on an accurate depth approximation for a scene. To this end, it is preferable for such camera-array systems to integrate multiple depth cameras (*e.g.* Kinect V2), thereby requiring a precise registration for the integrated depth sensors. Methods based on special calibration objects have been proposed to solve the multi-Kinect V2 registration problem by using the prebuilt geometric relationships of several easily-detectable common point pairs. However, for registration tasks incapable of knowing these precise geometric relationships, this kind of method is prone to fail. To overcome this limitation, a novel Kinect V2 registration approach in a coarse-to-fine framework is proposed in this paper. Specifically, both local color and geometry information is extracted directly from a static scene to recover a rigid transformation from one Kinect V2 to the other. Besides, a 3D convolutional neural network (ConvNet), *i.e.* 3DMatch, is utilized to describe local geometries. Experimental results show that the proposed Kinect V2 registration method using both color and deep geometry descriptors outperforms the other coarse-to-fine baseline approaches.**

I. Introduction

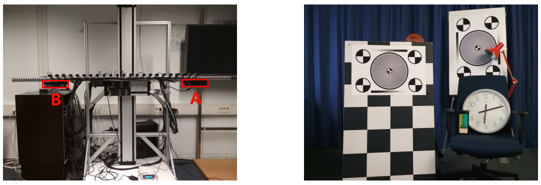
The second version of the Microsoft Kinect (Kinect V2) is one of the most widespread low-cost Time-of-Flight (ToF) sensors available in the market [1]. The comparison between the Kinect V2 and the first generation of Microsoft Kinect (Kinect V1) is well studied in [2], where the Kinect V2 has a higher accuracy but a lower precision than the Kinect V1 [3].

A. Motivation

The multi-camera rig illustrated in Fig. 1 (a) is a movable camera array [4] for capturing dynamic light fields [5]. The precise calibration of the two Kinect V2 sensors on this rig is critical to the dense 3D reconstruction of a large-scale and non-rigid scene [6], which can be further used for the novel view synthesis in the Free Viewpoint Video (FVV) [7] and Head-Mounted Display (HMD) [8] systems, together with the dynamic light fields captured by the sparse RGB camera array and densely reconstructed by [9]–[14]. Therefore, an automatic Kinect V2 registration method without relying on any calibration object would be highly desirable for this system, considering that the positions of the two Kinect V2 cameras may be changed for different scenes of varying sizes and the preparation phase of calibration object-based registration methods may be time-consuming and cumbersome.

B. Related Work

As for solving the registration problem of multiple depth cameras with using calibration objects, several methods have been proposed. Afzal *et al.* propose an RGB-D multi-view system calibration method, *i.e.* BAICP+, which combines Bundle Adjustment (BA) [15] and Iterative Closest Point (ICP) [16] into a single minimization framework [17]. The corners of a checkerboard are detected for the BA part of BAICP+. Kowalski *et al.* present a coarse-to-fine solution for the multi-Kinect V2 calibration problem, where a planar marker is used for the rough estimation of camera poses, which is later refined by an ICP algorithm [18]. Soleimani *et al.* employ three double-sided checkerboards placed at varying depths for an automatic calibration process of two opposing Kinect V2 cameras [19]. Córdova-Esparza *et al.* introduce a calibration tool for multiple Kinect V2 sensors using a 1D calibration object, *i.e.* a wand, which has three collinear points [20]. Regarding the Kinect V2 registration solution without using calibration objects, Gao *et al.* propose a coarse-to-fine Kinect V2 calibration approach using camera and scene constraints for two Kinect V2 cameras with a large displacement [21].



(a) A multi-camera system.

(b) A static scene.

Figure 1. The two Kinect V2 cameras are fixed on a movable multi-camera rig. The static scene shown in (b) is used for experiments.

In this paper, to solve the registration problem of two Kinect V2 cameras, a novel camera calibration method for Kinect V2 sensors using local color and geometry information is proposed. Specifically, an off-the-shelf feature detector is used for detecting interest points and describing local color information for them. Afterwards, a ConvNet-based 3D descriptor, 3DMatch [22], is utilized to describe local geometry information for these interest points. Both color and geometry descriptors are employed to estimate an initial rough rigid transformation between two Kinect V2 cameras, which can then be refined by an optional estimation refinement step if necessary. Experimental results prove the effectiveness of the proposed method by comparing it with baseline approaches.

II. Methodology

A. Preliminary

The two Kinect V2 cameras mounted on the multi-camera rig are denoted by $\mathbb{C}_\text{A}$ and $\mathbb{C}_\text{B}$, respectively. Since the intrinsic parameters and lens distortion of the ToF sensor in a Kinect V2 can be calibrated in advance or extracted from the factory calibration by using the Kinect for Windows SDK, the

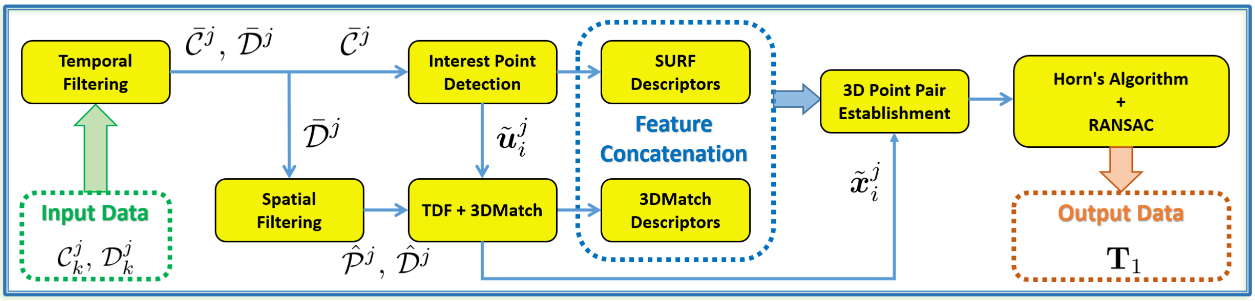

Temporal Filtering
$\bar{\mathcal{C}}^j$, $\bar{\mathcal{D}}^j$
$\bar{\mathcal{C}}^j$
Interest Point Detection
SURF Descriptors
Feature Concatenation
3D Point Pair Establishment
Horn's Algorithm + RANSAC
$\bar{\mathcal{D}}^j$
$\tilde{\boldsymbol{u}}_i^j$
$\tilde{\boldsymbol{x}}_i^j$
Input Data
$\mathcal{C}_k^j$, $\mathcal{D}_k^j$
Spatial Filtering
TDF + 3DMatch
3DMatch Descriptors
$\hat{\mathcal{P}}^j$, $\hat{\mathcal{D}}^j$
Output Data
$\mathbf{T}_1$

Figure 2. A flow chart of the proposed Kinect V2 registration method in the coarse estimation phase.

registration problem of two Kinect V2 cameras is interpreted as how to calculate a rigid transformation between these two cameras. Suppose a rigid transformation is expressed as

$$\mathbf{T}_i = \begin{pmatrix} \mathbf{R}_i & \boldsymbol{t}_i \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbb{SE}(3), \tag{1}$$

where $\mathbf{R}_i \in \mathbb{SO}(3)$ and $\boldsymbol{t}_i \in \mathbb{R}^3$. A coarse-to-fine camera registration framework [23], [24] is defined as estimating the rigid transformation from $\mathbb{C}_\mathrm{A}$ to $\mathbb{C}_\mathrm{B}$ via two steps:

$$\mathbf{T} = \mathbf{T}_2\mathbf{T}_1. \tag{2}$$

Here, for the rigid transformation matrix $\mathbf{T}_i$, $i \in \{1, 2\}$ stands for the case of using a coarse estimation method in Section II-B and the other case of using an estimation refinement approach in Section II-C, respectively. The camera coordinate system of the ToF sensor in a Kinect V2 camera is specified as the camera space of this Kinect V2. The intrinsic parameters of $\mathbb{C}_\mathrm{A}$ or $\mathbb{C}_\mathrm{B}$ are represented by the focal lengths f_x^j, f_y^j and the principal point $\left(c_x^j, c_y^j\right)^\mathrm{T}$, where $j \in \{a, b\}$. The lens distortion coefficients are utilized to eliminate distortions before saving any pair of registered color and depth images, denoted by $\mathcal{C}^j$ and $\mathcal{D}^j$, both of which are from the camera image plane of the ToF sensor in a Kinect V2.

B. *Coarse Estimation*

A marker that can be simultaneously captured by a pair of Kinect V2 cameras aids establishing reliable corresponding 2D point pairs on $\mathcal{C}^a$ and $\mathcal{C}^b$ [18]. One of these corresponding 2D point pairs is expressed as $\left(\boldsymbol{u}_i^a, \boldsymbol{u}_i^b\right)$, where $\boldsymbol{u}_i^j = \left(u_i^j, v_i^j, 1\right)^\mathrm{T}$, $j \in \{a, b\}$. The depth value d_i^j for a 2D point $\boldsymbol{u}_i^j$ is acquired from its respective depth image $\mathcal{D}^j$, *i.e.* $d_i^j = \mathcal{D}^j\left(v_i^j, u_i^j\right)$. Defining $s : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^4$ to be a back-projection function, which projects a 2D point $\boldsymbol{u}_i^j$ on the camera image plane to a 3D point $\boldsymbol{x}_i^j = \left(x_i^j, y_i^j, z_i^j, 1\right)^\mathrm{T}$ in the camera space,

$$s(\boldsymbol{u}_i^j, d_i^j) = \left(\frac{(u_i^j - c_x^j)d_i^j}{f_x^j}, \quad \frac{(v_i^j - c_y^j)d_i^j}{f_y^j}, \quad d_i^j, \quad 1 \right)^\mathrm{T}, \tag{3}$$

and $\boldsymbol{x}_i^j = s(\boldsymbol{u}_i^j, d_i^j)$. The 2D point pair $\left(\boldsymbol{u}_i^a, \boldsymbol{u}_i^b\right)$ is therefore able to be turned into a 3D point pair $\left(\boldsymbol{x}_i^a, \boldsymbol{x}_i^b\right)$ by (3). The coarse rigid transformation $\mathbf{T}_1$ is estimated by

$$\arg \min_{\mathbf{R}_1 \in \mathbb{SO}(3),\, \boldsymbol{t}_1 \in \mathbb{R}^3} \sum_{i=1}^{n} \frac{1}{2} \left\| \begin{pmatrix} \mathbf{R}_1 & \boldsymbol{t}_1 \\ \mathbf{0} & 1 \end{pmatrix} \boldsymbol{x}_i^a - \boldsymbol{x}_i^b \right\|_2^2. \tag{4}$$

The minimization problem in (4) can be turned into the Orthogonal Procrustes problem [25] and solved by the least-squares fitting algorithm [26] efficiently, requiring at least three corresponding 3D point pairs, *i.e.* $n \geqslant 3$.

However, preparing some special calibration objects for the Kinect V2 registration task is sometimes time- and effort-consuming. How to solve the Kinect V2 registration problem by only using the information from a nature scene is more challenging than the above case of using a marker. To deal with this problem, a novel coarse estimation framework is proposed and presented in Fig. 2. This framework exploits both color and geometry feature descriptors to estimate a rough rigid transformation $\mathbf{T}_1$ between two Kinect V2 cameras. Details about it are described as below:

1) Input Data: Due to the precision problem [3] of the ToF sensor of any Kinect V2, multi-frame depth information is used to improve the quality of the captured depth images. For a static scene and a static multi-camera system, m consecutive depth and color frames are captured by both $\mathbb{C}_\mathrm{A}$ and $\mathbb{C}_\mathrm{B}$ simultaneously. The input data for the coarse estimation framework are $\mathcal{C}_k^j$ and $\mathcal{D}_k^j$, where $k \in \mathbb{Z}^+$, $k \leqslant m$, and $j \in \{a, b\}$.

2) Temporal Filtering: A temporal mean filter is used here to calculate an average depth image $\bar{\mathcal{D}}^j$ for all the $\mathcal{D}_k^j$ images. Note that an underlying depth-validity check is also performed by this depth temporal mean filter. In particular, only depth image pixels with depth values larger than $0.5\,\mathrm{m}$ are treated as valid pixels for the accumulated weights. A corresponding average color image $\bar{\mathcal{C}}^j$ is accordingly generated by using all the $\mathcal{C}_k^j$ images and the same accumulated weights with valid pixel positions from the depth temporal filtering process.

3) Spatial Filtering: The mean depth image $\bar{\mathcal{D}}^j$ is then projected into a point cloud $\bar{\mathcal{P}}^j$ in the camera space of the Kinect V2 by using (3). However, the resulting point cloud $\bar{\mathcal{P}}^j$ may still have some outliers or noisy data, some of which are far away from the real captured scene. This will increase the volume allocation for the volumetric representation in the following steps, which may lead to a failure if limited memory is available in hardware, *e.g.* GPU. To handle this problem, a statistical spatial filtering method is utilized to trim the outliers of $\bar{\mathcal{P}}^j$. To be precise, each 3D point $\boldsymbol{x}_i^j$ in this point cloud has a mean distance t_i^j to its l nearest neighbor 3D points. A 3D point $\boldsymbol{x}_i^j$ will be removed if its distance t_i^j is not inside the range determined by the global distances mean and standard deviation. The filtered point cloud is denoted by $\hat{\mathcal{P}}^j$ and projected back onto the camera image plane by using

$$\pi(\boldsymbol{x}_i^j) = \left(\frac{f_x^j x_i^j}{z_i^j} + c_x^j, \quad \frac{f_y^j y_i^j}{z_i^j} + c_y^j, \quad 1 \right)^\mathrm{T}, \tag{5}$$

which generates a filtered depth image $\hat{\mathcal{D}}^j$ accordingly.

Algorithm 1: An ICP-based estimation refinement algorithm.

```
Input  : $\hat{\mathcal{P}}^j$ from Section II-B3, Rigid transformation $\mathbf{T}_1$.
Output: Rigid transformation $\mathbf{T}_2$.
/* Step 1: Transform $\hat{\mathcal{P}}^a$ from $\mathbb{C}_\mathrm{A}$ to $\mathbb{C}_\mathrm{B}$ coordinates */
foreach point $\boldsymbol{x}_i^a$ in $\hat{\mathcal{P}}^a$ do $\boldsymbol{x}_i^a \leftarrow \mathbf{T}_1\boldsymbol{x}_i^a$;
/* Step 2: Point cloud registration */
$\tau \leftarrow 0.005$;
$e \leftarrow +\infty$, $\check{e} \leftarrow 0$, $\dot{e} \leftarrow 0$;
$\mathbf{T}^a \leftarrow \mathbf{I}_4$, $\mathbf{T}^b \leftarrow \mathbf{I}_4$, $\dot{\mathbf{T}} \leftarrow \mathbf{I}_4$; /* $\mathbf{I}_n$: $n \times n$ identity matrix */
while true do
    $\check{\mathbf{T}}^a \leftarrow \mathbf{T}^a$;
    $\check{\mathbf{T}}^b \leftarrow \mathbf{T}^b$;
    $\check{e} \leftarrow e$;
    $e \leftarrow 0$;
    $\dot{\mathbf{T}}$, $\dot{e} \leftarrow$ ICP($\hat{\mathcal{P}}^a$, $\hat{\mathcal{P}}^b$); /* $\dot{e}$: Average error per point */
    foreach point $\boldsymbol{x}_i^a$ in $\hat{\mathcal{P}}^a$ do $\boldsymbol{x}_i^a \leftarrow \dot{\mathbf{T}}\boldsymbol{x}_i^a$;
    $\mathbf{T}^a \leftarrow \dot{\mathbf{T}}\mathbf{T}^a$;
    $e \leftarrow e + \dot{e}$;
    $\dot{\mathbf{T}}$, $\dot{e} \leftarrow$ ICP($\hat{\mathcal{P}}^b$, $\hat{\mathcal{P}}^a$);
    foreach point $\boldsymbol{x}_i^b$ in $\hat{\mathcal{P}}^b$ do $\boldsymbol{x}_i^b \leftarrow \dot{\mathbf{T}}\boldsymbol{x}_i^b$;
    $\mathbf{T}^b \leftarrow \dot{\mathbf{T}}\mathbf{T}^b$;
    $e \leftarrow e + \dot{e}$;
    if $e > \check{e}$ then
        $\mathbf{T}^a \leftarrow \check{\mathbf{T}}^a$;
        $\mathbf{T}^b \leftarrow \check{\mathbf{T}}^b$;
        break;
    if $\frac{\check{e}-e}{e} < \tau$ then break;
$\mathbf{T}_2 \leftarrow (\mathbf{T}^b)^{-1}\mathbf{T}^a$.
```

4) Interest Point Detection: The Speeded Up Robust Features (SURF) have robust and stable performance in computer vision and robotics applications [27]. The SURF interest point detector is used to detect 2D keypoints on the average color image $\bar{\mathcal{C}}^j$ from the temporal filtering step (Section II-B2). The coordinates of all the keypoints are fed to the next step for geometry feature calculation. Besides, for each detected 2D interest point $\tilde{\boldsymbol{u}}_i^j$, the SURF algorithm also generates a SURF descriptor $\tilde{\boldsymbol{\omega}}_i^j \in \mathbb{R}^{64}$, which is a normalized vector.

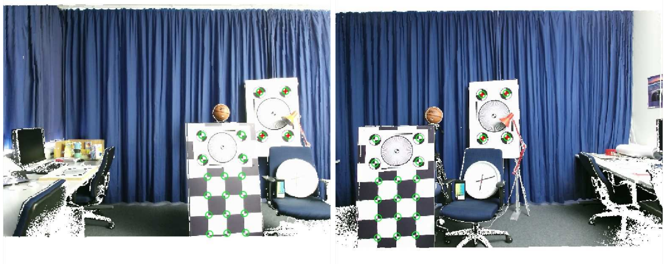
5) TDF and 3DMatch: The Truncated Distance Function (TDF) representation is a variation of Truncated Signed Distance Function (TSDF) [28]. The filtered point cloud $\hat{\mathcal{P}}^j$ is assigned to a volumetric grid of voxels to calculate the TDF value for each voxel. As for each 2D interest point $\tilde{\boldsymbol{u}}_i^j$, a corresponding 3D interest point $\tilde{\boldsymbol{x}}_i^j$ is computed by (3) with its depth information from $\hat{\mathcal{D}}^j$. A volumetric 3D patch for each $\tilde{\boldsymbol{x}}_i^j$ is then extracted from the volumetric grid, *i.e.*, $\tilde{\boldsymbol{x}}_i^j$ is in the center of a $30 \times 30 \times 30$ local voxel grid. The extracted volumetric 3D patch is finally fed into a pre-trained network of 3DMatch to generate a local geometry descriptor $\tilde{\boldsymbol{\epsilon}}_i^j \in \mathbb{R}^{512}$.

6) Feature Concatenation: To make full use of different advantages of the SURF and 3DMatch descriptors for the scene representation, a feature concatenation strategy is proposed as below:

$$\tilde{\boldsymbol{\rho}}_i^j = (1-\lambda)\tilde{\boldsymbol{\omega}}_i^j \oplus \lambda\tilde{\boldsymbol{\epsilon}}_i^j = \begin{pmatrix} (1-\lambda)\tilde{\boldsymbol{\omega}}_i^j \\ \lambda\tilde{\boldsymbol{\epsilon}}_i^j \end{pmatrix}, \lambda \in [0,1]. \tag{6}$$

The resulting concatenated descriptor is denoted by $\tilde{\boldsymbol{\rho}}_i^j \in \mathbb{R}^{576}$.

7) 3D Point Pair Establishment: After constructing the concatenated feature descriptor $\tilde{\boldsymbol{\rho}}_i^j$ for each 3D interest point $\tilde{\boldsymbol{x}}_i^j$, the reliable corresponding 3D point pairs in the two Kinect V2 camera spaces are established by means of the k-d tree data structure [29] and k-Nearest-Neighbors algorithm [30].



(a) Average color image $\bar{\mathcal{C}}^a$. (b) Average color image $\bar{\mathcal{C}}^b$.

Figure 3. The average color images from the temporal filtering step (Section II-B2). Green circles and red crosses stand for the corners of check patterns.

8) Horn's Algorithm and RANSAC: The final rigid transformation $\mathbf{T}_1$ from $\mathbb{C}_\mathrm{A}$ to $\mathbb{C}_\mathrm{B}$ for the coarse estimation step is calculated by using the Horn's algorithm [31] together with the RANdom SAmple Consensus (RANSAC) method [32] for solving the least squares problem defined in (4).

C. Estimation Refinement

The algorithm for estimation refinement is depicted in Algorithm 1. The input data for this algorithm are the rough rigid transformation $\mathbf{T}_1$ of the previous coarse estimation stage and point clouds $\hat{\mathcal{P}}^a$ and $\hat{\mathcal{P}}^b$ from the spatial filtering step (Section II-B3). The point cloud $\hat{\mathcal{P}}^a$ is firstly transformed into the camera coordinate system of $\mathbb{C}_\mathrm{B}$. Afterwards, the two point clouds in the same camera space are registered by using an ICP-based method, which in this case is equal to the camera pose refinement. The final estimation refinement result $\mathbf{T}_2$ is recovered from two intermediate rigid transformation matrices $\mathbf{T}^a$ and $\mathbf{T}^b$.

III. Experiments

A. Experimental Settings

1) Camera Setup: The equipment for capturing experimental data is a multi-camera system as shown in Fig. 1 (a). This system has two Kinect V2 cameras with similar orientations. The horizontal displacement between them is around 1.5 m. The Kinect for Windows SDK is leveraged to capture a static scene for both $\mathbb{C}_\mathrm{A}$ and $\mathbb{C}_\mathrm{B}$. The intrinsic parameters f_x^j, f_y^j, c_x^j, c_y^j and radial distortion coefficients [33] are extracted from the hardware of Kinect V2 sensors by using this SDK.

2) Static Scene: An example image of the static scene is exhibited in Fig. 1 (b). The positions of check patterns in the scene are adopted in the following evaluation metric step. The size of this scene is $5.5 \times 3.0 \times 3.6\,\mathrm{m}^3$ ($w \times h \times d$). The number of captured color or depth frames, *i.e.* m in Section II-B1, is equal to 31. The average color images of $\mathbb{C}_\mathrm{A}$ and $\mathbb{C}_\mathrm{B}$ described in Section II-B2 are presented in Fig. 3.

3) Evaluation Metric: The corners of the check patterns on the average RGB images $\bar{\mathcal{C}}^a$ and $\bar{\mathcal{C}}^b$ are manually labeled in order to establish several common-corner 2D point pairs. Afterwards, an automatic corner refinement approach with sub-pixel accuracy is employed to refine the coordinates of these 2D corner points [34]. Let a common-corner 2D point pair be denoted by $\left(\boldsymbol{u}_i^a, \boldsymbol{u}_i^b\right)$ as the description in Section II-B. This 2D point pair is then converted into a 3D point

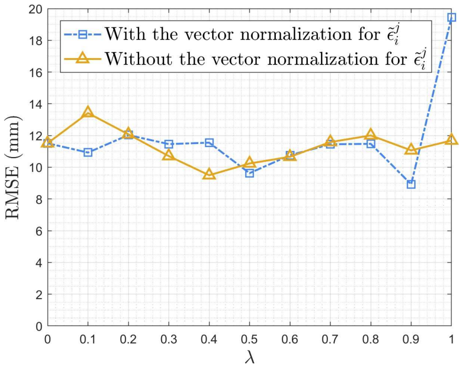


Figure 4. The influence of changing λ in the feature concatenation strategy, *i.e.* (6), on the registration performance of two Kinect V2 cameras.

pair $\left(\boldsymbol{x}_i^a, \boldsymbol{x}_i^b\right)$ by using (3) and $\hat{\mathcal{D}}^j$. Note that, because of the intensity-related distance error [35], [36] of any ToF sensor, the depth value d_i^j for a 2D corner point $\boldsymbol{u}_i^j$ is filtered by a specific filter in [37], where the depth information of only the white checks around $\boldsymbol{u}_i^j$ is taken into account. The Root-Mean-Square Error (RMSE) metric is applied to evaluate the performance of different Kinect V2 registration methods:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{T}\boldsymbol{x}_i^a - \boldsymbol{x}_i^b\right\|_2^2}. \tag{7}$$

Here, $n = 20$. All the common-corner 2D point pairs are indicated by green circles in Fig. 3. Besides, the four common-corner 2D point pairs represented by red crosses on a board are utilized to calculate the coarse estimation result in LiveScan3D [18]. Note that this board plays the same role as a marker.

4) Implementation Details: The SURF interest point detector and feature descriptor are implemented by referring to their implementations in OpenCV with default parameters. Each voxel in the volumetric grid of the TDF representation has the same size of $0.01^3\,\text{m}^3$. The pre-trained 3DMatch network from [22] has been optimized on multiple scene reconstruction datasets in diverse real-world environments at varying scales.

B. Results and Analysis

1) Quantitative Evaluation: The varying λ in Section II-B6 for the feature concatenation strategy has different impacts on the performance of the coarse estimation phase as shown in Fig. 4. The yellow solid line stands for the registration precision of changing λ in (6). It can be found that only using SURF descriptor ($\lambda = 0$) and using 3DMatch descriptor alone ($\lambda = 1$) have similar RMSE results ($\approx 11.6\,\text{mm}$), which indicates that both color and geometry descriptors in the coarse estimation stage are effective for the calibration of the two Kinect V2 cameras. Besides, when $\lambda = 0.4$, the best camera registration performance is achieved (RMSE $= 9.497\,\text{mm}$), which implies that the combination of both color and geometry information is beneficial for the camera registration task of Kinect V2 sensors. Since the 3DMatch descriptor $\tilde{\boldsymbol{\epsilon}}_i^j$ is not normalized, a vector normalization method is tried here through dividing $\tilde{\boldsymbol{\epsilon}}_i^j$ by a Euclidean norm before the concatenation operation for the feature descriptors. The blue dash line reveals the performance of feature concatenation using the normalized $\tilde{\boldsymbol{\epsilon}}_i^j$ at varying λ. When using the normalized 3DMatch descriptor alone ($\lambda = 1$), the RMSE value increases dramatically compared with the case of using the original 3DMatch descriptor alone, which suggests that the vector normalization for $\tilde{\boldsymbol{\epsilon}}_i^j$ is not helpful for the registration of the Kinect V2 cameras. Moreover, a reasonable best registration performance is achieved at $\lambda = 0.5$, which demonstrates that both color and geometry descriptors are of equal importance for the coarse rigid transformation estimation again.

Table I
THE RMSE RESULTS OF DIFFERENT METHODS.

Method	Coarse Estimation (mm)	Estimation Refinement (mm)
LiveScan3D [18]	12.714	20.116
Gao *et al.* [21]	79.037	32.416
Proposed	**9.497**	20.221

The performance comparison between the proposed method and baseline approaches is illustrated in Table I. Here, for the proposed method, $\lambda = 0.4$ without 3DMatch descriptor normalization is used for the performance comparison, which is explained by the detailed analysis as above. As can be seen from the table, the proposed Kinect V2 registration method with only using coarse estimation achieves the best performance, which proves the effectiveness of the proposed camera registration method for Kinect V2 sensors using both color and deep geometry information. However, the estimation refinement step does not reduce the RMSE values for LiveScan3D and the proposed method, which means that the ICP-based estimation refinement algorithm may get stuck in a local minimum that can be even worse than an initialization, *i.e.* the coarse estimation result. The estimation refinement step is effective only for method [21], whereas its performance is the worst among these three approaches, which suggests that estimation refinement will be a necessary step if the camera registration error of coarse estimation is large.

2) Qualitative Evaluation: The proposed Kinect V2 registration method is also evaluated qualitatively as illustrated in Fig. 5. Here, for each Kinect V2 camera, an integration algorithm in KinectFusion [38] is adopted to fuse all the depth images $\mathcal{D}_k^j$ into a 3D voxel grid using a volumetric TSDF representation [28]. Specifically, a projective point-to-point distance metric for the voxel-to-surface distance approximation and a constant weighting function are used in this integration algorithm [39]. Afterwards, the marching-cubes algorithm is utilized to extract a mesh standing for the zero-level isosurface encoded by the TSDF representation [40]. In Fig. 5, the yellow mesh comes from $\mathbb{C}_\text{A}$ and it has been transformed into the camera coordinates of $\mathbb{C}_\text{B}$ by using the rigid transformation result, *i.e.* $\mathbf{T}_1$, of the proposed method. The gray mesh is from $\mathbb{C}_\text{B}$. It is apparent that these two meshes coincide very well, which demonstrates that the proposed Kinect V2 registration method using feature concatenation strategy for both SURF and 3DMatch features is effective for the Kinect V2 calibration problem in this static scene.

IV. CONCLUSION

In this paper, a Kinect V2 registration method using color (SURF) and deep geometry (3DMatch) feature descriptors is presented. The proposed method is integrated into a coarse-to-fine framework and it achieves better performance in the

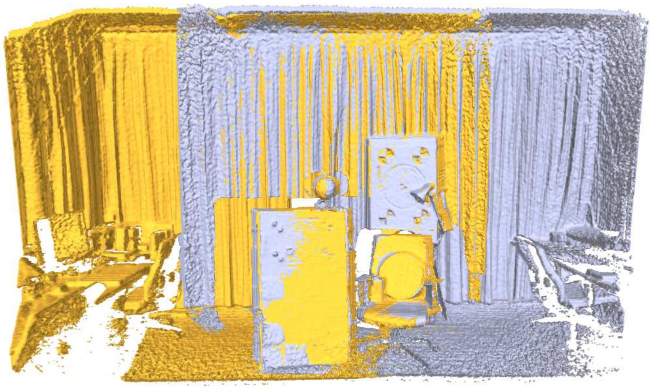


Figure 5. The visualized camera registration result of the proposed method using a TSDF representation. The yellow mesh is from $\mathbb{C}_\mathrm{A}$ and the gray mesh is from $\mathbb{C}_\mathrm{B}$. Both of them are in the camera space of $\mathbb{C}_\mathrm{B}$.

coarse estimation stage than in the estimation refinement phase for a static scene. Moreover, for the proposed method, using the combination of color and geometry features performs better than using color or geometry feature alone. Furthermore, the experimental performance comparison shows the superiority of the proposed method over other baseline approaches.

Acknowledgments

The work in this paper was funded from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 676401, European Training Network on Full Parallax Imaging, and the German Research Foundation (DFG) No. K02044/8-1.

References

[1] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the Kinect V2 time-of-flight camera," *Robotics and Autonomous Systems*, vol. 75, pp. 584–594, 2016. 1

[2] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight Kinect," *CVIU*, vol. 139, pp. 1–20, 2015. 1

[3] O. Wasenmüller and D. Stricker, "Comparison of Kinect V1 and V2 depth images in terms of accuracy and precision," in *ACCV Workshops*, 2016, pp. 34–45. 1, 2

[4] S. Esquivel, Y. Gao, T. Michels, L. Palmieri, and R. Koch, "Synchronized data capture and calibration of a large-field-of-view moving multi-camera light field rig," in *3DTV-CON Workshops*, 2016. 1

[5] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE J-STSP*, vol. 11, no. 7, pp. 926–954, 2017. 1

[6] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *CVPR*, 2015, pp. 343–352. 1

[7] A. Smolic, "3D video and free viewpoint video - from capture to display," *Pattern Recognition*, vol. 44, no. 9, pp. 1958–1968, 2011. 1

[8] J. Yu, "A light-field journey to virtual reality," *IEEE MultiMedia*, vol. 24, no. 2, pp. 104–112, 2017. 1

[9] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE TPAMI*, vol. 40, no. 1, pp. 133–147, 2018. 1

[10] Y. Gao and R. Koch, "Parallax view generation for static scenes using parallax-interpolation adaptive separable convolution," in *ICME Workshops*, 2018. 1

[11] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Accelerated shearlet-domain light field reconstruction," *IEEE J-STSP*, vol. 11, no. 7, pp. 1082–1091, 2017. 1

[12] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on EPI," in *CVPR*, 2017, pp. 1638–1646. 1

[13] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM TOG*, vol. 35, no. 6, pp. 193:1–193:10, 2016. 1

[14] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Image based rendering technique via sparse representation in shearlet domain," in *ICIP*, 2015, pp. 1379–1383. 1

[15] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - A modern synthesis," in *Vision Algorithms: Theory and Practice*, 2000, pp. 298–372. 1

[16] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE TPAMI*, vol. 14, no. 2, pp. 239–256, 1992. 1

[17] H. Afzal, D. Aouada, D. Font, B. Mirbach, and B. Ottersten, "RGB-D multi-view system calibration for full 3D scene reconstruction," in *ICPR*, 2014, pp. 2459–2464. 1

[18] M. Kowalski, J. Naruniec, and M. Daniluk, "LiveScan3D: A fast and inexpensive 3D data acquisition system for multiple Kinect v2 sensors," in *3DV*, 2015, pp. 318–325. 1, 2, 4

[19] V. Soleimani, M. Mirmehdi, D. Damen, S. Hannuna, and M. Camplani, "3D data acquisition and registration using two opposing kinects," in *3DV*, 2016, pp. 128–137. 1

[20] D.-M. Córdova-Esparza, J. R. Terven, H. Jiménez-Hernández, and A.-M. Herrera-Navarro, "A multiple camera calibration and point cloud fusion tool for Kinect V2," *SCP*, vol. 143, pp. 1–8, 2017. 1

[21] Y. Gao, S. Esquivel, R. Koch, M. Ziegler, F. Zilly, and J. Keinert, "A novel Kinect V2 registration method for large-displacement environments using camera and scene constraints," in *ICIP*, 2017, pp. 997–1001. 1, 4

[22] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *CVPR*, 2017, pp. 199–208. 1, 4

[23] Y. Gao, S. Esquivel, R. Koch, and J. Keinert, "A novel self-calibration method for a stereo-ToF system using a Kinect V2 and two 4K GoPro cameras," in *3DV*, 2017. 2

[24] Y. Gao, M. Ziegler, F. Zilly, S. Esquivel, and R. Koch, "A linear method for recovering the depth of Ultra HD cameras using a Kinect V2 sensor," in *IAPR MVA*, 2017, pp. 494–497. 2

[25] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966. 2

[26] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE TPAMI*, vol. PAMI-9, no. 5, pp. 698–700, 1987. 2

[27] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *ECCV*, 2006, pp. 404–417. 3

[28] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*, 1996, pp. 303–312. 3, 4

[29] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Comm. of the ACM*, vol. 18, no. 9, pp. 509–517, 1975. 3

[30] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. 3

[31] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987. 3

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 3

[33] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971. 3

[34] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IROS*, 2006, pp. 5695–5701. 3

[35] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-flight cameras in computer graphics," *CGF*, vol. 29, no. 1, pp. 141–159, 2010. 4

[36] M. Lindner, I. Schiller, A. Kolb, and R. Koch, "Time-of-flight sensor calibration for accurate range sensing," *CVIU*, vol. 114, no. 12, pp. 1318–1328, 2010. 4

[37] V. Garro, C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, "A novel interpolation scheme for range data with side information," in *CVMP*, 2009, pp. 52–60. 4

[38] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011, pp. 127–136. 4

[39] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3D reconstruction using signed distance functions." in *RSS*, 2013. 4

[40] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *SIGGRAPH*, vol. 21, no. 4, 1987, pp. 163–169. 4