Computers & Graphics 115 (2023) 446–460

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Progressive refinement imaging with depth-assisted disparity correction ☆

Markus Kluge [a,*], Tim Weyrich [b], Andreas Kolb [a]

[a] Computer Graphics Group, University of Siegen, Siegen, Germany
[b] Department of Computer Science, Friedrich-Alexander University (FAU), Erlangen-Nürnberg, Germany

ARTICLE INFO

Article history:
Received 8 February 2023
Received in revised form 16 July 2023
Accepted 18 July 2023
Available online 22 July 2023

Keywords:
RGB-D fusion
Progressive image refinement
3D reconstruction

ABSTRACT

In recent years, the increasing on-board compute power of mobile camera devices gave rise to a class of digitization algorithms that dynamically fuse a stream of camera observations into a progressively updated scene representation. Previous algorithms either obtain general 3D surface representations, often exploiting range maps from a depth camera, such as, Kinect Fusion, etc.; or they reconstruct planar (or distant spherical, respectively) 2D images with respect to a single (perspective or orthographic) reference view, such as, panoramic stitching or aerial mapping. Our work sets out to combine aspects of both, reconstructing a 2.5-D representation (color and depth) as seen from a fixed viewpoint, at spatially variable resolution. Inspired by previous work on "progressive refinement imaging", we propose a hierarchical representation that enables progressive refinement of both colors and depths by ingesting RGB-D images from a handheld depth camera that is carried through the scene. We evaluate our system by comparing it against state-of-the-art methods in 2D progressive refinement and 3D scene reconstruction, using high-detail indoor and outdoor data sets comprising medium to large disparities. As we will show, the restriction to 2.5-D from a fixed viewpoint affords added robustness (particularly against self-localization drift, as well as backprojection errors near silhouettes), increased geometric and photometric fidelity, as well as greatly improved storage efficiency, compared to more general 3D reconstructions. We envision that our representation will enable scene exploration with realistic parallax from within a constrained range of vantage points, including stereo pair generation, visual surface inspection, or scene presentation within a fixed VR viewing volume.

1. Introduction

The past decade has seen an emergence of interactive scene digitization systems that dynamically fuse a stream of sensor observations into a progressively updated scene representation. The key benefit of dynamic ("online") reconstruction over offline methods (where all data is captured first before a reconstruction happens in a post-process) is the ability to interactively capture more data where the current reconstruction indicates insufficient data so far [1].

This principle is now prominently used both for 2D imaging (e.g., panorama mode in mobile phone camera applications) and 3D model reconstruction; the latter was popularized through the introduction of affordable color+depth (RGB-D) cameras and immediately spawned the field of online scene digitization from handheld RGB-D cameras, pioneered by KinectFusion [2,3].

Even though a full 3D reconstruction (geometry and color) has the appeal of capturing more comprehensive aspects of a scene, and despite many modern mobile phones featuring RGB-D sensors, 2D imaging remains the most popular modality in the mainstream. We argue that, besides other reasons, that popularity is mainly due to most *output* devices being 2D, due to the tighter control over the output's appearance, but also due to our ability to take in a 2D scene at a single glance while 3D content requires an interface for navigation and exploration.

In recognition of the enduring importance of 2D scene imaging, recent work adapts the concept of online scene capture to the 2D domain, creating a *variable-resolution* RGB image from unstructured image collections. Kluge et al. [4] introduce interactive, *progressive refinement imaging* to bridge panoramic stitching and handheld "fusion-style" digitization. Similarly, Licorish et al. [5] use adaptive compositing of pre-registered images with variable resolution captured with a single camera with optical zoom. While these methods support the high-quality photometric integration of images across a wide range of object-space resolutions, they are, however, strictly limited either to a perfectly fixed vantage point, or to scenes with minimal depth disparity; in

☆ This article was recommended for publication by Prof. J Zheng.

* Correspondence to: Computer Graphics Group, University of Siegen, Hoelderlinstrasse 3, 57076 Siegen, Germany.
E-mail address: markus.kluge@uni-siegen.de (M. Kluge).

https://doi.org/10.1016/j.cag.2023.07.036
0097-8493/

particular, they are prone to parallax-induced misalignment artifacts whenever the camera is moved within the scene to obtain higher-resolution close-ups of objects.

In this work, we aim at overcoming this restriction by progressively reconstructing an auxiliary depth map alongside an image reconstruction in the spirit of Kluge et al. [4]. This adaptively refined depth map is used to compensate for parallax due to depth disparities and further assists with self-localization of the camera. In a departure from common approaches for scene reconstruction from RGB-D images, however, and more in line with image-based rendering, our method strictly decouples color data from the coarse and potentially incomplete geometry representation. Thus, the inherent difference in data quality between color and depth sensors is accommodated, which greatly increases robustness of the scene capture.

Just like online 3D scene reconstruction approaches, we take handheld RGB-D camera streams as input. Similarly to Kluge et al. [4], our approach hierarchically fuses color differences in a sparse Laplacian pyramid. Their approach naturally achieves texture consistency by blending not colors, but highpass-filtered image color details into that Laplacian hierarchy, assisted by local input alignment correction using optical flow. In order to also aggregate depth values, however, our approach has to overcome several challenges intrinsic to range images that make them harder to fuse than the typically high-quality color channels of RGB-D: (1) significantly increased noise, including outliers and missing data, often correlated with salient features like silhouettes, (2) lower effective resolution, and (3) relative alignment errors with respect to the color imager. To cope with these depth errors and artifacts, our progressive and adaptive *depth* refinement uses an explicit depth model instead of a Laplacian pyramid to prevent noise amplification. Moreover, we trade the standard averaging approach, frequently used in popular online 3D geometry reconstruction approaches, for a progressive per-pixel voting scheme.

The resulting system enables reliable image capture of general scenes, using an RGB-D camera where the operator first takes an overview shot before walking into the scene to take close-ups where added image detail is desired. By bridging between 2D and 3D approaches, our system manages to mitigate limitations of either modality. Parallax-induced errors of 2D imaging approaches are virtually eliminated, and texture inconsistencies, that to date require global post-optimization, yielding non-progressive and non-interactive systems [6,7], are resolved on the fly. Last but not least, by anchoring the reconstruction in the initial overview shot, camera-drift that plagues existing 3D scene reconstruction methods is eliminated.

We evaluate our system by comparing it against state-of-the-art methods in 2D progressive refinement and 3D scene reconstruction, using high-detail indoor and outdoor data sets comprising medium to large disparities. As we will show, the restriction to 2.5-D from a fixed viewpoint affords added robustness (particularly against self-localization drift, as well as backprojection errors near silhouettes), increased geometric and photometric fidelity, even in the presence of illumination changes, as well as greatly improved storage efficiency, compared to more general 3D reconstructions.

In summary, this paper contributes:

- *Disparity-corrected* adaptive image refinement that fuses observations into a high-quality, geometrically consistent, adaptive-resolution 2.5D image, even in the presence of *silhouettes* and strong *scene parallax*, while retaining photometric consistency.
- *Progressive and local* geometric and photometric optimization for *drift-free* color and depth alignment.
- *Decoupled color and depth representation*, using a sparse Laplacian for color and sparse Gaussian for depth, that straddles high color fidelity with artifact-prone depth readings.
- A bespoke *progressive per-pixel depth voting scheme* that outperforms conventional cumulative average weighting.

We envision that, apart from creating high-fidelity, adaptive-resolution 2D content, our depth-enhanced representation will enable scene exploration with realistic parallax from within a constrained range of vantage points, including stereo pair generation, visual surface inspection, or scene presentation within a fixed VR viewing volume.

2. Related work

Our progressive, high-quality, high-resolution RGB-D image reconstruction approach relates to both single-image refinement from photo collections, as well as to high-quality color reproductions for online 3D scene reconstruction methods. We now give a brief overview of the state-of-the-art in both domains.

2.1. Single-image refinement from photo collections

There are many methods for combining several RGB images into a single photo, which commonly require very specific conditions to be met. Relevant categories are panoramic mosaics [8], which expect images obtained by panning about the camera's pivot point, and photo montage approaches for combining a set of photographs into a single composite picture [9]. Conceptually, these methods solve the problem of image registration, i.e., *geometric consistency*, and image recombination, i.e., *photometric consistency*. However, Kluge et al. [4] have shown that applying these kinds of methods to imagery with highly variable object-space resolution and significant geometric and photometric distortions leads to failure, mainly due to (1) unsuccessful matching of the input frames, (2) unsuccessful image refinement, or (3) enforcing a panoramic mosaicing scenario with constant resolution.

Feature-based panorama stitching approaches for unordered data sets using, for instance, SIFT feature matching and multi-band blending [10] can solve the challenging data characteristics as demonstrated in methods like AutoStitch [11] and AutoPano [12] that utilize this approach. Photo Zoom [13] automatically constructs a high-resolution image from an unordered set of zoomed-in photos. This approach applies homographies for image registration and requires global post-processing comprising a recursive gradient domain fusion approach to tackle color inconsistencies.

Progressive refinement imaging [4] tackles the problem in a non-global fashion that allows processing large sequences of several 100 images. Photometric consistency is achieved using a sparsely occupied Laplacian pyramid in combination with an image fusion approach that retains the base color defined by an initial overview reference image. The incoming close-up images are aligned using a two-stage approach consisting of a coarse, feature-based registration and a local refinement using optical flow. Recent work [5] addresses adaptive compositing of different-resolution images by computing variable-resolution seams. The method assumes pre-registered images at different resolutions captured with a single camera with optical zoom and within a short period of time. As image refinement approaches for 2D RGB images intrinsically assume an almost planar (or infinitely far away) scenery, they are restricted in handling the disparity in non-planar scenes in closer vicinity to the camera. We will evaluate this limitation in Section 6.

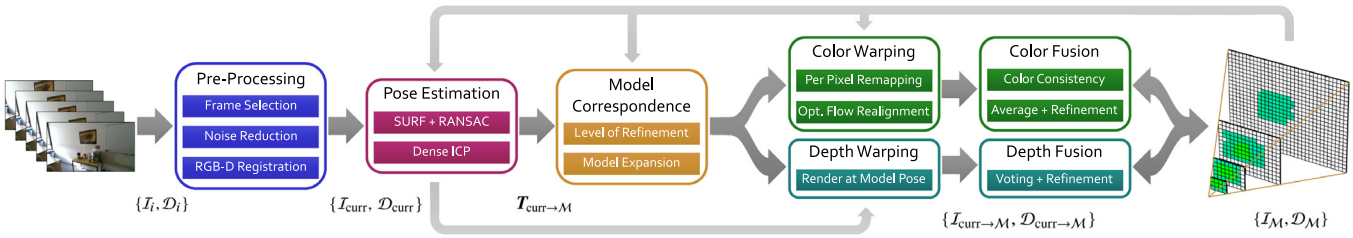

Fig. 1. Our proposed progressive refinement imaging pipeline for 3D scenes.

2.2. Photometrically optimized 3D scene reconstruction

Our objective of adaptive and progressive image refinement with disparity correction is inherently linked to 3D scene reconstruction methods that prioritize high-quality photometric optimization using RGB-D image sequences. These methods implicitly handle disparity by fusing depth information into a full 3D model.

Regarding scene geometry, high-quality photometric reconstruction is commonly achieved via post-optimization applied to a pre-reconstructed scene geometry using Structure-from-Motion [14,15] or KinectFusion [2,3]-like methods resulting in Truncated Signed Distance Function (TSDF) volumes [16], which are potentially converted into meshes [6,7,16–25]. Some of these methods further coarsen the mesh [21,22] or even extract semantics, i.e., planar or Manhattan-like representations [20,25]. Several methods modify the initially captured scene geometry to improve the overall 3D geometric and photometric consistency [16,19].

Commonly, there is a significant amount of photometric inconsistencies in a 3D-reconstructed scene, mainly due to sensor noise and inaccurate camera pose estimates. Photometric consistency is commonly achieved using pose refinement for keyframes [6, 21], potentially segmenting the model and applying intensity and gain correction or synthesizing textures from the RGB imagery [7,17,18,20,24,25], or using super-resolution approaches projecting individual observations into the keyframes [14,15,23]. Alternatively, the photometric information can be accumulated in a voxel grid with a higher resolution than the one used for fusing the geometric information [22]. Other methods aiming at high-quality photometric reconstruction use joint optimization for the camera poses, the scene's geometry and texture [19,25], or intrinsic material properties [16,25].

While all approaches mentioned above are not interactive or real-time capable, some methods reduce the computational complexity to achieve interactive framerates. Meilland and Comport [26] propose a 2.5D scene representation. They fuse low-resolution RGB-D image sequences into a single super-resolution 2560×1920 px RGB-D map applying a fixed super-resolution factor (4 in this case) and deblur the result in a post-processing step. Lee et al.'s TextureFusion approach [27] generates a full 3D model representing higher-resolved texture information using an axis-aligned parallel projection onto the implicit surface within individual TSDF voxels containing the iso-surface. This allows for real-time geometry reconstruction and texture fusion using standard weighted blending methods. Their follow-up work [28] allows for the real-time acquisition of photometric normals jointly represented with texture information.

2.3. NeRF and other learning-based approaches

Recently, *Neural Radiance Field (NeRF)* approaches have gained much attention, which generally learn an implicit latent representation of a radiance field captured at known camera poses [29]. There have been several attempts to enhance NeRF-like approaches towards the interactive processing of real-world RGB or RGB-D data. For example, the NeRF in the wild method [30] addresses photometric variations and transient objects in an unstructured photo collection with known camera poses, while the GNeRF approach [31] learns the camera pose parameters utilizing Generative Adversarial Networks (GANs) for this task. Moreover, neural implicit representations have been enhanced towards interactive RGB-D scene reconstruction [32,33]. The recent NICE-SLAM approach [33] achieves interactive frame rates of ∼5 fps. Still, compared to classical 3D scene reconstruction methods, the reconstruction quality of methods utilizing implicit neural representations is significantly lower than for classical approaches (see, e.g., the camera pose comparison in [33, Tab. 2]).

In summary, none of the existing methods can handle high-quality photometric and geometric RGB-D image refinement in an interactive progressive fashion. Most specifically, existing RGB-D approaches do not take advantage of adaptive fusion to achieve local photometric and geometric refinement. Conceptually, our approach has been inspired by the 2.5D scene representation from Meilland and Comport [26] to handle disparity properly, and by the efficient, spatially adaptive image refinement from Kluge et al. [4] that uses Laplacian pyramid-based image fusion for color consistency.

3. Method overview

Our proposed progressive RGB-D image refinement pipeline is depicted in Fig. 1. The input to our pipeline is a stream of RGB-D images $\{\mathcal{I}_i, \mathcal{D}_i\}$ comprising color and depth images for frame indices i. We expect the initial frame $\{\mathcal{I}_0, \mathcal{D}_0\}$ to be a reference frame that covers the region and viewing direction of interest of the observed scene for all following frames $\{\mathcal{I}_i, \mathcal{D}_i\}$, $i > 0$. Unlike the usual 360° lateral scan in scene reconstruction, "progressive refinement imaging" deliberately aims at a "walking closer to the scene"-like camera path. The overall assumption here is that by approaching the scene, subsequent frames provide novel geometric and photometric details of the scene. Taking the reference frame as initial model $\mathcal{M}$, our approach progressively refines this model by fusing the RGB-D stream into $\mathcal{M}$, yielding a geometric and photometric consistent RGB-D image with locally refined resolution. The model $\mathcal{M}$ comprises a Laplacian color pyramid $(\mathcal{I}_{\mathcal{M}})$ and a depth image $(\mathcal{D}_{\mathcal{M}})$ with locally adapted resolution (see Section 4.1 for a detailed motivation). The main components of our pipeline are as follows (see Table 1 for a list of conventions used).

Pre-Processing: As the main objective is to improve the photometric quality of the final image, we apply a *frame selection* to identify the frame $\{\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}\}$ with the sharpest color image within a small set of the latest consecutive input frames. Moreover, we perform *noise reduction* on the depth image to discard erroneous, e.g., flying pixels. Finally, we *register* the color and the depth image by generating a high-resolution RGB-D image. See Section 4.2 for further details.

Table 1
List of conventions.

$\mathcal{I}_i$, $\mathcal{D}_i$	ith input color and depth frame
$\mathcal{I}_{\text{curr}}$, $\mathcal{D}_{\text{curr}}$	Selected input frame of current iteration (observation)
$\mathcal{M}$	Model comprising components $\mathcal{I}_\mathcal{M}$, $\mathcal{D}_\mathcal{M}$
$\mathcal{I}_\mathcal{M}$, $\mathcal{D}_\mathcal{M}$	Pyramidal representations of accumulated color and depth, consisting of pyramid levels $\mathcal{I}^l_\mathcal{M}$, $\mathcal{D}^l_\mathcal{M}$ with level indices l
$c_\mathcal{M}$	Counter of observations fused into $\mathcal{I}_\mathcal{M}$ (per-pixel attribute)
$v_\mathcal{M}$	Voting counter of $\mathcal{D}_\mathcal{M}$ (per-pixel attribute)
$\boldsymbol{T}_{\text{curr}\to\mathcal{M}}$	Rigid camera transformation from observation to $\mathcal{M}$
$\mathcal{W}_{\mathcal{M}\to\text{curr}}$	Image-space mapping between $\mathcal{M}$ and the observation
$\mathcal{I}_{\text{curr}\to\mathcal{M}}$, $\mathcal{D}_{\text{curr}\to\mathcal{M}}$	$\mathcal{I}_{\text{curr}}$ and $\mathcal{D}_{\text{curr}}$ warped to $\mathcal{M}$'s image space
$\mathcal{I}^l_{\text{curr}\to\mathcal{M}}$	$\mathcal{I}_{\text{curr}\to\mathcal{M}}$ decomposed into Laplacian levels with indices l
$\mathcal{L}_{\text{curr}}$, $\mathcal{L}_\mathcal{M}$	Level-of-refinement of $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ and $\mathcal{I}_\mathcal{M}$ (per-pixel attribute)
$l_{\min}$	Corresp. level index of warped obs. within pyramid $\mathcal{M}$
$roi(\dots)$	Lateral boundaries of warped obs. on $\mathcal{M}$ (region of interest)
$\mathcal{F}_{\mathcal{M}\to\text{curr}}$	Flow field between $\mathcal{I}_\mathcal{M}$ and $\mathcal{I}_{\text{curr}\to\mathcal{M}}$
s^l_{curr}	Similarity score of $\mathcal{I}^l_{\text{curr}\to\mathcal{M}}$ (per-pixel attribute)
$\boldsymbol{K}_\mathcal{I}$, $\boldsymbol{K}_\mathcal{D}$	Intrinsic camera matrices of color and depth imager
$\mathcal{K}_{\text{prev}}$, $\mathcal{K}_{\text{curr}}$	2D keypoints of prev. and curr. iteration
$\mathcal{P}_{\text{prev}}$, $\mathcal{P}_{\text{curr}}$	3D keypoints of prev. and curr. iteration
$\mathcal{V}_{\text{curr}}$, $\mathcal{V}_\mathcal{M}$	Vertex maps of $\mathcal{D}_{\text{curr}}$ and $\mathcal{D}_\mathcal{M}$

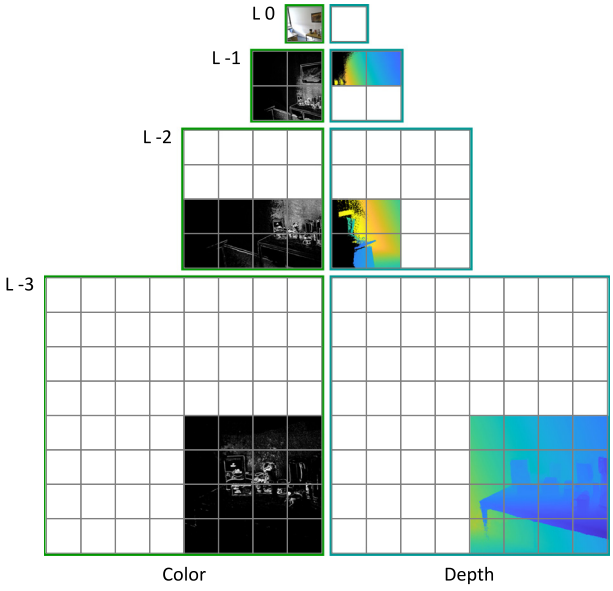

Fig. 2. An example layout of the model representation $\mathcal{M}$. Each pyramid level is regularly tiled with a fixed size. A tile is occupied by image data if refined data has been acquired; otherwise, it is unallocated. Color data is stored as sparsely occupied Laplacian pyramid in corresponding tiles across multiple levels, whereas depth data is stored as-is, within tiles that occupy the finest level of the respective depth observations.

Pose Estimation: The current camera pose, represented by the rigid transformation $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}$ between the currently selected frame $\{\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}\}$ and the model $\mathcal{M}$, is estimated in a two-stage process using sparse feature matching (*SURF*) and a subsequent *dense ICP* (see Section 4.3).

Model Correspondence: Dependent on the current frame's pose, we estimate the observation's potential to refine the model by determining a per-pixel *level-of-refinement* map. This may trigger an *expansion* of the model $\mathcal{M}$ by extending the color pyramid and the adaptive depth representation appropriately (see Fig. 2). For more details, see Section 4.4.

Color & Depth Warping: Due to their different nature, noise level, and purpose of the color and depth information, at this stage, both modalities are processed separately by splitting the reconstruction pipeline into two parallel strands (see Fig. 1). Our color warping is a per-pixel *remapping* using the estimated camera pose $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}$ and model depths $\mathcal{D}_\mathcal{M}$ to correct for parallaxes in the current color observation $\mathcal{I}_{\text{curr}}$. An *optical flow* is then applied for local re-alignment, resulting in $\mathcal{I}_{\text{curr}\to\mathcal{M}}$. The depth information, however, is warped via *rendering* the meshed depth map $\mathcal{D}_{\text{curr}}$ from the model's camera pose, yielding the warped depth map $\mathcal{D}_{\text{curr}\to\mathcal{M}}$ (see Section 4.5).

Color & Depth Fusion: The color fusion is based on a cumulative *averaging* scheme, *refining* the initial reference image by adding details in a frequency-oriented way. A *color consistency* check ensures the exclusion of inconsistent details. Depth fusion is performed using a combination of blending and replacement based on a progressive *voting* scheme, as the initial model depths can be very erroneous. For further details, see Sections 4.6 and 4.7.

4. Progressive refinement imaging for 3D scenes

Our depth-assisted progressive refinement imaging approach for 3D scenes is based on data obtained by a commodity, hand-held RGB-D camera such as Kinect v1, Xtion, or Kinect v2, that provides the RGB-D stream $\{\mathcal{I}_i, \mathcal{D}_i\}$ with color images $\mathcal{I}_i \in \mathbb{R}^3$ with RGB intensities and depth maps $\mathcal{D}_i \in \mathbb{R}$ with camera-to-surface-distances in meters.

4.1. Model (initialization)

We expect any capture to begin with an overview shot that defines the reference frame of the variable-resolution output image. Thus, the first frame, $i = 0$, sets a fixed reference viewpoint of the scene and initializes *model* $\mathcal{M}$, our representation for reconstructed RGB-D data. $\mathcal{M}$ consists of two components, $\mathcal{I}_\mathcal{M}$ and $\mathcal{D}_\mathcal{M}$, which represent the variable-resolution color image and depth map, respectively.

Model color component $\mathcal{I}_\mathcal{M}$ is a multi-scale representation based on [4], a sparsely occupied and dynamically expandable Laplacian pyramid [34], consisting of pyramid levels $\mathcal{I}^l_\mathcal{M}$, where the level index $l \in \mathbb{Z}$ decreases with finer resolution (i.e., receive negative indices). $\mathcal{I}_\mathcal{M}$ is initialized by the input color image $\mathcal{I}_0$, which serves as a reference for maintaining color consistency. Over time, new, finer Laplacian levels $\mathcal{I}^{l<0}_\mathcal{M}$ are appended to the bottom of the pyramid, refining the initial reference image as novel details are added from subsequent frames $\mathcal{I}_{i>0}$. As not all image regions are captured at the same level of object-space resolution when approaching the scene in a free-form camera path, $\mathcal{I}_\mathcal{M}$ is sparsely occupied. Therefore, each pyramid level $\mathcal{I}^l_\mathcal{M}$ is regularly tiled, where a tile (1024×1024 px) is allocated only if refined data was acquired. Moreover, each pixel has the following attributes: a counter $c_\mathcal{M} \in \mathbb{N}$, representing the number of fused observations (initialized with 1), and the model's level of refinement $\mathcal{L}_\mathcal{M} \in \mathbb{R}$, the so-far accumulated amount of detail (initialized with 0).

In addition, we introduce the model component $\mathcal{D}_\mathcal{M}$, an adaptively subdivided depth map representation. In contrast to color, the accumulated depth is not decomposed into band-pass filtered Laplacian levels but is stored as-is: we found that the difference operators produce artifacts in the range data due to amplifying noise, leading to erroneous model depths when merging frequencies of different observations. $\mathcal{D}_\mathcal{M}$ can be interpreted as a sparsely occupied Gaussian pyramid that shares the pyramidal structure of $\mathcal{I}_\mathcal{M}$ but has tiles allocated only at the finest level (see Fig. 2). The first input depth map $\mathcal{D}_0$ initializes $\mathcal{D}_\mathcal{M}$, and additionally, a voting counter $v_\mathcal{M} \in \mathbb{R}$ is stored as a per-pixel attribute, representing a depth's reliability (initialized with 1).

4.2. Input

The input to our reconstruction pipeline is a continuous stream of color images $\mathcal{I}_{i>0}$ and depth maps $\mathcal{D}_{i>0}$ that progressively refine the model color $\mathcal{I}_\mathcal{M}$ and model depth $\mathcal{D}_\mathcal{M}$.

Frame selection. To avoid merging highly redundant data and to reduce processing time, we select the sharpest of 15 subsequent frames for further processing if a maximum blur threshold $\varepsilon_\text{b} = 0.32$ is not exceeded. We follow [6] and use the blur metric from [35], applied to the color image $\mathcal{I}_i$. The selected frame of the current iteration $\{\mathcal{I}_\text{curr}, \mathcal{D}_\text{curr}\} = \{\mathcal{I}_i, \mathcal{D}_i\}$, the *current observation*, is then passed to the following pipeline stages.

Pre-processing. We first remove outliers from the depth map $\mathcal{D}_\text{curr}$ by discarding pixels incompatible with their local neighborhood (*flying pixels*). A pixel $\mathcal{D}_\text{curr}(x, y)$ is considered an inlier (i.e., not an outlier) if at least one pixel in its 4-neighborhood differs in depth by less than the tolerance $\varepsilon_\text{f} = 0.1$ m.

Subsequent bilateral filtering [36] of $\mathcal{D}_\text{curr}$ mitigates noise, smoothing homogeneous regions while preserving depth discontinuities. As parameterization, we use $\sigma_\text{s} = 2.5$ for the spatial Gaussian kernel and $\sigma_\text{r} = 0.03$ for the range kernel. For noisy outdoor scenery, we increase σ_r to 0.15.

RGB-D registration. If $\mathcal{I}_\text{curr}$ and $\mathcal{D}_\text{curr}$ are not pre-registered, we register both modalities using the extrinsic transformation $\boldsymbol{T}_{\mathcal{D}\to\mathcal{I}} = [\boldsymbol{R}_{\mathcal{D}\to\mathcal{I}}, \boldsymbol{t}_{\mathcal{D}\to\mathcal{I}}] \in \mathbb{SE}^3$ between both camera coordinate systems, with 3D rotation matrix $\boldsymbol{R}_{\mathcal{D}\to\mathcal{I}} \in \mathbb{SO}^3$ and translation vector $\boldsymbol{t}_{\mathcal{D}\to\mathcal{I}} \in \mathbb{R}^3$. As we prioritize high color resolution, we break with the 3D reconstruction tradition of transforming color images into the viewpoint of the depth camera and, instead, project depth $\mathcal{D}_\text{curr}$ onto the color camera's image plane. While the former only requires a simple backward remapping operation on $\mathcal{I}_\text{curr}$ for each pixel position $(x, y)^\top$ of $\mathcal{D}_\text{curr}$ using its depth value $\mathcal{D}_\text{curr}(x, y)$, the latter is more complex: we first triangulate $\mathcal{D}_\text{curr}$ (see Section 4.5 for details) and then render the resulting triangle mesh from the position and orientation of the color camera using $\boldsymbol{T}_{\mathcal{D}\to\mathcal{I}}$ and its intrinsic parameters, i.e., the principal point $(c_x^\mathcal{I}, c_y^\mathcal{I})^\top$ and the focal lengths $f_x^\mathcal{I}, f_y^\mathcal{I}$.

4.3. Camera pose estimation

To globally align the observation with the model $\mathcal{M}$, the current 6-DoF rigid camera transformation $\boldsymbol{T}_{\text{curr}\to\mathcal{M}} = [\boldsymbol{R}, \boldsymbol{t}] \in \mathbb{SE}^3$, with $\boldsymbol{T}_{\text{curr}\leftarrow\mathcal{M}} = \boldsymbol{T}_{\text{curr}\to\mathcal{M}}^{-1}$, needs to be estimated.

For this, 3D scene reconstruction approaches usually perform frame-to-model tracking by concatenating a chain of relative poses over all consecutive frames, which suffers from accumulating a temporal pose drift. This drift is the consequence of aligning the current frame with a proxy of the model, a rendering from the previous, already drift-affected pose. In our case, we benefit from the fact that refinement takes place in the reference pose, and we always align the current frame with the model itself. While we also use the previous pose as a prediction, it only serves as an initialization. This makes our system robust against self-localization drift, and we do not depend on loop closures to detect and correct error accumulation in a chain of relative poses.

Our pose estimation is based on a two-step, coarse-to-fine approach. First, we align the current frame with the "current" one of the previous pipeline run by searching for and matching sparse correspondences using scale-invariant, *speeded-up robust features* (SURF) [37]. A dense *iterative-closest-point* (ICP) algorithm [38,39] is then initialized with the resulting pre-alignment, estimating a final, fine-scale alignment $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}$ between the current frame and the model $\mathcal{M}$.

Pre-alignment using sparse keypoints. As we expect potentially large displacements between the current and the reference pose (see Section 4.2), we estimate a coarse pre-alignment using sparse photometric correspondences. First, a set of photometric SURF features with 2D keypoint locations $\mathcal{K}_\text{curr} \in \mathbb{R}^2$ are detected in the current color image $\mathcal{I}_\text{curr}$, using the Hessian feature threshold $\varepsilon_\text{h} = 1000$ and four SURF octaves with four scales in each octave. These features are then matched against the feature descriptors of keypoints $\mathcal{K}_\text{prev} \in \mathbb{R}^2$ of the previously selected "current" frame processed by our pipeline using RANSAC [40].

The keypoint sets $\mathcal{K}_\text{curr}$ and $\mathcal{K}_\text{prev}$ are pruned by filtering potential mismatches and error-prone keypoints. We pre-filter keypoint matches by applying Lowe's ratio test [41]. A keypoint is tested for its integrity by comparing its two best matches using their distance ratio. If both matches are similarly rated, the keypoint is discarded, with the intuition that a correct match is unique. We use a ratio threshold $\varepsilon_\text{r} = 0.675$.

Additionally, we filter keypoints $\mathcal{K}_\text{curr}$ in the vicinity of unreliable depths. While 2D keypoint locations are based on high-resolution color imagery, their 3D locations rely on coarse depth maps, which is highly prone to error at inaccurate depth discontinuities and surfaces with a flat angle to the camera. Therefore, we compute a binary mask $\mathcal{G} \in \mathbb{Z}_2$ of inhomogeneous areas: first, we generate morphologically eroded and dilated depth map versions $\mathcal{D}_\text{curr}^\text{min}$ and $\mathcal{D}_\text{curr}^\text{max}$, respectively, using a 5×5 box-shaped structuring element. Then, we exclude pixels by thresholding $\mathcal{D}_\text{curr}^\text{diff} = \mathcal{D}_\text{curr}^\text{max} - \mathcal{D}_\text{curr}^\text{min}$ with differences that exceed $\varepsilon_\text{d} = 0.03$ m.

Finally, we back-project 2D keypoint locations $\mathcal{K}_\text{curr}$ using their corresponding depths in $\mathcal{D}_\text{curr}$ and the input camera's intrinsic matrix $\boldsymbol{K}_\text{curr} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$ to get the 3D point set

$$\mathcal{P}_\text{curr} = \mathcal{D}_\text{curr}(\mathcal{K}_\text{curr})\, \boldsymbol{K}_\text{curr}^{-1} (\mathcal{K}_\text{curr}, 1)^\top \in \mathbb{R}^3 . \tag{1}$$

Knowing the correspondences between $\mathcal{P}_\text{curr}$ and $\mathcal{P}_\text{prev}$ by the feature matching process, we are able to compute a rigid transformation $\boldsymbol{T}_{\text{curr}\to\text{prev}}$ by minimizing the MMSE [42]. This results in the coarse pre-alignment $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}^\text{pre} = \boldsymbol{T}_{\text{curr}\to\text{prev}} \circ \boldsymbol{T}_{\text{prev}\to\mathcal{M}}$, using the previous pose estimation.

Final alignment using dense correspondences. For the final transformation $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}$, we directly align the current frame with the model $\mathcal{M}$ itself on a dense, fine-scale basis using the pre-alignment $\boldsymbol{T}_{\text{curr}\to\mathcal{M}}^\text{pre}$ as initialization. This is done by performing a dense Colored ICP [43], which we summarize in the following. Colored ICP optimizes for photometric consistency in addition to geometric consistency, which is formulated as the joint objective

$$E_\text{hybrid} = (1 - \sigma_\text{ICP}) E_\mathcal{I} + \sigma_\text{ICP} E_\mathcal{D} , \tag{2}$$

with $E_\mathcal{I}$ and $E_\mathcal{D}$ being the photometric and geometric least-squares objectives. We follow Park et al. [43] and set $\sigma_\text{ICP} = 0.968$. $E_\mathcal{D}$ is formulated as the traditional point-to-plane error metric,

$$E_\mathcal{D}(\boldsymbol{T}_{\text{curr}\to\mathcal{M}}) = \sum_{(\boldsymbol{p},\boldsymbol{q})\in\mathcal{R}} \left\langle \boldsymbol{T}_{\text{curr}\to\mathcal{M}}\, \mathcal{V}_\text{curr}(\boldsymbol{q}) - \mathcal{V}_\mathcal{M}(\boldsymbol{p}) , \mathcal{N}_\mathcal{M}(\boldsymbol{p}) \right\rangle^2 , \tag{3}$$

between the current input depth map $\mathcal{D}_\text{curr}$ and model depth $\mathcal{D}_\mathcal{M}$, back-projected to camera space, i.e., $\mathcal{V}_\text{curr}$ and $\mathcal{V}_\mathcal{M}$ (see Section 4.5). The model's normal map $\mathcal{N}_\mathcal{M}$ is determined from central-differences of $\mathcal{V}_\mathcal{M}$.

The photometric objective $E_\mathcal{I}$ is expressed as the squared differences of intensities

$$E_\mathcal{I}(\boldsymbol{T}_{\text{curr}\to\mathcal{M}}) = \sum_{(\boldsymbol{p},\boldsymbol{q})\in\mathcal{R}} \left(\mathcal{I}_\text{curr}(\boldsymbol{q}) - \mathcal{I}_\mathcal{M}^\text{comp}(\boldsymbol{p}) \right)^2 , \tag{4}$$

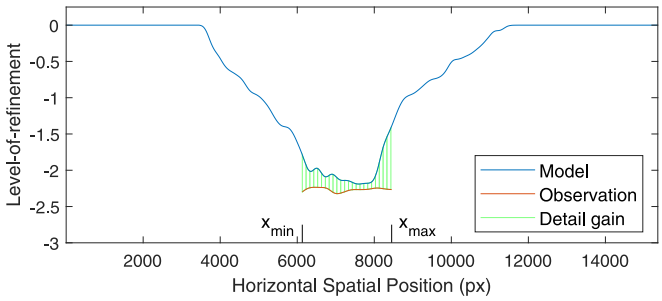

Fig. 3. A one-dimensional graphic representation of the level-of-refinement maps $\mathcal{L}_{\mathcal{M}}$ and $\mathcal{L}_{\text{curr}}$. The model's accumulated level of refinement (blue) is shown after the camera has been moved centrally towards the scene, with details accumulated up to level -2.2. The current observation (red) offers a higher object-space resolution (lower corresp. level), where the per-pixel gain in visual detail is colored in green (see Δ_{curr} in Eq. (12)). Its lateral boundaries within the model are x_{min} and x_{max} (region of interest), the minimum pyramid level is $l_{\text{min}} = -3$.

between the current input color image $\mathcal{I}_{\text{curr}}$ and $\mathcal{I}^{\text{comp}}_{\mathcal{M}}$, which is the re-composed model color image from the Laplacian pyramid $\mathcal{I}_{\mathcal{M}}$.

The dense correspondence set $\mathcal{R} = \{(\boldsymbol{p}, \boldsymbol{q})\}$ is determined via projective data association, that is, projecting each pixel in $\mathcal{D}_{\text{curr}}$ with location $\boldsymbol{q} \in \mathbb{N}^2$ onto $\mathcal{D}_{\mathcal{M}}$, getting the corresponding pixel location

$$\boldsymbol{p} = \boldsymbol{\pi}\left(\boldsymbol{K}_{\mathcal{M}}\, \boldsymbol{T}_{\text{curr}\to\mathcal{M}}\, \mathcal{D}_{\text{curr}}(\boldsymbol{q})\, \boldsymbol{K}^{-1}_{\text{curr}}\,(\boldsymbol{q}, 1)^{\top}\right) \in \mathbb{R}^2\,, \tag{5}$$

with $\boldsymbol{K}_{\text{curr}}$ and $\boldsymbol{K}_{\mathcal{M}}$ being the camera's intrinsic matrices of the current frame and the model, and $\boldsymbol{\pi}(x, y, z) = (x/z, y/z)^{\top}$, the de-homogenization. We use the Euclidean distance threshold $\varepsilon_{\text{dist}}$ and angle threshold $\varepsilon_{\text{angle}} = 45°$ as compatibility criteria to prune potential correspondences. We set $\varepsilon_{\text{dist}} = \{0.1\text{ m}, 0.065\text{ m}, 0.03\text{ m}\}$ for a three-level coarse-to-fine ICP and we soften the criterion for noisy outdoor footage to $\varepsilon_{\text{dist}} = \{0.3\text{ m}, 0.165\text{ m}, 0.03\text{ m}\}$.

4.4. Model correspondence

The correspondence between the observation and the model refers to the *region of interest* in the model $\mathcal{M}$ affected by the current frame, and the observation's *level of refinement*, representing the observation's potential to refine $\mathcal{M}$. Fig. 3 illustrates these properties with an example.

The current region of interest $roi = (x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}})$, i.e., the observation's lateral boundaries within model $\mathcal{M}$, is calculated by the forward projection of $\mathcal{D}_{\text{curr}}$ onto $\mathcal{D}_{\mathcal{M}}$ using Eq. (5).

The observation's level of refinement refers to the spatial sampling rate that is inverse-proportional to distance, i.e., the sampling rate increases the closer the camera is moved to the scene compared to the reference viewpoint. Thus, we determine the level-of-refinement map $\mathcal{L}_{\text{curr}} \in \mathbb{R}$ as the corresponding pyramid level in $\mathcal{M}$ per pixel. By back-projecting and transforming each model depth of $roi(\mathcal{D}_{\mathcal{M}})$ into the camera space of the current observation, we get its distance to the current frame's camera plane by extracting its z-component. The scale factor between both depths is then mapped to a pyramid level index, where the sampling rate for each level increases by one octave. The (fractional) number of octaves between both distances is given by

$$\mathcal{L}_{\text{curr}}(x, y) = \log_2 \frac{\left(\boldsymbol{T}_{\text{curr}\leftarrow\mathcal{M}}\, \mathcal{D}_{\mathcal{M}}(x, y)\, \boldsymbol{K}^{-1}_{\mathcal{M}}\,(x, y, 1)^{\top}\right)_z}{\mathcal{D}_{\mathcal{M}}(x, y)} \in \mathbb{R}\,, \tag{6}$$

where $(\cdot)_z$ is the z-component of a 3D point. We use the accumulated model depths for this estimate, as they are more accurate, complete, and reliable than observation depths. Here, a gain in level of refinement, i.e., $\mathcal{L}_{\text{curr}}(x, y) \leq \mathcal{L}_{\mathcal{M}}(x, y)$, indicates the observation's ability to contribute superior information for refining the model by updating its data in the fusion stage (Section 4.7).

We further determine the overall minimum pyramid level index $l_{\text{min}} = \lfloor \min(\mathcal{L}_{\text{curr}}) \rfloor \in \mathbb{Z}$. If this level is beyond the current level boundaries of $\mathcal{M}$, we expand the model as follows: a new level of unallocated tiles is appended to the bottom of Laplacian pyramid $\mathcal{I}_{\mathcal{M}}$. For the sparsely occupied Gaussian pyramid $\mathcal{D}_{\mathcal{M}}$, all tiles affected by the region of interest are up-sampled to l_{min}, using nearest-neighbor interpolation to avoid introducing flying pixels. The model's counters $c_{\mathcal{M}}$, $v_{\mathcal{M}}$ and the accumulated level-of-refinement $\mathcal{L}_{\mathcal{M}}$ inherit their values from coarser levels on demand, as needed during fusion.

4.5. Parallax-aware warping

Color warping. To allow a fusion with the model, a perspective warping of the color image $\mathcal{I}_{\text{curr}}$ into the model's image space is performed. In contrast to [4], which estimates a homography by assuming a (quasi) planar scenery, we, instead, have to rely on depth values for a disparity-corrective mapping between both image spaces. Therefore, we calculate the pixel mapping $\mathcal{W}_{\mathcal{M}\to\text{curr}} \in \mathbb{R}^2$ that relates model to observation locations

$$\mathcal{W}_{\mathcal{M}\to\text{curr}}(x, y) = \boldsymbol{\pi}\left(\boldsymbol{K}_{\text{curr}}\, \boldsymbol{T}_{\text{curr}\leftarrow\mathcal{M}}\, \mathcal{D}_{\mathcal{M}}(x, y)\, \boldsymbol{K}^{-1}_{\mathcal{M}}\,(x, y, 1)^{\top}\right)\,, \tag{7}$$

with $(x, y) \in [x_{\text{min}}, \ldots, x_{\text{max}}] \times [y_{\text{min}}, \ldots, y_{\text{max}}]$. That is, each regular lattice grid position within $roi(\mathcal{M})$ is mapped to an irregular sub-pixel coordinate in the current frame using refined model depths $\mathcal{D}_{\mathcal{M}}$ and camera transformation $\boldsymbol{T}_{\text{curr}\leftarrow\mathcal{M}}$.

The color image $\mathcal{I}_{\text{curr}}$ is then warped to $\mathcal{M}$ using a backward remapping $\mathcal{I}_{\text{curr}\to\mathcal{M}}(x, y) = \mathcal{I}_{\text{curr}}\big(\mathcal{W}_{\mathcal{M}\to\text{curr}}(x, y)\big)$, i.e., a resampling of $\mathcal{I}_{\text{curr}}$ at sub-pixel positions $\mathcal{W}_{\mathcal{M}\to\text{curr}}$ using bi-linear interpolation. As we will fuse color using Laplacian pyramids (see Section 4.7), we warp $\mathcal{I}_{\text{curr}}$ to the finest corresponding model level $\mathcal{M}^{l=l_{\text{min}}}$ at level index l_{min}.

Finally, by subtly smoothing $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ at depth discontinuities, we obtain a natural color transition between foreground and background objects instead of a binary one. For that, a Gaussian kernel with radius $r_{\text{G}} = 2\text{px}$ is used.

Note that warping the 2D image $\mathcal{I}_{\text{curr}}$ inevitably leads to inconsistencies with the model in occluded regions, which are addressed in the outlier removal stage (Section 4.6).

Depth warping. Changing the perspective of a 2.5D depth map requires retrieving the underlying 3D geometry represented by the discretized range values. We therefore convert $\mathcal{D}_{\text{curr}}$ to a polygon mesh by computing a vertex map $\mathcal{V}_{\text{curr}}(x, y) = \mathcal{D}_{\text{curr}}(x, y)\, \boldsymbol{K}^{-1}_{\text{curr}}(x, y, 1)^{\top}$, and then, neighboring vertices $\mathcal{V}_{\text{curr}}(x, y)^{\top}$, $\mathcal{V}_{\text{curr}}(x+1, y)^{\top}$, $\mathcal{V}_{\text{curr}}(x, y+1)^{\top}$ and $\mathcal{V}_{\text{curr}}(x+1, y+1)^{\top}$ are triangulated by choosing the diagonal with the shorter length. We omit triangles with edges longer than $\varepsilon_{\text{d}} = 0.03$ m to open the mesh at discontinuities. Finally, we obtain the warped depth map $\mathcal{D}_{\text{curr}\to\mathcal{M}}$ by rendering the mesh as seen from the model's camera, by setting the view matrix to $\boldsymbol{T}_{\text{curr}\leftarrow\mathcal{M}}$ and the viewport to roi, with the resolution of level l_{min}.

4.6. Local color consistency

After aiming for global consistency in the camera alignment stage (Section 4.3), we can now seek local consistency as the warped observation and the model share the same image space. This is done by matching the warped input frame $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ to the reference $\mathcal{M}$ on a per-pixel basis, using a two-step approach: first, $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ is re-aligned locally by estimating a per-pixel displacement w.r.t. $\mathcal{M}$. Second, pixels that are still inconsistent with the model are classified as outliers.

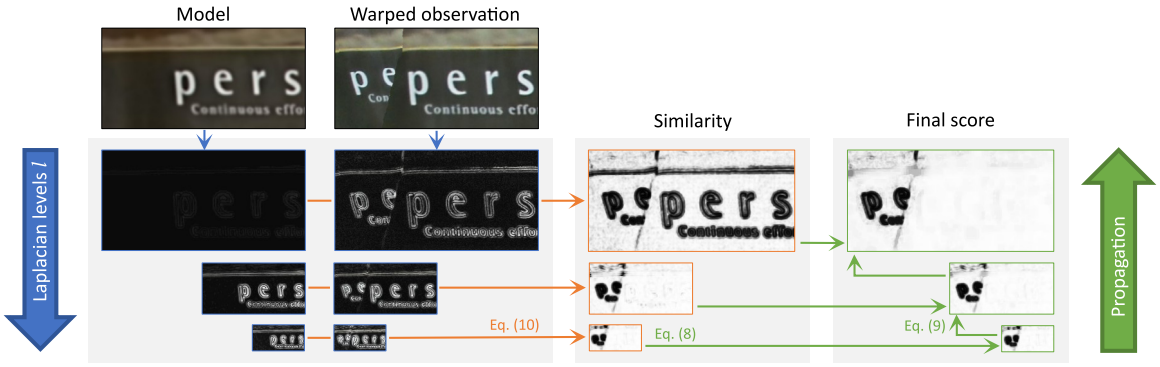

Fig. 4. Our proposed outlier removal scheme. Between the model $\mathcal{I}^l_{\mathcal{M}}$ and the warped observation $\mathcal{I}^l_{\text{curr}\to\mathcal{M}}$, the *similarity* is determined for each Laplacian level l ($SSIM^{\text{C,S}}$ in Eq. (10)). The information is then propagated upwards to compute the *final similarity score* (s^l_{curr} in Eq. (9)). Novel details not yet in the model are in a mismatch on the finest level but are correctly classified as inliers.

Local re-alignment. We follow [4] and compute a dense *Optical Flow* between grayscale variants of $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ and $\mathcal{I}^{\text{comp}}_{\mathcal{M}}$ using [44]. The resulting flow field $\mathcal{F}_{\mathcal{M}\to\text{curr}}$ provides the sub-pixel lateral motion to reduce local misalignments.

To account for various input scales, we use an adaptive number of scale levels for the optical flow algorithm, i.e., we use $-l_{\text{min}}$, the number of pyramid levels between model level $l = 0$ and l_{min}. We then re-align $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ w.r.t. $\mathcal{M}$ by applying the backward flow of $\mathcal{F}_{\mathcal{M}\to\text{curr}}$.

Local outlier removal. To avoid merging inconsistent color data, the warped color frame is searched pixel-wise for geometric discrepancies to detect mismatches that could not be re-aligned or regions that cannot be incorporated, e.g., due to occlusion. Inspired by [4], we detect outliers on band-pass filtered Laplacian levels, while explicitly omitting the top (Gaussian) level $l = 0$ to be resilient to photometric deviations due to local illumination changes. We, however, use a different outlier classification scheme than [4].

In this pipeline stage, the main challenge is to correctly classify novel details as inliers, even if they create discrepancies with the model. By comparing a Laplacian decomposition of the warped frame, $\mathcal{I}^l_{\text{curr}\to\mathcal{M}}$, with the Laplacian model pyramid $\mathcal{I}^l_{\mathcal{M}}$, we are able to exploit that true outliers are geometrically inconsistent across all levels, whereas novel details are in a mismatch on the finest level(s) only (see Fig. 4). Thus, we determine a per-pixel *similarity score* $s^l_{\text{curr}} \in \mathbb{R}$ w.r.t. $\mathcal{M}$ separately for each Laplacian level $l < 0$, starting with the coarsest Laplacian level $l = -1$:

$$s^{l=-1}_{\text{curr}} = SSIM^{\text{C,S}}\left(\mathcal{I}^{l=-1}_{\text{curr}\to\mathcal{M}}, \mathcal{I}^{l=-1}_{\mathcal{M}}\right), \tag{8}$$

where $SSIM^{\text{C,S}} \in \mathbb{R}$ is the similarity metric given in Eq. (10).

Since we can only distinguish outliers from novel details on coarser levels, where these frequencies are already present in the model, we then propagate the similarity score to the finest level $l = l_{\text{min}}$ by retaining high similarities from coarser levels:

$$s^l_{\text{curr}} = max\left(SSIM^{\text{C,S}}\left(\mathcal{I}^l_{\text{curr}\to\mathcal{M}}, \mathcal{I}^l_{\mathcal{M}}\right), [s^{l+1}_{\text{curr}}]_{\uparrow 2}\right), \tag{9}$$

where $[\dots]_{\uparrow 2}$ indicates an up-sampling by one octave. Fig. 4 illustrates this scheme, showing the computation of the similarity score and the effect of our propagation strategy.

As similarity metric $SSIM^{\text{C,S}}$, we use a variant of *SSIM* [45] suitable for being applied to Laplacian images. The original *SSIM* offers a *structural similarity index measure* between two intensity images $X \in \mathbb{R}$ and $Y \in \mathbb{R}$, with $SSIM \in [-1, +1]$ and can be broken down into three independent components: a comparison for luminance, contrast, and structure. Since we apply the metric on Laplacian levels, we discard the luminance component:

$$SSIM^{\text{C,S}}(X, Y) = max\left(\left[\frac{2\mu_X\mu_Y}{\mu_X^2 + \mu_Y^2}\right]^{\beta}\left[\frac{\sigma_{XY}}{\sigma_X\sigma_Y}\right]^{\gamma}, 0\right), \tag{10}$$

comprising the product of contrast and structure similarity. μ_X, μ_Y are the means of X and Y within a local window; σ_X^2, σ_Y^2 the variances; and σ_{XY} the covariance. We set the weighting parameters to $\beta = 1$, $\gamma = 1$ and clamp the result to ensure $SSIM^{\text{C,S}} \in [0, 1]$.

While the contrast comparison serves a similar purpose as the error metric of [4], we additionally compare the local structure instead of individual pixels. The local window size is set adaptively and increases according to finer pyramid levels l, starting with radius $r_0 = 1$px.

Finally, we classify pixels (x, y) on levels l as outliers if their similarity score $s^l_{\text{curr}}(x, y)$ falls below $\varepsilon_0 = 0.15$. In the following fusion stage, $s^l_{\text{curr}} \in [0, 1]$ is further used to weight inliers according to their achieved score (see Eq. (11)).

4.7. Fusion

In the final stage of the pipeline, the current frame $\{\mathcal{I}_{\text{curr}\to\mathcal{M}}, \mathcal{D}_{\text{curr}\to\mathcal{M}}\}$ is fused with the current model $\{\mathcal{I}_{\mathcal{M}}, \mathcal{D}_{\mathcal{M}}\}$.

Color fusion. Conceptually, our frequency-oriented color fusion approach follows Kluge et al. [4]. That is, we merge the Laplacian levels of the color pyramids while retaining the base color of the Gaussian level and, thus, enable progressive refinement without requiring local or global optimization for color harmonization. However, our approach designed for fusing warped observations of 3D scenes requires a different accumulation scheme.

The accuracy and reliability of the warped color $\mathcal{I}_{\text{curr}\to\mathcal{M}}$ is primarily limited by the underlying depth data due to inaccurate or even false depth estimates captured at low(er) resolution. Thus, in contrast to Kluge et al. [4], which is based on a planar scene and a replacement strategy, we apply a blending scheme of multiple observations, as each single, warped observation is not reliable enough by itself.

Our blending scheme only fuses inlier pixels (x, y) with a finer level of refinement, i.e., if $\mathcal{L}^l_{\text{curr}}(x, y) \leq \mathcal{L}^l_{\mathcal{M}}(x, y)$, to prevent coarser observations from degrading the model. We apply

$$\mathcal{I}^l_{\mathcal{M}} \leftarrow \frac{\mathcal{I}^l_{\mathcal{M}} + w_{\text{curr}} s^l_{\text{curr}} \mathcal{I}^l_{\text{curr}\to\mathcal{M}}}{1 + w_{\text{curr}} s^l_{\text{curr}}}, \tag{11}$$

to levels $l \in [l_{\text{min}}, \dots, -1]$ and, thus, update all corresponding Laplacian levels with data from the new observation. Here, $s^l_{\text{curr}} \in [0, 1]$ is the score determined in Section 4.6, which we use to lower the contribution of less reliable input color. Apart from that, the weight w_{curr} applied to the observation is computed as

$$w_{\text{curr}} = \Delta_{\text{curr}} + \frac{1}{c^l_{\mathcal{M}}}, \quad \text{with} \quad \Delta_{\text{curr}} = min\left(\left|\mathcal{L}^l_{\text{curr}} - \mathcal{L}^l_{\mathcal{M}}\right|, \Delta_{\text{max}}\right), \tag{12}$$

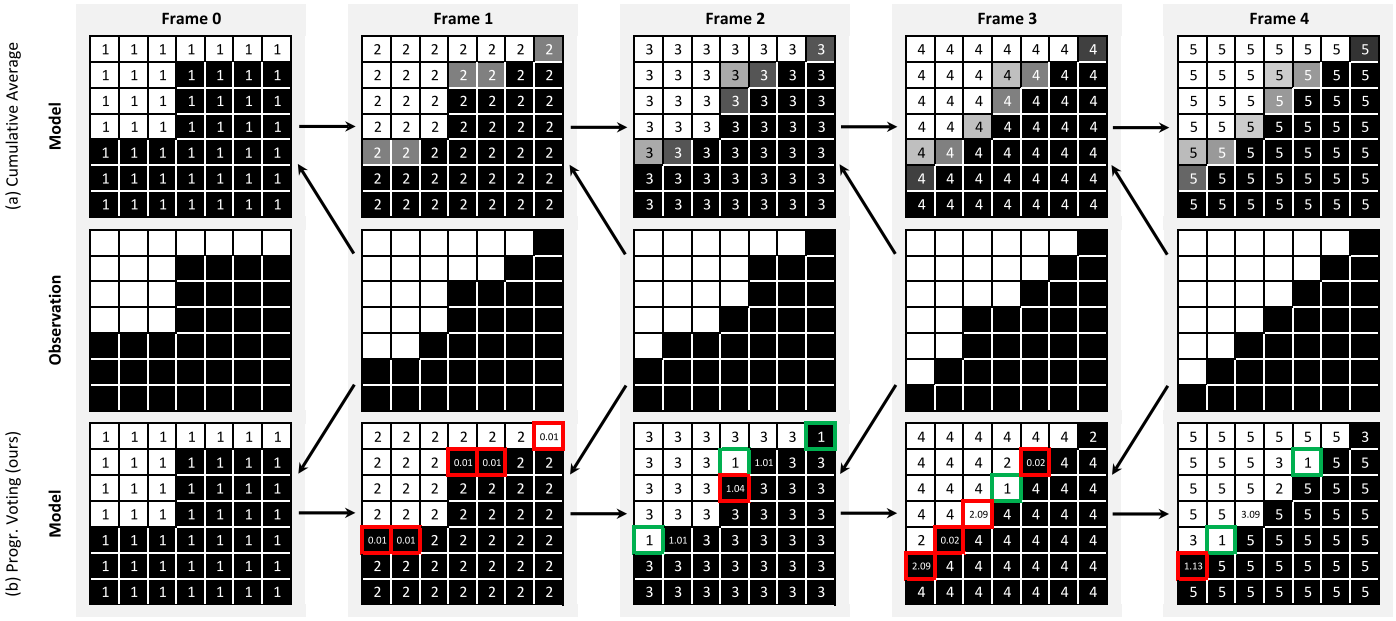

Fig. 5. Fusion of erroneous pixels at depth discontinuities for a set of example depth maps with foreground (black) and background depths (white). At frame 0, the model is initialized with the coarse observation depths (*left column*). At subsequent frames, the current observation (*middle row*) is fused with the model of the previous frame. (a) *Top row:* Cumulative average, where the pixel's counter is successively incremented. Blending of incompatible pixels results in flying pixels between the foreground and background depth. (b) *Bottom row:* Using our voting strategy, the model depths progressively approach the correct depths (shown in frame 4 for the observation) by replacing pixels at depth discontinuities. The pixel's voting counter is decremented (Eq. (16)) if the observation and its corresponding model depth are incompatible (highlighted in red); otherwise incremented. In case too many observations voted against a model pixel, i.e., a pixel's voting counter becomes negative, its depth is replaced by the current observation (highlighted in green), and the counter is reset to 1 (Eq. (17)).

where Δ_{curr} represents the gain in level of refinement (colored green in Fig. 3), $c^l_{\mathcal{M}}$ is the model's counter, and $\mathcal{L}^l_{\text{curr}}$ is the Gaussian decomposition of $\mathcal{L}_{\text{curr}}$. To reflect the amount of detail blended into the model so far, the model's level of refinement, $\mathcal{L}^l_{\mathcal{M}}$, is updated analogously to Eq. (11) as a weighted average, using

$$\mathcal{L}^l_{\mathcal{M}} \leftarrow \frac{\mathcal{L}^l_{\mathcal{M}} + w_{\text{curr}} s^l_{\text{curr}} \mathcal{L}^l_{\text{curr}}}{1 + w_{\text{curr}} s^l_{\text{curr}}}, \tag{13}$$

while the counter is incremented by

$$c^l_{\mathcal{M}} \leftarrow c^l_{\mathcal{M}} + 1. \tag{14}$$

With $\Delta_{\text{curr}} = 0$, Eq. (12) reduces to the basic blending scheme in incremental scene reconstruction, a *cumulative average* of samples [3,46], i.e., the observation's weight $w = 1/c_{\mathcal{M}}$ is decreasing continuously as the model's counter $c_{\mathcal{M}} \in [1, \dots, \infty]$ is incremented with each observation. In refinement imaging, this averaging scheme potentially prevents details captured by later observations from getting into the model. This happens specifically when many (early) observations with less details force up the weight. Our approach, therefore, takes the gain in level of refinement $\left|\mathcal{L}^l_{\text{curr}} - \mathcal{L}^l_{\mathcal{M}}\right|$ into account and combines it with the traditional confidence counter, defined by the number of observations ($1/c_{\mathcal{M}}$). To limit the maximum contribution of a single observation and, thus, to prevent the model from being replaced, we clamp Δ_{curr} at $\Delta_{\max} = 0.1$.

Depth fusion. The imperfect nature of depth images requires a different way of fusion, as no reliable initial reference depth is available, which could be used for (additive) refinement. Instead, inaccurate depths need to be corrected and false values have to be detected and replaced.

We filter observation depths $\mathcal{D}_{\text{curr}\to\mathcal{M}}$ that are incompatible with the model $\mathcal{D}_{\mathcal{M}}$, using the depth tolerance threshold $|\mathcal{D}_{\mathcal{M}} - \mathcal{D}_{\text{curr}\to\mathcal{M}}| \le \varepsilon_{\text{d}}$ as compatibility criterion. We then blend compatible pixels on pyramid level $l_{\min}$ by the weighted average

$$\mathcal{D}_{\mathcal{M}} \leftarrow \frac{v_{\mathcal{M}} \mathcal{D}_{\mathcal{M}} + \mathcal{D}_{\text{curr}\to\mathcal{M}}}{v_{\mathcal{M}} + 1}, \tag{15}$$

to improve the accuracy of model depths $\mathcal{D}_{\mathcal{M}}$ over time. However, in the case of initializing $\mathcal{D}_{\mathcal{M}}(x, y)$ with a false value, further observations will fail the compatibility test, inhibiting any refinement.

Therefore, we propose an incremental voting strategy to find a suitable model value progressively (see Fig. 5). With the intention that each new observation votes either for or against the reliability of a model pixel's depth, we interpret $v_{\mathcal{M}} \in \mathbb{R}$ as a *voting counter*. For each fusion that failed due to incompatibility with the model, we decrease a model pixel's counter, yielding the following counter update:

$$v_{\mathcal{M}} \leftarrow \begin{cases} v_{\mathcal{M}} - e^{-(v_{\mathcal{M}}/\sigma)^2}, & \text{if } |\mathcal{D}_{\mathcal{M}} - \mathcal{D}_{\text{curr}\to\mathcal{M}}| > \varepsilon_{\text{d}}. \\ v_{\mathcal{M}} + 1, & \text{otherwise}. \end{cases} \tag{16}$$

Here, $e^{-(v_{\mathcal{M}}/\sigma)^2}$ is used to control the amount of decrease in case of an incompatible observation. Our approach ensures a stable result once a model depth has been consolidated, while it quickly discards less reliable model values in favor of a more frequently observed depth value. For all our experiments, we set $\sigma = 10$.

In case a pixel's voting counter falls below 0, i.e., if $v_{\mathcal{M}} \le 0$, its depth value is replaced and the counter is reset:

$$\mathcal{D}_{\mathcal{M}} \leftarrow \mathcal{D}_{\text{curr}\to\mathcal{M}}, \quad v_{\mathcal{M}} \leftarrow 1. \tag{17}$$

Fig. 5 illustrates this voting scheme, showing the resulting fusion compared to a cumulative average. In the supplementary material, an alternative visualization is given, demonstrating the effect of the resulting weighting.

4.8. Final output

After the final frame of the RGB-D input sequence has passed the pipeline stages described in Sections 4.2 to 4.7, the model

pyramids $\mathcal{I}_{\mathcal{M}}$ and $\mathcal{D}_{\mathcal{M}}$ are recomposed to produce the final refined RGB-D image $\mathcal{I}^{\text{comp}}_{\mathcal{M}}$ and $\mathcal{D}^{\text{comp}}_{\mathcal{M}}$ from the fixed viewpoint $\boldsymbol{T}_{\mathcal{M}}$. That is, the Laplacian color pyramid $\mathcal{I}_{\mathcal{M}}$ is recomposed by upsampling and summing all Laplacian levels $\mathcal{I}^{l}_{\mathcal{M}}$. For the model depth $\mathcal{D}_{\mathcal{M}}$, all tiles are sampled up to the finest pyramid level existing in the model $\mathcal{M}$. Finally, after combining all tiles to a full image, $\{\mathcal{I}^{\text{comp}}_{\mathcal{M}}, \mathcal{D}^{\text{comp}}_{\mathcal{M}}\}$ is a refined version of the initial frame $\{\mathcal{I}_0, \mathcal{D}_0\}$, with a resolution up to a multiple of the initial resolution. Theoretically, by using the entire operating range of 8.0 m to 0.5 m for a typical RGB-D camera such as Kinect v2, the object-space resolution can be increased by a factor of 16, reaching several hundred megapixels for the final reconstruction (e.g., 530.8 MP when using a 2.1 MP image sensor). In our evaluation, however, a scale factor of 6 to 10 was reached for the outdoor data sets (see Section 6.1).

5. Implementation

Our reconstruction pipeline is implemented in C++, incorporating basic image processing operations from the OpenCV library. The pre-processing, outlier classification, and dense ICP are implemented on the GPU using CUDA. We use OpenCV's SURF feature detection for the camera pre-alignment, whereas Farnebäck's optical flow variant [44], provided by OpenCV, is used for local re-alignment. For rendering the input depth map from the reference pose, OpenGL is used by exploiting z-buffering. Lastly, the fusion of color and depth data is performed in image space using CUDA operations.

Although model color and depth share the same hierarchical structure (see Fig. 2), they are stored separately in two sparsely occupied image pyramids, each with additional layers for the associated attribute maps (e.g., the counter). Each pyramid level comprises a 2D array of pointers referring to the allocated image tiles currently in use.

To improve the computational efficiency of our online approach towards real-time applications, ideally, concurrent kernel scheduling should be applied to overlap data transfers and other operations by performing multiple CUDA operations simultaneously, which has yet to be realized in our current implementation.

6. Results

6.1. Data sets

Fig. 6 shows the reference images of the ten data sets we use for our evaluation. Besides the *Fountain* and the *LongOfficeHousehold* data sets, taken from Zhou and Koltun [6] and Sturm et al. [47], respectively, we created the following indoor as well as outdoor data sets that comprise medium to large disparities and, partially, very challenging situations in terms of reflective objects, fine scene details and high noise levels (dark/black objects). For each data set, $scale_{\max}$ denotes the maximum scale factor of object-space resolution with respect to the reference image that is featured by the input data ($scale_{\max} = 2.46$ for *Fountain* and $scale_{\max} = 2.77$ for *LongOfficeHousehold*).

CoffeeTable: This indoor scene comprises highly reflective objects, e.g., a coffee machine and a black metal box ($scale_{\max} = 4.38$).

BooksGlobe: An indoor scene that contains several books, a blanket, and a globe arranged on a couch/bed ($scale_{\max} = 2.25$).

VillageModel: An indoor scene that comprises a set of model houses arranged on a table in front of a display screen. This scene comprises very small, dark, and mainly diffuse objects ($scale_{\max} = 3.89$).

BrickWall: An outdoor scene with low depth variations that displays mainly diffuse stone colors ($scale_{\max} = 6.96$).

Table 2
Data set specifications. The data sets are acquired using the Asus Xtion Pro Live (pre-registered RGB-D: 640 × 480 px) and the Kinect v2 (pre-registered RGB-D: 1920 × 1080 px), comprising '# frames' frames, where '# fused frames' frames are selected by the specific method to be fused into the final result.

	Resolution (px)	# frames	# fused frames			
			Kluge20	*Fu21*	*Niessner13*, *Lee20*, *Ha21*	*Ours*
Fountain	640 × 480	1086	–	36	1086	59
LongOfficeH.	640 × 480	2488	–	–	–	31
CoffeeTable	1920 × 1080	2778	–	28	2778	186
BooksGlobe	1920 × 1080	370	–	5	370	26
VillageModel	1920 × 1080	2472	–	–	2472	162
BrickWall	1920 × 1080	7420	498	–	7420	496
Memorial	1920 × 1080	4037	356	–	4037	266
Statue	1920 × 1080	1515	–	–	1515	96
Cannon	1920 × 1080	677	–	–	677	43
FlowerBed	1920 × 1080	728	–	–	728	48

Memorial: This outdoor data set comprises mainly diffuse objects with medium disparities ($scale_{\max} = 9.71$).

Statue: An outdoor data set with statues at a fountain with large disparities and highly reflective water ($scale_{\max} = 5.70$).

Cannon: This outdoor data set contains a cannon (glossy, black) and large disparities ($scale_{\max} = 6.23$).

FlowerBed: An outdoor scene of an arrangement of flowers with very unreliable depth data due to semi-transparent leaves and very fine details ($scale_{\max} = 6.10$).

We display each reconstruction from the initial pose and zoom into a challenging sub-region as insets. We present all images in high resolution in the supplementary material. Table 2 summarizes the main data set specifications.

6.2. Ablation study

In this section, we evaluate the performance of our progressive refinement imaging using depth-assisted disparity correction by replacing core concepts of our pipeline with earlier approaches. We show the resulting effects in Fig. 7 for the *CoffeeTable* data set and in Fig. 8 for the *VillageModel* data set.

Outlier classification scheme. Fig. 7(a) shows the outlier removal to achieve local color consistency from Kluge et al. [4], and Fig. 7(d) depicts the result when applying our new *SSIM*-based Laplacian scheme presented in Section 4.6. The result obtained with our outlier removal scheme yield further color refinement, specifically at object borders with less reliable warped color information, avoiding misclassifying novel details as outliers.

Accumulation strategy for color fusion. In Fig. 7(b), the pyramidal color replacement strategy proposed by Kluge et al. [4] is shown, while Fig. 7(d) depicts the result obtained by our novel blending method described in Section 4.7. Comparing both results, we can see that the replacement scheme leads to strong artifacts at object boundaries and other areas with unreliable depth data, causing the reconstruction to suffer from noise and distorted colors. In contrast, our approach results in a geometric and photometric consistent reconstruction.

Weighting scheme for color fusion. Fig. 7(c) shows the color fusion result using a conventional cumulative averaging scheme used by, for instance, Newcombe et al. [3], and Fig. 7(d) gives the result when applying our new approach that takes the gain of visual detail into account, as described in Section 4.7. We observe that the classical weighting scheme is not able to incorporate as much color detail as our scheme, leading to a more blurred result.

Fig. 6. The unrefined reference images (initial frames) of the data sets.

Fig. 7. Ablation study for color reconstruction. (a) Our approach combined with the Laplacian outlier removal scheme from *Kluge20* [4]. (b) Our approach combined with the Laplacian color replacement strategy for color fusion from *Kluge20* [4]. (c) Our approach combined with the conventional cumulative average weighting, e.g., [3]. (d) Ours with the proposed *SSIM*-based outlier removal scheme and the proposed color blending with detail gain-based weighting.

Fig. 8. Ablation study for depth reconstruction. (a) Our approach, but with the conventional cumulative average weighting, e.g., [3]. (b) Ours with the proposed depth voting scheme. Depths are shown using a *Parula* colormap ranging from 1.5 m to 2.75 m.

Voting strategy for depth fusion. The effect of using our novel voting scheme for depth values presented in is given as a depth map in Fig. 8(b), compared to the application of a conventional depth averaging of compatible pixels as used by, for instance, Newcombe et al. [3], depicted in Fig. 8(a). The strength of our depth voting scheme gets specifically apparent at depth discontinuities, i.e., object silhouettes, where our approach refines the initial depths of the coarse object boundaries by detecting and replacing erroneous measurements.

6.3. Qualitative comparisons

As our approach provides a high-quality image refinement method robust to disparity and occlusions and, thus, aims at filling the gap between interactive 2D image refinement methods, online 3D reconstruction techniques with high-resolution textures, and offline texture optimization methods for 3D scene reconstruction, we compare our approach to the following state-of-the-art techniques in these contexts.

Kluge20: 2D interactive progressive refinement imaging [4] for (almost) planar scenes with only small amounts of disparity.

Niessner13: The online 3D scene reconstruction method using voxel hashing from Niessner et al. [48]. This approach is used by most of the color optimization methods, such as [6,7,19].

Lee20: The online 3D scene reconstruction method *TextureFusion* from Lee et al. [27] stores sub-voxel textures in the TSDF voxel grid cells containing the scene surface.

Ha21: The online 3D scene reconstruction method *NormalFusion* from Ha et al. [28], a follow-up work of [27], additionally obtains photometric normals, enabling geometric enhancement.

Fu21: The offline texture optimization proposed by Fu et al. [7]. We generate the initial scene reconstruction and camera poses using VoxelHashing [48] and select a subset of input frames based on the angle and distance between corresponding poses, as proposed in [7].

Ours: Our proposed method as described in Sections 3 and 4.

Comparison to 2D image reconstruction. We compare our approach to the 2D image reconstruction method *Kluge20* on the *BrickWall* and the *Memorial* data sets, which comprise a low to moderate amount of disparity; see Fig. 9.

For the *BrickWall* data set, *Kluge20* works robustly and yields quite good results. However, for the *Memorial* data set, the limitations of the geometric alignment using a homography lead to strong geometric ghosting artifacts, while our method is able to reconstruct the silhouettes and captures more details than *Kluge20*. Note that *Kluge20* does not generate results on any of the other data sets due to alignment failures. In the supplementary material, we additionally compare our method to the 2D photo stitching method *Autopano* [12], which is based on Brown et al.'s *AutoStitch* [10,11].

Comparison to online scene reconstruction. We reconstruct all data sets using the online 3D scene reconstruction approaches *Niessner13* (VoxelHashing), *Lee20* (NormalFusion), and *Ha21* (TextureFusion) as a comparison to our method; see Fig. 10. To achieve the

Fig. 9. Comparison with the 2D method from *Kluge20* [4]. See also Fig. 6 for a comparison with the unrefined reference image.

Table 3
Voxel sizes (mm) for the data sets, used by the competing methods.

	Fountain	*CoffeeTable*	*BooksGlobe*	*VillageModel*	*BrickWall*	*Memorial*	*Statue*	*Cannon*	*FlowerBed*
Niessner13	4	4	4	4	4	4	4	4	4
Lee20	4	4	4	4	8	5	4	4	6
Ha21	4	6	4	8	25	9	10	11	14

most detailed results, we used the smallest possible voxel size to successfully process a specific data set with 24 GB of GPU memory, if the reconstruction failed with the default size of 4 mm; see Table 3. Note that *Ha21* generates photometric normals as additional per-voxel attribute maps besides the texture patches, which requires a significant amount of memory, depending on the scene. The table with all hyper-parameters of the competing methods is given in the supplementary material.

All methods successfully reconstruct all scenes, but due to the nature of the 3D scene representation, 3D scene reconstruction methods potentially produce holes or incomplete color reconstructions. We observe further scene-dependent deficiencies, which we exemplify in the following. *Niessner13* exhibits, for example, local geometric inconsistencies (*Fountain*, *BooksGlobe*, *VillageModel*), as well as smoothed-out photometric reconstructions (*CoffeeTable*, *Statue*), but is partially able to reconstruct texture details (*Cannon*). *Lee20* partly reconstructs sharp details (*Fountain*) and silhouettes (*Memorial*), but also produces very blurry results (*CoffeeTable*, *BooksGlobe*, *Cannon*). Likewise, *Fu21* can partially reconstruct sharp details (*Fountain*, *BooksGlobe*) while delivering blurry results in other cases (*VillageModel*, *Cannon*).

Besides the *FlowerBed* data set, our method yields high-quality results regarding geometric and photometric consistency. Our method can successfully refine the reference image in geometrically homogeneous regions as well as at object silhouettes, and suppresses locally misaligned information (e.g., due to erroneous input range values).

The *FlowerBed* data set is very challenging, as it comprises many detailed silhouettes for which the range maps are not detailed and reliable enough. This leads to a large amount of outliers and to a comparably small amount of details that pass the outlier test and get incorporated into the reconstructed RGB-D image.

In addition, we provide a quantitative comparison using a synthetic data set with ground truth in the supplementary material. We evaluate all methods by employing different error metrics, both for the refined color and depth, revealing a significant advantage of our method.

Comparison to offline optimization. Fig. 11 shows the results of comparing our method to the offline, global post-optimization approach *Fu21* for the *Fountain, CoffeeTable*, and *BooksGlobe* data sets. Note that the *Fountain* data set footage comprises only limited amounts of close-ups of the specular tilework. For all three data sets, *Fu21* delivers geometrically good results, but there are photometric inconsistencies. Our method yields reconstructions with significantly improved photometric consistency, as the reference frame's illumination condition is retained. In the supplementary material, we further demonstrate the robustness of our pipeline against illumination changes and differences in white-balance or auto-exposure in the input footage.

Comparison of reconstructed depths. While our main output is a high-quality color reconstruction, the resulting depth map may have its uses (e.g., for stereo image generation). Therefore, we compare our depth map reconstruction to the scene reconstructions of *Niessner13*, *Lee20*, and *Ha21* by rendering the surface from the same viewpoint. The results of this experiment are shown in Fig. 12 for the *VillageModel* data set. While all approaches show competitive results, our method provides more consistent object silhouettes and fewer holes.

6.4. Robustness against self-localization drift

To demonstrate the robustness of our method against drift effects in camera tracking, we use the 360° data set *LongOffice-Household*, comprising 2488 RGB-D frames. Our system processes the first 326 frames, i.e., it selects 13 frames to be incorporated into the model. Later on, when the camera turns closer to the reference pose again, frames 1771–2488 are processed, from which 18 frames are selected. Fig. 13 shows the refinement before exiting the reference viewpoint (left) and the final refinement after re-entering the reference viewpoint (right), yielding a sharper reconstruction.

6.5. Performance

All experiments are performed using an AMD Ryzen Threadripper 3970X with 128 GB main memory and an NVIDIA GeForce RTX 4090 with 24 GB GPU memory. Table 4 states the timings for a complete reconstruction process of each method and the required peak memory. For *Niessner13, Lee20* and *Ha21*, we show the memory consumption using the minimal amount of pre-allocated data structure elements, determined using two passes. Note that in a true online scenario, this is not known beforehand, and thus, more memory would have been pre-allocated. Since the offline, post-processing method *Fu21* requires a large amount of processing time, up to several weeks (*CoffeeTable*), only three data sets are shown for this method; we stopped the *Statue* and *FlowerBed* data sets after ten and six days, respectively, when only the first of 30 iterations had been completed. For our method, the average frame rate over all data sets is 1.0 fps, with a minimum frame rate of 0.5 fps for the *Statue* data set and a maximum frame rate of 2.1 fps for the *Fountain* data set.

Fig. 10. Comparison with online scene reconstruction methods. See also Fig. 6 for a comparison with the unrefined reference image.

Table 4
Required resources.

	Total processing time (h:min:s)						Peak total main memory consumption (GB)						Peak total GPU memory consumption (GB)					
	Kluge20	*Niessner13*	*Lee20*	*Ha21*	*Fu21*	*Ours*	*Kluge20*	*Niessner13*	*Lee20*	*Ha21*	*Fu21*	*Ours*	*Kluge20*	*Niessner13*	*Lee20*	*Ha21*	*Fu21*	*Ours*
Fountain	–	0:00:11	0:00:27	0:00:43	142:19:20	0:00:28	–	1.78	7.01	9.55	7.60	1.14	–	2.16	6.18	13.47	0.36	2.77
LongOfficeH.	–	–	–	–	–	0:00:18	–	–	–	–	–	1.13	–	–	–	–	–	1.86
CoffeeTable	–	0:01:03	0:02:35	0:05:13	483:25:02	0:03:38	–	4.88	60.23	60.38	13.90	1.32	–	3.67	11.78	20.80	0.25	9.46
BooksGlobe	–	0:00:16	0:00:20	0:00:40	23:13:39	0:00:28	–	1.77	10.96	12.57	4.40	1.27	–	3.24	7.20	12.58	0.51	4.99
VillageModel	–	0:00:56	0:02:23	0:04:49	–	0:04:01	–	4.54	53.35	54.21	–	1.37	–	3.87	12.48	20.46	–	6.48
BrickWall	0:08:58	0:03:15	0:07:55	0:13:17	–	0:10:21	1.64	21.56	122.04	123.07	–	1.44	8.24	9.55	22.19	19.12	–	11.61
Memorial	0:04:47	0:01:45	0:03:43	0:08:56	–	0:05:04	1.71	10.36	85.12	86.15	–	1.33	7.54	5.26	21.29	22.49	–	8.47
Statue	–	0:00:39	0:01:32	0:02:51	–	0:03:10	–	5.06	35.42	35.21	–	1.48	–	4.60	16.49	19.63	–	12.33
Cannon	–	0:00:16	0:00:44	0:01:03	–	0:01:01	–	3.25	18.74	18.51	–	1.41	–	4.86	21.41	20.21	–	10.79
FlowerBed	–	0:00:20	0:00:43	0:01:07	–	0:00:55	–	4.57	19.62	19.48	–	1.34	–	5.87	21.71	20.12	–	8.18

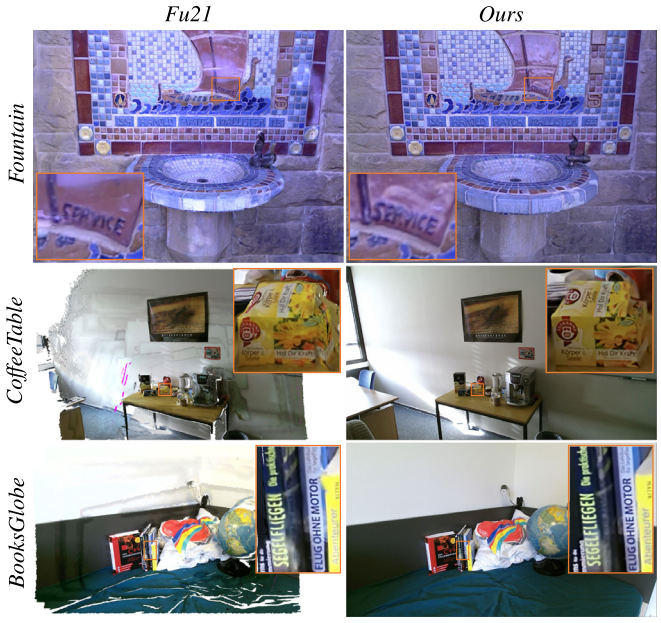

Fig. 11. Comparison with the offline, post-processing approach *Fu21* [7]. See also Fig. 6 for a comparison with the unrefined reference image.

6.6. Limitations

In order to enable refinement imaging with parallax effects in the scene, our method primarily depends on depth values to guide the alignment and disparity-corrective warping of the color information. However, in contrast to high-quality color data, depth images exhibit lower effective resolution and significantly increased noise, often correlated with visually important features like silhouettes. While our approach is explicitly designed for resilience against these low-quality characteristics, it is ultimately limited by the depth data provided.

Our method is not able to reliably refine RGB-D data sequences containing too fine-grained depth variations and silhouettes, resulting in too much unreliable depth information and outliers to be used for disparity correction. This is particularly evident in the *FlowerBed* data set, evaluated in Section 6.3, which comprises very detailed silhouettes in the color data for which the depth data's reliability is insufficient. Even in homogeneous depth areas, range estimations may exhibit increased noise and erroneous values, e.g., on specular surfaces (such as the coffee machine in the *CoffeeTable* data set). Since this directly affects the accuracy of the color warping, the local realignment may not be sufficient.

Furthermore, our pipeline maintains photometric and geometric consistency with respect to a reference image that needs to cover the scene of interest entirely. To avoid introducing photometric inconsistencies, unlike [4], we do not extend the lateral dimensions of the model to incorporate novel scene areas if the camera is exiting the region defined by the reference image.

7. Conclusions

We presented a novel progressive RGB-D image refinement pipeline that instantaneously produces a high-quality, geometrically and photometrically consistent RGB-D image reconstruction from RGB-D image sequences. Assisted by depth values to guide the alignment and to correct for disparity, our design allows for the refinement of general 3D scenes and, thus, fills the gap between 2D progressive refinement imaging and online 3D reconstruction techniques with high-resolution textures.

Colors and depths are hierarchically fused into an adaptive-resolution, progressively improving model of the scene, while strictly decoupling color data from the coarse and potentially incomplete geometry representation. Our pipeline modules are designed for resilience against low-quality, low-resolution depth information while refining the high-resolution color data in homogeneous depth regions as well as at object silhouettes. To that end, our method performs local color consistency operations in image space before applying a novel blending strategy for color fusion, taking the gain in visual detail into account. To benefit from progressively refined range values, depths are fused based on a novel depth voting scheme that allows for correcting inaccurate depth estimates.

CRediT authorship contribution statement

Markus Kluge: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Tim Weyrich:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Supervision. **Andreas Kolb:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2023.07.036.

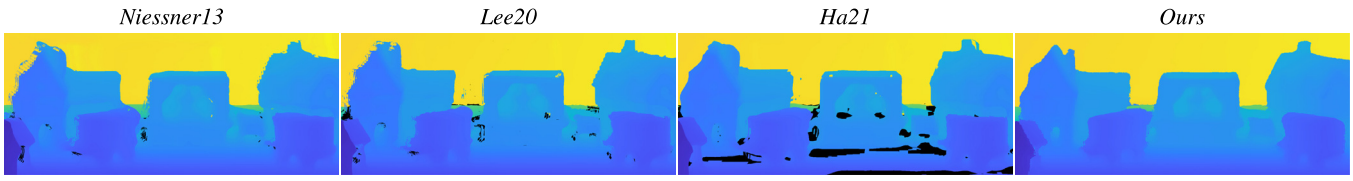

Fig. 12. Comparison of the reconstructed depths for the *VillageModel* data set, using a *Parula* colormap ranging from 1.5 m to 2.75 m.

Fig. 13. Robustness against self-localization drift. Refinement of the *LongOffice-Household* data set before exiting the reference viewpoint (left) and the final refinement after re-entering the reference viewpoint (right) using our approach. See also Fig. 6 for a comparison with the unrefined reference image.

References

[1] Rusinkiewicz S, Hall-Holt O, Levoy M. Real-time 3D model acquisition. ACM Trans Graph 2002;21(3):438–46.

[2] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proc. ACM symp. user interface softw. & tech.. 2011, p. 559–68.

[3] Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, et al. Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. Ieee; 2011, p. 127–36.

[4] Kluge M, Weyrich T, Kolb A. Progressive refinement imaging. Comput Graph Forum 2020;39(1):360–74.

[5] Licorish C, Faraj N, Summa B. Adaptive compositing and navigation of variable resolution images. In: Computer graphics forum, vol. 40. 2021, p. 138–50.

[6] Zhou Q-Y, Koltun V. Color map optimization for 3d reconstruction with consumer depth cameras. ACM Trans Graphics 2014;33(4):1–10.

[7] Fu Y, Yan Q, Liao J, Zhou H, Tang J, Xiao C. Seamless texture optimization for RGB-D reconstruction. IEEE Trans Vis Comput Graphics 2021.

[8] Szeliski R, Shum H-Y. Creating full view panoramic image mosaics and environment maps. In: Proc. SIGGRAPH. 1997, p. 251–8.

[9] Agarwala A, Dontcheva M, Agrawala M, Drucker S, Colburn A, Curless B, et al. Interactive digital photomontage. ACM Trans Graphics 2004;23(3):294–302.

[10] Brown M, Lowe DG. Automatic panoramic image stitching using invariant features. Int J Comput Vision (IJCV) 2007;74(1):59–73.

[11] Brown M. AutoStitch 3.0. 2018, Available from: http://matthewalunbrown.com/autostitch/autostitch.html. [Accessed 18 October 2022].

[12] Kolor. Kolor autopano giga 4.4.2. 2018, Available from: https://download.kolor.com/apg/stable/history. [Accessed 18 October 2022].

[13] Eisemann M, Eisemann E, Seidel H-P, Magnor M. Photo zoom: High resolution from unordered image collections. In: Proc. graphics interface. 2010, p. 71–8.

[14] Burns C, Plyer A, Champagnat F. Texture super-resolution for 3D reconstruction. In: Proc. international conference on machine vision applications. 2017, p. 350–3.

[15] Waechter M, Moehrle N, Goesele M. Let there be color! large-scale texturing of 3D reconstructions. In: Proc. Europ. conf. computer vision. 2014, p. 836–50.

[16] Maier R, Kim K, Cremers D, Kautz J, Nießner M. Intrinsic3d: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In: Proc. IEEE int. conf. computer vision. 2017, p. 3114–22.

[17] Bi S, Kalantari NK, Ramamoorthi R. Patch-based optimization for image-based texture mapping.. ACM Trans Graphics 2017;36(4). 106–1.

[18] Fu Y, Yan Q, Yang L, Liao J, Xiao C. Texture mapping for 3d reconstruction with RGB-D sensor. In: Proc. IEEE conf. computer vision and pattern recognition. 2018, p. 4645–53.

[19] Fu Y, Yan Q, Liao J, Xiao C. Joint texture and geometry optimization for RGB-D reconstruction. In: Proc. IEEE conf. computer vision and pattern recognition. 2020, p. 5950–9.

[20] Huang J, Dai A, Guibas LJ, Nießner M. 3Dlite: towards commodity 3D scanning for content creation. ACM Trans Graphics 2017;36(6). 203–1.

[21] Jeon J, Jung Y, Kim H, Lee S. Texture map generation for 3D reconstructed scenes. Vis Comput 2016;32(6):955–65.

[22] Liu S, Li W, Ogunbona P, Chow Y-W. Creating simplified 3D models with high quality textures. In: Proc. international conference on digital image computing: Techniques and applications. 2015, p. 1–8.

[23] Maier R, Stückler J, Cremers D. Super-resolution keyframe fusion for 3D modeling with high-quality textures. In: Proc. International conference on 3D vision. 2015, p. 536–44.

[24] Rouhani M, Fradet M, Baillard C. A multi-resolution approach for color correction of textured meshes. In: Proc. international conference on 3D vision. 2018, p. 71–8.

[25] Wang C, Guo X. Plane-based optimization of geometry and texture for RGB-D reconstruction of indoor scenes. In: Proc. international conference on 3D vision. 2018, p. 533–41.

[26] Meilland M, Comport AI. Super-resolution 3D tracking and mapping. In: Proc. IEEE int. conf. robotics and automation. 2013, p. 5717–23.

[27] Lee JH, Ha H, Dong Y, Tong X, Kim MH. Texturefusion: High-quality texture acquisition for real-time RGB-D scanning. In: Proc. IEEE conf. computer vision and pattern recognition. 2020, p. 1272–80.

[28] Ha H, Lee JH, Meuleman A, Kim MH. NormalFusion: Real-time acquisition of surface normals for high-resolution RGB-D scanning. In: IEEE conference on computer vision and pattern recognition. 2021.

[29] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. NeRF: Representing scenes as neural radiance fields for view synthesis. In: Proc. Europ. conf. computer vision. 2020, p. 405–21.

[30] Martin-Brualla R, Radwan N, Sajjadi MS, Barron JT, Dosovitskiy A, Duckworth D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proc. IEEE conf. computer vision and pattern recognition. 2021, p. 7210–9.

[31] Meng Q, Chen A, Luo H, Wu M, Su H, Xu L, et al. Gnerf: Gan-based neural radiance field without posed camera. In: Proc. IEEE int. conf. computer vision. 2021, p. 6351–61.

[32] Sucar E, Liu S, Ortiz J, Davison AJ. iMAP: Implicit mapping and positioning in real-time. In: Proc. IEEE int. conf. computer vision. 2021, p. 6229–38.

[33] Zhu Z, Peng S, Larsson V, Xu W, Bao H, Cui Z, et al. Nice-slam: Neural implicit scalable encoding for slam. In: Proc. IEEE conf. computer vision and pattern recognition. 2022, p. 12786–96.

[34] Burt P, Adelson E. The Laplacian pyramid as a compact image code. IEEE Trans Commun 1983;31(4):532–40. http://dx.doi.org/10.1109/TCOM.1983.1095851.

[35] Crete F, Dolmiere T, Ladret P, Nicolas M. The blur effect: perception and estimation with a new no-reference perceptual blur metric. In: Human vision and electronic imaging XII, vol. 6492. 2007, p. 64920I.

[36] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: Proc. IEEE int. conf. computer vision. IEEE; 1998, p. 839–46.

[37] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. In: European conference on computer vision. 2006, p. 404–17.

[38] Besl PJ, McKay ND. Method for registration of 3-D shapes. In: Sensor fusion IV: control paradigms and data structures, vol. 1611. 1992, p. 586–606.

[39] Chen Y, Medioni G. Object modelling by registration of multiple range images. Image Vision Comput 1992;10(3):145–55.

[40] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95.

[41] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vision 2004;60(2):91–110.

[42] Umeyama S. Least-squares estimation of transformation parameters between two point patterns. IEEE Trans Pattern Anal Mach Intell (PAMI) 1991;13(04):376–80.

[43] Park J, Zhou Q-Y, Koltun V. Colored point cloud registration revisited. In: Proc. IEEE int. conf. computer vision. 2017, p. 143–52.

[44] Farnebäck G. Two-frame motion estimation based on polynomial expansion. In: Proc. scandinavian conf. image analysis. 2003, p. 363–70.

[45] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process (TIP) 2004;13(4):600–12.
[46] Curless B, Levoy M. A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques. 1996, p. 303–12.
[47] Sturm J, Engelhard N, Endres F, Burgard W, Cremers D. A benchmark for the evaluation of RGB-D SLAM systems. In: Proc. of the international conference on intelligent robot systems. 2012.
[48] Nießner M, Zollhöfer M, Izadi S, Stamminger M. Real-time 3D reconstruction at scale using voxel hashing. ACM Trans Graphics 2013;32(6):169.