



HAL
open science

Fine Dense Alignment of Image Bursts through Camera Pose and Depth Estimation

Bruno Lecouat, Yann Dubois de Mont-Marin, Théo Bodrito, Julien Mairal,
Jean Ponce

► **To cite this version:**

Bruno Lecouat, Yann Dubois de Mont-Marin, Théo Bodrito, Julien Mairal, Jean Ponce. Fine Dense Alignment of Image Bursts through Camera Pose and Depth Estimation. 2023. hal-04337706

HAL Id: hal-04337706

<https://hal.science/hal-04337706v1>

Preprint submitted on 12 Dec 2023

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

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



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Fine Dense Alignment of Image Bursts through Camera Pose and Depth Estimation

Bruno Lecouat^{1,2,*} Yann Dubois de Mont-Marin^{2,*} Théo Bodrito^{2,*}
Julien Mairal⁴ Jean Ponce^{2,3}

Abstract

This paper introduces a novel approach to the fine alignment of images in a burst captured by a handheld camera. In contrast to traditional techniques that estimate two-dimensional transformations between frame pairs or rely on discrete correspondences, the proposed algorithm establishes dense correspondences by optimizing both the camera motion and surface depth and orientation at every pixel. This approach improves alignment, particularly in scenarios with parallax challenges. Extensive experiments with synthetic bursts featuring small and even tiny baselines demonstrate that it outperforms the best optical flow methods available today in this setting, without requiring any training. Beyond enhanced alignment, our method opens avenues for tasks beyond simple image restoration, such as depth estimation and 3D reconstruction, as supported by promising preliminary results. This positions our approach as a versatile tool for various burst image processing applications.

1. Introduction

This paper tackles the challenge of dense alignment in burst photography, a domain characterized by minimal camera movement and predominantly static scenes. We aim to align these image sequences accurately, quickly, and reliably.

Burst photography is increasingly pivotal in a range of image enhancement applications, as evidenced by recent advancements in high dynamic range imaging [23, 29], night photography [31], deblurring [14], or super-resolution [6, 28, 41]. In this context, a handheld camera captures a rapid sequence of images with slightly different viewpoints due to hand tremor, possibly with varying camera settings, over a brief duration. The alignment of these frames is a

critical precursor for these methods. However, current approaches to image registration between image pairs, such as homography or optical flow estimation, do not fully leverage the nature of burst sequences (multiple views of a quasi-static three-dimensional scene with slight camera motion). This limitation potentially leads to suboptimal outcomes. Precision in alignment is crucial for the quality of the enhanced images, and inaccuracies can significantly impair the final results, introducing artifacts like ghosting or zipping [28].

In this paper, instead of relying on traditional pairwise dense alignment of frames, we propose a novel global estimation approach tailored for image bursts, which explicitly considers the three-dimensional nature of the scene. Specifically, our approach takes full advantage of the small baseline feature by introducing a new parametrization of optical flows, consistent across different views, based on the image formation model. This model assumes a perspective camera with known intrinsic parameters, capturing a static scene comprising surfaces approximated as small planar patches. Given the small baseline, we anticipate minimal occlusions between views. Consequently, we simplify the representation of the 3D scene into a concise two-dimensional grid that encodes the depth and normals of these planar surfaces.

More precisely, our method employs a 2D grid to represent depth, normals from a reference view, and camera poses. While camera poses are optimized individually for each frame, structural parameters are shared across all views. This shared parameterization requires fewer parameters than traditional pairwise optical flow methods. It enhances the overall consistency and effectiveness of our alignment method while still preserving the expressivity necessary for accurately modeling motion induced from 3D scenes.

In practice, we solve a global optimization problem to align frames, minimizing patch photometric reprojection errors across all views within the reference frame. Optimizing for camera pose, depth, and normal parameters. In situations with parallax, our model adapts to determine camera motion and scene geometry that accounts for the relative movements between frames. When no parallax effects are present, the model defaults to fitting pose parameters for each frame with constant depth, similar to homography fitting.

* Authors marked with an asterisk (*) contributed equally to this work. This project started when B. Lecouat was a PhD student at Inria Paris.

¹Enhance Lab.

²Inria and DIENS (ENS-PSL, CNRS, Inria).

³Courant Institute and Center for Data Science, New York University.

⁴Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK.

⁵Corresponding author: bruno.lecouat@enhancelab.fr

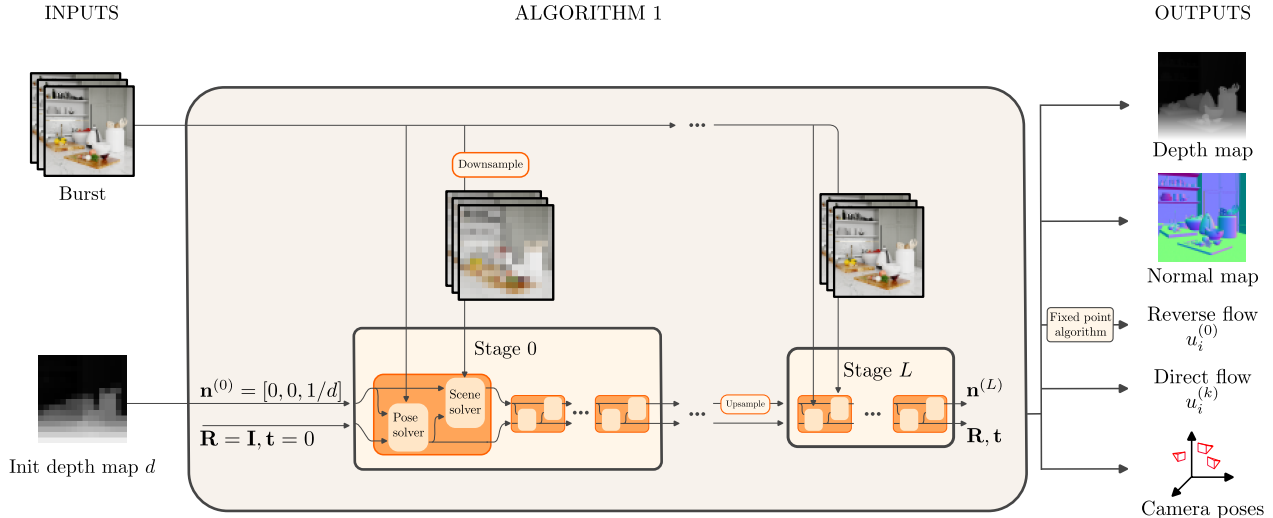


Figure 1. The global pipeline of our optimization-based method. It inputs a burst of images and an initialization depth map and outputs the direct and reverse flow between each image and the first one. Our method estimates the optical flow using the camera’s pose and 3D scene structure as optimization variables of the photometric reprojection errors in a reference frame then poses and depth maps can also be retrieved.

To achieve efficient optimization, we propose a new coarse-to-fine block-coordinate descent algorithm inspired by the parametric Lucas-Kanade algorithm [5] in its structure, using a variant of the Gauss-Newton algorithm for precise pose optimization on $SO(3)$ and gradient descent for depth and normal adjustments. We also introduce a novel fixed-point algorithm to infer depth maps for new camera positions. This algorithm is particularly advantageous for our specific needs but also holds potential for broader applications. It enables us to calculate reverse optical flows and adapt reference views to other views, which is essential for downstream tasks like super-resolution and low-light photography and can also be used to detect occlusions.

We validate our approach with synthetic bursts built with photorealistic rendering software. To validate our approach with real-world data, we also demonstrate applications with real bursts captured with a Pixel 6 pro smartphone to night photography denoising and super-resolution. Quantitative and qualitative experiments with synthetic and real data show that our method consistently gives accurate registration results even when little or no parallax is present and consistently outperforms the state-of-the-art in the burst setting, outperforming learning-based methods for flow estimation such as RAFT [40].

Beyond flow estimation, our model demonstrates exceptional versatility and efficiency in small baseline scenarios. It not only achieves convergence in pose and depth to meaningful values but also surpasses specialized methods in these areas. Distinct from conventional 3D methods that typically separate camera pose estimation and dense reconstruction into different steps, our method directly tackles dense optimization within a joint estimation framework.

Essentially, our approach acts as a multifunctional tool in burst photography, with the dual capability to accurately estimate flow and precisely determine depth and pose. It is helpful across a wide range of downstream tasks and sets a new benchmark for processing small motion scenes—characterized by its simplicity, accuracy, and robustness.

Contributions. Below, we summarize our key contributions, highlighting how our approach serves as a versatile tool applicable to various burst image processing tasks:

1. **State-of-the-art dense alignment for burst imagery:** we propose a novel optimization algorithm that outperforms deep-learning methods in dense alignment. This precision is especially useful for tasks requiring fine alignments, like burst super-resolution.
2. **Accurate pose and depth estimation in small motion:** our algorithm provides state-of-the-art camera pose and depth estimation results in scenarios with minimal motion, effectively capturing 3D scene structures from bursts with small baselines. This performance is achieved where standard SFM methods such as COLMAP [37] struggle.
3. **Novel fixed-point algorithm for depth inference:** we propose a new fixed-point algorithm for deducing depth maps at novel camera positions, enhancing our method’s utility in reversing optical flows and warping reference views onto other views, with potential applications beyond the scope of this paper.

2. Related work

Burst photography. Burst photography is a technique that involves capturing a sequence of images to improve the overall quality of a photograph by reducing noise [23], enhancing details [6, 7, 16, 28, 32, 33, 41], and improving dynamic

range [29]. Traditionally, algorithms for burst photography rely on a registration step to align frames [15].

Recent advancements explore machine learning, specifically deep learning, for burst photography, often eliminating the need for traditional registration [16, 32]. However, many such algorithms are based on supervised learning, demanding paired datasets of degraded raw bursts and high-quality sRGB images for training. The reliance on simulated raw bursts generated from ground truth sRGB images introduces a potential mismatch between training and real-world data distributions. Real-world bursts may exhibit different degradations or involve a sensor mismatch, leading to artifacts [8]. Self-supervised learning methods [8, 34] have emerged to address these issues. Furthermore, the computational demands of deep learning models pose challenges for integration into embedded devices [15], limiting their practical utility under constraints of limited resources.

In contrast, we present an efficient approach to image alignment specifically designed for burst image data. This approach does not rely on machine learning and can serve as a versatile tool in various burst processing applications, whether they involve learning-based components or not.

Multi-frame image registration.

A straightforward method for image alignment in burst photography involves aligning frames with a reference frame, as demonstrated in [23, 41]. Some works have explored the multi-view setting to enhance registration quality, such as [3, 4, 19], which introduced various optimization-based approaches for multi-view image registration. However, these approaches are limited to simple motion models, such as translations. In contrast, our method is more general and takes into account the three-dimensional nature of the scene.

Depth reconstruction from small motions. Popular 3D reconstruction methods rely on geometric approaches such as structure from motion (SfM) [37]. These methods use geometric constraints and depend on keypoint correspondences to reconstruct a sparse 3D scene. Subsequently, dense 3D representations can be estimated based on the sparse reconstruction, as done by Colmap [37]. Bundle adjustment is a critical step for refining the estimated 3D structure and camera poses of a scene. This process involves optimizing 2D image keypoints, corresponding 3D points, and camera calibration parameters iteratively to minimize the reprojection error, leading to a more accurate scene reconstruction.

Several 3D reconstruction methods have been specifically tailored for scenarios involving small motions to reconstruct depth maps. For instance, Im et al. [24] have adapted SfM to small motion settings, whereas [21] have proposed an efficient method using feature tracking for pairwise key points and bundle adjustment algorithms adapted to small motions. Additionally, this method estimates the intrinsic parameters of the camera as well as distortion parameters to achieve a

better fit with the data. In a different approach, [12] introduces a neural depth model and uses an inertial measurement unit (IMU) and lidar measurements to respectively initialize camera poses and the depth map. Then, [11] eliminates the need to initialize with a depth map model, although initialization with such a model may still yield improved results.

In contrast, our method serves a different purpose than depth estimation, with our primary goal being accurate image alignment. As shown in Sec. 4, our dense depth estimation procedure is more suitable for this task than approaches based on bundle adjustment with sparse keypoints.

3. Method

The proposed method aims to robustly and accurately estimate the optical flow and its inverse between a reference image and other images within a burst sequence. Given the nature of a burst, where movements are small, this approach provides the opportunity to directly address the problem densely, in contrast to [21], which relies on prior sparse matching between pairs of views. Densely approaching the problem enables high flow accuracy compared to other existing methods. In order to address the problem both densely and robustly, the key idea of the method is to parameterize the flows for each view using a common dense map characterizing the scene in the reference view and the relative positions of the views with respect to the reference frame.

Our formation model is detailed in the first paragraph below, leading to the formulation of flow estimation by optimizing the dense structure map and the relative positions of the views. This optimization is achieved by minimizing the photometric reprojection error through the direct flow induced by the parameters, which is the loss that best characterizes the quality of the induced flow. The challenges of this minimization problem are outlined in the second paragraph. The minimization procedure uses a block coordinate descent between the dense structure map and the relative poses, described in the third paragraph. This approach stabilizes the optimization process.

It also enables a coarse-to-fine approach for the dense parameterization of the scene. Finally, our formation model also allows the calculation of inverse optical flow through a fixed-point algorithm, detailed in the fourth paragraph. The global pipeline is illustrated in Fig. 1.

Image formation model. We consider a rigid scene described by a piecewise surface and $K+1$ internally calibrated pinhole cameras $(C_k)_{k=0..K}$. A point \mathbf{u}_i in \mathbf{G} a regular grid of the C_0 camera plane, is the projection of a point \mathbf{x}_i of the scene surface. We denote by π_i , the affine plane tangent to the scene in \mathbf{x}_i parameterized, by its (non-unit) normal n_i such that $\pi_i = \{\mathbf{y} \in \mathbb{R}^3, \mathbf{n}_i^\top \mathbf{y} = 1\}$. A patch $P(\mathbf{u}_i)$ around \mathbf{u}_i is the projection of a patch around \mathbf{x}_i in π_i , and its image in the camera plane C_k is given by a homography uniquely

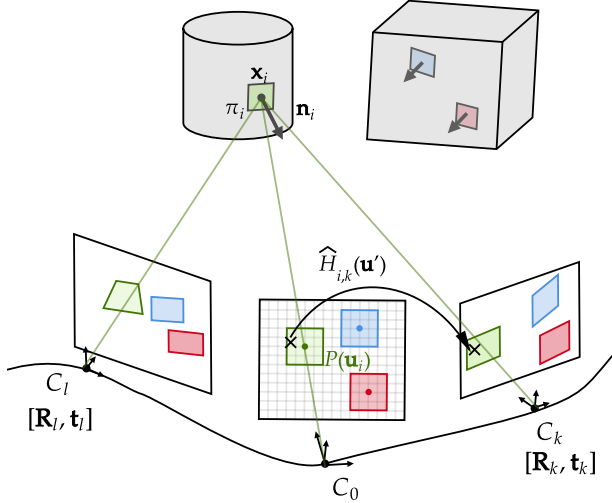


Figure 2. Image formation model with a patch and its local homography flow.

defined by the plane π_i and the extrinsic parameters $\mathbf{R}_k, \mathbf{t}_k$ of the other camera (Fig. 2). For \mathbf{u}' in the patch $P(\mathbf{u}_i)$, we have the direct flow locally expressed as a homography:

$$\hat{H}_{i,k}(\mathbf{u}') = \psi(H_{i,k}[\mathbf{u}', 1]^\top) \quad (1)$$

$$H_{i,k} = \mathbf{R}_k + \mathbf{t}_k \mathbf{n}_i^\top, \quad (2)$$

where $H_{i,k}$ is the homography matrix for the patch $P(\mathbf{u}_i)$ in the camera plane of C_k , $[\mathbf{u}', 1]$ is the homogeneous representation of \mathbf{u}' and $\psi : x, y, z \rightarrow x/z, y/z$ is the standard projection. The parameters of this flow are the non-unit normal \mathbf{n}_i characterizing the plane π_i and the pose $\mathbf{R}_k, \mathbf{t}_k$. It is important to note, as in [11, 22], that \mathbf{n}_i is not a homogeneous vector defined up to scale and has three full degrees of freedom. The formation model is summed up in Fig. 2.

Minimization problem. The parameters of our formation model is $n = (\mathbf{n}_i)_{i \in G}$ the dense map over a regular grid G parametrizing the scene structure and $R, t = (\mathbf{R}_k, \mathbf{t}_k)_{k=1..K}$ the pose parameters of each C_k relative to C_0 . As the objective is to estimate the optical flow between view k and the reference view, we optimize the parameters $\mathbf{R}, \mathbf{t}, \mathbf{n}$ so that the flows derived from local homographies $H_{i,k}$ minimize the photometric reprojection error between the images I_0 and I_k . More specifically, we solve the minimization problem:

$$\min_{\mathbf{n}, \mathbf{R}, \mathbf{t}} \frac{1}{2} \sum_{k=1}^K \sum_{i \in G} \sum_{\mathbf{u}' \in P(\mathbf{u}_i)} \rho(|I_0(\mathbf{u}') - I_k(\hat{H}_{i,k}(\mathbf{u}'))|^2), \quad (3)$$

where ρ is a robust loss function as in [39]. Indeed, the formation model does not account for occlusion phenomena. When a pixel \mathbf{u}' in the C_0 plane is the projection of a point \mathbf{x} that is not visible in camera C_k , $\hat{H}_{i,k}(\mathbf{u}')$ is essentially the projection of another point \mathbf{y} that is not on the same

scene element as \mathbf{x} . Consequently, it is likely that the color $I_k(\hat{H}_{i,k}(\mathbf{u}'))$ deviates significantly from $I_0(\mathbf{u}')$. The function ρ reduces the importance of large values, effectively filtering out such cases.

Optimization procedure. As usual in structure from motion literature [22], there is a global scale ambiguity since for every $\lambda > 0$, jointly replacing \mathbf{t} by $\lambda \mathbf{t}$ and \mathbf{n} by $1/\lambda \mathbf{n}$ does not change the homography matrices $H_{i,k}$ and nor the loss. To prevent this ambiguity from hindering the convergence of the optimization procedure, our algorithm relies on a *block coordinate descent* that alternates between steps on the plane map \mathbf{n} and steps on the relative poses \mathbf{R}, \mathbf{t} . Indeed, when \mathbf{n} is fixed, there is no longer any ambiguity about the value that \mathbf{t} can take, and the same applies to \mathbf{n} when \mathbf{t} is fixed. Gradually, the ambiguity boils down to the scale induced by the parameters' initialization. In addition, the optimization problem (3) is not convex and a good initialization is crucial to enable the algorithm's convergence. In the case of small movements, it is reasonable to initialize the pose with $\mathbf{R}_k = I$ and $\mathbf{t}_k = 0$. Therefore, it is necessary to have a good initialization of the plane parameters. Our method relies on an initialization based on a very coarse and low-resolution estimation of the scene depth in the reference image. Starting from a depth map $\mathbf{z} = (z_i)_{i \in G^{(0)}}$ on a very low-resolution grid $G^{(0)}$ (typically 16×16), we can initialize the plane map as $n^{(0)} = ([0, 0, 1/z_i])_{i \in G^{(0)}}$. This corresponds to initializing the planes as fronto-parallel and located at a distance z_i from the reference camera. It is important to note that this initialization resolves the scale ambiguity and initializes in a good region, thereby avoiding certain local minima. However, it does not need to be extremely precise. As observed in Section 4, the performance of our method is minimally impacted by the quality of the initialization. This paper uses the smallest monocular network of shallow resolution from [36], which has negligible inference cost, to initialize the algorithm.

From this initialization, we adopt a *coarse-to-fine* strategy as in [30, 39] for optimizing the plan map. Specifically, we define a sequence $G^{(0)}, \dots, G^{(L)}$ of L regular grids, each twice as fine as the previous one, with $G^{(L)}$ having the same resolution as the burst $(I_k)_{k=0..K}$. We also denote $I_k^{(l)}$ as the downsampled version of the image I_k to the resolution of $G^{(l)}$. Our optimization strategy is as follows:

- We perform the steps for poses \mathbf{R}, \mathbf{t} using the high-resolution grid $G^{(L)}$, a linear interpolation of the current estimate of the plane map $\mathbf{n}^{(l)}$ to the resolution of $G^{(L)}$, and using the high-resolution images $(I_k)_{k=0..K}$. For these steps, we employ a proximal Gauss-Newton algorithm tailored to the fact that rotation matrices belong to the Lie group $SO(3)$ and the minimization problem (3) is a robust nonlinear least squares problem [25]. The small number of variables (six times the number of images) makes the computation of the required Jacobians

tractable. Details about the Gauss-Newton step and the closed form of the Jacobians are provided in Appendix A. Using the Lie group exponential representation of rotation and employing a second-order optimization method are crucial elements of our method for achieving high precision. We empirically show the advantages of these choices in the ablation study presented in Appendix E.

- We perform the steps on the plane map parameters $\mathbf{n}^{(l)}$ at scale l using the gradient descent variation **Adam** [26], with the loss calculated using the grid $\mathbf{G}^{(l)}$, \mathbf{R} , \mathbf{t} , and the images at resolution l : $(I_k^{(l)})_{k=0..K}$. Using a method with moments like Adam accelerates the convergence of the procedure.
- Every few alternate steps on \mathbf{R} , \mathbf{t} on one side and $\mathbf{n}^{(l)}$ on the other side, we double the resolution of the plane map $\mathbf{n}^{(l)}$ and move to the next scale with $\mathbf{n}^{(l+1)}$.

The procedure is summarized in the pseudocode in Algorithm 1. In the case of a dense approach, a coarse-to-fine strategy is crucial. Since we use a photometric loss, the gradients and Jacobians depend on the spatial gradients of the I_k images and contain only sub-pixel information. When the alignment error is larger than one pixel, this can lead to convergence issues. At lower scales of the coarse-to-fine approach, pixels cover a larger area, allowing us to benefit from the information. As we move to higher scales, we increase the precision we aim to achieve. Finally, note that the original minimization problem is properly solved during the last stage of the coarse-to-fine approach. The previous stages can be interpreted as a procedure to generate the right initialization for the original minimization problem.

Algorithm 1 Multiscale block coordinate descent

Require: $L \geq 0, N \geq 0, \beta$
Require: $I = (I_k)_{k=0..K}, d$ \triangleright Burst and 16×16 resolution depth map

- 1: $d \leftarrow \text{Mono}(I_0)$ \triangleright Low resolution monocular depth estimation
- 2: $n \leftarrow [0, 0, 1/d]$ \triangleright dimension 16×16
- 3: $R \leftarrow I$ $\triangleright K$ matrices 3×3
- 4: $t \leftarrow 0$ $\triangleright K$ vectors of size 3
- 5: $l \leftarrow 0$
- 6: **while** $l \leq L$ **do** \triangleright Multiscale loop
- 7: $n \leftarrow \text{interpolate}(n, 2 \times \text{resolution}(n))$ \triangleright Double resolution
- 8: $I_- \leftarrow \text{sample}(I, \text{resolution}(n))$ \triangleright Low resolution image
- 9: $m \leftarrow 0$
- 10: **while** $m \leq M$ **do** \triangleright Block descent
- 11: $n_+ \leftarrow \text{interpolate}(n, \text{resolution}(I))$ \triangleright High resolution
- 12: $R, t \leftarrow \text{PGN}(R, t, n_+, I)$ \triangleright Pose Newton step
- 13: $n \leftarrow \text{ADAM}(R, t, n, I_-)$ \triangleright Some steps with ADAM
- 14: $m \leftarrow m + 1$
- 15: **end while**
- 16: $l \leftarrow l + 1$
- 17: **end while**
- 18: **return** R, t, n

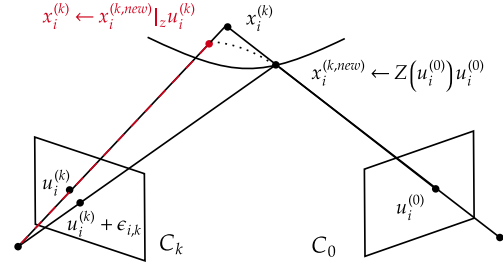


Figure 3. We have the depth map $(z_i^{(0)})_{i \in \mathbf{G}^{(L)}}$ in the reference view (C_0), with interpolation we construct Z that gives the depth for any \mathbf{u}_0 . We initialize a depth map $(z_i^{(k)})_{i \in \mathbf{G}^{(L)}}$ in view k with a copy of the depth map in view 0. Then, we use $z^{(k)}$ on the regular grid to induce a direct flow into the view 0 and query the depth $Z(F(\mathbf{u}_i^{(k)}))$. Reprojecting the obtained depth map gives the new iterate of $z^{(k)}$. At convergence, the direct flow induced by $z^{(k)}$ is the reverse flow from view 0 to view k .

Outputting the flows, poses, depth map and normal map.

After the convergence of the algorithm, we obtain \mathbf{R} , \mathbf{t} , and $\mathbf{n}^{(L)}$ that minimize the problem (3) on a grid of maximum resolution, thus minimizing the photometric error of the flow. We then obtain an estimation of the **direct flow** for each image. For $\mathbf{u}_i^{(0)} \in \mathbf{G}^{(L)}$:

$$\mathbf{u}_i^{(k)} = \hat{H}_{i,k}(\mathbf{u}_i^{(0)}) = \psi(\mathbf{R}_k \mathbf{u}_i^{(0)} + 1/z_i^{(0)} \mathbf{t}_k), \quad (4)$$

where, from $\mathbf{n}^{(L)}$, we recover a high-resolution depth map in the reference view: $z_i^{(0)} = 1/([\mathbf{u}_i^{(0)}, 1]^\top \mathbf{n}^{(L)})$. Normalizing $\mathbf{n}^{(L)}$ also provides a high-resolution normal map. Finally, our algorithm directly estimates the camera poses \mathbf{R} , \mathbf{t} . For certain applications, such as super-resolution, we need the inverse flow, i.e., $\mathbf{u}_i^{(0)} = F(\mathbf{u}_i^{(k)})$. The flow inverse is generally unstable, so PyTorch [35] does not implement the forward warp. As the movements are small for a burst, the depth map in view C_k will be close to that in view C_0 . Moreover, the depth map in view C_k allows generating a direct flow $\mathbf{u}_i^{(0)} = F(\mathbf{u}_i^{(k)})$ as in (4) using the inverse poses: $[\mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k]$. So, if the depth map is correct, we should obtain the identity by composing with the initial direct flow. This allows designing a fixed-point algorithm presented in Fig. 3 and detailed in Appendix B. Pixels for which the fixed point does not converge correspond to the occluded pixel, and the occlusion masks are presented in Appendix G.

4. Experiments

We conduct experiments on synthetic bursts and showcase practical applications using real bursts captured with a Pixel 6 Pro smartphone. These applications include night photography and 3D reconstruction, serving as proof of concept. Additionally, we have included preliminary experiments on burst super-resolution in Appendix K.

Method	EPE	RMSE	NPE1	NPE2	NPE3	EPE	RMSE	NPE1	NPE2	NPE3
	↓	↓	↑	↑	↑	↓	↓	↑	↑	↑
	Blender 1 (small motion)					Blender 2 (micro motion)				
DfUSMC [21] *	1.4466	2.1723	0.5315	0.7488	0.8477	4.1356	4.5676	0.2267	0.4278	0.5497
RCVD [27]*	5.9556	7.678	0.0957	0.2534	0.3763	0.4007	0.5316	0.8676	0.9825	0.9959
Saop [11] *	9.7262	12.5891	0.101	0.2457	0.3402	2.0430	2.3563	0.5684	0.7645	0.8424
Homography	2.8102	4.7107	0.4998	0.6627	0.7405	<u>0.3008</u>	<u>0.3772</u>	<u>0.9003</u>	<u>0.9921</u>	<u>0.9982</u>
Farnebäck [17]	2.6852	4.8478	0.5299	0.6612	0.7278	2.0892	3.8154	0.6480	0.7296	0.7642
RAFT [40]	<u>0.9013</u>	<u>1.5396</u>	<u>0.7348</u>	<u>0.9069</u>	<u>0.9443</u>	0.4857	0.5765	0.8664	0.9857	0.9963
Ours	0.7439	1.4324	0.7841	0.9084	0.9456	0.2321	0.2820	0.9366	0.9972	1.0000

Table 1. Optical flow errors. The optical flow was predicted from the extrinsic camera parameters and depth maps for the models marked with an asterisk.

Method	Pose				Depth					
	Left 12 (m)↓	ATE (m)↓	Geom (m)↓	Biinvrot 12 (deg)↓	Abs rel ↓	Sqr rel ↓	RMSE↓	Delta 1↑	Delta 2 ↑	Delta 3 ↑
Dataset	Blender 1 (small motion)									
Colmap [37]	✗				✗					
DfUSMC[21]	<u>0.0117</u>	<u>0.0108</u>	<u>0.0094</u>	<u>0.1948</u>	<u>0.2107</u>	<u>0.4864</u>	<u>0.9683</u>	<u>0.7723</u>	<u>0.8877</u>	0.9409
Saop [11]	0.0274	0.0229	0.0204	0.6369	0.5818	1.8768	1.7900	0.3958	0.6009	0.7198
RCVD [27]	0.0168	0.0162	0.0140	0.2158	0.3111	0.5382	1.2368	0.5294	0.814	<u>0.9524</u>
Ours	0.0066	0.0056	0.0050	0.1806	0.1381	0.2391	0.8688	0.8358	0.9263	0.9761
Dataset	Blender 2 (micro motion)									
Colmap [37]	✗				✗					
DfUSMC[21]	<u>0.0046</u>	<u>0.0026</u>	<u>0.0024</u>	<u>0.1918</u>	0.3093	0.9543	2.0499	0.5722	0.7785	0.9187
Saop [11]	0.0078	0.0043	0.0040	0.2678	0.2936	0.8326	2.0020	0.5794	0.7976	0.9263
RCVD [27]	0.0168	0.0162	0.0140	0.2158	<u>0.1898</u>	<u>0.3492</u>	<u>1.3745</u>	<u>0.6726</u>	<u>0.8816</u>	<u>0.9693</u>
Ours	0.0022	0.0022	0.0020	0.0245	0.1383	0.1962	1.1521	0.7996	0.9819	0.9983

Table 2. Pose and depth errors metrics on the two proposed synthetic bursts datasets.

Synthetic burst simulation. We require photorealistic bursts containing ground truth depth and camera poses for evaluating our approach and concurrent methods, but existing public multi-view stereo datasets we are aware of lack the needed characteristics due to non-static scenes or excessively large frame baselines that do not align with our specific use cases.

We generated two photorealistic synthetic datasets using CYCLES, the path tracing engine of Blender [13]. We used a set of twelve publicly available indoor scenes made by 3D artists, with detailed and varied scene compositions.

Ten scenes come from [1], and two scenes are from [2]. Each burst of the dataset consists of 20 frames, with a resolution of 512x512 pixels.

We skipped the post-processing denoising step at the end of the rendering to avoid temporal flickering artifacts and mitigated render noise by using a large number of samples (4096). The camera trajectories and orientations are crafted as follows: a few keyframes was positioned manually to out-

line the global path, and the other keyframes were obtained with Bezier interpolation.

We generated two datasets: Blender 1 with small baselines and Blender 2 with micro-baselines. The first dataset exhibits larger parallax effects, while the second dataset has reduced parallax effects. Detailed characteristics of these datasets are provided in the Appendix D.

Evaluation on synthetic data. We initialize our algorithm on synthetic data with a 16×16 coarse depth map using the shallow network from [36]. For the evaluation, we follow the standard practice to evaluate pose, depth, and flow, described in [20, 27]. For all the methods, as depth estimation and pose are known up to an unknown scale, we align the predicted depth and the ground truths using median scaling. For pose evaluation, we compute the scale factor as $s = \arg \min_s \|T - s\hat{T}\|^2$, where $T = [t_0, \dots, t_N]$. In addition, we use the canonic left-invariant distance in $SE(3)$ that combines rotational and translation parts in one quantity; see [10, 42] for details. We report the distance

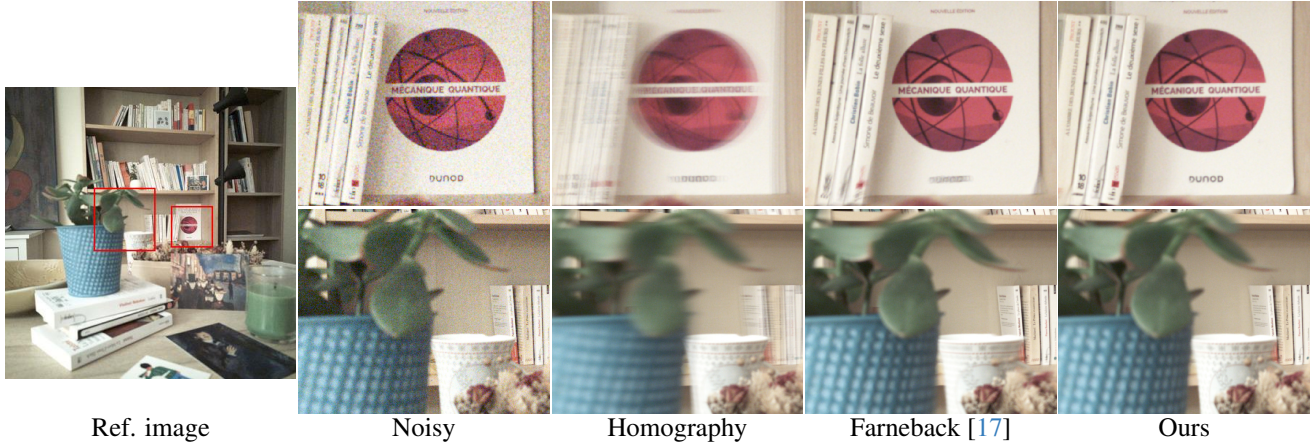


Figure 4. Burst denoising for night photography on real bursts exploiting alignment of various algorithms. Left: Full image with bounding boxes highlighting the region of interest. Top line: background region is misaligned for concurrent methods. Bottom line: The cup is misaligned for other methods. Homography misaligned the plant as well. It is best seen by zooming on a computer screen.

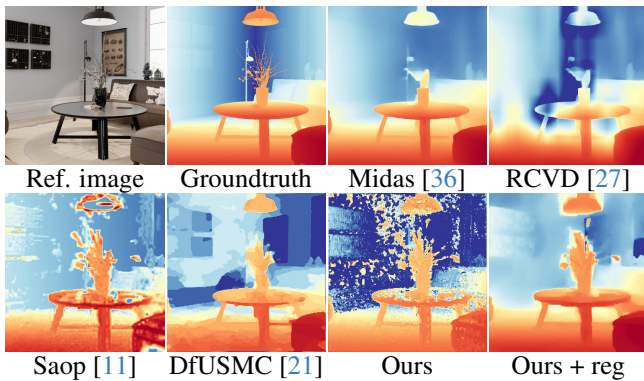


Figure 5. Depth estimation from a synthetic image burst. It is one of the scenes generated with Blender used in the dataset *Blender 2*. We present our result w/o regularisation (*Ours*) and with determinant penalization (*Ours + reg*) for smoother results, see the Appendix C.

between the ground truth pose and the estimated pose. It reads $d([R, t], [R', t'])^2 = \|t' - t\|_2^2 + \lambda \|\log(R^\top R')\|_2^2$. For λ , we use the median value of the ground truth depth. $\|\log(R^\top R')\|_2$ is the canonic metric on the set of rotation $SO(3)$ and is also reported independently. Unlike other methods in the literature [27], we choose not to present relative pose error (RPE) as a good RPE may not correlate with good alignment metrics and rely on a time coherent burst. To evaluate the ATE, we did not align the estimated poses with the ground truth poses with rigid transformation, as is common in the SLAM community. Indeed, our loss 3 and, more generally, the flow is not invariant by a solid transformation of the poses. As the final goal of our method is alignment, performance evaluation up to a rigid transformation would not be informative.

For optical flow evaluation, we conducted comparisons on our synthetic datasets. We utilized a state-of-the-art deep

optical flow method [40] by registering all frames pairwise with a reference. Additionally, we employed a standard homography and the Farneback optical flow [17] for comparison. Furthermore, we computed optical flow errors for other concurrent methods [11, 21, 27] using the camera projection model as in Eq. (4) and their estimated pose and depth maps. Leveraging the assumption of a static scene, our method consistently outperformed [40] regarding flow accuracy.

We conducted comparisons of our pose and depth estimation method with methods introduced in [27], [11], and [21], utilizing publicly available codebases. To ensure a fair comparison, we initialized the method from [11] with the same depth map as the one we used for our own initialization.

We compare our method with a monocular depth estimation model Midas [36]. However, monocular methods estimate depth up to an affine transformation, whereas flow estimation is not invariant by affine reparameterization. Using affine registration lacks full relevance to evaluate the quality of the result, so the performances in Table 2 are obtained after rescaling only. For a fair depth map comparison, we also evaluate our method and others against Midas with an affine registration. Results are presented in Appendix F.

3D reconstructions quality on synthetic data and real bursts.

We evaluate qualitatively our depth reconstructions on synthetic data from our dataset and real bursts captured with a Pixel 6 Pro smartphone.

Visualizations of reconstructed depth maps are provided in Figure 5. Our depth map can have a noisy aspect on a texture-less structure. This is a normal feature as our optimization is not well conditioned on uniform surfaces, as small variations in inferred depth will not affect the reprojection photometric loss. This noisy effect can be mitigated by adding spatial regularization for the scene steps. But this

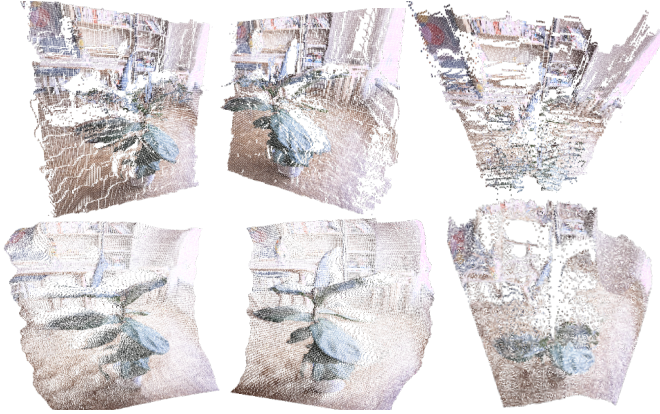


Figure 6. **Top** : point cloud reconstruction with DfUSMC [21]. **Bottom** : point cloud reconstructed with our method. We show respectively left, right, and top views of the two point clouds.

trades with lower performance in terms of flow and pose metrics on synthetic data. We observed that no spatial regularization parameters give the best image alignment and pose estimation results. We detail the spatial regularization in Appendix C.

For real scenes, we showcase the high-quality depth reconstructions achievable with our method in Figure 7. We input RAW image bursts from the Pixel 6 Pro smartphone, and perform demosaicking using bilinear filtering. We initialize our algorithm with a low-resolution depth map from the phone sensor. We compare our results with depth maps obtained from a monocular method [36], RCVD [27], Saop [11], and DfUSMC [21]. Furthermore, we provide visualizations of reconstructed point clouds in Figure 6.

Low-light photography on real bursts.

To demonstrate the robustness and accuracy of our alignment method for downstream tasks, we conducted a low-light photography experiment as a proof of concept. This scenario is challenging as it involves aligning frames with a low signal-to-noise ratio. We captured night bursts using a Pixel 6 Pro smartphone under low light conditions, using a short exposure time and high ISO settings to reduce motion blur. We aligned these frames using our method and other concurrent alignment algorithms, including a simple homography and dense optical flow using the Farneback implementation from OpenCV [9].

To reduce noise, we averaged the aligned frames, using a straightforward denoising approach. While our focus was on highlighting the registration quality of our method, it’s worth noting that a more sophisticated fusion algorithm could be employed to enhance image quality and reduce artifacts, as seen in previous works [23, 31].

In Figure 4, we provide visual comparisons of our results. We observed that due to the nonplanar nature of the scene, the homography-based approach failed to align objects in the

foreground and background, resulting in a blurry appearance in the denoised image. In contrast, the optical flow model exhibited greater flexibility, successfully aligning objects in both the foreground and background. However, some elements, such as the white book in the background or certain patterns on the white cup in the foreground, were still not perfectly aligned.

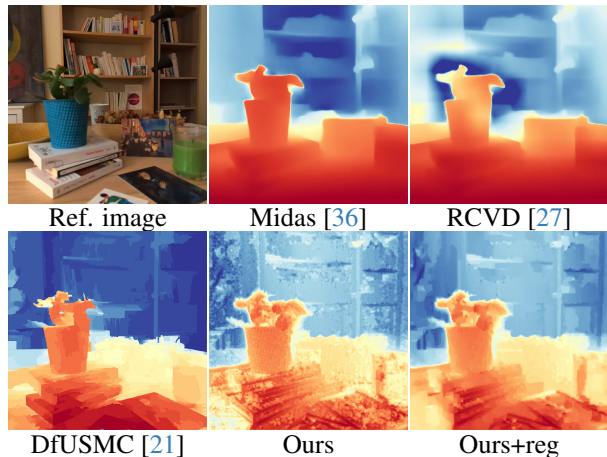


Figure 7. Depth estimation from a real burst. We present our result w/o regularisation (*Ours*) and with determinant penalization (*Ours* + *reg*) for smoother results; see Appendix C for more details.

Depth initialization. Figure 8 shows the impact of the depth map’s initialization on our method’s performance. We gradually increase the variance of a Gaussian random noise added to the 16×16 initialization depth map and evaluate the performance of our algorithm on our synthetic dataset with various depth, pose, and alignment metrics. This experiment demonstrates that our method is robust to noise on the initialization depth map. The model only requires a noisy estimate to converge to the right solution.

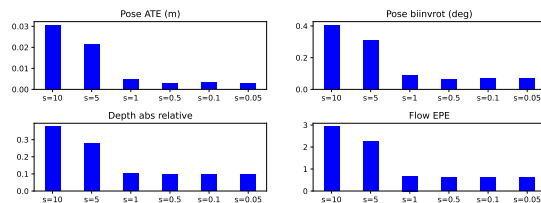


Figure 8. Noise on the initialization depth map. Our method is robust to noise; the performance degrades when the noise’s variance is larger than 1 meter.

5. Conclusion

Our approach offers a comprehensive and versatile solution for burst photography. It excels in accurately estimating flow, depth, and pose, setting a new benchmark for processing

small motion scenes. Future enhancements include integrating intrinsic camera parameter estimation like focal length to improve accuracy, refining our model for lens-induced distortions, and exploring more advanced camera models such as thin-lens to account for defocus effects.

Acknowledgments

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). JM was supported by the ERC grant number 101087696 (APHE-LAIA project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003). JP was partly supported by the Louis Vuitton/ENS chair in artificial intelligence and a Global Distinguished Professor appointment at the Courant Institute of Mathematical Sciences and the Center for Data Science of New York University.

References

- [1] Evermotion Archinteriors vol.43. https://evermotion.org/shop/show_product/archinteriors-vol-43/12555, . 6
- [2] Architecture topics. <https://www.youtube.com/watch?v=Gn1biEB5PbQ>, . 6
- [3] Cecilia Aguerrebere, Mauricio Delbracio, Alberto Bartesaghi, and Guillermo Sapiro. Fundamental limits in multi-image alignment. *IEEE Transactions on Signal Processing*, 64(21): 5707–5722, 2016. 3
- [4] Cecilia Aguerrebere, Mauricio Delbracio, Alberto Bartesaghi, and Guillermo Sapiro. A practical guide to multi-image alignment. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1927–1931. IEEE, 2018. 3
- [5] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56:221–255, 2004. 2
- [6] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9209–9218, 2021. 1, 2
- [7] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2460–2470, 2021. 2
- [8] Goutam Bhat, Michaël Gharbi, Jiawen Chen, Luc Van Gool, and Zhihao Xia. Self-supervised burst super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10605–10614, 2023. 3
- [9] Gary Bradski. The opencv library. *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 8
- [10] Gregory S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 1*. Birkhäuser Boston, 2009. 6
- [11] Ilya Chugunov, Yuxuan Zhang, and Felix Heide. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. *arXiv preprint arXiv:2212.12324*, 2022. 3, 4, 6, 7, 8, 13, 16
- [12] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Xuaner Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2852–2862, 2022. 3, 16
- [13] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 6
- [14] Mauricio Delbracio and Guillermo Sapiro. Burst deblurring: Removing camera shake through fourier burst accumulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2385–2393, 2015. 1
- [15] Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *Annual Review of Vision Science*, 7:571–604, 2021. 3
- [16] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burstformer: Burst image restoration and enhancement transformer. 2023. 2, 3
- [17] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer, 2003. 6, 7
- [18] Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13(10):1327–1344, 2004. 16
- [19] Sina Farsiu, Michael Elad, and Peyman Milanfar. Constrained, globally optimal, multi-frame motion estimation. In *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, pages 1396–1401. IEEE, 2005. 3
- [20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 6
- [21] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5413–5421, 2016. 3, 6, 7, 8, 16
- [22] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 4
- [23] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 1, 2, 3, 8
- [24] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. High quality structure from small motion for rolling shutter cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 837–845, 2015. 3

- [25] Stephen J. Wright Jorge Nocedal. *Numerical Optimization*. Springer New York, 2006. 4
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [27] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 6, 7, 8, 14, 16
- [28] Bruno Lecouat, Jean Ponce, and Julien Mairal. Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2, 16, 18
- [29] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. High dynamic range and super-resolution from raw image bursts. *ACM Transactions on Graphics*, 41(4), 2022. 1, 3
- [30] Cheng Lei and Yee-Hong Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1562–1569, 2009. 4
- [31] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T Barron, Dillon Sharlet, Ryan Geiss, et al. Handheld mobile photography in very low light. *ACM Trans. Graph.*, 38(6):164–1, 2019. 1, 8
- [32] Ziwei Luo, Youwei Li, Shen Cheng, Lei Yu, Qi Wu, Zhihong Wen, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Bsr: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 998–1008, 2022. 2, 3
- [33] Nancy Mehta, Akshay Dudhane, Subrahmanyam Murala, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Gated multi-resolution transfer network for burst restoration and enhancement. *arXiv preprint arXiv:2304.06703*, 2023. 2
- [34] Ngoc Long Nguyen, Jérémy Anger, Axel Davy, Pablo Arias, and Gabriele Facciolo. Self-supervised multi-image super-resolution for push-frame satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1121–1131, 2021. 3
- [35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5
- [36] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 4, 6, 7, 8, 13, 14, 16
- [37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2, 3, 6
- [38] Joan Solà, Jérémy Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *CoRR*, abs/1812.01537, 2018. 11
- [39] Javier Sánchez. The Inverse Compositional Algorithm for Parametric Registration. *Image Processing On Line*, 6: 212–232, 2016. <https://doi.org/10.5201/ipol.2016.153>. 4
- [40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2, 6, 7
- [41] Bartłomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (ToG)*, 38(4):1–18, 2019. 1, 2, 3
- [42] Ernesto Zacur, Matias Bossa, and Salvador Olmos. Left-invariant riemannian geodesics on spatial transformation groups. *SIAM Journal on Imaging Sciences*, 7(3):1503–1557, 2014. 6

Appendix

A. Closed form Jacobean for Gauss-Newton step

From Eq. (3), we recall that the residual of the robust least square for which we have to compute the Jacobian is the flat vector \mathbf{r} with coordinates indexed by $k = 1..K$, $i \in \mathbf{G}$, $\mathbf{u}' \in P_i$ with total dimension noted KD with $D = \#(\mathbf{G})\#(P)$ size of grid by size of patch:

$$r_{k,i,\mathbf{u}'} = I_0(\mathbf{u}') - I_k(\hat{H}_{i,k}(\mathbf{u}')). \quad (5)$$

We note the twist ξ_k such that $\mathbf{R}_k, \mathbf{t}_k = \text{Exp}(\xi_k)$. We want to find the Jacobian \mathbf{J}_r with the variable flat vector $\xi = (\xi_k)_{k=1..K}$. Then we note that \mathbf{J}_r is of dimensions KD, K and that \mathbf{J}_r is diagonal by K block of dimension D, K . We note \mathbf{J}_k these blocks. Using the expression of the homography matrix in (2), the block \mathbf{J}_k have D rows of the form $(\nabla\phi_{k,i,\mathbf{u}'})^\top$ where:

$$\phi_{k,i,\mathbf{u}'}(\xi) = I_k(\psi((\mathbf{R} + \mathbf{t}\mathbf{n}_i^\top)\mathbf{u}')) \quad (6)$$

$$= I_k\left(\psi\left(\mathbf{R}\frac{1}{\mathbf{n}_i^\top\mathbf{u}'}\mathbf{u}' + \mathbf{t}\right)\right), \quad (7)$$

with $\psi(x, y, z) = [x/z, y/z]^\top$ and $\mathbf{R}, \mathbf{t} = \text{Exp}(\xi)$. So if we note $\mathbf{X}_{i,\mathbf{u}'} = \frac{1}{\mathbf{n}_i^\top\mathbf{u}'}\mathbf{u}'$ in \mathbb{R}^3 and $\Lambda_{\mathbf{X}} : SE(3) \rightarrow \mathbb{R}^3$ the action on $\mathbf{X} \in \mathbb{R}^3$ that takes an element of $[\mathbf{R}, \mathbf{t}]$ in $SE(3)$ and gives its action on \mathbf{X} : $\Lambda_{\mathbf{X}}(\mathbf{R}, \mathbf{t}) = \mathbf{R}\mathbf{X} + \mathbf{t}$ we can simplify $\phi_{k,i,\mathbf{u}'}$ to a simple composition and compute its gradient with a chain rule:

$$\phi_{k,i,\mathbf{u}'} = I_k \circ \psi \circ \Lambda_{\mathbf{X}_{i,\mathbf{u}'}} \circ \text{Exp} \quad (8)$$

$$(\nabla\phi_{k,i,\mathbf{u}'})^\top = \nabla I_k^\top \mathbf{J}_\psi \mathbf{J}_{\Lambda_{\mathbf{X}_{i,\mathbf{u}'}}} \mathbf{J}_{\text{Exp}}. \quad (9)$$

Note that $\Lambda_{\mathbf{X}}$ takes input on the group $SE(3)$, and Exp has an output on the same group. However, as described in [38], using the so-called *left jacobian* suffices. ∇I_k is the spatial gradient of the image I_k calculated using a convolution and a Sobel kernel and evaluated in a coordinate using bilinear interpolation. The individual Jacobians are reported dropping indexes in Table 3.

In practice, exploit the diagonal structure of \mathbf{J}_r in our implementation.

Domains	Function	Jacobian
$\mathbb{R}^3 \rightarrow \mathbb{R}^2$	$\psi(x, y, z) = [x/z, y/z]^\top$	$\mathbf{J}_\psi = \frac{1}{z}[\mathbf{I}_2 \mid -\psi(x, y, z)]$
$SE(3) \rightarrow \mathbb{R}^3$	$\Lambda_{\mathbf{X}}(\mathbf{R}, \mathbf{t}) = \mathbf{R}\mathbf{X} + \mathbf{t}$	$\mathbf{J}_{\Lambda_{\mathbf{X}}} = [\mathbf{R} \mid -\mathbf{R}[\mathbf{X}]_\times]$
$\mathbb{R}^6 \rightarrow SE(3)$	$\text{Exp}(\xi)$ as eq (172) in [38]	\mathbf{J}_{Exp} as eq (179a) in [38]

Table 3. Closed form of functions needed to calculate the residual jacobian. \mathbf{I}_2 is the identity matrix of size 2

B. Fixed point algorithm for reverse flow estimation

We have a depth map in the reference view $(z_i^{(0)})_{i \in \mathbf{G}}$ and we note $\gamma_i^{(0)} = 1/z_i^{(0)}$ the associated disparity.

Given a disparity γ , a relative pose \mathbf{R}, \mathbf{t} and \mathbf{u} a point in the camera plane of the first view, we can calculate the image \bar{u} a point on the second camera plane and $\bar{\gamma}$ the projected disparity in the second view frame:

$$\bar{u}(\mathbf{u}, \gamma, \mathbf{R}, \mathbf{t}) = \psi(\mathbf{R}[\mathbf{u}, 1] + \gamma\mathbf{t}) \quad (10)$$

$$\bar{\gamma}(\mathbf{u}, \gamma, \mathbf{R}, \mathbf{t}) = \gamma\omega(\mathbf{R}[\mathbf{u}, 1] + \gamma\mathbf{t}). \quad (11)$$

In particular, given a regular grid of $\mathbf{u}_i^{(0)}$ in the reference view and the relative position of other views, $\mathbf{R}_k, \mathbf{t}_k$, we have the direct flow:

$$\bar{\mathbf{u}}_i^{(k)} = \bar{u}(\mathbf{u}_i^{(0)}, \gamma_i^{(0)}, \mathbf{R}_k, \mathbf{t}_k), \quad (12)$$

$\bar{\mathbf{u}}_i^{(k)}$ is not a regular grid in the view C_k , it is the image of a regular grid in the view C_0 . The direct flow warp I_k as an image $I_k^{(0)}$ in the camera plane C_0 . It is called a backward warp. But for some applications, we also need the warp of the image

I_0 as an image $I_0^{(k)}$ in the view C_k . This can be done using the direct flow $\bar{\mathbf{u}}_i^{(k)}$ and a forward warp, but it is known as not numerically stable. Instead, it can be computed using a backward warp and the *reverse flow*. The reverse flow is the other way around; the regular grid is $\mathbf{u}_i^{(k)}$ in the camera plane C_k and we want to find its antecedent $\bar{\mathbf{u}}_i^{(0)}$ in the camera plane of C_0 . The reverse can be computed using (10) using the inverse of the relative position and the disparity map in the view C_k . The inverse of the relative position is the inverse in $SE(3)$, and it is $\mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k$. On the other hand, the disparity map in the view C_k is not known. However, using the inverse relative position, the disparity $\gamma^{(k)}$ in a point $\mathbf{u}^{(k)}$ in the camera plane of C_k must match the known one in C_0 :

$$\bar{\gamma}(\mathbf{u}^{(k)}, \gamma^{(k)}, \mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k) = \Gamma^{(0)}(\bar{\mathbf{u}}(\mathbf{u}^{(k)}, \gamma^{(k)}, \mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k)), \quad (13)$$

where $\Gamma^{(0)}$ is the disparity function on the whole camera plane of C_0 using interpolation and the depth map $(z_i^{(0)})_{i \in \mathbf{G}}$. This equation can be interpreted as the reprojection of the disparity in C_k must match the disparity in C_0 evaluated in the flow induced by the disparity in C_k . It is an implicit equation for $\gamma^{(k)}$. Using (11) again from C_0 to C_k , it can be converted as a fixed point equation $\gamma^{(k)} = F(\gamma^{(k)})$ when defining F as :

$$F(\gamma) = \bar{\gamma}(\mathbf{u}, \Gamma^{(0)}(\mathbf{u}), \mathbf{R}_k, \mathbf{t}_k) \quad \text{with} \quad \mathbf{u} = \bar{\mathbf{u}}(\mathbf{u}^{(k)}, \gamma, \mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k).$$

Then we can estimate the disparity map $(\gamma_i^{(k)})_{i \in \mathbf{G}}$ in view C_k using a regular grid $\mathbf{u}_i^{(k)}$ in the view C_k and using a fixed point algorithm with a function F for every pixel using $\mathbf{u}_i^{(k)}$.

We build the sequence for $\gamma_{i,m}^{(k)}$ with $m > 0$ as:

$$\gamma_{i,m+1}^{(k)} = \bar{\gamma}(\mathbf{u}_{i,k,m}, \Gamma^{(0)}(\mathbf{u}_{i,k,m}), \mathbf{R}_k, \mathbf{t}_k) \quad \text{with} \quad \mathbf{u}_{i,k,m} = \bar{\mathbf{u}}(\mathbf{u}_i^{(k)}, \gamma_{i,m}^{(k)}, \mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k), \quad (14)$$

as the motion baseline is small, we initialize the disparity map in view C_k by the one in C_0 : $\gamma_{i,0}^{(k)} = \gamma_i^{(0)}$ and we can use the composition of the two flows (direct and reverse) as a convergence error:

$$\epsilon_{i,k,m} = \left\| \mathbf{u}_i^{(k)} - \bar{\mathbf{u}}(\mathbf{u}_{i,k,m}, \Gamma^{(0)}(\mathbf{u}_{i,k,m}), \mathbf{R}_k, \mathbf{t}_k) \right\| \quad (15)$$

The value of i and k for which the sequence does not converge correspond to the occlusion of the element projected in $\mathbf{u}_i^{(k)}$ between view C_k and C_0 . We can build an occlusion mask using the convergence criterion. Examples of these masks are available in appendix G. For the value of i and k for which the sequence does converge $\gamma_i^{(k)}$.

Finally, the reverse flow is given by:

$$\bar{\mathbf{u}}_i^{(0)} = \bar{\mathbf{u}}(\mathbf{u}_i^{(k)}, \gamma_i^{(k)}, \mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k). \quad (16)$$

C. Determinant regularization

The idea behind this regularization is that when the gradient is small, we will favor the direction of descent for the structure that deforms the current flow the least. To do this, we look at the flow effect on the center of the patches regularly distributed on the $\mathbf{G}^{(l)}$ grid. We note $i = i_x, i_y$ the i elements of the grid $\mathbf{G}^{(l)}$ with $i_x = 1..W_l$ and $i_y = 1..H_l$. We note (\mathbf{u}_{i_x, i_y}) the center point of the pixel in the corresponding image plane, and we suppose that the coordinates of \mathbf{u}_{i_x, i_y} are normalized and evolve in a range $[-1, 1]$. A parallelogram constituted by the points $(\mathbf{u}_{i_x, i_y}, \mathbf{u}_{i_x+1, i_y}, \mathbf{u}_{i_x+1, i_y+1}, \mathbf{u}_{i_x, i_y+1})$ thus has a normalized area of $4/(H_l W_l)$. We compare independently, for each view k and each grid mesh element, the normalized area of the mesh after application of the local homographic flow and the constant area noted $\bar{\mathbf{u}}_{i_x, i_y}^{(k)} = \hat{H}_{(i_x, i_y), k}(\mathbf{u}_{i_x, i_y})$. We penalize the ratio of the area of each parallelogram before and after the homography flow to 1. The penalization reads:

$$P = \sum_{k=1}^K \sum_{i_x=1}^{W_l-1} \sum_{i_y=1}^{H_l-1} \left| \frac{\mathcal{A}_{i_x, i_y}^{(k)}/2}{4/(H_l W_l)} - 1 \right| \quad (17)$$

$$\mathcal{A}_{i_x, i_y}^{(k)} = \det(\bar{\mathbf{u}}_{i_x+1, i_y}^{(k)} - \bar{\mathbf{u}}_{i_x, i_y}^{(k)}, \bar{\mathbf{u}}_{i_x, i_y+1}^{(k)} - \bar{\mathbf{u}}_{i_x, i_y}^{(k)}) \quad (18)$$

$$+ \det(\bar{\mathbf{u}}_{i_x, i_y+1}^{(k)} - \bar{\mathbf{u}}_{i_x+1, i_y+1}^{(k)}, \bar{\mathbf{u}}_{i_x+1, i_y}^{(k)} - \bar{\mathbf{u}}_{i_x+1, i_y+1}^{(k)}), \quad (19)$$

where \mathcal{A}_{i_x, i_y} is the double of the area of the parallelogram $(\bar{\mathbf{u}}_{i_x, i_y}^{(k)}, \bar{\mathbf{u}}_{i_x+1, i_y}^{(k)}, \bar{\mathbf{u}}_{i_x+1, i_y+1}^{(k)}, \bar{\mathbf{u}}_{i_x, i_y+1}^{(k)})$ using determinant on the two halves triangle as illustrated in figure 9.

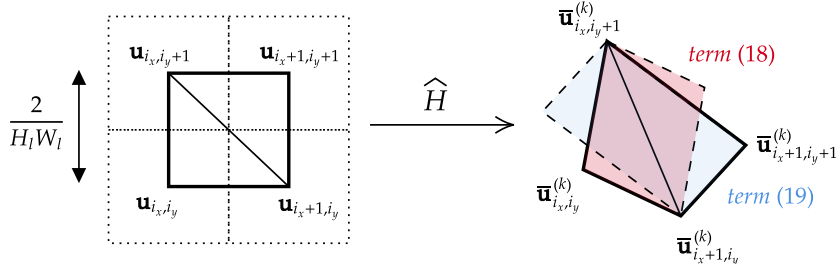


Figure 9. Illustration of the determinant regularization.

D. Additional details on the datasets and on the experiments

Proposed datasets. We generated two datasets: Blender 1 with small baselines and Blender 2 with micro-baselines. The first dataset exhibits larger parallax effects, while the second dataset has reduced parallax effects. Detailed characteristics of these datasets are provided in Table 4.

Experiments. In our experiments, whose results are reported in Table 1 and Table 2, we evaluated the performance of the Saop method [11] by calculating the average results across all scenes where Saop successfully converged. On the *blender 1* dataset, we excluded one scene where Saop did not converge. Excluding this scene for Saop does not change the methods’ ranking and our experiments’ conclusion.

E. Ablation study

We make an ablation study to understand the impact of the different choices in our modeling and algorithm. We compare the global algorithm to an identical algorithm using the same hyperparameters but, respectively, without the exponential parametrization of the motion, without the newton step, using spatial regularization (total variation and determinant), without the plan parametrization, with patches of size one, i.e., a pixel-wise loss and without the multiscale approach. We report the performance on the flow estimate in Table. 6, and depth/pose in Table. 7.

F. Comparison with monocular method

Monocular depth estimation methods can only estimate depth up to an affine transformation. Therefore, we evaluate them up to an affine correction. It does not make sense to compare them to the binocular method with linear correction as in Table 2. On the other hand, to compare them to the latter, we must recalculate the error of each of the methods in Table 2 with an affine correction. The results are reported in Table 5.

G. Estimated occlusion mask

We use the fixed point algorithm described in B on the depth map obtained at the optimization’s last step and note the points for which the fixed point algorithm does not converge. We use a threshold and a maximum number of iterations to construct the non-convergent set. This set constitutes a partial occlusion mask. It can be used in downstream tasks to avoid aggregating erroneous information because it is occluded. Fig. 10 shows examples of masks on synthetic data.

H. Depthmaps

We provide additional examples of depth maps from both synthetic bursts (Fig. 11) and real bursts (Fig. 12). All disparity maps were aligned to the ground truth with an affine transform by using the least square criterion of [36].

Dataset	Scenes	Frames	Std baselines (m)	Std rotations (deg)	Max depth (m)	Min depth (m)	Mean depth (m)
Blender 1	15	20	0.116	0.20	0.316	11.234	3.73
Blender 2	10	20	0.010	0.29	1.92	19.453	6.21

Table 4. Main characteristics of the two proposed datasets.

Method	Abs rel ↓	Sqr rel ↓	RMSE ↓	Delta 1 ↑	Delta 2 ↑	Delta 3 ↑
Blender 1 (small motion)						
Midas [36]	<u>0.1589</u>	1.0747	<u>1.3148</u>	0.8019	<u>0.951</u>	<u>0.9824</u>
RCVD [27]	0.2038	<u>1.0622</u>	1.3888	0.698	0.9191	0.9684
Ours	0.1544	0.2229	0.9258	<u>0.7881</u>	0.9544	0.9911
Blender 2 (micro motion)						
Midas [36]	0.0790	0.0786	0.7166	0.9429	0.9929	0.9986
RCVD [27]	<u>0.0971</u>	<u>0.1131</u>	<u>0.8244</u>	<u>0.9149</u>	0.988	0.9973
Ours	<u>0.1763</u>	<u>0.2875</u>	<u>1.3711</u>	<u>0.6857</u>	0.9594	<u>0.9976</u>

Table 5. Depth errors metrics on the two proposed synthetic bursts datasets.

Method	EPE	RMSE	NPE1	NPE2	NPE3	EPE	RMSE	NPE1	NPE2	NPE3
	↓	↓	↑	↑	↑	↓	↓	↑	↑	↑
Blender 1 (small motion)						Blender 2 (micro motion)				
Base	0.7439	1.4324	0.7841	0.9084	0.9456	0.2321	0.2820	0.9366	0.9972	1.0000
with regularization	0.7641	1.4596	0.7732	0.9024	0.9432	0.2660	0.3286	0.9297	0.9937	0.9997
with $k = 1$ (pixelwise)	0.8102	1.4705	0.7512	0.8909	0.9377	0.2834	0.3482	0.9220	0.9940	0.9997
w/o plan parametrization	544.3828	3151.1105	0.2983	0.4619	0.5569	219.6589	1234.7524	0.6671	0.7301	0.7420
w/o exponential parametrization	0.7685	1.4629	0.7721	0.9013	0.9421	0.2658	0.3294	0.9294	0.9937	0.9997
w/o newton step	0.7676	1.4630	0.7725	0.9015	0.9422	0.2741	0.3402	0.9272	0.9933	0.9997

Table 6. Optical flow errors.

Method	Pose				Depth					
	Left l2 (m) ↓	ATE (m) ↓	Geom (m) ↓	Biinvrot l2 (deg) ↓	Abs rel ↓	Sqr rel ↓	RMSE ↓	Delta 1 ↑	Delta 2 ↑	Delta 3 ↑
Blender 1 (small motion)										
Base	0.0066	0.0056	0.0050	0.1806	0.1381	0.2391	0.8688	0.8358	0.9263	0.9761
with regularization	0.0072	0.0062	0.0053	0.1850	0.1399	0.2462	0.8777	0.8344	0.9236	0.9759
with $k = 1$ (pixelwise)	0.0073	0.0062	0.0054	0.1883	0.1538	0.2673	0.9217	0.8087	0.9220	0.9734
w/o plan parametrization	0.0267	0.0250	0.0221	0.4317	0.4518	0.7926	1.7602	0.2319	0.5076	0.7142
w/o exponential parametrization	0.0073	0.0062	0.0054	0.1865	0.1392	0.2398	0.8694	0.8340	0.9252	0.9763
w/o newton step	0.0073	0.0062	0.0054	0.1861	0.1393	0.2397	0.8696	0.8342	0.9253	0.9763
Blender 2 (micro motion)										
Base	0.0022	0.0022	0.0020	0.0245	0.1383	0.1962	1.1521	0.7996	0.9819	0.9983
with regularization	0.0023	0.0022	0.0020	0.0256	0.1766	0.2935	1.3640	0.6943	0.9498	0.9948
with $k = 1$ (pixelwise)	0.0024	0.0024	0.0022	0.0287	0.1750	0.2928	1.3629	0.6932	0.9503	0.9952
w/o plan parametrization	0.0040	0.0039	0.0037	0.0430	0.2571	0.5270	1.7727	0.5005	0.8881	0.9857
w/o exponential parametrization	0.0022	0.0022	0.0020	0.0261	0.1755	0.2908	1.3590	0.6981	0.9510	0.9949
w/o newton step	0.0023	0.0022	0.0021	0.0258	0.1818	0.3040	1.3837	0.6807	0.9458	0.9946

Table 7. Pose and depth errors metrics on the two proposed synthetic bursts datasets.

For a fair comparison, we also show the results of DfUSMC without their additional depth map filtering, which is essential to obtain a visually appealing depth map. However, this step introduces a stratification of the depth map, which is not present in our method.

I. Pose estimation visualization

To visualize the positions the algorithm approximates, we can look at the translation part of the positions. Because our images come from a burst, we use the temporal coherence of the series of pictures and can trace the trajectory of the camera center during the burst. After rescaling, we compare the trajectory approximated by the algorithm to the trajectory used to create the burst in Blender. Fig. 13 shows examples of trajectories for different images of the Blender 2 dataset during the last three stages.

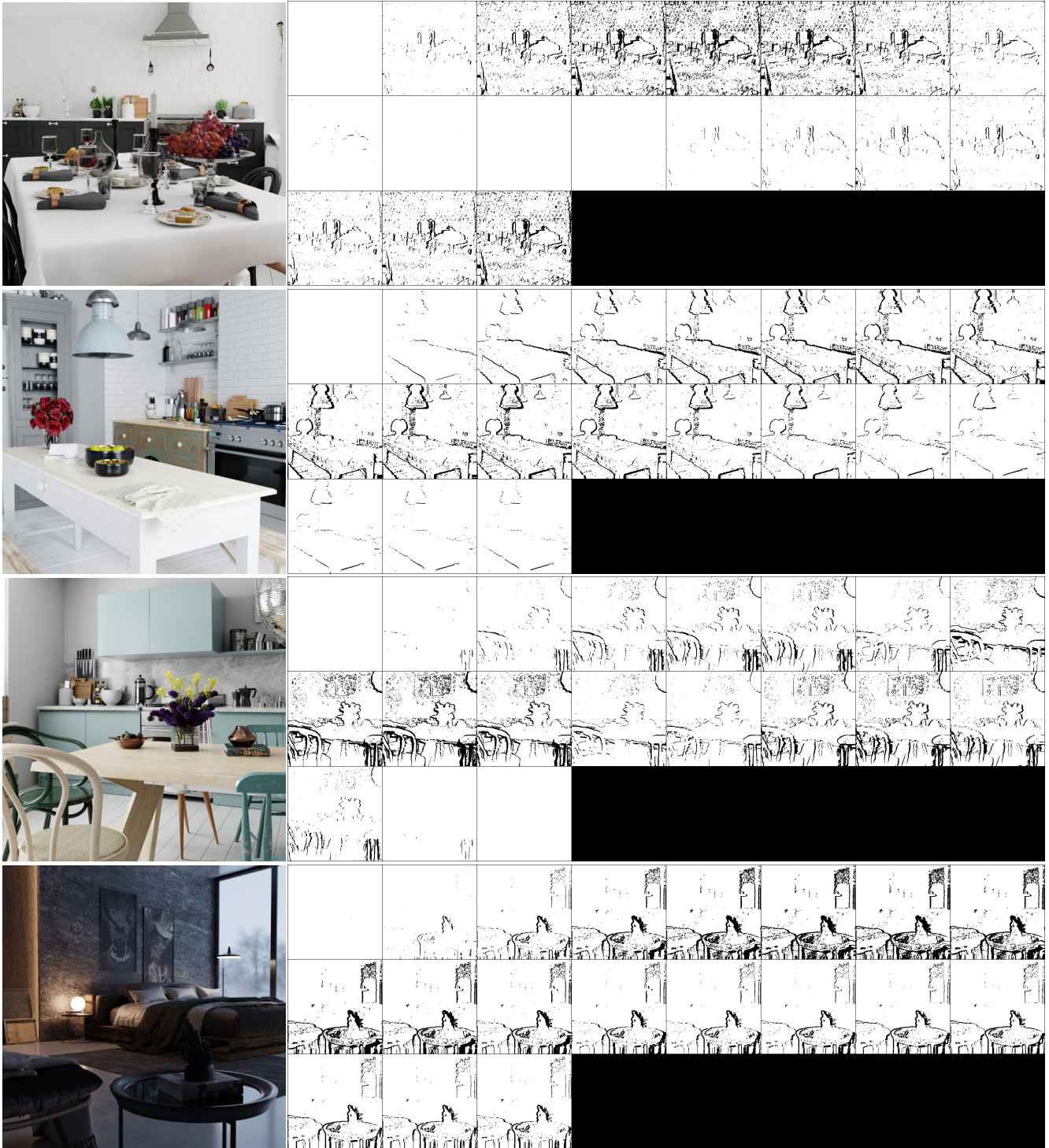


Figure 10. Partial occlusion mask obtained using the fixed point algorithm for four examples of the Blender 2 dataset.

J. Visual inspection of the registration of real frames

Fig. 14 visually demonstrates the alignment quality achieved with our method on a real burst. To assess the alignment quality, we generate images by overlaying the green and blue channels of the warped source images onto the red channel of the

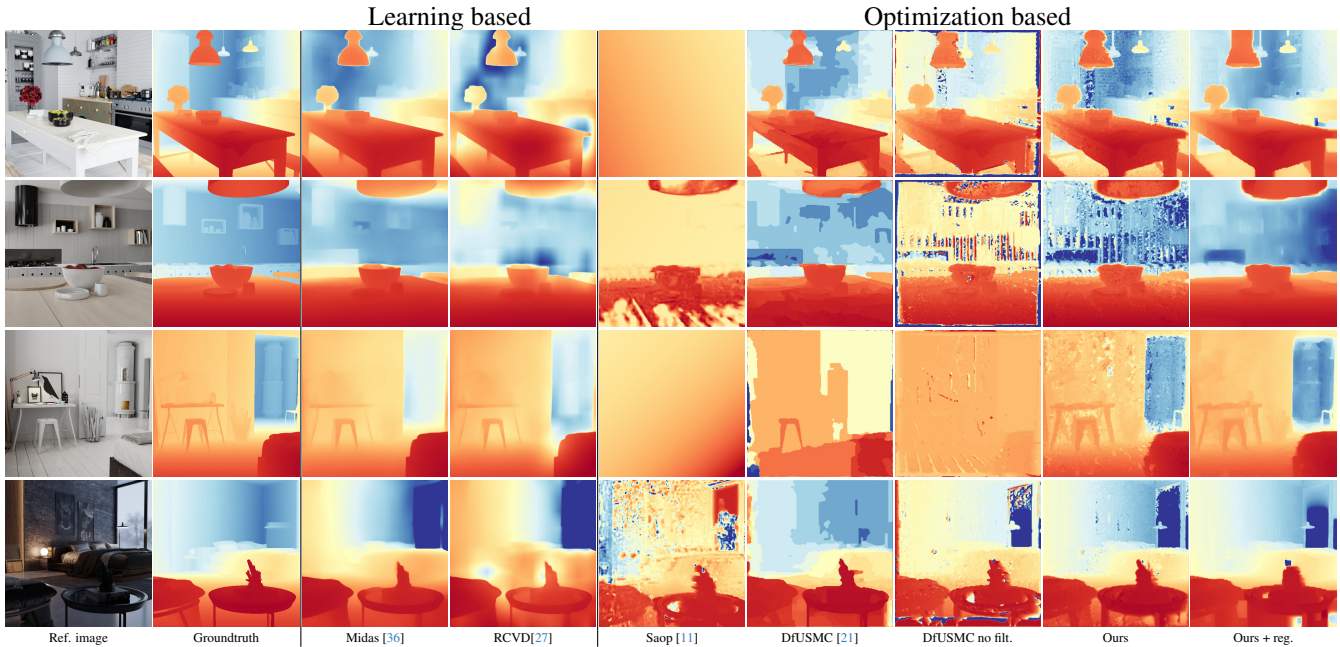


Figure 11. Depth estimation from synthetic bursts (*Blender 2* dataset).

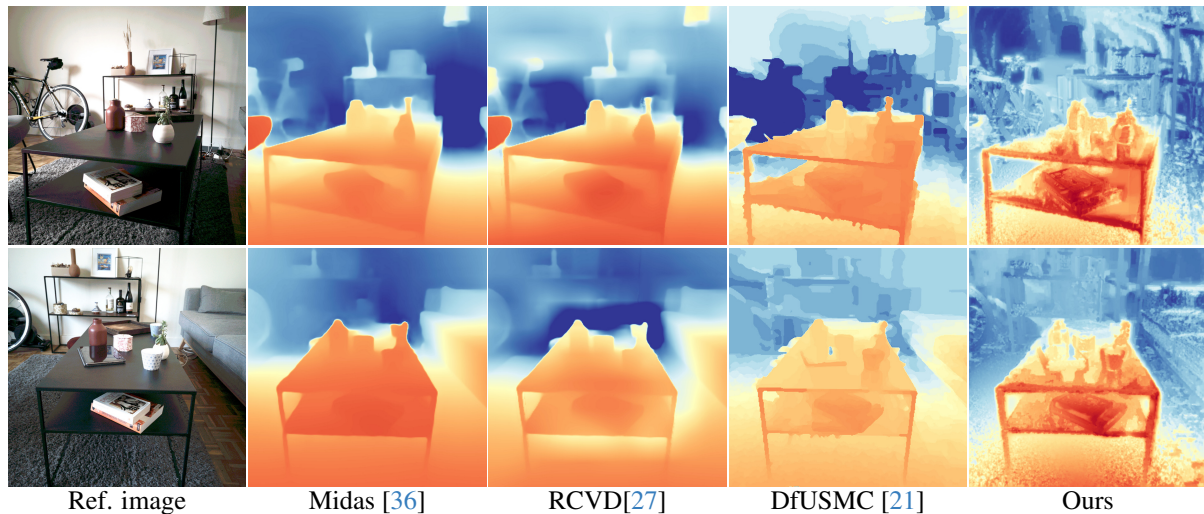


Figure 12. Depth estimation from real bursts.

target image, following a similar approach as [12]. In this example, we observe that the majority of the frames exhibit a good alignment, while a few frames (5 out of 15) show inadequate alignment particularly in certain regions of the foreground (see for example the books or the plant).

K. Super-resolution on real bursts

To showcase the ability of our method to produce fine alignments on real images, we perform burst super-resolution (SR) with our alignments. To achieve the task, we use the popular inverse problem framework employed in [18, 28]. To recover the high-resolution image \mathbf{x} from a set of K noisy and low-resolution observations \mathbf{y}_i with $i \in [0, K]$ we solve the minimization problem $\min_{\mathbf{x}} \sum_i^K \|DBW_i\mathbf{x} - \mathbf{y}_i\|_2^2$, with a gradient descent algorithm.

D is a decimation operator that reduces spatial resolution, B is a blurring operator, and W is a warp parametrized by the optical flow. In our experiments, DB is chosen as the average pooling operator following [28]. The gradient can be derived as

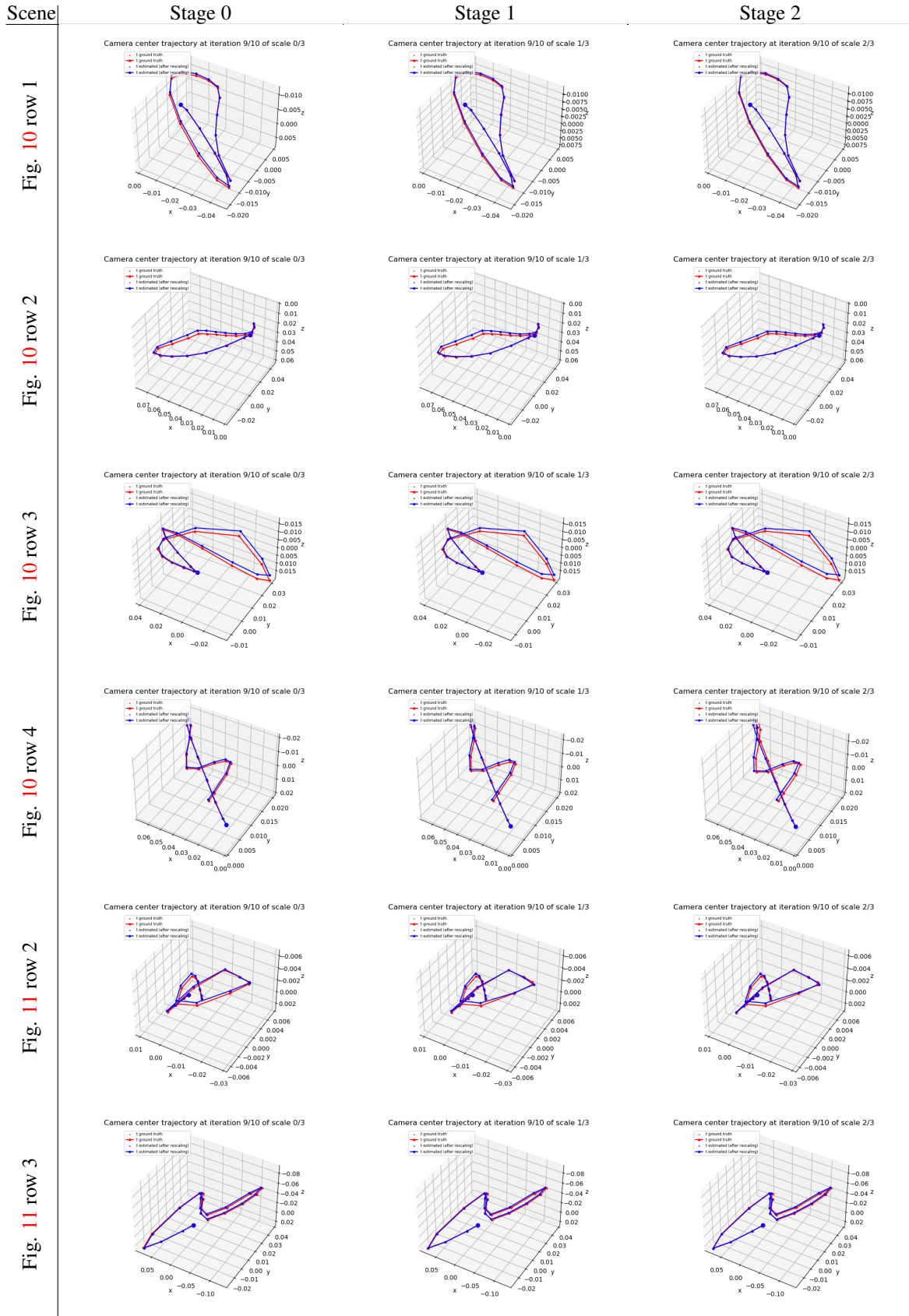


Figure 13. Trajectory at different scales of the coarse to fine approach for all the scenes shown in Fig. 10 and Fig. 11.



Figure 14. Qualitative alignment results of our method on a real burst. Images are generated by superimposing the warped source images on the target image.

$$\sum_i^K W_i^\top B^\top D^\top (DBW_i \mathbf{x} - \mathbf{y}_i).$$

The optical flow to warp the reference high-resolution image \mathbf{x} candidate is estimated in two steps using our method and then the fixed point algorithm presented in Sec. 3 to infer the motion field of interest. We perform super-resolution on RGB images in linear space demosaicked RAW frames with bilinear filtering. Joint super-resolution and demosaicking is left for future work.

We visually compare our results in Figure 15. Our algorithm can recover fine details, including, for instance, the fine texture on the rum bottle or the hair of the doll, that were not distinguishable in the original frames.



Figure 15. burst super-resolution on real raw bursts exploiting our alignment method. Top: low-resolution crops. Bottom: super-resolution exploiting our alignment method. Data kindly provided by the authors of [28]. It is best seen by zooming aggressively on a computer screen.