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-04337706
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\textsuperscript{1,2,*} Yann Dubois de Mont-Marin\textsuperscript{2,*} Théo Bodrito\textsuperscript{2,*} Julien Mairal\textsuperscript{4} Jean Ponce\textsuperscript{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.

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. 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.

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.
\textsuperscript{1}Enhance Lab.
\textsuperscript{2}Inria and DIENS (ENS-PSL, CNRS, Inria).
\textsuperscript{3}Courant Institute and Center for Data Science, New York University.
\textsuperscript{4}Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK.
\textsuperscript{5}Corresponding author: bruno.lecouat@enhancelab.fr
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
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 sparse 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 \( u_i \) in \( G \) a regular grid of the \( C_0 \) camera plane, is the projection of a point \( x_i \) of the scene surface. We denote by \( \pi_i \), the affine plane tangent to the scene in \( x_i \) parameterized, by its (non-unit) normal \( n_i \) such that \( \pi_i = \{ y \in \mathbb{R}^3, n_i^T y = 1 \} \). A patch \( P(u_i) \) around \( u_i \) is the projection of a patch around \( x_i \) in \( \pi_i \), and its image in the camera plane \( C_k \) is given by a homography uniquely...
When a pixel \( u \) where \( \rho \) where \( H \) x the projection of another point \( y \) formation model does not account for occlusion phenomena. I that the flows derived from local homographies \( R \) objective is to estimate the optical flow between view \( C \) parametrizing the scene structure and \( n \) Minimization problem. The parameters of our formation \( n \) of freedom. The formation model is summed up in Fig. 2.

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

defined by the plane \( \pi \), and the extrinsic parameters \( R_k, t_k \) of the other camera (Fig. 2). For \( u' \) in the patch \( P(u_i) \), we have the direct flow locally expressed as a homography:

\[
\hat{H}_{i,k}(u') = \psi(H_{i,k}[u', 1]^T) \quad (1)
\]

\[
H_{i,k} = R_k + t_k n_i^\top, \quad (2)
\]

where \( H_{i,k} \) is the homography matrix for the patch \( P(u_i) \) in the camera plane of \( C_k \), \([u', 1]\) is the homogeneous representation of \( u' \) and \( \psi : x, y, z \to x/z, y/z \) is the standard projection. The parameters of this flow are the non-unit normal \( n_i \) characterizing the plane \( \pi \) and the pose \( R_k, t_k \). It is important to note, as in [11, 22], that \( n_i \) 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 \( n = (n_i)_{i \in G} \) the dense map over a regular grid \( G \) parametrizing the scene structure and \( R, t = (R_k, t_k)_{k=1..K} \) 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 \( R, t, n \) 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:

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

where \( \rho \) is a robust loss function as in [39]. Indeed, the formation model does not account for occlusion phenomena. When a pixel \( u' \) in the \( C_0 \) plane is the projection of a point \( x \) that is not visible in camera \( C_k \), \( \hat{H}_{i,k}(u') \) is essentially the projection of another point \( y \) that is not on the same scene element as \( x \). Consequently, it is likely that the color \( I_k(\hat{H}_{i,k}(u')) \) deviates significantly from \( I_0(u) \). The function \( \rho \) 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 \( t \) by \( \lambda t \) and \( n \) by \( 1/\lambda n \) 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 \( n \) and steps on the relative poses \( R, t \). Indeed, when \( n \) is fixed, there is no longer any ambiguity about the value that \( t \) can take, and the same applies to \( n \) when \( t \) 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 \( R_k = I \) and \( t_k = 0 \). 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 \( z = (z_{i})_{i \in G^{(0)}} \) on a very low-resolution grid \( G^{(0)} \) (typically \( 16 \times 16 \)), we can initialize the plane map as \( n^{(0)} = ([0, 0, 1/2])_{i \in G^{(0)}} \). This corresponds to initializing the planes as fronto-parallel and located at a distance \( z \) 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 \( G^{(0)}, \ldots, G^{(L)} \) 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 \( R, t \) using the high-resolution grid \( G^{(L)} \), a linear interpolation of the current estimate of the plane map \( u^{(l)} \) to the resolution of \( G^{(L)} \), and using the high-resolution images \( (I_k^{(l)})_{k=0..K} \). 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 \( n^{(l)} \) at scale \( l \) using the gradient descent variation Adam [26], with the loss calculated using the grid \( G^{(l)} \), \( R \), \( t \), \( n \), \( I \) and the images at resolution \( l: (I^{(l)}_{k})_{k=0..K} \). Using a method with moments like Adam accelerates the convergence of the procedure.
- Every few alternate steps on \( R \), \( t \) on one side and \( n^{(l)} \) on the other side, we double the resolution of the plane map \( n^{(l)} \) and move to the next scale with \( n^{(l+1)} \).

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 \( \times \) 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 \( \times \) 16
3: \( R \leftarrow I \) \( \triangleright \) \( K \) matrices \( 3 \times 3 \)
4: \( t \leftarrow 0 \) \( \triangleright \) \( K \) vectors of size 3
5: \( l \leftarrow 0 \)
6: while \( l \leq L \) do \( \triangleright \) Multiscale loop
7: \( n \leftarrow \text{interpole}(n, 2 \times \text{resolution}(n)) \) \( \triangleright \) Double resolution
8: \( I_{\perp} \leftarrow \text{sample}(I, \text{resolution}(n)) \) \( \triangleright \) Low resolution image
9: \( m \leftarrow 0 \)
10: while \( m \leq M \) do \( \triangleright \) Block descent
11: \( n_{+} \leftarrow \text{interpole}(n, \text{resolution}(I)) \) \( \triangleright \) High resolution
12: \( R, t \leftarrow \text{PGN}(R, t, n_{+}, I) \) \( \triangleright \) Pose Newton step
13: \( n \leftarrow \text{ADAM}(R, t, n, I_{\perp}) \) \( \triangleright \) 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 \( (z^{(0)})_{i} \in G^{(l)} \) in the reference view \( (C_{0}) \), with interpolation we construct \( Z \) that gives the depth for any \( u_{0} \). We initialize a depth map \( (z^{(k)})_{i} \in G^{(l)} \) 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 \( Z(F(u_{i}^{(k)})) \). 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 \( R \), \( t \), and \( n^{(L)} \) 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 \( u_{i}^{(0)} \in G^{(L)} \):

\[
u_{i}^{(k)} = H_{i,k}(u_{i}^{(0)}) = \psi(R_{k}u_{i}^{(0)} + 1/z_{i}^{(0)}t_{k}),
\]

where, from \( n^{(L)} \), we recover a high-resolution depth map in the reference view: \( z_{i}^{(0)} = 1/(|u_{i}^{(0)}, 1|)n_{i}^{(L)} \). Normalizing \( n^{(L)} \) also provides a high-resolution normal map. Finally, our algorithm directly estimates the camera poses \( R \), \( t \). For certain applications, such as super-resolution, we need the inverse flow, i.e., \( u_{i}^{(0)} = F(u_{i}^{(k)}) \). 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 \( u_{i}^{(0)} = F(u_{i}^{(k)}) \) as in (4) using the inverse poses: \( [R_{k}^{\top}, -R_{k}t_{k}] \). 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.
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.

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 outline 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 x 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 $T = [t_0, \ldots, t_N]$. 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
Ref. image

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.

Ref. image Groundtruth Midas [36] RCVD [27]

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
\[ d([R, t], [R', t'])^2 = \|t' - t\|^2_2 + \lambda \| \log(R^T R') \|^2_2.\]

For \( \lambda \), we use the median value of the ground truth depth. \( \| \log(R^T R') \|^2_2 \) 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
trades with lower performance in terms of flow and pose metrics on synthetic data. We observed that no spatial regularization plan 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.

**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 $16 \times 16$ 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.

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 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


Appendix

A. Closed form Jacobian 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 \( r_{k, i, u'} \) with coordinates indexed by \( k = 1..K, i \in G, u' \in P_i \) with total dimension noted \( KD \) with \( D = \#(G)\#(P) \) size of grid by size of patch:

\[
r_{k, i, u'} = I_0(u') - I_k(\hat{H}_{i, k}(u')).
\]  

We note the twist \( \xi_k \) such that \( R_{k, i}, t_k = \exp(\xi_k) \). We want to find the Jacobian \( J_k \) with the variable flat vector \( \xi = (\xi_k)_{k=1..K} \).

Then we note that \( J_{\xi} \) is of dimensions \( KD, K \) and that \( J_{\xi} \) is diagonal by \( K \) block of dimension \( D, K \). We note \( J_k \) these blocks. Using the expression of the homography matrix in (2), the block \( J_k \) have \( D \) rows of the form \( (\nabla \phi_{k, i, u'})^\top \) where:

\[
\phi_{k, i, u'}(\xi) = I_k(\psi((R + u^\top n^\top)u')) \\
= I_k\left(\psi\left(R\frac{1}{n^\top u'}u' + t\right)\right),
\]

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

\[
\begin{align*}
\phi_{k, i, u'}(\xi) &= I_k \circ \psi \circ \Lambda_X, u' \circ \exp \\
(\nabla \phi_{k, i, u'})^\top &= \nabla I_k^\top J_{\psi} J_{\Lambda_X, u'} J_{\exp}.
\end{align*}
\]

Note that \( \Lambda_X \) 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. \( \nabla I_k \) 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 \( J_{\xi} \) in our implementation.

<table>
<thead>
<tr>
<th>Domains</th>
<th>Function</th>
<th>Jacobian</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{R}^3 \rightarrow \mathbb{R}^2 )</td>
<td>( \psi(x, y, z) = [x/z, y/z]^\top )</td>
<td>( J_{\psi} = \frac{1}{z}[I_2 \mid -\psi(x, y, z)] )</td>
</tr>
<tr>
<td>( SE(3) \rightarrow \mathbb{R}^3 )</td>
<td>( \Lambda_X(R, t) = RX + t )</td>
<td>( J_{\Lambda_X} = [R - R[X]_x] )</td>
</tr>
<tr>
<td>( \mathbb{R}^6 \rightarrow SE(3) )</td>
<td>( \exp(\xi) ) as eq (172) in [38]</td>
<td>( J_{E_{\exp}} ) as eq (179a) in [38]</td>
</tr>
</tbody>
</table>

Table 3. Closed form of functions needed to calculate the residual jacobian. \( I_2 \) is the identity matrix of size 2

B. Fixed point algorithm for reverse flow estimation

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

Given a disparity \( \gamma \), a relative pose \( R, t \) and \( u \) 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:

\[
\begin{align*}
\bar{u}(u, \gamma, R, t) &= \psi(R[u, 1] + \gamma t) \\
\bar{\gamma}(u, \gamma, R, t) &= \gamma \omega \left(R[u, 1] + \gamma t\right). 
\end{align*}
\]

In particular, given a regular grid of \( u_i^{(0)} \) in the reference view and the relative position of other views, \( R_k, t_k \), we have the direct flow:

\[
\bar{u}_i^{(k)} = \bar{u}\left(u_i^{(0)}, \gamma_i^{(0)}, R_k, t_k\right),
\]

\( \bar{u}_i^{(k)} \) 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 $I^{(k)}_0$ in the view $C_k$. This can be done using the direct flow $\bar{u}^{(k)}_i$ 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 $u^{(k)}_i$ in the camera plane $C_k$, and we want to find its antecedent $\bar{u}^{(0)}_i$ 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 $R^{k}_k = -R^{k}_k t_k$. 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 $u^{(k)}_i$ in the camera plane of $C_k$ must match the known one in $C_0$:

$$\gamma(u^{(k)}_i, \gamma^{(k)}_i, R^{k}_k, -R^{k}_k t_k) = \Gamma^{(0)} \left( \bar{u}(u^{(k)}_i, \gamma^{(k)}_i, R^{k}_k, -R^{k}_k t_k) \right),$$

(13)

where $\Gamma^{(0)}$ is the disparity function on the whole camera plane of $C_0$ using interpolation and the depth map $(z_i^{(0)})_{i \in G}$. 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:

$$F(\gamma) = \bar{\gamma} \left( u, \Gamma^{(0)}(u), R_k, t_k \right) \quad \text{with} \quad u = \bar{u} \left( u^{(k)}_i, \gamma, R^{k}_k, -R^{k}_k t_k \right).$$

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

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

$$\gamma^{(k)}_{i,m+1} = \bar{\gamma} \left( u_{i,k,m}, \Gamma^{(0)}(u_{i,k,m}), R_k, t_k \right) \quad \text{with} \quad u_{i,k,m} = \bar{u} \left( u^{(k)}_i, \gamma^{(k)}_{i,m}, R^{k}_k, -R^{k}_k t_k \right),$$

(14)

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

$$\epsilon_{i,k,m} = \left\| u^{(k)}_i - \bar{u} \left( u^{(k)}_i, \gamma^{(k)}_{i,m}, R^{k}_k, -R^{k}_k t_k \right) \right\|$$

(15)

The value of $i$ and $k$ for which the sequence does not converge correspond to the occlusion of the element projected in $u^{(k)}_i$ 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 $\gamma^{(k)}_i$.

Finally, the reverse flow is given by:

$$\bar{u}^{(0)}_i = \bar{u} \left( u^{(k)}_i, \gamma^{(k)}_i, R^{k}_k, -R^{k}_k t_k \right).$$

(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 $G^{(l)}$ grid. We note $i = i_x, i_y$ the $i$ elements of the grid $G^{(l)}$ with $i_x = 1..W_l$ and $i_y = 1..H_l$. We note $(u_{i_x,i_y})$ the center point of the pixel in the corresponding image plane, and we suppose that the coordinates of $u_{i_x,i_y}$ are normalized and evolve in a range $[-1, 1]$. A parallelogram constituted by the points $(u_{i_x+1,i_y}, u_{i_x+1,i_y+1}, u_{i_x,i_y+1}, u_{i_x,i_y+1})$ 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 $\bar{u}^{(k)}_{i_x,i_y} = \bar{H}_{(i_x,i_y),k}$. We penalize the ratio of the area of each parallelogram before and after the homography flow to 1. The penalization reads:

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

(17)

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

(18)

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

(19)

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scenes</th>
<th>Frames</th>
<th>Std baselines (m)</th>
<th>Std rotations (deg)</th>
<th>Max depth (m)</th>
<th>Min depth (m)</th>
<th>Mean depth (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blender 1</td>
<td>15</td>
<td>20</td>
<td>0.116</td>
<td>0.20</td>
<td>0.316</td>
<td>11.234</td>
<td>3.73</td>
</tr>
<tr>
<td>Blender 2</td>
<td>10</td>
<td>20</td>
<td>0.010</td>
<td>0.29</td>
<td>1.92</td>
<td>19.453</td>
<td>6.21</td>
</tr>
</tbody>
</table>

Table 4. Main characteristics of the two proposed 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.
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
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 $x$ from a set of $K$ noisy and low-resolution observations $y_i$ with $i \in [0, K]$ we solve the minimization problem $\min_x \sum^K_i \| DBW_i x - y_i \|_2^2$, 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.
The optical flow to warp the reference high-resolution image $x$ 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.