MBA-VO: Motion Blur Aware Visual Odometry

Peidong Liu[1] Xingxing Zuo[1] Viktor Larsson[1] Marc Pollefeys[1,2]
[1]Department of Computer Science, ETH Zürich
[2]Microsoft Mixed Reality and AI Lab, Zürich

Abstract

Motion blur is one of the major challenges remaining for visual odometry methods. In low-light conditions where longer exposure times are necessary, motion blur can appear even for relatively slow camera motions. In this paper we present a novel hybrid visual odometry pipeline with direct approach that explicitly models and estimates the camera's local trajectory within the exposure time. This allows us to actively compensate for any motion blur that occurs due to the camera motion. In addition, we also contribute a novel benchmarking dataset for motion blur aware visual odometry. In experiments we show that by directly modeling the image formation process, we are able to improve robustness of the visual odometry, while keeping comparable accuracy as that for images without motion blur. Both the code and the datasets can be found from https://github.com/ethliup/MBA-VO.

1. Introduction

Visual odometry (VO) determines the relative camera motion from captured images. As a fundamental block for many vision applications, such as robotics and virtual/augmented/mixed reality, great progress has been made during the last two decades. There have been many algorithms proposed in the literature: ranging from classical geometric approaches, deep learning based approaches to hybrid approaches. The geometric approaches recover the motion based on multi-view geometric constraints. Both the reprojection error (e.g. ORB-SLAM [23]) and the photometric consistency (e.g. DSO [7]) are commonly used constraints for the optimization. Deep learning based approaches formulate the problem as an end-to-end regression problem. Current state-of-the-art networks are still not able to achieve comparable performance to the classical approaches for large scale environments. Hybrid approaches usually embed a deep network inside a classical pipeline, to further improve their accuracy and robustness.

While many state-of-the-art algorithms have been proposed, motion blur is still a major challenge remaining for visual odometry methods. Motion blur is one of the most common artifacts that degrade images. It usually occurs in low-light conditions where longer exposure times are necessary. This affects both feature based approaches (e.g. ORB-SLAM [23]), which struggle to detect keypoints, and direct methods (e.g. DSO [7]) which rely on strong image gradients for their alignment. While relocalization strategies can partially mitigate the problem by allowing the VO to recover after losing track, VO would still fail if the camera continues to move in un-explored areas.

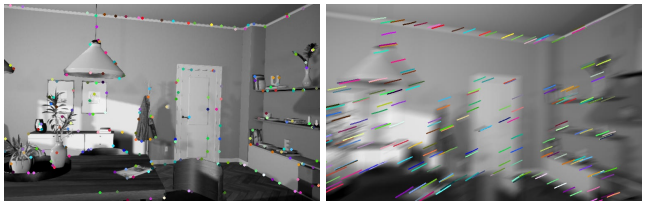


Figure 1. *Motion Blur Aware Visual Odometry*. We propose a full pipeline for performing motion blur aware visual odometry. By explicitly modelling the image formation process during tracking, we can actively compensate for motion blur in the direct image alignment.

In this paper, we thus propose a novel hybrid visual odometry method which is robust to motion blur. As conventional algorithms, our method consists of a front-end tracker and a back-end mapper. During tracking, instead of estimating the camera pose at a particular point in time, we estimate the local camera motion trajectory within the exposure time for each frame. This allows us to explicitly model the motion blur in the image and leverage it for tracking. We assume that the reference keyframe image is sharp, which is achieved by applying a deep deblurring network on the original motion blurred image. Since keyframes are usually sampled with a frequency much lower than frame-rate (and are less sensitive to latency), we can thus take advantage of a powerful deep network for keyframe deblurring (e.g. [32]). To estimate the camera motion trajectory during image capture, we locally re-blur the sharp reference keyframe image that is then used for direct image alignment against current tracked frame. The back-end jointly optimizes the camera poses and scene geometry based on the deblurred keyframe images, by maximizing the photometric consistency. We

build our method on the popular DSO [7] framework.

As another contribution, we also propose a novel benchmarking dataset targeting motion blur aware VO. Our dataset contains sequences with varying levels of motion blur. Time synchronized ground truth trajectories are also provided by an accurate indoor motion capturing system. By making this dataset publicly available to the community, we hope to encourage further research on making VO robust, which is important for real-world deployments.

We evaluate our approach with both synthetic dataset and real datasets. The experimental results demonstrate that we are able to improve the robustness of the visual odometry, while keeping comparable accuracy as that for images without motion blur. Furthermore, our motion blur aware VO (called *MBA-VO*) is also able to run in real-time on a laptop with a Nvidia GeForce RTX 2080 graphic card.

2. Related Work

Visual odometry: Existing works on visual odometry can be categorized into three main groups: classical geometric approaches, deep learning based approaches and hybrid approaches.

Classical geometry-based approaches recover the camera motion from multi-view constraints. These methods can be further divided into direct approaches and feature-based approaches. Direct approach relies on the photometric consistency assumption across multiple view within a short time interval. They jointly optimize the camera poses, 3D scene structure as well as camera intrinsic parameters by maximizing the photometric consistency. The representative works are LSD-SLAM [8], DSO [7] and their many variants [30, 20, 19, 10]. Different from direct method, feature-based methods extract a set of sparse keypoints from the raw images which are then matched across different views. Both the camera poses and 3D scene geometry are estimated by enforcing consistency between the keypoint locations and the projections of the scene structure. Dating back from the early work by Davison et al. [6] and Nister et al. [25], to the more recent ORB-SLAM [23], many feature-based approaches have been proposed in the literature. More details can be found from a recent review paper from Cadena et al. [4]. Deep learning-based approaches usually formulate the problem as an end-to-end regression problem. Although several pioneering works [39, 40, 34] and their variants have been proposed in the past years, they are still in their infancy compared to geometric approaches in terms of scalability and performance. Recently there have been a series of hybrid methods [33, 2, 38], which try to embed deep networks into classical geometric frameworks. These frameworks aim to leverage the benefits of both approaches to robustify the visual odometry.

Almost all those algorithms assume the input images are of good quality. However, due to environmental conditions (e.g. low light), low quality of image is sometimes unavoidable in real world applications, which can then drastically reduce the performance of VO systems. In this paper we propose to tackle one of the most common challenging cases, motion blurred images. The early works from Pretto et al. [28] and Lee et al. [17] have been proposed to improve the robustness of sparse keypoint based VO against motion blur. Pretto et al. [28] propose to detect motion blur robust sparse invariant features. The work from Lee et al. [17] is perhaps the one which is most similar to ours. In [17], the authors assume the motion between neighbouring frames is smooth and try to linearly interpolate the motion within the exposure time. For each frame the initial motion is extrapolated from previous frames using a motion model and this prediction is used to re-blur the patches from the keyframe. The re-blurred patches are used to establish explicit sparse correspondences between the new frame and the keyframe. The camera poses and scene structure are then estimated from these correspondences. In this work we take a similar approach to [17], where we re-blur patches extracted from the keyframe. In contrast to [17], which relies heavily on the initial motion prediction to make hard decisions on correspondences, we directly optimize the local camera motion trajectory used to re-blur the patches and instead implicitly solve the data association problem using a direct image alignment approach.

Image deblurring: Motion deblurring methods can be categorized into classic optimization based approaches and modern deep learning based approaches. We will only focus our attention on several representative single image deep deblurring networks, since they are most related to our work. Recently, the performance of single image deblurring algorithms has been boosted significantly by deep neural networks. The early work for image deblurring by Xu et al. [35] is a shallow network with four hidden layers, which is trained end-to-end with known ground truth sharp images. Hradis et al. [14] later propose a 15 layer network for text image deblurring. The network was further enlarged to 40 layers in a multi-scale manner by Nah et al. [24], resulting a network with 120 layers for three pyramid scales. Adversarial loss [12] was also introduced to improve the deblurring performance by Kupyn et al. [15]. Another two concurrent works from Tao et al. [32] and Zhang et al. [37] also achieve the state-of-the-art performance by using recurrent neural networks (RNN). To further improve the generalization performance, Liu et al. [21] recently propose a self-supervised single image deblurring network. Although these networks achieve remarkable performance, they usually cannot run at frame-rate even with a high-end GPU. Recently, Kupyn et al. [16] propose a light-weight network which is able to run in real-time, with an expense of slightly downgraded deblurring quality. In this work, we will ex-

plore the networks from Tao et al. (better quality but slow) [32] and Kupyn et al. (lower quality but fast) [16].

Existing dataset for robust visual odometry: In the last decade there have been several benchmark datasets proposed for evaluating visual odometry and SLAM methods [31, 11, 13, 1, 3, 27, 22, 29]. Some datasets focus on evaluating specific aspects or settings; e.g. autonomous-driving [11], illumination changes [26] and long-term re-localization [5]. In this paper we propose a new benchmark dataset for evaluating visual odometry which specifically targets at motion blur. While images with motion blur appear in some of the previous datasets (e.g. [31, 29], see Section 4), it is not their main focus. We provide sequences with varying degrees of motion blur, which allows us to more precisely evaluate the breaking points of different methods.

3. Method

In this section we present our motion blur aware visual odometry. We build on Direct Sparse Odometry (DSO) from Engel et al. [7]. The proposed pipeline consists of three main parts: a motion blur aware visual tracker, a keyframe deblurring network and a local mapper.

The front-end tracker estimates the camera motion trajectory within the exposure time of current blurry frame, relative to the latest sharp keyframe image. Each new keyframe is processed with the motion deblurring network. The local mapper then jointly optimizes the camera poses and the scene structure based on the recovered latent sharp keyframe images. We use the same local mapper as in DSO [7] and the main technical contribution of our work is the motion blur aware tracker, which we will detail in the following sections.

Motion blur image formation model: The physical image formation process of a digital camera, is to collect photons during the exposure time and convert them into measureable electric charges. This process can be mathematically modelled as integrating over a set of virtual sharp images:

$$\mathbf{B}(\mathbf{x}) = \lambda \int_0^{\tau} \mathbf{I}_{\mathrm{t}}(\mathbf{x}) \mathrm{dt}, \tag{1}$$

where $\mathbf{B}(\mathbf{x}) \in \mathbb{R}^{\mathrm{W}\times\mathrm{H}\times 3}$ is the captured image, W and H are the width and height of the image respectively, $\mathbf{x} \in \mathbb{R}^2$ represents the pixel location, λ is a normalization factor, τ is the camera exposure time, $\mathbf{I}_{\mathrm{t}}(\mathbf{x}) \in \mathbb{R}^{\mathrm{W}\times\mathrm{H}\times 3}$ is the virtual sharp image captured at timestamp t within the exposure time. Motion in the camera during the exposure time will result in different virtual images $\mathbf{I}_{\mathrm{t}}(\mathbf{x})$ for each t, resulting in a blurred image $\mathbf{B}(\mathbf{x})$. The model can be discretely approximated as

$$\mathbf{B}(\mathbf{x}) \approx \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{I}_{\mathrm{i}}(\mathbf{x}), \tag{2}$$

where n is the number of discrete samples.

The amount of motion blur in an image thus depends on the motion during the exposure time. For shorter exposure time, the relative motion will be small even for a quickly moving camera. Conversely, for long exposure time (e.g. in low light conditions), even a slowly moving camera can result in a motion blurred image.

Direct image alignment with sharp images: Before introducing our direct image alignment algorithm with blurry images, we first review the original algorithm with sharp images. Direct image alignment algorithm serves as the core block for direct visual odometry approaches. It estimates the camera pose of current tracked frame by maximizing the photometric consistency between the latest keyframe and current frame. It can be formally defined as follows:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{i=0}^{m-1} \left\| \mathbf{I}_{\mathrm{ref}}(\mathbf{x}_i) - \mathbf{I}_{\mathrm{cur}}(\hat{\mathbf{x}}_i) \right\|_2^2, \tag{3}$$

where $\mathbf{T} \in \mathbf{SE}(3)$ is the transformation matrix from the reference image $\mathbf{I}_{\mathrm{ref}}$ to the current image $\mathbf{I}_{\mathrm{cur}}$, m is the number of sampled pixels for motion estimation, $\mathbf{x}_i \in \mathbb{R}^2$ is the location of the i^{th} pixel, $\hat{\mathbf{x}}_i \in \mathbb{R}^2$ is the pixel location corresponding to pixel $\mathbf{x}_i$ in current image $\mathbf{I}_{\mathrm{cur}}$. Robust loss function (e.g. huber loss) is usually also applied to the error residuals for robust pose estimation. The image points $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are related by the camera pose $\mathbf{T}$ and the depth d_i as

$$\hat{\mathbf{x}}_i = \pi(\mathbf{T} \cdot \pi^{-1}(\mathbf{x}_i, d_i)), \tag{4}$$

where $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the camera projection function, which projects point in 3D space to image plane; $\pi^{-1} : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^3$ is the inverse projection function, which transforms a 2D point from image to 3D space by back-projecting with the depth d_i. The formulation can also be extended to multi-frames, which can be used to jointly optimize the camera poses, 3D scene structures and camera intrinsic parameters (i.e. also known as photometric bundle adjustment).

Direct VO methods assume that photoconsistency (i.e. equation 3) holds for the correct transformation $\mathbf{T}$. However, if the images $\mathbf{I}_{\mathrm{ref}}$ and $\mathbf{I}_{\mathrm{cur}}$ are affected by different motion blur, the photoconsistency loss will no longer be valid since the local appearance for correctly corresponding points will differ. This scenario is unavoidable in settings with highly non-linear trajectories, e.g. tracking in augmented/mixed/virtual reality applications, which usually result in images with different levels of motion blur.

Motion trajectory modeling: To correctly compensate for the motion blur we need to model the local camera trajectory during the exposure time. One approach is to only parameterize the final camera pose and then linearly interpolate between the previous frame and the new estimate. From the interpolation we can then create the virtual images necessary to represent the motion blur, as in equation (2). However, this approach might fail for camera trajectories with very abrupt directional changes, which are quite common for hand-held and head-mounted cameras.

To ensure robustness, instead, we choose to parameterize the local camera trajectory independently of the previous frame. To be specific, we parameterize two camera poses, one at the beginning of the exposure $\mathbf{T}_{\text{start}} \in \mathbf{SE}(3)$ and one at the end $\mathbf{T}_{\text{end}} \in \mathbf{SE}(3)$. Between the two poses we linearly interpolate poses in the Lie-algebra of $\mathbf{SE}(3)$. The virtual camera pose at time $t \in [0, \tau]$ can thus be represented as

$$\mathbf{T}_t = \mathbf{T}_{\text{start}} \cdot exp(\frac{t}{\tau} \cdot log(\mathbf{T}_{\text{start}}^{-1} \cdot \mathbf{T}_{\text{end}})), \tag{5}$$

where τ is the exposure time. For more details on the interpolation and derivations of the related Jacobian, please see the supplementary material.

The goal of our motion blur-aware tracker is now to estimate both $\mathbf{T}_{\text{start}}$ and $\mathbf{T}_{\text{end}}$ for each frame. If the two poses are close, we know that the corresponding frame has very little motion blur. In this work we only considered linear interpolation between the two poses, but e.g. higher order splines could be used as well which could then represent more complex camera motions. However, in our experiments we found that the linear model worked well enough, since the exposure time is usually relatively short.

Direct image alignment with blurry images: Our motion blur-aware tracker works by performing direct alignment between the keyframe, which we assume is sharp, and the current frame which can suffer from motion blur. To leverage photometric consistency in the alignment, we thus need to either de-blur the new frame or re-blur the keyframe. In our work we chose the latter since re-blurring is in general easier and more robust compared to motion deblurring, especially for severe motion blurred images.

Each sampled pixel in $\mathbf{I}_{\text{ref}}$ with known depth can be transferred into the current (blurry) image $\mathbf{B}_{\text{cur}}$ using (4). For each projected point we select its nearest neighbour integer position pixel, in current blurry image. Assuming that the 3D point lies on a fronto-parallel plane (with respect to $\mathbf{I}_{\text{ref}}$), we can use this plane to transfer the selected pixel back into the reference view. Details can be found in Fig. 2. To synthesize the re-blurred pixel from the reference view (so that we can compare against the real captured pixel intensity), we now interpolate between $\mathbf{T}_{\text{start}}$ and $\mathbf{T}_{\text{end}}$. For each virtual view $\mathbf{T}_t$, which is uniformly sampled within

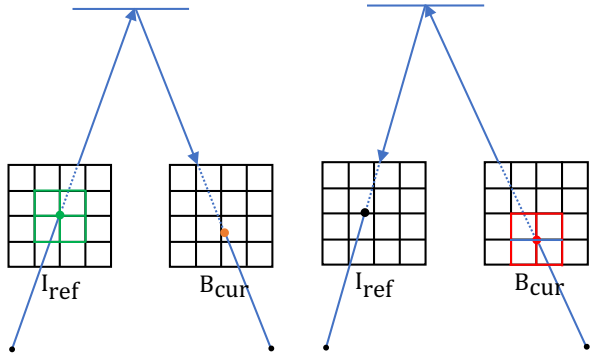


Figure 2. Pixel point transfer strategies. Note that we assume the pixel center lies at the grid intersection, e.g. the green grid is considered as a 3×3 patch.

$[0, \tau]$, we transfer the pixel coordinate (i.e. the red pixel in Fig. 2) back into the reference image and retrieve the image intensity values using bi-linear interpolation. The re-blurred pixel intensity is then created by averaging over the intensity values (as in (2)):

$$\hat{\mathbf{B}}_{\text{cur}}(\mathbf{x}) = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{I}_{\text{ref}}(\mathbf{x}_{\frac{i\tau}{n-1}}), \tag{6}$$

where $\mathbf{x}_{\frac{i\tau}{n-1}} \in \mathbb{R}^2$ corresponds to the transferred point at time $t = \frac{i\tau}{n-1}$ in the sharp reference frame, n is the number of virtual frames used to synthesize the blurry image [1]. The tracker then optimizes over the start-pose and end-pose to minimize the photoconsistency loss between the real captured intensities in current frame and the synthesized pixel intensities from the reference image (i.e. via re-blurring),

$$\mathbf{T}^*_{\text{start}},\ \mathbf{T}^*_{\text{end}} = \underset{\mathbf{T}_{\text{start}},\ \mathbf{T}_{\text{end}}}{\text{argmin}} \sum_{i=0}^{m-1} \left\| \mathbf{B}_{\text{cur}}(\mathbf{x}_i) - \hat{\mathbf{B}}_{\text{cur}}(\mathbf{x}_i) \right\|_2^2. \tag{7}$$

In practice, most direct image alignment methods use local patches for better convergence. Different from direct image alignment algorithm for sharp images, which usually selects the local patch from the reference image (e.g. the green 3×3 grid on the left of Fig. 2), we instead select the local patch from the current blurry image (e.g. the red 3×3 grid on the right of Fig. 2) since this simplifies the re-blurring step of our pipeline.

More details on the transfer: To further demonstrate the relationship between $\mathbf{x} \in \mathbb{R}^2$ and $\mathbf{x}_{\frac{i\tau}{n-1}} \in \mathbb{R}^2$ from Eq. (6), we define following notations for the ease of illustration. We denote the depth of the fronto-parallel plane as d, which is the estimated depth of the corresponding sampled keypoint from $\mathbf{I}_{\text{ref}}$ (i.e. the green pixel in Fig. 2); we further denote the camera pose of the virtual frame $\mathbf{I}_i$ captured at

[1] We use a fixed number of virtual frames in our experiments. However, it can be dynamically changed according to the level of blur to save computational resources.

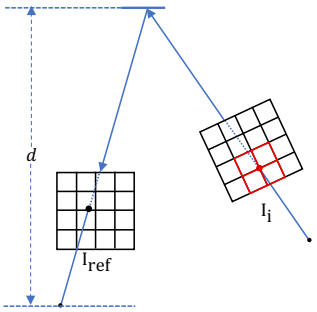


Figure 3. Geometric relationship between $\mathbf{x} \in \mathbb{R}^2$ (i.e. the red pixel) of the virtual sharp image $\mathbf{I}_i$ and $\mathbf{x}_{\frac{i\tau}{n-1}} \in \mathbb{R}^2$ (i.e. the black pixel) of the reference image $\mathbf{I}_{\text{ref}}$.

timestamp $\frac{i\tau}{n-1}$ relative to the reference keyframe $\mathbf{I}_{\text{ref}}$ as $\mathbf{T}_i \in \mathbf{SE}(3)$, which can be computed from Eq. (5) as

$$\mathbf{T}_i = \mathbf{T}_{\text{start}} \cdot exp(\frac{i}{n-1}\tau \cdot log(\mathbf{T}_{\text{start}}^{-1} \cdot \mathbf{T}_{\text{end}})), \quad (8)$$

where $\mathbf{T}_{\text{start}} \in \mathbf{SE}(3)$ and $\mathbf{T}_{\text{end}} \in \mathbf{SE}(3)$ are the relative camera poses (which are defined from the current camera coordinate frame to the reference camera coordinate frame) of the current blurry image, at the beginning and end of the image capturing respectively, τ is the camera exposure time. Note that the fronto-parallel plane is defined in the reference camera frame, it might not be fronto-parallel with respect to the i^{th} virtual camera frame. To avoid confusion, we illustrate the relationship in Fig. 3. By proper algebraic manipulations, we can obtain $\mathbf{x}_{\frac{i\tau}{n-1}}$ as

$$\mathbf{x}_{\frac{i\tau}{n-1}} = \pi(\mathbf{T}_i \cdot \mathbf{p}_{3\text{d}}), \quad (9)$$

$$\mathbf{p}_{3\text{d}} = \frac{d - p_z}{\lambda} \begin{bmatrix} x, & y, & z \end{bmatrix}^{\text{T}}, \quad (10)$$

$$\lambda = 2x \cdot q_0 + 2y \cdot q_1 + z \cdot q_2 \quad (11)$$

$$q_0 = q_x q_z - q_w q_y, \quad (12)$$

$$q_1 = q_x q_w + q_y q_z, \quad (13)$$

$$q_2 = q_w^2 - q_x^2 - q_y^2 + q_z^2, \quad (14)$$

$$\begin{bmatrix} x, & y, & z \end{bmatrix}^{\text{T}} = \pi^{-1}(\mathbf{x}), \quad (15)$$

where $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the camera projection function, (q_w, q_x, q_y, q_z) is the quaternion representation of the rotation matrix of $\mathbf{T}_i$ and (p_x, p_y, p_z) is the translation vector of $\mathbf{T}_i$, d is the depth of the plane with respect to the reference key-frame, $\pi^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the camera back projection function such that $x^2 + y^2 + z^2 = 1$. Detailed algebraic derivations as well as related Jacobian can be found in our supplementary material.

4. Datasets

In this section we give an overview of the datasets that we consider in our experimental evaluation. While there are many different datasets for evaluating visual odometry methods, we found that there is no suitable dataset that specifically targets at motion blurred images, although some datasets have sub-sequences that contain motion blur, e.g. in the ETH3D SLAM Benchmark [29] and TUM RGB-D [31].

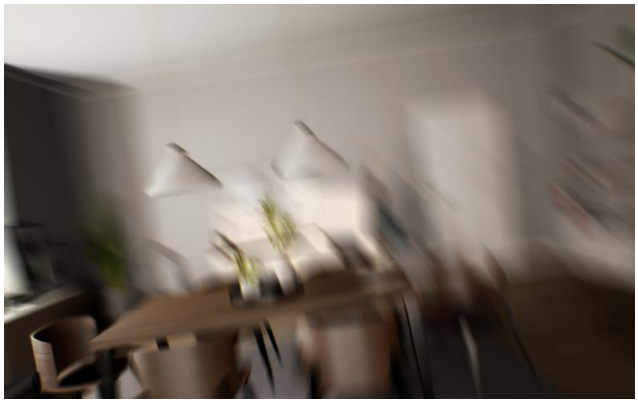


Figure 4. *Sample image from our synthetic ArchVizInterior dataset*. The camera motions are taken from the ETH3D Benchmark [29] which are then re-rendered in synthetic scene by Unreal game engine.

ETH3D [29] / ArchVizInterior: In the ETH3D SLAM benchmark [29], the image sequences; *camera_shake_1*, *camera_shake_2* and *camera_shake_3* have severe motion blur. The three sequences were captured with a camera being quickly shaken back and forth. In addition to the motion blur, the sequences are difficult due to the very poorly textured scene (mainly containing a white circular table with very few distinguishing landmarks). We experiment with both DSO [7] and ORBSLAM [23] on these sequences and find that both methods fail to initialize on this dataset.

To investigate if the failures are due to the poorly textured scene or the motion blur, we render a synthetic photo-realistic dataset using the same motion trajectories, exposure time and frame rate. The dataset is rendered by the Unreal game engine with the free ArchVizInterior scene model[2]. We use the scripts provided by Liu et al. [18] for the dataset creation. A sample image can be found in Fig. 4. Sample videos on the dataset can be found in our supplementary material. Since this dataset provides perfect ground truth sharp images, which are paired with the motion blurred images, we use it for our ablation studies.

TUM-RGBD [31]: The hand-held SLAM sequences from the TUM-RGBD dataset [31] also contain motion blurred images. The dataset is collected with the Microsoft Xbox Kinect sensor, which contains a rolling shutter color camera and a time-of-flight depth camera. Since the dataset targets at evaluating the performance of RGBD camera based SLAM methods, the effect of motion blurred images is not their focus. Furthermore, it has been shown that direct approach is more sensitive to rolling shutter effect [29, 36].

[2]https://www.unrealengine.com

It is thus better to have dataset which is collected from a global shutter camera to avoid the effect of rolling shutter mechanism. We also evaluate our method with the TUM-RGBD dataset in the next section.

Proposed Motion Blur Benchmarking Dataset: To more clearly show the benefit of our method we propose a new benchmark dataset for evaluating visual odometry which specifically targets motion blur. By making this dataset publicly available to other researchers, we hope to encourage further research on making visual odometry robust.

Our dataset is collected with a global shutter camera at a resolution of 752×480 pixels and a frame rate of 27 fps. The ground truth trajectory is provided by an indoor motion capturing system[3] at 100 Hz. Extrinsic parameters between the marker for motion capture and camera is calibrated by the hand-eye calibration approach, such that the ground truth trajectory can align with the camera motion trajectory. A total number of 18 motion blurred sequences are collected. The dataset contains images with varying levels of motion blur. More details on the dataset can be found in the supplementary video. Figure 5 shows some example images from the new dataset with varying motion blur.

5. Experimental Evaluation

Implementation details: The original tracker of DSO [7] does semi-dense direct image alignment. For efficiency, we sub-sample the high gradient pixels to obtain sparse keypoints, which are uniformly distributed within the image. We further define a 9×9 local patch around each sampled sparse keypoints for better convergence. The energy function is optimized in a coarse to fine manner and robust huber loss function is also applied for robustness. Our tracker is implemented and evaluated on a laptop grade Nvidia RTX 2080 graphic card. It takes 34.4 ms on average to process a single blurry image, which is suitable for real time applications.

We experiment with two state-of-the-art deblurring networks, SRNDeblurNet [32] and DeblurGANv2 [16]. We use the official pretrained models and generalize them to our datasets without any finetuning. In particular, we consider the mobile network of DeblurGANv2 since it can run in real time on a high-end GPU. SRNDeblurNet takes around 140 ms second to process a single image at a resolution of 752×480 pixels on a laptop grade Nvidia RTX 2080 graphic card. However, it delivers higher quality deblurred images compared to the DeblurGANv2 mobile network and the time consumption is already sufficient for local mapping. There are also more advanced deblurring networks being proposed recently, such as the work from Gao et al. [9]. However, they are usually more time consuming than SRNDeblurNet [32], and is not suitable to be integrated into our local mapper.

Baseline methods and evaluation metrics: Two state-of-the-art monocular VO pipelines are selected for the benchmark. In particular, we select ORB-SLAM [23], which is the representative of sparse feature based approach. As a representative for direct approaches we compare with DSO [7]. For quantitative comparisons, we measure the RMSE of absolute trajectory error (i.e. RMSE ATE) [31], since it is the focus of VO algorithms and is commonly used by the literature [7, 29]. The estimated trajectory is first aligned with the ground truth by matching poses with the same timestamps. The RMSE of ATE is then computed by averaging the translational differences between the aligned trajectories. In addition, we also use the percentage of frame drops to measure the robustness of different algorithms.

Motivating example: To clearly motivate the need for motion blur aware VO, we first evaluate the performance of ORB-SLAM and DSO on the sharp images and the corresponding motion blurred images respectively, on the ArchVizInterior dataset. See Section 4 for details on the dataset. Table 1 demonstrates that motion blurred images affect both ORB-SLAM and DSO, in terms of estimated trajectory accuracy and robustness. Although there is no large accuracy drop for ORB-SLAM, there are significant frame drops. ArchVizInterior dataset collects images around a local area and scene overlaps exist among almost every image. Even though there are many frame drops, ORB-SLAM can still recover back due to its relocalization module, once the image is in better quality, assuming there is enough visual overlap with previously mapped areas. However, a reset might need to perform if the camera tranverses in unexplored scenes. There is no frame drops for DSO. However, its accuracy drops with a large margin compared to that with sharp images.

Note that we rendered the dataset with the same trajectories from ETH3D [29] dataset. The results demonstrate that the camera motion is not the main factor, which leads the failure of both ORB-SLAM and DSO on ETH3D dataset. It further justifies our motivation to create new datasets.

Ablation studies: Since ArchvizInterior dataset has ground truth sharp images, which are paired with the motion blurred images, we conduct ablation studies with it for better comparisons. Our ablation studies consist of two parts, the selection of the deblurring network and experiments to demonstrate the effectiveness of our motion blur aware tracker.

We evaluate the generalization performance as well as efficiency of both deblurring networks, i.e. SRNDeblurNet [32] and DeblurGANv2-mobileNet [16] on the ArchvizInterior dataset. The evaluation is conducted with a laptop grade Nvidia RTX 2080 graphic card. Table 2 demonstrates that the DeblurGANv2-mobileNet is able to run in real time

[3] https://www.vicon.com

Figure 5. Examples images from the proposed dataset for benchmarking visual odometry from motion blurred image sequences. The dataset contains multiple sequences with varying levels of motion blur.

	ORB-SLAM [23]						DSO [7]					
	ArchViz-1		ArchViz-2		ArchViz-3		ArchViz-1		ArchViz-2		ArchViz-3	
	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)
Sharp	0.020	0	0.005	0	0.014	0	0.020	0	0.004	0	0.014	0
Blur	0.033	22.1	0.012	1.1	0.101	19.5	0.213	0	0.166	0	0.129	0
Deblur	0.018	15.6	0.007	2.8	0.020	16.7	0.207	0	0.161	0	0.048	0

Table 1. *The performance of both ORB-SLAM and DSO on the ArchVizInterior dataset*. *Sharp*, *Blur* and *Deblur* denote the pipeline is running on the ground truth sharp images, motion blurred images and the deblurred images by DeblurGANv2 [16] respectively. The FD column shows the percentage of dropped frames. Both the ATE and FD metrics are the smaller the better.

	PSNR (dB) ↑	SSIM ↑	Time (ms)
Blur image	26.80	0.7887	N.A.
DeblurGANv2m [16]	28.66	0.8156	38.1
SRNDeblurNet [32]	30.01	0.8491	140.3

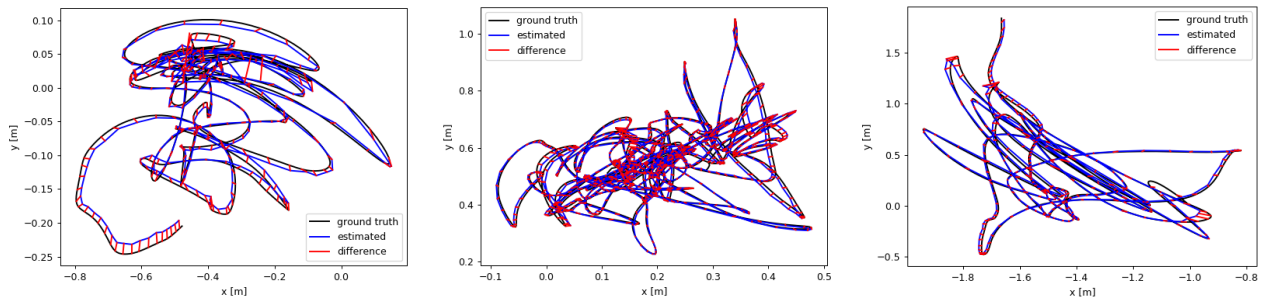
Table 2. Generalization performance of DeblurGANv2-mobileNet [16] and SRNDeblurNet [32] on the ArchVizInterior dataset.

on a high-end GPU. However, its deblurring performance is worse than SRNDeblurNet as an expense. To verify if current performance of DeblurGANv2-mobileNet is sufficient to improve the performance of VO algorithms, we deblurred every image of ArchVizInterior dataset by DeblurGANv2-mobileNet. We run both ORB-SLAM and DSO with the deblurred images. The experimental results shown in Table 1 demonstrate that it can only improve the performance of VO algorithms with motion blurred images with a small margin. The reason is that the DeblurGANv2 [16] has limited generalization performance as an expense for smaller model size. It demonstrates that the naive way to deblur every input frame (with an efficient deblurring network for real time operation), and feed the deblurred images to a standard VO pipeline is not the correct way to make VO robust to motion blur. It justifies our motivation to do hybrid motion blur aware VO, which can take advantage of a more powerful deblurring network with a larger model size. Our hybrid approach recovers the camera motion of severe blurred images by the motion blur aware direct image alignment algorithm, without the need to deblur them in frame rate. Since SRNDeblurNet takes around 140 ms to process a 752×480 pixels resolution image, which is sufficient to deblur selected key-frame image, and delivers better deblurring performance, we thus use it for our pipeline.

To study the effectiveness of MBA-VO, we experiment with sharp images and blurry images respectively. Experimental results from Table 3 demonstrate that MBA-VO is able to achieve similar performance as both ORB-SLAM and DSO if the images are not motion blurred. For motion blurred images, MBA-VO is able to achieve competitive accuracy as that for sharp images without any frame drops. To further demonstrate the effectiveness of our motion blur aware tracker, we set the camera exposure time to be 0 during the estimation of the camera poses (i.e. Eq. (7)). It enforces the tracker to assume the current blurry images as sharp images and do normal pose estimations instead. The other settings are kept the same (e.g. we still use SRNDeblurNet to deblur the keyframe images). The resulted ATE metrics are 0.22 m, 0.1558 m and 0.2113 m respectively for the ArchVizInterior dataset. The experimental results thus demonstrate the necessity to do motion blur aware tracking.

Fig. 6 demonstrates the estimated trajectories of MBA-VO on the motion blurred sequences from ArchVizInterior dataset. Both the quantitative and qualitative results demonstrate the effectiveness of our proposed algorithm for motion blurred image sequences.

Evaluation with TUM RGB-D dataset: To evaluate the performance of MBA-VO with real motion blurred images, we select three sequences with large motion blur from the TUM RGB-D dataset [31]. In particular, we select the *fr1-desk*, *fr1-desk2* and *fr1-room* from the handheld SLAM category. Since the camera is hand held, which is similar to head-mounted camera, hand shaken would result in fast rotational motion though the translational velocity is small.



ArchViz-1 ArchViz-2 ArchViz-3

Figure 6. Estimated trajectories of MBA-VO from the motion blurred image sequences of the ArchVizInterior dataset. It demonstrates that MBA-VO can estimate accurate trajectories, although the camera motions are very challenging.

	ArchViz-1		ArchViz-2		ArchViz-3	
	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)
Sharp	0.020	0	0.010	0	0.016	0
Blur	0.026	0	0.018	0	0.020	0

Table 3. *The performance of MBA-VO on the ArchVizInterior dataset*. *Sharp* and *Blur* denote the pipeline is running on the ground truth sharp images and motion blurred images respectively. FD is the percentage of dropped frames.

	fr1-desk		fr1-desk2		fr1-room	
	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)
ORBSLAM	0.178	5.1	**0.301**	33.8	**0.066**	46.5
DSO	0.496	0	0.776	0	0.299	0
MBA-VO	**0.102**	0	0.399	0	0.144	0

Table 4. *Comparison on TUM RGB-D dataset [31]*. ORB-SLAM suffers from significant frame drops, although it provides accurate estimates. The proposed method, MBA-VO, improves on DSO and provides more accurate estimates with no frame drops.

For augmented/virtual/mixed reality applications, head rotational motion is the main cause of severe motion blur.

Table 4 demonstrates the performance of MBA-VO against ORB-SLAM and DSO on sequences with large motion blur from TUM RGB-D dataset. It demonstrates that MBA-VO is able to improve the accuracy over the original DSO algorithm, while is also more robust compared to sparse feature based approach, with motion blurred images. Note that ORB-SLAM suffers from significant frame-drops, although it generally provides more accurate poses for these sequences (low ATE).

Evaluation with our real-world dataset: Since the goal of the TUM RGB-D dataset is not evaluating the robustness of monocular VO/SLAM algorithms, we created a specific large real dataset with varying levels of motion blur (see Section 4). We evaluate ORB-SLAM, DSO and MBA-VO with it. Due to space limit, we present the experimental results from a subset of the datasets in Table 5. More experimental results can be found from our supplementary material. The experimental results illustrate that ORB-SLAM has significant frame drops in general for all the sequences, while it usually is more accurate. Compared to DSO, which has no frame drops, MBA-VO has better accuracy. In general, MBA-VO achieves better robustness and accuracy, compared to both DSO and ORB-SLAM on the real motion blurred image sequences.

	ORB-SLAM [23]		DSO [7]		MBA-VO	
	ATE (m)	FD (%)	ATE (m)	FD (%)	ATE (m)	FD (%)
Seq0	0.127	7.0	0.272	0	**0.058**	0
Seq1	0.084	36.8	0.433	0	**0.069**	0
Seq2	0.199	11.9	0.196	0	**0.045**	0
Seq3	x	x	0.404	0	**0.162**	0
Seq4	x	x	x	x	**0.132**	0

Table 5. *The performance of MBA-VO on our dataset*. x denotes the corresponding algorithm fails on that particular sequence. It demonstrates that MBA-VO improves the accuracy of DSO, while being robust to motion blur with no frame drops.

Discussions: The experimental results demonstrate that DSO [7] generally performs worse than ORB-SLAM [23] for motion blurred images, in terms of the ATE metric. It is caused by the datasets we evaluated on have many more severely blurred images. In this case, ORB-SLAM [23] simply discards the severe blurred images without affecting the accuracy of the remaining frames. In contrast, DSO [7] does not drop the frames, leading to the overall loss of accuracy due to including more challenging frames in the estimation. Note that the ATE metric is only computed from the successfully tracked frames for ORB-SLAM.

6. Conclusion

We present a hybrid visual odometry algorithm which is robust to motion blur. We also propose a novel benchmarking dataset targeting motion blur aware visual odometry. Experimental results demonstrate that our algorithm improves the accuracy and robustness over existing methods on both synthetic and real-world datasets. We believe both our method and dataset would be a valuable step towards the era of robust visual odometry.

References

[1] José-Luis Blanco-Claraco, Francisco-Ángel Moreno-Dueñas, and Javier González-Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *International Journal of Robotics Research (IJRR)*, 2014. 3

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM: learning a compact, optimisable representation for dense visual SLAM. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[3] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *International Journal of Robotics Research (IJRR)*, 2016. 3

[4] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 2016. 2

[5] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *International Journal of Robotics Research (IJRR)*, 2016. 3

[6] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision (ICCV)*, 2003. 2

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2017. 1, 2, 3, 5, 6, 7, 8

[8] Jacob Engel, Thomas Schops, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular slam. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[9] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[10] Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. Ldso: Direct sparse odometry with loop closure. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 3

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, 2014. 2

[13] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *International Conference on Robotics and Automation (ICRA)*, 2014. 3

[14] Michal Hradiš, Jan Kotera, Pavel Zemčík, and Filip Šroubek. Convolutional neural networks for direct text deblurring. In *British Machine Vision Conference (BMVC)*, 2015. 2

[15] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[16] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 6, 7

[17] Hee Seok Lee, Junghyun Kwon, and Kyoung Mu Lee. Simultaneous localization, mapping and deblurring. In *International Conference on Computer Vision (ICCV)*, 2011. 2

[18] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[19] Peidong Liu, Marcel Geppert, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. Towards robust visual odometry with a multi-camera system. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[20] Peidong Liu, Lionel Heng, Torsten Sattler, and Marc Pollefeys. Direct visual odometry for a fisheye-stereo camera. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017. 2

[21] Peidong Liu, Joel Janai, Marc Pollefeys, Torsten Sattler, and Andreas Geiger. Self-supervised linear motion deblurring. In *IEEE Robotics and Automation Letters (RAL)*, 2020. 2

[22] András L Majdik, Charles Till, and Davide Scaramuzza. The zurich urban micro aerial vehicle dataset. *International Journal of Robotics Research (IJRR)*, 2017. 3

[23] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1, 2, 5, 6, 7, 8

[24] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[25] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[26] Seonwook Park, Thomas Schöps, and Marc Pollefeys. Illumination change robustness in direct visual slam. In *International Conference on Robotics and Automation (ICRA)*, 2017. 3

[27] Bernd Pfrommer, Nitin Sanket, Kostas Daniilidis, and Jonas Cleveland. Penncosyvio: A challenging visual inertial odometry benchmark. In *International Conference on Robotics and Automation (ICRA)*, 2017. 3

[28] Alberto Pretto, Emanuele Menegatti, Maren Bennewitz, Wolfram Burgard, and Enrico Pagello. A visual odometry framework robust to motion blur. In *International Conference on Robotics and Automation (ICRA)*, 2009. 2

[29] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 5, 6

[30] David Schubert, Nikolaus Demmel, Lukas von Stumberg, Vladyslav Usenko, and Daniel Cremers. Rolling-shutter

modelling for direct visual-inertial odometry. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019. 2

[31] Jurgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012. 3, 5, 6, 7, 8

[32] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 6, 7

[33] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[34] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[35] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *Neural Information Processing Systems (NIPS)*, 2014. 2

[36] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. In *IEEE Robotics and Automation Letters (RAL)*, 2018. 5

[37] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson W.H. Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[38] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[39] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[40] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2