Embedded Dense Camera Trajectories in Multi-Video Image Mosaics by Geodesic Interpolation-based Reintegration

Lars Haalck, Benjamin Risse
University of Münster
{lars.haalck, b.risse}@uni-muenster.de

Abstract

Dense registrations of huge image sets are still challenging due to exhaustive matchings and computationally expensive optimisations. Moreover, the resultant image mosaics often suffer from structural errors such as drift. Here, we propose a novel algorithm to generate global large-scale registrations from thousands of images extracted from multiple videos to derive high-resolution image mosaics which include full frame rate camera trajectories. Our algorithm does not require any initialisations and ensures the effective integration of all available image data by combining efficient and highly parallelised key-frame and loop-closure mechanisms with a novel geodesic interpolation-based reintegration strategy. As a consequence, global refinement can be done in a fraction of iterations compared to traditional optimisation strategies, while effectively avoiding drift and convergence towards inappropriate solutions. We compared our registration strategy with state-of-the-art algorithms and quantitative evaluations revealed millimetre spatial and high angular accuracy. Applicability is demonstrated by registering more than $110{,}000$ *frames from multiple scan recordings and provide dense camera trajectories in a globally referenced coordinate system as used for drone-based mappings, ecological studies, object tracking and land surveys.*

1. Introduction

Registering multiple images taken at different times, viewpoints and/or with different sensors into a common coordinate system is a classical problem in computer vision [55]. Given the multitude of use-cases for registered images, including robotics, remote sensing, cartography, medicine and ecology, it is no surprise that a huge variety of different registration strategies have been developed. Despite this variety of applications, the majority of algorithms utilise the same three steps, namely feature detection, feature matching and image transformation calculations [48]. These steps can be complemented by an optional stitching step which combines the images based on these transformations [50].

Historically, algorithms for image registration are separated into planar image registration (camera scans over a more or less planar scene using mainly translational motion), panoramic image registration (camera rotates around its optical centre to create a viewing sphere), or arbitrary scene reconstruction techniques (neither camera motion nor underlying geometry is constrained) [47]. Planar and panoramic image registration techniques result in two-dimensional image mosaics [37, 32], whereas arbitrary scene reconstruction algorithms aim to derive the three-dimensional structure of the scene, hence also called structure from motion (SfM) algorithms [43, 54]. Moreover, these algorithms can be separated based on their real-time capabilities in either sequential (local) or global registration strategies [6].

Unfortunately, the optimisation problem of the registration tasks inevitably suffers from ambiguities [36] so that image mosaicing techniques are usually constrained in terms of their underlying geometric transformations [47] or complemented by additional sensor readings like camera motion estimates (i.e. extrinsic camera parameters) [29, 49]. Also, global SfM strategies as well as simultaneous localisation and mapping (SLAM) algorithms can not guarantee drift-free results in very long monocular recordings due to so-called critical configurations [51]. Examples for critical configurations are recordings in which the image plane is parallel to the ground plane (e.g. as in aerial scan surveys), videos with strong lens distortion (e.g. when using a wide angle lens), or imaging conditions in which the camera moves continuously in the direction of its optical axis (e.g. in autonomous driving applications). Even though, 2D image mosaics bypass critical configurations they still suffer from perspective drift, inappropriate initialisations and long computational times [50]. Moreover, existing algorithms capable of processing continuous videos rely on preceding key-frame selection strategies resulting in irregular temporal samplings and the loss of intermediate camera positions.

1.1. Related Work

Two-dimensional image registration is one of the traditional problems in computer vision. Early work mainly focused on the geometric properties between a few frames and was of a more theoretical nature [3, 48, 5]. Later, image registration was applied to a variety of different use-cases such as document scanning [38, 27] and required more and more computational resources to identify the spatial correspondences between growing sets of images [55].

In particular, for translational scan applications, the number of frames quickly exceeded the resources available since the camera motion is not restricted to rotational transformations [50]. Therefore, a variety of domain-specific heuristics and strategies were proposed to stitch large-scale mosaics for all kinds of applications. For example, local optimisation strategies were introduced to enable close to real-time robotic applications [39, 4]. Others used adjusted registration techniques to stitch two or more image streams from temporally synchronised cameras [14, 33, 25]. Medical imaging used registration techniques to generate high-resolution reconstructions from multiple endoscopic images featuring an extremely small field of view [30, 44], to generate high-resolution imagery of the human skin [10], or to compute retinal image mosaics [20]. Similarly, microscopic image data has been stitched to generate high-resolution reconstructions for biological applications [40]. Novel data acquisition strategies such as recordings done by Unmanned Aerial Vehicles (UAVs) or by submarines also gave rise to a variety of registration algorithms specifically targeting the integration of huge image datasets for survey applications [7, 28, 9]. For example, Randall *et al*. used pairwise stitching to produce video mosaics for underwater applications using super-pixel and dominant camera motion heuristics [26]. Similarly, UAV-based recordings have been used to monitor wildlife densities for ecological purposes [29]. In a different approach, UAV recordings were used for solar plant reconstruction applications [54, 23].

Given the recent success of deep learning methods in computer vision, these techniques have also been applied to image registration [21]. In particular, convolutional neural networks (CNNs) are used in order to register images in medical [22, 45], stereo imaging [53, 13] and remote sensing applications [52, 15].

Most of the above mentioned global approaches are limited to a few hundred frames. This is also true for most deep learning-based registration techniques which have been surveyed recently using only up to 80 images at a 1000×1000 pixel resolution for benchmarking purposes [21]. In contrast, Ferrer *et al*. were among the first who pushed the boundaries by one order of magnitude [11]. Even though the authors report the successful stitching of up to 20,000 frames, external sensors were used to precisely initialise the camera extrinsics, making this approach infeasible in applications in which no additional camera motion information is available. Moreover, the algorithm was not tuned towards computational efficiency and does not ensure the integration of all frames leading to sparse camera trajectories with temporally irregular distributions [11]. This drawback is particularly apparent if video data is used which intrinsically increases the number of images due to potentially high frame rates. Consequently, processing video data for wide-view high-resolution image generation such as landscape surveys still suffers greatly from incorrect registration estimates [19].

Effective global registration of thousands of frames without the integration of additional external cues is still an unsolved problem in computer vision (see recent survey [50]). In fact, the authors explicitly claim the demand for scalable image registration techniques which do not suffer from errors due to incremental transformations [50]. And since key-frame based registration algorithms do not reintegrate intermediate camera positions, none of the existing techniques can be used to extract per-frame image (e.g. drone) locations solely based on image data or to continuously compensate the camera motion for object tracking purposes where detections in every frame are projected using the respective transformation.

1.2. Contribution

In this paper, we propose a scalable and fast multi-video registration algorithm which enforces the usage of all frames from continuous recordings and thus provides dense camera trajectories within the reconstruction. Our strategy combines two superimposing processing steps, namely a highly optimised key-frame selection and registration approach with a novel reintegration technique. In summary the first step utilises (i) multi-video integration by using an inter-video maximum spanning tree, (ii) cascaded hashing-based sparse key-frame selection which is optimised for video sequences, and (iii) non-linear global optimisation on the resultant key-frames. Subsequently, intermediate images (i.e. not included in the key-frames) are incorporated by using geodesic interpolation-based reintegration. This ensures invertible and smooth similarity transformations so that the subsequent final optimisation does not suffer from local or ill-defined optima.

Our algorithm has several advantages compared to existing methods. Firstly, our approach does not require any additional cues such as camera intrinsics or extrinsics to generate highly accurate large-scale image registrations and mosaics. Secondly, the separation allows for unprecedented parallelisations and efficient graph-based computations which drastically reduces the computational complexity of the underlying optimisation problem. Thirdly, full frame-rate 2.5D camera motion trajectories are available which guarantee equitemporal sampling according to

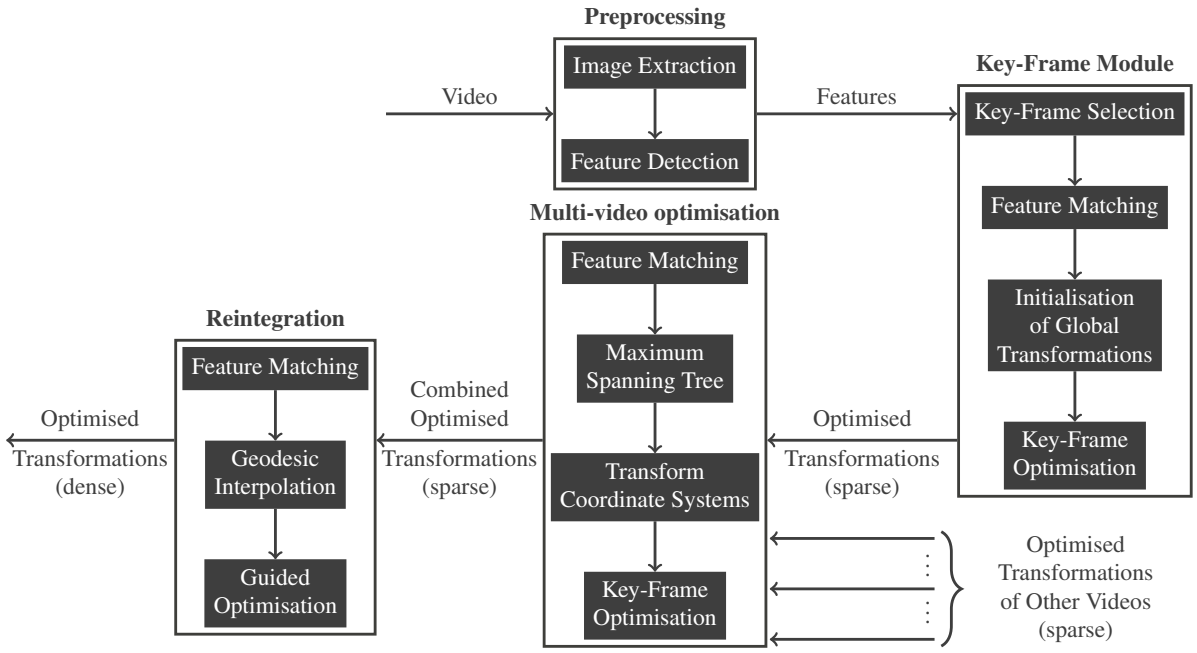


Figure 1: Structure of the algorithm involving four processing modules preprocessing, key-frame selection, multi-video optimisation and reintegration. For details see text.

the sampling rate of the cameras without skipping frames. Finally, the full potential of all available frames from all videos is used to alleviate potential drift and other errors.

In a series of experiments, we demonstrate that our algorithms outperforms existing global registration strategies in both, accuracy and computational time while avoiding critical configurations by design. Additional quantitative evaluations reveal millimetre spatial accuracy and a median angular error of 3° with a median absolute deviation of 3°. Scalability of our algorithm is further demonstrated by registering more than 110,000 frames from multiple videos in about three hours and 30 minutes without suffering from strong drift, erroneous initialisations or suboptimal key-frames.

2. Method

The proposed algorithm consists of four modules, namely preprocessing, key-frame selection, multi-video optimisation and reintegration and is outlined in Figure 1.

2.1. Preprocessing

In the preprocessing stage, a video is extracted into a stack of images and feature points are detected in every image. Let $\mathcal{I} = \{I_1, \dots, I_N\}$ be the set of N images and let $\mathcal{F} = \{F_1, \dots, F_N\}$ be their corresponding feature vectors, where $F_i \subset \mathbb{R}^2$ is a set of 2D feature points. The images are extracted from the same video so we can assume $I_i \in \mathbb{R}^{n \times m}$ for all $i \in \{1, \dots, N\}$ and some $n, m \in \mathbb{N}$.

2.2. Key-Frame Module

In order to significantly reduce the number of processed frames in the later defined optimisation problems, a small representative subset $\tilde{\mathcal{I}} = \{I_{\kappa(1)}, \dots, I_{\kappa(K)}\} \subset \mathcal{I}$ of size K is selected in the key-frame module, where $\kappa(\cdot)$ maps an index of a key-frame to an index in the full image set. To improve readability, we will use the tilde notation $\tilde{I}_i$ instead of $I_{\kappa(i)}$, $\tilde{F}_i$ instead of $F_{\kappa(i)}$ and $\tilde{M}_{i,j}$ instead of $M_{\kappa(i),\kappa(j)}$. Based on these key-frames, an optimisation problem is solved. Given sparse global transformations for the key-frames, the reintegration module integrates the frames omitted in the key-frame selection stage to obtain dense transformations as described in Section 2.4.

Key-frames are selected based on two criteria: they should have a good overlap for the resulting mosaic and enough feature matches to guide the optimisation robustly to a satisfying solution. To achieve this, two thresholds are predefined, marking the minimal and maximal desired pixel shift between matching feature points (e.g. 30% and 50% shift with respect to the image dimensions). All frames in this shift window are extracted and the frame with the highest amount of feature matches satisfying a similarity transformation is defined as the next key-frame [16]. The algorithm checks possible candidates of successive frames in parallel and stops when no more feature matches are found or the shift is too high. To be robust against blurry frames (e.g. through motion blur) or extreme parallax leading to poor matches and subsequently poor similarities, the upper

threshold is extended dynamically to guarantee sufficient matches in these cases.

Afterwards, feature matching is performed on these key-frames only using the method described in [18] which employs cascaded hashing of ORB features [41] as it is very efficient and reliably detects loop closure candidates. To ensure local feature matches between successive key-frames, we also match these frames in a sliding window approach. This ensures, that a point of revisiting (i.e. loop) in the scene is sufficiently covered by feature matches as well as guaranteeing good local matches for later stitching. Let $M_{i,j} \subset F_i \times F_j$ be the matches with respect to a similarity between features of two different images I_i and I_j. Let $T_{i,j} \in \mathbb{R}^{3\times3}$ be this (homogeneous) similarity transformation warping corresponding feature points from image I_i to image I_j for some i and j parametrised by a simple 2D translation, a uniform scaling factor and some rotation angle and determined by using RANSAC [12]. This transformation warps feature points f_i into another image I_j according to $\hat{f}_i = T_{i,j} f_i = g\left(f_i; T_{i,j}\right)$. Afterwards, we define global transformations in a unified coordinate system to formulate an optimisation problem. In contrast to the pairwise transformations above, the global transformations encode a mapping from image I_i to I_1 for some $i \in \{2, \ldots, N\}$. They are initialised by concatenating pairwise transformations and later refined by non-linear optimisation.

This initialisation suffers from a multiplicative error, accumulating over time and leading to severe drifts in the resulting mosaic, where points of revisiting do not align (Figure 2a). Therefore, an optimisation problem is solved, which distributes alignment errors across all images and effectively closed loops in the resulting mosaic.

The optimisation problem is then formulated as

$$\min J\left(\tilde{T}_1, \ldots, \tilde{T}_K\right) = \sum_{1\le i\le K} \sum_{i\le j\le K} \sum_{(f_k, f_l)\in \tilde{M}_{i,j}} \underbrace{\|g(f_k; \tilde{T}_j^{-1}\tilde{T}_i) - f_l\|_2^2 + \|g(f_l; \tilde{T}_i^{-1}\tilde{T}_j) - f_k\|_2^2}_{= h\left(f_k, f_l; \tilde{T}_i, \tilde{T}_j\right)}, \quad (1)$$

minimizing the (symmetric) reprojection error for every image pair $\tilde{I}_i$ and $\tilde{I}_j$ by rewriting the pairwise transformation $\tilde{T}_{i,j}$ as $\left(\tilde{T}_j^{-1}\tilde{T}_i\right)$ in the new global setting, where the symmetric error is used for stability [48]. This non-linear optimisation problem can be solved by using the Levenberg-Marquardt algorithm and results in drift-aware global key-frame registrations (Figure 2b) and is implemented using the Ceres Solver in this pipeline [1].

2.3. Multi-Video Optimisation

In a multi-video scenario, each video is first processed separately using the preprocessing and key-frame module as described above, leading to optimal transformations for all key-frames in each video. Next we build a graph, where each node in the tree represents a video. Two nodes share an edge if there is a frame in the first and a frame in the second video with matching feature points, where the number of matching feature points defines the weight of this edge, thus multiple edges between the same two videos are possible. For this inter-video matching step, we again use the method described in [18]. We now build a maximum spanning tree with respect to the number of feature matches given this graph which is subsequently used for the initialisation of the optimisation problem defined in Equation 1. Each video has already been optimised in its own coordinate system with its own (arbitrary) scale, where the identity mapping was assigned to the respective first frame. Given the spanning tree, we can initialise the common coordinate system by selecting one of the videos as a reference video. Subsequently, transformations of all videos adjacent to the reference video in the spanning tree are pre-multiplied by the pairwise transformations needed to transfer them into the same coordinate system. This achieves that videos with possibly different start and endpoints can be combined which would not be possible otherwise.

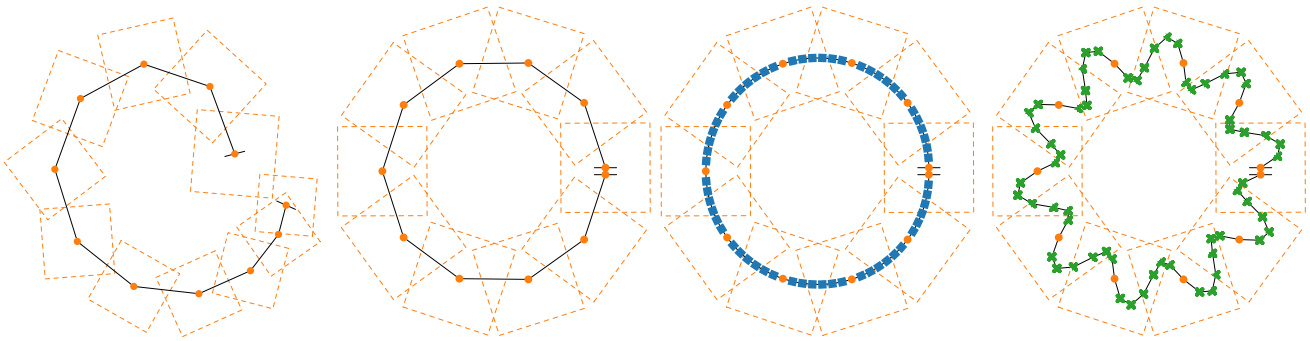
2.4. Reintegration

After performing the optimisation in the previous step, transformations in a shared coordinate system are only given for the sparse key-frame subset of all extracted images. Without the transformations of all intermediate frames, we can only reconstruct an highly sparse camera trajectory.

To reintegrate the omitted frames from the key-frame selection into our problem formulation, we first define an estimate of these transformations in terms of geodesic interpolation. Given the transformations $\tilde{T}_i$ and $\tilde{T}_j$ of the previous and next key-frame, e.g. the adjacent key-frames, we can interpolate the transformations of the intermediate frames by using the geodesic interpolation method:

$$\begin{aligned} F\left(\lambda; \tilde{T}_i, \tilde{T}_j\right) &= \tilde{T}_i \left(\tilde{T}_i^{-1} \cdot \tilde{T}_j\right)^{\lambda} \\ &= \tilde{T}_i \cdot \exp\left(\lambda \log\left(\tilde{T}_i^{-1} \cdot \tilde{T}_j\right)\right), \ \lambda(m) := \frac{m}{l+1}. \end{aligned} \quad (2)$$

The interpolation factor λ is defined by the number of frames l between two consecutive key-frames for all intermediate frames $m \in \{1, \ldots, l\}$. Importantly, the set of 2D homogeneous similarities is a closed subgroup of all invertible 2D matrices $\mathrm{GL}(3)$ and therefore itself a Lie matrix group which makes geodesic interpolation applicable [17, 46]. In contrast to naive linear interpolation, this ensures that all intermediate transformations are (invertible) similarities again since the exponential mapping stays a similarity, which is an important property for the subsequent optimisation step. Moreover, the use of geodesic interpolation allows a natural generalisation of the proposed



(a) Initialisation (b) Optimisation (c) Geodesic interpol. (d) Global refinement

Figure 2: Illustration of the different optimisation stages. Key-frames are outlined in dashed orange lines, where their centres are pictured as orange circles. (a) Initialisation results in a sparse trajectory disturbed by accumulated drift. (b) The first optimisation closes the gap between the start and end point and distributing multiplicative errors across all images. (c) Reintegration is done using geodesic interpolation, where projected centres of intermediate frames are pictured in blue squares. (d) Global refinement recovers motion in-between key-frames outlined in green crosses (i.e. dense camera trajectory).

algorithm to Isometries, Affinities and Homographies since they are again closed subgroups of GL (3). Given these interpolated transformations, we can reconstruct the full trajectory of the camera, but as a direct consequence of the interpolation, movements in-between key-frames are not visible as illustrated in Figure 2c.

To refine these first estimates of the transformations, we again make use of an optimisation formulation very similar to Equation 1, this time keeping the transformations of the selected key-frames fixed and only refining the transformations of the reintegrated frames, which can be done in parallel as each sub-problem is independent of each other:

$$\begin{aligned} &\min J\left(T_m\right) = \\ &\sum_{(f_k, f_l) \in M_{\kappa(i),m}} h\left(f_k, f_l; T_{\kappa(i)}, T_m\right) \\ &\sum_{(f_n, f_o) \in M_{m,\kappa(j)}} h\left(f_n, f_o; T_m, T_{\kappa(j)}\right) \end{aligned} \tag{3}$$

for all $m \in (\kappa(i), \kappa(i+1))$. We perform this step for all intermediate frames with sufficient feature matches with the adjacent key-frames. For all others (i.e. blurry frames or frames with extreme parallax), we use geodesic interpolation with respect to the refined transformations of adjacent frames including non-key-frames after this optimisation step to get a good approximation.

Given these matches, we can guide the optimisation problem to a solution, where the concatenated transformation using the intermediate frame resembles the direct transformation given by our previous optimisation step:

$$\tilde{T}_{i,i+1} = T_{\kappa(i),\kappa(j)} \approx T_{m,\kappa(i+1)} \cdot T_{\kappa(i),m}, \tag{4}$$

for all $m \in (\kappa(i), \kappa(i+1))$ as illustrated in Figure 3.

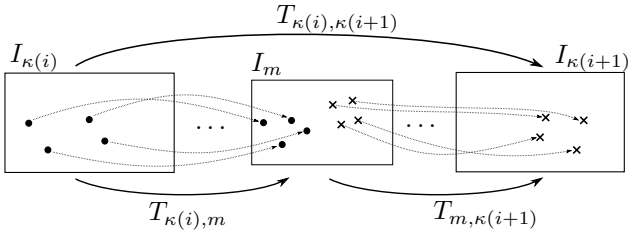
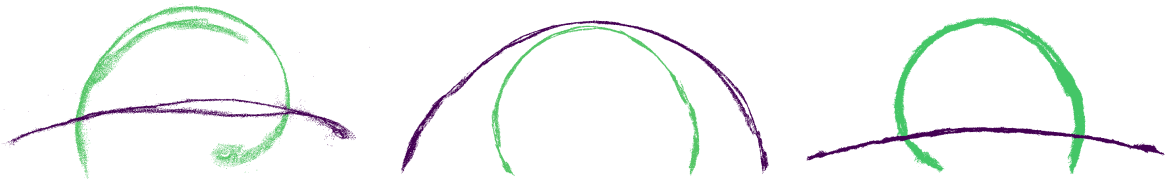


Figure 3: Guided optimisation in the reintegration step.

All in all, this results in a maximally dense approximation of the overall camera motion in a shared coordinate system for every frame and thus allows us to recover the full camera trajectory with high precision (Figure 2d). In contrast to using accumulated pairwise transformations on frames between key-frames similar to the initialisation in Section 2.2, geodesic interpolation in combination with a refinement, leads to better initialisation and, thus, faster convergence.

3. Results

In order to analyse the accuracy and applicability of the proposed method, our algorithm is evaluated in three different experiments. Firstly, to illustrate the limited applicability of 3D-based methods like SLAM or SfM for this use-case, a video containing a loop is processed with ORB-SLAM2 [35], OpenMVG [34], the OpenCV high-level stitching API [2], our method and its extensive counterpart, where no key-frame selection is done and matching is done on all image pairs. For this experiment, a video with 6362 frames and a single point of revisiting is used (Section 3.1). Secondly, a quantitative accuracy analysis is conducted using a Vicon motion capture system in an area of about four by three metres in which markers are at-



(a) ORB-SLAM2 (b) OpenMVG: key-frames (c) OpenMVG: full

Figure 4: Comparison of different 3D pipelines on a single video of a fully planar scene containing a loop with (magenta) and without (green) camera intrinsics from a camera calibration. In the case of OpenMVG, the experiment was conducted once on all frames and once on key-frames only. Calibrated and uncalibrated describe if camera intrinsics from a camera calibration have been supplied.

tached to the camera body. In total, three different camera scenes are generated ranging from 874 to 2946 frames. The ground-truth camera trajectory was then compared with the trajectory of the proposed method. In order to identify the best possible combination of image features and matching, we evaluated ORB, SIFT [31] and SuperPoint [8] features in combination with BeyondSift [18], a k-NN matcher and SuperGlue [42] respectively. In the cases where ORB is not used directly, the results of the matching method described above is only used to get possible candidates which are then matched with the respective other methods. In order to evaluate our reintegration and refinement step, we again compare our results with the extensive counterpart, where all frames have been used in each step (Section 3.2). In a third evaluation, a variety of different hand-held videos are registered in a unified mosaic to demonstrate the applicability of our method. These multi-video scenarios vary from $\sim$ 33,000 images to more than 110,000 frames (Section 3.3).

3.1. Comparison to other pipelines

As shown in Figure 4 extensive scan recordings result in a critical configuration, where missing or inaccurate calibration leads to ambiguities in the determination of radial distortion parameters and curvature of the underlying scene [51]. When an exact calibration is not given or not feasible (i.e. because of zooming of the camera while recording), these methods produce unreliable results which are not usable for 3D registrations. In addition, ORB-SLAM2 uses a key-frame selection and thus only returns a sparse subset of camera transformations, thus, not constructing dense camera trajectories. OpenMVG on the other hand, does not employ key-frame selection and, therefore, takes two days to finish. When passing our selected key-frames to OpenMVG, the computation time reduces to a few minutes but the ambiguities in the reconstruction are more severe. OpenCVs high-level stitching API in "scan" mode, takes 15 minutes on our key-frames and two days on all frames, resulting in invalid image mosaics in both cases. The extensive pipeline crashed with an overflow in the optimisation step in Ceres due to the size of the problem. In contrast, our pipeline takes 10 to 30 minutes depending on the used feature type on all 6362 frames (10/12/30 minutes for ORB/SIFT/SuperPoint). The results are summarised in Table 1. Other methods described in Section 1.1 could not be evaluated here as comparable methods like [11] need external sensor data that was not available in our datasets.

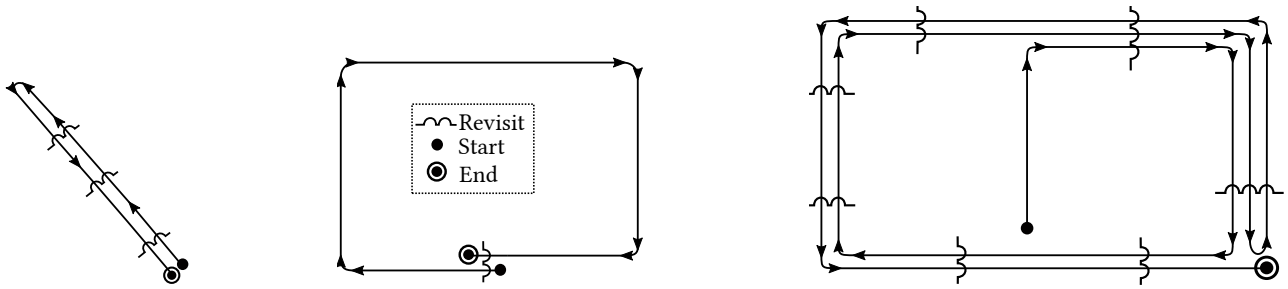
	[35]	[34]	[2]	Ext.	Ours
3D Reconstruction	✓	✓	✗	✗	✗
Avoids Crit. Conf.	✗	✗	✓	✓	✓
Dense Cam. Track	✗	✓	✓	✓	✓
Time [in minutes]	45	2880	3480	n.a.	10

Table 1: Comparison of different pipelines ([35]: ORB-SLAM2; [34]: OpenMVG; [2]: OpenCV; Ext: Extensive) on a video sequence with 6362 frames. The result of the OpenCV Stitcher Pipeline was not usable and in the extensive case the optimisation crashed due to an overflow in Ceres.

3.2. Quantitative Analysis using Vicon Data

In three different experiments with multiple revisiting points as illustrated in Figure 5, the reconstruction was quantified by comparing the iterative closest point distances between the aligned trajectories (positions and angles), after aligning both trajectories in the coordinate system of the Vicon system. The experiments termed *line* and *loop* were repeated three times, whereas the last experiment was repeated twice.

Table 2 shows the resulting spatial distances and angle distances (absolute value of the angle difference) between the ground-truth and the reconstructed trajectories, comparing the distances before and after the final refinement step. The median and median absolute deviation (MAD) [24] are used instead of a conventional mean and variance because start- and endpoint of the measurements by the Vicon system and the recordings are not completely synchronised and



(a) line (b) loop (c) complex

Figure 5: Movement patterns for the different Vicon experiments.

would, thus, distort the measurements through outliers. Evidently, the final refinement steps improve the overall accuracy significantly, where the refined trajectory is close to the extensive versions and in some cases even better. The cases in which the extensive versions are worse can be explained with ill-posed matches: Our key-frame selection supplies frames with good overlap satisfying a similarity constraint. Poor matches (i.e. due to parallax) in adjacent frames introduce noise in the overall optimisation problem that have negative effects on all remaining frames. In our refinement step, we keep the key-frame transformations fixed and optimise intermediate frames independently to mitigate this issue. The best feature type for our pipeline with respect to speed and accuracy is SIFT, due to a poorly optimised k-NN matcher for binary features in OpenCV. For this case, the overall median distance is 7.6 mm with MAD of 4.1 mm in the reintegrated and 5.7 mm with a MAD of 2.7 mm in the refined case, where the angular distance is 3° with a MAD of 4° in the reintegrated and 3° with a MAD of 3° in the refined case. Note, that the pipeline in the complex scene in the case of SuperPoint/Superglue was cancelled after four days.

3.3. Qualitative Results

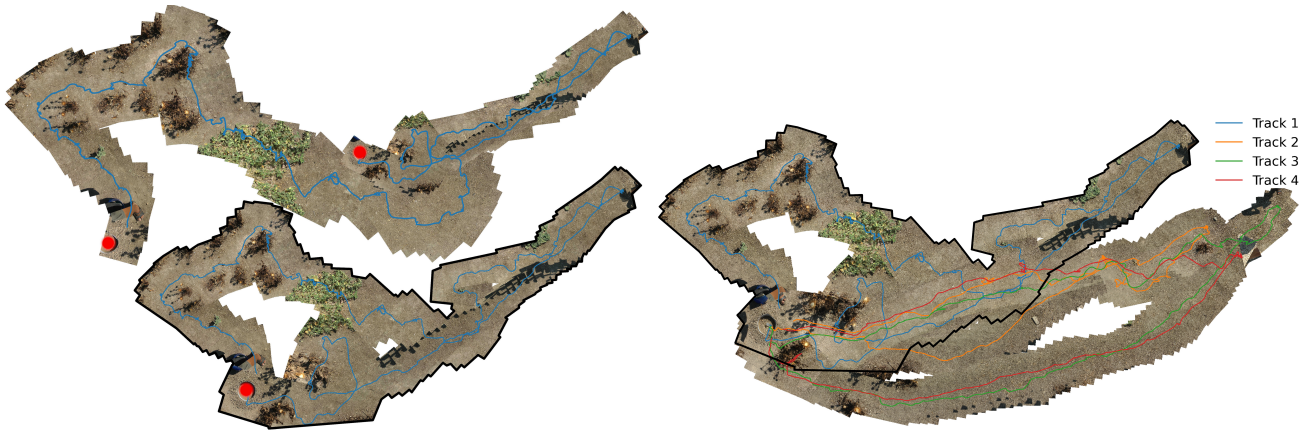
In Figure 6, the results of different steps of the algorithm are illustrated for four videos with a frame rate of 50 frames per second and with $33{,}278$ frames in total. The single video case before any optimisation is outlined in the top part of Figure 6a, where severe drift leads to the two red dots (point of revisit) not being aligned. After the optimisation step, the dots are aligned as pictured in the bottom part of Figure 6a. Figure 6b shows multiple videos from the same environment combined in one mosaic, where different camera trajectories are outlined in different colours. In additional real-world experiments, we pushed the number of processed frames further by registering $114{,}470$ camera positions without any external initialisations, which took 217 minutes on an Intel Xeon E5-2695 CPU, of which 130 minutes were for feature detection, 15 minutes for matching, 15 minutes for key-frame selection and 57 minutes for optimisation. Our evaluations confirm that our algorithm can be used to generate scalable high-resolution image mosaics with embedded full frame rate camera trajectories from multiple monocular and non-calibrated video recordings which are applicable to a wide range of scan applications. The proposed key-frame selection reduces the number of processed frames in the first optimisation problem to 1-3% in the tested videos with a hand-held camera and a frame rate of 50 frames per second.

4. Conclusion

In this paper, we proposed a novel multi-video registration strategy capable of registering thousands of frames into a large-scale image mosaic without suffering from structural errors such as drift. This is achieved by splitting the task into two superimposing processing steps, namely key-frame based global registration and dense geodesic transformation-based reintegration. Graph structures and optimised feature selection schemes are used where possible to enable large-scale registrations of thousands of images into a single global solution. Our two-step processing procedure reduces the computational complexity of the underlying optimisation problem and allows for unprecedented parallelisation. In particular, the dense optimisation of all frames is detached from redundant intermediate frames (contribution less to inter-video revisits and loops) by using graph-based exhaustive matching and loop-closure mechanisms which are optimal across videos in terms of their shared matches. Once a common coordinate system is established based on the key-frames, potential drift or other errors arising from intermediate frames can be alleviated by using geodesic interpolation-based reintegration. This dense reintegration strategy offers several advantages. Most importantly, frames initialised by geodesic interpolation can be optimised by using only a few global iterations. Moreover, we observed that the locations of the previously registered key-frames only change negligibly in the final refinement step, since the reintegrated frames only superimpose

	Line (≈ 1000 Frames)			Loop (≈ 1300 Frames)			Complex (≈ 3000 Frames)		
	ε_d [mm]	ε_a [°]	t [s]	ε_d [mm]	ε_a [°]	t [s]	ε_d [mm]	ε_a [°]	t [s]
ORB Reint.	9.1/5.3	1/2	80	6.0/3.7	9/4	97	10.5/5.8	3/3	272
ORB Ref.	6.7/4.1	0/2	+ 4	5.8/3.5	9/4	+ 3	9.9/5.6	3/3	+ 6
ORB Full	6.7/4.0	0/2	2673	5.5/2.8	9/3	4657	9.7/5.9	3/3	19722
SIFT Reint.	6.5/3.6	1/2	75	6.8/3.8	9/4	90	8.7/4.3	3/2	200
SIFT Ref.	5.7/3.0	0/2	+ 4	5.8/3.5	9/4	+ 4	5.7/2.3	3/2	+ 7
SIFT Full	4.9/2.5	0/2	1568	7.4/4.0	9/4	1691	7.6/4.1	3/2	7252
SP+SG Reint.	7.5/5.2	0/2	86	5.2/2.7	9/4	110	23.4/16.0	3/3	390
SP+SG Ref.	6.9/5.4	0/2	+ 4	5.1/2.6	9/4	+ 4	21.3/15.4	3/3	+ 6
SP+SG Full	5.9/3.9	0/2	80544	5.2/2.8	9/4	128809	*n.a.*	*n.a.*	> 4d

Table 2: Quantitative results regarding the experiments from Figure 5 given as median/MAD for ORB, SIFT and SuperPoint together with SuperGlue before (only reintegrated) and after (refined) the final optimisation step in comparison with the full extensive counterparts.



(a) (b)

Figure 6: Qualitative results of a real-world video. (a) The top figure shows a single video without optimisation. The green bushes as well as the red points which mark the common start and end point of the video are misaligned. The second figure show the same video after optimisation. (b) Multiple videos have been aligned using the multi-video optimisation module.

the global solution derived from the most informative key-frames. When combined with the accuracy measures of our quantitative evaluation, this provides evidence that our two-step registration procedure indeed naturally splits the huge dataset while preserving the relevant characteristics for each processing step. Our results indicate that in some cases our algorithm can even achieve better results than the exhaustive counterpart. Furthermore, our real-world experiment demonstrates that straight-forward scan recordings of uncalibrated consumer grade cameras can be used to compute a high-resolution image mosaic of the underlying scenery while registering more than 110,000 frames in about 3 hours and 30 minutes. Embedding the dense camera trajectory into the image mosaic enables a variety of applications such as per-frame localisations of drones in aerial survey imagery and camera motion compensated object tracking.

In the proposed algorithm, image stitching across videos might introduce visual artefacts in the image overlaps, which can be addressed by advanced blending strategies. Future work should consider a frame selection using additional visual heuristics, for example minimizing shadows or parallax. Moreover, post-processing methods like gain-compensation can be used to further improve the visual quality of the resulting mosaic [3].

Acknowledgements

Lars Haalck gratefully acknowledges the *Heinrich Böll Stiftung* for their support and all authors would like to thank the *Microsoft AI for Earth* initiative. Moreover, we would like to thank Michael Mangan, Barbara Webb and Antoine Wystrach for providing data and their valuable support and suggestions throughout this project.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Matthew Brown and David Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74:59–73, 08 2007.

[4] Mateusz Brzeszcz, Toby P. Breckon, and Ken Wahren. Real-time Mosaicing from Unconstrained Video Imagery for UAV Applications. In *Proceedings of the 26th International Conference on Unmanned Air Vehicle Systems.*, pages 359–374, 2011.

[5] David Capel and Andrew Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 885–891, 1998.

[6] Javier Civera, Andrew Davison, Juan Magallón, and J. Montiel. Drift-free real-time sequential mosaicing. *International Journal of Computer Vision*, 81:128–137, 2009.

[7] David Corrigan, Ken Sooknanan, Jennifer Doyle, and Colm Lordan. A low-complexity mosaicing algorithm for stock assessment of seabed-burrowing species. *IEEE Journal of Oceanic Engineering*, 44:386–400, 2019.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. *arXiv*, 2017.

[9] Armagan Elibol, Nuno Gracias, and Rafael Garcia. Fast topology estimation for image mosaicing using adaptive information thresholding. *Robotics and Autonomous Systems*, 61(2):125 – 136, 2013.

[10] Khuram Faraz, Walter Blondel, Marine Amouroux, and Christian Daul. Towards skin image mosaicing. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications*, pages 1–6, 2016.

[11] J. Ferrer, Armagan Elibol, Olivier Delaunoy, Nuno Gracias, and Rafael Garcia. Large-area photo-mosaics using global alignment and navigation data. In *OCEANS 2007*, pages 1–9, 2007.

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] Zehua Fu and Mohsen Ardabilian Fard. Learning confidence measures by multi-modal convolutional neural networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1321–1330, 2018.

[14] Heng Guo, Shuaicheng Liu, Tong He, Shuyuan Zhu, Bing Zeng, and Moncef Gabbouj. Joint video stitching and stabilization from moving cameras. *IEEE Transactions on Image Processing*, 25:5491–5503, 2016.

[15] Qiangliang Guo, Jin Xiao, and Xiaoguang Hu. New Keypoint Matching Method Using Local Convolutional Features for Power Transmission Line Icing Monitoring. *Sensors*, 18(3):698, 2018.

[16] Lars Haalck, Michael Mangan, Barbara Webb, and Benjamin Risse. Towards image-based animal tracking in natural environments using a freely moving camera. *Journal of Neuroscience Methods*, 330:108455, 2020.

[17] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations, An Elementary Introduction*. Springer, 2015.

[18] Lei Han, Guyue Zhou, Lan Xu, and Lu Fang. Beyond SIFT using Binary features for Loop Closure Detection. *International Conference on Intelligent Robots and Systems*, 2017.

[19] Fangning He, Tian Zhou, Weifeng Xiong, Seyyed Meghdad Hasheminnasab, and Ayman Habib. Automated Aerial Triangulation for UAV-Based Mapping. *Remote Sensing*, 10(12):1952, 2018.

[20] Thomas Kohler, Axel Heinrich, Andreas Maier, Joachim Hornegger, and Ralf P Tornow. Super-resolved retinal image mosaicing. In *2016 IEEE 13th International Symposium on Biomedical Imaging*, pages 1063–1067, 2016.

[21] Kavitha Kuppala, Sandhya Banda, and Thirumala Rao Barige. An overview of deep learning methods for image registration with focus on feature-based approaches. *International Journal of Image and Data Fusion*, 2020.

[22] Maxime Lafarge, Pim Moeskops, Mitko Veta, Josien Pluim, and Koen Eppenhof. Deformable image registration using convolutional neural networks. In *Medical Imaging 2018: Image Processing*, volume 10574, pages 192 – 197, 2018.

[23] Sara Lafkih and Youssef Zaz. Solar panel monitoring using a video frames mosaicing. *2016 International Renewable and Sustainable Energy Conference (IRSEC)*, pages 247–250, 2016.

[24] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764 – 766, 2013.

[25] Jing Li, Wei Xu, Jianguo Zhang, Maojun Zhang, Zhengming Wang, and Xuelong Li. Efficient Video Stitching Based on Fast Structure Deformation. *IEEE Transactions on Cybernetics*, 45(12):2707–2719, 2015.

[26] Yan Li, Carly J. Randall, Robert van Woesik, and Eraldo Ribeiro. Underwater video mosaicing using topology and superpixel-based pairwise stitching. *Expert Systems with Applications*, 119:171–183, 2018.

[27] Jian Liang, Daniel DeMenthon, and David Doermann. Mosaicing of camera-captured document images. *Computer Vision and Image Understanding*, 113(4):572–579, 2009.

[28] Julie Linchant, Jonathan Lisein, Jean Semeki, Philippe Lejeune, and Cédric Vermeulen. Are unmanned aircraft systems (UASs) the future of wildlife monitoring? A review of accomplishments and challenges. *Mammal Review*, 45(4):239 – 252, 2015.

[29] Jonathan Lisein, Julie Linchant, Philippe Lejeune, Philippe Bouché, and Céderic Vermeulen. Aerial Surveys Using an Unmanned Aerial System (UAS): Comparison of Different Methods for Estimating the Surface Area of Sampling Strips. *Tropical Conservation Science*, pages 506–520, 2013.

[30] K E Loewke, D B Camarillo, W Piyawattanametha, M J Mandella, C H Contag, S Thrun, and J K Salisbury. In Vivo

Micro-Image Mosaicing. *IEEE Transactions on Biomedical Engineering*, 58(1):159 – 171, 2010.

[31] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.

[32] Philip F McLauchlan and Allan Jaenicke. Image mosaicing using sequential bundle adjustment. *Image and Vision Computing*, 20(9):751–759, 2002.

[33] Edgardo Molina, Wai Lun Khoo, Hao Tang, Zhigang Zhu, and Arthur Ardeshir Goshtasby. Video Image Registration. In *Theory and Applications of Image Registration*, Theory and Application of Image Registration, pages 357–396. John Wiley & Sons, 2017.

[34] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[35] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[36] Andriy Myronenko and Xubo Song. Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[37] Achala Pandey and Umesh C. Pati. Panoramic Image Mosaicing: An Optimized Graph-Cut Approach. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, volume 43, pages 299–305, 2015.

[38] Dhruven Praiapati and Krutl. J. Danvarwala. Various Document Image Mosaicing Method in Image processing: A Survey. In *2015 International Conference on Signal Processing and Communication Engineering Systems*, pages 281–285, 2015.

[39] K. Sai Venu Prathap, S. A. K. Jilani, and P. Ramana Reddy. A Real-time Image Mosaicing Using Onboard Computer. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems, Advances in Intelligent Systems and Computing*, volume 668, pages 359–367. Springer Singapore, 2018.

[40] Stephan Preibisch, Stephan Saalfeld, and Pavel Tomancak. Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics*, 25(11):1463–1465, 2009.

[41] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[42] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. *arXiv*, 2019.

[43] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4104 – 4113, 2016.

[44] Sharmishtaa Seshamani, Michael D. Smith, Jason J. Corso, Marcus O. Filipovich, Ananth Natarajan, and Gregory D. Hager. Direct global adjustment methods for endoscopic mosaicking. In *Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, volume 7261, pages 447 – 455. SPIE, 2009.

[45] J M Sloan, K A Goatman, and J P Siebert. Learning Rigid Image Registration - Utilizing Convolutional Neural Networks for Medical Image Registration. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 2: BIOIMAGING*, pages 89–99. SciTePress, 2018.

[46] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Scale drift-aware large scale monocular slam. In *In Proceedings of Robotics: Science and Systems*, 2010.

[47] R. Szeliski. Image mosaicing for tele-reality applications. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.

[48] R. Szeliski. Image Alignment and Stitching: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2006.

[49] Esin Turkbeyler and Chris Harris. Building Aerial Mosaics II, 2009.

[50] Zhaobin Wang and Zekun Yang. Review on image-stitching techniques. *Multimedia Systems*, 2020.

[51] Changchang Wu. Critical Configurations for Radial Distortion Self-Calibration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 25 – 32, 2014.

[52] Armand Zampieri, Guillaume Charpiat, and Yuliya Tarabalka. Coarse to fine non-rigid registration: a chain of scale-specific neural networks for multimodal image alignment with application to remote sensing. *CoRR*, 2018.

[53] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip H S Torr. GA-Net: Guided Aggregation Net for End-To-End Stereo Matching. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–194, 2019.

[54] S Zhu, R Zhang, L Zhou, and T Shen. Very Large-Scale Global SfM by Distributed Motion Averaging. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018.

[55] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.