Mesoscopic photogrammetry with an unstabilized phone camera

Kevin C. Zhou,* Colin Cooke, Jaehee Park, Ruobing Qian,
Roarke Horstmeyer, Joseph A. Izatt, Sina Farsiu
Duke University, Durham, NC
kevin.zhou@duke.edu

Abstract

We present a feature-free photogrammetric technique that enables quantitative 3D mesoscopic (mm-scale height variation) imaging with tens-of-micron accuracy from sequences of images acquired by a smartphone at close range (several cm) under freehand motion without additional hardware. Our end-to-end, pixel-intensity-based approach jointly registers and stitches all the images by estimating a coaligned height map, which acts as a pixel-wise radial deformation field that orthorectifies each camera image to allow plane-plus-parallax registration. The height maps themselves are reparameterized as the output of an untrained encoder-decoder convolutional neural network (CNN) with the raw camera images as the input, which effectively removes many reconstruction artifacts. Our method also jointly estimates both the camera's dynamic 6D pose and its distortion using a nonparametric model, the latter of which is especially important in mesoscopic applications when using cameras not designed for imaging at short working distances, such as smartphone cameras. We also propose strategies for reducing computation time and memory, applicable to other multi-frame registration problems. Finally, we demonstrate our method using sequences of multi-megapixel images captured by an unstabilized smartphone on a variety of samples (e.g., painting brushstrokes, circuit board, seeds).

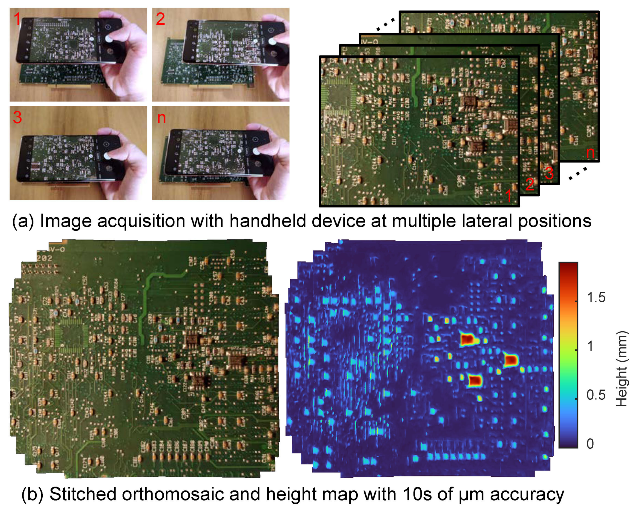

Figure 1. Our method jointly stitches multi-megapixel images acquired under freehand motion at close range and reconstructs high-accuracy height maps without precalibration of camera distortion.

1. Introduction

The photogrammetric problem of reconstructing 3D representations of an object or scene from 2D images taken from multiple viewpoints is common and well studied, featuring prominently in techniques such as multi-view stereo (MVS) [15], structure from motion (SfM) [51, 57, 46], and simultaneous localization and mapping (SLAM) [13]. Implicit in these 3D reconstructions is knowledge of the camera parameters, such as camera position, orientation, and distortions, which are jointly estimated in SfM and SLAM. Photogrammetry tools have been developed and applied to both long-range, macro-scale applications [41], such as building-scale reconstructions or aerial topographical mapping, and close-range, meter-scale applications [33], such as industrial metrology. However, comparatively less work has been done to push photogrammetry to mesoscopic (mm variation) and microscopic scales, where additional issues arise, such as more limited depths of field and increased impact of camera distortion. Existing approaches at smaller scales typically require very careful camera distortion precalibration, expensive cameras, dedicated setups that allow well-controlled camera or sample motion (e.g., with a dedicated rig), or attachment of control points to the object [33].

Here, we show that a smartphone is capable of obtaining quantitative 3D mesoscopic images of objects with 100 µm- to mm-scale height variations at tens-of-micron accuracies with unstabilized, freehand motion and without precalibration of camera distortion (Fig. 1). To achieve this, we present a new photogrammetric reconstruction algorithm that simultaneously stitches the multi-perspective images after warping to a common reference frame, reconstructs sample's 3D height profile, and estimates the camera's position, orientation, and distortion (via a piecewise linear, non-parametric model) in an end-to-end fashion without relying on feature point extraction and matching. Our careful modelling of distortions is especially important for mesoscopic applications. Our method also features a reparameterization of the camera-centric height maps as the outputs of a single *untrained* convolutional neural network (CNN) with the raw camera images at the input (akin to the deep image prior (DIP) [52]), which is optimized instead of the height map itself. Since the camera-centric height maps are by design coaligned with the camera images, they are automatically registered once the camera images are registered. As we will demonstrate, both the use of an untrained CNN and careful modeling of distortion substantially reduce reconstruction artifacts, thus allowing high-accuracy height map estimation without camera precalibration or stabilization.

2. Related work

The majority of general-purpose implementations of SfM, MVS, and photogrammetry, as well as for more general image registration and stitching problems are centered around feature points [57, 14, 46, 37, 44]. A typical multi-step pipeline starts with extraction of feature points, such as scale-invariant features (SIFT [32]), which are then matched across multiple images. The pipeline then crudely estimates the 3D point cloud of the object along with the cameras' positions and poses, which are then refined with bundle adjustment (BA). Our method differs from BA in that it is based on pixel-wise image registration without requiring inference of point correspondences, and operates on rasterized 2D height rather than 3D point clouds, making it more amenable to incorporation of CNNs.

Prior to the development of robust feature point descriptors, pixel-intensity-based techniques for depth estimation from image sequences were also common [36, 24, 35, 23, 20, 48, 40]. Our method falls in this category. In more recent years, there has been a resurgence of interest in direct approaches [39, 11, 9, 3, 10], including those incorporating deep learning. In particular, our work is similar to previous self-supervised deep learning methods that train a CNN on consecutive frames of videos to generate camera-centric depth estimates based on dense pairwise viewpoint warping [65, 53, 54, 31, 17, 42, 60, 34, 61, 18]. However, they differ in that we jointly warp each frame to a common, potentially unseen reference (e.g., a world reference frame) and stitch them together, while these methods consider consecutive pairs of frames and stay within camera reference frames without stitching to form a larger field of view. Although these methods don't require labels, they still require datasets for training, unlike our method. Other deep learning methods have used multiple frames to estimate depth; however, unlike our method, many of these techniques require known camera poses [55, 26, 59, 27, 25, 38] or reference a keyframe [55, 26, 59, 27, 25, 49, 50, 62, 56], and *all* of these multi-frame methods require supervision and ignore camera distortion. More generally, none of the deep learning approaches mentioned in this paragraph estimate or acknowledge camera distortion, with one exception [18], which uses a quartic polynomial radial distortion model. However, as we will show, such a model is too limited for obtaining high-accuracy results in high-resolution mesoscopic imaging applications.

While our method uses a CNN, it does not require training on and therefore does not inherit any biases from a dataset. Rather, the CNN serves as a drop-in, compression-based regularizer without the need for training or generalization beyond the current sample under investigation. As such, our work is also similar to recent work on using DIP [52] to fill in gaps in camera-centric depth images based on warping to nearby view, which requires knowledge of the camera poses [16]. While our CNN-based regularizer was inspired by DIP, it differs in that our CNN takes the camera images as input rather than random noise, thus allowing a single shared CNN and therefore a fixed number of parameters regardless of the number of images in the sequence.

3. Our approach

Our photogrammetric method is an end-to-end, feature-free, multi-image registration technique formulated as an inverse problem. It is reminiscent of a technique previously developed for jointly registering and combining microscopic images from multiple angles for optical superresolution [64], and of other pixel-intensity-based multi-image alignment techniques developed for a variety of tasks, such as digital superresolution [43, 2]. Here, our goal is to register and stitch the multi-view camera images into a single composite mosaic using a plane-plus-parallax framework [29, 45], where the image deformation parameters are the camera parameters and the sample height map. The key insight of our approach is to use these parameters to co-rectify the camera images so that they appear to have been taken from a common perspective, thus allowing joint estimation of the stitched RGB mosaic and coaligned height map.

In the following subsections, we describe in detail the camera model employed to accurately account for distortion and smartphone focal plane shift, the perspective rectification models, the multi-image registration framework, and the CNN-based regularization framework. We also propose strategies for dealing with large, multi-megapixel image that alleviate time and memory costs to produce larger, high-resolution RGBH reconstructions (H = height).

3.1. Image deformation model

Pinhole camera and thin lens model. Assuming an ideal camera, image formation can be described by a pinhole

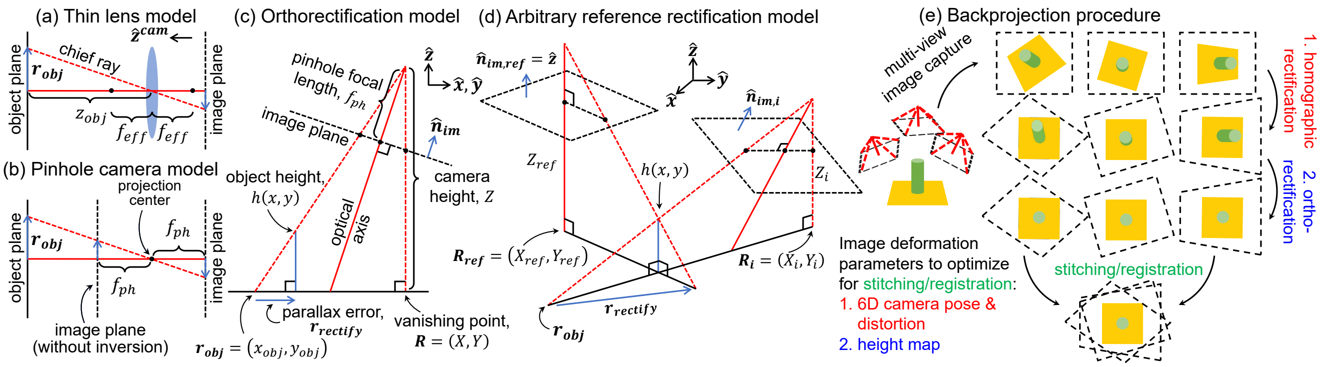

Figure 2. Components of the image deformation model. The chief rays of the thin lens model (a) and the pinhole camera model (b) are equivalent, but have differing definitions of focal length. In general, the scene is not flat (height variation, $h(x, y)$) and the camera is tilted (orientation $\mathbf{\hat{n}_{im}}$). The orthorectification model (c) assumes a reference frame whose projection center is at infinity, such that all projection lines are parallel to $\mathbf{\hat{z}}$. The radial vector field $\mathbf{r_{rectify}}(x, y)$, which converges at the vanishing point ($\mathbf{R}$) and whose magnitudes are $\propto h(x, y)$, performs such a per-pixel rectification, so that the resulting image no longer has perspective distortion. Rectifying the i^{th} image to an arbitrary reference (b) requires a different $\mathbf{r_{rectify}}$. The backprojection procedure of the camera images (e) to facilitate their stitching/registration requires optimization of the 6D camera pose and distortion, and the height map, which performs orthorectification.

camera model, whereby a 3D scene is projected onto a 2D image plane along lines that converge at a point, the center of projection, with a pinhole camera focal length, f_{ph}. The camera itself, however is governed by an effective focal length, f_{eff}, and is related to f_{ph} by the thin lens equation,

$$1/z_{obj} + 1/f_{ph} = 1/f_{eff}, \tag{1}$$

where z_{obj} is the distance of the object being imaged from the pinhole camera focus, which corresponds to the position of the thin lens. Thus, the projection lines of the pinhole camera model correspond to chief rays in the lens model (Fig. 2a,b). When imaging scenes at very far distances ($z_{obj} \gg f_{ph}$), $f_{ph} \approx f_{eff}$, a good assumption for long-range but less so for very close-range applications.

These models do not predict the limited depths of field associated with closer working distances. To avoid this issue, we assume that the user only attempts to manipulate 2 (xy translation) out of the 6 camera pose parameters, which justifies designating an object plane (xy plane) to which the sample height variations may be referenced, assumed to be within the depth of field. In practice, to account for freehand instability, we still model all 6 degrees of freedom for each image: let the camera's 3D orientation be parameterized by a unit normal vector, $\mathbf{\hat{n}_{im}}$, pointing along the optical axis, and an in-image-plane rotation angle, θ; let the camera's 3D position be designated by the position of its center of projection location, $(X, Y, Z = z_{obj})$. In the supplement, we describe the procedure for homographic rectification of the camera images in a common 2D world reference (Fig. 2e).

Camera (un)distortion. Camera lenses and sensor placement are not perfect, giving rise to image distortion. This can pose problems for close-range, mesoscale applications, as the 3D-information-encoding parallax shifts become more similar in magnitude to image deformation due to camera distortion. Distortion models commonly separate radial and tangential components, which are often expanded as even-order polynomials [6, 12]. In some cases, the distortion center should also be optimized [21]. However, as we will show, we found even a 64-order polynomial radial undistortion model insufficiently expressive for our mesoscopic application. Instead, we opted for a nonparametric model, which has been shown to be more flexible in handling general distortions [7, 47]. Specifically, we used a piecewise linear radial undistortion model, whereby the radially dependent relative magnification factor is discretized into n_r points, $\{\widetilde{M}_t\}_{t=0}^{n_r-1}$, spaced by δ_r with intermediate points linearly interpolated:

$$\widetilde{M}(r)=\left(1+\lfloor\frac{r}{\delta_r}\rfloor-\frac{r}{\delta_r}\right)\widetilde{M}_{\lfloor\frac{r}{\delta_r}\rfloor}+\left(\frac{r}{\delta_r}-\lfloor\frac{r}{\delta_r}\rfloor\right)\widetilde{M}_{\lfloor\frac{r}{\delta_r}\rfloor+1}, \tag{2}$$

where $\lfloor\cdot\rfloor$ is the flooring operation, $0 \le r < (n_r - 1)\delta_r$ is the radial distance from the distortion center, which is also optimized, and $\widetilde{M}(r) = 1$ if there's no distortion. Thus, for a given point in the image, $\mathbf{r_{im}}$, the distortion correction operation is given by

$$\mathbf{r_{im}} \leftarrow \widetilde{M}(|\mathbf{r_{im}}|)\mathbf{r_{im}}, \tag{3}$$

which is applied before backprojection. A piecewise linear model, unlike high-order polynomials, also has the advantage of being trivially analytically invertible, allowing easy computation of both image distortion and undistortion. This is important because, while BA typically uses a distortion model, our method requires an *un*distortion model, as we first backproject camera images to form the reconstruction.

Orthorectification. To extend our image deformation model to allow registration of scenes with height variation, we need to warp each backprojected image to a common reference in a pixel-wise fashion. One such option is orthorectification (Fig. 2c), which can be interpreted as rectifying to a world reference. As such, the effective camera origin is at infinity, so that our images are governed by true length scales, regardless of proximity to the camera (i.e., no perspective distortion). For each camera image backprojection, we estimate a radial deformation field,

$$\mathbf{r_{rectify}}(\mathbf{r_{obj}}) = \Delta r(\mathbf{r_{obj}} - \mathbf{R})|\mathbf{r_{obj}} - \mathbf{R}|^{-1}, \quad (4)$$

which is a function of position in the object plane, $\mathbf{r_{obj}} = (x_{obj}, y_{obj})$, and moves each pixel a signed distance of Δr towards the vanishing point, $\mathbf{R} = (X, Y)$, the point to which all lines normal to the object plane appear to converge in the homographically rectified camera image. Δr is directly proportional to the height at the new rectified location,

$$h(\mathbf{r_{obj}} + \mathbf{r_{rectify}}) = -Z\Delta r|\mathbf{r_{obj}} - \mathbf{R}|^{-1}. \quad (5)$$

Each image has its own height map, forming an augmented RGBH image, which is pixel-wise orthorectified by Eq. 5.

Orthorectification is a limiting case of the more general, arbitrary reference rectification (Fig. 2d); for details, see the supplement.

Accounting for focal plane shift. Most smartphones have an autofocus feature where the sensor or lens position automatically adjusts to sharpen some part of the image. This feature is only relevant for close-range applications, as can be seen in Eq.1, and manifests as a dynamically adjusted $f_{ph,i}$, while for long-range applications, $f_{ph,i}$ remains fixed at f_{eff}. This issue is intertwined with the well-known limitation of photogrammetry that inferring absolute scale requires something of known length in the scene. In particular, Eq. 5 alone is insufficient to obtain quantitative height maps, because Z_i and $f_{ph,i}$ are ambiguous up to a scale factor related to the camera's magnification for the i^{th} image:

$$M_i = f_{ph,i}/Z_i. \quad (6)$$

Combining Eqs. 1 and 6 and solving for Z_i (=$z_{obj,i}$), we get

$$Z_i = f_{eff}(1 + M_i^{-1}), \quad (7)$$

which, when combined with Eq. 5, yields the ambiguity-free, absolute height estimate by the i^{th} image,

$$h_i(\mathbf{r_{obj}}+\mathbf{r_{rectify}}) = -f_{eff}\Delta r_i|\mathbf{r_{obj}} - \mathbf{R_i}|^{-1}\left(1 + M_i^{-1}\right). \quad (8)$$

Note that M_i contains global scale information and thus must be precalibrated for at least one i. While in principle one can optimize $f_{ph,i}$ or M_i for the remaining values of i, we used an approximation, $M_i \approx M_0Z_0/Z_i$, which assumes the camera autofocused once and maintained a similar f_{ph} throughout acquisition. This approximation yields

$$h_i(\mathbf{r_{obj}} + \mathbf{r_{rectify}}) \approx -f_{eff}\frac{\Delta r_i}{|\mathbf{r_{obj}} - \mathbf{R_i}|}\left(1 + \frac{1}{M_0}\frac{Z_i}{Z_0}\right). \quad (9)$$

Ignoring the lens equation (Eq. 1) and thus the focal plane shift results in biased height estimation (see supplement).

Putting it all together. Homographic rectification using the pinhole/thin-lens models, camera undistortion, and orthorectification (or arbitrary-reference rectification) via the height map, as described in this section (3.1), together constitute the backprojection step (Fig. 2e, 3). Let these image deformation parameters be collectively denoted as $\mathbf{w}$.

3.2. Multi-frame image stitching and registration

Given the current estimate of the image deformation parameters, $\mathbf{w}$, we simultaneously backproject all the images to form an estimate of the RGBH reconstruction, $\mathbf{B}$, with the coaligned height map stacked as the fourth channel:

$$\mathbf{B} \leftarrow \mathbf{0}, \qquad \mathbf{B}[\mathbf{x_w}, \mathbf{y_w}] \leftarrow \mathbf{D_{RGBH}}, \quad (10)$$

where $(\mathbf{x_w}, \mathbf{y_w})$ are the flattened coordinates corresponding to the pixels of $\mathbf{D_{RGBH}}$, which are the flattened RGB images augmented with the camera-centric height maps. If a pixel of $\mathbf{B}$ is visited multiple times, the values are averaged.

To guide the optimization, we next generate forward predictions of the camera images, $\hat{\mathbf{D}}_{\mathbf{RGBH}}$, by using the exact same backprojection coordinates, $(\mathbf{x_w}, \mathbf{y_w})$, to *reproject* back into the camera frames of reference and compute the mean square error (MSE) with the original camera images (Fig. 3). The idea is that if the backprojected images are consistent with each other at the pixels where they overlap, then the forward predictions will be more accurate. We then use gradient descent to minimize the MSE with respect to the image deformation parameters, that is

$$\min_{\mathbf{w}} ||\hat{\mathbf{D}}_{\mathbf{RGBH}} - \mathbf{D_{RGBH}}||^2. \quad (11)$$

To avoid local minima, we adopted a multi-scale strategy, whereby both $\mathbf{D_{RGBH}}$ and $\mathbf{B}$ were subject to a downsampling procedure that was relaxed over time. Further, we didn't update the height map until we reached the lowest downsampling factor. If the scene consisted of non-repetitive structures and the camera images exhibited a lot of overlap, initializing each image to the same position was often a good initial guess. However, if this failed, we initialized using sequential cross-correlation-based estimates.

3.3. Regularization

Reparameterization with a CNN. We reparameterized the camera-centric height maps as the output of a CNN with the respective RGB images as the inputs. Instead of optimizing for the per-image height maps, we optimize the weights of a single untrained CNN as a DIP, whose structure alone exhibits a bias towards "natural" images, as empirically demonstrated on multiple image-reconstruction-related tasks [52], including 3D reconstruction [63]. While the DIP is often an overparameterization, in our case the

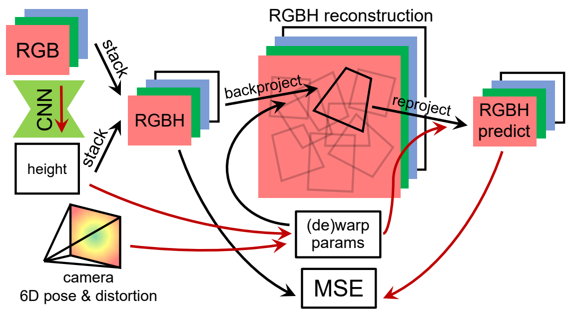

Figure 3. Method overview. A stack of RGB images (only one shown for clarity) is augmented with their camera-centric height maps (RBGH), which are reparameterized as outputs of an untrained CNN. The camera-centric height maps and camera parameters serve as the image deformation parameters, which rectify the RGBH images so that they can be registered and stitched (RGBH reconstruction). We use the same warping coordinates to reproject the images to form a prediction, whose MSE with RGBH is minimized with respect to the CNN and camera parameters. Arrows indicate forward differentiable operations, but only red arrows indicate paths that allow backpropagation (Sec. 3.4).

single CNN has fewer parameters than the total number of height map pixels that we otherwise would have directly optimized, thus offering further means of regularization [22]. Furthermore, we used an encoder-decoder network architecture without skip connections, forcing the information to flow through a bottleneck. Thus, the degree of compression in the CNN is an interpretable regularization hyperparameter, where restricting information flow may force the network to discard artifacts (see supplement for architecture). Finally, we note that the network doesn't need to generalize beyond the current image sequence.

Camera-centric height map consistency. Although we directly optimize for camera-centric height maps, they ultimately follow the same backprojection and reprojection procedure that the RGB images undergo, as described earlier. However, while the contribution of RGB pixels to the MSE in Eq. 11 serves as feedback for registration, the contribution of the height values to the MSE is primarily to make the camera-centric height maps more consistent irrespective of the backprojection result. This can be useful, for example, when filling in height values at the vanishing points, which are blind spots, as $h \propto \mathbf{r}_{\mathbf{rectify}}(\mathbf{R}) = 0$. Since RGB values and height values are not directly comparable, we introduce a regularization hyperparameter that scales their relative contributions.

3.4. Reducing computation time and memory

To make our method tractable on a GPU, we used gradient checkpointing [8, 19] and CPU memory swapping [30], as well as two novel strategies, which we discuss next.

Blocking backpropagation through the reconstruction Instead of computing the total gradient of the loss with respect to the image deformation parameters, which would require backpropagation across every path that leads to the deformation parameters, we compute partial gradients using only the paths that lead to the deformation parameters without going through the reconstruction (red arrows in Fig. 3). Writing out the relevant terms in the chain rule expansion of the gradient of the loss with respect to the image deformation parameters (Sec. 3.1), we have

$$\frac{dL}{d\mathbf{w}} = \frac{\partial L}{\partial \hat{\mathbf{D}}_{\mathbf{RGBH}}} \left(\mathbf{J}_{\hat{\mathbf{D}}_{\mathbf{RGBH}}}(\mathbf{w}) + \cancelto{0}{\mathbf{J}_{\hat{\mathbf{D}}_{\mathbf{RGBH}}}(\mathbf{B}) \mathbf{J}_{\mathbf{B}}(\mathbf{w})} \right), \tag{12}$$

where $\mathbf{J}_{\mathbf{y}}(\mathbf{x})$ denotes the Jacobian of $\mathbf{y}$ with respect to $\mathbf{x}$ and L is the loss. Doing so ends up saving memory and time because it avoids computing expensive derivatives associated with the reconstruction backprojection and reprojection steps. An intuitive interpretation is that at each iteration the reconstruction serves as a temporarily static reference to which all the images are being registered, as opposed to also explicitly registering the reconstruction to the images. Note, however, that both registration directions are governed by the same parameters, and that this "static" reference still updates at every iteration.

Batching with a running-average reconstruction. At every iteration of the optimization, we require an estimate of the reconstruction, which itself requires joint participation of all images in the dataset to maximize the available information. This can be problematic as it requires both the reconstruction and the entire dataset to be in GPU memory at the same time. Batching is a standard solution, which at first glance would only work for the reprojection step, as the projection step requires all images to form the reconstruction. A two-step approach to overcome this requirement is to realize that, because of our strategy of blocking backpropagation paths through the reconstruction, we can simply generate the reconstruction incrementally in batches without worrying about accumulating gradients. Once the temporarily static reconstruction is generated given the current estimates of the image deformation parameters, we could then update the parameters by registering batches of images to the reconstruction. This approach, however, is inefficient because it requires two passes through the dataset per epoch, and the parameters are only updated during one of the passes.

We introduce a more efficient, one-step, end-to-end strategy where each batch updates *both* the reconstruction *and* the parameters by keeping track of a running average of the reconstruction. In particular, the update rule for the reconstruction after the $(j+1)^{th}$ gradient step when presented with the j^{th} batch as a list of warped coordinates and their associated RGB values, $(\mathbf{x}_{w,j}, \mathbf{y}_{w,j}, \mathbf{D}_j)$, is given by

$$\begin{gathered} \mathbf{B}_{j+1} \leftarrow \mathbf{B}_j \\ \mathbf{B}_{j+1}[\mathbf{x}_{w,j}, \mathbf{y}_{w,j}] \leftarrow m\mathbf{B}_j[\mathbf{x}_{w,j}, \mathbf{y}_{w,j}] + (1-m)\mathbf{D}_j \end{gathered} \tag{13}$$

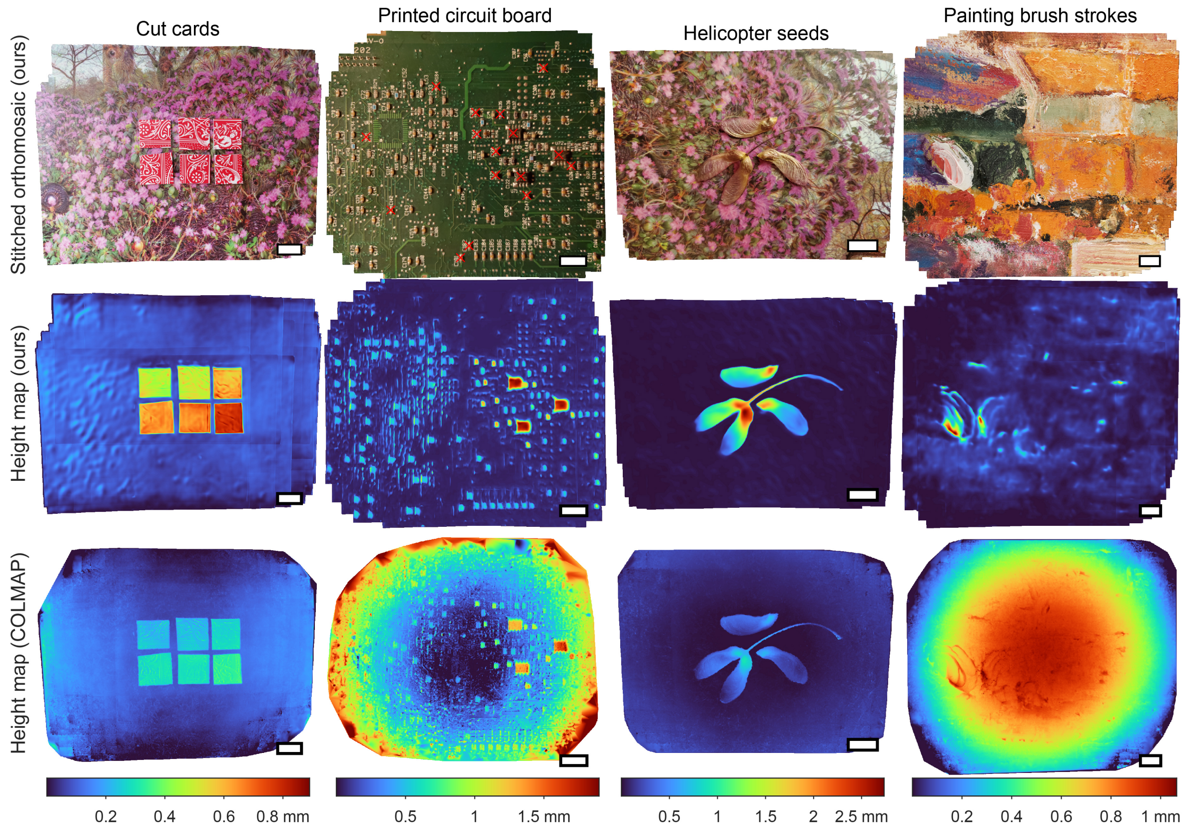

Figure 4. Stitched orthomosaics and height maps for various samples with mm-scale height variation. Height values for the cut cards and PCB components (red $\times$'s) are quantified in Tables 1 and 2, respectively. COLMAP consistently underestimates heights. Scale bars, 1 cm.

where $0 < m < 1$ is the momentum controlling how rapidly to update **B**. The batch is specified very generally in Eq. 13, and can correspond to any subset of pixels from the dataset, whether grouped by image or chosen from random spatial coordinates. Only the spatial positions of the reconstruction visited by the batch are updated in the backprojection step, and the loss is computed with the same batch after the reprojection step. As a result, we only need one pass though the dataset per epoch. This method is general and can be applied to other multi-image registration problems.

4. Experiments

Using the rear wide-angle camera of a Samsung Galaxy S10+ (f_{eff} = 4.3 mm) and freehand motion, we collected multiple image sequence datasets consisting of 21-23 RGB 1512×2016 images ($2\times$-downsampled from 3024×4032). While our method does not require it, we attempted to keep the phone approximately parallel and at a constant height (5-10 cm) from the sample while translating the phone laterally, to keep as much of the sample as possible within the limited depth of field associated with such close working distances. To obtain absolute scale, we estimated the magnification of the first image of each sequence using reference points of known separation in the background.

We implemented our algorithm in TensorFlow [1] (code and data at `https://github.com/kevinczhou/mesoscopic-photogrammetry`), which we ran on an Intel Xeon Silver 4116 processor augmented with an 11-GB GPU (Nvidia RTX 2080 Ti). We used the same CNN architecture for all experiments, tuned on an independent sample to balance resolution and artifact reduction (see supplement for architectures). Gradient descent via Adam [28] was performed for 10,000 iterations (see supplement for discussion on number of iterations) for each sample with a batch size of 6. We set $n_r = 30$ (Eq. 2) and $m = 0.5$ (Eq. 13).

We compare our method to the open-source, feature-based SfM tool, COLMAP [46], which has been shown to outperform competing general-purpose SfM tools [5]. We use COLMAP's full SfM pipeline, with shared camera models and focal lengths, and converted the dense point cloud reconstructions to height maps for comparisons. See supplement for detailed hyperparameter settings.

Accuracy and precision characterization. We first created a calibrated phantom sample consisting of standard playing cards (~0.3-mm thick), cut into six 1-2-cm^2 squares and attached 0-5 layers of tape (50-70 µm thick per layer) to alter their heights. We measured the thicknesses of the tape-

Card # (G. T.)	Ours Acc.	Ours Prec.	CM Acc.	CM Prec.	CM (scaled) Acc.	CM (scaled) Prec.
bkgd (0)	59.4	45.5	235.6	59.6	19.2	157.8
#1 (295)	34.9	37.7	74.9	26.8	41.8	71.0
#2 (350)	2.0	36.5	24.0	24.6	2.4	65.2
#3 (420)	38.5	54.9	6.7	18.4	31.7	48.9
#4 (485)	6.0	26.2	55.6	20.6	9.4	54.7
#5 (555)	32.5	35.5	120.6	15.4	47.3	40.8
#6 (625)	10.5	28.4	151.6	18.1	14.1	47.9
mean	26.3	37.8	95.6	26.2	23.7	69.5

Table 1. Accuracy (abs. error from ground truth (G. T.)) and precision (st. dev.) of our method vs. COLMAP (CM) vs. CM rescaled to match G. T. of the cut card sample (Fig. 4). All units are µm.

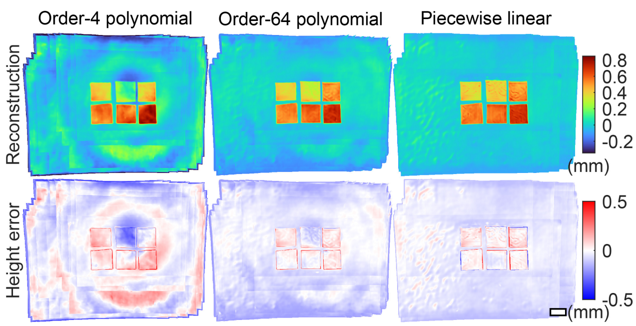

Figure 5. Visual comparison of height map reconstructions of the cut cards sample using different undistortion models. Note the ring artifacts with the polynomial models. Scale bar, 1 cm.

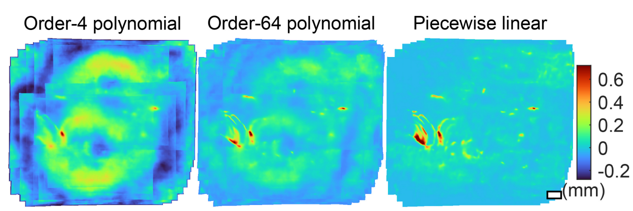

Figure 6. Strong ring artifacts in the painting sample due to uncompensated distortion in polynomial models. Scale bar, 1 cm.

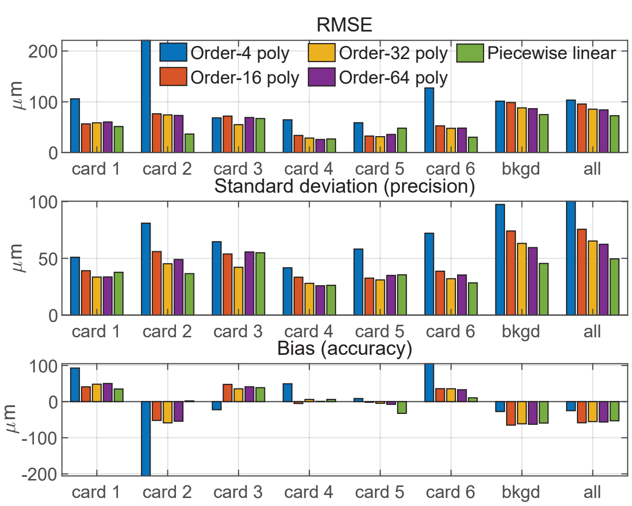

Figure 7. Height estimation performance on the 6 cut cards, background, and whole image for various undistortion models.

backed cut cards using calipers with 20-µm accuracy (Starrett EC799A-12/300), and arranged them on a flat, feature-rich surface [4]. We regarded these measurements as the ground truths, $\mathbf{h_{gt}}$ (column 1 of Table 1).

The stitched orthomosaics and height map reconstructions by our method and COLMAP are shown in the first column of Fig. 4, noting that COLMAP underestimates the card heights. To quantify accuracy and precision of both methods, we manually segmented the six cut cards and background based on the RGB orthomosaic, and computed the mean and standard deviation of the height values of these seven regions (Table 1). Because the height maps have an arbitrary global shift, we used the shift that minimizes the MSE between the mean height estimates and the ground truths: $\Delta h = mean(\mathbf{h_{gt}} - \mathbf{h_{est}})$. While COLMAP underestimates heights and therefore has low absolute accuracy, we hypothesized that the relative accuracy might be high. To test this, we additionally scaled COLMAP's height estimates by the factor that minimizes MSE between mean height estimates and ground truths, given by $cov(\mathbf{h_{gt}}, \mathbf{h_{est}})/var(\mathbf{h_{est}}) \approx 2.65$ (Table 1). Only our method has simultaneously high accuracy (26.3 µm) and precision (37.8 µm) and without the need to rescale.

Results on multiple mm-scale samples. Next, we compared our method and COLMAP on a printed circuit board (PCB), helicopter seeds, and the brush strokes on a painting (Fig. 4). COLMAP not only consistently underestimates heights, but also exhibits unpredictable surface curvatures, possibly due to our imaging procedure operating near a degenerate configuration for radial distortion estimation [58]. Our method, however, is robust to this degeneracy. We quantified the height accuracy and precision on 16 of the PCB components using caliper estimates as the ground truth and manual segmentation based on the RGB orthomosaics (Table 2). We caution that this analysis is less rigorous than that of the cut cards sample, as some components are not completely flat (slight elevations along the shorter edges), and due to greater difficulty in estimating height with the calipers (we alleviated this problem statistically by using the mean of 10 measurements as the ground truth). These caveats may explain why the accuracy (62.5 µm) and precision estimates (95.8 µm) are inflated compared to those of the cut cards sample. However, it is clear that our method does not have COLMAP's underestimation issue. Further evidence of this is that the optimal rescale value for our method is $1.028 \approx 1$ (which we did not use). While we do not have ground truths for the helicopter or painting samples, the calipers estimated the helicopter seeds to be $\sim$2.3 mm at their thickest part, qualitatively consistent with our method; while this is not a rigorous measurement due to non-rigidity of the sample, it at least suggests that COLMAP is underestimating heights.

Importance of undistortion. Figs. 5 and 6 show spuri-

		bkgd	C120	C121	C126	C142	C147	C181	C182	C203	C57	R135	R144	R175	R184	U70	U71	U72	mean
	G.T.	0	1253	1257	1269	618	1282	632	632	1354	677	533	548	476	427	1772	1771	1778	–
Ours	Acc.	119.8	41.4	15.8	79.0	60.1	163.6	5.8	56.0	43.5	94.3	122.4	89.3	21.0	52.7	44.3	27.5	31.9	62.9
Ours	Prec.	–	138.4	123.5	133.2	32.2	175.1	45.3	34.2	86.7	139.2	117.0	61.0	17.6	52.9	101.1	141.6	134.3	95.8
CM	Acc.	455.7	435.4	557.7	193.9	116.6	90.7	401.6	225.0	456.9	548.0	74.6	173.3	452.5	305.4	502.1	388.9	168.3	326.3
CM	Prec.	–	108.0	140.2	83.4	248.1	163.4	118.5	95.5	133.8	51.7	98.7	91.3	105.0	70.4	90.9	95.3	123.6	113.6

Table 2. Quantification of accuracy and precision (in µms) of our method and COLMAP (CM) on PCB components (Fig. 4, red ×'s).

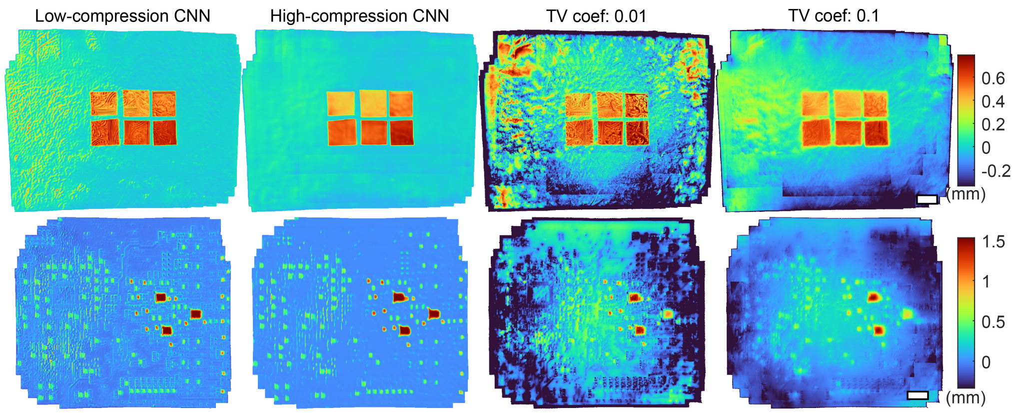

Figure 8. Height map reconstructions of the cut cards and PCB samples using high- and low-compression CNNs (architectures 1 and 4; see supplement for architectures and additional results) and high and low levels of total variation (TV) regularization. Zoom in to see fine features. Scale bars, 1 cm.

ous rings when camera distortions are not sufficiently modeled (cf., [47]). In particular, although for conventional macro-scale applications it is often sufficient to use a low-order even polynomials (e.g., order-4 [18]), for mesoscopic applications, even using an order-64 polynomial undistortion model leaves behind noticeable ring artifacts, while our piecewise linear model (30 segments) effectively eliminates them. Fig. 7 quantifies the performance of multiple undistortion models via root-MSE (RMSE), precision, and accuracy of heights of the cut cards sample. Not only does our piecewise linear model generally have better precision and overall RMSE than the polynomial models, but also it does not exhibit biases caused by the ring artifacts (e.g., card 2), which cannot be corrected by a global scale or shift. See the supplement for a full comparison of undistortion models on all four samples in Fig. 4, as well as the estimated radial undistortion profiles and distortion centers.

Effectiveness of CNN regularizer. CNN reparameterization is crucial to our method. As the CNN is an encoder-decoder network without skip connections, the degree of compression can be adjusted by the number of parameters and downsampling layers. Fig. 8 shows that varying the degree of compression controls the amount of fine detail transferred to the height map without affecting the flatness of the field of view or blurring edges. Thus, the degree of compression in the CNN is an interpretable means of tuning the regularization. However, if we optimize the camera-centric height maps directly with total variation (TV) regularization, we see many artifacts, even when the regularization is strong enough to blur sharp edges. See the supplement for results with more CNN architectures and TV regularization levels for all four samples in Fig. 4, plus the hyperparameter-tuning sample. Although CNNs are generally more computationally complex than TV, CNNs don't make as strong assumptions and therefore can express more complex and general solutions. In particular, while TV is most effective for samples that have piecewise smooth height variation, CNNs can in principle be used for samples with a more diverse set of features.

5. Conclusion

We have presented a feature-free, end-to-end photogrammetric algorithm applied to mesoscopic samples with tens-of-µm accuracy over cms fields of view. Our method features a novel use of CNNs/DIPs, which effectively removes many artifacts from the height maps. We also showed that careful modeling of distortion is important for obtaining accurate height values. Qualitative and quantitative comparisons show that our method outperforms COLMAP, a feature-based SfM tool. Our work fills the gap between 3D computer vision and microscopy, pushing resolution limits of models used in 3D computer vision with consumer-grade cameras. We expect our method to find application in industrial metrology, studying historical artwork, and biomedicine.

Funding. National Science Foundation CBET-1902904.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. 6

[2] Cecilia Aguerrebere, Mauricio Delbracio, Alberto Bartesaghi, and Guillermo Sapiro. A practical guide to multi-image alignment. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1927–1931. IEEE, 2018. 2

[3] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Asian Conference on Computer Vision*, pages 324–341. Springer, 2016. 2

[4] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Massachusetts, USA:, 2017. 7

[5] Simone Bianco, Gianluigi Ciocca, and Davide Marelli. Evaluating the performance of structure from motion pipelines. *Journal of Imaging*, 4(8):98, 2018. 6

[6] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966. 3

[7] Federico Camposeco, Torsten Sattler, and Marc Pollefeys. Non-parametric structure-based calibration of radially symmetric cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2200, 2015. 3

[8] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 5

[9] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014. 2

[10] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2

[11] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 2

[12] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001. 3

[13] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 1

[14] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18. Citeseer, 2014. 2

[15] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1

[16] Pallabi Ghosh, Vibhav Vineet, Larry S Davis, Abhinav Shrivastava, Sudipta Sinha, and Neel Joshi. Deep depth prior for multi-view stereo. *arXiv preprint arXiv:2001.07791*, 2020. 2

[17] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019. 2

[18] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8977–8986, 2019. 2, 8

[19] Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, 26(1):19–45, 2000. 5

[20] KJ Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *Proceedings of the IEEE workshop on visual motion*, pages 156–157. IEEE Computer Society, 1991. 2

[21] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1309–1321, 2007. 3

[22] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv preprint arXiv:1810.03982*, 2018. 5

[23] Joachim Heel. Direct estimation of structure and motion from multiple frames. Technical report, Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 1990. 2

[24] Berthold KP Horn and EJ Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, 1988. 2

[25] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2651–2660, 2019. 2

[26] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[27] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 2

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[29] Rakesh Kumar, Prabu Anandan, and Keith Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 685–688. IEEE, 1994. 2

[30] Tung D Le, Haruki Imai, Yasushi Negishi, and Kiyokuni Kawachiya. Tflms: Large model support in tensorflow by graph rewriting. *arXiv preprint arXiv:1807.02037*, 2018. 5

[31] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018. 2

[32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[33] Thomas Luhmann. Close range photogrammetry for industrial applications. *ISPRS journal of photogrammetry and remote sensing*, 65(6):558–569, 2010. 1

[34] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 2

[35] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989. 2

[36] Larry H Matthies, Richard Szeliski, and Takeo Kanade. Incremental estimation of dense depth maps from image sequences. In *CVPR*, pages 366–374, 1988. 2

[37] 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. 2

[38] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. *arXiv preprint arXiv:2003.10432*, 2020. 2

[39] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 2

[40] John Oliensis. Direct multi-frame structure from motion for hand-held cameras. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 889–895. IEEE, 2000. 2

[41] GN Peggs, Paul G Maropoulos, EB Hughes, AB Forbes, S Robson, M Ziebart, and Balasubramanian Muralikrishnan. Recent developments in large-scale dimensional metrology. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(6):571–595, 2009. 1

[42] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 2

[43] Dirk Robinson, Sina Farsiu, and Peyman Milanfar. Optimal registration of aliased images using variable projection with applications to super-resolution. *The Computer Journal*, 52(1):31–42, 2009. 2

[44] Ewelina Rupnik, Mehdi Daakir, and Marc Pierrot Deseilligny. Micmac–a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*, 2(1):1–9, 2017. 2

[45] Harpreet S Sawhney. 3d geometry from planar parallax. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 929–934, Los Alamitos, CA, USA, Jun 1994. IEEE Computer Society. 2

[46] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 2, 6

[47] Thomas Schops, Viktor Larsson, Marc Pollefeys, and Torsten Sattler. Why having 10,000 parameters in your camera model is better than twelve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2544, 2020. 3, 8

[48] Gideon P Stein and Amnon Shashua. Model-based brightness constraints: On direct estimation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):992–1015, 2000. 2

[49] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[50] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 2

[51] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 1

[52] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 2, 4

[53] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 2

[54] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018. 2

[55] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257. IEEE, 2018. 2

[56] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. *arXiv preprint arXiv:1912.09697*, 2019. 2

[57] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 1, 2

[58] C. Wu. Critical configurations for radial distortion self-calibration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 25–32, 2014. 7

[59] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 2

[60] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[61] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 2

[62] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018. 2

[63] Kevin C Zhou and Roarke Horstmeyer. Diffraction tomography with a deep image prior. *Optics Express*, 28(9):12872–12896, 2020. 4

[64] Kevin C Zhou, Ruobing Qian, Simone Degan, Sina Farsiu, and Joseph A Izatt. Optical coherence refraction tomography. *Nature Photonics*, 13(11):794–802, 2019. 2

[65] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 2