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).

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 modeling 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 RGB 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 images 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
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,
\[
\frac{1}{z_{obj}} + \frac{1}{f_{ph}} = \frac{1}{f_{eff}},
\]
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 two 3D translations 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, \( \hat{b}_{im} \), pointing along the optical axis, and an in-image-plane rotation angle, \( \theta \); 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, \( \{M_i\}_{i=0}^{n_r-1} \), spaced by \( \delta_r \), with intermediate points linearly interpolated:
\[
\hat{M}(r) = \left( 1 + \left\lfloor \frac{r}{\delta_r} \right\rfloor - \frac{r}{\delta_r} \right) M(\lfloor \frac{r}{\delta_r} \rfloor) + \left( \frac{r}{\delta_r} - \left\lfloor \frac{r}{\delta_r} \right\rfloor \right) \hat{M}(\lfloor \frac{r}{\delta_r} \rfloor + 1),
\]
where \( \lfloor \cdot \rfloor \) is the flooring operation, \( 0 \leq r < (n_r - 1)\delta_r \) is the radial distance from the distortion center, which is also optimized, and \( \hat{M}(r) = 1 \) if there’s no distortion. Thus, for a given point in the image, \( r_{im} \), the distortion correction operation is given by
\[
r_{im} \leftarrow \hat{M}(\lfloor r_{im} / \delta_r \rfloor) r_{im},
\]
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 undistortion 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,

\[ r_{\text{rectify}}(r_{\text{obj}}) = \Delta r(r_{\text{obj}} - R)|r_{\text{obj}} - R|^{-1}, \]  

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

\[ h(r_{\text{obj}} + r_{\text{rectify}}) = -Z\Delta r|r_{\text{obj}} - R|^{-1}. \]

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 \( f_{\text{eff}} \). 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:

\[ M_i = f_{ph,i}/Z_i. \]

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

\[ Z_i = f_{\text{eff}}(1 + M_i^{-1}), \]

which, when combined with Eq. 5, yields the ambiguity-free, absolute height estimate by the \( i \)th image,

\[ h_i(r_{\text{obj}} + r_{\text{rectify}}) = -f_{\text{eff}} \Delta r_i|r_{\text{obj}} - R_i|^{-1} \left( 1 + M_i^{-1} \right). \]

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

\[ h_i(r_{\text{obj}} + r_{\text{rectify}}) \approx -f_{\text{eff}} \frac{\Delta r_i}{|r_{\text{obj}} - R_i|} \left( 1 + \frac{1}{M_0} \frac{Z_i}{Z_0} \right). \]

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 \( w \).

### 3.2. Multi-frame image stitching and registration

Given the current estimate of the image deformation parameters, \( w \), 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:

\[ \mathbf{B} \leftarrow 0, \quad \mathbf{B}[x_w, y_w] \leftarrow \mathbf{D}_{\text{RGBH}}. \]

where \( (x_w, y_w) \) are the flattened coordinates corresponding to the pixels of \( \mathbf{D}_{\text{RGBH}} \), 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, \( \mathbf{D}_{\text{RGBH}} \), by using the exact same backprojection coordinates, \( (x_w, y_w) \), 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

\[ \min_w \| \mathbf{D}_{\text{RGBH}} - \mathbf{D}_{\text{RGBH}} \|^2. \]

To avoid local minima, we adopted a multi-scale strategy, whereby both \( \mathbf{D}_{\text{RGBH}} \) and \( \mathbf{B} \) were subject to a down-sampling 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
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 $h \propto \text{rectify}(R) = 0$. 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

$$\frac{dL}{dw} = \frac{\partial L}{\partial \hat{D}_{\text{RGBH}}} \left( J_{\text{D}_{\text{RGBH}}}(w) + J_{\text{D}_{\text{RGBH}}}(B)J_B(w) \right)^0,$$

(12)

where $J_y(x)$ denotes the Jacobian of $y$ with respect to $x$ 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, $(x_{w,j}, y_{w,j}, D_j)$, is given by

$$B_{j+1} \leftarrow B_j$$

$$B_{j+1}[x_{w,j}, y_{w,j}] \leftarrow mB_j[x_{w,j}, y_{w,j}] + (1 - m)D_j$$

(13)
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 re-projection step. As a result, we only need one pass through 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+ (\(f_{\text{eff}} = 4.3\) mm) and freehand motion, we collected multiple image sequence datasets consisting of 21-23 RGB 1512 \(\times\) 2016 images (2\(\times\)-downsampled from 3024 \(\times\) 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 \(\mu\)m thick per layer) to alter their heights. We measured the thicknesses of the tape-
Importance of undistortion. Figs. 5 and 6 show spuri-
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


