Structure from Motion on XSlit Cameras

Wei Yang, Yingliang Zhang, Jinwei Ye*, Yu Ji, Zhong Li, Mingyuan Zhou, and Jingyi Yu*, *Member, IEEE*

Abstract—We present a structure-from-motion (SfM) framework based on a special type of multi-perspective camera called the cross-slit or XSlit camera. Traditional perspective camera based SfM suffers from the scale ambiguity which is inherent to the pinhole camera geometry. In contrast, an XSlit camera captures rays passing through two oblique lines in 3D space and we show such ray geometry directly resolves the scale ambiguity when employed for SfM. To accommodate the XSlit cameras, we develop tailored feature matching, camera pose estimation, triangulation, and bundle adjustment techniques. Specifically, we devise a SIFT feature variant using non-uniform Gaussian kernels to handle the distortions in XSlit images for reliable feature matching. Moreover, we demonstrate that the XSlit camera exhibits ambiguities in pose estimation process which can not be handled by existing work. Consequently, we propose a 14 point algorithm to properly handle the XSlit degeneracy and estimate the relative pose between XSlit cameras from feature correspondences. We further exploit the unique depth-dependent aspect ratio (DDAR) property to improve the bundle adjustment for the XSlit camera. Synthetic and real experiments demonstrate that the proposed XSlit SfM can conduct reliable and high fidelity 3D reconstruction at an absolute scale.

Index Terms—Multi-perspective imaging, Generalized Structure from Motion, Camera motion estimation, Feature matching, Bundle adjustment

1 Introduction

A perspective camera collects rays passing through a common 3D point (i.e., the Center of Projection or CoP) and produces images similar to those visiable for human eyes. However, this model is rare for insect eyes. For example, many insects have compound eyes consisting of thousands of individual eye units or ommatidia. Located on a convex surface, these ommatidia can receive light from different directions, thus enabling the compound eye to have a very large field-of-view which is extremely helpful for detecting fast movement. Notice that without a common CoP, the perspective camera model no longer applies to the compound eye. Instead, it adopts the *multi-perspective* model and combines rays from different viewpoints. Despite the incongruity of views, a multi-perspective image is able to preserve the spatial coherence of a scene and depict its details that are simultaneously inaccessible from a single view within a single context.

Early works on multi-perspective stereo matching have laid the theoretical foundation for studying multi-perspective camera. The seminal work of Seitz [1] characterizes all possible multi-perspective stereo pairs and concludes that the epipolar geometry, if existing, has to be a doubly ruled surface. Pajdla [2] reaches similar results. Pless [3] further derives the Generalized Epipolar Constraint (GEC) for generic camera models. Yu et al. [4] propose the General Linear Camera (GLC) that characterizes all possible linear multi-perspective cameras under the light field ray space. In particular, the GLC reveals that a total of 8 fundamental multi-perspective cameras on the basis of all multi-perspective cameras where previous stereo matching algorithms are directly applicable for 3D reconstruction.

Among the GLC models, the most fundamental one is the cross-slit (XSlit) camera which captures rays passing through two oblique line slits in 3D space, as it can be viewed as the "simplest" camera following pinhole projections [5]. Many traditional cameras are degenerated cases of the XSlit camera. For example, when the two slits intersect, it degenerates into a pinhole camera with the intersection point being the pinhole; when one of the two slits moves to infinity, it degenerates into the pushbroom camera; when both slits go infinitely far, it degenerates into an orthographic camera. Furthermore, reflections and refractions can both be modeled with the XSlit camera under caustic surface models since the two slits can provide a special set of surface ruling of the caustic surfaces [6].

One key advantage of the XSlit camera against other multi-perspective cameras is that one can use off-the-shelf optical components to construct the XSlit camera. Ji et al. [7] construct a prototype XSlit lens using a pair of cylindrical lenses coupled with slit-shaped apertures in place of a single spherical thin lens. The lens can be mounted on commodity camera to directly produce XSlit images. XSlit defocus blurs analogous to shallow depth-of-field effects in perspective cameras provide additional depth cues that can help recover scene depth and design better coded aperture imaging systems.

This paper discusses how to use the XSlit camera in place of pinhole cameras for 3D reconstruction. In computer vision, two most commonly used 3D reconstruction techniques are stereo matching and structure-from-motion (SfM). For the former, Ye et al. [8] proposed to rotate instead of translate the camera to produce parallax and thereafter conduct feature matching. In particular, they have shown that under rotational settings which they call rotational XSlit (R-XSlit) pair, the images form valid epipolar geometry that

- * *Corresponding authors.*
- *W. Yang, Y. Zhang, Y. Ji, Z. Li and M. Zhou are with DGene (pre. PlexVR), 3500 Thomas RD, Santa Clara, CA, U.S.A. Contact E-mail: wei.yang@dgene.com*
- *J. Ye is now with the Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA, U.S.A.*
- *J. Yu is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China.*

Manuscript received XXX XX, 201X; revised XXX XX, 201X.

obeys the Seitz model. This significantly reduces the search space for feature matching and fits well with many existing stereo solutions such as graph-cut [9]. In contrast, there is very little work on employing the XSlit camera for SfM.

The basic principles of SfM have been thoroughly studied in computer vision. The process consists of three major steps: it first uses techniques such as SIFT [10] or SURF [11] to extract feature points in images; next, SfM establishes feature correspondences and subsequently conducts camera pose estimation by solving for the fundamental matrix or homography matrix between pairwise views; finally it conducts triangulation to reconstruct 3D points and iteratively refines the estimated camera poses. A major drawback in perspective SfM is scale ambiguity [12] caused by the projection process, i.e., one cannot determine the actual scale of the recovered 3D scene even with calibrated cameras [13].

We show that scale ambiguity is a singularity inherited from pinhole cameras and can be directly eliminated by using XSlit cameras. Specifically, we demonstrate aspect ratio distortions in XSlit images provide a vital cue on depth and can directly resolve scale ambiguity. Yet XSlit SfM compared with the perspective one incurs two additional challenges: 1) the XSlit projection is non-linear where the pose solver for perspective cameras is not directly applicable; and 2) although aspect ratio distortions are helpful for resolving scale ambiguity, they make feature matching much more difficult due to distortions.

In this paper, we present the first XSlit-based SfM framework capable of recovering 3D scene geometry from images at an absolute scale. We explore the Depth-Dependent Aspect Ratio (DDAR) property in the XSlit camera and demonstrate how DDAR can be used for resolving scale ambiguity. We first show that, similar to pinhole cameras, there exists a fundamental matrix to correlate two XSlit images captured by the same camera under different poses. We show how to reduce the dimensions in the XSlit fundamental matrix so that absolute translation and rotation matrices can be solved via a linear system. We further tailor a new, robust feature matching algorithm on correlating XSlit images under strong distortions. Finally, we propose a novel error metric based on DDAR for bundle adjustment to iteratively refine the estimated camera parameters and scene geometry. Synthetic and real experiments demonstrate that our XSlit SfM can recover both camera motions and scene geometry at an absolute scale with high fidelity and reliability.

For clarification, We reuse the DDAR discussion (Sec.5.2.1) presented in our previous work [14]. Notice this paper only re-uses the DDAR analysis from [14] for the completeness of the bundle adjustment discussion, the problem and motivation and are completely different. The paper aims to re-invent the SfM pipeline, which uses multiple XSlits images for 3D reconstruction, while [14] focused on XSlit distortion analysis.

2 Related Work

Our work is closely related to SfM and multi-view geometry for non-central cameras (the XSlit in particular). Since the literature is huge, we only discuss the most relevant works in this section.

SfM has been extensively studied in computer vision and great success has been achieved in robotics [15], autonomous navigation [16], large-scale 3D reconstruction [17], [18] etc. Most existing works rely on perspective cameras. However, SfM from a perspective camera suffers from scale ambiguity [19]. Conventional approaches for this problem include using a stereo camera setup with known baseline [15], [20] where the scale factor is determined by triangulating feature points in the stereo pair. Clipp et al. [21] recovered the scale by tracking features on two non-overlapping cameras. In the case of a single perspective camera, it is necessary to have constraints on or priors of either the camera motion or the scene geometry to recover the scale. Scaramuzza et al. [22] used the camera-to-ground distance to keep track of the camera motion and estimate the scale. Davison et al. [15] employed a pattern of known size to compute the absolute scale of the entire scene. Pollefeys et al. [23] utilized an additional GPS sensor to get the exact dimension. In this paper, we exploit a unique attribute of multi-perspective cameras, i.e., aspect ratio distortions, to directly resolve the problem of scale ambiguity.

Remarkable progress has been made in employing multiple non-central cameras for geometry analysis. In the seminar work [3], Pless derived the constraint, i.e. GEC, in terms of the rotation and translation between camera viewpoints imposed by point(ray) correspondences. The GEC suggests a 17-point algorithm that could be used to solve the camera motion linearly. Stewénius et al. [24] further proposed a non-linear 6-point algorithm based on Gröbner-Basis. Later, Li et al. [25] pointed out that there were degeneracies in the 17-point algorithm for certain configurations which lead to a family of solutions. Kim and Kanade [26] further proposed a method to find and prove the degenerate cases of the 17-point algorithm. An XSlit camera [27] as a general camera collects rays passing through two oblique (neither coplanar nor parallel) slits. The XSlit cameras exhibit unique degeneracy that haven't been studied and can not be solved using methods presented in [3], [24], [25], which we discuss in detail in Sec. 3.2. We specifically handle the XSlit degeneracy by exploiting the XSlit constraints and reduce the 6×6 pose matrix to 4×4.

Regarding multiple XSlit cameras, previous studies mainly focused on the epipolar geometry of a pair of XSlit cameras. Feldman et al. [28] derived a 6×6 fundamental matrix using the Veronese mapping and proved that a pair of XSlit cameras can have valid epipolar geometry if they share a slit or the slits intersect in four pairwise distinct points. Trager et al. [5] presented a $2 \times 2 \times 2 \times 2$ epipolar tensor, which is sensitive to noise and not generic. Our work is most relevant to [29], which reduced the GEC to a 4×4 essential matrix on reduced Plücker vectors. However, its deduction was based on a simplified XSlit system and no solution for relative pose was reported. We derive a general constraint on XSlit cameras and propose a effective solution for motion estimation.

In addition, our work is related to the other two major components of SfM, i.e. the feature matching and bundle adjustment. By far, most feature extractors, such as SIFT [10], SURF [11], ORB [30], KAZE [31] and the recent DSP-SIFT [32], [33] are designed for perspective images, i.e., to address the translation, rotation and uniform scaling de-

formations. They cannot handle unique distortions in XSlit images. Affine covariant features, such as Harris and Hessian Affine [34], and MSER [35], are helpful for addressing this problem, but they normally generate a relatively small number of correspondences. An exception is the Affine SIFT (ASIFT) feature descriptor [36], which simulates the images with different camera orientations from a frontal position. ASIFT can generate a great many correspondences when applied to XSlit images since the affine transformation can approximate local XSlit distortions. However, there still exist a large number of mismatches which make the RANSAC unstable. Another way is to manually select and match the feature points in two XSlit images [28]. Apparently, manual selection is not an option for the SfM task because a significant large number of feature correspondence are required for reliable 3D reconstruction. Instead, we present an accurate feature extraction and matching method for large distortions of XSlit images by toning the SIFT feature with non-uniform Gaussian kernels. When applying bundle adjustment, an essential step of SfM, to general cameras, researchers proposed several error metrics, including the object space error [37], angular error [38] and shortest transversal error [39]. In this paper, we introduce a novel error metric based on the XSlit distortion and demonstrate that it can improve the traditional re-projection error based method.

3 XSlit Imaging Model

We model XSlit with ray space geometry. We first demonstrate that similar to the perspective case, there exists a fundamental matrix to correlate a pair of images taken by XSlit cameras at different 3D locations. We then reduce the dimensions in the fundamental matrix by imposing XSlit geometric constraints and solve for the transformation via a linear system.

3.1 XSlit Camera Geometry

We draw on the ray space geometry [40] to analyze XSlit ray structures and further derive the epipolar geometry in XSlit images. To represent a ray in ray space, we adopt the Two-Plane Parametrization (2PP) [41] that is widely used in previous works on light fields. With 2PP, a ray is represented by its intersections with two parallel planes Π_{uv} and Π_{st}. For simplicity, we use the image plane at $z = 0$ as Π_{uv} and the plane at $z = 1$ as Π_{st}. Consequently, a rays direction can be written as $[\sigma, \tau, 1] = [s - u, t - v, 1]$. Then we have a 4D parameter $[u, v, \sigma, \tau]$ for each ray in space.

To construct an XSlit camera, we assume the image plane ($z = 0$) is parallel to the two slits' planes. As shown in Fig. 1, the two slits are at depth z_1 and z_2 and form angle θ_1 and θ_2 w.r.t the u-axis, where $z_2 > z_1$ and $\theta_1 \neq \theta_2$. With this configuration, a ray collected by the XSlit camera follows the linear constraints:

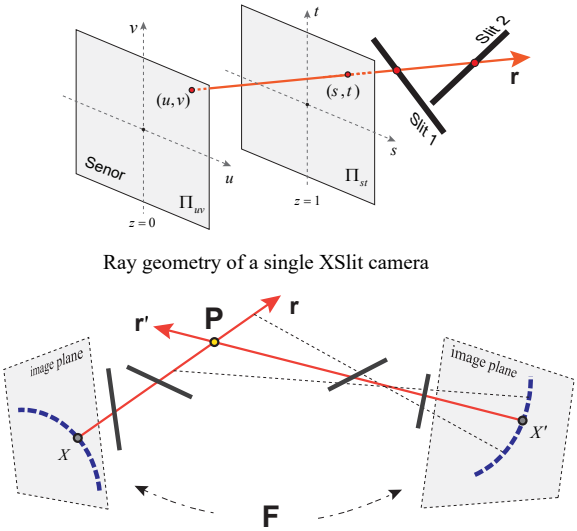
$$\sigma = (\mathsf{A}u + \mathsf{B}v)/\mathsf{E} \qquad \tau = (\mathsf{C}u + \mathsf{D}v)/\mathsf{E} \tag{1}$$

where

$\mathsf{A} = z_2 \cos\theta_2 \sin\theta_1 - z_1 \cos\theta_1 \sin\theta_2$, $\mathsf{B} = (z_1 - z_2) \cos\theta_1 \cos\theta_2$

$\mathsf{C} = (z_1 - z_2) \sin\theta_1 \sin\theta_2$, $\mathsf{D} = z_1 \cos\theta_2 \sin\theta_1 - z_2 \cos\theta_1 \sin\theta_2$

$\mathsf{E} = z_1 z_2 \sin(\theta_2 - \theta_1)$.



Ray geometry of a single XSlit camera

The two view geometry of a pair of XSlit cameras

Fig. 1. XSlit ray geometry. Top: Ray geometry of a single XSlit camera. Bottom: XSlit images captured from different viewpoints are correlated by a fundamental matrix $\mathbf{F}$.

We call Eqn. 1 the XSlit constraints. Previous studies reached similar conclusions in various forms [4], [27], [42].

3.2 XSlit Fundamental Matrix

Given a reference XSlit image $\mathbb{X}$ and a target XSlit image $\mathbb{X}'$ captured at different viewpoints, our goal is to align $\mathbb{X}'$ to $\mathbb{X}$ via a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$. Consider a 3D scene point $\mathbf{P}$ as shown in Fig. 1, we can "project" $\mathbf{P}$ to $\mathbb{X}$ and $\mathbb{X}'$ by using corresponding rays $\mathbf{r}[u, v, \sigma, \tau]$ and $\mathbf{r}'[u', v', \sigma', \tau']$ passing through the point. Assume that the world coordinate is aligned with the reference image $\mathbb{X}$, we first transform $\mathbf{r}'$ into the world coordinate $\mathbf{r}^*[u^*, v^*, \sigma^*, \tau^*]$. Since $\mathbf{r}$ and $\mathbf{r}^*$ pass through $\mathbf{P}$, their ray coordinates should satisfy a bilinear constraint [6]: $\dfrac{u - u^*}{v - v^*} = \dfrac{\sigma - \sigma^*}{\tau - \tau^*}$. Its vector form can be written as:

$$\mathbf{d}^{\mathsf{T}} \cdot \mathbf{m}^* + \mathbf{m}^{\mathsf{T}} \cdot \mathbf{d}^* = 0 \tag{2}$$

where $\mathbf{d} = [\sigma, \tau, 1]^{\mathsf{T}}$, $\mathbf{m} = [-v, u, \chi]^{\mathsf{T}}$, $\chi = v\sigma - u\tau$ and $\mathbf{d}^* = [\sigma^*, \tau^*, 1]^{\mathsf{T}^*}$, $\mathbf{m}^* = [-v^*, u^*, \chi^*]^{\mathsf{T}}$, $\chi^* = v^*\sigma^* - u^*\tau^*$.

Similarly, we define $\mathbf{d}'$ and $\mathbf{m}'$ for $r'[u', v', \sigma', \tau']$. Since the two image coordinates in $\mathbb{X}$ and $\mathbb{X}'$ are correlated by transformation matrices $\mathbf{R}$ and $\mathbf{t}$, we can derive the relationship between $\mathbf{d}'$, $\mathbf{m}'$ and $\mathbf{d}^*$, $\mathbf{m}^*$ as:

$$\mathbf{d}^* = \mathbf{R} \cdot \mathbf{d}' \qquad \mathbf{m}^* = \mathbf{R} \cdot \mathbf{m}' - [\mathbf{t}]_\times \mathbf{R} \cdot \mathbf{d}' \tag{3}$$

Substituting Eqn. 3 into Eqn. 2, we have:

$$\begin{bmatrix} \mathbf{d}^{\mathsf{T}} & \mathbf{m}^{\mathsf{T}} \end{bmatrix} \mathbf{E}_{6\times 6} \begin{bmatrix} \mathbf{d}' \\ \mathbf{m}' \end{bmatrix} = 0 \tag{4}$$

This reveals that XSlit images captured by the same camera but at different locations are correlated by a general essential matrix as $\mathbf{E}_{6\times 6} = \begin{bmatrix} -[\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & 0 \end{bmatrix}$.

Since both $\mathbf{d}$ and $\mathbf{m}$ are determined by the image pixel coordinate $[u, v]$, Eqn. 4 reveals that there exists an essential

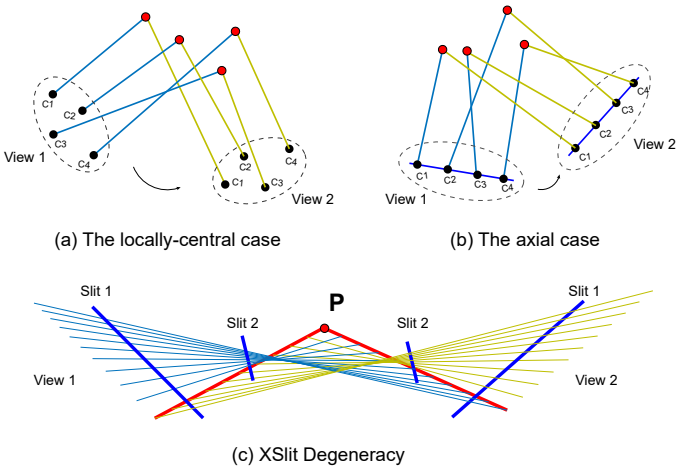


Fig. 2. The degenerated cases of the generalized epipolar constraint. Top: the locally-central case, where for every 3D point the projection center is fixed across views, and the axial case, where all image rays must intersect a common line, identified by [25] and [26]. Bottom: The XSlit degeneracy where corresponding rays across views must lie on a doubly ruled surface.

matrix $\mathbf{E}_{6\times6}$, similar to the one in the perspective case, for two XSlit images captured at different viewpoints. Our derivation so far is performed in the ray space and is consistent with the GEC [3], which uses Plücker formulation for more general cameras.

XSlit Degeneracy Analysis: Intuitively, we can solve for the matrix as a linear system by treating $\mathbf{E_t} = -[\mathbf{t}]_\times\mathbf{R}$ and $\mathbf{R}$ as two independent unknowns and using 17 pairs of correspondences to solve the following equations

$$\mathbf{d}^\mathsf{T}\mathbf{E_t}\mathbf{d}' + \mathbf{d}^\mathsf{T}\mathbf{R}\mathbf{m}' + \mathbf{m}^\mathsf{T}\mathbf{R}\mathbf{d}' = 0 \tag{5}$$

In reality, the linear equations may exhibit degeneracies due to the ray constraints imposed by a general camera model. To demonstrate that XSlit is a degenerated case, we will show that a linear family of solutions exist. Specifically, we use $\mathbf{v}_i$ to represent a point on the slit i and $\mathbf{w}_i$ to represent the slit direction. The moment of the ray $\mathbf{m} = (\mathbf{v}_i + \alpha_i\mathbf{w}_i) \times \mathbf{d}$, where α_i is a scalar to make sure the $\mathbf{v}_i + \alpha_i\mathbf{w}_i$ is the intersection of the ray and the slit i. Eqn. 5 can be rewritten as:

$$\begin{aligned}\mathbf{d}^\mathsf{T}\mathbf{E_t}\mathbf{d}' + \mathbf{d}^\mathsf{T}\mathbf{R}\big[(\mathbf{v}_i + \alpha_i'\mathbf{w}_i) \times \mathbf{d}'\big] \\ + \big[(\mathbf{v}_i + \alpha_i\mathbf{w}_i) \times \mathbf{d}\big]^\mathsf{T}\mathbf{R}\mathbf{d}' = 0\end{aligned} \tag{6}$$

Further simplify the equation, we have:

$$\mathbf{d}^\mathsf{T}\mathbf{E}_{\mathbf{t}+\mathbf{v}_i}\mathbf{d}' + \alpha_i'\mathbf{d}^\mathsf{T}\mathbf{R}(\mathbf{w}_i \times \mathbf{d}') + \alpha_i(\mathbf{w}_i \times \mathbf{d})^\mathsf{T}\mathbf{R}\mathbf{d}' = 0 \tag{7}$$

The general solution to Eqn. 7 is $(\lambda\mathbf{E_t} - \mu\mathbf{E}_{\mathbf{v}_i}, \lambda\mathbf{R} + \mu\mathbf{w}_i\mathbf{w}_i^\mathsf{T})$. Notice that there are two slits in a XSlit camera. If $(-\mathbf{E}_{\mathbf{v}_1}, \mathbf{w}_1\mathbf{w}_1^\mathsf{T})$ and $(-\mathbf{E}_{\mathbf{v}_2}, \mathbf{w}_2\mathbf{w}_2^\mathsf{T})$ are independent, we can recover the true solution $(\mathbf{E_t}, \mathbf{R})$ unambiguously. However, this is not true as we can set the coordinate system as follows: let z axis intersect with both slits and x axis is the half-vector of the two slits directions. In this coordinate system, $\mathbf{w}_1\mathbf{w}_1^\mathsf{T} = \mathbf{w}_2\mathbf{w}_2^\mathsf{T}$ and $\mathbf{v}_1 \propto \mathbf{v}_2$. The previously identified general solution satisfy both equations determined by the two slits. Hence the XSlit camera corresponds to a degenerated case and the 17-pt algorithm won't work. Previous degeneracy analysis [25], [26] mainly focused on the multi-camera rig and identified the locally-central case as in Fig. 2(a) and the axial case as in Fig. 2(b). One crucial requirement for Li et al.'s method [25] to be applicable is that all the ambiguity lies in the determination of the $\mathbf{R}$ part and the $\mathbf{E_t}$ part of the solution is unchanged by the ambiguity. Li et al.'s method handles the axial case by setting the origin of the coordinate system on the axis which lead $\mathbf{v} = 0$ in previous general solution. Such trick can not be applied to the XSlit camera and there is no easy solution to enforce the ubiquity on $\mathbf{E_t}$, see Fig. 2.

To handle the XSlit degeneracy and further reduce the dimension in $\mathbf{E}_{6\times6}$, we exploit the XSlit constraints (Eqn. 1) to decompose $[\mathbf{d}\ \mathbf{m}]^\mathsf{T}$ into two matrices as:

$$\begin{bmatrix}\mathbf{d}\\ \mathbf{m}\end{bmatrix} = Kp^\mathsf{T} \tag{8}$$

where

$$K = \begin{bmatrix} 0 & -\mathsf{B} & \mathsf{A} & 0 \\ 0 & -\mathsf{D} & \mathsf{C} & 0 \\ \hline & & I_{4\times4} & \end{bmatrix}$$

and $p = [1, -v, u, \chi]$. Notice that K is only related to the configuration of the two slits, so we call it XSlit intrinsic matrix, where p is determined by pixel coordinate $[u, v]$. Substituting K, K' and p, p' into Eqn. 4, we have:

$$p^\mathsf{T}\mathbf{F}p' = 0 \qquad \text{where } \mathbf{F} = K^\mathsf{T}\mathbf{E}_{6\times6}K' \tag{9}$$

Recall that p is a 1×4 vector, our XSlit fundamental matrix $\mathbf{F}$ is a 4×4 matrix with its last element being zero. As a result, we should be able to solve the unknown elements in $\mathbf{F}$ with 14 pairs of corresponding points between $\mathbb{X}$ and $\mathbb{X}'$ by applying Singular Value Decomposition (SVD).

3.3 Pose Estimation

Next, we use the fundamental matrix $\mathbf{F}$ to solve for camera pose transformation matrices $\mathbf{R}$ and $\mathbf{t}$. Notice that we cannot achieve this directly from Eqn. 9 since the underdetermined XSlit intrinsic matrix K is uninvertible. To address this problem, we apply QR matrix decomposition on K and convert K into the multiplication of an orthogonal matrix $\mathcal{Q}$ and an upper triangular matrix $\mathcal{R}$, i.e., $K = \mathcal{Q}\mathcal{R}$. Decomposing K' similarly and substituting K, K' into Eqn. 9, we have

$$\hat{\mathbf{F}} = \mathcal{R}_4^{-\mathsf{T}}\mathbf{F}\mathcal{R}_4'^{-1} = \left(\mathcal{Q}^\mathsf{T}\begin{bmatrix}-[\mathbf{t}]_\times\mathbf{R} & \mathbf{R}\\ \mathbf{R} & 0\end{bmatrix}\mathcal{Q}'\right)_4 \tag{10}$$

where the subscript 4 means 4×4 sub-matrix from the upper left corner. Remove trivial zero elements in the equation, we can rewrite Eqn. 10 as

$$\hat{\mathbf{F}} = \begin{bmatrix}\mathrm{M} & \mathrm{V}\\ \mathrm{U} & 0\end{bmatrix} = \begin{bmatrix}\mathcal{Q}_1^\mathsf{T} & \mathcal{Q}_3^\mathsf{T}\\ 0 & \mathbf{e}_3^\mathsf{T}\end{bmatrix}\begin{bmatrix}-[\mathbf{t}]_\times\mathbf{R} & \mathbf{R}\\ \mathbf{R} & 0\end{bmatrix}\begin{bmatrix}\mathcal{Q}_1' & 0\\ \mathcal{Q}_3' & e_3\end{bmatrix} \tag{11}$$

where M is a 3×3 sub-matrix of $\hat{\mathbf{F}}$, U and V are 3×1 vectors, $\mathcal{Q}_1$, $\mathcal{Q}_3$ are 3×3 sub-matrices of $\mathcal{Q}$ and $\mathcal{Q}_1'$, $\mathcal{Q}_3'$ are

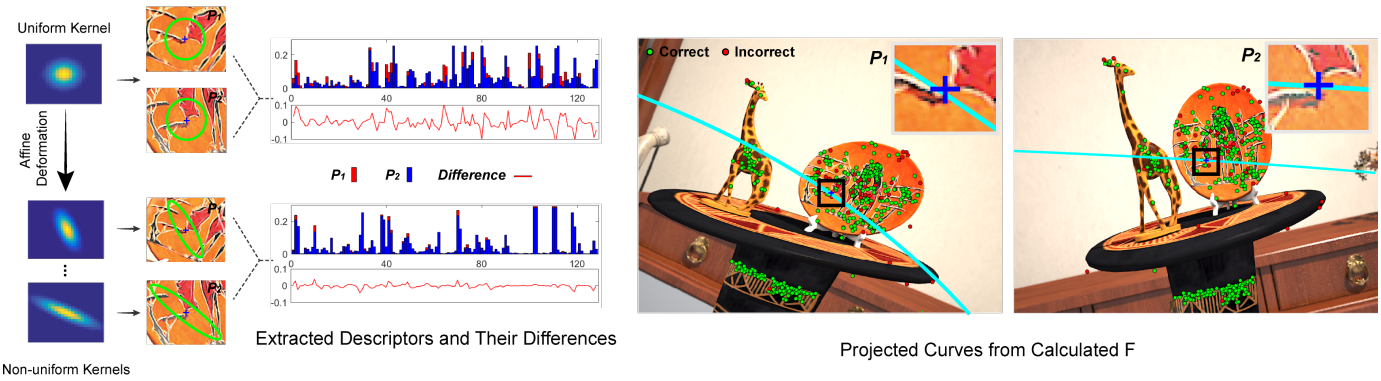


Fig. 3. An example of our feature matching technique. Left: We disturb the Guassian kernel in SIFT using affine deformation and apply them to patches in XSlit image pairs. Middle: A better match can be find by non-uniform kernels. Right: Projected curves using the fundamental matrix calculated from correspondences produced by our method.

3×3 sub-matrices of $\mathcal{Q}'$, and $\mathsf{e}_3 = [0, 0, 1]^\mathsf{T}$. From Eqn. 11, we can derive the following constraints on $\mathbf{R}$ and $\mathbf{t}$:

$$\mathrm{U} = \mathsf{e}_3^\mathsf{T} \cdot \mathbf{R}\mathcal{Q}_1' \tag{12a}$$

$$\mathrm{V} = \mathcal{Q}_1^\mathsf{T}\mathbf{R} \cdot \mathsf{e}_3 \tag{12b}$$

$$\mathrm{M} = \mathcal{Q}_1^\mathsf{T}(-[\mathbf{t}]_\times \mathbf{R})\mathcal{Q}_1' + \mathcal{Q}_3^\mathsf{T}\mathbf{R}\mathcal{Q}_1' + \mathcal{Q}_1^\mathsf{T}\mathbf{R}\mathcal{Q}_3' \tag{12c}$$

From Eqn. 12a and Eqn. 12b, we have:

$$\mathbf{R} \cdot (\mathcal{Q}_1'^{-\mathsf{T}}\mathrm{U}^\mathsf{T}) = \mathsf{e}_3 \qquad \mathbf{R} \cdot \mathsf{e}_3 = \mathcal{Q}_1^{-\mathsf{T}}\mathrm{V} \tag{13}$$

Add another constraint according to the rotation matrix properties.

$$\mathbf{R} \cdot [\mathcal{Q}_1'^{-\mathsf{T}}\mathrm{U}^\mathsf{T} \times \mathsf{e}_3] = \mathsf{e}_3 \times \mathcal{Q}_1^{-\mathsf{T}}\mathrm{V} \tag{14}$$

where $\times$ means the cross product. From Eqn. 13 and 14, we can find an initial estimation $\mathbf{R}$ and then get the optimal solution by enforcing orthogonality using SVD. Then we can solve the translation vector $\mathbf{t}$ from Eqn. 12c by plugging in $\mathbf{R}$.

3.4 Scale Ambiguity

Our formulation reveals why perspective SfM exhibits scale ambiguity: recall that the perspective camera is a special case of XSlit camera where the two slits are at the same distance $z_1 = z_2 = f$, and f is the focal length. In Eqn. 1, we have $\mathsf{B} = \mathsf{C} = 0$ and $\mathsf{A} = \mathsf{D}$. The ray constraints for a perspective camera degenerate to $\sigma = -\frac{1}{f}u, \tau = -\frac{1}{f}v$. As a result, for a perspective camera, the higher order terms $\chi = v\sigma - u\tau = 0$ and Eqn. 12a and Eqn. 12b become trivial, and Eqn. 12c degenerates into $\mathrm{M} = \mathcal{Q}_1^\mathsf{T}(-[\mathbf{t}]_\times \mathbf{R})\mathcal{Q}_1'$. This reveals that we can only recover the direction of $\mathbf{t}$ but not its actual scale.

From a geometric perspective, if we scale the entire scene along with the camera positions by a factor k, the projections of the scene points in the image will be exactly the same. However, it will be very different for XSlit cameras: if we scale both the scene and the camera positions, the absolute distance between camera and the object will change, thus leads to a different XSlit projection and yields a different solution to Eqn. 12c. In other words, the scale ambiguity is automatically resolved in XSlit SfM. However, this is under the assumption that one can reliably establish feature correspondences, which, however, is much more difficult in the XSlit camera due to distortions.

4 XSlit Feature Matching

Different from a perspective image, an XSlit image exhibits various distortions, making it difficult to conduct correspondence matching using conventional techniques.

4.1 Distortion Analysis

It is well known that perspective images exhibit wide-angle and telephoto distortions. Distant objects may seem abnormally large or small relative to objects closer to the pinhole camera. But perspective images still preserve strong geometric cues about the scene, e.g. lines are projected as lines and frontal parallel objects keep their aspect ratios. Researchers developed various techniques, such as orientation estimation and searching in scale space, to well address the case.

In contrast, the XSlit cameras exhibit caustic distortions [43] in which straight lines in the scene appear curved in the image. Furthermore, XSlit images exhibit different extension and compression rates along the two slits directions, which makes the frontal parallel squares projecting into rectangles. Such distortions nullify the feature extraction and matching methods developed for perspective images. We refer the readers to [44] on a comprehensive discussion on XSlit image distortions and their causes.

Most feature extractors, such as SIFT [10], SURF [11], ORB [30], KAZE [31] are designed for perspective images, i.e., to address the translation, rotation and uniform scaling deformations. Even with a small change in viewpoint, the XSlit distortions may still lead to strong and apparent deformations that we can not describe simply by similar transformations.

An exception is the Affine SIFT (ASIFT) feature descriptor [36], which simulates the images with different camera orientations from a frontal position. The simulated images are obtained by transforming the original image according to longitude angle ϕ, latitude angle ψ and transition tilt t as follows:

$$I_t = R(\phi)T(t)R(\phi) \cdot I_o \tag{15}$$

where I_o is the original image and I_t is the simulated image. $R(x)$ is the rotation matrix in 2D with angle x. $T(t)$ is a diagonal matrix with first eigenvalue $t > 1$ and the second one equal to 1.

ASIFT can generate a great many correspondences when applied to XSlit images since the affine transformation can approximate local XSlit distortions. However, there still exist a large number of mismatches.

4.2 Matching with Non-Uniform Gaussian Kernels

To properly handle distortions in XSlit images, we develop a new feature matching algorithm based on non-uniform Gaussian kernels. Similar to ASIFT, we sample SIFT features in different subspaces in order to undistort the XSlit image patch. However, the difference is that we use non-uniform Gaussian kernels to sample subspaces instead of perspective warping used in ASIFT.

Specifically, an affine transformation can be defined by a rotation angle θ, a shear factor s and a scale factor r:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix} \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (16)$$

We then apply affine transformation matrix (Eqn. 16) to a Gaussian kernel to obtain non-uniform Gaussian kernels g as

$$g(x, y, \sigma, \theta, s, r) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \quad (17)$$

We use the non-uniform Gaussian kernels g to transform the XSlit image patch I into feature subspaces. The non-uniform Gaussian kernels are applied at both the key point detection and the descriptor extraction stage. Per our experience, apply the kernels for the key point detection provides more candidates for the feature matching while won't affect the precision much. And we extract the feature descriptor only on the patch covered by the non-uniform kernel. The blocks for the histogram accumulation are deformed accordingly. Compared to uniform kernels, more accurate matching can be found via. non-uniform patches as shown in Fig. 3. In our feature subspace, we can compensate various distortions properly. However, mismatched correspondences can still occur occasionally. To address this problem, we perform bi-directional search for valid correspondences. In particular, we first make feature matching from the reference image to the target image and then reverse the search direction and match features from the target to the reference. We only keep those correspondences that exist in both rounds.

Fig. 3 illustrates our feature matching algorithm vs. the state-of-the-arts. The left-top bar chart shows that our algorithm outperforms other methods in terms of both the number of detected feature points and the mismatch rate. Although ASIFT detects more feature points than our algorithm, its mismatch rate is extremely high. We also show the "epipolar curves" produced by the fundamental matrix calculated from our features and ASIFT features. Apparently, our curves establish more accurate correspondences and manage to achieve sub-pixel accuracy.

We first evaluate the non-uniform Gaussian based feature matching algorithm. We show that our feature detector is able to handle drastic changes in viewpoint, which is locally analogous to strong geometric distortions in XSlit images. We test our algorithm with the Graffiti dataset [45] which contains images captured with viewing angles ranging from 20° to 60°. Fig. 4(a) shows a subset of matched feature points produced by our algorithm. The viewing angle difference between the two images is 60°. The overlaid parallelograms illustrate the affine transformations used to generate feature points. This example shows that our algorithm is able to handle images with big changes in viewpoint.

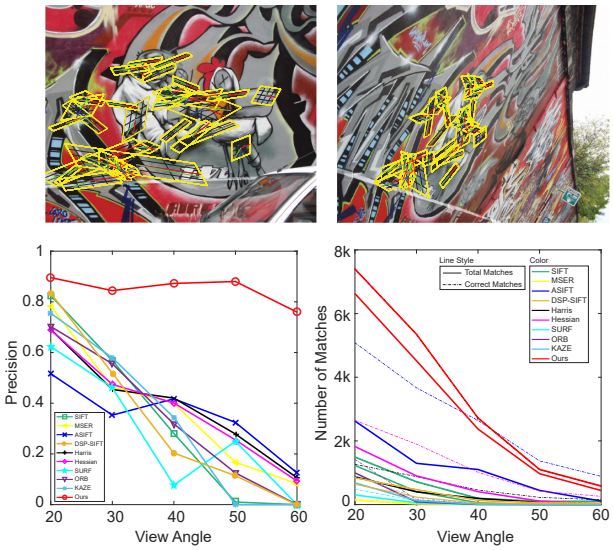


Fig. 4. Feature matching evaluation. Top: Subset of feature points generated by our algorithm. The overlaid parallelograms illustrate the affine transformations used to produce features; Bottom-left: Precision curves in comparison with state-of-the-arts; Bottom-right: Numbers of valid features (correctly matched) and all matched features w.r.t. different viewing angles.

	Matches		RANSAC	
	Total	Correct	Iter	Time(s)
SIFT	201	125	1.8e4	9.34
MSER	16	7	-	-
ASIFT	900	354	1.4e4	26.18
DSP-SIFT	140	65	7.2e5	284.22
Harris	249	129	2258	1.38
Hessian	294	156	1913	3.38
SURF	30	4	-	-
ORB	40	18	675	0.20
KAZE	98	54	531	0.21
Ours	540	469	184	0.27

TABLE 1
Comparisons of our feature matching technique vs. the state-of-the-art methods on XSlit image pair.

To quantitatively evaluate the performance, we make a precision curve, showing the ratio between the number of valid features and all matched features. Specifically, we compare our method with the state-of-the-art feature detectors, such as SIFT,DSP-SIFT, SURF, ORB, KAZE, Harris, Hessian, MSER and ASIFT. Valid feature points are defined as a pair of corresponding points within 1.5 pixels after being warped by the estimated homography. In Fig. 4, the bottom-left image shows the precision curve w.r.t the change in viewing angle and the right one shows the number of valid features and all matched features. Although the precision curves of all other feature detectors descend rapidly when the viewing angle variation increases, the ratio remains high in our method. This is because the non-uniform Gaussian kernels adopted in our approach are effective in handling

large distortions. Although in some cases the ASIFT detects more feature points in total, its mismatch rate is very high. To further show the effectiveness of our feature matching method on XSlit images, we compare on the XSlit image pair shown in Fig. 3. The match result is shown in Tab. 1 and our method reports the highest number of correct matches as well as the precision. The precision of the matched features is important for XSlit images as the RANSAC framework choose matches randomly for pose estimation. We require 14 correspondences for pose estimation which is much larger than 5 or 8 pairs in perspective cameras. With 50% correct matches, the chance is 6E-5 that all 14 matches are inliers. We compare the performance of RANSAC in Tab. 1 that our method is significant faster while reporting sufficient number of matches.

5 Scene Reconstruction

After extracting correspondences and estimating camera poses, we set out to recover 3D scene geometry (i.e., 3D point cloud) by triangulating camera projection rays.

5.1 Points Triangulation

Consider a 3D point $\mathbf{P}[x, y, z]$ in a camera view with world to camera transformation matrix $\mathbf{R}$ and vector $\mathbf{t}$. Assume $\mathbf{P}$ is projected by ray $\mathbf{r}[u, v, \sigma, \tau]$ onto the image plane. The point projection in XSlit can be formulated as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \lambda \cdot \mathbf{R}^{-1} \begin{bmatrix} \sigma \\ \tau \\ 1 \end{bmatrix} = \mathbf{R}^{-1} \cdot \Big(\begin{bmatrix} u \\ v \\ 0 \end{bmatrix} - \mathbf{t} \Big) \tag{18}$$

where λ is the ray propagation factor.

Our goal is to calculate the 3D coordinate (x, y, z) of the scene point $\mathbf{P}$. We treat λ as an independent variable, then Eqn. 18 becomes a linear constraint. Given N views, we can formulate three equations for each viewpoint using Eqn. 18. With $3N$ equations in total, we stack these equations into a linear system and solve the $N + 3$ unknowns (x, y, z and N λ values) by applying SVD.

5.2 Bundle Adjustment

Recall that in perspective camera based SfM, bundle adjustment aims to refine both camera poses and scene geometries as follows:

$$E_p = \sum_i^m \sum_j^n w_i^j \Big\| \Phi_p \Big[K_p,\ \mathcal{T}_p(\mathbf{P}_j,\ \langle \mathbf{R}_i, \mathbf{t}_i \rangle) \Big] - \mathrm{x}_{ij} \Big\|^2 \tag{19}$$

where w_i^j is a binary variable indicating the visibility of the jth 3D point in camera i (1 means visible). K_p is the intrinsic matrix of the perspective camera; $\mathcal{T}_p$ transforms the 3D point $\mathbf{P}_j$ into the camera coordinate using $\mathbf{R}_i$, $\mathbf{t}_i$; and Φ_p is the perspective projection function defined as:

$$\Phi_p \Big[K_p,\ \mathcal{T}_p(\mathbf{P},\ \langle \mathbf{R}, \mathbf{t} \rangle) \Big] = K_p(\mathbf{R}P + \mathbf{t}) \tag{20}$$

Bundle adjustment for SfM with a perspective camera utilizes the 2D coordinates re-projection errors of detected feature points as the metric to do the final refinement. Similar to the perspective case, we can write the re-projection error for XSlit camera as:

$$E_d = \sum_i^m \sum_j^n w_i^j \Big\| \Phi \Big[K,\ \mathcal{T}(\mathbf{P}_j,\ \langle \mathbf{R}_i, \mathbf{t}_i \rangle) \Big] - \mathrm{x}_{ij} \Big\|^2 \tag{21}$$

where K is the intrinsic matrix of XSlit; $\mathcal{T}$ transforms the 3D point $\mathbf{P}_j$ into the camera coordinate like $\mathcal{T}_p$ in Eq. 19 ; and Φ is the XSlit projection function defined as:

$$\Phi(K, \mathbf{P}) = \mathsf{E} \cdot \begin{bmatrix} \mathsf{E} + \mathsf{A}z & \mathsf{B}z \\ \mathsf{C}z & \mathsf{E} + \mathsf{D}z \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{22}$$

One problem of standard BA is it suffers from drift along optical axis which is manifested in continuous stretching of the estimated trajectory compared to ground truth. While position estimation perpendicular to motion heading is more accurate than along the optical axis [46]. The rooted cause is that the re-projection error of standard BA is insensitive to deviations along the projection ray as all 3D points on the ray will project to the same pixel.

In the XSlit case, we exploit a unique type of distortion, i.e., aspect ratio distortion, to improve the repojection error based bundle adjustment. In perspective cameras, images of a frontal-parallel 3D object keep the aspect ratio (AR) invariant to the depth. In contrast, the aspect ratio of an object changes with its depth in an XSlit camera. This depth dependent aspect ratio (DDAR) property provides additional cues to diversify the optimization objective along the projection ray direction which can help to improve the result of bundle adjustment.

5.2.1 Depth Dependent Aspect Ratio Analysis

To analyze the aspect ratio distortion of a XSlit camera, we first discuss its projection model analogous to pinhole projection. When we map a 3D point $\mathbf{P}[x, y, z]$ to image pixel $p(u, v)$ via a XSlit camera, we can describe it as follows: firstly, we decompose the x-y components of $\mathbf{P}$ into three basis vectors, $\mathbf{v_1}[\cos\theta_1, \sin\theta_1, 0]$, $\mathbf{v_2}[\cos\theta_2, \sin\theta_2, 0]$, $\mathbf{e_3}[0, 0, 1]$ and represent it as $[\kappa_x, \kappa_y, z]$. Then we project the $[\kappa_x, \kappa_y]$ components to $[\kappa_u, \kappa_v]$. Each component can be viewed as pinhole projection since they are parallel to the slits. Finally, we obtain the mapping from $\mathbf{P}$ to p.

$$\mathbf{P} = \kappa_x \mathbf{v_1} + \kappa_y \mathbf{v_2} + z\mathbf{e_3}$$

We then project $\kappa_x \mathbf{v_1}$ and $\kappa_y \mathbf{v_2}$ independently. Notice that the two components are at depth z. And $\kappa_x \mathbf{v_1}$ is parallel to slit 1 and $\kappa_y \mathbf{v_2}$ to slit 2. Their projections emulate the pinhole projection except that the focal lengths are different:

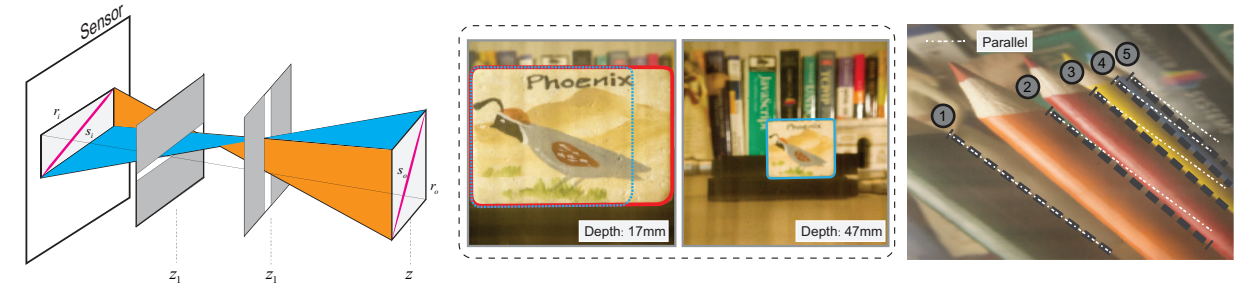
$$\kappa_u = -\frac{z_2}{z - z_2}\kappa_x \qquad \kappa_v = -\frac{z_1}{z - z_1}\kappa_y \tag{23}$$

Since the XSlit mapping is linear, we can combine κ_u and κ_u to compute p.

$$p = \kappa_u \mathbf{v_1} + \kappa_v \mathbf{v_2} \tag{24}$$

where κ_u and κ_v are the linear representations of p on the basis of $\mathbf{v_1}$ and $\mathbf{v_2}$.

Equation 23 reveals that κ_x and κ_y are projected to κ_u and κ_v with different scales on the two directions parallel to the slits. In other words, with the change of depth, the



(a) XSlit projection analogous to perspective projections along two slits directions (b) Depth Dependent Aspect Ratio of XSlit images (c) Depth Dependent Slope of XSlit images

Fig. 5. XSlit camera exhibits depth dependent aspect ratio and depth dependent line slopes.

ratio will change accordingly. Specifically, we can calculate the ratio as:

$$\frac{\kappa_u}{\kappa_v} = \frac{z_2(z - z_1)}{z_1(z - z_2)} \frac{\kappa_x}{\kappa_y} \tag{25}$$

This is fundamentally different from the pinhole/perspective case where the ratio remains static across depth. Recall that the pinhole camera can be viewed as a special XSlit camera where the two slits intersect, i.e., they are at the same depth $z_1 = z_2$. In that case, Eqn. 25 degenerates to $\frac{\kappa_x}{\kappa_y} = \frac{\kappa_u}{\kappa_v}$, i.e., the aspect ratio is invariant to depth.

We use $r_o = \frac{\kappa_x}{\kappa_y}$ to represent the base aspect ratio, and $r_i = \frac{\kappa_u}{\kappa_v}$ the aspect ratio after XSlit projection. From Eqn. 25, we can derive:

$$z_1(z - z_2)r_i - z_2(z - z_1)r_o = 0 \tag{26}$$

Furthermore, we can consider a line frontal-parallel to the XSlit camera as the diagonal of a parallelogram, whose sides are along the directions of the two slits. Given a line with slope s and a point $\mathbf{P}_1[x_1, y_1, z]$ on it, we have $\mathbf{P}_2[x_1 + 1, y_1 + s, z]$ on the line. We can map it to a line with slope s' on XSlit image, where $\mathbf{P}_1$ and $\mathbf{P}_2$ map to points $p_1(u_1, v_1)$ and $p_2(u_1 + c, v_1 + cs')$ respectively. According to the definition of r_o, we can decompose the segment $\mathbf{P}_1$-$\mathbf{P}_2$ to the direction of the two slits and take the ratio of the two components to get r_o:

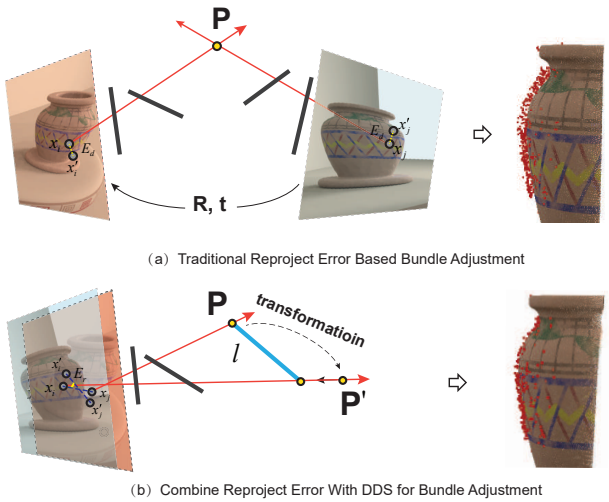
$$r_o = \frac{\sin\theta_2 - s\sin\theta_2}{s\cos\theta_1 - \cos\theta_2} \tag{27}$$

r_i is also calculated from Eqn. 27 by simply substituting s with s'. Eqn. 27 and 26 reveal that the slopes of a frontal-parallel line and its projection should satisfy certain criteria, see Fig. 5. Such inference cannot work in the pinhole camera since the slope of the frontal-parallel line is always the same as observed slope.

5.2.2 Depth Dependent Slope Error

Our aspect ratio analysis leads to a new type of error metric. Specifically, we integrate the depth-dependent slope (DDS) as an additional constraint. DDS indicates that the slope of a frontal-parallel line segment in an XSlit image changes according to its depth to the camera.

For each pair of XSlit views $\mathbb{X}_i$ and $\mathbb{X}_j$ with rotation $\mathbf{R}_{ij}$ and translation $\mathbf{t}_{ij}$ as relative pose from view j to view i. The 3D Point $\mathbf{P}_k$ projects to pixel x_{ik} in $\mathbb{X}_i$, and pixel x_{jk} in $\mathbb{X}_j$. Instead of thinking camera is moving and the scene



(a) Traditional Reproject Error Based Bundle Adjustment

(b) Combine Reproject Error With DDS for Bundle Adjustment

Fig. 6. Two error metrics for XSlit bundle adjustment. (a) left: Reprojection error E_d: distance between the projected and observed points; (b) left: Depth-dependent slope (DDS) error E_r: difference between the projected and observed slopes; Right: Bundle adjustment (BA) comparison without (a) and with DDS (b) error metric.

is static, alternatively we assume that the camera is static and the scene moves reversely (e.g. $\mathbf{P}_k$ moves reversely to $\mathbf{P}'_k = \mathbf{R}_{ij}^T\mathbf{P}_k - \mathbf{R}_{ij}^T\mathbf{t}_{ij}$). Notice this imitation is valid as the generated images under both views $\mathbb{X}_i$ and $\mathbb{X}_j$ remain unchanged. Under our assumption, the 3D point $\mathbf{P}_k$ moves to $\mathbf{P}'_k$ while its projection moves from pixel x_{jk} to x_{ik}. We denote the slope of line segment connecting x_{ik} and x_{jk} as s_α. According to Eqn. 27, we can get the observed aspect ratio after projection as: $r_\alpha = (\sin\theta_2 - s_\alpha\sin\theta_1)/(s_\alpha\cos\theta_1 - \cos\theta_2)$.

Per our discussion in Sec. 5.2.1, the aspect ratio before projection only applies to frontal-parallel line segments. We consider the frontal-parallel line segment ℓ passing through $\mathbf{P}_k$ and intersecting with the projection ray of $\mathbf{P}'_k$, as shown in Fig. 6(b). Since ℓ is frontal-parallel, we can easily calculate its slope s_β by tracking $\mathbf{P}'_k$ along its projection ray to $\mathbf{P}_k$'s depth z_k. Similarly, we can also calculate the aspect ratio r_β before projection. We align the world coordinate system with the first XSlit camera view $i = 0$ and combine all errors from view j to the first view as our final DDS error. Hence the error metric based on DDS is as follows:

$$E_r = \sum_j^N \sum_k^K (v_i^j \| z_1(z_k - z_2)r_\alpha - z_2(z_k - z_1)r_\beta \|^2)_{i=0} \tag{28}$$

We combine the re-projection error and the DDS error as our final objective function for optimizing the viewpoint transformation matrices $\mathbf{R}$, $\mathbf{t}$ and the 3D point coordinate $\mathbf{P}$.

$$\mathbf{P}, \mathbf{R}, \mathbf{t} \leftarrow \arg\min_{\mathbf{P},\mathbf{R},\mathbf{t}} (E_d + \lambda E_r) \tag{29}$$

Optimization: It's a non-linear least squares problem and we use the Levenberg-Marquardt (LM) algorithm to solve for the optimal camera pose and structure. To accommodate the LM algorithm, we represent the camera pose as a 6 dimensional vector $\xi = (\omega, \mathbf{t})$, where ω is the axis/angle representation of the rotation matrix. We use subscript c to represent the point is in the camera coordinate, i.e. $\mathbf{P}_c = \mathcal{T}(\mathbf{P}, \xi)$. The Jacobi matrices of the re-projection error residuals in Eqn. 29 can be computed using the chain rule:

$$\frac{\partial \Phi}{\partial \mathbf{P}} = \frac{\partial \Phi}{\partial \mathbf{P}_c}\frac{\partial \mathcal{T}}{\partial \mathbf{P}} \quad \frac{\partial \Phi}{\partial \xi} = \frac{\partial \Phi}{\partial \mathbf{P}_c}\frac{\partial \mathcal{T}}{\partial \xi} \tag{30}$$

where $\frac{\partial \Phi}{\partial \mathbf{P}_c}$ can be easily calculated from Eqn. 22. $\frac{\partial \mathcal{T}}{\partial \mathbf{P}}$ and $\frac{\partial \mathcal{T}}{\partial \xi}$ are the derivatives of the transformation function, which are the same with the traditional SfM method.

To simplify the derivation of the Jacobi matrix of the DDS error, we set $z_k = 0$ in our implementation. Thus the residual function in Eqn. 28 becomes the direct difference between r_α and r_β. And s_β becomes the slope of the line segment connecting the projections of $\mathbf{P}$ and $\mathbf{P}'$ on the image plane, i.e.

$$s_\beta = \frac{q_x - q'_x}{q_y - q'_y} \quad q = \Phi(\mathbf{P}) \quad q' = \Phi(\mathbf{P}') \tag{31}$$

The computation of the Jacobi matrix depends only on r_β as

$$\begin{aligned} \frac{\partial r_\beta}{\partial \mathbf{P}} &= \frac{\partial r_\beta}{\partial s_\beta}\left(\frac{\partial s_\beta}{\partial q}\frac{\partial \Phi}{\partial \mathbf{P}_c}\frac{\partial \mathcal{T}}{\partial \mathbf{P}} + \frac{\partial s_\beta}{\partial q'}\frac{\partial \Phi}{\partial \mathbf{P}'_\mathbf{c}}\frac{\partial \mathcal{T}^{-1}}{\partial \mathbf{P}}\right) \\ \frac{\partial r_\beta}{\partial \xi} &= \frac{\partial r_\beta}{\partial s_\beta}\frac{\partial s_\beta}{\partial q'}\frac{\partial \Phi}{\partial \mathbf{P}'_\mathbf{c}}\frac{\partial \mathcal{T}^{-1}}{\partial \xi} \end{aligned} \tag{32}$$

$\mathcal{T}^{-1}$ is the reverse coordinate transformation according to ξ. From Eqn. 30 and 32 we can compute the Jacobi matrix J for the problem defined in Eqn. 29. For the state vector $\mathbf{x} = \{\xi, \mathbf{P}\}$ which contains all camera poses and 3D point positions, we have its LM step prediction equations as:

$$(J^T J + \lambda S)\delta \mathbf{x} = -J^T r(\mathbf{x}) \tag{33}$$

where $r(\mathbf{x})$ is the residuals for the non-linear least squares problem in Eqn. 29. S is the diagonal of J. From the feature correspondences and sequentially estimated pose of XSlit cameras, we can do triangulation and obtain the initial state vector $\mathbf{x}_0$. The state vector can be updated in each iteration as $\mathbf{x}_{i+1} = \mathbf{x}_i + \lambda \delta \mathbf{x}_i$ till we find the optimal solution for both camera poses and feature point positions.

In Fig. 6, we show illustrations of our two error metrics and the BA results. We observe that our DDS error metric effectively improves the reconstruction accuracy.

6 Experimental Validations

In this section, we perform experiments using both synthetic and real data to evaluate our XSlit SfM framework.

6.1 Camera Pose Estimation

We first evaluate our pose estimation algorithm through simulation. In our experiment, we set up the XSlit camera as $z_1 = 1$, $z_2 = 2$ and $\theta_1 = 0$, $\theta_2 = 90°$. We render images at a resolution of 800×600. The camera is moved by a rotation matrix with Euler angles $[30°, 30°, -30°]$ and translation vector $[2, 3, 0]$.

First, to support our XSlit degeneracy analysis in Sec. 3.2 and demonstrate that Li et al.'s method [25] can not handle the XSlit camera, we randomly generate 100 3D points and project those points onto the XSlit camera. We then add Gaussian noise with standard deviation $= 1$ to the projected pixels. We feed these noisy data to our algorithm and Li's method [25]. We use the angular difference between two rotation matrices as the rotation error. Given two rotation matrices $\mathbf{R_1}$ and $\mathbf{R_2}$, their angular difference is calculated as follows:

$$D_a = \cos^{-1}[(\mathrm{tr}(\mathbf{R_1} * \mathbf{R_2}^{\mathrm{T}}) - 1)/2] \tag{34}$$

The translation error is directly measured by the Euclidean distance between the two translation vectors. We simulate 1000 random tests. Fig. 7 shows the histogram of the rotation errors and translation errors of our algorithm and [25]. Li et al.'s method [25] shows huge errors in both rotation and translation suggests it can not handle the ambiguities in XSlit cameras. The reason is that [25] requires the ambiguities lie only in the determination of the rotation matrix part which is not the case for the XSlit camera as analyzed in Sec. 3.2. In contrast, our method exhibit accurate estimated poses suggesting the degeneracy is well handled.

We then evaluate the robustness of our algorithm w.r.t the noise ratio and the point-to-camera distance. The point-to-camera distance is measured by the depth sensitivity of the XSlit camera $r_z = z_2(z_2 - z_1)/z_1$, where z_1 and z_2 are the distances of the two slits. The results are shown in Fig. 8, which demonstrates that our algorithm can keep errors at a low level despite the growth of noise level.

To conduct more comprehensive comparisons, we further compare our pose estimation method with the implementations of Stewénius's six-point algorithm [47], Li et al.'s method [25], non-linear optimization method and the GE [48] method provided by the OpenGV library [49]. Our proposed method, Li et al.'s method and the non-linear optimization method directly produce the optimal estimation. In comparison, the six-point algorithm produces 64 solutions which include the optimal one produced by ours. To remove the ambiguities, one needs to integrate six-points algorithm into a RANSAC framework. Hence we first compare our method with Li et al.'s method and non-linear optimization method w.r.t noise. We then add outliers to the simulation data and test the methods under the RANSAC framework.

We add Gaussian noise to the projected pixels and use the noisy data as inputs to different methods. For each noise level, we simulate 500 tests and take the mean value as the final error. We make the error curve w.r.t noise level. During our experiment, we find the Li et al.'s method doesn't work for the XSlit camera since there is additional ambiguity in XSlit cameras. And we notice that the non-linear optimization method is very sensitive to the initial pose as shown

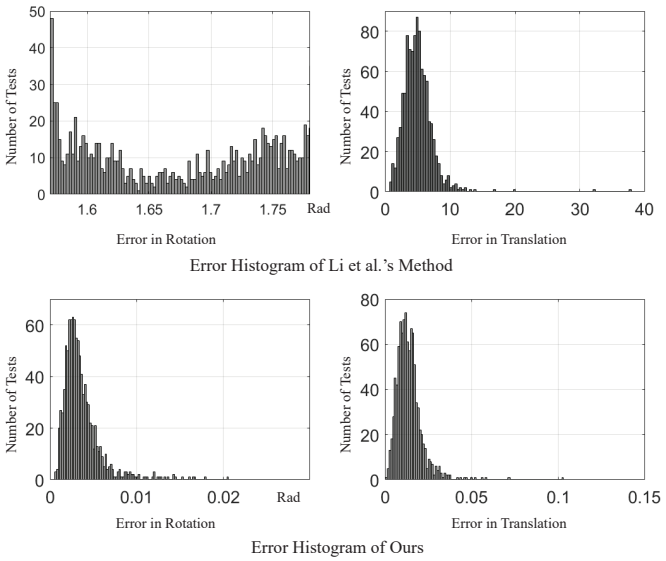


Fig. 7. Comparison with Li et al.'s method [25] under 1000 random tests with Gaussian noise. The figure shows the histogram of translation error and rotation error. The horizontal and vertical axes correspond to error and tests number respectively. The top rows shows the error of Li et al.'s method [25] and bottom row is ours.

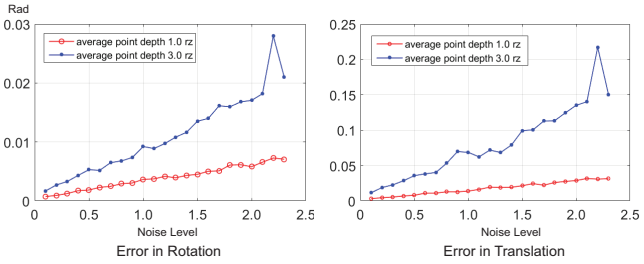


Fig. 8. The robustness of our algorithm w.r.t the noise ratio and the point-to-camera distance.

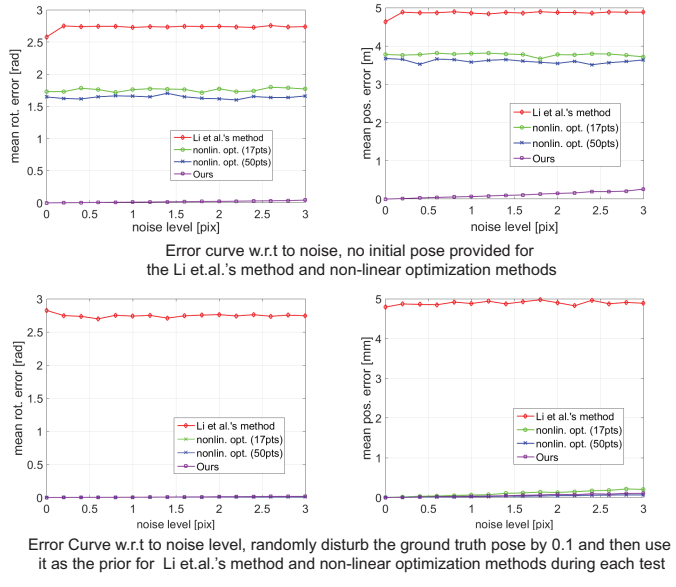


Fig. 9. Error curve of Li et al.'s method, non-linear optimization method and our method w.r.t to noise. Li et al.'s method does not work as it only handles ambiguities in axial and central cameras. Without reliable prior, non-linear optimization method can not produce accurate estimation.

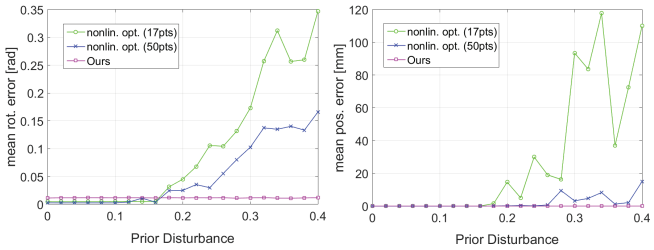


Fig. 10. Error curve w.r.t the accuracy of initial pose.

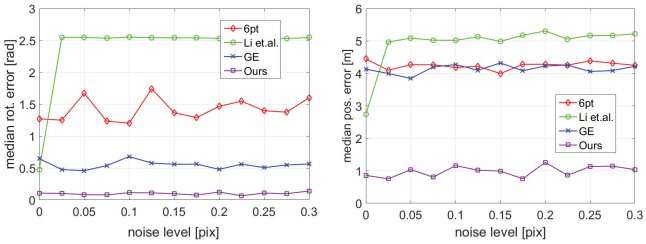


Fig. 11. Error curve of the 6-point algorithm, Li et al.'s method, GE method and our method w.r.t to noise in RANSAC framework. 0.15 fraction of original data size outliers are added.

in Fig 9. Without the initial pose (Fig 9 top), the non-linear optimization method generally produces very large errors.

To further analyze how priors can affect the performance of non-linear optimization methods, we fix the noise level to 1. Then we randomly disturb the ground truth pose according to a scale and use the disturbed pose as priors for non-linear optimization methods. We simulate 100 tests and take the mean value as the final error. We make the error curve w.r.t prior disturbance as shown in Fig. 10. As we can see, for accurate estimation, the initial pose should be fairly close to the ground truth (with only 0.15 disturbance). Not relying on the initial pose, our method can generate highly accurate estimations.

Next, we add outliers to synthetic data and compare our method with Stewénius's six-point algorithm [47], the Li et al.'s method and GE algorithm. We first add 0.15 fraction of the original data size outliers, and test the robustness of the methods w.r.t noise, as shown in Fig 11. Having found that the 6-point algorithm yields extreme large errors occasionally, we use the median error for better comparison. We observe that all other methods fail while ours succeed. We believe that the GE method fails for lack of reliable priors. In terms of the 6-point algorithm, we find that even without any noise and outliers, its implementation in OpenGV still generates large errors occasionally. We think this is because the 6-point algorithm generate 64 solutions per 6 correspondence, making the RANSAC framework more complicated, so that it is difficult to remove those ambiguities reliably. In our comparison, we use the one with the smallest error with the ground truth among all returned solutions of the 6-point algorithm (which is the most "nice" to algorithms returning multiple solutions).

We evaluate how the fraction of the outliers affects the performance. We fix the noise level to 1 and gradually add more outliers to synthetic data. For each fraction of outlier,

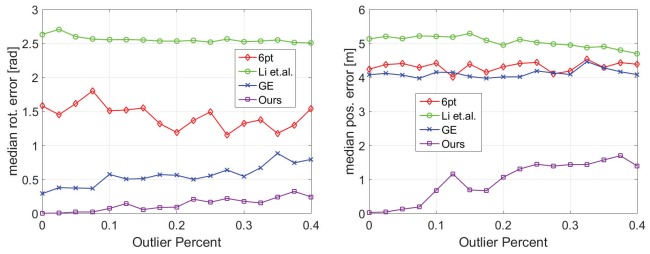


Fig. 12. Error curve of the 6-point algorithm, Li et al.'s method, GE method and our method w.r.t to outlier fraction in RANSAC framework. Gaussian noise with std 1 is added into the data.

we simulate 50 tests. The curve is shown in Fig 12.

According to the above experiments, our method is more robust and reliable than others for pose estimation of an XSlit camera.

6.2 Point Cloud Reconstruction

Synthetic Data: We use ray tracing to render synthetic XSlit images. Specifically, we implement an XSlit camera model in the open source ray tracer POV-Ray [50].

We first test on a simple ladybug scene which contains very few feature points. We use an XSlit camera with $z_1 = 1$ and $z_2 = 3$ to capture the ladybug image. The size of the ladybug is $9 \times 5 \times 5$. We place the XSlit camera about 15 units away from the ladybug and rotate around it. We render an image in every 10° with 6 images in total. The image resolution is 800×600. We follow the incremental SfM pipeline. We first perform feature matching and pose estimation on cameras 1 and 2, and triangulate rays from the matched feature points. We then do the same on cameras 2, 3, and so on until all cameras are involved.With the pose estimation and triangulation of all the five camera pairs, we merge and transform all recovered point clouds into the camera 1's coordinates. Finally, we run our bundle adjustment algorithm to refine both camera poses and point clouds. Our results (both point clouds and camera poses) are shown in Fig. 13 (top row). We superimpose the recovered point clouds onto the ground truth mesh. Although our point clouds are sparse due to the limitation of available feature points, the recovered points fit well on the ground truth and the 3D points are recovered with absolute scales.

We then test on a more complex water pot scene. With a size of $20 \times 10 \times 20$, the water pot has a high resolution texture with fine details that can provide more feature points. We use an XSlit camera with $z_1 = 3$ and $z_2 = 9$. We place the XSlit camera about 35 units away from the pot and rotate it around the object. We render 8 images in total with 15° steps. The image resolution is 800×600. Our reconstruction process is the same as that in the ladybug scene. The results are shown in Fig. 13 (bottom row). We recover denser point clouds this time and the reconstruction also fits the ground truth well.

Real Data: To capture real data, we construct a real XSlit lens using two cylindrical lenses [8]. The two cylindrical lenses have focal lengths of 25mm (closer to the sensor) and 75mm (farther away from the sensor) respectively. The principal axises of the two lenses are orthogonal and we use two 1mm wide slit apertures to form sharp images. The

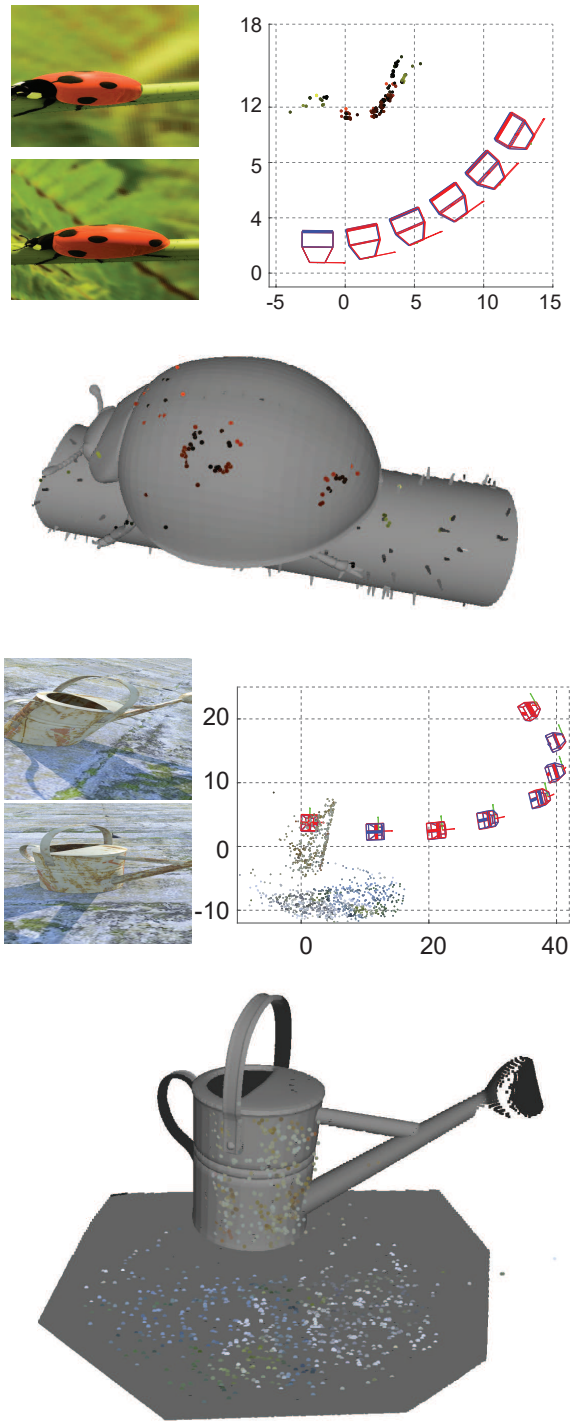


Fig. 13. Results on two synthetic data. Top row shows the ladybug scene and bottom row shows the water pot scene. For each scene, we show two sample XSlit images, recovered point clouds, estimated camera positions (blue), ground truth camera positions (red), and our point cloud superimposed on the ground truth mesh.

distance between the two lenses is adjustable, ranging from 5cm to 12cm. Our camera setup is shown in Fig. 14.

We first test our method on a simple checker scene (as shown in Fig. 15). We take a cube as our reconstruction target and put checkerboards on its surface to provide feature points. In this experiment, we manually extract checker corners and use them to estimate camera poses. We use

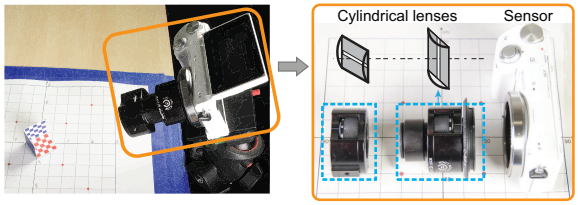


Fig. 14. Left: Our experimental setup; Right: Our real XSlit camera construction.

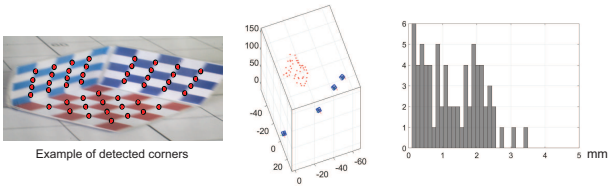


Fig. 15. Results of the checker cube scene. Left: Our recovered camera poses and 3D points; Right: Histogram of distance errors.

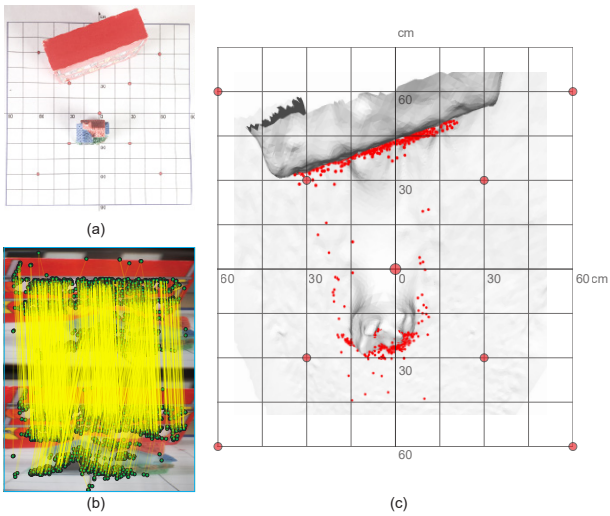


Fig. 16. Results of the toy scene. (a) Scene setup; (b) Matched feature points; (c) Recovered point cloud superimposed on the ground truth mesh.

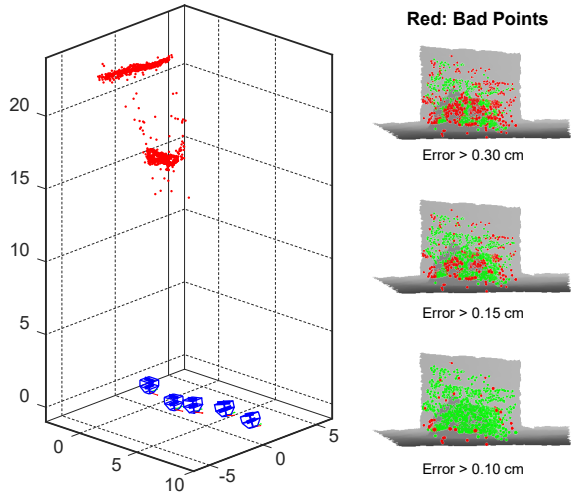


Fig. 17. Estimated camera poses for the house toy scene.

triangulation to recover the 3D points. The estimated camera poses and 3D points are shown in Fig 15(left). We use the distance errors between neighboring corners to evaluate our reconstruction since the checker corners are uniformly-spaced by 5mm. Fig 15(right) shows the histogram of the distance errors.

We then construct another scene by placing two toys on a printed coordinate grid, as shown in Fig. 16(a). We perform our feature matching algorithm on captured XSlit images. Fig. 16(b) shows the feature matching results for one XSlit image pair. We then estimate camera poses and triangulate the 3D point clouds as shown in Fig. 17. In addition, we take 12 images using a perspective camera and compute a surface mesh using a SfM software AgiSoft [51]. We treat the mesh as ground truth and resolve the scale ambiguity in the perspective case using the coordinate grid. We align the two reconstructions and superimpose our point cloud on the ground truth mesh. As shown in Fig. 16(c), the two reconstructions are consistent and our XSlit SfM estimates the 3D point clouds with absolute scale. We show the estimated camera poses in Fig. 17 and Tab. 2. To further demonstrate the accuracy of our reconstruction, we show the bad points with error defined as the nearest distance between the reconstructed point and the mesh. The percentages of the bad points with 0.10cm, 0.15cm and 0.30cm threshold respectively are shown in Tab. 3. Half of points is within 0.10cm distance with the mesh and almost all points are within the 0.30cm range. We overlay the bad points on the scene depth map and show it in Fig. 17. Consider the low quality of the real XSlit images, our method can recover the scene accurately.

View	Rotation (Eular Angles in degrees)	Traslation (cm)
0	[0°, 0°, 0°]	[0, 0, 0]
1	[0.10°, −4.94°, 0.09°]	[2.17, 0.04, -0.62]
2	[0.21°, −9.67°, 0.12°]	[3.87, 0.07, -0.36]
3	[−0.29°, −16.32°, 0.70°]	[6.58, 0.23, 0.05]
4	[−0.64°, −22.28°, 0.99°]	[8.88, 0.28, -0.141]

TABLE 2
Recovered XSlit camera poses shown in Fig. 17.

Error	Total	<0.1cm	<0.15cm	<0.3cm
Percentage of Points	1368	48.0%	71.7%	96.5%

TABLE 3
Percentage of the correct points in the reconstructed point cloud.

7 Conclusion and Discussions

We have presented a novel SfM framework based on a special type of multi-perspective camera called the XSlit. Our XSlit SfM directly eliminates the scale ambiguity observed in the perspective camera. We have shown that similar to the pinhole camera, the fundamental matrix can also be used for correlating a pair of XSlit images. We have further developed techniques to reduce the degree of freedom of the fundamental matrix for more reliable pose estimation. To use the XSlit for SfM, we have tailored techniques for feature matching, triangulation, and bundle adjustment techniques, by actively exploiting unique aspect ratio distortion properties in the XSlit images. Despite being largely theoretical,

we have validated our formulation and solutions through synthetic and real experiments.

To our knowledge, this is the first XSlit SfM framework. As discussed earlier, the XSlit is one of the most fundamental multi-perspective cameras where the pinhole, orthographic and pushbroom cameras are all its special cases. A very important future direction is to develop a unified solution for all four types of cameras where one can easily adjust individual components (e.g., feature matching, bundle adjustment etc) to accommodate special imaging properties of corresponding cameras. Our current work is still largely theoretical as high quality XSlit cameras are not yet accessible. In our implementation of the XSlit, we relay two cylindrical lenses coupled with slit-shaped aperture. Although effective, such an XSlit implementation has a relatively small baseline (i.e., the distance between the two slits) and therefore it can only acquire aspect ratio changes within a short range. Constructing a large baseline XSlit camera will be costly as it is difficult to fabricate large form cylindrical lens. A more feasible solution would be adopt a cylindrical catadioptric mirror where the reflection image can be approximated as an XSlit image, which is our immediate future work.

Finally, we plan to investigate integrating perspective and XSlit cameras into a single hybrid imaging system. For example, by constructing a hybrid XSlit-perspective camera pair, we may employ advantages for both cameras, e.g, one for reliable point cloud reconstruction and the other for solving scale ambiguity. Finally, a special type of pushbroom or XSlit images are panoramas synthesized from videos captured under translational camera movements. These images have become widely popular in VR content production and are can be easily accessed in large volumes. We hence seek to adopt SfM in-the-wide approach (e.g., using internet images [52]) and test our framework on internet panorama images.

REFERENCES

[1] S. M. Seitz and J. Kim, "The space of all stereo images," *International Journal of Computer Vision*, vol. 48, no. 1, pp. 21–38, 2002.

[2] T. Pajdla, "Stereo with oblique cameras," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 161–170, 2002.

[3] R. Pless, "Using many cameras as one," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. II–587.

[4] J. Yu and L. McMillan, "General linear cameras," in *European Conference on Computer Vision*, 2004, pp. 14–27.

[5] M. Trager, B. Sturmfels, J. Canny, M. Hebert, and J. Ponce, "General models for rational cameras and the case of two-slit projections," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1935–1943.

[6] J. Yu and L. McMillan, "Modelling reflections via multiperspective imaging," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 117–124.

[7] Y. Ji, J. Ye, and J. Yu, "Depth reconstruction from the defocus effect of an xslit camera," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2015, pp. CM4E–3.

[8] J. Ye, Y. Ji, and J. Yu, "A rotational stereo model based on xslit imaging," in *IEEE International Conference on Computer Vision*, 2013, pp. 489–496.

[9] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[12] R. Szeliski and S. B. Kang, "Shape ambiguities in structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 506–512, 1997.

[13] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, p. 133, 1981.

[14] W. Yang, H. Lin, S. Bing Kang, and J. Yu, "Resolving scale ambiguity via xslit aspect ratio analysis," in *IEEE International Conference on Computer Vision*, 2015, pp. 3424–3432.

[15] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera." in *IEEE International Conference on Computer Vision*, vol. 3, 2003, pp. 1403–1410.

[16] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover." DTIC Document, Tech. Rep., 1980.

[17] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[18] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," in *ACM Transactions on Graphics*, vol. 25, no. 3, 2006, pp. 835–846.

[19] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[20] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[21] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley, "Robust 6DoF motion estimation for non-overlapping, multi-camera systems," in *IEEE Workshop on Applications of Computer Vision*, 2008, pp. 1–8.

[22] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints," in *IEEE International Conference on Computer Vision*, 2009, pp. 1413–1419.

[23] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell *et al.*, "Detailed real-time urban 3D reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008.

[24] D. Nistér and H. Stewénius, "A minimal solution to the generalised 3-point pose problem," *Journal of Mathematical Imaging and Vision*, vol. 27, no. 1, pp. 67–79, 2007.

[25] H. Li, R. Hartley, and J.-h. Kim, "A linear approach to motion estimation using generalized camera models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[26] J.-S. Kim and T. Kanade, "Degeneracy of the linear seventeen-point algorithm for generalized essential matrix," *Journal of Mathematical Imaging and Vision*, vol. 37, no. 1, pp. 40–48, 2010.

[27] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, "Mosaicing new views: The crossed-slits projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 741–754, 2003.

[28] D. Feldman, T. Pajdla, and D. Weinshall, "On the epipolar geometry of the crossedslits projection," *IEEE International Conference on Computer Vision*, pp. 988–995, 2003.

[29] P. Sturm, "Multi-view geometry for general camera models," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 206–212.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[31] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *European Conference on Computer Vision*. Springer, 2012, pp. 214–227.

[32] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," in *IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 5097–5106.

[33] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1482–1491.

[34] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision*, 2002, pp. 128–142.

[35] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
[36] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM journal on imaging sciences*, vol. 2, no. 2, pp. 438–469, 2009.
[37] G. Schweighofer and A. Pinz, "Fast and globally convergent structure and motion estimation for general camera models." in *British Machine Vision Conference*, 2006, pp. 147–156.
[38] M. Lhuillier, "Effective and generic structure from motion using angular error," in *International Conference on Pattern Recognition*, 2006, p. 0.
[39] B. Micušık and T. Pajdla, "Autocalibration & 3D reconstruction with non-central catadioptric cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 58–65.
[40] J. Ye and J. Yu, "Ray geometry in non-pinhole cameras: a survey," *The Visual Computer*, vol. 30, no. 1, pp. 93–112, 2014.
[41] M. Levoy and P. Hanrahan, "Light field rendering," in *ACM SIGGRAPH*, 1996, pp. 31–42.
[42] J. Ponce, "What is a camera?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1526–1533.
[43] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, "A perspective on distortions," in *IEEE Conference onComputer Vision and Pattern Recognition*, 2003, p. 594.
[44] Y. Ding and J. Yu, "Multiperspective distortion correction using collineations," in *Asian Conference on Computer Vision*, 2007, pp. 95–105.
[45] http://www.robots.ox.ac.uk/ vgg/research/.
[46] V. Ovechkin and V. Indelman, "BAFS: Bundle adjustment with feature scale constraints for enhanced estimation accuracy," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 804–810, 2018.
[47] H. Stewénius, D. Nistér, M. Oskarsson, and K. strëm, "Solutions to minimal generalized relative pose problems," in *Workshop on Omnidirectional Vision*, 2005.
[48] L. Kneip and H. Li, "Efficient computation of relative pose for multi-camera systems," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 446–453.
[49] L. Kneip and P. Furgale, "OpenGV: A unified and generalized approach to real-time calibrated geometric vision," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 1–8.
[50] "Pov-Ray," http://povray.org/.
[51] "Agisoft Photoscan," http://www.agisoft.com/.
[52] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1434–1441.

Wei Yang received the BEng and MS degrees from the Huazhong University of Science and Technology and Harbin Institute of Technology respectively, and the PhD degree from the University of Delaware (UDel) in Dec 2017. He joined the DGene. Co (Prev. Plex-VR) as a research scientist in Mar 2018. His research interests include computer vision and computer graphics, with special focus in computational photography and 3D reconstruction.

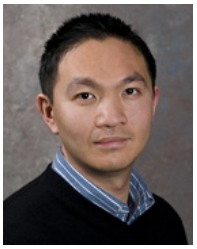
Yingliang Zhang received the B.E. degree of communication engineering from Ningbo University, China, in 2014. He is pursuing the Ph.D degree of computer science from ShanghaiTech University, China. His research interests include image-based 3D reconstruction, light field rendering, and light field reconstruction.

Jinwei Ye received her PhD degree from University of Delaware in 2014. She received her B.Eng. in Electrical Engineering from Huazhong University of Science and Technology. After her PH.D., she worked with the US Army Research Laboratory (ARL) and Canon U.S.A., Inc. Currently, she is an assistant professor at the Division of Computer Science and Engineering at Louisiana State University. Her research interests lie in the fields of computational photography and computer vision.

Yu Ji received his PhD degree in computer science from the University of Delaware in 2014. He received his M.Sc. in digital media from Nanyang Technological University in 2011 and Bachelor degree in Electrical Engineering from Huazhong University of Science and Technology in 2009. He is now a principal scientist at Plex VR. His research interests include computational photography, computer vision, and computer graphics.

Zhong Li received his MSc degree in Computer Science from the University of Missouri in 2015. He is now working toward the Ph.D. degree in the Department of Computer and Information Sciences, University of Delaware. His research interests include computational photography, computer graphics, and computer vision. He is a student member of the IEEE.

Mingyuan Zhou received his M.E. degree in Computer Engineering from Stevens Institute of Technology in 2014, and B.E. degree in Intelligence Science and Technology from Beijing Information Science and Technology University, China, in 2011. He is now a Ph.D. student at the Department of Computer and Information Sciences, University of Delaware. His research interests include image processing, computer graphics and computer vision.



Jingyi Yu is the executive dean in the School of Information Science and Technology at ShanghaiTech University. He received B.S. from Caltech in 2000 and Ph.D. from MIT in 2005. He is also affiliated with the University of Delaware. His research interests span a range of topics in computer vision and computer graphics, especially on computational photography and non-conventional optics and camera designs. He is a recipient of the NSF CAREER Award, the AFOSR YIP Award, and the Outstanding Junior Faculty Award at the University of Delaware. He has served as general chair, program chair, and area chair of many international conferences such as CVPR, ICCV, ECCV, ICCP and NIPS. He is currently an Associate Editor of IEEE TPAMI, IEEE TIP and Elsevier CVIU, and will be program chair of ICPR 2020 and IEEE CVPR 2021.