Prepare for Ludicrous Speed: Marker-based Instantaneous Binocular Rolling Shutter Localization

Juan Carlos Dibene*
Stevens Institute of Technology

Yazmin Maldonado†
Instituto Tecnológico de Tijuana

Leonardo Trujillo‡
Instituto Tecnológico de Tijuana

Enrique Dunn§
Stevens Institute of Technology

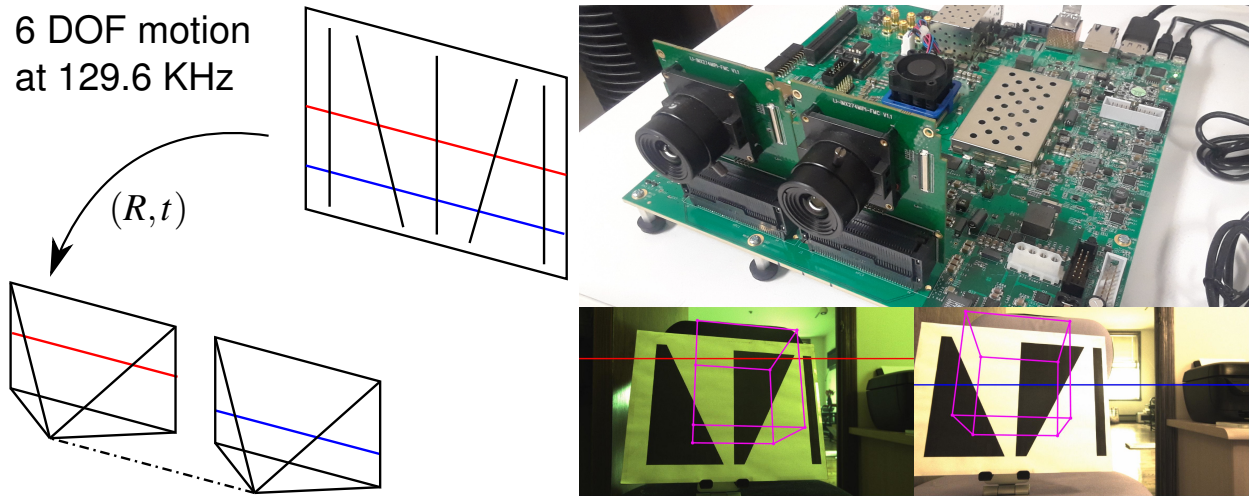

Fig. 1. Our proposed geometric framework. We estimate the absolute pose of a binocular camera system using only a pair of concurrent rolling shutter scanlines (red and blue) observing a special planar pattern, achieving drift-free, real-time, low-latency (1.5 μs from pixel readout to pose estimation), high frequency (129.6 KHz) operation. We illustrate our method by using a pair of scanlines from real imagery to estimate the absolute pose and render a 3D cube (magenta) on the pattern.

Abstract— We propose a marker-based geometric framework for the high-frequency absolute 3D pose estimation of a binocular camera system by using the data captured during the exposure of a single rolling shutter scanline. In contrast to existing approaches enforcing temporal or motion models among scanlines (e.g. linear motion, constant velocity or small motion assumptions), we strive to determine the pose from instantaneous binocular capture (i.e. without using data from previous scanlines) and achieve drift-free pose estimation. We leverage the projective invariants of a novel rigid planar pattern, to both define a geometric reference as well as to determine 2D-3D correspondences from raw edge detection measurements from individual scanlines. Moreover, to tackle the ensuing multi-view estimation problem, achieve real-time operation, and minimize latency, we develop a pair of custom solvers leveraging our geometric setup. To mitigate sensitivity to noise, we propose a geometrically consistent measurement refinement mechanism. We verify the quality of our solvers by comparing with state of the art general solvers for absolute pose estimation of generalized cameras. Finally, we demonstrate the effectiveness of our proposed approach with an FPGA-based implementation which achieves a localization throughput of 129.6 KHz with a 1.5 μs latency.

Index Terms—Absolute pose estimation, Cross ratio, Rolling shutter, FPGA

1 Introduction

Camera pose estimation aims to localize an observer w.r.t. a known geometric reference and has a range of applications from autonomous navigation to virtual/augmented reality. Standard practice for accuracy-driven scenarios, is for fine-grain geometric analysis to rely on associating local image primitives with known environmental landmarks in order to yield pose estimates compliant to a given image formation model. Motivated by the ubiquity of digital cameras deploying "rolling shutter" (**RS**) hardware, pose estimation modules have been extended from the pin-hole camera model (applicable for global shutter), to incorporate the dynamic 1D capture characteristics of these sensors. Moreover, explicit (local) motion models governing viewpoint changes among distinct sequential scanlines, have enabled tasks such as imaging aberration correction and scanline-level egomotion estimation. However, absolute pose estimation from single RS scanline inputs has been hitherto ignored in the literature. This omission may be attributed to the limited geometric context available for instantaneous RS capture and stringent latency requirements involved in real-time system operation.

We propose a solution to the problem of determining the absolute

*e-mail: jdibenes@stevens.edu
†e-mail: yaz.maldonado@tectijuana.edu.mx
‡e-mail: leonardo.trujillo@tectijuana.edu.mx
§e-mail: edunn@stevens.edu

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Table 1. Comparison of our proposed geometric framework with existing works for estimating the pose and motion of RS cameras.

Method	Estimated Pose	Known Spatial Reference	Observation Scope	Motion Prior	Motion Representation	Capture Setup	Solver Type	Solution Cardinality
[2]	Absolute	SfM Sparse 3D	Whole Frame	Linear Motion	Double Linearized SE(3)	Monocular	Minimal (Polynomial)	20
[3]	Absolute	SfM Sparse 3D	Whole Frame	Linear Motion	Linearized SE(3)	Monocular	Minimal (Polynomial)	8
[4]	Relative	None	Single Scanlines	Linear Motion	Linearized SE(3)	Camera cluster	Linear Least Squares	1
[5]	Relative	None	Single Scanlines	Linear Motion	Linearized SE(3)	Camera cluster	Linear Least Squares	1
[19]	Absolute	Shape-from-Template	Whole Frame	Deformable 3D	Linearized SE(3)	Monocular	Levenberg-Marquardt	1
[23]	Absolute	SfM Sparse 3D	Whole Frame	Linear Motion	Linearized SE(3)	Monocular	Semi Definite Programming	1
[34]	Absolute	SfM Sparse 3D	Whole Frame	Linear Motion	Zero Angular Motion	Monocular	Minimal (Polynomial)	8
[20]	Absolute	3D marker	Single Scanlines	None	Full SE(3)	Monocular	Polynomial	1
[21]	Absolute	Planar Line Pattern	Frame + Scanline	None	Full SE(3)	Binocular	Polynomial	1
Our 6-pt	Absolute	Planar Line Pattern	Single Scanlines	None	Full SE(3)	Binocular	Minimal (Polynomial)	8
Our 10-pt	Absolute	Planar Line Pattern	Single Scanlines	None	Full SE(3)	Binocular	Geometric (Closed Form)	1

3D pose of a binocular rolling shutter camera system using only the data captured during the exposure of a single scanline per camera. In contrast to existing works, we do not use whole image frames to perform pose estimation. The motivation for using single scanlines is that pose estimation is no longer bound to the frame rate of the camera, and instead, it depends on the scanline frequency, which is orders of magnitude higher. Also, by only using scanlines captured at the same point in time, our geometric framework obviates the need for intra-frame camera motion models (such as linear motion, constant velocity, or small motion assumptions) and determines instantaneous absolute 6-DOF pose, as if it were a binocular system of 1D cameras. In doing so, we address the following open questions within the context of RS-based multi-view geometry: What constitutes a persistent, reliable, sufficient and minimal geometric context for localization based on 1D observations? What are the trade-offs between minimal and non-minimal solvers in this context?

Our answers to both these questions are reached through 1) the development of a novel planar pattern and a geometric model leveraging planar projective invariants found among sets of intersecting lines observed by binocular capture, and 2) the algebraic derivation of an optimal residual model w.r.t. a set Euclidean geometric constraints pursuant to a generalized camera instance. Accordingly, we summarize our contributions as:

- ***Depth from cross ratio***. A pattern design and geometrical analysis for determining in simple closed form the depths of the observed points on the pattern from monocular capture, enabling high speed pose estimation.
- ***Custom Non-Perspective N-Point solvers***: A minimal solver estimating the full 6 degrees of freedom (DOF) absolute camera pose from six observations and a computationally streamlined solver from ten (redundant) observations.
- ***Optimal residual model***: To mitigate noise sensitivity, we derive an optimal multi-view observation refinement using redundant measurements to determine a set of coupled input values that satisfy our system's imaging geometry.

1.1 Related Work

Applications of the Cross Ratio. The cross ratio of four collinear points is invariant under projection to an image, enabling geometric object recognition without estimating the camera pose [24]. For example, Han [10] used the cross ratio to estimate the speed of cars w.r.t. a camera without knowing the location of the car on the road. Ramalingam *et al.* [31] use cross ratios for wide-baseline matching and 3D reconstruction. Yu *et al.* [45] use the cross ratio for designing fiducial markers for a robot positioning system. Dhall *et al.* [8] use the cross ratio for detecting traffic cones for autonomous driving. Zhang *et al.* [44] use cross ratios across frames for action recognition. Nakai *et al.* [27] use cross ratios to perform camera-based document image retrieval robust to perspective distortion. The cross ratio is also used in the context of line-scan (1D) cameras. Given the challenges of establishing point correspondences from 1D capture due to the limited geometric context available, there are line-scan camera calibration methods that rely on special markers to obtain point correspondences from the cross ratios of the 1D observations of the marker, which are then used to estimate the intrinsics and extrinsics (pose) of the camera [12]. Li *et al.* [20] use a 3D marker consisting of two orthogonal planes containing 20 lines each. From the 1D observations of the marker lines, point correspondences are established and an homogeneous linear system is constructed. Then, this linear system is solved using SVD to obtain the camera parameters, which are then refined using non-linear optimization. Li *et al.* [21] use a 2D marker, shown at a least two different orientations, and an additional calibrated 2D camera to perform the calibration of a line-scan camera. The pose between the two cameras is unknown but fixed. Su *et al.* [37] use the same 3D marker as [20] for the calibration of a hyperspectral line-scan camera. Usamentiaga *et al.* [38] present a robust linear method for line-scan camera calibration assuming the point correspondences have been obtained beforehand.

The Perspective-Three-Point (P3P) Problem. P3P is the task of finding the orientation and translation (pose) of a calibrated camera from three 2D-3D point correspondences. Initially solved by Grunert in 1841, a review of the major direct solutions up to 1991 is given by Haralick *et al.* [11]. Traditionally, P3P is posed as a system of three multivariate polynomial equations derived from the law of cosines. Then, this polynomial system is reduced to a quartic univariate polynomial yielding up to four different poses. Gao *et al.* [42] provide a complete solution classification describing the criteria under which the P3P problem has one, two, three or four solutions. P3P was first formulated without the law of cosines by Kneip *et al.* [17], where camera pose is determined directly without computing distances among observed features to yield superior numerical stability and robustness. Ke and Roumeliotis [15] present an algebraic solution that avoids computing unnecessary intermediate results for faster run times and increased accuracy. Persson and Nordberg [30] introduce a solver that never computes geometrically invalid or duplicate solutions. Wang *et al.* [40] give an intuitive relationship between the solutions of the P3P problem and the roots of associated the quartic equation.

Generalized P3P. While P3P assumes a single projection center for all 3D landmark viewing rays, the case where the rays do not share a common projection center is the generalized P3P problem, also called non-central or non-perspective three point (NP3P). This corresponds to the case of a multi-camera system. Nistér [29] describes a minimal solution for the NP3P problem, which results in an octic polynomial arising from the intersection of a ruled quartic surface and a circle. The odd monomials of this octic vanish in the case of central P3P. Merzban *et al.* [26] formulate the NP3P problem as a system of three multivariate polynomial equations where the unknowns are the distances to the 3D landmarks. Using Sylvester matrix resultants, the real roots of a univariate polynomial of degree eight yield the pose parameters. Kneip *et al.* [16] present Unified PnP which works for both the central and non-central cases and handles more that three point correspondences.

Pose Estimation for RS Cameras. The pose estimation methods mentioned so far assume a camera model that is valid for global shutter cameras but not for RS cameras. However, due to the ubiquity of RS cameras, there has been a great interest in the development of models

and algorithms suitable for these type of cameras [7, 14, 18, 25, 32, 33, 35, 36, 39, 46, 47]. A minimal solution for the RS absolute pose problem from six points is given by Albl *et al.* [2]. It uses a double linearized camera model and the standard P3P algorithm. Saurer *et al.* [34] present another minimal solution for RS pose estimation. They introduce an additional linear velocity in the camera projection matrix to model the motion of the RS camera and solve for the pose and velocity using five 2D-3D point correspondences, obtaining up to eight solutions. Albl *et al.* [3] solve for the RS absolute pose with known vertical direction using five points. Magerand *et al.* [23] describe a polynomial projection model for RS cameras and a constrained global optimization of its parameters to compute the uniform motion of a known object. Lao *et al.* [19] propose a method for absolute pose estimation for RS cameras, considering RS distortions due to camera ego-motion as virtual deformations of a known template captured by a global shutter camera. The camera pose is computed by registering the deformed scene on the original template. Albl *et al.* [1] introduce a special two RS camera configuration and an associated motion model to correct RS distortions and generate synthetic global shutter imagery. Wang *et al.* [41] present a robust RS stereo depth map estimation pipeline, which adapts to the changes in baseline due to camera motion.

High Speed Tracking. Bapat *et al.* [4] developed a 6-DOF markerless head pose tracker using a cluster of ten RS cameras. Their tracker considers the RS camera as a high frequency 1D sensor. This allows them to track the 6-DOF head pose from individual image rows instead of whole frames. Bapat *et al.* [5] leverage RS and radial distortion to achieve superior performance w.r.t. high frequency camera pose estimation. The constraints introduced by radial distortion allows them to reduce the amount of observational redundancy and work with as little as four cameras. Narita *et al.* [28] present a high speed tracker for non-rigid surfaces using a fiducial marker comprised of dot clusters painted onto the surface which can be detected even when strongly deformed. Blate *et al.* [6] implemented a 50 KHz 28 μs latency head tracking instrument. The instrument consists of a helmet equipped with 4 LED emitters and a pair of duo-lateral photodiode sensors facing the target helmet. Guzel *et al.* [9] present a fast incremental pose estimator for RS cameras.

1.2 Novelty and Relevance of our Framework

Comparison with Existing Methods. Table 1 presents a summary of the differences between our proposed geometric framework and existing methods for estimating the pose and the motion of RS cameras. While existing RS absolute pose analyze a set of non-concurrent pixel rows to estimate a motion (or registration) model from which to determine the pose of a specific pixel row, our approach computes absolute pose from single scanlines. Moreover, while [2, 3, 23, 34] rely on whole-frame 2D feature extraction; [19] relies on modeling 3D/2D pattern deformation through image registration (i.e. *shape-from-template*), requiring content analysis over an image region. Compared to [2, 3, 34], we do not impose a temporal model on the camera nor assume linear or small motion. Therefore, *our model is unaffected (geometrically) by motion intensity and non-linearity*. However, in practice, motion blur due to fast motion reduces the performance of our method. Similarly to line-scan camera calibration methods, we use a marker to obtain point correspondences from 1D observations by leveraging the cross ratio. However, from our pattern design and geometric analysis we determine in simple closed forms the depths of the observed points on the pattern (given that our cameras are calibrated) which allows us to develop a novel simple pose solver from ten observations. In contrast to [12, 20, 21, 37, 38], our streamlined ten point method does not require solving systems of equations (linear least squares in [12] and SVD in [20, 21, 37, 38]) to determine the pose of the camera. This simplicity, together with single scanline readouts, ultimately enables high-frequency low latency real-time operation, as demonstrated by our FPGA implementation. Moreover, we develop a novel minimal pose solver from six observations with better accuracy but which is mainly of theoretical interest due to its higher computational burden. While [4] is markerless, it uses a ten-camera cluster (four or more cameras in [5]) for relative motion estimation, which is subject to drift and applicable only to short time-horizon applications. Our method requires only two cameras plus the planar line pattern and does not rely on data from previous scanlines or past motion estimations. Therefore, *our method does not drift and its performance does not degrade over time*.

Motivation. Virtual 3D content renderings for online AR/VR systems must compensate for observer motions based on the estimates of a viewer pose tracker. To eradicate visualization artifacts, tracking throughput should outpace rendering throughput, which should, in turn, outpace user perception. When considering high frequency displays, such as the 16,000 FPS system of Lincoln *et al.* [22], the performance bottleneck is acquiring tracking data at least an order of magnitude faster than today's technology. Besides high speed, low latency is necessary to avoid virtual imagery lagging behind the user's motion. Current approaches for single-scanline RS pose estimation [4, 5] focus on the algorithmic and geometric aspects of the estimation process, but instead of operating online, they analyze recorded (or buffered) data. Per the discussion in [6], the achievement of a real-time (i.e. online) implementation for such frameworks is still a non-trivial and open engineering challenge. Our pipelined FPGA implementation integrates a comprehensive solution to the camera localization task, as it performs feature extraction (1D edge detection), automatic pattern detection, and absolute pose estimation, all within the time needed to stream a single row of pixels from the camera. Importantly, we provide the first reported system able to operate online from live capture imagery. Indeed, our geometric framework enables the operation of systems such as [22], as it provides pose estimates with a frequency upwards of 100 KHz. Our geometric framework targets operation in a controlled or structured environment where our proposed planar line pattern is available. In particular, we target indoor AR/VR applications (needing high frequency pose estimation and reduced latency) using a head-mounted display (HMD) with a pair of high-resolution large-FOV cameras and the pattern projected to a persistently observable large flat surface (e.g. the ceiling, which is usually not cluttered).

2 Geometric Formulation

Landmark Estimation from 1D Scanlines. As common practice in multi-view geometric analysis, we strive to associate local features to a geometric reference. Hence, we use a known 2D line pattern to facilitate persistent landmark capture across independent pixel rows. In this way, our image features are the edge detections along each scanline. Accordingly, we aim to associate such landmarks with a meaningful geometric context. We leverage projective invariances to localize each observed 1D scanline w.r.t. the known 2D line pattern with the following geometric rationale.

Lemma 1 (Planar Incidence Localization)**.** *For an arbitrary line ℓ_o intersecting four known lines $\{\ell_A, \ell_B, \ell_C, \ell_D\}$, of which only ℓ_A is not part of a common pencil, the intersection of ℓ_o and ℓ_A is uniquely determined by the cross ratio of incidence displacements along ℓ_o.*

Proof. The cross ratio of incidence locations $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ of a line ℓ_o with four distinct lines $\{\ell'_A, \ell_B, \ell_C, \ell_D\}$ of a pencil $\mathscr{P}_\mathbf{x}$, is a well known projective invariant for any ℓ_o not passing through the pencil's generator $\mathbf{x}$ (see Fig. 2**a**). Replacing ℓ'_A by another line $\ell_A \notin \mathscr{P}_\mathbf{x}$ passing through $\mathbf{a}$ yields the same cross ratio since the incidence locations remain unchanged while eliminating invariance w.r.t. ℓ_o as the incidence points no longer define a projectivity. Hence, for a known set $\{\ell_A, \ell_B, \ell_C, \ell_D\}$ the cross ratio value identifies an additional implied line ℓ'_A, which may be intersected with an arbitrary known line ℓ_A to determine the location $\mathbf{a} = \ell'_A \times \ell_A$ (see Fig. 2**b**). □

This result provides a viewpoint-invariant closed-form expression of a point in the projective plane from a set of 1D measurements. Importantly, for a suitable parametrization, the displacement of the intersection point along the known line is a linear function of the cross ratio. The following corollary exploits this insight.

Corollary 1 (Line Localization)**.** *The coordinates of the line ℓ_o intersecting a set of five known lines $\{\ell_A, \ell_B, \ell_C, \ell_D, \ell_E\}$, of which only ℓ_A and ℓ_E are not part of a common pencil, are determined by two cross ratios of the intersection displacements along ℓ_o.*

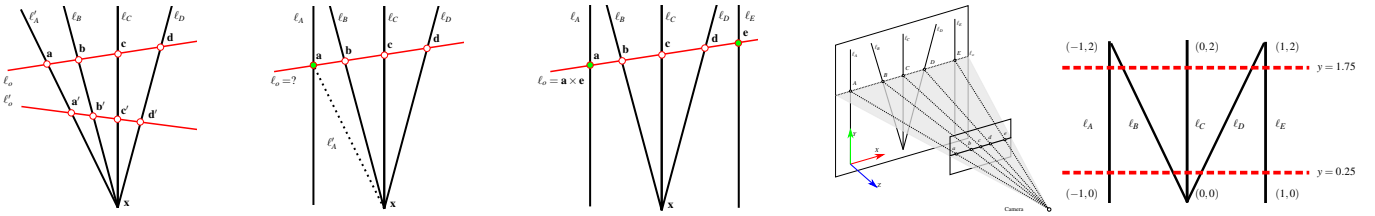

Fig. 2. **a**) The cross ratio is a projective invariant so the cross ratios $(\mathbf{a},\mathbf{b};\mathbf{c},\mathbf{d})$ and $(\mathbf{a}',\mathbf{b}';\mathbf{c}',\mathbf{d}')$ are equal. **b**) The position of $\mathbf{a}$ on ℓ_A is uniquely determined by $(\mathbf{a},\mathbf{b};\mathbf{c},\mathbf{d})$. **c**) The position of $\mathbf{e}$ on ℓ_E is uniquely determined by $(\mathbf{e},\mathbf{d};\mathbf{c},\mathbf{b})$ and the line ℓ_o observed on the pattern is determined by $\mathbf{a}$ and $\mathbf{e}$. **d**) The points A, B, C, D, and E on the pattern can be computed solely from the points a, b, c, d, and e observed in a single image row. **e**) The geometry of our pattern on the xy plane. For cross ratio numerical stability, we avoid observing line segments close to the line intersection vertex points and register against a printed template corresponding to the vertical range $y \in [0.25, 1.75]$.

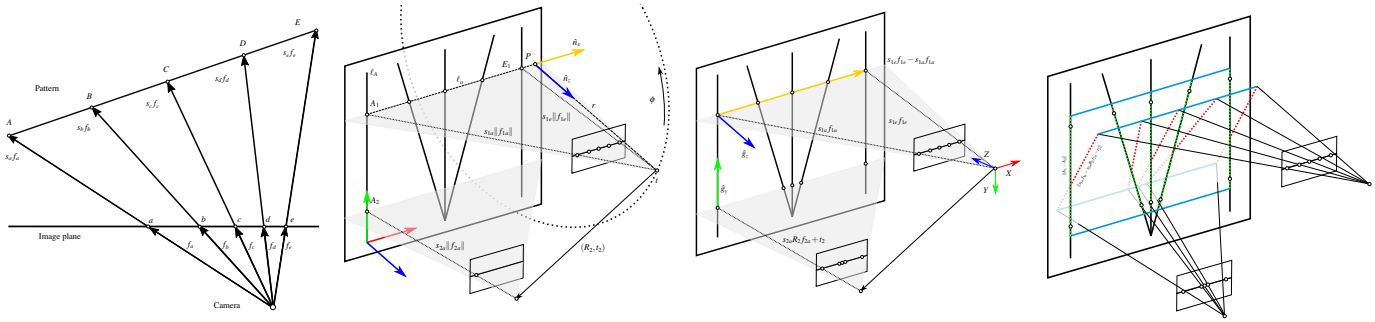

Fig. 3. **a**) Bird's eye view of a camera observing points A, B, C, D, and E on the pattern. The unknowns are the scaling factors s_a, s_b, s_c, s_d, and s_e. Using this triangle, we solve for the pattern-to-camera distances. **b**) Six-point pose estimation. The five points observed by camera 1 constrain its position on a 3D circle centered on ℓ_o. For points on the circle corresponding to a valid camera rig translation t, the ray f_{2a} of camera 2 intersects the line ℓ_A. From the (up to eight) valid values of t, we solve directly for R. **c**) Ten-point pose estimation. Using our estimated camera-to-pattern distances, we can build the world frame on the camera space and solve directly for (R,t) uniquely. **d**) The effect of noise. The small circles denote the real points before noise. The 3D points on the pattern determined by the cross ratio and their corresponding 3D points obtained by stretching the rays from the cameras are not related by a rigid transformation, so the distance between the points in camera space (red) is not the same as the distance between their corresponding points on the pattern (green).

Proof. The proof follows directly from Lemma 1, as it suffices to compose a pair of overlapping line sets, $\mathbf{L}^a = \{\ell_A, \ell_B, \ell_C, \ell_D\}$ and $\mathbf{L}^e = \{\ell_B, \ell_C, \ell_D, \ell_E\}$, to determine the displacement of the points of intersection along ℓ_A and ℓ_E, with ℓ_o as their join (see Fig. 2**c**). □

Establishing 2D-3D Correspondences. The above results effectively upgrades a set of five 1D observations in $\mathbb{P}^1$ into a line element in $\mathbb{P}^2$, yielding that for a set of five known lines we can determine, uniquely and in closed form, the intersection points $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e} \in \mathbb{P}^2$ along each line in our pattern. Moreover, while $\mathbf{a}$ and $\mathbf{e}$ are determined explicitly from the corresponding cross ratio, the intersection coordinates of the remaining points are trivially computed from the cross product of each pair of line coordinates (e.g. $\mathbf{b} = \ell_o \times \ell_B$). We further impose Euclidean structure to our known line set and our 1D observations, in order to ascertain 2D-3D correspondences. More specifically, 1) we define our known 2D line pattern to establish an absolute Euclidean reference by residing on a specific 3D plane, where the Euclidean 3D locations of our line intersection are denoted as $\{A,B,C,D,E\}$; while 2) we map the set of 1D observations from a single scanline into image coordinates through the known intrinsics of our calibrated camera (see Fig. 2**d**).

Pattern Geometry. Our pattern allows to obtain five 2D-3D point correspondences from a single image row. The observed points A, B, C, D and E on the pattern are a function of the cross ratios $(a,b;c,d)$ and $(e,d;c,b)$. For our pattern (see Fig. 2**e**), the observed 3D points are

$$A = \begin{bmatrix} -1 & 4r_1 - 2 & 0 \end{bmatrix}^\top \tag{1}$$

$$B = \begin{bmatrix} 1 - r_1 - r_5 & 2r_1 + 2r_5 - 2 & 0 \end{bmatrix}^\top / (r_5 - r_1 + 1) \tag{2}$$

$$C = \begin{bmatrix} 0 & 2r_1 + 2r_5 - 2 & 0 \end{bmatrix}^\top \tag{3}$$

$$D = \begin{bmatrix} r_1 + r_5 - 1 & 2r_1 + 2r_5 - 2 & 0 \end{bmatrix}^\top / (r_1 - r_5 + 1) \tag{4}$$

$$E = \begin{bmatrix} 1 & 4r_5 - 2 & 0 \end{bmatrix}^\top \tag{5}$$

$$r_1 = (a,b;c,d) = ((c-b)(d-a)) / ((c-a)(d-b)) \tag{6}$$

$$r_5 = (e,d;c,b) = ((c-d)(b-e)) / ((c-e)(b-d)) \tag{7}$$

Binocular Rolling Shutter. So far we have described how to ascertain the in-plane coordinates of a set of five collinear points observed in a single scanline. This (degenerate) datum constraints the center of projection of the observing (pin-hole) camera to lie on a 3D circle centered on and orthogonal to ℓ_0 (see Fig. 3**b**). Notably, we leverage a set of five 1D observations to determine 5 DOF for camera pose, where the remaining DOF determines its location along the circumference. To resolve this ambiguity, we require an additional in-plane observation in general position (i.e. not on ℓ_0). We attain such information from a concurrent scanline observation from a second calibrated camera with known relative pose which captures a distinct region of the line pattern. We trivially avoid both cameras imaging the same 2D line in the plane, by controlling image capture configuration (e.g. adjusting the relative orientation among the camera pair or controlling the offset between the scanline capture of the cameras).

Solving for Generalized Camera Pose. Our capture scenario equates to a generalized camera whose pose estimation, using the available 3D points, constitutes a NPNP instance. A unique characteristic of our formulation is that the 3D points are intermediate representations derived from input image observations. While this enables a solution

defined exclusively in terms of edge detection locations along an input scanline, the sensitivity to measurement errors is inherently amplified. Moreover, we target implementing an absolute pose estimator from single scanlines on an embedded system (FPGA) for real-time high frequency operation. Existing NP3P algorithms, such as [29] and [26], require solving an octic polynomial, an expensive iterative operation not amenable to hardware implementation, while [16] requires solving a least squares problem necessitating expensive linear algebra submodules. Hence, to achieve reduced computational burden (enabling real-time low-latency high frequency operation with low hardware resources usage) and improved numerical stability, we develop a pair of custom geometric solvers leveraging our specific pattern geometry.

Leveraging Line Pattern Geometry. We begin by imposing specific structure onto our line pattern and leveraging the collinearity of our observations. The goal is to leverage Euclidean constraints emerging from the configuration of the line set to determine distance values from a camera to the observed points on the pattern uniquely and in closed form. Using these distances, we can find the pose of the camera system by enforcing invariance between in-plane distances (i.e. world reference system) among our landmarks and the distances of their estimates in the cameras 3D reference frame. We define the pattern lines ℓ_A, ℓ_C, and ℓ_E to be parallel to the world Y axis and their x coordinate in increasing order. We assume chirality constraints are satisfied, in the sense that all points are in front of the cameras. This pattern manipulation provides an outstanding result: *the distances along each of the viewing rays are known in simple closed-forms directly from the image observations*.

Pattern-to-Camera Distances. The distance from the center of projection of the camera to the points A, B, C, D, and E on the pattern can be determined by solving the triangle shown in Fig. 3**a**. We assume that the intrinsic matrix K of the camera is known. From the law of cosines and by similar triangles we have that

$$\|A-E\|^2 = s_a^2\|f_a\|^2 + s_e^2\|f_e\|^2 - 2(s_a f_a \cdot s_e f_e) \tag{8}$$

$$(s_a(a-c))/(s_e(c-e)) = \|A-C\|/\|C-E\| \tag{9}$$

Combining (8) and (9), and by similar triangles we find that

$$s_a = \lambda_a\|A-E\|/\|\lambda_a f_a - \lambda_e f_e\| \tag{10}$$

$$s_e = \lambda_e\|A-E\|/\|\lambda_a f_a - \lambda_e f_e\| \tag{11}$$

$$s_c = (\|C-E\|s_a + \|A-C\|s_e)/\|A-E\| \tag{12}$$

$$s_b = s_c\|B-D\|(c-d)/((b-d)\|C-D\|) \tag{13}$$

$$s_d = s_c\|B-D\|(b-c)/((b-d)\|B-C\|) \tag{14}$$

$$\lambda_a = \|A-C\||c-e| \tag{15}$$

$$\lambda_e = \|C-E\||a-c| \tag{16}$$

We determine 3D feature coordinates in the camera coordinate system simply by scaling the vector $f_j = K^{-1}[j, row, 1]^\top$ $(j \in \{a,b,c,d,e\})$ associated with each observation (the actual distances are $s_j\|f_j\|$). It is important to note that these distances are dependent on the five observations (i.e. they are not attainable through incidence localization of a single point).

2.1 A Minimal Six-point Solver

The six DOF absolute pose of the camera system can be determined using six points (one for each DOF): five from one camera and one from the other. The intuition is to leverage the Euclidean constraints attained from five collinear 2D-3D correspondences from the same scanline and use the remaining observation as an additional ray-to-line incidence constraint. Fig. 3**b** illustrates the geometry of the problem. We assume w.l.o.g. that five points come from camera 1 and one point from camera 2, as similar geometric arguments can be made in the alternative scenario. The five points constrain the position of camera 1 on a 3D circle that is centered on and orthogonal to ℓ_o. The center P of the circle and its radius r are

$$P = A_1 + \alpha(E_1 - A_1) \tag{17}$$

$$r = \sqrt{s_{1a}^2\|f_{1a}\| - \alpha^2\|E_1 - A_1\|^2} \tag{18}$$

$$\alpha = (\lambda_{1a}^2\|f_{1a}\|^2 - \lambda_{1e}^2\|f_{1e}\|^2)/(2\|\lambda_{1a}f_{1a} - \lambda_{1e}f_{1e}\|^2) + 1/2 \tag{19}$$

The position t of camera 1 is parametrized using stereographic projection with the parameter $\phi \in (-1,1)$. Values $\phi \notin (-1,1)$ refer to positions behind the pattern which are not considered.

$$t = P + r\begin{bmatrix}\hat{n}_x & \hat{n}_z \times \hat{n}_x & \hat{n}_z\end{bmatrix}\begin{bmatrix}0 & \frac{2\phi}{\phi^2+1} & \frac{1-\phi^2}{\phi^2+1}\end{bmatrix}^\top \tag{20}$$

$$\hat{n}_x = \hat{\mathbf{u}}(E_1 - A_1) \tag{21}$$

$$\hat{n}_z = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top \tag{22}$$

$$\hat{\mathbf{u}}(v) = v/\|v\| \tag{23}$$

The orientation R of camera 1 is given by the relationship between the rays $\hat{f}_1$ in camera space and the rays $\hat{f}_1'$ in world space

$$R = \begin{bmatrix}\hat{f}_{1a}' & \hat{g}_y' & \hat{f}_{1a}' \times \hat{g}_y'\end{bmatrix}\begin{bmatrix}\hat{f}_{1a} & \hat{g}_y & \hat{f}_{1a} \times \hat{g}_y\end{bmatrix}^\top \tag{24}$$

$$\hat{f}_{1a}' = \hat{\mathbf{u}}(A_1 - t) \tag{25}$$

$$\hat{g}_y' = \hat{\mathbf{u}}((A_1 - t) \times (E_1 - t)) \tag{26}$$

$$\hat{f}_{1a} = \hat{\mathbf{u}}(f_{1a}) \tag{27}$$

$$\hat{g}_y = \hat{\mathbf{u}}(f_{1a} \times f_{1e}) \tag{28}$$

As camera 1 moves about the circle, so does the ray f_2 corresponding to the single point from camera 2, according to the relative pose (R_2, t_2) of camera 2 w.r.t. camera 1. Per Fig. 3**b**, such ray corresponds to a point on ℓ_A, described as

$$A_2 = R(s_{2a}R_2 f_{2a} + t_2) + t \tag{29}$$

The points on the circle that are a valid position of camera 1 are those where the ray f_{2a} intersects ℓ_A. Since the x and z coordinates are constant for all points on ℓ_A, this yields a polynomial system of two equations with two unknowns: ϕ and s_{2a}.

$$[A_2]_z = [\ell_A]_z \tag{30}$$

$$[A_2]_x = [\ell_A]_x \tag{31}$$

With a slight abuse of notation, $[p]_m$ refers to the m coordinate component of the vector p. Solving for s_{2a} in (30) and substituting into (31) gives an eighth degree polynomial in ϕ. The real roots of this polynomial in $(-1,1)$ encode the possible poses of the camera.

2.2 Ten-point Pose Estimation

Performing five-point line localization independently per each camera determines two disjoint sets of 2D-3D correspondences, all of them coplanar. We leverage the line pattern geometry, the known plane-to-camera distances, and the known relative pose among the cameras to define a succinct geometric formulation of absolute pose. The pose of the camera system can be uniquely determined in closed form from a triplet of 3D points, two localized in camera 1 and the other in camera 2. Note that ascertaining the pattern-to-camera distances requires five observations per scanline, leading us to denote this model a ten-point solver. The absolute camera pose is

$$R = \begin{bmatrix}\hat{g}_y \times \hat{g}_z & \hat{g}_y & \hat{g}_z\end{bmatrix}^\top \tag{32}$$

$$t = A_1 - s_{1a}Rf_{1a} \tag{33}$$

where

$$\hat{g}_y = \hat{\mathbf{u}}\left((s_{1a}f_{1a} - s_{2a}R_2 f_{2a} - t_2)/[A_1 - A_2]_y\right) \tag{34}$$

$$\hat{g}_z = \hat{\mathbf{u}}\left((s_{1e}f_{1e} - s_{1a}f_{1a}) \times \hat{g}_y\right) \tag{35}$$

This amounts to constructing the world frame in camera space as shown in Fig. 3**c**. We leverage the facts that 1) ℓ_A is parallel to the Y axis of the world frame and 2) the vector $s_{1e}f_{1e} - s_{1a}f_{1a}$ points in the positive direction of the X axis of the world frame.

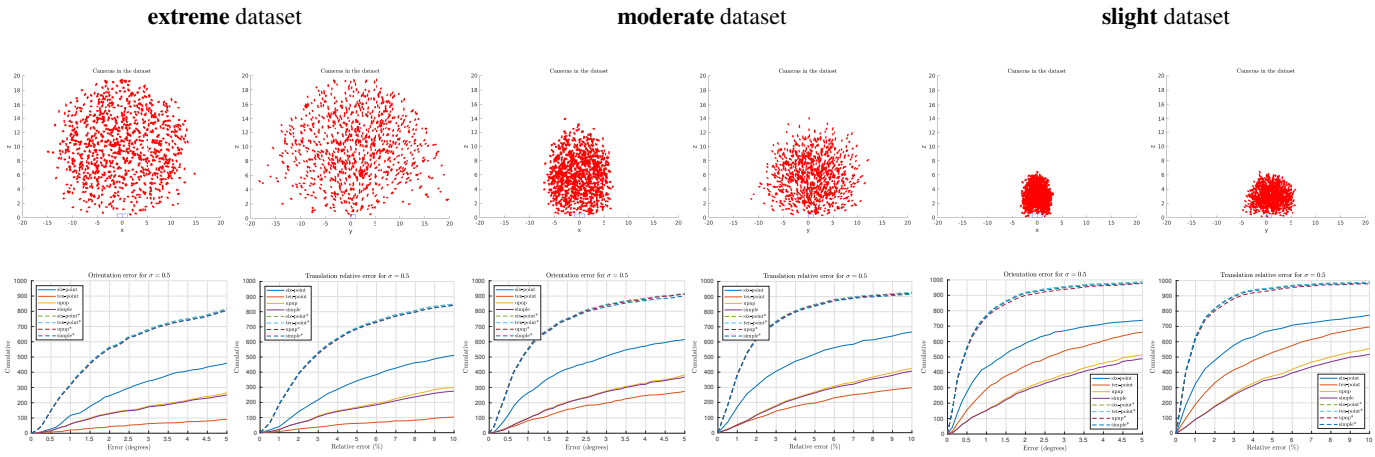

Fig. 4. Results on synthetic data. Top) Spatial distribution of cameras (red) for our synthetic datasets. The pattern is the (blue) rectangle centered on $x = 0$, $y = 0$, and $z = 0$. Bottom) Orientation and relative translation errors for the six-point, ten-point, upnp [16] and simple [26] methods operating on input corrupted with 1D Gaussian noise $\sigma = 0.5$ pixels. The $*$ suffix denotes methods benefiting from input measurement refinement. The cumulative histograms show the number of samples (out of 1000) whose error is less than the value given by the horizontal axis.

2.3 Measurement Refinement

Both the six-point and ten-point methods are sensitive to noise, predominantly due to our reliance on cross ratio evaluations to determine the 3D points on the pattern. When the measured 1D points contain noise, the associated 2D-3D correspondences may violate our known binocular geometry and/or planar scene definition. For noise-corrupted inputs, we may detect independent 2D lines whose orientation is not compliant with the known binocular viewing geometry or determine a pair of 3D lines that are not coplanar. Thus, the computed distances between the 3D points of camera 1 and camera 2 may not match the distances between their corresponding 3D points on the pattern (e.g. $\|s_{1a}f_{1a} - s_{2a}R_2f_{2a} - t_2\| \neq \|A_1 - A_2\|$), so the geometry of the pattern is broken (see Fig. 3**d**). Therefore, there may not exist a rigid transformation (R,t) between the points in the camera space and the points on the pattern, i.e., the expression

$$R = \begin{bmatrix} h'_x & h'_y & h'_x \times h'_y \end{bmatrix} \begin{bmatrix} h_x & h_y & h_x \times h_y \end{bmatrix}^{-1} \tag{36}$$

where

$$h_x = s_{1a}f_{1a} - s_{1e}f_{1e} \tag{37}$$

$$h_y = s_{1a}f_{1a} - s_{2a}R_2f_{2a} - t_2 \tag{38}$$

$$h'_x = A_1 - E_1 \tag{39}$$

$$h'_y = A_1 - A_2 \tag{40}$$

does not yield a valid rotation matrix R. However, since both the six-point and ten-point methods always return a valid R by design, the estimated pose will be inaccurate.

To improve the accuracy of the six-point and ten-point methods, we adjust the x coordinate of the measured 1D points such that they are congruent with a possible pose of the camera rig while minimizing the reprojection error on the x axis. This is posed as a constrained minimization problem

$$\min \sum_{j \in \{a,b,c,d,e\}} (j_1 - j'_1)^2 + (j_2 - j'_2)^2 \tag{41}$$

s.t.

$$\|s_{1a}f_{1a} - s_{2a}R_2f_{2a} - t_2\|^2 = \|A_1 - A_2\|^2 \tag{42}$$

$$\|s_{1c}f_{1c} - s_{2c}R_2f_{2c} - t_2\|^2 = \|C_1 - C_2\|^2 \tag{43}$$

$$\|s_{1e}f_{1e} - s_{2e}R_2f_{2e} - t_2\|^2 = \|E_1 - E_2\|^2 \tag{44}$$

where j_i and j'_i denote, respectively, the adjusted and the original measurements ($j \in \{a,b,c,d,e\}, i \in \{1,2\}$). From the twenty-five possible inter-camera constraints, this particular triplet was determined to sufficiently encode the rigidity of the pattern and provide a good trade-off between the quality of the results and the complexity of the minimization problem. Note that the distances between points belonging to the same camera always match the distances between their corresponding points on the pattern, so no additional information can be obtained from points of the same camera.

3 Experiments

We evaluated the quality of our six-point and ten-point solvers, the sensitivity of our geometric framework to noise, motion blur and image resolution, and the efficacy of our measurement refinement step.

3.1 Synthetic Data

We generated three synthetic datasets of 1000 random poses (R,t) each with uniform distribution. These datasets were labeled **slight**, **moderate** and **extreme**, according to their considered incidence angles and distances of the camera rig w.r.t. the pattern. We used these datasets to evaluate the robustness of the pose estimation methods under arbitrary poses. The baseline comparisons used were upnp [16] and the method of [26] (denoted as simple), which are state of the art general solvers for absolute pose estimation of generalized cameras. For the camera rig, we used simulated cameras with a resolution of 4K (3840x2160) and with calibration values (K_1, K_2, R_2, t_2) similar to our real camera rig.

For each pose, we projected the pattern into each image row of both cameras and kept only the rows where all the five lines of the pattern are observed within the row. Then, we uniformly selected one random row from camera 1 and one random row from camera 2, verifying that both rows do not observe the same line on the pattern. Using random rows tests the robustness of the methods against the separation between the scanlines (in practice, this separation angle is a hyper-parameter to optimize). From these rows, we extracted the horizontal coordinates a, b, c, d, and e where the lines of the pattern were observed and added Gaussian noise with $\sigma = 0.5$ pixels to evaluate robustness against noise.

Finally, we use the noisy 2D points as inputs for all pose estimation methods. For six-point, the 6th point is a_2 which maximizes the distance to e_2 and yields better results. For upnp and simple we use the ten points to obtain the 2D-3D correspondences (a_1,A_1), (e_1,E_1), and (a_2,A_2) used as inputs. In the case of multiple solutions, we selected the pose that minimizes the reprojection error of (e_2,E_2). The four methods were compared with and without measurement refinement. The results, along with the geometry of the datasets, are shown in Fig. 4.

From the experiments using the **extreme** dataset, we observe that the sensitivity to noise increases greatly when the camera rig observes the pattern from long distances or with extreme angles of incidence.

Even under extreme conditions, our optimization method can salvage most of the pose estimations. The reason the four methods perform badly without measurement refinement is that the noise is amplified greatly due to the 3D points on the pattern being obtained from the cross ratios of the noisy 2D measurements. For refined measurements, the median orientation error is less than 2 degrees and the median relative translations error is less than 3% for all methods.

The experiments using the **moderate** dataset show that even under moderate conditions, the unrefined measurements are unsuitable for pose estimation. For refined measurements, the median orientation error is less than 1 degree and the median relative translation error is less than 1.5% for all methods.

The experiments using the **slight** dataset show that the unrefined measurements are suitable for pose estimation only when the camera rig is close to the pattern and its orientation is not far from fronto-parallel. For refined measurements, the median orientation error is less than 0.5 degrees and the median relative translation error is less than 1%.

For all three datasets with unrefined measurements, we observe that the six-point method performs much better than the other three. This is because the pose estimation error for the six-point method is dominated by the measurement errors of camera 1. The other three methods mix the errors from both cameras, which may break the geometry of the pattern for the 3D points in camera space, and yield higher estimation error. For refined measurements, the four methods perform similarly.

Our results demonstrate our custom solvers have comparable performance w.r.t. the state of the art methods when measurement refinement is performed, or when the pattern is not far from the camera and the incidence angles are not large. However, the ten-point method has a unique solution while six-point, upnp, and simple can yield up to eight different solutions. Also, for the ten-point solver the output geometric variables are defined in closed form using basic arithmetic and trigonometric operations, which makes the ten-point solver the most suitable for a real-time low-latency high frequency hardware implementation. Therefore, *we focus exclusively on the ten-point solver hereafter*.

Effect of Motion Blur. We generated 1000 synthetic row images using Unity to evaluate the performance of the ten-point solver (with refinement) under motion blur at different velocities (Fig. 5). We use a translational velocity of 1.5 m/s and set the rotational velocity to 120 deg/s, 240 deg/s, 360 deg/s, and 480 deg/s. The distance from the camera to the pattern is about 50 cm. Each row has an integration time (exposure) of 2.31 ms (300 rows time). We passed the blurred rows to our pipeline. The results are shown in Table 2. At 480 deg/s our edge detector cannot detect the pattern.

Table 2. Mean orientation and relative translation error for 1000 synthetic row samples with motion blur. Translational velocity is 1.5 m/s.

	Rotational Velocity			
	120 deg/s	240 deg/s	360 deg/s	480 deg/s
Orientation Error	0.79°	1.25°	18.90°	N/A
Relative Translation Error	1.35%	1.83%	25.62%	N/A

Table 3. Mean orientation and relative translation error for the real dataset scaled at different resolutions.

	Resolution			
	3840x2160	1920x1080	1280x720	960x540
Orientation Error	0.74°	0.72°	0.72°	1.23°
Relative Translation Error	1.28%	1.26%	1.26%	1.92%

3.2 Real Data

We evaluated the performance of the ten-point solver on a dataset of static images captured with a pair of 4K RS cameras. In the absence of motion, all of the image rows have the same pose and we can obtain an approximation of the camera rig pose using P3P, which we used as our ground truth. Given that, by design, we know the 3D coordinates of the extreme points of the lines of the pattern, we used their detections in image 1 as 2D-3D correspondences for a P3P instance. We verified the quality of the estimated pose for each image pair by reprojecting the pattern onto both images.

Binocular Scanline Divergence. For each row v_1 in image 1 where the pattern is visible, we read row $v_2 = v_1 + v$ in image 2. If the pattern is visible in v_2, we estimate the pose by applying the ten-point solver on v_1 and v_2. We chose a row separation value of $v = 300$. This value was selected empirically after measuring the effect of the row separation on pose estimation error (see Fig. 5). As our stereo setup is close to rectified, smaller separations approach the degenerate configuration of shared collinear observations. Values of $v > 300$ yield diminishing returns and further reduce the number of samples we can extract from each image pair. If geometric accuracy were the sole consideration, we would set v to half the height of the image (1080) to minimize the error caused by the scanline separation. However, this would also reduce the maximum range of operation for our system, since the pattern would have to be observed in a larger region of the images. Our chosen value of v balances estimation robustness and the system's range of operation.

Results Analysis. Fig. 6 summarizes the orientation and relative translation errors for the ten-point solver for each pair of images in the dataset. As with our synthetic experiments, we observe our measurement refinement procedure is most effective on images with very large incidence angles and images far from the camera (nearing the maximum range of operation given by the chosen row separation $v = 300$). We also observe that the estimation error is dominated by the incidence angle w.r.t. the camera, such that we can obtain good results for incidence angles up to ~ 40 degrees without refinement, even if the pattern is far from the camera. This is because the accuracy of the cross ratio estimation depends on the distance between the closest pair of points [13], and this is most affected by having an oblique view of the plane. Edge detection is performed using the derivative of Gaussian ($\sigma = 2$ and window size of 1x13). Subpixel precision is achieved by fitting a parabola at the peaks of the response of the derivative of Gaussian filter. The parabola is fitted using three points: the peak, and the two points around it.

Effect of Image Resolution. We evaluate the performance degradation for lower resolution cameras. We shrink the images in the real dataset to 1920x1080, 1290x720, and 960x540. Results are shown in Table 3. Reduced horizontal resolution effectively favors capture instances where the 1D measurements are closer to each other, making the cross ratio numerically unstable, and rendering the solver more sensitive to noise. Reduced vertical resolution affects pose estimation throughput, given by rows $\times$ FPS.

Automatic Pattern Detection. We now address the challenge of automatically detecting our pattern within an arbitrary scene, using as input only our edge detection locations. This is a critical step for online deployment, since our geometric solvers lack verification mechanisms to determine the validity of the inputs provided to them. Assuming the pattern is observed without occlusion, we strive to identify a set of consecutive edges which are projectively compliant with our pattern's geometry. Toward this end, we augment our pattern with three additional vertical lines denoted as ℓ_{m6}, ℓ_{m7}, and ℓ_{m8}. These lines, along with ℓ_A, ℓ_C, and ℓ_E, are all parallel and define a total of $\binom{6}{4} = 15$ constant cross ratios. By keeping track of the last eight detected edges in the image scanline (which potentially correspond to observations of ℓ_A, ℓ_B, ℓ_C, ℓ_D, ℓ_E, ℓ_{m6}, ℓ_{m7}, and ℓ_{m8}) and verifying their cross ratios, we can determine whether these observations belong to the pattern (see Fig. 5). Note that in principle, just a single additional parallel line (i.e. ℓ_{m6}) is sufficient to identify a constant cross ratio. However, empirical edge measurement errors renders this process unreliable. The redundancy afforded by using three additional vertical lines robustly reduces false positives, at the expense of only moderate rates of missed detections (see Table 4).

4 FPGA Implementation Proof of Concept

To validate the feasibility of our geometric framework for real-time, low-latency, high frequency absolute pose estimation, we implemented the ten-point solver without measurement refinement on an FPGA.

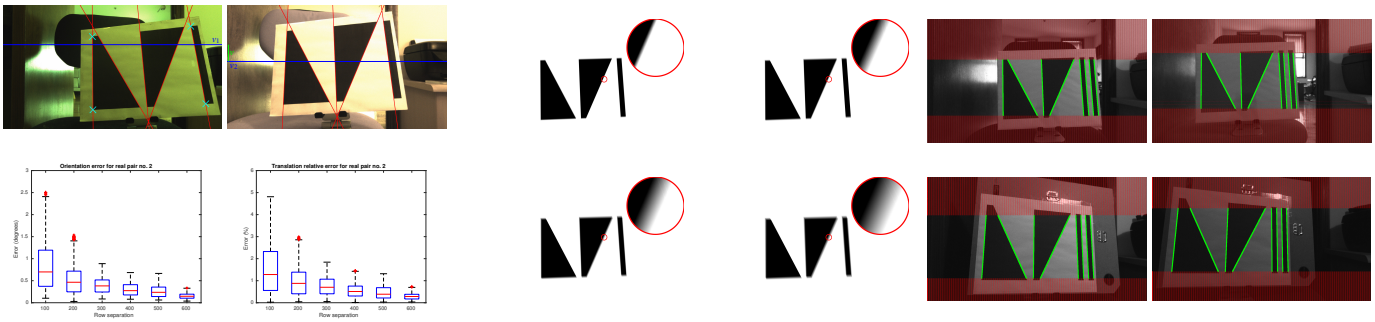

Fig. 5. Left) Effect of the row separation (v) on the accuracy of the ten-point solver. An illustration of a pair of processed scanlines (blue), their row separation (green), and the ground truth pose from P3P reprojected onto both images (red) with boxplots describing the distribution of orientation and relative translation errors evaluated over scanline pairs (mutually observing the pattern) having different row separation values. Pose estimates are computed without measurement refinement to avoid compensating for the effect of row separation. Center) Synthetic images (left camera) with motion blur (120 deg/s, 240 deg/s, 360 deg/s, and 480 deg/s). Right) Automatic pattern detection. We augment our pattern with a triplet of additional lines defining a set of constant cross ratios whose verification enables us to automatically determine if the pattern template is fully observed in a given input pixel scanline. Depicted vertical bars (red) illustrate rows where the pattern is not detected.

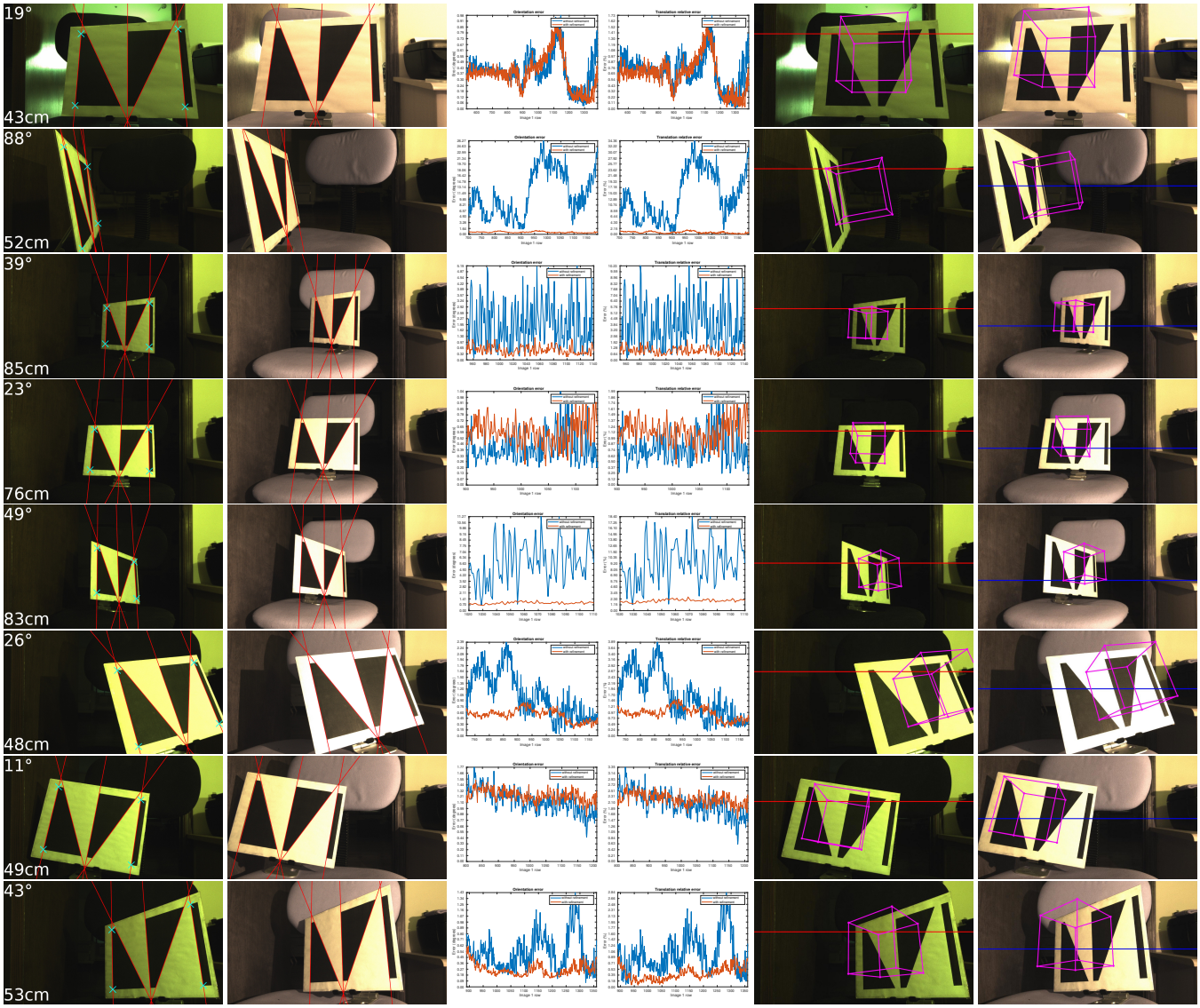

Fig. 6. Results on real data. Left) Captured image pairs with superimposed ground truth pose (points used for P3P ground truth shown in green), incidence angle and distance to camera rig. Middle) The orientation and relative translation errors for the ten-point solver are shown for every row of image 1 where the pattern was observed and has a corresponding row in image 2 where the pattern was also observed. Right) AR rendering results using the pose estimated by the ten-point solver. The scanlines used to compute the pose are depicted as horizontal lines (red and blue).

Table 4. Performance of the automatic pattern detector for the entire image (2160 rows) for a set of four real images.

	Additional lines		
	ℓ_{m6}	ℓ_{m6}, ℓ_{m7}	$\ell_{m6}, \ell_{m7}, \ell_{m8}$
Average False Positives	297	230	21
Average False Negatives	0	6	16

Table 5. FPGA resource usage for our scanline-level synchronization, derivative of Gaussian subpixel edge detector, automatic pattern detector implementations, and ten-point solver. Part is xczu9eg-ffvb1156 (see [43] for part specifications, including total available resources).

	Resource Type				
	LUT	LUTRAM	FF	BRAM	DSP
Synchronization	0.16%	0.00%	0.07%	0.11%	0.00%
Edge detector	0.55%	0.02%	0.54%	0.00%	0.00%
Pattern detector	9.66%	1.30%	7.94%	0.00%	3.49%
Ten-point solver	12.65%	4.43%	9.73%	0.00%	9.84%
Total	23.02%	5.75%	18.28%	0.11%	13.33%

As demonstrated by our experiments with synthetic and real data, we can use the ten-point solver without measurement refinement for short range absolute pose estimation.

Our experimental setup consists of a Zynq UltraScale+ MPSoC ZCU102 evaluation kit with two LI-IMX274MIPI-FMC cameras (see Fig. 1), which capture 4K (3840x2160) images and video at 60 FPS. The FPGA receives two pixels from the cameras per clock cycle (300 MHz) so every image row has a duration of 1920 clock cycles (6.4 μs) which is the upper-bound on compute latency for our estimation to run on real-time, i.e. the total latency of our implementation must be less than 1920 clock cycles to run in real-time.

Scanline-level Synchronization. The cameras are configured by sending commands through an I2C bus. We achieved scanline-level synchronization by implementing a control module on the FPGA that issues I2C commands (stored in a small ROM) to each camera with configurable delay. The control module has two I2C channels, each connected to one camera, and each I2C channel has an independent timer which controls when the commands are sent to the cameras. The timers have a resolution of one clock cycle so we can control when each camera is activated with very high precision. Both timers are initialized with distinct values such that the cameras capture different rows but the rows are captured at the same time. The value of both timers decreases by one every clock cycle. When the timer of an I2C channel reaches 0, the channel starts sending I2C commands to its camera to initialize it and begin capture. We selected timer values such that camera 1 is enabled first and camera 2 is enabled after a set number of lines from camera 1 (v) have passed. Scanline-level synchronization only happens once, during the initialization of the system.

Subpixel Edge Detector. The FPGA implementation of the derivative of Gaussian edge detector is fully pipelined (can accept two new pixels every clock cycle) and outputs subpixel coordinates with a latency of 54 clock cycles (180 ns). The subpixel coordinates are represented as signed 24-bit fixed-point numbers with 13 bits integer and 11 bits fractional parts. Thus, our edge detector has a resolution of $1/2048$ pixels. Compared to our MATLAB implementation, the FPGA implementation of the edge detector is exact to the whole pixel and the subpixel detection error is about 0.023 pixels.

Automatic Pattern Detector. The automatic pattern detector FPGA module is fully pipelined (can accept one 1D edge measurement on every clock cycle) and has a latency of 105 clock cycles (350 ns). Internally, the module uses a shift register to keep track of the last eight 1D measurements which potentially correspond to ℓ_A, ℓ_B, ℓ_C, ℓ_D, ℓ_E, ℓ_{m6}, ℓ_{m7}, and ℓ_{m8}.

Ten-point Solver. Our ten-point solver FPGA implementation is a fully pipelined design which accepts pattern data (ten 1D measurements) every clock cycle, and outputs the corresponding pose with a latency of 301 clock cycles (1 μs). The ten-point solver FPGA implementation uses IEEE 754 single precision floating point numbers (32-bit), so we performed additional experiments to measure the pose estimation error caused by the reduced precision and verify that the error is within an acceptable margin w.r.t. the double precision (64-bit) MATLAB implementation. We benchmarked on the synthetic data without noise and the precaptured real data. The maximum orientation error is 0.03 degrees and the maximum relative translation error is 0.01% w.r.t. the MATLAB implementation.

Throughput of Estimation Pipeline. The total latency of our implementation is $54 + 105 + 301 = 460$ clock cycles (1.5 μs, 24% of the upper-bound). Note that this latency is from pixel readout to pose estimation. Therefore, our implementation can work with 4K data in real-time and achieve a low latency absolute pose estimation with a frequency of rows $\times$ FPS $=$ 129.6 KHz. The total FPGA resource usage is about 23% (see Table 5).

5 Limitations

Pattern observability is a critical factor that depends on system design and deployment choices. Plainly put, in order to estimate 3D pose for every image scanline, the pattern needs to be the dominant item in the camera views. Our geometric formulation is not robust to occlusions, since all the lines of the pattern must be observed in the scanlines to be able to estimate the pose. This also includes the case where the pose of the camera does not allow observing all of the lines of the pattern (e.g. when the scanlines are almost parallel to the pattern lines or the pattern leaves the field of view of the cameras). The reported pose estimation frequency of 129.6 KHz constitutes the throughput attainable in our particular capture implementation as a function of the camera row frequency (FPS x Height). As our tracking module outpaces input capture (i.e. it completes estimation before the next pixel row has been completely read), our effective throughput for pose estimation renders the 129.6 KHz throughput as a strict lower bound. Clearly, for instances where the pattern is not detected, we will not be able to compute a pose, but the effective throughput of our per-row pose estimation pipeline is independent of this. As image blur affects the quality of 1D edge measurements, our method is sensitive to extreme velocities of egocentric motion. In this regard, further analysis of the empirical tradeoffs between camera gain, exposure times and capture setup is required. Finally, our model does not account for lens aberrations, requiring high-quality optics nominally free of radial distortions with sufficient depth of field.

6 Discussion and Conclusions

Our work has focused on the geometric principles and system integration aspects of developing the first reported online single-shot absolute pose tracker for rolling shutter. To this end, we leveraged insights from line incidence relations within the projective plane into absolute pose estimates for a generalized camera in Euclidean space. Although the geometric models developed are tractable, given that 1) we use the cross ratio as the basic unit of geometric analysis, and 2) 2D-3D correspondences need to be "lifted" from the input 1D observations; the mechanisms used to mitigate measurement error play a crucial role in deploying a functional localization instance. While both our six-point and ten-point solvers benefit from running measurement correction as a pre-process, the former is more robust and accurate, while the latter is considerably more efficient and straightforward to implement. Although aimed at instantaneous estimation, our approach is suitable for integration within larger systems enforcing RS temporal consistency for outlier filtering and increased robustness. Empirical feasibility was demonstrated through an FPGA-based implementation of our ten-point solver, highlighting the potential for low latency applications like tracking for AR/VR HMDs. The developed geometric insights and challenges addressed in this work constitute an effort towards more robust and general solutions capable of real-time low-latency high frequency operation.

REFERENCES

[1] C. Albl, Z. Kukelova, V. Larsson, M. Polic, T. Pajdla, and K. Schindler. From two rolling shutters to one global shutter. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2502–2510, 2020. doi: 10.1109/CVPR42600.2020.00258

[2] C. Albl, Z. Kukelova, and T. Pajdla. R6p - rolling shutter absolute pose problem. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2292–2300, June 2015. doi: 10.1109/CVPR.2015.7298842

[3] C. Albl, Z. Kukelova, and T. Pajdla. Rolling shutter absolute pose problem with known vertical direction. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3355–3363, June 2016. doi: 10.1109/CVPR.2016.365

[4] A. Bapat, E. Dunn, and J. Frahm. Towards kilo-hertz 6-dof visual tracking using an egocentric cluster of rolling shutter cameras. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2358–2367, Nov 2016. doi: 10.1109/TVCG.2016.2593757

[5] A. Bapat, T. Price, and J. Frahm. Rolling shutter and radial distortion are features for high frame rate multi-camera tracking. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4824–4833, June 2018. doi: 10.1109/CVPR.2018.00507

[6] A. Blate, M. Whitton, M. Singh, G. Welch, A. State, T. Whitted, and H. Fuchs. Implementation and evaluation of a 50 khz, 28μs motion-to-pose latency head tracking instrument. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1970–1980, 2019. doi: 10.1109/TVCG.2019.2899233

[7] Y. Dai, H. Li, and L. Kneip. Rolling shutter camera relative pose: Generalized epipolar geometry. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4132–4140, June 2016. doi: 10.1109/CVPR.2016.448

[8] A. Dhall, D. Dai, and L. Van Gool. Real-time 3d traffic cone detection for autonomous driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 494–501, 2019. doi: 10.1109/IVS.2019.8814089

[9] A. E. Güzel, D. Hisar, L. Claesen, and H. F. Uğurdağ. Fast incremental least square pose estimation for hardware implementation with rolling shutter camera. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2020. doi: 10.1109/SIU49456.2020.9302192

[10] I. Han. Car speed estimation based on cross-ratio using video data of car-mounted camera (black box). *Forensic Science International*, 269:89–96, 2016. doi: 10.1016/j.forsciint.2016.11.014

[11] R. M. Haralick, D. Lee, K. Ottenburg, and M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 592–598, June 1991. doi: 10.1109/CVPR.1991.139759

[12] R. Horaud, R. Mohr, and B. Lorecki. On single-scanline camera calibration. *IEEE Transactions on Robotics and Automation*, 9(1):71–75, 1993. doi: 10.1109/70.210796

[13] D. Huynh. The cross ratio: A revisit to its probability density function. *Procedings of the British Machine Vision Conference 2000*, pp. 27–27, 2000. doi: 10.5244/c.14.27

[14] E. Ito and T. Okatani. Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4512–4520, July 2017. doi: 10.1109/CVPR.2017.480

[15] T. Ke and S. I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4618–4626, July 2017. doi: 10.1109/CVPR.2017.491

[16] L. Kneip, H. Li, and Y. Seo. Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., *Computer Vision – ECCV 2014*, pp. 127–142. Springer International Publishing, Cham, 2014.

[17] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*, pp. 2969–2976, June 2011. doi: 10.1109/CVPR.2011.5995464

[18] Y. Lao and O. Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4795–4803, 2018.

[19] Y. Lao, O. Ait-Aider, and A. Bartoli. Rolling shutter pose and ego-motion estimation using shape-from-template. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds., *Computer Vision – ECCV 2018*, pp. 477–492. Springer International Publishing, Cham, 2018.

[20] D. Li, G. Wen, B. Wei Hui, S. Qiu, and W. Wang. Cross-ratio invariant based line scan camera geometric calibration with static linear data. *Optics and Lasers in Engineering*, 62:119–125, 2014. doi: 10.1016/j.optlaseng.2014.03.004

[21] D. D. Li, G. Wen, and S. Qiu. Cross-ratio–based line scan camera calibration using a planar pattern. *Optical Engineering*, 55(1):1 – 10, 2016. doi: 10.1117/1.OE.55.1.014104

[22] P. Lincoln, A. Blate, M. Singh, T. Whitted, A. State, A. Lastra, and H. Fuchs. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1367–1376, 2016. doi: 10.1109/TVCG.2016.2518038

[23] L. Magerand, A. Bartoli, O. Ait-Aider, and D. Pizarro. Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds., *Computer Vision – ECCV 2012*, pp. 456–469. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[24] S. J. Maybank. Probabilistic analysis of the application of the cross ratio to model based vision: Misclassification. *International Journal of Computer Vision*, 14(3):199–210, 1995.

[25] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras, 2005.

[26] M. H. Merzban, M. Abdellatif, and A. A. Abouelsoud. A simple solution for the non perspective three point pose problem. In *2014 International Conference on 3D Imaging (IC3D)*, pp. 1–6, Dec 2014. doi: 10.1109/IC3D.2014.7032594

[27] T. Nakai, K. Kise, and M. Iwamura. Hashing with local combinations of feature points and its application to camera-based document image retrieval. *Proc. CBDAR05*, pp. 87–94, 2005.

[28] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1235–1248, 2017. doi: 10.1109/TVCG.2016.2592910

[29] D. Nister. A minimal solution to the generalised 3-point pose problem. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. I–I, June 2004. doi: 10.1109/CVPR.2004.1315081

[30] M. Persson and K. Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[31] S. Ramalingam, M. Antunes, D. Snow, G. Hee Lee, and S. Pillai. Line-sweep: Cross-ratio for wide-baseline matching and 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[32] V. Rengarajan, A. N. Rajagopalan, and R. Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2773–2781, June 2016. doi: 10.1109/CVPR.2016.303

[33] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys. Rolling shutter stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 465–472, 2013.

[34] O. Saurer, M. Pollefeys, and G. H. Lee. A minimal solution to the rolling shutter pose estimation problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1328–1334, Sep. 2015. doi: 10.1109/IROS.2015.7353540

[35] O. Saurer, M. Pollefeys, and G. H. Lee. Sparse to dense 3d reconstruction from rolling shutter images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3337–3345, June 2016. doi: 10.1109/CVPR.2016.363

[36] D. Schubert, N. Demmel, V. Usenko, J. Stuckler, and D. Cremers. Direct sparse odometry with rolling shutter. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 682–697, 2018.

[37] D. Su, A. Bender, and S. Sukkarieh. Improved cross-ratio invariant-based intrinsic calibration of a hyperspectral line-scan camera. *Sensors*, 18(6), 2018. doi: 10.3390/s18061885

[38] R. Usamentiaga, D. F. Garcia, and F. J. de la Calle. Line-scan camera calibration: a robust linear approach. *Appl. Opt.*, 59(30):9443–9453, Oct 2020. doi: 10.1364/AO.404774

[39] S. Vasu, M. Mohan M.R., and A. N. Rajagopalan. Occlusion-aware rolling

shutter rectification of 3d scenes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 636–645, June 2018. doi: 10.1109/CVPR.2018.00073

[40] B. Wang, H. Hu, and C. Zhang. New insights on multi-solution distribution of the p3p problem. *Image and Vision Computing*, 103:104009, 2020. doi: 10.1016/j.imavis.2020.104009

[41] K. Wang, C. Liu, K. Wang, and S. Shen. Depth estimation under motion with single pair rolling shutter stereo images. *IEEE Robotics and Automation Letters*, pp. 1–1, 2021. doi: 10.1109/LRA.2021.3063695

[42] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug 2003. doi: 10.1109/TPAMI.2003.1217599

[43] Xilinx. Ultrascale architecture and product data sheet: Overview. `https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf`.

[44] Yeyin Zhang, Kaiqi Huang, Yongzhen Huang, and Tieniu Tan. View-invariant action recognition using cross ratios across frames. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3549–3552, 2009. doi: 10.1109/ICIP.2009.5414338

[45] J. Yu, W. Jiang, Z. Luo, and L. Yang. Application of a vision-based single target on robot positioning system. *Sensors*, 21(5), 2021. doi: 10.3390/s21051829

[46] B. Zhuang, L. Cheong, and G. H. Lee. Rolling-shutter-aware differential sfm and image rectification. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 948–956, Oct 2017. doi: 10.1109/ICCV.2017.108

[47] B. Zhuang, Q.-H. Tran, P. Ji, L.-F. Cheong, and M. Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4551–4560, 2019.