

Minimal Solvers for 3D Geometry from Satellite Imagery

Enliang Zheng, Ke Wang, Enrique Dunn, and Jan-Michael Frahm
 The University of North Carolina at Chapel Hill
 {ezheng, kewang, dunn, jmf}@cs.unc.edu

Abstract

We propose two novel minimal solvers which advance the state of the art in satellite imagery processing. Our methods are efficient and do not rely on the prior existence of complex inverse mapping functions to correlate 2D image coordinates and 3D terrain. Our first solver improves on the stereo correspondence problem for satellite imagery, in that we provide an exact image-to-object space mapping (where prior methods were inaccurate). Our second solver provides a novel mechanism for 3D point triangulation, which has improved robustness and accuracy over prior techniques. Given the usefulness and ubiquity of satellite imagery, our proposed methods allow for improved results in a variety of existing and future applications.

1. Introduction

Commercial satellite imagery, captured on a variety of hardware platforms, has become widely available by multiple vendors. It is a valuable source of information and has received much attention by the computer vision community. For instance, it has driven work on geo-localization [20], meteorological and oceanographic forecasts [19], urban change detection [24], and road detection [25].

To effectively leverage this imagery, the relationship between the underlying 3D scene geometry and its 2D image has to be known. This is typically provided by a camera model. However, defining a precise physical camera model is complicated due to the complex hardware configuration and the capture process. It is standard practice to construct an approximation to the mapping between the 3D object space and the image.

The representation that has been used to approximate this mapping for over two decades is the Rational Polynomial Coefficient (RPC) model. It was first introduced by Hartley and Saxena [14] and greatly contributed to the thriving of satellite imagery applications. The primary reason for its success is its ability to maintain the full accuracy of physical sensor models, its unique characteristic of sensor independence, and its real-time calculation [18]. It is

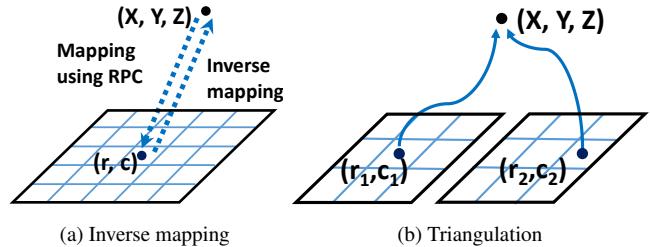


Figure 1: The two problems solved in our paper.

included in the standards of the NITF image format [1], and is available from most satellite image vendors [2].

However, one challenge faced by many works utilizing satellite imagery and the RPC model is that the bidirectional mapping between the 3D object and 2D image spaces may be unknown. Specifically, vendors supplying the satellite imagery may only provide the one-directional mapping function from the 3D object space to 2D image (known as a forward RPC model). The inverse mapping, termed the inverse RPC model, utilizes 2D image coordinates and an associated altitude in order to infer the resulting 3D coordinate. Approaches attempting to solve for this inverse mapping often lack accuracy [31]. In this work, we propose two novel high-accuracy minimal solvers (see Fig. 1) for use with satellite imagery only requiring the forward RPC.

The first minimal solver for the inverse mapping targets the domain of extracting depth information from pairs of satellite images (stereo). Previous works have explored this domain [11, 10, 18], but suffer from various limitations. For instance, a typical approach relying on volumetric models can have high memory requirements [5]. Furthermore, recent image-to-image approaches rely on the existence of an inverse RPC model, which may be missing or inaccurate [15, 26, 30]. Our approach avoids these issues by proposing an efficient minimal solver, which bypasses the issue of explicitly constructing a single inverse RPC model for the entire image. Instead, each coordinate is processed independently, providing a highly-accurate inverse mapping conforming to the provided forward RPC model.

The second novel minimal solver targets the application of 3D point triangulation. In typical computer vision applications, 3D point triangulation is the task of finding the optimal 3D point position given correspondences among feature points in calibrated images. However, triangulation under the forward RPC model is much more difficult, mainly due to the increased complexity of the camera model [29]. Moreover, since no explicit epipolar constraints exist for the RPC camera model, feature mismatches cannot be easily detected and removed. Our approach solves these issues by first proposing an effective strategy for a minimal solver based triangulation under the forward RPC model, and then by making the observation that only three of the four coordinates in a correspondence are needed to solve the equations. The fourth coordinate can then be used both for robustness (enabling us to reject false matches), as well as for refinement (leveraging redundant information). This is novel compared to typical triangulation schemes, and we demonstrate the usefulness and effectiveness of this strategy in our results.

Given the usefulness and ubiquity of satellite imagery and the RPC model, our proposed methods allow for improved results in a variety of existing and future applications. For low-level problems and stereo computations, our exact image-to-object space mapping increases the accuracy of 3D reconstructions. For high-level tasks, we provide improved feature matching and outlier rejection mechanisms to improve the results.

2. Related Work

Most applications leveraging satellite imagery for reconstruction and analysis require a known camera model in order to correctly function. A popular technique to provide the required camera calibration is the RPC model [14]. After its introduction, Tao *et al.* [28] studied the characteristics of the RPC model and proposed an iterative approximation solution to estimate RPC models from the physical sensor model or using ground control points. Hu *et al.* [17] proposed to exploit control information to further improve the accuracy of the solution for RPC models. To further boost accuracy, Dial *et al.* [9] leverage bundle adjustment to refine the the RPC coefficients. All of the above methods provide numerical approximations to the RPCs for satellite imagery. However, numerical stability is and was a major challenge in these solutions, which had typically been mitigated by leveraging additional information such as sensor characteristics or ground control points [28, 9]. In this paper we propose a solution to achieve a numerically stable and accurate solution to determine the image-to-object-space mapping given an RPC model describing the object-to-image-space transformation.

Once the RPC model is known, computer vision and photogrammetry applications have leveraged them in a va-

riety of applications. For instance, given a pair of matching features from two stereo satellite images, Tao *et al.* [29] reconstruct the corresponding approximate 3D point by an iterative approach that relies on the first-order Taylor approximation of the RPC model. However, lacking a rigorous analytical convergence proof and outlier removal mechanisms, the automatic processing of satellite images is impeded. Several other approaches also provide methods to perform dense stereo reconstruction from satellite images [8, 16, 12, 26]. Hirschmüller [15] used an approximation of global optimization to efficiently solve the dense stereo problem. Oh *et al.* [26] rectify the satellite images by approximating the epipolar curve with piecewise straight lines, and then search for correspondences along the line of the rectified images using normalized cross correlation measures. It is unclear if the linear approximations work for images with large distortions. Additionally, this method also requires that the inverse RPC parameters are available. Wang *et al.* [30] efficiently compute depthmaps of large satellite images by testing a reduced set of disparity hypotheses for each pixel. These methods are very effective when handling high-resolution satellite images, but a bi-directional mapping is required for such methods to work. We propose a novel minimal solver to accurately establish this bi-directional mapping.

3. RPC model

There are two broadly used classes of sensor models, the physical sensor model and the generalized sensor model. The physical sensor model deduces the imaging characteristics from the physical imaging process of the specific sensor used for acquisition. The parameters describe the position and orientation of the sensor with respect to an object-space coordinate system. These physical sensor models typically yield high modeling accuracy when used for mapping. However, the physical models across different satellites may vary significantly and do not provide a decent generalization. Accordingly, image analysis algorithms often have to be tailored for the specific satellite.

To overcome this limitation of specificity and to achieve generalization across sensor platforms (satellites), generalized sensor models were proposed. In the generalized sensor model, the transformation between the image and the object space is represented as a general mapping function without modeling the specific physical imaging process of the satellite. There are a number of common representations of the mapping function, for instance polynomials or rational functions. In either of these models, the parameters do not carry any physical meanings related to the imaging process as is the case for the physical sensor models, so the algorithms are fixed and can be directly apply across different satellites. Utilizing the RPC model to replace physical sensor models in photogrammetric mapping is becoming a

standard way for accurate, economical, and fast mapping. Next, we describe the RPC model in detail.

To boost accuracy, the RPC model typically operates in a normalized space. The normalization functions, for the unnormalized object space coordinates \tilde{X} , \tilde{Y} , and \tilde{Z} , are denoted as Φ^X , Φ^Y , and Φ^Z , given by:

$$\begin{aligned} X &= \Phi^X(\tilde{X}) = (\tilde{X} - X_o)/X_s, \\ Y &= \Phi^Y(\tilde{Y}) = (\tilde{Y} - Y_o)/Y_s, \\ Z &= \Phi^Z(\tilde{Z}) = (\tilde{Z} - Z_o)/Z_s, \end{aligned} \quad (1)$$

with X_o , Y_o , and Z_o denoting the offset of object coordinates, and X_s , Y_s and Z_s representing the corresponding scale factors for the coordinate directions. Similarly, the image coordinates are normalized through:

$$r = (\tilde{r} - r_o)/r_s, \quad c = (\tilde{c} - c_o)/c_s, \quad (2)$$

with r denoting the normalized row coordinate of the image and c the normalized column in the image, and r_o, c_o being the offsets of the image coordinates. The scale factors for the image coordinates are denoted by r_s, c_s . The normalization parameters are set to makes sure that the 3D region covered by the satellite image has normalized coordinates within the cube of $[-1, 1] \times [-1, 1] \times [-1, 1]$, and that the normalized image coordinates are within the range of $[-1, 1] \times [-1, 1]$. There parameters are provided by the satellite image vendors with the RPC model. For convenience, we denote $M = (X, Y, Z)$, $\tilde{M} = (\tilde{X}, \tilde{Y}, \tilde{Z})$, and the 3D point normalization as $M = \Phi(\tilde{M})$.

Then, the forward RPC model relates the normalized object coordinates X, Y, Z to the normalized image coordinates (r, c) :

$$r = \frac{P^1(X, Y, Z)}{P^2(X, Y, Z)}, \quad c = \frac{P^3(X, Y, Z)}{P^4(X, Y, Z)}, \quad (3)$$

with the P^i , $i = 1, 2, 3, 4$ being polynomials of degree three. Also, notice that in Eq. (3) the row and column of the image coordinates are independently computed. The P^i , $i = 1, 2, 3, 4$ are of the form (superscript i is omitted for simplicity)

$$\begin{aligned} P = a_0 + a_1X + a_2Y + a_3Z + a_4XY + a_5XZ \\ + a_6YZ + a_7X^2 + a_8Y^2 + a_9Z^2 + a_{10}XYZ \\ + a_{11}X^2Y + a_{12}X^2Z + a_{13}XY^2 + a_{14}Y^2Z \\ + a_{15}XZ^2 + a_{16}YZ^2 + a_{17}X^3 + a_{18}Y^3 + a_{19}Z^3, \end{aligned} \quad (4)$$

where a_j , $j = 0, \dots, 19$ are the polynomial coefficients. It is understood that distortions caused by the optical projection during the satellite image capture can generally be represented by the ratios of first-order terms of the P^i . Corrections such for the earth curvature, the atmospheric refraction, etc., can be well approximated by the second-order

terms. Additional unmodeled distortions are approximated through the high-order components, where one example of these distortions is the camera vibration that can be modeled by the third-order terms [28]. Typically, the low-order monomial terms in Eq. (4) have a significantly larger coefficients than high-order monomials.

The inverse RPC model performs image-to-space transformation. Given the input of the normalized row and column (r, c) , and the normalized altitude Z of the corresponding 3D point, it computes associated X and Y . The model is almost the same to that of forward RPC model as follows,

$$X = \frac{P^5(r, c, Z)}{P^6(r, c, Z)}; \quad Y = \frac{P^7(r, c, Z)}{P^8(r, c, Z)}; \quad (5)$$

where P^i , $i = 5, 6, 7, 8$ has the form of polynomial as in Eq. (4). While some satellite image vendors deliver both the forward and the inverse RPC model with their data, most of them only provide the forward RPC model. We also find estimating the inverse RPC model is difficult without detailed knowledge of the sensor and/or ground control points. In the remainder of this paper, we use the term RPC model to represent forward RPC model since we assume the inverse RPC is unknown.

4. Inverse mapping

This section proposes the parameterizations of the inverse mapping, a minimal solver based on a polynomial equation system, and our proposed solution using the Gröbner basis method. We close the discussion with the application of the novel minimal solver to stereo estimation.

4.1. Inverse Mapping Parameterization

Unlike traditional methods that explicitly model the inverse mapping and estimate its parameters, our method maps each individual point without any assumption about the model of the inverse mapping (inverse RPC).

To compute the inverse mapping, we rewrite the RPC mapping from Equation (3) as:

$$\begin{cases} P^1(X, Y, Z) - rP^2(X, Y, Z) = 0 \\ P^3(X, Y, Z) - cP^4(X, Y, Z) = 0 \end{cases} \quad (6)$$

which defines a polynomial equation system. Assuming a known normalized altitude Z and after expanding each of P^i in Eq. (6), we can obtain a polynomial equation system of two equations. Each has degree three in the unknown variables X and Y . Using computational algebra software (e.g. Maple or Macaulay2 [13]) it can be confirmed that the equation system of Eq. (6) has up to nine solutions. Next, we describe the Gröbner bases method for our solution.

4.2. Solving polynomial equations

In order to solve the polynomial equation system of Eq. (6), we leverage the Gröbner basis method which has

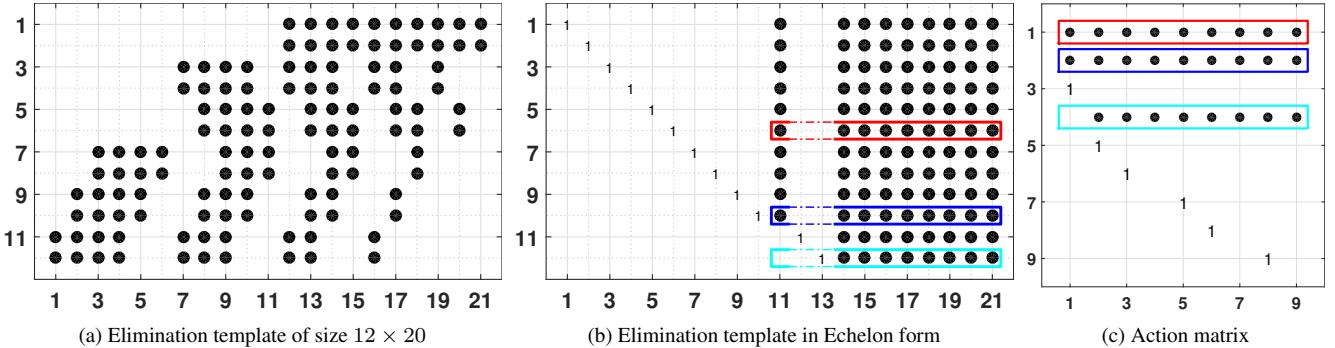


Figure 2: The Gröbner basis method for inverse mapping problem. The black dots represent non-zero values. The value of the three rows in the action matrix is obtained from the three rows in figure (b).

achieved great success in solving minimal problems encountered in computer vision [27, 23]. From the existing standard approaches [4, 22, 27, 3], we opt to use the efficient and accurate method proposed by Kukelova *et al.* [22] as we empirically found it to produce very accurate results.

Solving a polynomial equation system typically involves two steps. The first step is an offline process in which the specific behavior of the problem is studied, and a pattern for the solution is determined. The second step, an online process, uses the determined pattern to set up and solve the problem for a given input.

Starting with the offline process, it consists of two main parts. In the first part, the problem (Eq. (6)) is modeled in a computer algebra software package (such as Macaulay2 [13]). We then leverage the built-in functions of the software package to first compute the Gröbner bases, and then the bases of the quotient ring (which in turn defines the number of solutions to our problem). During this computation, in order to avoid issues of numerical precision, the problem is represented using a finite field (which uses fixed, discrete values instead of a floating-point representation). For these computations, we use the GrevLex as the monomial ordering, as this typically results in fewer overall computations compared to other alternative ordering schemes (such as lexicographic) [7].

The second part of the offline process seeks to construct a modified set of polynomial equations that can be used for action matrix construction. In order to achieve this, we first construct an elimination template (e.g. Fig. 2a) which represents our initial equations (the first two rows of the template correspond to Eq. (6)). Here, the polynomial equations are represented as $\mathbf{M}\mathbf{X} = \mathbf{0}$, where \mathbf{X} is a vector of monomial terms, and \mathbf{M} is a matrix of polynomial coefficients (the elimination template [3]). Given these initial equations, we then multiply them by various monomial terms, yielding additional entered rows in the elimination template. The purpose of these additional entries is to en-

able the successful construction of an action matrix in a following step. Given the elimination template, we then perform Gauss-Jordan elimination to reduce the matrix to echelon form. Once in echelon form, the matrix has certain properties which enable the construction of an action matrix by leveraging the bases of the quotient ring computed above. The exact method, properties, and theory for how to construct the elimination template and action matrix are complex and beyond the scope of this paper. Therefore, we refer readers to [7, 22, 23] for more details.

By successfully constructing the action matrix, we now have a pattern for how the original set of equations can be solved. The online portion of the processing now utilizes actual input values to form the elimination template, perform Gauss-Jordan elimination, and construct the action matrix. Then, the final numerical solutions to the polynomial equations can be computed from the eigenvectors of the action matrix. As much of this process has been detailed before [7, 22, 23], we now only provide the details of the offline procedures for our specific problem.

For our inverse mapping problem, the polynomial equation system consists of two polynomials, with each containing 10 monomials (Eq. (6)). The polynomial equations are multiplied by the following monomials $\{X, Y, X^2, XY, Y^2\}$, and the resulting equations are added to form the elimination template (Fig. 2a). Then, the echelon form of the elimination template is computed by Gauss-Jordan elimination (Fig. 2b), and the action matrix is constructed. Here, the negated values of row 6, 10 and 12 in Fig. 2b are used for action matrix construction as shown in Fig. 2c. After constructing the action matrix, the 7th and 8th elements of the eigenvectors are the solutions of the polynomial equation system.

4.3. Inverse mapping in stereo

After describing how to solve for the inverse mapping, we next detail its usage within the context of stereo estima-

tion for a pair of stereo images I_1 and I_2 . For convenience of notation, we use the subscript 1 and 2 to denote the associated properties for I_1 and I_2 .

In traditional image based stereo for pinhole cameras, it is known that points corresponding to point p_1 in the first image I_1 can only be found along the corresponding epipolar line $l_2(p_1)$ in the second image I_2 . The epipolar line $l_2(p_1)$ effectively represents the depth ambiguity of the point p_1 in the second image I_2 , i.e. the position along the line corresponds to the depth of the point. Having this reduction to a one-dimensional search space (epipolar line) significantly reduces the computational expense of the stereo estimation. Leveraging the inverse mapping in combination with the RPC model, we are also able to establish a one dimensional search space for image-to-image stereo from satellite images. In contrast to the pinhole camera model that has an epipolar line, the satellite images have epipolar curves with unknown analytical formulation [26]. However, using the inverse RPC mapping we can still numerically identify the epipolar curve in the second image.

Starting from an image point $(\tilde{r}_1, \tilde{c}_1)$ in pixel coordinates, we first apply the image point normalization of Eq. (2) to obtain the normalized point (r_1, c_1) in the first satellite image. To explore the ambiguity space for the normalized altitude Z_1 of the corresponding object point, we can sample the normalized altitude in the range $[-1, 1]$ (Sec. 3). For each specific sample \hat{Z}_1^i , the inverse mapping can be applied to compute the corresponding normalized 3D object point $\hat{M}_1 = (\hat{X}_1^i, \hat{Y}_1^i, \hat{Z}_1^i)$. The set of samples $\{\hat{Z}_1^i\}$ defines a curve in the object space corresponding to the depth ambiguity of the image point $(\tilde{r}_1, \tilde{c}_1)$. After de-normalizing from the normalized 3D points \hat{M}_1^i to real object space, apply the normalization of the second image and the RPC model of I_2 to compute the 2D point $(\hat{r}_2^i, \hat{c}_2^i)$ on I_2 . In practice we adjust the sampling rate to produce a sub-pixel sampling in the second satellite image. Fig. 3a shows a 2D point and its corresponding epipolar curve for a pair of stereo satellite images. The particular epipolar curve in Fig. 3b is close to a line, but epipolar curves generally show different curvatures in different parts of the image. As in the case of standard stereo estimation, the limited altitude range further shortens the epipolar curve in the second image. This leads to a further reduced search space in the second satellite image.

5. Triangulation

We now describe our method for robust satellite sparse point triangulation, which allows us to efficiently remove outlier correspondences during feature triangulation. We address the problem of obtaining the normalized 3D point from a pair of correspondences with normalized coordinates (r_1, c_1) in image I_1 and (r_2, c_2) in image I_2 with known RPC models. Our key observation is that it is sufficient to use three out of the four coordinate values to triangulate the

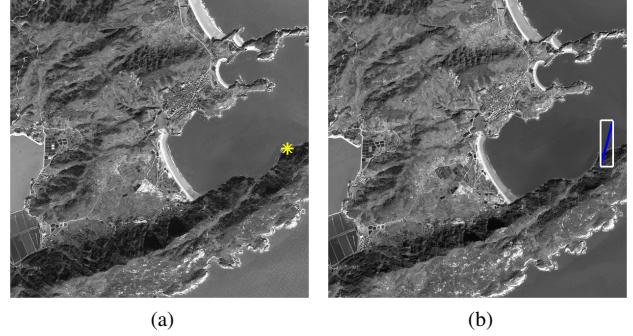


Figure 3: The yellow point on the left image and its corresponding blue epipolar curve on the right image. The white rectangle is the region of feature candidates for matching.

3D point. Hence, the last coordinate value can be used for correspondence verification. Without loss of generality, we use r_1, c_1 and r_2 for triangulation, and c_2 for verification.

For a 3D point $(\tilde{X}, \tilde{Y}, \tilde{Z})$ in the real object space that is observed by I_1 and I_2 , and given the known RPC models (Eq. (3)) for images I_1 and I_2 , we can construct the following polynomial equation system

$$\left\{ \begin{array}{l} P_1^1(\Phi_1^X, \Phi_1^Y, \Phi_1^Z) - r_1 P_1^2(\Phi_1^X, \Phi_1^Y, \Phi_1^Z) = 0 \\ P_1^3(\Phi_1^X, \Phi_1^Y, \Phi_1^Z) - c_1 P_1^4(\Phi_1^X, \Phi_1^Y, \Phi_1^Z) = 0 \\ P_2^1(\Phi_2^X, \Phi_2^Y, \Phi_2^Z) - r_2 P_2^2(\Phi_2^X, \Phi_2^Y, \Phi_2^Z) = 0, \end{array} \right. \quad (7)$$

where Φ^X , Φ^Y and Φ^Z are the known normalization functions for the point in the object space given by Eq. (1). Eq. (7) defines a system of three polynomial equations, and each of the polynomial has degree 3 in the unknown variables \tilde{X} , \tilde{Y} and \tilde{Z} . Similarly to the inverse mapping, Eq. (7) has in general up to 27 solutions, some of which have real values. Solving Eq. (7) again uses the framework of the Gröbner basis as briefly described below.

Solving the polynomial equation system for triangulation is similar to that of inverse mapping, but slightly more complicated. The original 3 polynomials are multiplied by all the 34 monomials in variables $\{\tilde{X}, \tilde{Y}, \tilde{Z}\}$ of degrees from 1 up to 4, and added to the original 3 equations. The number of total equations is $35 * 3 = 105$, but within which only 93 are linearly independent. By choosing 93 linearly independent equations, we have the elimination template of size 93×120 (Fig. 4a). After computing the echelon form (Fig. 4b), the negated values in the submatrix from the rows $\{36, 62, 63, 78, 79, 80, 89\}$ and columns $\{64, 83—85, 95—100, 104—120\}$ are used in row $\{1—3, 5—7, 11\}$ of the action matrix (Fig. 4c). The 24th, 25th, and 26th elements of the eigenvectors of the action matrix are solutions to the polynomial equation system.

After estimating the 3D object point \tilde{M}^* , we next need to evaluate the obtained solution leveraging the remaining

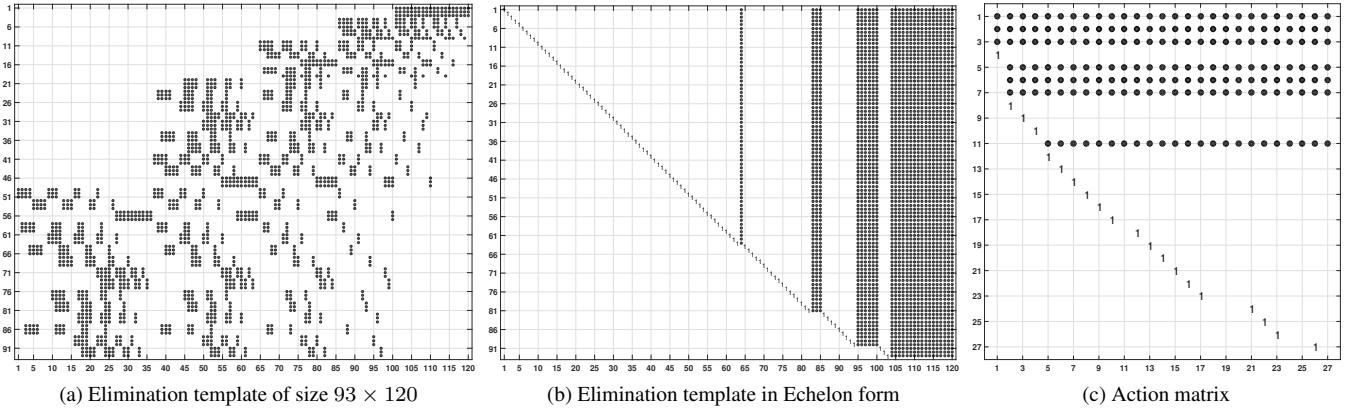


Figure 4: The Gröbner basis method for triangulation problem. The eliminate template shapes may vary depending on which linearly independent equations are chosen, but the shape of the echelon form is fixed.

point coordinate \tilde{c}_2 . During this, \tilde{c}_2 acts as a role to remove the outlier correspondences and the fake real solutions of the polynomial equation system. The estimated 3D point $\tilde{M}^* = (\tilde{X}^*, \tilde{Y}^*, \tilde{Z}^*)$ is projected onto I_2 to compute the column \tilde{c}_2^* . If the distance $\tilde{d}^* = |\tilde{c}_2 - \tilde{c}_2^*|$ is below a predefined threshold (e.g. 2 pixels in our experiments), the correspondence is considered to be correct. Since Eq. (7) generates multiple real valued solutions, each real valued solution is tested for correctness. If none of the solutions pass the test, the 2D match is considered to be an outlier. For correct matches (inlier), typically only one solution passes the inlier test. Whenever multiple solutions pass the test, we accept the one with the smallest distance \tilde{d}^* .

Beyond serving for validation, the redundancy during the triangulation can also be used to further refine the estimated 3D point \tilde{M}^* starting from the obtained solution. We propose to determine the refined 3D point by minimizing the projection error in both images through

$$\min_{\tilde{M}} \sum_{i=1}^2 \left\| \frac{P_i^1(\Phi_i(\tilde{M}))}{P_i^2(\Phi_i(\tilde{M}))} - r_i \right\|_2^2 + \left\| \frac{P_i^3(\Phi_i(\tilde{M}))}{P_i^4(\Phi_i(\tilde{M}))} - c_i \right\|_2^2, \quad (8)$$

where Φ_i are the normalization functions. We leverage the Levenberg-Marquardt algorithm to perform the minimization of the non-linear cost function.

Triangulation requires estimating the correspondences from image features in two images. Since the satellite images are typically huge and contain millions of features, it is inappropriate to exhaustively compute the pairwise distance between every two features. Not only this is computationally expensive, but also the feature descriptor is less distinctive in presence of millions of features. To tackle this problem, we use the observation that the epipolar curve typically covers a small region of the satellite image. For the feature point shown in Fig. 3a, we only search within the white

rectangle for correspondence (Fig. 3b). The white rectangle is determined by the two end points of the epipolar curve, which is readily computed using the inverse mapping.

6. Experiments

To evaluate inverse mapping and triangulation, we use both synthetic data and real data for quantitative and qualitative evaluations. Both solvers are implemented using C++ with double-precision arithmetic. To evaluate the numerical errors, we also implement the triangulation method with 64-bit significand (mantissa) as opposed to 53-bit for double precision. The running time on Intel Xeon E5-2597 @2.70 GHz for each of the implementations are 0.064ms, 0.9ms and 164ms, respectively. Accordingly, the efficiency of our implementations with double arithmetic is sufficient to enable diverse application scenarios.

6.1. Inverse mapping

To quantitatively evaluate our proposed inverse RPC mapping estimation, we generate synthetic 3D points from a random uniform distribution within the cube of $[-1, 1] \times [-1, 1] \times [-1, 1]$. In order to yield the image coordinates, these 3D points are projected into virtual images I_1 and I_2 using their known RPCs to produce normalized image coordinates. To approximate realistic experimentation scenarios, we use four sets of real RPC model parameters extracted from two different satellites, including GeoEye-1 and Worldview-2. Each test is performed on 10,000 random samples for each set of RPC parameters. Error characteristics were stable across the four different sets of RPCs. Hence, quantitative results describe the performance over the aggregated sample of 40,000 tests.

Accuracy. Our proposed inverse mapping computes the normalized latitude and longitude of the 3D object point. The estimated normalized coordinates (X^*, Y^*) are com-

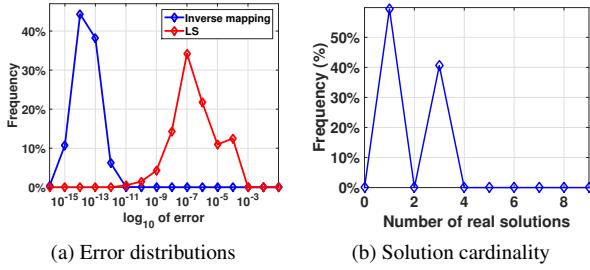


Figure 5: Our inverse mapping has $7.8 \cdot 10^{-14}$ median error. LS in Fig. 5a represents the method proposed by [31].

pared with the ground truth using the Euclidian distance $\|(\mathbf{X}^*, \mathbf{Y}^*) - (\mathbf{X}^g, \mathbf{Y}^g)\|_2$, where the superscript g denotes the ground truth. The resulting error distribution is shown in Fig. 5a. As can be seen from Fig. 5a, our proposed algorithm typically has errors of less than 10^{-11} , which is well below the demands of common applications.

To compare with one of the state-of-the-art methods in estimating the inverse mapping, we implemented the method of Yang [31]. Their method uses the inverse RPC model in Eq. (5), and given the ground truth data between 3D and 2D correspondences, the parameters of the model is estimated by fitting the data through least squares estimation. To this end, we used ground truth synthetic data for parameter estimation. We note that estimation errors above a magnitude of 10^{-4} correspond to an error of one meter in the real object space for a typical satellite image that covers 10 kilometers in one dimension. Accordingly, Fig. 5a illustrates how our proposed algorithm significantly outperforms the method in [31] (LS).

Furthermore, we apply another error measure that intuitively shows the superiority of our method over [31]. A 2D image point with an altitude is inversely mapped into the object space and then projected back onto the same image to generate a new 2D point. The error is defined as the Euclidean distance in pixels between the two points, which ideally should equal to zero. For each pixel across the whole image, we try 100 normalized altitudes randomly sampled within the valid range $[-1, 1]$, and report the average error over the 100 trials. Fig. 6 shows the error distribution across a image of resolution 7168×9216 . We can see our method has low and uniform errors across the whole image, while the method by [31] generates prominent and structured errors (larger than 2 pixels for some regions).

Solution Cardinality. Depending on RPC model parameters and the values of the specific estimation instance, we observed in our experiments either one or three real valued solutions. Fig. 5b illustrates the distribution of the number of real valued solutions. It can be seen that our attained solutions are predominantly unique. Moreover, there is always only one solution with both of the estimated values

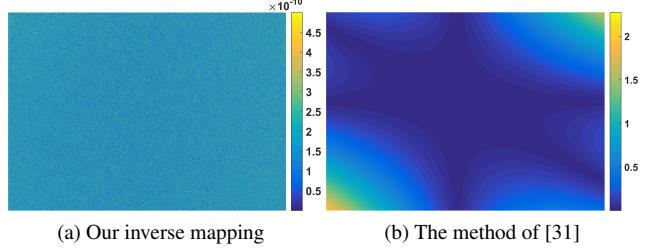


Figure 6: The error distribution of inverse mapping across the whole image.

X^* and Y^* within the valid range of $[-1, 1]$. This behavior follows from the geometric intuition that two 3D points with the same altitude in the real world should not be projected onto the same pixel.

6.2. Triangulation

Accuracy. We use the coordinate-wise components $\{\tilde{r}_1, \tilde{c}_1, \tilde{r}_2\}$ of a match positioning within the satellite imagery to estimate the 3D point by the triangulation solver, and project onto I_2 to compute the estimated column, denoted as \tilde{c}_2^* . The projection error is defined as $|\tilde{c}_2 - \tilde{c}_2^*|$. The triangulation error is defined as the Euclidian distance between the refined 3D point (Eq. (8)) and the ground truth. The error distributions in Figs. 7a and 7b show both of the projection errors and triangulation errors are very small. Moreover, we find that the accuracy of triangulation to be inferior to that of our estimated inverse mapping. We attribute this behavior to the numerical errors from the G-J elimination of the much larger elimination template. To validate this assumption, we also implement the solver in high-precision floating point arithmetic with 64 bits of significand (mantissa), in addition to the double precision implementation (53-bit mantissa). The results in Figs. 7a and 7b support numerical precision to be a significant component of the estimation error associated with our framework.

Stability. In practice our estimation process is faced both with measurement errors and/or potential mismatches in the 2D correspondences. To test the stability against measurement noise, we additively perturb input coordinate measurements with zero-mean Gaussian noise with different standard deviations. Fig. 7c shows the projection error across different noise levels. From Fig. 7c, it can be seen if the Gaussian noise has deviation of 1 pixel, the 90th percentile of the projection error is around 1.8 pixels. The projection error is used for outlier correspondences rejection, so we can optionally choose 2 pixels as the threshold for inlier testing. We note that the estimated triangulation error has real scale. Moreover, for the RPC parameters being used, each pixel corresponds to 0.5 meter in the object

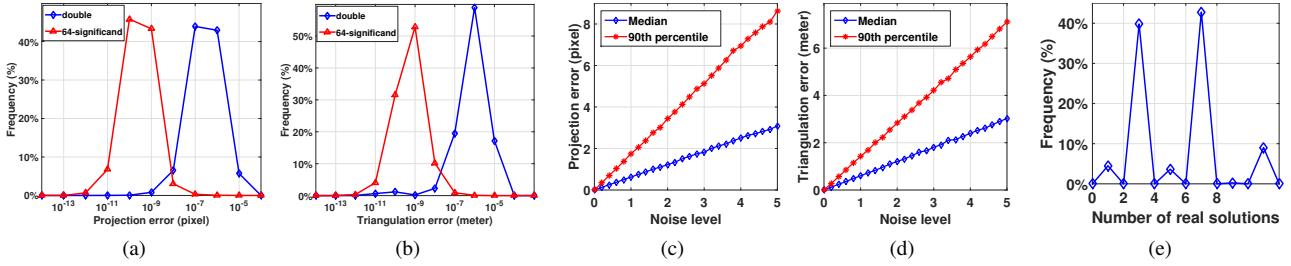


Figure 7: (a, b) Errors on noise-free data. (c, d) Median and 90th percentile errors on noisy data. (e) Solution cardinality.

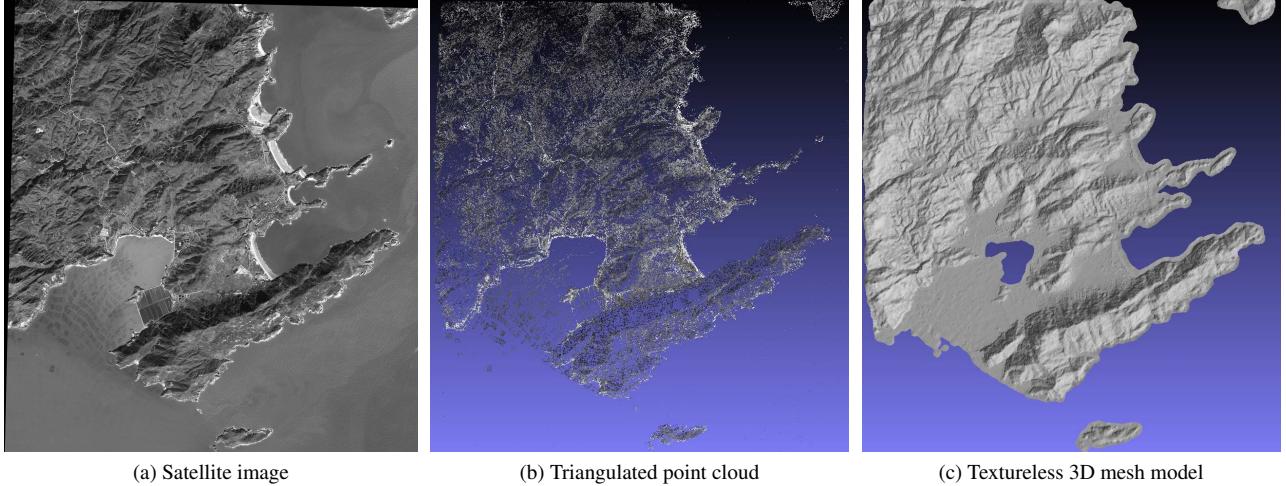


Figure 8: Triangulation results on the real satellite images. The images have a pixel resolution of 20460×20460 .

space. Given that, the triangulation error shown in Fig. 7d have reasonably small errors. We also test the implementation with 64-bit significand in presence of noise, but it has little benefit in errors, which means the measurement noise dominates the errors. Considering the efficiency vs. accuracy trade-off, the implementation with double precision should be used in most real applications. The number of real solutions is shown in Fig. 7e. There are typically multiple real solutions within the valid range, but only one of them has small projection error.

Real images. Although the synthetic data illustrates the correctness and effectiveness of our estimation methods, we also test the algorithm on a pair of real satellite images captured by the Worldview-2 satellite. To compute the 2D Correspondences, BRIEF [6] is used as local feature descriptor, and pairwise Hamming distance is used to measure feature similarity. One of the images used for stereo triangulation and the associated 3D point cloud are shown in Fig. 8a and Fig. 8b. We generate 3D mesh using Poisson surface reconstruction [21], where the point normal information required by Poisson surface reconstruction is set to $(0, 0, 1)$. Fig. 8c shows the textureless mesh model.

7. Conclusion

We propose two novel minimal solvers for satellite imagery based computer vision. The first minimal solver targets image based stereo estimation by establishing an accurate inverse mapping from image space to object space. Our evaluations demonstrate the advantage of the proposed solver over existing methods. The second minimal solver aims at 3D triangulation from satellite images. Based on this solver, we are able to propose a robust estimator and a refinement for increased accuracy of the triangulation.

Acknowledgement: Supported by the Intelligence Advance Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL), contract FA8650-12-C-7214. The U.S Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

References

- [1] The compendium of controlled extensions (ce) for the national imagery transmission format (nitf). National Imagery and Mapping Agency, November 2000. Version 2.1.
- [2] Astrium. *Pléiades Imagery User Guide*, v2.0 edition, October 2012.
- [3] M. Bujnák. *Algebraic solutions to absolute pose problems*. PhD thesis, Czech Technical University, 2012.
- [4] M. Byrod, K. Josephson, and K. Astrom. Fast and stable polynomial equation solving and its application to computer vision. *IJCV*, 2009.
- [5] F. Calakli, A. O. Ulusoy, M. I. Restrepo, J. L. Mundy, and G. Taubin. High Resolution Surface Reconstruction from Multi-view Aerial Imagery. In *3DIMPVT*, 2012.
- [6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*. 2010.
- [7] D. A. Cox, J. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e*. Springer-Verlag, 2007.
- [8] P. d’Angelo and G. Kuschk. Dense multi-view stereo from satellite imagery. In *IGARSS*, 2012.
- [9] G. Dial and J. Grodecki. Block adjustment with rational polynomial camera models. In *ISPRS*, 2002.
- [10] G. Dial and J. Grodecki. Satellite image block adjustment simulations with physical and rpc camera models. In *ISPRS*, pages 23–28, 2004.
- [11] G. Dial and J. Grodecki. Rpc replacement camera models. In *ISPRS*, volume 34, 2005.
- [12] S. Gehrke, K. Morin, M. Downey, N. Boehrer, and T. Fuchs. Semi-global matching: An alternative to lidar for dsm generation. In *Proceedings of the 2010 Canadian Geomatics Conference and Symposium of Commission I*, 2010.
- [13] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. <http://www.math.uiuc.edu/Macaulay2/>.
- [14] R. Hartley and T. Saxena. The cubic rational polynomial camera model. In *Image Understanding Workshop*, volume 649, page 653, 1997.
- [15] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, Feb 2008.
- [16] H. Hirschmüller, M. Buder, and I. Ernst. Memory efficient semi-global matching. *ISPRS*, 3:371–376, 2012.
- [17] Y. Hu and C. Tao. Updating solutions of the rational function model using additional control information. *Photogrammetric engineering and remote sensing*, 2002.
- [18] Y. Hu, V. Tao, and A. Croitoru. Understanding the rational function model: methods and applications. *ISPRS*, 20:6, 2004.
- [19] T. Isambert, J.-P. Berroir, and I. Herlin. A multi-scale vector spline method for estimating the fluids motion on satellite images. In *ECCV*, pages 665–676. Springer, 2008.
- [20] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *CVPR*, pages 1–6. IEEE, 2007.
- [21] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 2013.
- [22] Z. Kúkelová, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV*, 2008.
- [23] Z. Kúkelová, T. Pajdla, and M. Bujnak. *Algebraic methods in computer vision*. PhD thesis, Czech Technical University, Prague, Czech republic, 2012.
- [24] W. Li, X. Li, Y. Wu, and Z. Hu. A novel framework for urban change detection using vhr satellite images. In *ICPR*, volume 2, pages 312–315. IEEE, 2006.
- [25] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *ECCV*, pages 210–223. Springer, 2010.
- [26] J. Oh, C. K. Toth, and D. A. Grejner-Brzezinska. Automatic georeferencing of aerial images using stereo high-resolution satellite images. *Photogrammetric Engineering & Remote Sensing*, 77(11):1157–1168, 2011.
- [27] H. Stewénius. *Gröbner basis methods for minimal problems in computer vision*. Citeseer, 2005.
- [28] C. V. Tao and Y. Hu. A comprehensive study of the rational function model for photogrammetric processing. *Photogrammetric Engineering and Remote Sensing*, 67(12):1347–1358, 2001.
- [29] C. V. Tao and H. Yong. 3d reconstruction methods based on the rational function model. *Photogrammetric engineering and remote sensing*, 68(7):705–714, 2002.
- [30] Y. Wang, K. Wang, E. Dunn, and J.-M. Frahm. Stereo under sequential optimal sampling: A statistical analysis framework for search space reduction. In *CVPR*, pages 485–492. IEEE, 2014.
- [31] X. Yang. Accuracy of rational function approximation in photogrammetry. In *ISPRS*, pages 22–26, 2000.