



Lunar Crater Identification in Digital Images

John A. Christian¹ · Harm Derksen² · Ryan Watkins³

Accepted: 2 September 2021 / Published online: 21 October 2021
© The Author(s) 2021

Abstract

It is often necessary to identify a pattern of observed craters in a single image of the lunar surface and without any prior knowledge of the camera's location. This so-called “lost-in-space” crater identification problem is common in both crater-based terrain relative navigation (TRN) and in automatic registration of scientific imagery. Past work on crater identification has largely been based on heuristic schemes, with poor performance outside of a narrowly defined operating regime (e.g., nadir pointing images, small search areas). This work provides the first mathematically rigorous treatment of the general crater identification problem. It is shown when it is (and when it is not) possible to recognize a pattern of elliptical crater rims in an image formed by perspective projection. For the cases when it is possible to recognize a pattern, descriptors are developed using invariant theory that provably capture all of the viewpoint invariant information. These descriptors may be pre-computed for known crater patterns and placed in a searchable index for fast recognition. New techniques are also developed for computing pose from crater rim observations and for evaluating crater rim correspondences. These techniques are demonstrated on both synthetic and real images.

Keywords Spacecraft navigation · Invariant theory · Lunar exploration · Terrain relative navigation · Craters · Conics · Projective geometry · Pattern recognition · HEALPix

✉ John A. Christian
john.a.christian@gatech.edu

Harm Derksen
ha.derksen@northeastern.edu

Ryan Watkins
ryan.n.watkins@nasa.gov

¹ Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

² Department of Mathematics, Northeastern University, Boston, MA 02115, USA

³ Arctic Slope Regional Corporation Federal, Arlington, VA 22202, USA

1 Introduction

Future lunar exploration missions are expected to rely on optical measurements (e.g., images from a camera) to navigate independently of Earth-based operators. Although the use of images for spacecraft navigation—called optical navigation (OPNAV)—is a well-established practice for conventionally navigated spacecraft [108, 109], autonomous on-board OPNAV remains an emerging technology. Recent decades have witnessed great technological advancement and expanding acceptance of autonomous vision-based navigation for the exploration of other celestial bodies (e.g., Moon, Mars, asteroids). These advancements in real-time, on-board OPNAV are exemplified by the technological progression from simple estimation of lander velocity with the Mars Exploration Rover's DIMES system in 2004 [19] to autonomous feature tracking that will soon be demonstrated on the OSIRIS-REx mission to asteroid Bennu [105] and during landing of the Mars Perseverance (Mars 2020) rover [64].

This work focuses on OPNAV for lunar missions using digital images captured by a conventional camera (images from a pushbroom camera [47] are a different problem and are not discussed here). Images from a conventional camera follow the geometry of perspective projection. How to best use such images of the Moon for navigation depends on many factors—with distance and lighting often being the two most important considerations. It is usually best to use horizon-based OPNAV [21, 26, 108] when far away from the Moon and when a large portion of the lunar disk is contained within the image. As the vehicle gets closer to the Moon (e.g., low lunar orbit, lunar lander descent), it becomes more appropriate to observe specific landmarks on the lunar surface. These landmarks could be specific types of morphological features (e.g., craters) or a patch of unique-looking terrain.

The principal difficulty with landmark-based OPNAV is the need to match points in an image to points in an onboard map. This has led to the continued use of horizon-based OPNAV at close distances where it would otherwise be better to use surface features. It has also led to navigation with unknown landmarks [8, 23, 80, 137] or visual odometry [27]. Unfortunately, horizon-based OPNAV is not always practical (especially at close ranges) and unknown landmarks do not generally provide full state observability. Consequently, the ability to match surface features to a map is a required capability for autonomous vision-based navigation for many missions.

Many landmark-based OPNAV algorithms rely on the real-time rendering of an onboard digital elevation map (DEM) [3, 44, 64, 105]. The standard approach is to render the expected appearance of landmark patches and then compare this to regions of the navigation image, usually by means of a 2D cross-correlation. This, however, requires the vehicle to carry onboard 3D models for each landmark patch (and, sometimes, complete DEMs for certain portions of the body) and also have the ability to render these patches in real-time. Further, the rendering step requires *a priori* state knowledge, thus precluding this technique from supporting lost-in-space landmark-based OPNAV. More troubling, however, is that the rendering-based template matching approach feeds navigation state data into the measurement generation process, creating scenarios where corrupted navigation states render otherwise

good images ineffectual—potentially leading to unnecessary filter reinitializations, trajectory aborts (e.g., during lunar descent), or other undesirable events.

Landmark-based OPNAV using craters [18, 49, 86, 110] is an alternative strategy that, when properly implemented, does not necessarily require *a priori* state data. Furthermore, OPNAV with craters does not require onboard DEMs or an onboard rendering capability. Instead, such systems require an image processing algorithm to detect craters and a pattern matching algorithm to match these observations to a catalog of known craters. This manuscript focuses on the second part (crater pattern matching) of this problem.

There are two general classes of crater matching algorithms: (1) tracking and (2) lost-in-space. In the case of crater tracking, it is possible to use *a priori* state information to predict where catalog craters should appear in an image. These predictions are used to associate observed craters with model craters through a variety of schemes of varying sophistication. We generally find that explicit crater matching is undesirable and more robust tracking performance is achievable using probabilistic techniques (as exemplified by anonymous feature tracking (AFT) [88]). In the case of lost-in-space crater matching, no *a priori* state information is available. Lost-in-space algorithms, therefore, must be able to recognize a crater pattern from its appearance in an image regardless of camera pose (i.e., camera position and attitude). Such a capability is necessary for filter (re)initialization. It is also useful in providing state-independent measurements (if required) or a state-independent verification of crater tracking correspondences. This work focuses on solving the lost-in-space crater identification problem.

We briefly note that, while autonomous spacecraft navigation was the original motivation for this work, crater pattern recognition is equally useful for the registration of scientific images [138].

In this manuscript, we provide a comprehensive treatment of lunar crater pattern recognition in digital images formed by a conventional camera (i.e., under perspective projection). We begin by outlining the philosophical framework for such a system (Section 2) and then review important background material (Section 3). This is followed by the detailed mathematical development in Sections 4–9. Numerical results and examples are shown in Section 10. This work contains a number of key results, which are summarized here:

1. Almost all lunar impact craters are elliptical, and many are nearly circular (Section 3.1). Rather than being a heuristic design choice for easy crater identification, we discuss why craters are necessarily elliptical in shape from a perspective of planetary science and impact mechanics. Recent advancements in our knowledge of the lunar crater population suggests that a circular crater assumption (common amongst crater identification algorithms) is not well-supported by the data for small craters (diameter less than about 30 km).
2. Invariant theory is the proper mathematical framework for recognizing a pattern of crater rims in an image (Section 5). Elliptical craters on the lunar surface

project to elliptical features in an image (Section 4). For some patterns of craters, there exist algebraic quantities that remain unchanged (i.e., invariant) regardless of the camera pose and that may be computed from just the projected crater rims in a single image. Such quantities we call *projective invariants*. Invariant theory provides the means by which we may determine the existence (or not) of these projective invariants.

3. There are no projective invariants for arbitrarily placed conics (e.g., elliptical crater rims) on the surface of an arbitrarily shaped body. We provide the first known proof of this fact (Section 5.4.3). This result is significant, as it produces a theoretical obstacle for solving the lost-in-space crater identification problem about very irregular bodies, such as some asteroids and comets. Fortunately, the Moon is not an arbitrarily shaped body.
4. Projective invariants exist for arbitrarily placed conics (e.g., elliptical crater rims) lying on a nondegenerate quadric surface (e.g., sphere, ellipsoid), which is an excellent approximation for the shape of the Moon. A different set of invariants exists for conics lying on a common plane. The existence of invariants for conics on a nondegenerate quadric surface is a novel result (Section 5.5), while invariants for conics on a plane is a known result (Section 5.2). Since the Moon is nearly spherical on the regional/global level and nearly planar on the local level, projective invariants generally exist for craters lying on the surface of the Moon. In both cases (global and local), we show how to efficiently compute these invariants from only the projected conics (i.e., the contour of the crater rim) that appear in a digital image (Section 6).
5. There are no algebraically independent invariants for conics (e.g., elliptical crater rims) on a nondegenerate quadric surface or on a plane beyond the ones developed in this work. All other invariants a researcher may conceive for one of these two cases may be written as an algebraic function of the invariants discussed herein. Thus, our invariants describe all the independent information about a crater rim pattern that is pose invariant and useful in constructing a descriptor that may be indexed. We provide the first known proof of this fact for the case of conics on a nondegenerate quadric surface (Section 6.1.3). Given these findings, we suggest future investigation of invariants for patterns of lunar crater rims be focused on ease of computation or numerical stability rather than attempting to extract additional independent information from the crater rim contours (since there is no additional independent information to be had that remains unchanged with camera viewpoint).
6. The projective invariants for a triad of lunar craters may be used to form a feature descriptor that is insensitive to camera viewpoint (Section 7). These descriptors may be computed for known crater patterns and stored in a searchable catalog (i.e., an index). Matches to an observed crater pattern in any image are found by simply finding the nearest neighbor for its descriptor in the index. In cases where it is desirable for the matching to be permutation sensitive, the descriptor is nothing more than the projective invariants concatenated into a

vector. Alternatively, in cases where we wish the matching to be insensitive to crater ordering, we apply invariant theory a second time to construct *projective and permutation (p^2) invariants*. This is a novel result for crater pattern descriptors.

7. Lunar crater identification is a multi-scale pattern recognition problem. Local, regional, and global crater patterns are built from individual craters of different (increasing) diameters and occur over different (increasing) extent on the lunar surface. For example, a 1 km crater may contribute to a local crater pattern but not a global crater pattern, while a 100 km crater may contribute to a global crater pattern but not a local crater pattern. To address this challenge, we present the first known use of Hierarchical Equal Area isoLatitude Pixelization (HEALPix) to construct a hierarchy of crater catalogs (Section 8).
8. A substantial amount of navigation information is contained within the shapes of projected crater rims in an image. Many existing crater identification algorithms, however, compute camera pose using only the center coordinates of the craters—thus ignoring a great deal of actionable information. We discuss how to compute the camera position using the contours of projected image conics, rather than just the conic center coordinates. Our solution is non-iterative and finds the camera position in the least squares sense when observing $d \geq 2$ craters (Section 9.1).
9. Crater match hypotheses are verified by comparing the observed crater rims to what is expected were the hypothesized match true. Most existing crater identification algorithms perform this verification by comparing only the coordinates of crater centers, which generally requires ≥ 5 craters locations to agree. Instead, we develop a novel distance metric as a measure of how dissimilar two ellipse contours are from one another (Section 9.2) and use this to compare the observed and expected crater rims. Our metrics satisfy the three classical axioms for a metric (minimality, symmetry, triangle inequality) along with a fourth requirement of similarity invariance. Using the entire crater rim permits pattern verification with just the three craters used to perform the index match and does not require additional craters for verification. Thus, as compared to past methods, the new distance metric permits crater identification over more sparsely cratered regions of the Moon.
10. There is no single crater identification algorithm that is always best. The correct solution depends on orbital regime (or descent trajectory), camera specifications, on-board computational and memory resources, operational cadence, and other mission-specific parameters. There are critical crater identification design decisions to be made for the crater catalog (raw source of data), index scale (local, regional, global, or a combination), index organization, and invariant descriptor type (projective invariants or p^2 invariants). Therefore, rather than present a single crater identification algorithm, we suggest a *framework* solving this problem and provide the reader with tools for each step. Once understood, the elements in this manuscript may be used to identify craters for a wide

diversity of mission types. A few illustrative examples are provided to enhance understanding of how this might work.

Finally, we note that this work relies on mathematics and techniques that may be obscure to the average astrodynamist and spacecraft navigator. Therefore, we adopt a rather explanatory approach to facilitate understanding, dispel common misconceptions, and encourage adoption of the techniques herein.

2 A Framework for Lost-in-Space Terrain Relative Navigation (TRN)

The principal difficulty with recognizing 3D objects from their projection in a digital image is that the appearance of these objects changes with camera viewpoint and illumination geometry. The recognition of landmarks on the lunar surface inherits this generic computer vision difficulty. In many cases (including lunar landmark identification), we may circumvent this problem entirely by choosing to represent objects with a descriptor that remains the same (is *invariant*) regardless of camera viewpoint. Such an object recognition philosophy was influentially advocated for in [97] and [147], and we adopt this philosophy here.

There are a variety of object attributes that may remain unchanged with camera viewpoint and illumination geometry, including color, texture, and geometric structure. Finding these invariant attributes, however, requires a great deal of care. Of note, we know that no such invariant exists for an arbitrary patch of terrain with constant albedo and with an approximately Lambertian reflectance [17]. This suggests that terrain patches (or maplets)—as is often used in correlation-based TRN [3, 44, 64, 105]—may not be well-suited for lost-in-space TRN, though this is an important topic of future work. We also observe that popular image feature descriptors—such as SIFT [83], SURF [7], BRIEF [78], ORB [124], and others—do not describe the type of invariance required for lost-in-space TRN. They are, by construction, not the appropriate tool for the problem considered in this manuscript. There are two primary reasons for this. First, most of these descriptors are only formally invariant for a similarity transformation (translation, rotation, scaling), though many have been shown to be robust for a modest amount of affine shear. While there do exist feature descriptors that are formally affine invariant [92], it was convincingly argued by Lowe [83] that such invariance is often undesirable within the context of feature descriptors. Regardless, none of these descriptors are especially robust to extreme projective transformations—making them more appropriate for matching features between two similar images than for matching a single arbitrary image to a prebuilt database of features on a 3D object. The second (and more significant) problem with classical feature descriptors for lost-in-space TRN is that none of them are illumination geometry invariant, as discussed at length in [27]. In brief, all of these feature descriptors work by finding and describing regions of unique 2D patterns in an image. For TRN above an airless body (e.g., Moon, asteroid), the 2D pattern observed in an image is formed by the way that light is reflected off the terrain and towards the camera. When the lighting geometry changes, the 2D pattern in the image changes—thus the feature

descriptor for the same surface patch will change. Hence, the feature descriptors are not illumination invariant.

An alternative to landmark patches and feature descriptors is to look for explicit geometric features. Crater rims are one such geometric feature that can be easily recognized and consistently localized, even when observed from different camera viewpoints and under different lighting conditions. We find the geometry describing the pattern of multiple crater rims to be one of the most accessible attributes from which viewpoint invariant information may be constructed.

For any given object geometry, such as a pattern of crater rims, there may be a variety of viewpoint-invariant properties—but not all of these necessarily provide equal perceptual salience [63]. Thus, at a minimum, we seek non-constant invariants¹ that can effectively differentiate between many different crater rim patterns. Such non-trivial invariants do not always exist for craters lying on the surface of arbitrary 3D celestial bodies, though they do exist for a body such as the Moon. Even when invariants do exist, finding a way to compute them is not always straightforward. However, as we show in this manuscript, the additional effort to develop these invariants produces powerful pattern recognition capabilities.

The identification of perceptually salient invariants under the action of perspective projection (e.g., a digital image of the Moon collected with a conventional camera) allows us to quickly distinguish between two different objects (e.g., two different crater patterns) possessing two different invariant descriptors. Therefore, if we compute the invariants for a particular lunar crater pattern, these invariants may be concatenated into a descriptor for that pattern and stored in a catalog (or *index*) for future comparison. We repeat this procedure for each known crater pattern pre-flight and produce a very large index of patterns and their corresponding descriptors. Then, when an image is acquired in-flight, the invariants may be computed for an observed crater pattern and concatenated into the same type of pattern descriptor that we stored within the index. Therefore, the index may be queried for known patterns with descriptor values similar to that of the observed crater pattern. Efficient data structures allow such queries to be performed very fast, usually in $\mathcal{O}(\log n)$ time.

Matches from the index may be used to construct crater match hypotheses, which must be verified before acceptance. The primary means for verification is to use a hypothesized crater correspondence to compute the camera's location. This hypothesized camera location is used to reproject expected crater rims into the image, which are compared to the observed crater rims. If the reprojected pattern matches the observations, the match hypothesis is accepted. If not, the match hypothesis is rejected, and we attempt a different hypothesis.

This framework is summarized in Fig. 1. The remainder of this manuscript is dedicated to providing the details for each step of this process.

¹By non-constant or non-trivial invariants we mean functions that have the same value for different viewpoints of the same object, but whose value can change if the object being viewed changes.

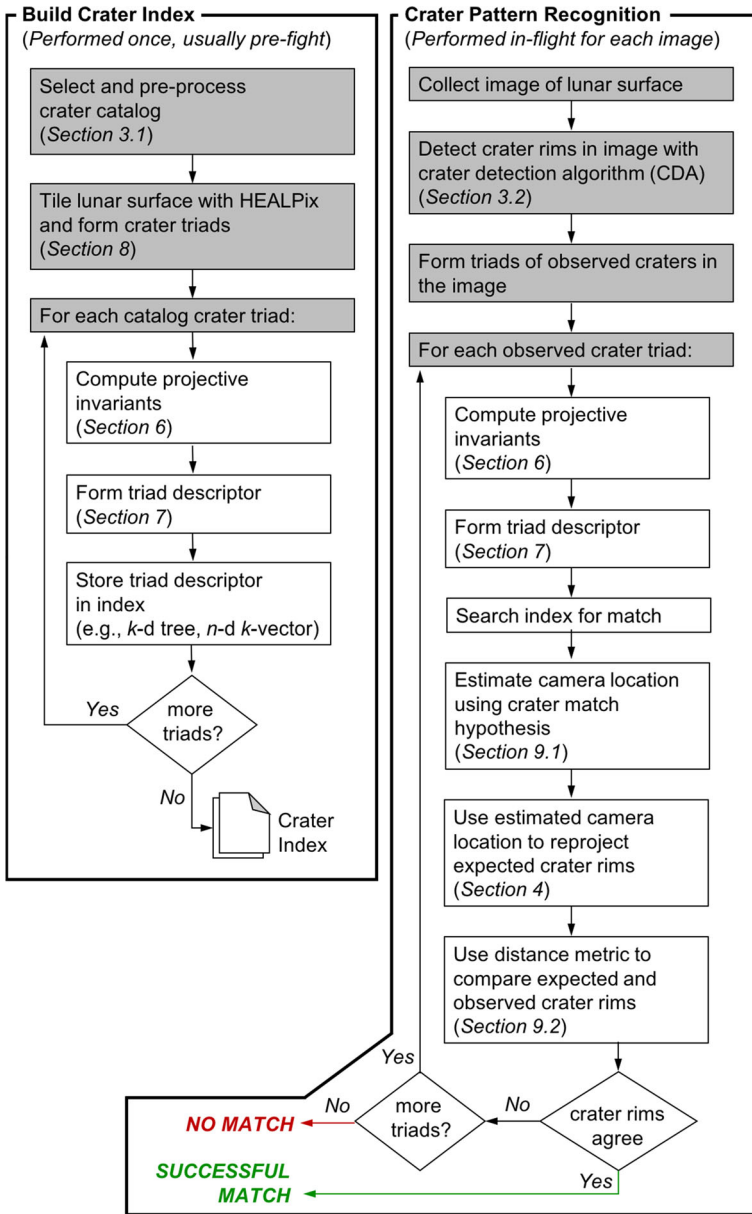


Fig. 1 The crater identification framework has two major components: index construction and pattern recognition. The index construction step (left) is often time-intensive since we must loop through all possible triads and then store them in an efficient data structure. The crater pattern recognition step (right) is fast and is performed for each image

3 Background

3.1 Observations on Lunar Crater Morphology and Size-Frequency Distribution

Lunar crater identification algorithms often begin with the assumption that the crater rims are either circular or elliptical in shape. It is also generally assumed that craters are well distributed about the Moon and that the crater catalog is static (new craters are not being added or removed). We find, however, that many recent crater identification algorithms are built on assumptions not supported by a modern understanding of the lunar crater population—and, therefore, are not likely to perform well across the full diversity of orbital regimes. A thorough understanding of lunar crater morphology and size-frequency distributions must precede the development of a computational algorithm to recognize patterns of these features.

3.1.1 Crater Catalogs

To recognize crater patterns in images of the lunar surface, we must first know where the craters are located and how they are shaped. Such data is contained in lunar crater catalogs. In most cases, however, these catalogs were built for planetary science purposes and not for image registration purposes—with the specific scientific question motivating the catalog construction often influencing the contents of the catalog. This must be considered before repurposing a catalog for spacecraft navigation, since not all lunar crater catalogs are appropriate for building an index for crater pattern identification.

There are a variety of lunar crater catalogs available; e.g., [52, 112, 119], and others [126, 140]. These catalogs were created with differing scientific objectives and from different raw data sets—thus, there should be no expectation of a one-to-one correspondence between their entries. For example, the minimum crater diameter is different for each catalog (and some catalogs even focus on very specific size ranges). As another example, some catalogs include all crater-like objects (e.g., [119]), while others only include craters the authors are confident originated from an impact event (e.g., [112]). Moreover, some catalogs attempt to only count the craters produced by direct impacts, thus ignoring those craters thought to be produced by secondary impacts (i.e., ejecta from the primary impact). This highlights the importance of selecting the crater catalog that best matches a particular mission's crater identification needs.

Crater catalogs are a useful tool for studying the statistical properties of the entire lunar crater population. To obtain global and comprehensive coverage, we usually exchange the detailed understanding of individual craters for a general understanding of all craters. Clearly, a detailed study of any single crater would yield a more sophisticated understanding of that crater—and likely produce crater information (e.g., size, shape, depth) that differs slightly from its corresponding entry in a global crater catalog.

The most comprehensive crater catalog to date is that of [119], containing about 1.3 million lunar impact craters having a diameter larger than about 1–2 km. This is the database we choose to use in this work.

3.1.2 The Elliptical Crater Assumption

The natural shape of an impact crater is an ellipse. Almost all lunar craters are nearly elliptical, and many craters are nearly circular [14]. The same is true for crater populations on other large celestial bodies [54]. Clearly, no crater rim is a perfect ellipse or perfect circle and this is only an approximation of the general shape.

There are a variety of factors that determine the size and shape of an impact crater. With regards to the shape of lunar crater rims, the most important factors are the impactor velocity, impactor size, and lunar surface properties. Lower impact angles, ϕ (see Fig. 2), tend to produce craters of higher ellipticity—where we define *ellipticity*, $\epsilon = a/b \geq 1$, as the ratio of crater rim semi-major to semi-minor axis. The sensitivity of the crater ellipticity to this impact angle is a function of the *cratering efficiency* (the ratio of crater diameter to impactor diameter) [14, 39], which acts as a model surrogate for impactor size and lunar surface properties. An empirical fit to hydrocode simulations and laboratory experiments shows excellent agreement between the impact angle at which $\epsilon = 1.1$ occurs ($\phi_{\epsilon=1.1}$) and a simple power law relation [39],

$$\phi_{\epsilon=1.1} = (90 \text{ deg}) \left(\frac{D_{90}}{L} \right)^{-0.82} \tag{1}$$

where D_{90} is the crater diameter for a vertical impact and L is the impactor diameter (i.e., D_{90}/L is the cratering efficiency for a vertical impact). Thus, an impact with $\phi < \phi_{\epsilon=1.1}$ will produce a crater with $\epsilon > 1.1$. The cutoff angle $\phi_{\epsilon=1.1}$ is generally between 3–5 deg (for sand, $D_{90}/L \sim 60$) and 30 deg (for brittle rock, $D_{90}/L \sim 4$) [29, 39, 91].

Given the rather oblique impact angle required for an elliptical crater, it follows that a large percentage of lunar craters should be nearly circular. For an airless body such as the Moon, the cumulative distribution function (CDF) for the impact angle when the impactors are assumed to arrive from random directions is [130]

$$p(\phi \leq \phi_0) = \sin^2 \phi_0 \tag{2}$$

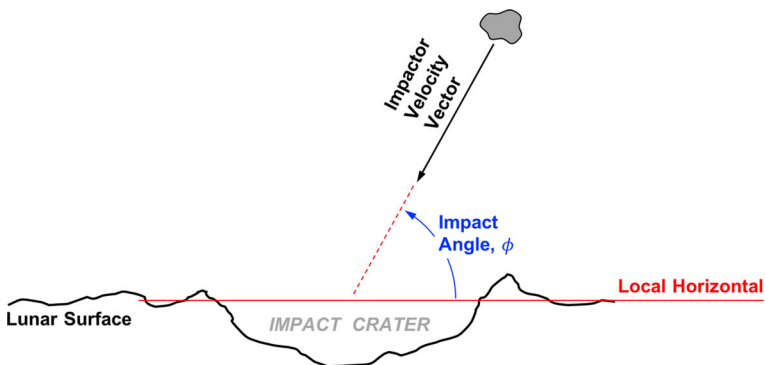


Fig. 2 Geometry (side view) of impact crater creation, with the impact angle being defined as the angle above the local horizontal

This model suggests that about 7.6% of the crater population should have an impact angle less than 5 deg and that about 25% of the crater population should have an impact angle less than 30 deg. The database of [119], however, found a higher percentage of elliptical craters than might otherwise be expected and proposed a few hypotheses for this observation. Of particular note is that Robbins’s database includes both primary and secondary impact craters, whereas many ellipticity studies (e.g., [14]) only consider primary craters.

Regardless of the underlying physical cause, the newly available global distribution of crater ellipticity from [119] makes clear that a circular crater assumption is not valid at the local level (i.e., when viewing craters of diameter smaller than about 30 km). This finding is significant, as many recent crater identification algorithms assume a circular crater model—a choice that is not well supported by the data. To see this, consider the ellipticity distribution as a function of crater size as shown in Fig. 3. We immediately see that over half of the entire crater population has an ellipticity greater than 1.1 (the threshold planetary scientists usually use to define an “elliptical” crater [14, 39]). Many of these craters, however, are not morphologically well preserved—and, therefore, are not good candidates for crater-based navigation since they will not produce good ellipse fits. If we restrict ourselves to well-defined craters whose catalog shape is supported by over 90% of its circumference, we see a substantially larger percentage of the craters are nearly circular, especially when considering craters larger than 30 km.

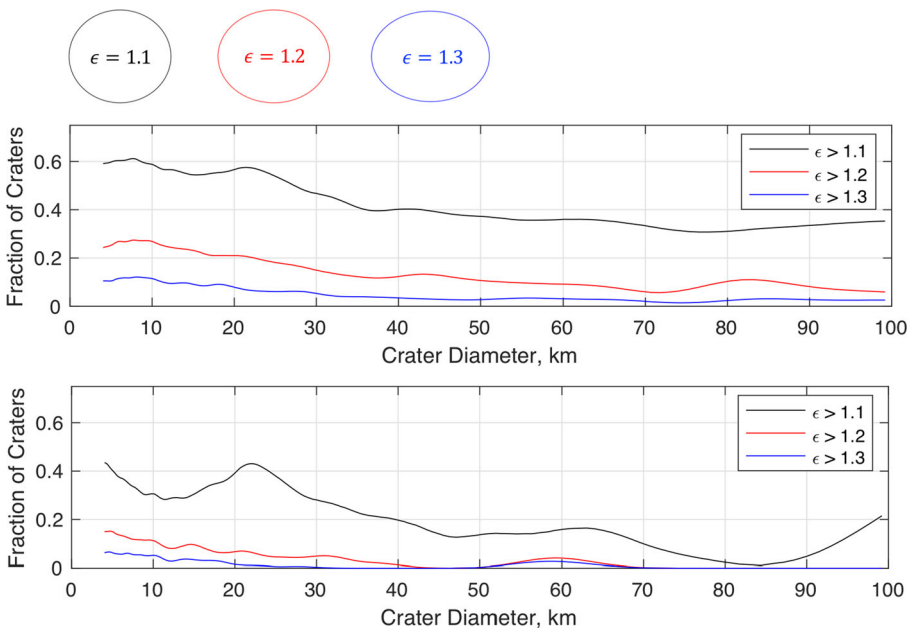


Fig. 3 Lunar craters with well-defined rims tend to be less elliptical than the general population of craters. Plots show the fraction of craters above a specified ellipticity for every catalog crater (top) and for only those with well-defined crater rims (bottom). A visualization of crater ellipticity is shown above the plots for intuition and context. Results are based on post-processing of catalog data from [119]

Topography or layered strength differences can create other irregular crater shapes [89], with a few examples shown in Fig. 4. On the Moon, simple craters ($\lesssim 10\text{--}20$ km in diameter) often have bowl-shaped morphologies, with small flat floors and few internal topographic features. Some may have deposits on the floor that originate from mass wasting or ponding of impact melt. The rims of simple craters are generally uplifted (unless significant degradation has occurred, as in the case of old simple craters), and the crater is surrounded by an ejecta blanket. More energetic impacts create larger, more complex craters. Complex craters ($\gtrsim 10\text{--}20$ km) have more varied morphologies, including central peaks, terraced walls, and flattened floors. Central peaks are formed by rebounding of material from deeply buried rocks, making these features especially interesting for studying materials excavated from lower in the crust. Larger complex craters can often have more than one central peak structure. Surrounding rocks in the wall of complex craters can also collapse, forming terrace-like structures. As impact features increase in size, they also increase in complexity. Craters larger than 300 km are usually classified as basins.

3.1.3 Lunar Crater Size-Frequency Distribution

Analyzing crater populations plays a major role in establishing planetary chronologies and in dating geologic units. There is general consensus that the lunar cratering projectile flux was relatively constant within the last 3 Gyr [102, 133], with the period before 3 Gyr experiencing variable impact rates as a result of the hypothesized late heavy bombardment.

Cumulative crater size-frequency distributions (SFDs), which describe the number of craters greater than a given diameter per measured area, are the usual data product used to estimate the age of planetary surfaces. Crater SFDs are conventionally determined by binning measured craters based on their diameters [46, 55], though modern practitioners are transitioning to more formal statistical approaches (e.g., using empirical density functions) [120]. These SFDs are often used to produce *production functions* that describe the rate at which new craters occur.

The most commonly used production function for estimating lunar surface ages is the Neukum Production Function (NPF) [101, 102]. The NPF models the formation

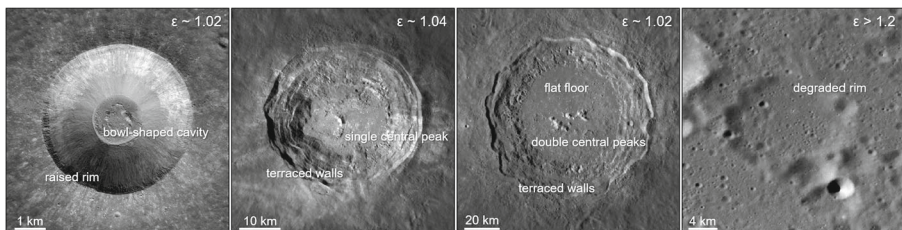


Fig. 4 Craters of different sizes and ages exhibit a variety of morphologies. (A) Lichtenburg B (33.3 N, 61.5 W), a 5 km simple bowl-shaped crater with a raised rim. (B) Aristarchus (23.7 N, 47.4 W), a 40 km complex crater with terraced walls and a single central peak. (C) Copernicus (9.62 N, 20.08 W), a 93 km complex crater with double central peaks, terraced walls, and a flat floor. (D) Riccius W (38.9 S, 25. 2 E), a 19 km degraded crater with an infilled cavity and much of the rim no longer clearly visible

rate of craters between 10 m and 300 km, and is generally used to estimate ages for geologic units on the Moon [102]. The NPF is shown in Fig. 5 and is given by

$$\log_{10}(N) = \sum_{k=0}^m a_k [\log_{10}(D)]^k \quad (3)$$

where D is the crater diameter, N is the number of craters of size D or smaller per unit area per unit time, and the coefficients a_k (for $m = 11$) are given in [102]. The NPF suggest that that we should expect the production of new craters on the lunar surface to occur at a rate of 1.7×10^{-5} new craters larger than 4 km per km^2/Gyr . For the Moon, this results in approximately 635 new craters per Gy (or about 1.7×10^{-9} craters/day). Assuming impacts follow a Poisson distribution, we may straightforwardly compute the probability of an impact creating a crater larger than 4 km during a specified period of time. If a crater catalog must be valid for a period of days to a few years, the probability of a new crater appearing is essentially zero. Thus, it is quite reasonable to assume a static crater catalog.

We note that recent work analyzing new impact craters as detected by the Lunar Reconnaissance Orbiter (LRO) Camera [133] found that the current cratering rate may potentially be higher than what the NPF predicts. Regardless, the expected discrepancies are small within the present context and the static crater catalog assumption remains valid.

3.2 Crater Detection

Successful crater detection is a prerequisite for crater identification. While not the focus of the present work, a brief review of crater detection algorithms (CDAs) is appropriate.

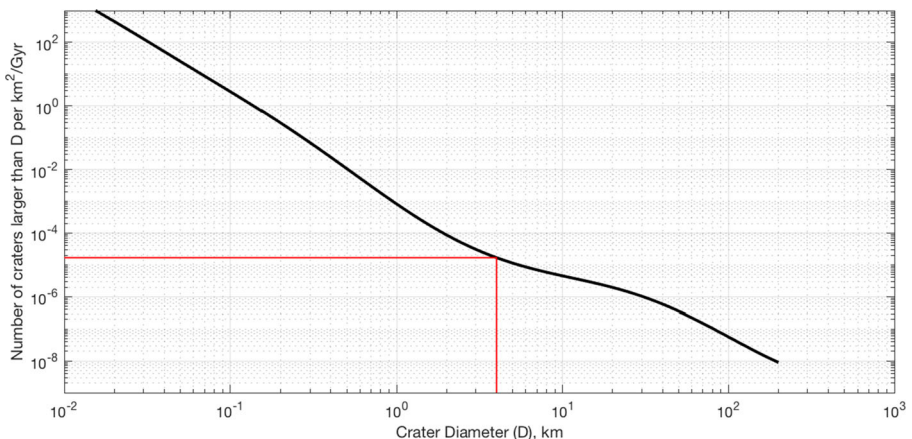


Fig. 5 The Neukum Production Function (NPF) describes the cumulative number of new lunar craters per km per Gyr below a specified diameter. The red line shows the NPF evaluated at a diameter of 4 km

There have been a number of recent surveys of CDAs for both planetary science [33, 125] and spacecraft navigation [142] applications. We observe here that the crater detection requirements for planetary science and navigation are often quite different—meaning algorithms developed for one application are not necessarily transferable to another application. There are two primary differences. First, from a scientific standpoint, the CDA objective is often to find all of the craters. For navigation, however, the objective is not to find every crater, but only enough to navigate. Any crater found that is not in the database becomes clutter that must be rejected. Thus, an algorithm that returns every crater is not necessarily better for navigation, and we are more interested in algorithms that preferentially return only well-modeled craters that exist in our on-board catalog. The second primary difference between scientific and navigation CDAs is suitability for real-time use. CDAs for autonomous navigation must be fast enough to run in real-time on a space-qualified computing platform and robust enough to not require human supervision. No such requirements are generally placed on CDAs for scientific analyses.

Of special note is the tremendous progress made in CDAs with machine learning over the last five years. Some notable algorithms include: DeepMoon² [131], Python crater detection algorithm (PyCDA)³ [69], LunaNet [37], CraterIDNet [139], and others [9, 33, 40].

With well over 100 different algorithms contained in the CDA surveys [33, 125, 142], we find no need to discuss the topic further here. The interested reader is directed to these surveys and the references therein.

3.3 Crater Identification

Lost-in-space crater identification from a single image is a pattern recognition problem. When challenged with an image containing a set of observed craters, the task is to find the corresponding craters from within a large catalog of known craters.

This problem is not new, and numerous attempts have been made in the past to recognize crater patterns. However, the lack of an existing invariant for non-coplanar ellipses has forced past work to make numerous simplifying assumptions or employ *ad hoc* recognition schemes. These restrictive assumptions are relaxed by the developments introduced in this work. We now review some of the most notable prior work in crater identification and discuss their drawbacks using insights from invariant theory. As the reader will see, miscounting of independent invariants is a pervasive problem in the crater identification literature—leading to descriptors that are incomplete, redundant (i.e., not all elements are independent), or both. This recurring problem is one of the principal motivations of the present work.

Hanak [49, 50] uses an adaptation of star identification algorithms [95, 127] to recognize crater triads—an interpretation of the crater identification problem that has become influential within the spacecraft navigation community. Hanak chooses to

²<https://github.com/silburt/DeepMoon>

³<https://github.com/AlliedToasters/PyCDA>

index the crater pattern using the triangle's interior angles, which is not an invariant under general perspective projection. Therefore, the approach of [49, 50] operationally restricts usability to a nadir pointing camera with a modest field-of-view, which restricts the crater pattern to a similarity transformation where inter-crater angles remain invariant. If the camera is not nadir pointing, or if the spacecraft is far away from the Moon and craters are not nearly coplanar, the descriptor is no longer pose invariant and matching performance may be poor. Once potential matches are made, the authors of [49] check these hypotheses against four additional metrics (three ratios of crater diameters to inter-crater distances, and a ± 1 flag indicating if the triangle is clockwise or counterclockwise going from shortest to longest leg), which are also not projective invariants. We observe, therefore, that the authors of [49, 50] develop a six element descriptor, with none of the elements being a projective invariant. We contrast this to the seven known invariants for three coplanar conics. It is possible, therefore, to build a seven-element descriptor (see Section 6.2.2) that holds more descriptive power than [49] and that is also a projective invariant.

Yu, et al., assume coplanar craters under affine transformation [145], and develop an invariant descriptor based on the ratio of crater areas. The ratio of areas of two closed coplanar curves as invariant for affine transformations is discussed in [41]. Identification using only area ratios, however, neglects the spatial relationships between the craters and significantly reduces the descriptiveness of these invariants. Like the work of Hanak [49], the methods from [145] are also limited to nearly nadir pointing images of nearly coplanar craters.

Cheng, et al., develop a crater identification scheme by matching pairs of coplanar craters [18, 20] using invariants from [41] and [115]. A pair of coplanar conics possess two unique invariants, which may be computed preflight and indexed into a catalog for in-flight matching. The descriptive power of only two conics, however, is low—so multiple pairs must be compared, which is straightforward to accomplish. Additionally, the constraint of using coplanar conics limits the flight regimes where this technique works well. Despite these minor drawbacks, we consider the work of [18, 20] to be the most theoretically informed lost-in-space crater identification approach developed to date.

Park, et al., consider a pattern of three coplanar craters [110] by looking at invariants of six coplanar points, where the six points are formed by the intersection of the triangle connecting the crater centers with the crater conics. These six coplanar points are split into six possible groupings of five points, from which the two unique invariants for each set of points may be computed using results from [41]. Then, for each set of points, these two invariants are made insensitive to point ordering using the projective and permutation invariants (also called p^2 invariants) from [76]. The pair of invariants for five coplanar points from [41] produce five p^2 invariants [76]. The authors of [110] concatenate the five p^2 invariants for each of the six sets of points to create a 30-element descriptor ($5 \times 6 = 30$) for the crater triad. However, as discussed in [76], only two of the five p^2 invariants for a set of five coplanar points are independent (which should be expected, since making the two invariants from [41] insensitive to point ordering can not add new invariants). Therefore, the six sets of five points are really described by vector of $6 \times 2 = 12$ invariants, such that

the 30-element descriptor from [110] has 18 redundant elements that do not enhance discriminative power. Moreover, the six sets of five points are not constructed from unique points (they are the six possible five-point combinations of the same six points), thus the remaining 12 invariants are not all independent. We observe that six coplanar image points possess only four independent projective invariants. We also observe that $d \geq 2$ coplanar conics possess only $5d - 8$ independent projective invariants (seven invariants for a triad of craters), as discussed in Section 5.2. Consequently, the 30-element descriptor of [110] has only four independent elements, 26 redundant (dependent) elements, and is missing three independent invariants. We note here that the seven invariants for three coplanar craters may be computed directly from the observed image conics as discussed in Section 6.2, without the need to compute sets of points. Likewise, invariants for clusters of four or more coplanar craters may also be computed directly [53]. The practical mechanics of computing such invariants are discussed in Section 6.2.2. If one wishes to build a permutation invariant descriptor (e.g., with p^2 invariants), we discuss how to do this for triads of craters in Section 7.

Maass, et al., propose an assortment of crater identification techniques [86], though we only discuss their lost-in-space technique here. They assume circular craters, thereby allowing explicit estimation of the crater normal in the camera frame (there are two possibilities [22, 129]). Given a set of three craters and their normals, the authors of [86] consider the $2^3 = 8$ possible configurations and choose the one where the normals are most similar (which presupposes a nearly planar configuration). More craters can be added to the pattern by iteratively solving a binary global optimization problem, with the cost function being the curvature energy of a cubic spline through a Delaunay triangulation of the crater centers. This resolves the 3D crater configuration (or reconstruction) to an unknown scale. Thus, Maass, et al., define a pattern descriptor using a pair of independent interior angles of a 3D crater triad, which we observe is a projective invariant since it is a reconstruction of the 3D geometry (and not the triangle in the image). Maass, et al., augment this check with the two independent ratios of crater radii. Pattern matches are verified by comparing the reprojected centers of additional craters. This usually requires two additional craters, thus requiring an image to possess five craters to yield a successful match. The method of [86] just described has three major drawbacks. First, we know from Section 3.1.2 that the circular crater assumption is often not valid, with 25%–50% of small craters having an ellipticity $\epsilon > 1.1$. This effects the ability to accurately estimate the surface normal direction (e.g., a nadir-pointing image of an elliptical crater with $\epsilon = 1.1$ would have a normal direction error over 20 deg if it was incorrectly assumed circular). Second, the two parameter pattern descriptor (or four parameters if you count the radii ratios, though these do not appear to be explicitly used in the search) are not maximally descriptive—we know there to be seven invariant parameters for a triad of nearly coplanar craters (see Sections 5.2 and 6.2.2). Thus, while the descriptor is indeed pose invariant (if the craters are circular), there are at least three independent pieces of information that remain for describing the crater pattern. Third, by verifying matches using only crater centers, more craters are required. Verifying using the crater rim reprojection (see Section 9.2) reduces the number of craters required.

There are a variety of other crater matching schemes [31, 68, 84], all of which make assumptions similar to the above methods.

This work is most similar to that of [18, 20], though we relax the requirement that craters be coplanar—thus enabling a substantial expansion of the orbital regimes where crater matching may be performed. We also allow for the simultaneous consideration of more craters by using triads instead of pairs.

4 Geometry of the Projection of a Single Crater

4.1 Mathematical Representation of a Crater

Since impact craters are known to be elliptical in shape, many lunar crater databases store the ellipse parameters for the best-fit ellipse to the rim of each cataloged crater. The most common database parameterization is the (1) latitude/longitude of the crater center $\{\varphi, \lambda\}$, (2) lengths of crater ellipse semimajor and semiminor axes $\{a, b\}$, and the orientation of the crater ellipse ψ (often measured counterclockwise from East). This is the parameterization used in the Robbins database [119].

In this work, we assume that a crater is a planar feature defined by the 2D elliptical curve that best describes the crater's rim. Circular craters are a special case ($a = b$) that is naturally handled without special treatment.

Let ξ_M be a three-dimensional (3D) point in the selenographic (i.e., Moon-centered, Moon-fixed) frame,

$$\xi_M = \begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix} \quad (4)$$

which may be represented in homogeneous coordinates as

$$\bar{\xi}_M = \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} \quad (5)$$

A crater is a planar feature, so let the plane \mathcal{P}_i containing crater i be described by the 4×1 vector π_i , such that

$$\mathcal{P}_i = \{\bar{\xi} \in \mathbb{P}^3 \mid \pi_i^T \bar{\xi} = 0\} \quad (6)$$

Here, \mathbb{P}^n denotes projective space, which is discussed in greater detail in Ref. [51] and Ref. [43].

4.1.1 Two-dimensional (2D) Crater Description

Within the plane \mathcal{P}_i , define a 2D coordinate system with origin at the ellipse center and aligned with the ellipse principal axes (see Fig. 6). In this case, we arrive at the canonical form of an ellipse, where the point $[x', y']$ lies on the ellipse if

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1 \tag{7}$$

where a is the ellipse semi-major axis and b is the ellipse semi-minor axis. Now, define another 2D coordinate system within \mathcal{P}_i with a different origin and different coordinate axis orientation (see Fig. 6). Let the center of the ellipse be located at $[x_c, y_c]$ and the ellipse principal axes be rotated by an angle ψ . A point $[x, y]$ in this coordinate system is related to a point $[x', y']$ by a Euclidian transformation,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} = \begin{bmatrix} (x - x_c) \cos \psi + (y - y_c) \sin \psi \\ -(x - x_c) \sin \psi + (y - y_c) \cos \psi \end{bmatrix} \tag{8}$$

Substituting this result into Eq. 7 yields the implicit equation

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + G = 0 \tag{9}$$

where

$$A = a^2 \sin^2 \psi + b^2 \cos^2 \psi \tag{10}$$

$$B = 2(b^2 - a^2) \cos \psi \sin \psi \tag{11}$$

$$C = a^2 \cos^2 \psi + b^2 \sin^2 \psi \tag{12}$$

$$D = -2Ax_c - By_c \tag{13}$$

$$E = -Bx_c - 2Cy_c \tag{14}$$

$$G = Ax_c^2 + Bx_c y_c + Cy_c^2 - a^2 b^2 \tag{15}$$

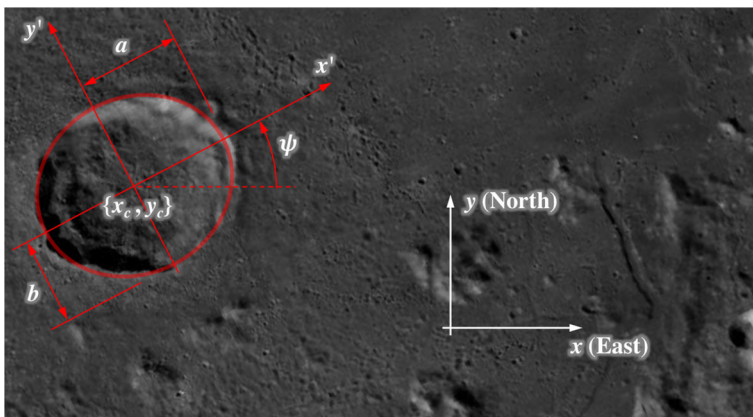


Fig. 6 Visualization of ellipse parameterization in \mathcal{P}_i

Observe that Eq. 9 is the generic equation for a conic, which happens to be an ellipse when $B^2 - 4AC < 0$. A simple calculation with Eqs. 10–12 shows that $B^2 - 4AC = -4a^2b^2$, which will always satisfy the ellipse inequality constraint when $a \geq b > 0$. The reverse mapping (implicit coefficients to explicit parameters) is straightforward and left to the reader (though the expression may be found in [25]).

The implicit equation from Eq. 9 may be written as a quadratic form using homogeneous coordinates. Therefore, letting a 2D point in the crater plane \mathcal{P}_i be given by

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{16}$$

which may be written in homogeneous coordinates as

$$\bar{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{17}$$

we observe that Eq. 9 may be rewritten as

$$\bar{\mathbf{x}}^T \mathbf{C} \bar{\mathbf{x}} = 0 \tag{18}$$

where

$$\mathbf{C} \propto \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & F/2 \\ D/2 & F/2 & G \end{bmatrix} \tag{19}$$

The expression in Eq. 18 describes a *conic locus*, which is the locus of 2D points forming the path of the conic (see Fig. 7). The matrix \mathbf{C} describing a conic is a 3×3 , real-valued, symmetric matrix of ambiguous scale. The ambiguous scale means that \mathbf{C} has only 5 degrees-of-freedom (which is also obvious from Eq. 9) and that \mathbf{C} and $k\mathbf{C}$ (with k being a real-valued, non-zero scalar) describe the same conic. If \mathbf{C} is full rank (as it is with an ellipse), we have a proper conic—otherwise it is degenerate

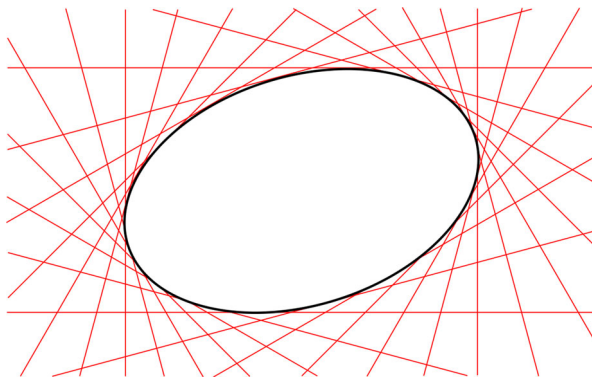


Fig. 7 Visualization of simple conic locus (black) and lines belonging to the conic envelope (red)

(either a pair of lines or a double line). Furthermore, if the conic is an ellipse, C is indefinite—always having two eigenvalues of one sign and a third eigenvalue of the opposite sign, with none of the eigenvalues being zero (since it is a proper conic).

Since C is full rank for an ellipse, $\det(C) \neq 0$ and

$$C^{-1} = \det(C)^{-1} C^* \propto C^* \tag{20}$$

where C^* is the adjugate matrix of C .

In projective geometry, points are dual to lines [51, 128]. A 2D point written in homogeneous coordinates (\bar{x}) lies on the line ℓ if it satisfies the constraint $\ell^T \bar{x} = 0$. Clearly, \bar{x} and ℓ can be exchanged in this relation—hence the point–line duality. To apply this duality property to conics, first observe that the line $\ell \propto C\bar{x}$ is tangent to the conic C if \bar{x} is a point on the conic [51]. For a proper conic, one may therefore compute $\bar{x} \propto C^{-1}\ell$. Substitution into Eq. 18 yields

$$\bar{x}^T C \bar{x} = (C^{-1}\ell)^T C (C^{-1}\ell) = \ell^T C^{-T} C C^{-1} \ell = 0 \tag{21}$$

and, since C is symmetric and $C^{-1} \propto C^*$,

$$\ell^T C^* \ell = 0 \tag{22}$$

where ℓ is a line tangent to the conic. The utility of using adjugate matrices instead of inverses will become readily apparent as we proceed.

The expression in Eq. 22 describes a *conic envelope*, which is the family of tangent lines that encapsulate the ellipse (see Fig. 7).

4.1.2 Three-dimensional (3D) Crater Description

The elliptical crater rim is a 2D feature in 3D space. Thus, as we will show, the crater catalog data may be used to construct both the crater’s plane and the disk quadric describing the 3D crater. The disk quadric is the natural and mathematically rigorous way to represent a 3D conic feature.

Consider catalog crater i with a geometric center at lunar latitude φ_i and longitude λ_i . The 3D selenographic position of the crater center is

$$p_{M_i}^{(c)} = \rho_i \begin{bmatrix} \cos \varphi_i \cos \lambda_i \\ \cos \varphi_i \sin \lambda_i \\ \sin \varphi_i \end{bmatrix} \tag{23}$$

where ρ_i is the distance of the plane \mathcal{P}_i from the center of the Moon. The crater’s center point is assumed to lie in the crater plane, $\pi_i^T \bar{p}_{M_i}^{(c)} = 0$.

The Moon is very nearly a sphere. Therefore, to an excellent approximation, the crater normal is parallel to the selenographic crater center location $p_{M_i}^{(c)}$. If the celestial body were an oblate spheroid, the local “up” direction may be computed through

an additional simple calculation [12, 15]. Regardless, assuming a spherical Moon, define a local East-North-Up (ENU) coordinate system, such that

$$\mathbf{u}_i = \frac{\mathbf{p}_{M_i}^{(c)}}{\|\mathbf{p}_{M_i}^{(c)}\|} \tag{24}$$

$$\mathbf{e}_i = \frac{\mathbf{k} \times \mathbf{u}_i}{\|\mathbf{k} \times \mathbf{u}_i\|} \tag{25}$$

$$\mathbf{n}_i = \frac{\mathbf{u}_i \times \mathbf{e}_i}{\|\mathbf{u}_i \times \mathbf{e}_i\|} \tag{26}$$

where $\mathbf{k}^T = [0 \ 0 \ 1]$ describes the selenographic description of the lunar pole. With the coordinate axes defined, construct the attitude transformation matrix (passive interpretation of a rotation [146]),

$$\mathbf{T}_M^{E_i} = [\mathbf{e}_i \ \mathbf{n}_i \ \mathbf{u}_i] \tag{27}$$

that transforms a vector expressed in the ENU frame of crater i (defined as E_i) to the same vector expressed in the selenographic frame M .

It is acknowledged that the ENU frame is poorly defined at the poles. In this case, we may replace (25) and (26) with a different convention. Any consistent and well-defined convention for defining the basis vectors of the local horizontal will work.

Since \mathbf{u}_i is normal to the plane and ρ_i is the perpendicular distance to the center of the Moon, the vector $\boldsymbol{\pi}_i$ describing the plane \mathcal{P}_i may be computed as

$$\boldsymbol{\pi}_i^T = [\mathbf{u}_i^T \ -\rho_i] \tag{28}$$

Now, consider a 3D point $\boldsymbol{\xi}_{E_i}$ lying on the rim of crater i as expressed in the crater’s ENU frame E_i . This 3D point may be transformed into selenographic coordinates,

$$\boldsymbol{\xi}_M = \mathbf{p}_{M_i}^{(c)} + \mathbf{T}_M^{E_i} \boldsymbol{\xi}_{E_i} \tag{29}$$

which may be compactly written in homogeneous coordinates

$$\bar{\boldsymbol{\xi}}_M = \begin{bmatrix} \mathbf{T}_M^{E_i} & \mathbf{p}_{M_i}^{(c)} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \bar{\boldsymbol{\xi}}_{E_i} \tag{30}$$

Since the point $\bar{\boldsymbol{\xi}}_{E_i}$ lies in the plane \mathcal{P}_i , it is evident that

$$\bar{\boldsymbol{\xi}}_{E_i} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \tag{31}$$

Thus, defining the columns of $T_M^{E_i}$ according to

$$T_M^{E_i} = [t_{1_i} \ t_{2_i} \ t_{3_i}] \tag{32}$$

we arrive at a relation between the 2D ellipse point \bar{x}_{E_i} in the crater plane [see Eq. 17] and its corresponding 3D point in the selenographic frame

$$\bar{\xi}_M = \begin{bmatrix} H_{M_i} \\ k^T \end{bmatrix} \bar{x}_{E_i} \tag{33}$$

where $k^T = [0 \ 0 \ 1]$ and H_{M_i} is the 3×3 matrix

$$H_{M_i} = [t_{1_i} \ t_{2_i} \ p_{M_i}^{(c)}] = [T_M^{E_i} S \ p_{M_i}^{(c)}] \tag{34}$$

and where S is the 3×2 matrix

$$S = \begin{bmatrix} I_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \tag{35}$$

Consider a quadratic surface in \mathbb{P}^3 (e.g., a sphere, an ellipsoid). Such surfaces are generally called *quadrics*, with a 3D point $\bar{\xi}$ lying on the surface if

$$\bar{\xi}^T Q \bar{\xi} = 0 \tag{36}$$

where Q is a 4×4 symmetric matrix describing the surface. This is the 3D analog to the 2D expression in Eq. 18.

The 3D surface defined by Eq. 36 is called a *quadric locus*. The 3D conic locus (a curve) cannot be represented by a quadric alone. Instead the conic defines a proper quadric cone (where Q is a 4×4 matrix of rank 3) and the 3D conic is formed by the intersection this quadric cone with the plane \mathcal{P}_i . Therefore, define the Moon-centered quadric cone (i.e., a cone with vertex at the center of the Moon) as

$$\mathcal{X}_i = \{\bar{\xi} \in \mathbb{P}^3 \mid \bar{\xi}^T Q_i \bar{\xi} = 0\} \tag{37}$$

such that the conic locus describing the elliptical crater is formed by the conic section

$$\mathcal{C}_i = \mathcal{X}_i \cap \mathcal{P}_i \tag{38}$$

Just as points and lines are dual in \mathbb{P}^2 , points and planes are dual in \mathbb{P}^3 (a fact that should be evident from Eq. 6). It follows therefore, that one may construct a dual quadric, Q_i^* , such that the *quadric envelope* is given by

$$\mathcal{D}_i^* = \{\pi \in \mathbb{P}^3 \mid \pi^T Q_i^* \pi = 0\} \tag{39}$$

where π is a plane tangent to the quadric. The quadric envelope provides a more natural way to describe a 3D conic.

The quadric envelope for a 3D conic is generally called the *disk quadric* [128], and defines all the planes tangent to the conic curve (see Fig. 8). This may be visualized as all the planes tangent to an ellipsoid $x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$ and letting $c \rightarrow 0$, thus resulting in a 3D disk (i.e., the 3D ellipsoid is collapsed to a plane and resembles a pancake or dinner plate). The disk quadric is, therefore, defined by a 4×4 matrix

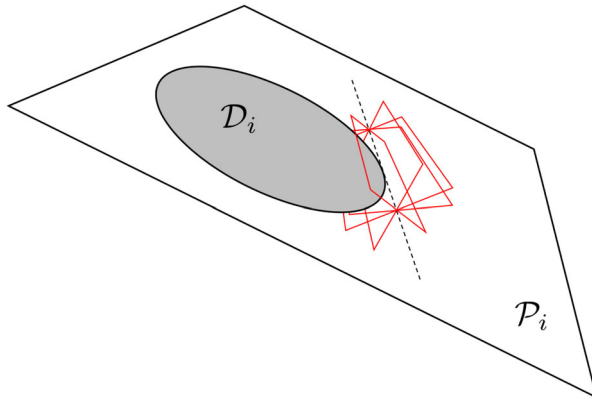


Fig. 8 Visualisation of the disk quadric that defines all of planes tangent to \mathcal{D}_i lying in the plane \mathcal{P}_i . A few example tangent planes are illustrated in red

\mathcal{Q}_i^* that has rank 3. We may compute the disk quadric directly from the crater’s conic envelope as

$$\mathcal{Q}_i^* \propto \begin{bmatrix} \mathbf{H}_{M_i} \\ \mathbf{k}^T \end{bmatrix} \mathbf{C}^* \begin{bmatrix} \mathbf{H}_{M_i} \\ \mathbf{k}^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{H}_{M_i} \mathbf{C}_i^* \mathbf{H}_{M_i}^T & \mathbf{H}_{M_i} \mathbf{C}_i^* \mathbf{k} \\ \mathbf{k}^T \mathbf{C}_i^* \mathbf{H}_{M_i}^T & \mathbf{k}^T \mathbf{C}_i^* \mathbf{k} \end{bmatrix} \tag{40}$$

which is a 4×4 symmetric matrix of rank 3.

The disk quadric is clearly a more direct way to represent the 3D conic than the intersection of a proper quadric cone and a plane. Thus, the disk quadric is the representation of choice when one must concisely describe a 3D conic, such as a crater located on the surface of the Moon.

Each 3D conic (as defined by the disk quadric) corresponds to a symmetric 4×4 matrix of rank 3, up to a scalar. The symmetric 4×4 matrices form a 10-dimensional vector space, so a nonzero 4×4 symmetric matrix corresponds to a point in \mathbb{P}^9 . The constraint that the rank is ≤ 3 defines a hypersurface in the projective space \mathbb{P}^9 given by the vanishing of the determinant. The set \mathcal{L} of all symmetric 4×4 matrices of rank 3 is an open subset in this hypersurface that parametrizes the set of conics. The dimension of \mathcal{L} (and the hypersurface) is $9 - 1 = 8$.

4.2 Homography and Action of a Projective Camera on a Crater Disk Quadric

If a crater rim is a planar conic, then its projection into an image is also a conic. The projective transformation from one plane to another is a homography, thus allowing the appearance of the projected crater ellipse to be computed analytically.

If \mathcal{C}_i is a 3D conic (see Eq. 38 and Table 1), we define its perspective projection into an image as $\mathcal{A}_i = \pi(\mathcal{C}_i)$, where $\pi : \mathbb{P}^3 \setminus \{\bar{\mathbf{r}}_M\} \rightarrow \mathbb{P}^2$ and $\bar{\mathbf{r}}_M$ is the camera location. Thus, we describe the conic locus for \mathcal{A}_i within the image as

$$\mathcal{A}_i = \{\bar{\mathbf{u}} \in \mathbb{P}^2 \mid \bar{\mathbf{u}}^T \mathbf{A}_i \bar{\mathbf{u}} = 0\} \tag{41}$$

Table 1 Summary of geometric primitives related to a 3D crater

Coordinate-Free Symbol	Homogeneous Coordinate Representation	Description
\mathcal{P}_i	$\boldsymbol{\pi}_i$	Plane containing crater rim
\mathcal{C}_i	\mathcal{C}_i	Conic locus of crater rim contour within \mathcal{P}_i
\mathcal{C}_i^*	\mathcal{C}_i^*	Conic envelope of crater rim contour within \mathcal{P}_i
\mathcal{X}_i	\mathcal{Q}_i	Quadric cone with apex at Moon center passing through \mathcal{C}_i
\mathcal{D}_i^*	\mathcal{Q}_i^*	Disk quadric describing 3D crater

where $\bar{\boldsymbol{u}}^T = [u \ v \ 1]$ is the homogeneous coordinate representation of the image pixel coordinate $[u, v]$. We now introduce a coordinate system to make this projection explicit.

Consider a calibrated camera with selenographic position \boldsymbol{r}_M and attitude \boldsymbol{T}_C^M that views lunar crater i . If $\boldsymbol{\xi}_M$ is the selenographic position of a point on the rim of crater i , then its location in the camera frame is simply

$$\boldsymbol{\xi}_C = \boldsymbol{T}_C^M (\boldsymbol{\xi}_M - \boldsymbol{r}_M) \tag{42}$$

As discussed in Ref. [26], we can construct the pinhole camera by letting $\bar{\boldsymbol{x}}_C \propto \boldsymbol{\xi}_C$, where $\bar{\boldsymbol{x}}_C$ is the projected image plane coordinate of the 3D point $\boldsymbol{\xi}_C$. With a digital camera, however, we never observe a point $\bar{\boldsymbol{x}}_C$, but instead we observe the 2D pixel coordinate $\boldsymbol{u}^T = [u \ v]$ (or $\bar{\boldsymbol{u}}^T = [u \ v \ 1]$) corresponding to that location in the image plane. Assuming the camera frame convention from Refs. [24] and [26], we place the $+z$ axis out of the camera and along the optical axis, the $+x$ axis to the image right and in the direction of increasing pixel column number, and the y axis completing the right-handed system. Furthermore, let the origin of the pixel u - v system be in the upper left-hand corner of the image, the u axis be to the right (increasing pixel column count), the v axis be down (increasing pixel row count), and integer values of $[u, v]$ occurring at the pixel centers. With these conventions, the conversion between image plane coordinates ($\bar{\boldsymbol{x}}_C$) and digital image pixel coordinates ($\bar{\boldsymbol{u}}$) is described by a simple affine transformation,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} d_x & \alpha & u_p \\ 0 & d_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_C \tag{43}$$

or, more compactly,

$$\bar{\boldsymbol{u}} = \boldsymbol{K}\bar{\boldsymbol{x}}_C \tag{44}$$

where the camera calibration matrix \boldsymbol{K} is a function of the five calibration parameters: d_x (and d_y) is the ratio of the focal length to pixel pitch in the x (or y) direction, α is

the detector skewness, and $[u_p, v_p]$ is the coordinates of the principal point (where the optical axis intersects the image). We generally have excellent knowledge of \mathbf{K} for spacecraft OPNAV applications from in-flight calibration with star field images [13, 24].

Therefore, the 3D point $\bar{\xi}_M$ projects to pixel coordinate $\bar{\mathbf{u}}$ in the image according to

$$\bar{\mathbf{u}} \propto \mathbf{P}_C^M \bar{\xi}_M \tag{45}$$

where \mathbf{P}_C^M is the camera projection matrix for a specified absolute pose,

$$\mathbf{P}_C^M = \mathbf{K} \begin{bmatrix} \mathbf{T}_C^M & -\mathbf{r}_C \end{bmatrix} = \mathbf{K} \mathbf{T}_C^M \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathbf{r}_M \end{bmatrix} \tag{46}$$

The action of a projective camera on a quadric envelope is known to follow [51]

$$\mathbf{A}_i^* \propto \mathbf{P}_C^M \mathbf{Q}_i^* \left(\mathbf{P}_C^M \right)^T \tag{47}$$

where \mathbf{Q}_i^* is from Eq. 40 and where \mathbf{A}_i (which may be computed from \mathbf{A}_i^*) describes the conic in the image plane tracing the apparent outline of the quadric. This allows for the analytic projection of the crater’s disk quadric into its apparent ellipse in the image plane.

The result of Eq. 47 may also be viewed as a homography. Returning to Eq. 45, substitute the result from Eq. 33 for $\bar{\xi}_M$,

$$\bar{\mathbf{u}} = \mathbf{K} \begin{bmatrix} \mathbf{T}_C^M & -\mathbf{r}_C \end{bmatrix} \begin{bmatrix} \mathbf{H}_{M_i} \\ \mathbf{k}^T \end{bmatrix} \bar{\mathbf{x}}_{E_i} \tag{48}$$

$$\bar{\mathbf{u}} = \mathbf{H}_{C_i} \bar{\mathbf{x}}_{E_i} \tag{49}$$

where we define \mathbf{H}_{C_i} to be the 3×3 matrix describing the homography between the crater plane \mathcal{P}_i and the camera’s image plane

$$\mathbf{H}_{C_i} = \mathbf{K} \begin{bmatrix} \mathbf{T}_C^M & -\mathbf{r}_C \end{bmatrix} \begin{bmatrix} \mathbf{H}_{M_i} \\ \mathbf{k}^T \end{bmatrix} = \mathbf{P}_C^M \begin{bmatrix} \mathbf{H}_{M_i} \\ \mathbf{k}^T \end{bmatrix} \tag{50}$$

Since \mathbf{H}_{C_i} is a homography, the conic locus and conic envelope may be analytically transformed to the image plane. Observing that

$$0 = \bar{\mathbf{x}}_{E_i}^T \mathbf{C}_i \bar{\mathbf{x}}_{E_i} = \left(\mathbf{H}_{C_i}^{-1} \bar{\mathbf{u}} \right)^T \mathbf{C}_i \left(\mathbf{H}_{C_i}^{-1} \bar{\mathbf{u}} \right) \tag{51}$$

$$= \bar{\mathbf{u}}^T \left(\mathbf{H}_{C_i}^{-T} \mathbf{C}_i \mathbf{H}_{C_i}^{-1} \right) \bar{\mathbf{u}} \tag{52}$$

$$= \bar{\mathbf{u}}^T \mathbf{A}_i \bar{\mathbf{u}} = 0 \tag{53}$$

we arrive at the result

$$\mathbf{A}_i \propto \mathbf{H}_{C_i}^{-T} \mathbf{C}_i \mathbf{H}_{C_i}^{-1} \tag{54}$$

$$\mathbf{H}_{C_i}^T \mathbf{A}_i \mathbf{H}_{C_i} \propto \mathbf{C}_i \tag{55}$$

It follows, therefore, that

$$A_i^* \propto H_{C_i} C_i^* H_{C_i}^T \tag{56}$$

$$H_{C_i}^{-1} A_i^* H_{C_i}^{-T} \propto C_i^* \tag{57}$$

The result of Eq. 56 is clearly the same as Eq. 47. We often find it convenient to make explicit the relative scales of A_i and C_i , which may be done by introducing the scale s_i . For example,

$$H_{C_i}^T A_i H_{C_i} = s_i C_i \tag{58}$$

5 Existence of Invariants from Projections of Crater Patterns

The central premise of this work is that invariant theory is the proper framework for describing, indexing, and matching patterns of lunar crater rims as seen in digital images. Recognizing that invariant theory will be unfamiliar to many space scientists and engineers, we find it worthwhile to develop the concept fully.

Much of the existing literature pursues an *ad hoc* approach for the construction of crater pattern descriptors. The lack of a formal framework has led to other authors proposing descriptors of varying dimension (ranging from two for a pair of craters [18, 20] to 30 for a triad of craters [110]). However, for a given type of crater pattern there is a specified number of independent projective invariants. Descriptors with fewer elements than this do not fully exploit the perceptually salient information in the image. Descriptors with more elements than this increase complexity of the matching process without adding any perceptually salient information (the redundant invariants are all algebraic functions of the independent invariants). Thus, *ad hoc* schemes for developing crater pattern descriptors generally provide suboptimal performance, either in terms of descriptive power or computational complexity.

5.1 Preliminaries on Invariant Theory

In invariant theory we study functions on a space that remain unchanged under a group of symmetries of that space. Invariant theory originated in the 19th century with pioneering work by Cayley, Clebsch, Gordan, Sylvester, Hilbert, and others. Groups of particular interest were the general linear group GL_n of invertible $n \times n$ matrices, the special linear group SL_n of $n \times n$ matrices with determinant 1, orthogonal groups, and finite groups (see [35, 36]). The space of binary forms of degree d (homogeneous polynomials in 2 variables of degree d) with the SL_2 -action described below was extensively studied in the 19th century [106]. If $a(x, y) = a_0x^d + a_1x^{d-1}y + \dots + a_dy^d$ is a binary form of degree d , then a matrix

$$\begin{bmatrix} p & q \\ r & s \end{bmatrix}$$

acts on it by

$$\begin{bmatrix} p & q \\ r & s \end{bmatrix} \cdot a(x, y) = a(px + ry, qx + sy) = a'_0x^d + a'_1x^{d-1}y + \dots + a'_dy^d,$$

where a'_0, \dots, a'_d are polynomial expressions in $a_0, a_1, \dots, a_d, p, q, r, s$. For $d = 2$, the discriminant $a_1^2 - 4a_0a_2$ is invariant under the SL_2 -action and every other polynomial invariant is a polynomial expression in the discriminant. Hilbert's papers [56, 57] on invariant theory form the foundation of modern algebraic geometry in which one studies algebraic varieties. An algebraic variety is a set that is defined by polynomial equations, and an algebraic group is an algebraic variety that also has a group structure. Over the 20th century, invariant theory has been generalized to an action of arbitrary algebraic group \mathcal{G} on an arbitrary algebraic variety \mathcal{V} .

5.1.1 Polynomial Versus Rational Invariants

Traditionally the focus has been on *polynomial* invariants; i.e., polynomial functions that remain unchanged under the group symmetries. In practice, we generally desire a set of *fundamental* invariants. That is, we seek a set of invariants f_1, f_2, \dots, f_r such that every other polynomial invariant g is a polynomial expression in the fundamental invariants: $g = G(f_1, f_2, \dots, f_r)$ for some polynomial $G(x_1, \dots, x_r)$ in r variables. In the language of commutative algebra, the set of all polynomial functions on the variety \mathcal{V} form a commutative ring, which we write as $\mathbb{C}[\mathcal{V}]$ (or $\mathbb{R}[\mathcal{V}]$ if we work over the real numbers). The subring generated by f_1, f_2, \dots, f_r is the set of all polynomial expressions in f_1, f_2, \dots, f_r , which we may write compactly as $\mathbb{C}[f_1, \dots, f_r]$. We also define the *invariant ring* as the set of all invariant polynomials, forming a subring denoted as $\mathbb{C}[\mathcal{V}]^{\mathcal{G}}$.

Now, observe that f_1, f_2, \dots, f_r are fundamental invariants if and only if $\mathbb{C}[\mathcal{V}]^{\mathcal{G}} = \mathbb{C}[f_1, f_2, \dots, f_r]$. Hilbert's Finiteness Theorem [56, 57] states that there is a finite list of fundamental invariants for groups such as GL_n , SL_n , and orthogonal groups. More generally, there is a finite system of fundamental invariants when the symmetry group is reductive [48, 99]. For a precise definition of reductive, see Ref. [61]. Not all groups are reductive though, and typically, translation groups may not be reductive. There are examples where there is no finite set of fundamental invariants [98]. In many situations the smallest possible number of fundamental polynomial invariants is finite but extremely large compared to the dimension of the space.

In practical applications, such as computer vision and spacecraft optical navigation, it may not be feasible to find a fundamental system of invariants. One remedy is to work with separating invariants (see [34, §2.4]), and it is even simpler to work with *rational invariants*. Thus, instead of dealing with polynomial functions, we will consider rational functions.

Let $\mathbb{C}(\mathcal{V})$ be the set of all rational functions on \mathcal{V} , and $\mathbb{C}(\mathcal{V})^{\mathcal{G}}$ be the set of all *invariant* rational functions on \mathcal{V} . We assume that \mathcal{V} is an irreducible variety so that $\mathbb{C}(\mathcal{V})$ has the algebraic structure of a field—specifically, it is the quotient field of the ring $\mathbb{C}[\mathcal{V}]$. We now see that $\mathbb{C}(\mathcal{V})^{\mathcal{G}}$ is a subfield of $\mathbb{C}(\mathcal{V})$. The reader should be warned that the quotient field of $\mathbb{C}[\mathcal{V}]^{\mathcal{G}}$ can be strictly smaller than $\mathbb{C}(\mathcal{V})^{\mathcal{G}}$. For example, it is possible that there are only constant polynomial invariants while there are non-constant rational invariants. This is another reason why it may be preferable to work with rational invariants instead of polynomial invariants.

Rational functions are not defined for all points in the space, but are defined for almost all points—which is good enough for practical crater identification. Invariant rational functions f_1, f_2, \dots, f_r form a system of fundamental rational invariants if every other rational invariant g is a rational expression in f_1, f_2, \dots, f_r ; i.e., of the form $G_1(f_1, \dots, f_r)/G_2(f_1, \dots, f_r)$ where $G_1(x_1, \dots, x_r)$ and $G_2(x_1, \dots, x_r)$ are polynomials with coefficients in \mathbb{C} and $G_2(f_1, \dots, f_r) \neq 0$. There always exists a finite system of fundamental *rational invariants* f_1, f_2, \dots, f_r . In fact, it is possible to choose $r \leq 1 + \dim \mathcal{V}$ (see Remark 1).

5.1.2 Real Versus Complex Rational Invariants

Though the problem of spacecraft optical navigation (and computer vision, more generally) deals with geometric configurations over the real numbers, we will often work over the complex numbers. There is not a big difference. In the typical setup, we have a real algebraic variety $\mathcal{V}_{\mathbb{R}}$ that parametrizes certain geometric configurations. For example, $\mathcal{V}_{\mathbb{R}}$ could be a subvariety of the real projective space $\mathbb{P}_{\mathbb{R}}^m$ defined by homogeneous polynomial equations with real coefficients. The same equations define also a projective variety $\mathcal{V}_{\mathbb{C}} \subset \mathbb{P}_{\mathbb{C}}^m$ in complex projective space if we view those equations over the complex numbers. We now assume that $\mathcal{V}_{\mathbb{R}}$ is Zariski dense in $\mathcal{V}_{\mathbb{C}}$, which means that any complex polynomial that vanishes on $\mathcal{V}_{\mathbb{R}}$ also vanishes on $\mathcal{V}_{\mathbb{C}}$. In this case, we can view the field $\mathbb{R}(\mathcal{V}_{\mathbb{R}})$ of real rational functions on $\mathcal{V}_{\mathbb{R}}$ as a subfield of $\mathbb{C}(\mathcal{V}_{\mathbb{C}})$. Moreover, the real and complex part of a rational function in $\mathbb{C}(\mathcal{V}_{\mathbb{C}})$ are rational functions on $\mathcal{V}_{\mathbb{R}}$. This gives a decomposition

$$\mathbb{C}(\mathcal{V}_{\mathbb{C}}) = \mathbb{R}(\mathcal{V}_{\mathbb{R}}) \oplus \mathbb{R}(\mathcal{V}_{\mathbb{R}})i,$$

where $i = \sqrt{-1}$. We also assume that $\mathcal{G}_{\mathbb{C}}$ is a complex algebraic group acting on the variety $\mathcal{V}_{\mathbb{C}}$, such that the subgroup $\mathcal{G}_{\mathbb{R}}$ of real group elements acts on $\mathcal{V}_{\mathbb{R}}$ and $\mathcal{G}_{\mathbb{R}}$ is Zariski dense in $\mathcal{G}_{\mathbb{C}}$. Then, because $\mathcal{G}_{\mathbb{R}} \subset \mathcal{G}_{\mathbb{C}}$ is Zariski dense, a complex rational function is invariant under $\mathcal{G}_{\mathbb{C}}$ if and only if it is invariant under the group $\mathcal{G}_{\mathbb{R}}$. Thus, we find that

$$\mathbb{C}(\mathcal{V}_{\mathbb{C}})^{\mathcal{G}_{\mathbb{C}}} = \mathbb{C}(\mathcal{V}_{\mathbb{C}})^{\mathcal{G}_{\mathbb{R}}} = (\mathbb{R}(\mathcal{V}_{\mathbb{R}}) \oplus \mathbb{R}(\mathcal{V}_{\mathbb{R}})i)^{\mathcal{G}_{\mathbb{R}}} = \mathbb{R}(\mathcal{V}_{\mathbb{R}})^{\mathcal{G}_{\mathbb{R}}} \oplus \mathbb{R}(\mathcal{V}_{\mathbb{R}})^{\mathcal{G}_{\mathbb{R}}}i.$$

In particular, if f_1, f_2, \dots, f_r are real rational invariant functions that generate the field $\mathbb{R}(\mathcal{V}_{\mathbb{R}})^{\mathcal{G}_{\mathbb{R}}}$ over \mathbb{R} (i.e., $\mathbb{R}(\mathcal{V}_{\mathbb{R}})^{\mathcal{G}_{\mathbb{R}}} = \mathbb{R}(f_1, f_2, \dots, f_r)$), then they also generate $\mathbb{C}(\mathcal{V}_{\mathbb{C}})^{\mathcal{G}_{\mathbb{C}}}$ over \mathbb{C} (i.e., $\mathbb{C}(\mathcal{V}_{\mathbb{C}})^{\mathcal{G}_{\mathbb{C}}} = \mathbb{C}(f_1, f_2, \dots, f_r)$).

On the other hand, if $\mathbb{C}(\mathcal{V}_{\mathbb{C}})^{\mathcal{G}_{\mathbb{C}}} = \mathbb{C}(f_1, f_2, \dots, f_r)$ for some *complex* rational invariants f_1, f_2, \dots, f_r , then the real and complex parts of f_1, f_2, \dots, f_r generate the field $\mathbb{R}(\mathcal{V}_{\mathbb{R}})^{\mathcal{G}_{\mathbb{R}}}$ of real rational invariants.

5.1.3 Counting Independent Rational Invariants

As we have seen, the proper counting of independent invariants is a recurring problem in the crater identification literature. Moreover, the usual method of counting invariants used in classical computer vision studies (e.g., a degrees-of-freedom approach) is not extensible to the problem at hand. We find it necessary, therefore, to develop

a more formal framework. This framework is shown to reproduce known results for simple cases, before being extended to the problem of non-coplanar conics.

Suppose that L is a field and K is a subfield. If $f_1, f_2, \dots, f_r \in L$, then $K(f_1, f_2, \dots, f_r)$ is the set of all rational expressions in f_1, \dots, f_r with coefficients in K . This is also called the field that is generated by f_1, f_2, \dots, f_r over K . So by definition, invariant rational functions f_1, f_2, \dots, f_r are fundamental rational invariants for the action of \mathcal{G} on \mathcal{V} if and only if $\mathbb{C}(f_1, f_2, \dots, f_r) = \mathbb{C}(\mathcal{V})^{\mathcal{G}}$.

In spacecraft navigation, computer vision, and other applications it is often desirable to know the maximum number of functionally independent invariants. This relates to the notion of transcendence degree of a field extension. If f_1, f_2, \dots, f_r lie in a field, then we say that f_1, f_2, \dots, f_r are algebraically dependent over a subfield K if there exists a nonzero polynomial P with coefficients in K with $P(f_1, f_2, \dots, f_r) = 0$. The transcendence degree of L over the subfield K is the supremum over all r for which there exists $f_1, f_2, \dots, f_r \in L$ that are algebraically independent over K . Let $\text{trdeg}(L/K)$ be the transcendence degree of L over K . If L is also a subfield of another field M , then we have the following relation (see [62, Theorem VI.1.11]):

$$\text{trdeg}(M/K) = \text{trdeg}(M/L) + \text{trdeg}(L/K).$$

Remark 1 If \mathcal{V} is a variety, then $\text{trdeg}(\mathbb{C}(\mathcal{V})/\mathbb{C})$ is equal to the dimension of \mathcal{V} . If $s = \dim \mathcal{V}$ then we can choose f_1, f_2, \dots, f_s algebraically independent. If $L = \mathbb{C}(f_1, f_2, \dots, f_s)$ then we have

$$s = \text{trdeg}(\mathbb{C}(\mathcal{V})/\mathbb{C}) = \text{trdeg}(K/\mathbb{C}) + \text{trdeg}(\mathbb{C}(\mathcal{V})/K) = s + \text{trdeg}(\mathbb{C}(\mathcal{V})/K)$$

This shows that $\mathbb{C}(\mathcal{V})/K$ is algebraic; i.e., every element of $\mathbb{C}(\mathcal{V})$ is algebraic over K . If $\mathbb{C}(\mathcal{V}) \neq K$, then there exists an element $f_{s+1} \in \mathbb{C}(\mathcal{V})$ with

$$\mathbb{C}(\mathcal{V}) = K(f_{s+1}) = \mathbb{C}(f_1, f_2, \dots, f_{s+1})$$

by the Theorem of the Primitive Element (see [72, Theorem 4.6]).

We are particularly interested in $\text{trdeg}(\mathbb{C}(\mathcal{V})^{\mathcal{G}}/\mathbb{C})$ which is the maximal number of rational invariants that are algebraically independent (over \mathbb{C}). Let $\mathcal{G} \cdot v = \{g \cdot v \mid g \in \mathcal{G}\}$ be the orbit of $v \in \mathcal{V}$ under the \mathcal{G} -action and let $\mathcal{G}_v = \{g \in \mathcal{G} \mid g \cdot v = v\}$ be the stabilizer of v . Then we have $\dim \mathcal{G} = \dim(\mathcal{G} \cdot v) + \dim \mathcal{G}_v$. If $s = \max_{v \in \mathcal{V}} \dim(\mathcal{G} \cdot v)$ is the largest dimension of an orbit, then almost all orbits in \mathcal{V} have dimension s and we call s the generic dimension of an orbit in \mathcal{V} . The following results follows from Rosenlicht’s Theorem [122].

Theorem 1 *If s is the dimension of a generic orbit in \mathcal{V} , then the maximum number of algebraically independent invariants is equal to $\dim \mathcal{V} - s$.*

Proof Rosenlicht proved in [122] that there exist a \mathcal{G} -stable nonempty (Zariski) open subset $\mathcal{U} \subseteq \mathcal{V}$ that has a geometric quotient. A geometric quotient is an algebraic variety \mathcal{U}/\mathcal{G} together with a surjective morphism $\pi : \mathcal{U} \rightarrow \mathcal{U}/\mathcal{G}$ such that the fibers of π are exactly all \mathcal{G} -orbits and π also has additional properties such as $\mathbb{C}(\mathcal{U}/\mathcal{G}) = \mathbb{C}(\mathcal{U})^{\mathcal{G}} = \mathbb{C}(\mathcal{V})^{\mathcal{G}}$. The fibers of π have dimension at most s and almost

all fibers have dimension s . (Actually one can show that all the fibers have dimension s .) The maximum number of algebraically independent invariants is

$$\text{trdeg}(\mathbb{C}(\mathcal{V})^{\mathcal{G}}/\mathbb{C}) = \text{trdeg}(\mathbb{C}(\mathcal{U}/\mathcal{G})/\mathbb{C}) = \dim \mathcal{U}/\mathcal{G} = \dim \mathcal{U} - s = \dim \mathcal{V} - s.$$

□

5.2 Rational Invariants for Conics In \mathbb{P}^2

The result of Theorem 1 is straightforwardly applied to the problem of counting the rational invariants for coplanar conics (i.e. a d -tuple of conics in \mathbb{P}^2). We know from past results that two coplanar conics have two invariants [41, 96, 115], that three coplanar conics have seven invariants [116], and that a d -tuple of conics have $5d - 8$ invariants [53]. This is now briefly shown, before moving on to the more nuanced problem of non-coplanar conics.

Therefore, as an illustration, we can count the number of independent rational invariants for d conics in \mathbb{P}^2 . A conic in \mathbb{P}^2 corresponds to a nonzero quadratic homogeneous polynomial in 3 variables, up to scalar (see Eq. 9). So the variety of all conics can be identified with \mathbb{P}^5 because such a polynomial has $6 = 5 + 1$ coefficients. Let \mathcal{V} be the variety of d -tuples of conics in \mathbb{P}^2 , Then $\mathcal{V} \cong (\mathbb{P}^5)^d$ has dimension $5d$. For \mathcal{G} we take the group PGL_3 which has dimension 8. For $d = 1$ the group PGL_3 acts transitively on all nondegenerate quadratic forms. This means that PGL_3 has a Zariski dense orbit in $\mathcal{V} = \mathbb{P}^5$, so the dimension of a generic orbit is $s = 5$ and the number of independent rational invariants is $\dim \mathcal{V} - s = 5 - 5 = 0$. For $d = 2$, we can explicitly compute the stabilizer of a pair of conics (e.g., $x^2 + 2y^2 - z^2 = 0$ and $2x^2 + y^2 - z^2 = 0$) and note that it is finite. This implies that the dimension of a generic orbit is equal to $\dim \mathcal{G} = 8$. This, in turn, implies that the dimension of a generic orbit is equal to 8 for all $d \geq 2$ and the number of algebraically independent invariants is $\dim \mathcal{V} - s = 5d - 8$.

5.3 Rational Invariants for Conics in \mathbb{P}^3

A method for the robust identification of non-coplanar crater patterns has eluded spacecraft navigators since autonomous crater-based navigation was first explored over 25 years ago. The result is that most existing algorithms constrain the problem to local (nearly coplanar) crater patterns or to only nadir pointing images. To solve this problem requires us to first understand conics in \mathbb{P}^3 .

5.3.1 Counting independent rational invariants for conics in \mathbb{P}^3

Let \mathcal{Z} be the 8-dimensional variety introduced at the end of Section 4.1.2 parameterizing conics in \mathbb{P}^3 , and let $\mathcal{V} = \mathcal{Z}^d$ be the space of d conics in \mathbb{P}^3 . The 15-dimensional group $\mathcal{G} = \text{PGL}_4$ acts on \mathbb{P}^3 , \mathcal{Z} , and \mathcal{V} . If $d = 1$ then the group acts transitively on all nondegenerate conics so that $\mathcal{V} = \mathcal{Z}$ has a dense orbit and there are no non-constant rational invariants. Thus, we search for invariants when $d \geq 2$.

Suppose that $d = 2$ and consider the pair of conics C_1 and C_2 . If the conics are generic enough, then the stabilizer of the pair (C_1, C_2) is finite. To see this, suppose

that $g \in \text{PGL}_4$ lies in the connected component of the identity in the stabilizer of $(\mathcal{C}_1, \mathcal{C}_2)$. Then g must also fix the planes \mathcal{P}_1 and \mathcal{P}_2 through \mathcal{C}_1 and \mathcal{C}_2 respectively. Further, g must also fix the intersections $\mathcal{C}_2 \cap \mathcal{P}_1 = \{\bar{c}_2^{(1)}, \bar{c}_2^{(2)}\}$ and $\mathcal{C}_1 \cap \mathcal{P}_2 = \{\bar{c}_1^{(1)}, \bar{c}_1^{(2)}\}$ and because g lies in the connected component of the identity, it fixes all the points $\bar{c}_1^{(1)}, \bar{c}_1^{(2)}, \bar{c}_2^{(1)}, \bar{c}_2^{(2)}$ on the line $\mathcal{P}_1 \cap \mathcal{P}_2$ individually. There are two points $\bar{r}_1^{(1)}, \bar{r}_1^{(2)}$ on \mathcal{C}_1 such that the tangent lines to \mathcal{C}_1 at $\bar{r}_1^{(1)}$ and $\bar{r}_1^{(2)}$ go through $\bar{c}_2^{(1)}$. So g must also fix $\bar{r}_1^{(1)}$ and $\bar{r}_1^{(2)}$. Among $\bar{r}_1^{(1)}, \bar{r}_1^{(2)}, \bar{c}_1^{(1)}, \bar{c}_1^{(2)}$ in the plane \mathcal{P}_1 there are no three points on the line (Fig. 9), because they all lie on the same conic \mathcal{C}_1 . This implies that g must fix all the points in the plane \mathcal{P}_1 . Similarly it must fix all the points in the plane \mathcal{P}_2 . From this follows that g is the identity element in PGL_4 . Thus, the dimension of a general orbit is equal to $\dim \text{PGL}_4 = 15$, and the same is true for all $d \geq 2$. It follows, therefore, that the maximal number of algebraically independent rational invariants for $d \geq 2$ conics is $\dim \mathcal{V} - 15 = 8d - 15$.

This result agrees with the past literature that has identified one independent rational invariant for a pair of conics in \mathbb{P}^3 [65, 114]. Our result, however, is more general and prepares us to study which invariants (if any) may be recovered from the projection of conics into an image.

5.4 Calculating Invariants from Projections

In Section 5.3.1, we found the invariants for a d -tuple of conics in \mathbb{P}^3 . These, however, are not the invariants we need because we do not measure the crater rims directly in \mathbb{P}^3 . Instead, we image the lunar surface with a camera. Thus, we are in search of rational invariants that may be computed from only the projected crater rims that we see in an image. Such invariants do not always exist.

5.4.1 Invariants from Projections and the Action of PGL_4

We begin by discussing the general setup for computing invariants from projections. To do this, we consider models \mathcal{M} in \mathbb{C}^3 space that are parameterized by some variety

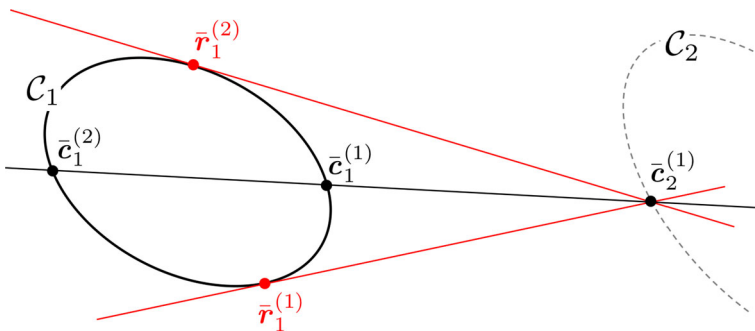


Fig. 9 Since $\bar{r}_1^{(1)}, \bar{r}_1^{(2)}, \bar{c}_1^{(1)}, \bar{c}_1^{(2)}$ all lie on \mathcal{C}_1 no three of these points may be on the same line

\mathcal{V} . For example, \mathcal{V} can consist of all d -tuples of points in $3D$, or all d -tuples of conics. We also have camera position $\bar{o} \in \mathbb{C}^3$ and a perspective projection $\pi : \mathbb{C}^3 \setminus \{\bar{o}\} \rightarrow \mathbb{P}^2$. We are interested in rational functions on \mathcal{V} (functions depending on the model \mathcal{M}) that can be computed from the projected image $\pi(\mathcal{M})$ in a way that is independent on the camera position \bar{o} . We can extend \mathbb{C}^3 to the projective space \mathbb{P}^3 and also extend the camera projection to a map $\mathbb{P}^3 \setminus \{\bar{o}\} \rightarrow \mathbb{P}^2$, where \bar{o} is equal to $o \in \mathbb{C}^3 \subset \mathbb{P}^3$, but viewed in \mathbb{P}^3 . The group PGL_4 acts on \mathbb{P}^3 and contains the group of affine transformations of \mathbb{C}^3 . The group PGL_4 does not act on \mathbb{C}^3 , but it acts on the field of rational functions on \mathbb{C}^3 , because \mathbb{C}^3 and \mathbb{P}^3 have the same rational functions. We assume that the class of models parameterized by \mathcal{V} is closed under the action of PGL_4 (for example, PGL_4 takes conics to conics). Then PGL_4 also acts on the rational functions on \mathcal{V} .

Suppose that f is a rational function on \mathcal{V} such that $f(\mathcal{M})$ can be computed from the projected image $\pi(\mathcal{M})$ in such a way that it does not depend on the choice of the position \bar{o} of the camera. So there exists a function h such that $f(\mathcal{M}) = h(\pi(\mathcal{M}))$ for every choice of \bar{o} . Let $\mathcal{G}_{\bar{o}}$ be the image of the group

$$\left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x & y & z & w \end{bmatrix} \mid w \neq 0 \right\}$$

in PGL_4 . An element $g \in \mathcal{G}_{\bar{o}}$ fixes \bar{o} and all the lines through \bar{o} , so we have

$$f(g \cdot \mathcal{M}) = h(\pi(g \cdot \mathcal{M})) = h(\pi(\mathcal{M})) = f(\mathcal{M}).$$

So f is invariant under the action of $\mathcal{G}_{\bar{o}}$. Since this is true for every choice of \bar{o} , we see that f is invariant under the group \mathcal{G} generated by all $\mathcal{G}_{\bar{o}}$, $\bar{o} \in \mathbb{P}^3$. One can verify that $\mathcal{G} = \text{PGL}_4$. To see this, for example, we observe that the group \mathcal{G} is closed under conjugation with elements in PGL_4 . This implies that \mathcal{G} is a non-trivial normal subgroup of PGL_4 . Since PGL_4 is known to be a simple group, we get $\mathcal{G} = \text{PGL}_4$. This shows that any rational function that can be computed from a camera projection in a way that is independent on the choice of the position of the camera must be invariant under the action of PGL_4 .

5.4.2 Invariants from Projections of Points in \mathbb{P}^3

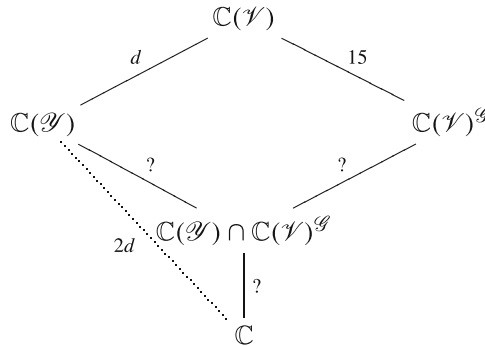
The simplest example of invariants from projections is the case of a d -tuple of points in \mathbb{P}^3 . It is well known that d arbitrarily placed 3D points possess no invariants that one may compute from their projection in an image [16, 28]. Thus, a cloud of random 3D points cannot be indexed for recognition by a pose invariant descriptor, which is a critically important fact. We reproduce this known result here in our more formal framework. In doing so, we develop some of the ideas and tools necessary for the case of conics in \mathbb{P}^3 but within the context of a simpler (and familiar) example.

Therefore, for example, suppose we have d points in \mathbb{P}^3 . For $d \geq 5$ there are $3d - 15$ independent invariants for the action of PGL_4 (because for $d \geq 5$, the generic stabilizer in $(\mathbb{P}^3)^d$ is finite). Perspective projection for some fixed cameragives d

points in \mathbb{P}^2 . Unfortunately, however, no non-constant rational invariant can be computed from just the image. This may seem counter-intuitive, and we now show why this is the case.

Suppose we fix a model $\mathcal{M} \in (\mathbb{P}^3)^d$ consisting of d points. The image may change in appearance with varying camera viewpoint, with the projection being governed by the action of PGL_4 . The variety of the possible images from the model as we move the camera cuts out a variety $\mathcal{U} \subseteq (\mathbb{P}^2)^d$ of dimension at most $\dim \text{PGL}_4 = 15$. For $d \geq 8$, the dimension of \mathcal{U} is strictly smaller than the dimension of $(\mathbb{P}^2)^d$ and one might expect to have at least $2d - 15$ independent invariants that can be computed from the images, but this is wrong as we will see. The problem is that Theorem 1 does not apply. The variety \mathcal{U} of possible images is not an orbit for any group action on $(\mathbb{P}^2)^d$: the group PGL_4 does not act on $(\mathbb{P}^2)^d$, and \mathcal{U} may be bigger than any PGL_3 orbit.

To proceed, suppose that $\mathcal{V} = (\mathbb{P}^3)^d$ and $\mathcal{Y} = (\mathbb{P}^2)^d$, $\bar{o} \in \mathbb{P}^3$ is the position of the camera, and $\pi : \mathcal{V} \rightarrow \mathcal{Y}$ is the projection. (The map π is only defined for elements in $(\mathbb{P}^3 \setminus \{\bar{o}\})^d$.) If $h \in \mathbb{C}(\mathcal{Y})$ is a rational function, then we can pull it back to get a rational function $\pi^*h = h \circ \pi \in \mathbb{C}(\mathcal{V})$, where \circ is the composition. Using the inclusion $\pi^* : \mathbb{C}(\mathcal{Y}) \hookrightarrow \mathbb{C}(\mathcal{V})$ we may view $\mathbb{C}(\mathcal{Y})$ as a subfield of $\mathbb{C}(\mathcal{V})$. We are interested in rational invariants that can be computed from the image in \mathcal{Y} . Such invariants are exactly elements in the intersection field $\mathbb{C}(\mathcal{Y}) \cap \mathbb{C}(\mathcal{V})^{\mathcal{G}}$. We get the following diagram of field extensions with their transcendence degrees:



We will see that $\mathbb{C}(\mathcal{Y}) \cap \mathbb{C}(\mathcal{V})^{\mathcal{G}} = \mathbb{C}$, which means that there are no non-constant rational invariants that can be computed from the image. Suppose that $f \in \mathbb{C}(\mathcal{Y}) \cap \mathbb{C}(\mathcal{V})^{\mathcal{G}}$, or to be more precise, there exists an $h \in \mathbb{C}(\mathcal{Y})$ such that $f = h \circ \pi \in \mathbb{C}(\mathcal{V})^{\mathcal{G}}$. We will show that f (and h) must be constant.

To understand what is happening, we introduce an equivalence relation \sim on $(\mathbb{P}^2)^d$, the variety of possible images of d points in \mathbb{P}^2 . We say $\mathcal{I}_1 \sim \mathcal{I}_2$ is true when there exists a model \mathcal{M} such that $\mathcal{I}_1, \mathcal{I}_2$ both appear as images of \mathcal{M} (but possible from different camera positions). The relation \sim is *not* an equality relation. It satisfies the reflexivity axiom ($\mathcal{I} \sim \mathcal{I}$) and the symmetry axiom ($\mathcal{I}_1 \sim \mathcal{I}_2 \Leftrightarrow \mathcal{I}_2 \sim \mathcal{I}_1$). However, it does not satisfy the transitivity axiom (if $\mathcal{I}_1 \sim \mathcal{I}_2$ and $\mathcal{I}_2 \sim \mathcal{I}_3$, then $\mathcal{I}_1 \sim \mathcal{I}_3$). If π and π' are the projections with respect to different camera positions/orientations, then π' is obtained from π by a projective linear transformation g ,

so that $\pi'(\mathcal{M}) = \pi(g \cdot \mathcal{M})$ for any model \mathcal{M} . If $\mathcal{I} \sim \mathcal{I}'$, then we have $\mathcal{I} = \pi(\mathcal{M})$ and $\mathcal{I}' = \pi'(\mathcal{M})$ for some model and some camera projections π, π' . This means that $h(\mathcal{I}) = h(\pi(\mathcal{M})) = f(\mathcal{M}) = f(g \cdot \mathcal{M}) = f(\pi(g \cdot \mathcal{M})) = h(\mathcal{I}')$.

Let \equiv be the equivalence relation generated by \sim . So we say that $\mathcal{I} \equiv \mathcal{I}'$ if and only if there is a finite sequence of images $\mathcal{I} = \mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_r = \mathcal{I}'$ such that $\mathcal{I}_0 \sim \mathcal{I}_1, \mathcal{I}_1 \sim \mathcal{I}_2, \dots, \mathcal{I}_{r-1} \sim \mathcal{I}_r$. If $\mathcal{I} \equiv \mathcal{I}'$ and $\mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_r$ are as above, then $f(\mathcal{I}) = f(\mathcal{I}_0) = f(\mathcal{I}_1) = \dots = f(\mathcal{I}_r) = f(\mathcal{I}')$.

We will now show that $\mathcal{I} \equiv \mathcal{I}'$ for all images $\mathcal{I}, \mathcal{I}' \in \mathcal{Y}$. Let $(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) \in (\mathbb{P}^2)^d = \mathcal{Y}$ and $\bar{q}_1 \in \mathbb{P}^2$. Define $\bar{o} \in \mathbb{P}^3$ as the position of a camera and $\pi : \mathbb{P}^3 \rightarrow \mathbb{P}^2$ as the camera projection. Let $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_d, \mathcal{N}_1$ be the lines through \bar{o} in \mathbb{P}^3 corresponding to the points $\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d, \bar{q}_1$ respectively. Now, as shown in Fig. 10, define \mathcal{P} as the plane through the lines \mathcal{L}_1 and \mathcal{N}_1 (which intersect at \bar{o}). Let us choose another camera position \bar{o}' in the plane \mathcal{P} randomly. We randomly choose points $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_d$ on the lines $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_d$ respectively (i.e., the points $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_d$ are in general position). Let \mathcal{L}'_j be the line through \bar{o}' and \bar{a}_j for all j . Since \bar{o}' and \bar{a}_1 lie in the plane \mathcal{P} , so does the line \mathcal{L}'_1 . Therefore, the lines \mathcal{L}'_1 and \mathcal{N}_1 intersect at some point \bar{b}_1 . Now we have $(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) = \pi(\bar{a}_1, \bar{a}_2, \dots, \bar{a}_d)$ and $(\bar{p}'_1, \bar{p}'_2, \dots, \bar{p}'_d) = \pi'(\bar{a}_1, \bar{a}_2, \dots, \bar{a}_d)$ so $(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) \sim (\bar{p}'_1, \bar{p}'_2, \dots, \bar{p}'_d)$. Moreover, we have $(\bar{p}'_1, \bar{p}'_2, \dots, \bar{p}'_d) = \pi'(\bar{b}_1, \bar{a}_2, \dots, \bar{a}_d)$ and $(\bar{q}_1, \bar{p}_2, \dots, \bar{p}_d) = \pi(\bar{b}_1, \bar{a}_2, \dots, \bar{a}_d)$, so $(\bar{p}'_1, \bar{p}'_2, \dots, \bar{p}'_d) \sim (\bar{q}_1, \bar{p}_2, \dots, \bar{p}_d)$. From this follows that $(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) \equiv (\bar{q}_1, \bar{p}_2, \dots, \bar{p}_d)$. Repeating this argument shows that for any points $\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d, \bar{q}_1, \bar{q}_2, \dots, \bar{q}_d \in \mathbb{P}^2$ we have

$$(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) \equiv (\bar{q}_1, \bar{p}_2, \dots, \bar{p}_d) \equiv (\bar{q}_1, \bar{q}_2, \bar{p}_3, \dots, \bar{p}_d) \equiv \dots \equiv (\bar{q}_1, \bar{q}_2, \dots, \bar{q}_d).$$

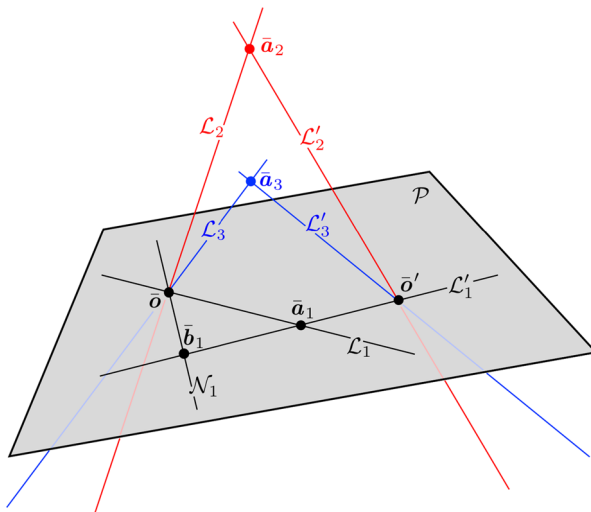


Fig. 10 Visualization of geometric quantities used to show the non-existence of invariants from the projection of arbitrary points in \mathbb{P}^3 . Illustrated here are the line-point sets $\mathcal{L}_1 \cap \mathcal{L}'_1 = \bar{a}_1, \mathcal{N}_1 \cap \mathcal{L}'_1 = \bar{b}_1, \mathcal{L}_2 \cap \mathcal{L}'_2 = \bar{a}_2$, and $\mathcal{L}_3 \cap \mathcal{L}'_3 = \bar{a}_3$ that result from the two camera locations \bar{o} and \bar{o}' . All quantities in black lie in the plane \mathcal{P}

So we have $(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) \equiv (\bar{q}_1, \bar{q}_2, \dots, \bar{q}_d)$. We conclude that $h(\bar{p}_1, \bar{p}_2, \dots, \bar{p}_d) = h(\bar{q}_1, \bar{q}_2, \dots, \bar{q}_d)$. In other words, h and f are constant.

5.4.3 Invariants from Projections of Conics in \mathbb{P}^3

The corollary to a d -tuple of arbitrary points in \mathbb{P}^3 is a d -tuple of arbitrarily placed conics in \mathbb{P}^3 . We will show that there are no non-constant invariants for d conics in \mathbb{P}^3 that can be computed from a perspective projection. The argument is similar to the argument in Section 5.4.2 using an equivalence relation on d -tuples of conics in \mathbb{P}^2 . We use an explicit geometric construction to show that all d -tuples are equivalent.

As before, let \mathcal{Z} be the variety of conics in \mathbb{P}^3 defined at the end of Section 4.1.2. Then, let $\mathcal{V} = \mathcal{Z}^d$ be the variety of d -tuples of conics in \mathbb{P}^3 and $\mathcal{Y} = (\mathbb{P}^5)^d$ be the variety of d -tuples of conics in \mathbb{P}^2 . We define a relation \sim on \mathcal{Y} by $(A_1, A_2, \dots, A_d) \sim (A'_1, A'_2, \dots, A'_d)$ if there exists two camera projections π and π' and conics (C_1, C_2, \dots, C_d) with $\pi(C_1, C_2, \dots, C_d) = (A_1, A_2, \dots, A_d)$ and $\pi'(C_1, C_2, \dots, C_d) = (A'_1, A'_2, \dots, A'_d)$. Let \equiv be the equivalence relation generated by \sim .

Suppose that $B_1, A_1, A_2, \dots, A_d \subseteq \mathbb{P}^2$ are conics. Let $\ell^{(1)}, \ell^{(2)}$ be two lines in \mathbb{P}^2 that are tangent to both A_1 and B_1 with A_1, B_1 within the same region of $\mathbb{P}^2 \setminus (\ell^{(1)} \cup \ell^{(2)})$ (see Fig. 11).

We denote the point at which $\ell^{(j)}$ is tangent to A_1 (respectively B_1) by $\bar{a}_1^{(j)}$ (respectively $\bar{b}_1^{(j)}$). Also, let \bar{q} be the intersection point of $\ell^{(1)}$ and $\ell^{(2)}$. Choose a point $\bar{o} \in \mathbb{P}^3$ (the center of the camera) and let $\pi : \mathbb{P}^3 \setminus \{\bar{o}\} \rightarrow \mathbb{P}^2$ be the camera projection. Let $\mathcal{X}_i = \pi^{-1}(A_i) \cup \{\bar{o}\}$ be the cone with top \bar{o} that corresponds to $A_i \subset \mathbb{P}^2$. Also, define $\mathcal{Y}_1 = \pi^{-1}(B_1) \cup \{\bar{o}\}$. The lines $\ell^{(j)}$ correspond to the plane $\mathcal{L}^{(j)} = \pi^{-1}(\ell^{(j)}) \cup \{\bar{o}\}$ for $j = 1, 2$. The planes $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ intersect in the line $\mathcal{Q} = \pi^{-1}(\bar{q}) \cup \{\bar{o}\}$, as shown in Fig. 12. We choose another point \bar{o}' on the line \mathcal{Q} and $\pi' : \mathbb{P}^3 \setminus \{\bar{o}'\} \rightarrow \mathbb{P}^2$ is the camera perspective projection with respect to the camera position \bar{o}' . We choose some random plane \mathcal{P} in \mathbb{P}^3 . Then

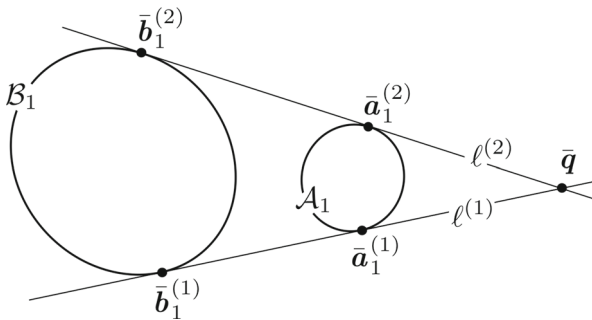


Fig. 11 Visualization of geometric quantities used to show the non-existence of invariants from the projection of arbitrary conics in \mathbb{P}^3 . The conics and lines in this graphic all lie within the image plane. The backprojection of these planar quantities into \mathbb{P}^3 form quadric cones (for A_1 and B_1) or planes (for $\ell^{(1)}$ and $\ell^{(2)}$)

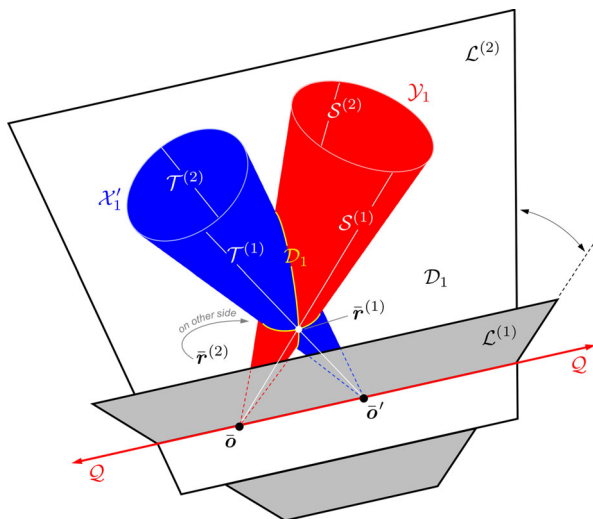


Fig. 12 Visualization of geometric quantities used to show the non-existence of invariants from the projection of arbitrary conics in \mathbb{P}^3 . The lines $\mathcal{T}^{(1)}$ and $\mathcal{S}^{(1)}$ lie in plane $\mathcal{L}^{(1)}$ (lines are white, plane is gray). The lines $\mathcal{T}^{(2)}$ and $\mathcal{S}^{(2)}$ lie in plane $\mathcal{L}^{(2)}$

$C_j = \mathcal{P} \cap \mathcal{X}_j$ is a conic in \mathbb{P}^3 and $\pi(C_j) = \mathcal{A}_j$. Let \mathcal{X}'_j be the cone with top \bar{o}' through C_j . By construction, we have $\pi(C_1, C_2, \dots, C_d) = (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_d)$ and $\pi'(C_1, C_2, \dots, C_d) = (\mathcal{A}'_1, \mathcal{A}'_2, \dots, \mathcal{A}'_d)$ for some conics $\mathcal{A}'_1, \dots, \mathcal{A}'_d$ in \mathbb{P}^2 . We will construct a conic \mathcal{D}_1 in \mathbb{P}^3 with $\pi(\mathcal{D}_1) = \mathcal{B}_1$ and $\pi'(\mathcal{D}_1) = \mathcal{A}'_1$. Note that \mathcal{X}'_1 is a cone with top \bar{o}' that is tangent to the planes $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$. The intersection $\mathcal{S}^{(j)} = \mathcal{Y}_1 \cap \mathcal{L}^{(j)} = \pi^{-1}(\bar{\mathbf{b}}_1^{(j)}) \cup \{\bar{o}\}$ is a line through \bar{o} , and $\mathcal{T}^{(j)} = \mathcal{X}'_1 \cap \mathcal{L}^{(j)}$ is a line through \bar{o}' for $j = 1, 2$. Let $\bar{\mathbf{r}}^{(j)}$ be the intersection point of the lines $\mathcal{S}^{(j)}$ and $\mathcal{T}^{(j)}$ in the plane $\mathcal{L}^{(j)}$ for $j = 1, 2$. Then we have $\pi(\bar{\mathbf{r}}^{(j)}) = \bar{\mathbf{b}}_1^{(j)}$. Moreover we choose a point $\bar{\mathbf{r}}^{(3)} \neq \bar{\mathbf{r}}^{(1)}, \bar{\mathbf{r}}^{(2)}$ that lies on $\mathcal{Y}_1 \cap \mathcal{X}'_1$. Let \mathcal{R} be the plane through $\bar{\mathbf{r}}^{(1)}, \bar{\mathbf{r}}^{(2)}, \bar{\mathbf{r}}^{(3)}$. Define $\mathcal{D}_1 = \mathcal{R} \cap \mathcal{X}'_1$. Then \mathcal{D}_1 is tangent to the planes $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ at the points $\bar{\mathbf{r}}^{(1)}$ and $\bar{\mathbf{r}}^{(2)}$ respectively. Now $\pi(\mathcal{D}_1)$ and \mathcal{B}_1 both are tangent to $\ell^{(1)}$ and $\ell^{(2)}$ at the points $\bar{\mathbf{b}}^{(1)} = \pi(\bar{\mathbf{r}}^{(1)})$ and $\bar{\mathbf{b}}^{(2)} = \pi(\bar{\mathbf{r}}^{(2)})$ respectively, and both contain the point $\pi(\bar{\mathbf{r}}^{(3)})$. It follows that $\pi(\mathcal{D}_1) = \mathcal{B}_1$ because a conic in \mathbb{P}^2 is determined by 3 points and the tangent lines at 2 of the points. One would expect the intersection $\mathcal{Y}_1 \cap \mathcal{X}'_1$ of two quadratic surfaces to be a curve of degree 4. In this case, the curve is reducible, and a union of two conics, and \mathcal{D}_1 is one of them.

It follows that $(\mathcal{B}_1, \mathcal{A}_2, \mathcal{A}_3, \dots, \mathcal{A}_d) = \pi(\mathcal{D}_1, C_2, \dots, C_d)$ and $(\mathcal{A}'_1, \mathcal{A}'_2, \dots, \mathcal{A}'_d) = \pi'(\mathcal{D}_1, C_2, \dots, C_d)$, and therefore $(\mathcal{B}_1, \mathcal{A}_2, \dots, \mathcal{A}_d) \sim (\mathcal{A}'_1, \mathcal{A}'_2, \dots, \mathcal{A}'_d)$. If we combine this with $(\mathcal{A}'_1, \mathcal{A}_2, \dots, \mathcal{A}'_d) \sim (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_d)$ then we get $(\mathcal{B}_1, \mathcal{A}_2, \dots, \mathcal{A}_d) \equiv (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_d)$. Repeating the argument shows that

$$(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_d) \equiv (\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_d)$$

for all conics $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_d, \mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_d$. This implies that there is no nonconstant rational invariant for d conics in \mathbb{P}^3 that can be computed from a projection.

That no rational invariants exist for the projection of d arbitrary conics is an important finding (which we believe to be a novel result). This clearly precludes the use of invariants for lost-in-space crater identification about an arbitrarily shaped body. Fortunately, celestial bodies large enough to exhibit substantial cratering do not have an arbitrary shape. The surface of these bodies can usually be modeled regionally—if not globally—as a nondegenerate quadric surface. Specifically, we note that planets, moons, and dwarf planets are generally ellipsoidal in global shape [90] (this is certainly the case for the Moon), while the smaller minor planets are often globally irregular but regionally ellipsoidal. Comets nuclei often show no such structure. Further, many large bodies (e.g., the Moon) appear nearly planar over sufficiently small portions of the surface. We know from before that invariants exist for coplanar conics (Section 5.2). We now show that invariants also exist for non-coplanar conics lying on a quadratic surface (a quadric).

5.5 Counting Invariants for Conics on a Nondegenerate Quadric Surface

As we have seen, there are no nontrivial invariants for d -tuples of arbitrary conics in 3D space that can be computed from a camera image. In the case of craters on the Moon, the conics we are interested in are not arbitrary, but lie on the surface of the Moon. Thus, instead of considering d -tuples of arbitrary conics, we consider the variety \mathcal{V} of d -tuples of conics for which there exists a nondegenerate quadric surface that contains all of them. We find projective invariants to exist for $d \geq 3$ conics lying on the same nondegenerate quadric surface.

5.5.1 Pairs of Conics on a Nondegenerate Quadric Surface

First consider the case $d = 2$. Suppose that \mathcal{C}_1 and \mathcal{C}_2 are two conics that lie in the planes given by the linear equations $f_1 = 0$ and $f_2 = 0$ respectively. Then \mathcal{C}_1 and \mathcal{C}_2 lie in the degenerate quadric surface $f_1 f_2 = 0$. Suppose both conics also lie on a nondegenerate surface defined by a quadratic equation $g = 0$. This surface is not unique, because for every t , the surface defined by $g + t f_1 f_2 = 0$ contains both conics. If two conics \mathcal{C}_1 and \mathcal{C}_2 lie on a nondegenerate surface defined by $g = 0$, then \mathcal{C}_1 is defined by $f_1 = g = 0$ and \mathcal{C}_2 is defined by $f_2 = g = 0$. The intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ is defined by $f_1 = f_2 = g = 0$ and consists of two (possibly complex) points. On the other hand, if $\mathcal{C}_1 \cap \mathcal{C}_2$ consists of two points, one can show that both conics lie on a nondegenerate quadric surface. To find a pair $(\mathcal{C}_1, \mathcal{C}_2)$ in \mathcal{V} , we can choose the conic \mathcal{C}_1 arbitrarily in the 8-dimensional variety of conics, we choose the plane \mathcal{P}_2 that contains \mathcal{C}_2 in the 3-dimensional space of hyperplanes, Now the conic \mathcal{C}_2 is an element of the 5-dimensional variety of conics in \mathcal{P}_2 , but there are two constraints because the conic has to go through the two points of $\mathcal{C}_1 \cap \mathcal{P}_2$. So the dimension of the variety \mathcal{V} is $8 + 3 + (5 - 2) = 14$. The 15-dimensional group PGL_4 acts on \mathcal{V} . Let \mathcal{H} be the connected component of the identity in the stabilizer of $(\mathcal{C}_1, \mathcal{C}_2) \in \mathcal{V}$. The conics \mathcal{C}_1 and \mathcal{C}_2 intersect in two (possibly complex) points \bar{e}_1 and \bar{e}_2 . So g will fix the points \bar{e}_1 and \bar{e}_2 . If \mathcal{L} is the line through \bar{e}_1 and \bar{e}_2 then g will map \mathcal{L} to itself. If \bar{e}_3 is a 3rd point on \mathcal{L} then g may map \bar{e}_3 to another point of \mathcal{L} . However, if g also fixes the point \bar{e}_3 then a similar argument as in

Section 5.3.1 shows that g must fix the planes \mathcal{P}_1 and \mathcal{P}_2 pointwise, and must be the identity. This shows that the stabilizer of the pair $(\mathcal{C}_1, \mathcal{C}_2)$ is at most 1-dimensional. Since the generic stabilizer has dimension at most 1, the dimension of a generic orbit is at least $15 - 1 = 14$. Since $\dim \mathcal{V} = 14$, this implies that there are no rational invariants for a pair of conics lying on a nondegenerate quadric surface.

5.5.2 Many ($d \geq 3$) Conics on a Nondegenerate Quadric Surface

Let us now assume that $d \geq 3$. If a quadric surface S contains 3 distinct conics $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ then S is the unique quadric surface that contains these three conics. To see this, suppose that S' is another quadric surface through the 3 conics. Suppose that S and S' are defined by the equations $h = 0$ and $h' = 0$ respectively, where h and h' are homogeneous quadratic polynomials in 4 variables. Let \mathcal{P}_i be the plane through \mathcal{C}_i given by $f_i = 0$ where f_i is a linear function. Let $\bar{a} \in \mathbb{P}^3$. We can multiply h and h' with nonzero scalars such that $(h - h')(\bar{a}) = 0$. In the plane \mathcal{P}_i , the restriction of the quadratic polynomial $h - h'$ vanishes on $\mathcal{C}_i \cup \{\bar{a}\}$. This implies that $h - h'$ is divisible by f_i for $i = 1, 2, 3$. But then $h - h'$ is divisible by $f_1 f_2 f_3$ and $h - h'$ must be zero.

To parametrize d conics that lie on a quadric surface, we can first choose the surface that is given by a quadratic polynomial in 4 variables up to a scalar. Such a polynomial has 10 coefficients, so the quadric surface is determined by $10 - 1 = 9$ parameters. Now, each of the conics is determined by a hyperplane section of the quadric surface. Hyperplanes are parameterized by points in \mathbb{P}^3 . So the variety \mathcal{V} of d -tuples of conics that lie on a common quadric surface has dimension $9 + 3d$. The stabilizer of a generic point in \mathcal{V} is finite, so the dimension of a generic orbit is $15 = \dim \text{PGL}_4$. So the number of independent rational invariants is $(9 + 3d) - 15 = 3d - 6$ for $d \geq 3$.

6 Computing Invariants from Crater Rims in an Image

The results of Section 5 established the existence of invariants for conics lying on either a plane or a nondegenerate quadric surface. While the coordinate-free approach used in the previous section is a powerful tool for studying such invariants, we must ultimately impose a coordinate system for practical computation of these invariants from data. Doing so is straightforward, but does require some additional mathematical machinery. The details are now discussed.

6.1 Geometry of Invariants for Non-Coplanar Conics

6.1.1 Pair of Arbitrary Non-Coplanar Conics

Two arbitrary 3D conics do not generally intersect one another, even over the complex numbers. This may be seen through the intersection of surfaces (illustrated in Fig. 13). The line formed by the intersection of the two conic planes intersects each conic at two places, producing four intersection points. The cross-ratio of these

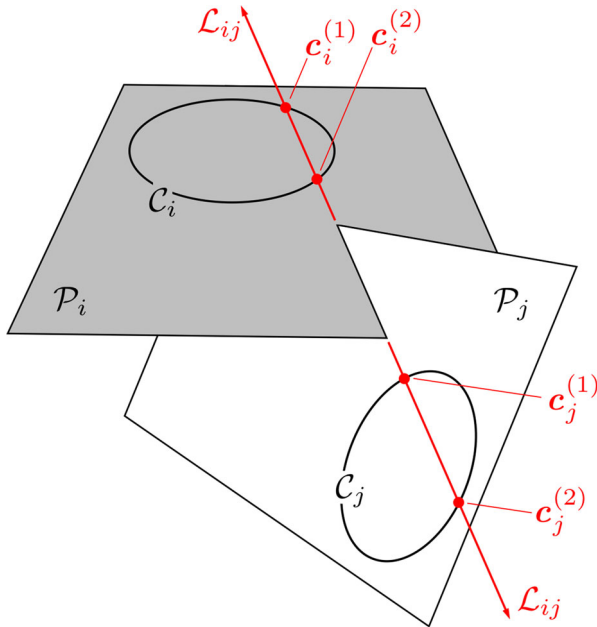


Fig. 13 Two arbitrarily placed non-coplanar conics do not intersect one another. The common 3D line \mathcal{L}_{ij} formed by the intersection of their planes intersects each of the two conics in two places, creating four (usually distinct) points. The two intersection points for a particular conic are complex valued if \mathcal{L}_{ij} does not physically intersect that conic. The four colinear intersection points (possibly complex valued) may be used to form a cross-ratio, which is the single unique 3D invariant for a pair of 3D conics

four colinear points is the unique 3D invariant for a pair of non-coplanar conics. This was observed in [87] and [114], which we now rederive by other means as we build towards a methods for computing projective invariants. Further, the results of Section 5.3.1 tell us that this is the only 3D invariant for a pair of non-coplanar conics. We also show that this 3D invariant does not lead to a useful projective invariant that may be constructed from an image of these two conics.

Suppose we have crater i described by the 3D conic \mathcal{C}_i that lies in plane \mathcal{P}_i . Let the Moon-centered quadric cone of crater i be given by \mathcal{X}_i as described in Eq. 37. By construction, this cone must pass through the conic locus \mathcal{C}_i , and we interpret \mathcal{C}_i as the conic section from Eq. 38.

Now, suppose we have two craters: crater i and crater j . We compute their intersection by substitution of Eq. 38 as,

$$\mathcal{C}_i \cap \mathcal{C}_j = (\mathcal{P}_i \cap \mathcal{X}_i) \cap (\mathcal{P}_j \cap \mathcal{X}_j) = (\mathcal{P}_i \cap \mathcal{P}_j) \cap \mathcal{X}_i \cap \mathcal{X}_j \tag{59}$$

Define the 3D line \mathcal{L}_{ij} as the intersection of planes \mathcal{P}_i and \mathcal{P}_j ,

$$\mathcal{L}_{ij} = \mathcal{P}_i \cap \mathcal{P}_j \tag{60}$$

such that

$$C_i \cap C_j = \mathcal{L}_{ij} \cap \mathcal{X}_i \cap \mathcal{X}_j = (\mathcal{L}_{ij} \cap \mathcal{X}_i) \cap (\mathcal{L}_{ij} \cap \mathcal{X}_j) \tag{61}$$

The intersection of a line with a quadric cone, e.g. $\mathcal{L}_{ij} \cap \mathcal{X}_i$, produces two points. Define the intersection points of this line with crater i as $\{c_i^{(1)}, c_i^{(2)}\}$. It is easy to see, as illustrated in Fig. 13, that these intersection points are not generally the same for two arbitrary conics. Therefore,

$$C_i \cap C_j = \{c_i^{(1)}, c_i^{(2)}\} \cap \{c_j^{(1)}, c_j^{(2)}\} = \emptyset \tag{62}$$

Observe that the four points $\{c_i^{(1)}, c_i^{(2)}, c_j^{(1)}, c_j^{(2)}\}$ are colinear since they all lie on the line \mathcal{L}_{ij} . They are also distinct, meaning their cross ratio ρ_{ij} is a 3D invariant of the conic pair [114]

$$\rho_{ij} = \text{Cr}(c_i^{(1)}, c_i^{(2)}, c_j^{(1)}, c_j^{(2)}) \tag{63}$$

The difficulty with this 3D invariant is that it is not recoverable from a single image of the two conics. The two 3D conics will project into two coplanar conics in the image plane. The image conics have four (often complex valued) intersection points, which are not necessarily colinear. In the absence of other constraints, these intersection points in the image plane are not related to the 3D line \mathcal{L}_{ij} or its intersection points $\{c_i^{(1)}, c_i^{(2)}, c_j^{(1)}, c_j^{(2)}\}$ with the two 3D conics. This is because the two conics do not actually intersect one another. That no projective invariant exists from a pair of non-coplanar conics agrees with the findings of Section 5.4.3. The non-existence of projective invariants for three or more arbitrarily placed non-coplanar conics follows by similar arguments.

In the special case where the conics do intersect one another, their intersection is preserved when imaged from an arbitrary pose by a projective camera. One of the most flexible ways to ensure two 3D conics intersect over the complex numbers is to constrain them to lie on a nondegenerate quadric surface (e.g., sphere, ellipsoid). This is now shown.

6.1.2 Pair of Non-Coplanar Conics on a Nondegenerate Quadric Surface

Consider lunar craters modeled as conics lying on a nondegenerate quadric surface \mathcal{S} . Under this assumption, we may model the surface of Moon by the quadric locus parameterized by Q_M ,

$$\mathcal{S} = \left\{ \bar{\xi} \in \mathbb{P}^3 \mid \bar{\xi}^T Q_M \bar{\xi} = 0 \right\} \tag{64}$$

We observe here that the quadric surface need not be a global surface fit, but only a good approximation of the portion of the lunar surface where the craters reside. Indeed, some crater clusters may use one quadric surface while other crater clusters use a different quadric surface. This choice in no way affects the theoretical developments that follow.

Now, consider crater i lying on plane \mathcal{P}_i . For the 3D rim of a crater i to also be on the surface of the Moon, it must lie on the intersection of this plane with the \mathcal{S} . Therefore, the 3D conic locus for crater i must be the intersection

$$\mathcal{C}_i = \mathcal{P}_i \cap \mathcal{S} \tag{65}$$

which produces a circular crater if \mathcal{S} is a sphere. An ellipsoidal (non-spherical) body would generally produce an elliptical crater. The discussion that follows holds for both spherical and ellipsoidal bodies.

Further, as in Section 6.1.1 and described in Eq. 38, the crater may also be viewed as the conic section produced by intersecting plane \mathcal{P}_i with the Moon-centered quadric cone \mathcal{X}_i . However, since the crater must lie on both \mathcal{X}_i and \mathcal{S} , it is clear that we can also write the 3D crater conic as,

$$\mathcal{C}_i = \mathcal{X}_i \cap \mathcal{S} \tag{66}$$

Therefore, we see that

$$\mathcal{C}_i = \mathcal{X}_i \cap \mathcal{S} = \mathcal{P}_i \cap \mathcal{S} \tag{67}$$

Suppose we have two craters: \mathcal{C}_i and \mathcal{C}_j . Because \mathcal{C}_i and \mathcal{C}_j are formed by the intersections of planes \mathcal{P}_i and \mathcal{P}_j with the quadric surface \mathcal{S} , we observe that,

$$\mathcal{C}_i \cap \mathcal{C}_j = (\mathcal{P}_i \cap \mathcal{S}) \cap (\mathcal{P}_j \cap \mathcal{S}) = (\mathcal{P}_i \cap \mathcal{P}_j) \cap \mathcal{S} = \mathcal{L}_{ij} \cap \mathcal{S} \tag{68}$$

where \mathcal{L}_{ij} is the 3D line formed by the intersection of the two crater planes (see Eq. 60). This is illustrated in Fig. 14.

If the two craters physically intersect, the line \mathcal{L}_{ij} pierces \mathcal{S} and the intersection points are real. If the craters do not physically intersect, the intersection occurs over the complex numbers. Let $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ be the two intersection points

$$\{\bar{s}_{ij}^{(1)}, \bar{s}_{ij}^{(2)}\} = \mathcal{L}_{ij} \cap \mathcal{S} \tag{69}$$

Therefore, define \mathcal{P}_{ij} as the plane passing through the center of the Moon and the line \mathcal{L}_{ij} (or, equivalently passing through the center of the Moon, $\bar{s}_{ij}^{(1)}$, and $\bar{s}_{ij}^{(2)}$). Clearly, the numerical stability of computing \mathcal{P}_{ij} in this manner is poor if the craters are close to one another and \mathcal{P}_i and \mathcal{P}_j are nearly coplanar. Better numeric stability may be achieved by taking a different approach.

The conic intersection from Eq. 68 may also be written in terms of the quadric cones instead of the planes,

$$\mathcal{C}_i \cap \mathcal{C}_j = (\mathcal{X}_i \cap \mathcal{S}) \cap (\mathcal{X}_j \cap \mathcal{S}) = (\mathcal{X}_i \cap \mathcal{X}_j) \cap \mathcal{S} \tag{70}$$

Without constraints, the intersection of two cones $\mathcal{X}_i \cap \mathcal{X}_j$ is a set of four lines passing through the center of the Moon that pierce \mathcal{S} at eight different points. Four of the points will be on the wrong side and come from conics that do not actually exist; these can be ignored. When the conics lie on a nondegenerate quadric surface, we

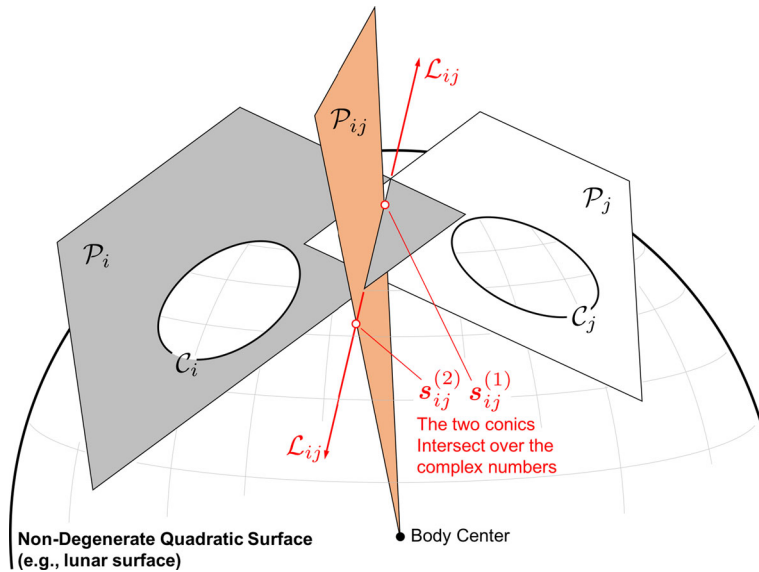


Fig. 14 Two non-coplanar conics C_i and C_j lying on a nondegenerate quadric surface must intersect in two points (which happen to be complex valued in this example). The two intersection points lie on the line \mathcal{L}_{ij} formed by the intersection of the two crater planes. The intersection points also lie in the plane \mathcal{P}_{ij} formed by the join of \mathcal{L}_{ij} and the body center

find the other four intersection points are a set of repeated intersections. To see this, combine the above results to find,

$$(\mathcal{X}_i \cap \mathcal{X}_j) \cap \mathcal{S} = C_i \cap C_j = \mathcal{L}_{ij} \cap \mathcal{S} = \{\bar{s}_{ij}^{(1)}, \bar{s}_{ij}^{(2)}\} \tag{71}$$

The advantage of using the quadric cone intersections to describe $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ is that this remains well defined and more numerically stable as the craters become close and their planes become nearly parallel. This is especially important when viewing craters at a local level, where craters are nearly coplanar.

Since the points $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ lie on C_i , the projection of these points lie on the projection of C_i (which we called \mathcal{A}_i). The same holds for C_j , and its projection \mathcal{A}_j . Therefore, the projection of $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ is recoverable as two of the intersection points of the image conics. That is,

$$\bar{u}_{ij}^{(k)} = \mathbf{P}_C^M \bar{s}_{ij}^{(k)} \subset \mathcal{A}_i \cap \mathcal{A}_j, \quad k = 1, 2 \tag{72}$$

where \mathbf{P}_C^M is the projection matrix from Eq. 46 and the pixel coordinate $\bar{u}_{ij}^{(k)}$ is the projection of $\bar{s}_{ij}^{(k)}$ into the image.

The challenge, therefore, is to determine which of the four intersection points of \mathcal{A}_i and \mathcal{A}_j are the projection of $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$. There are six possible combinations of these four points, corresponding to six possibilities for the projection of line \mathcal{L}_{ij} into the line ℓ_{ij} . We now discuss how to uniquely compute ℓ_{ij} .

Consider the two conics \mathcal{A}_i and \mathcal{A}_j described by the conic locus matrices A_i and A_j . We recall that

$$\bar{u}^T A_i \bar{u} = 0 \text{ and } \bar{u}^T A_j \bar{u} = 0 \tag{73}$$

It follows that we can form a pencil of conics parameterized by the scalar ratio $\lambda : \mu$ that also pass through the same four intersection points as A_i and A_j ,

$$\bar{u}^T (\lambda A_i + \mu A_j) \bar{u} = 0 \tag{74}$$

If we force the matrix $\lambda A_i + \mu A_j$ to be a degenerate conic, then it becomes two lines (or a double line) and we may find its intersection with the two conics. We find six possibilities, one of which is the line ℓ_{ij} that we seek.

Letting $\mu = 1$ for easy computation, we force the conic to be degenerate by setting the determinant to zero,

$$\det (\lambda A_i + A_j) = 0 \tag{75}$$

Since both A_i and A_j are full rank and well-conditioned, we may rewrite this as

$$\det [A_j (-A_i)^{-1} - \lambda I_{3 \times 3}] = 0 \tag{76}$$

which is a simple 3×3 eigenvalue problem. When there is no actual intersection of the two conics in the image, only one of the resulting degenerate conics will produce real valued lines [117]. The specific line we seek is ℓ_{ij} , which is the projection of \mathcal{L}_{ij} . As the intersection of two non-coplanar planes, \mathcal{L}_{ij} is real valued by construction. Thus, its projection ℓ_{ij} is also real valued. Therefore we always seek the eigenvalue leading to a degenerate conic of real valued lines, discarding the four complex valued line possibilities. It will soon become apparent which eigenvalue to choose.

Therefore, suppose we choose one of the eigenvalues to construct the degenerate conic formed by

$$B_{ij} = \lambda A_i + A_j \tag{77}$$

where we observe that B_{ij} has rank 2. The remaining steps are now to find the lines described by B_{ij} , determine if they are real valued, and then determine which one is ℓ_{ij} .

A degenerate conic made of two lines, g and h , is defined by the symmetric matrix

$$B_{ij} = g h^T + h g^T \tag{78}$$

where our task is now to find g and h given B_{ij} . There are a variety of ways to accomplish this task. Our specific approach is similar to the framework outlined in [117]. Therefore, proceeding in this way, we first recall that the intersection of the two lines in homogeneous coordinates is computed as [51]

$$\bar{z} = g \times h \tag{79}$$

Further, observing that

$$[\bar{z} \times] = \mathbf{g}\mathbf{h}^T - \mathbf{h}\mathbf{g}^T \tag{80}$$

we find that that

$$\mathbf{B}_{ij} + [\bar{z} \times] = (\mathbf{g}\mathbf{h}^T + \mathbf{h}\mathbf{g}^T) + (\mathbf{g}\mathbf{h}^T - \mathbf{h}\mathbf{g}^T) = 2\mathbf{g}\mathbf{h}^T \tag{81}$$

is a rank one matrix with columns proportional to \mathbf{g} and rows proportional to \mathbf{h} . We must now find \bar{z} .

To do this, a quick calculation will confirm that

$$\mathbf{B}_{ij}^* = (\mathbf{g}\mathbf{h}^T + \mathbf{h}\mathbf{g}^T)^* = -(\mathbf{g} \times \mathbf{h})(\mathbf{g} \times \mathbf{h})^T = -\bar{z}\bar{z}^T \tag{82}$$

Or, in terms of the original ellipses,

$$\mathbf{B}_{ij}^* = (\lambda\mathbf{A}_i + \mathbf{A}_j)^* = -\bar{z}\bar{z}^T \tag{83}$$

The diagonals of $-\mathbf{B}_{ij}^*$ are the squares of the elements of \bar{z} . Thus, defining the columns and elements of \mathbf{B}_{ij}^* as

$$\mathbf{B}_{ij}^* = [\mathbf{b}_1^* \ \mathbf{b}_2^* \ \mathbf{b}_3^*] = \begin{bmatrix} b_{11}^* & b_{12}^* & b_{13}^* \\ b_{21}^* & b_{22}^* & b_{23}^* \\ b_{31}^* & b_{32}^* & b_{33}^* \end{bmatrix} \tag{84}$$

any column of \mathbf{B}_{ij}^* may be scaled to find \bar{z} according to

$$\bar{z} = -\mathbf{b}_k^* / \sqrt{-b_{kk}^*} \tag{85}$$

where any $k \in \{1, 2, 3\}$ with $b_{kk} \neq 0$ will work. Best numerical performance is achieved by selecting the value for k that yields $\max_k \|\mathbf{b}_k^*\|$. If the ellipses do not intersect, only one eigenvalue from Eq. 76 will produce a real valued estimate of \bar{z} . The eigenvalue to choose is the one that makes the diagonal of \mathbf{B}_{ij}^* negative.

Therefore, with \bar{z} known, compute \mathbf{D}

$$\mathbf{D} = \mathbf{B}_{ij} + [\bar{z} \times] = 2\mathbf{g}\mathbf{h}^T \tag{86}$$

we can then compute \mathbf{g} from any non-zero column of \mathbf{D} and \mathbf{h} from any non-zero row of \mathbf{D} . In general, it is best to find the element of \mathbf{D} with the largest absolute value and pick the corresponding row and column for \mathbf{g} and \mathbf{h} .

Given the two lines \mathbf{g} and \mathbf{h} , one of these corresponds to ℓ_{ij} . The other does not. Each of the lines \mathbf{g} and \mathbf{h} divide \mathbb{P}^2 into two regions. Since the line we seek comes from the projection of $\mathcal{L}_{ij} = \mathcal{P}_i \cap \mathcal{P}_j$, the two conics must not lie in the same region. That is, we seek a line that passes between the two conics. Only one line will satisfy this condition and this is the line we choose for ℓ_{ij} .

If we were to compute the intersection of ℓ_{ij} with either \mathcal{A}_i or \mathcal{A}_j we would obtain two points that are the projection $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$. Fortunately such a computation is not necessary since we work directly with ℓ_{ij} in subsequent discussions.

Briefly, we observe that \mathcal{C}_i and the two points $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ all lie in the plane \mathcal{P}_i . While a projective invariant exists for two points and a conic (all coplanar) [41], the points must not lie on the conic. Since $\bar{s}_{ij}^{(1)}$ and $\bar{s}_{ij}^{(2)}$ lie on \mathcal{C}_i by construction, we seek an alternative way of constructing an invariant.

Consider instead the 2D line ℓ_{ij} formed by the projection of 3D line \mathcal{L}_{ij} (or, equivalently, by the join of image points $\bar{u}_{ij}^{(1)}$ and $\bar{u}_{ij}^{(2)}$). While there is not an invariant for a single line and a conic, there is an invariant for two lines and a conic [123]. This motivates the study of invariants for a triad of craters.

6.1.3 Triad of Non-Coplanar Conics on a Nondegenerate Quadric Surface

Suppose a projective camera observes three craters: \mathcal{C}_i , \mathcal{C}_j , and \mathcal{C}_k . The intersection of the corresponding planes (\mathcal{P}_i , \mathcal{P}_j , and \mathcal{P}_k) produces the three 3D lines \mathcal{L}_{ij} , \mathcal{L}_{ik} , and \mathcal{L}_{jk} . Three planes intersect in a point, which is also the location where the 3D line formed by two planes intersects the third plane. This point, defined as a_{ijk} , is the apex of the pyramid formed by the three planes,

$$a_{ijk} = \mathcal{P}_i \cap \mathcal{P}_j \cap \mathcal{P}_k = \mathcal{L}_{ij} \cap \mathcal{P}_k = \mathcal{L}_{ik} \cap \mathcal{P}_j = \mathcal{L}_{jk} \cap \mathcal{P}_i \tag{87}$$

Now, let the three 3D craters \mathcal{C}_i , \mathcal{C}_j , and \mathcal{C}_k project to 2D conics \mathcal{A}_i , \mathcal{A}_j , and \mathcal{A}_k in the image. For any given pair of 3D conics (e.g., \mathcal{C}_i and \mathcal{C}_j), the 3D line through their intersection points (e.g., \mathcal{L}_{ij}) is coplanar with both of these 3D conics and it is possible to recover its projection (e.g., ℓ_{ij}) from just the projected image conics (e.g., \mathcal{A}_i and \mathcal{A}_j). Thus, for a triad of observed craters in an image, we may compute the three lines ℓ_{ij} , ℓ_{ik} , and ℓ_{jk} . Since \mathcal{C}_i , \mathcal{L}_{ij} , \mathcal{L}_{ik} are all coplanar, an invariant exists in the image using \mathcal{A}_i , ℓ_{ij} , and ℓ_{ik} . The same is true for craters \mathcal{C}_j and \mathcal{C}_k .

We may show that such an invariant exists by use of a cross ratio, and this invariant may be computed directly by use of a Cayley-Klein metric. The usual cross-ratio applies to four points on a line. However, by the duality of points and lines in \mathbb{P}^2 , we can also form a cross-ratio of four lines passing through a point. Therefore, as shown in Fig. 15, consider a conic \mathcal{A}_i . Let the two lines ℓ_{ij} and ℓ_{ik} be described by the 3×1 vectors ℓ_{ij} and ℓ_{ik} , and let these two lines intersect at the point $\bar{p} = \ell_{ij} \times \ell_{ik}$. Using the pole-polar relation for a conic, we may find the two lines from \bar{p} that are tangent to the conic \mathcal{A}_i , which we call w_1 and w_2 . Since these four lines go through the point \bar{p} , we may form a cross ratio that is a projective invariant

$$\text{Cr}(\ell_{ij}, w_1, w_2, \ell_{ik}) \tag{88}$$

We can now form a classical Cayley-Klein metric as

$$d(\ell_{ij}, \ell_{ik}) = \frac{1}{2} \ln [\text{Cr}(\ell_{ij}, w_1, w_2, \ell_{ik})] \tag{89}$$

which, from hyperbolic geometry, is known to be equivalent to

$$\alpha_i = \cosh [d(\ell_{ij}, \ell_{ik})] = \frac{\|\ell_{ij}^T \mathbf{A}_i^* \ell_{ik}\|}{\sqrt{(\ell_{ij}^T \mathbf{A}_i^* \ell_{ij})(\ell_{ik}^T \mathbf{A}_i^* \ell_{ik})}} \tag{90}$$

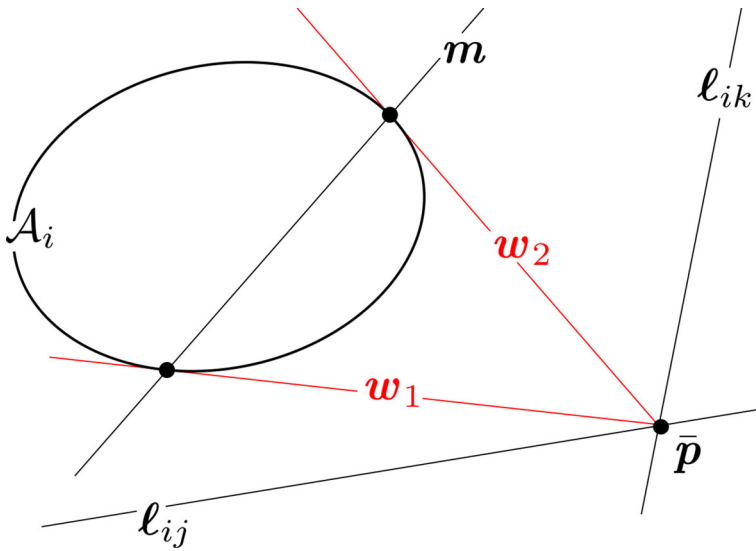


Fig. 15 Visualization of contributing components to invariant for a conic \mathcal{A}_i and two lines ℓ_{ij} and ℓ_{jk} . The lines w_1 and w_2 pass through \bar{p} and are tangent to \mathcal{A}_i . The line m is the polar of point \bar{p} with respect to \mathcal{A}_i

where the adjugate matrix A_i^* describes the conic envelope of \mathcal{A}_i . To understand the need for the absolute value in the numerator, recall from Section 4 that ℓ_{ij} and $-\ell_{ij}$ describe the same line (same for ℓ_{ik} and $-\ell_{ik}$). Likewise, A_i^* and $-A_i^*$ describe the same conic envelope. Thus, since the appropriate choice of sign is not always clear *a priori*, the absolute value in the numerator is a simple way to robustly remove the sign ambiguity and ensure that $\cosh [d(\ell_{ij}, \ell_{ik})] \geq 1$ for all $d(\ell_{ij}, \ell_{ik})$. Absolute values are not necessary in the denominator since this is quadratic in ℓ_{ij} , ℓ_{ik} , and A_i^* . We may also write, therefore,

$$\alpha_i^2 = \cosh^2 [d(\ell_{ij}, \ell_{ik})] = \frac{(\ell_{ij}^T A_i^* \ell_{ik})^2}{(\ell_{ij}^T A_i^* \ell_{ij})(\ell_{ik}^T A_i^* \ell_{ik})} \tag{91}$$

Since the cross ratio from Eq. 88 is a projective invariant, it follows immediately that the Cayley-Klein metric from Eq. 89 and its hyperbolic cosine in Eqs. 90 and 91 are also projective invariants.

At first, the Cayley-Klein development may seem of purely academic interest. However, it is of critical importance when building a searchable index for lunar crater identification. It is only through this relation that we fully understand why the metric for α_i^2 from Eq. 91 (the typical invariant found in the literature for a coplanar set of a conic and two lines [123]) loses descriptiveness for small distances $d(\ell_{ij}, \ell_{ik})$. The spacecraft navigator will immediately see that this is analogous to how small inter-star angles θ lose descriptiveness when cataloged as $\cos \theta$ instead of by θ (i.e., we

need to tighten the matching tolerance as $\theta \rightarrow 0$ when using a $\cos \theta$ index). Thus, we prefer to index on d instead of α^2 .

We usually choose to only consider craters that do not physically intersect (i.e. do not overlap), since many CDAs have difficulty providing high-quality crater rim fits for intersecting craters. Moreover, if restrict ourselves to only non-overlapping craters, then we are guaranteed that the cross ratio is positive. This implies that $d \geq 0$ and is real, and that the denominator on the right-hand side of Eq. 90 is also real. Since $d \geq 0$, we know that

$$d = \operatorname{arcosh} \alpha_i = \ln \left[\alpha_i + \sqrt{\alpha_i^2 - 1} \right] \tag{92}$$

Therefore, we may relate the cross-ratio, the Cayley-Klein metric (ℓ_{ij}, ℓ_{ik}) , and the invariant α_i according to

$$\ln \left[\alpha_i + \sqrt{\alpha_i^2 - 1} \right] = d(\ell_{ij}, \ell_{ik}) = \frac{1}{2} \ln [\operatorname{Cr}(\ell_{ij}, \mathbf{w}_1, \mathbf{w}_2, \ell_{ik})] \tag{93}$$

where we choose to construct the invariant to index with the Cayley-Klein metric,

$$J_i = d(\ell_{ij}, \ell_{ik}) = \operatorname{acosh} \left\{ \frac{\|\ell_{ij}^T \mathbf{A}_i^* \ell_{ik}\|}{\sqrt{(\ell_{ij}^T \mathbf{A}_i^* \ell_{ij}) (\ell_{ik}^T \mathbf{A}_i^* \ell_{ik})}} \right\} \tag{94}$$

An identical procedure produces corresponding invariants for craters j and k

$$J_j = d(\ell_{ij}, \ell_{jk}) = \operatorname{acosh} \left\{ \frac{\|\ell_{ij}^T \mathbf{A}_j^* \ell_{jk}\|}{\sqrt{(\ell_{ij}^T \mathbf{A}_j^* \ell_{ij}) (\ell_{jk}^T \mathbf{A}_j^* \ell_{jk})}} \right\} \tag{95}$$

$$J_k = d(\ell_{ik}, \ell_{jk}) = \operatorname{acosh} \left\{ \frac{\|\ell_{ik}^T \mathbf{A}_k^* \ell_{jk}\|}{\sqrt{(\ell_{ik}^T \mathbf{A}_k^* \ell_{ik}) (\ell_{jk}^T \mathbf{A}_k^* \ell_{jk})}} \right\} \tag{96}$$

We know from our review of the literature that many previous attempts at crater identification have accidentally employed feature descriptors with elements that are not independent. Thus, before proceeding further, it is necessary to ensure that the invariants J_i , J_j , and J_k are independent. Since we know from Section 5.5 that a d -tuple of conics on a quadric surface has $3d - 6$ invariants (leading to $9-6=3$ invariants for this case), it follows that if the three invariants J_i , J_j , and J_k are independent then we have found all the algebraically independent invariants that exist. This is now shown.

We observe that the invariants $\alpha_i^2, \alpha_j^2, \alpha_k^2$ (and hence also $\alpha_i, \alpha_j, \alpha_k$ and J_i, J_j, J_k) are algebraically independent. We calculate these invariants for a particular model, since it suffices to show that the invariants are independent when restricted to special subclasses of models. We assume that the lunar surface is the sphere $x^2 + y^2 + z^2 = 1$, and that the three craters are the circles defined by intersecting the sphere with the planes given by $x = t_1, y = t_2$ and $z = t_3$ respectively. Within the plane $z = t_3$ we have the crater given by $x^2 + y^2 = 1 - t_3^2$ and the lines $x = t_1$ and $y = t_2$. So we get

$$A_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & t_3^2 - 1 \end{bmatrix} \text{ and } A_3^* = \begin{bmatrix} t_3^2 - 1 & 0 & 0 \\ 0 & t_3^2 - 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{97}$$

$$\ell_{13} = \begin{bmatrix} 1 \\ 0 \\ -t_1 \end{bmatrix} \text{ and } \ell_{23} = \begin{bmatrix} 0 \\ 1 \\ -t_2 \end{bmatrix} \tag{98}$$

Thus, we may compute α_3^2

$$\alpha_3^2 = \frac{(\ell_{13} A_3^* \ell_{23})^2}{(\ell_{12}^T A_3^* \ell_{12})(\ell_{13}^T A_3^* \ell_{13})} = \frac{t_1^2 t_2^2}{(t_1^2 + t_3^2 - 1)(t_2^2 + t_3^2 - 1)} \tag{99}$$

and, by symmetry, we also find

$$\alpha_2^2 = \frac{t_1^2 t_3^2}{(t_1^2 + t_2^2 - 1)(t_2^2 + t_3^2 - 1)} \tag{100}$$

$$\alpha_1^2 = \frac{t_2^2 t_3^2}{(t_1^2 + t_2^2 - 1)(t_1^2 + t_3^2 - 1)} \tag{101}$$

It is possible to verify that the Jacobi matrix

$$J(\alpha_1^2, \alpha_2^2, \alpha_3^2) = \left(\frac{\partial \alpha_i^2}{\partial t_j} \right)_{1 \leq i, j \leq 3} \tag{102}$$

is invertible for some choice of t_1, t_2, t_3 (e.g., $t_1 = t_2 = t_3 = 1/2$). This implies that $\alpha_1^2, \alpha_2^2, \alpha_3^2$ (and, therefore, J_i, J_j, J_k) are algebraically independent [75].

Despite the rather long development up to this point, the practical implementation of this method for computing the invariants J_i, J_j, J_k is straightforward and is summarized in Algorithm 1. The reader should note that the invariants J_i, J_j, J_k are computed directly from the measured image conics A_1, A_2, A_3 without requiring any knowledge of the camera position or orientation. This permits lost-in-space crater identification.

Algorithm 1 Pseudocode for computing invariants for a triad of non-coplanar conics lying on a common non-degenerate quadric surface. Inputs are the apparent conics in an image and outputs are invariants.

```

1: procedure  $\{J_1, J_2, J_3\} = \text{NONCOPLANARCRATERINVARIANTS}(A_1, A_2, A_3)$ 
2:   for  $\{i, j\} = \{1, 2\}, \{2, 3\}, \{3, 1\}$  do
3:      $\{\lambda_n\}_{n=1}^3 = \text{eig}[A_j(-A_i)^{-1}]$ 
4:     for  $n = 1$  to 3 do
5:       compute  $B_{ij} = \lambda_n A_i + A_j$  ▷ (77)
6:       compute  $B_{ij}^*$  as adjugate matrix of  $B_{ij}$ 
7:       find  $k$  as index of the largest magnitude diagonal of  $B_{ij}^*$ 
8:       compute  $\bar{z} = -b_k^* / \sqrt{-b_{kk}^*}$  ▷ (85)
9:       if  $\bar{z}$  is real then
10:        Break
11:       compute  $D = B_{ij} + [\bar{z} \times]$  ▷ (86)
12:       find indices  $k, \ell$  as the largest entry  $\|D_{k\ell}\|$  in  $D$ 
13:       compute  $\mathbf{h} = [D_{k1}, D_{k2}, D_{k3}]^T$ 
14:       compute  $\mathbf{g} = [D_{\ell 1}, D_{\ell 2}, D_{\ell 3}]^T$ 
15:       compute  $\ell_{ij}$  a line from  $\{\mathbf{h}, \mathbf{g}\}$  that passes between centers of ellipses  $i, j$ 
16:       compute  $A_1^*, A_2^*, A_3^*$  ▷ (20)
17:       compute  $J_1, J_2, J_3$  ▷ (94), (95), (96)
18: return  $\{J_1, J_2, J_3\}$ 

```

The utility of this framework is now briefly demonstrated. Consider a regional crater pattern such as the one shown in Fig. 16, where there is notable curvature of the Moon. This example has about 18.1 deg between the surface normal of crater A_j and A_k , thus necessitating the consideration of non-coplanar invariants. Two substantially different views of the same crater pattern are shown. The image on the left is nearly nadir pointing and the image on the right is pointed over 30 deg off nadir. As can be seen, the non-coplanar invariants J_1, J_2 , and J_3 are identical in both images since

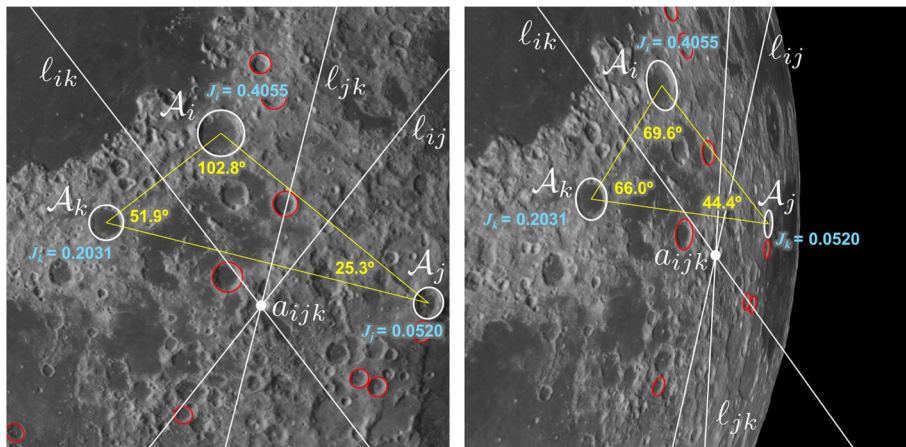


Fig. 16 Example of invariants J_i, J_j , and J_k (blue) for the same triad of craters (A_i, A_j , and A_k) seen from two different vantage points. The invariants remain unchanged under perspective projection from any viewing geometry. The triangle interior angles (yellow) are not a projective invariant. Synthetic images of the lunar surface were produced using PANGU Planet Surface Simulation Software developed by the Space Technology Centre at the University of Dundee, Scotland [111]

they are formal projective invariants. This makes evident the power of such invariants for crater pattern recognition.

It should be stressed that only specific items in Fig. 16 are coplanar, with the vast majority of items being non-coplanar. Consequently general combinations of lines or crater ellipses from left image cannot be transformed to the right image with a common homography.

Also shown in Fig. 16 are the crater triad interior angles (yellow). We highlight these here because crater triangle interior angles are often proposed as descriptors for lost-in-space crater pattern recognition, e.g. [49]. It is clear, however, that the interior angles are not projective invariants and do not provide a robust means of pattern indexing if off-nadir viewing is possible.

6.2 Geometry of Invariants for Coplanar Conics

When viewing only a very small portion of the lunar surface, the observed craters are often nearly coplanar. Smaller craters are also more elliptical. Thus, patterns of small craters close to one another are better described using a triad of coplanar craters. Here, we revisit the problem of invariants of two and three coplanar conics using the framework of Semple and Kneebone and [128] and of Quan [115, 116]

6.2.1 Pairs of Coplanar Conics

A pair of coplanar conics has two projective invariants [41]. We saw in Section 6.1.1 that two arbitrarily placed 3D conics have no intersections and in Section 6.1.2 that two 3D conics on a nondegenerate quadric surface have two intersections. We observe now that two coplanar conics always have four intersections. These four intersection points (possibly over the complex numbers, possibly repeated) persist under the action of a projective camera, which permits easy computation of two invariants.

Four specified points on a conic may be used to construct a cross-ratio, which is a projective invariant. It may initially be unclear why this is the case since the four points around the ellipse are not collinear, thus a brief explanation is warranted. As discussed earlier, using the duality of points and lines in \mathbb{P}^2 , we can also form a projective invariant by the cross-ratio of four lines passing through a point. Chasles' theorem [128] states that the cross-ratio is a constant for the pencil of four lines from four points on a conic to a fifth point also on the conic (Fig. 17). Therefore, we may directly form two invariants: the first by finding the cross ratio of the lines from the four intersection points to any other point on first conic, and the second by finding the cross ratio of the lines from the same four intersection points to any other point on the second conic.

Therefore, let $\{\bar{a}_k\}_{k=1}^4 = \mathcal{A}_i \cap \mathcal{A}_j$ be the four intersection points of the conics in the image plane and let \bar{p} be an arbitrary point on conic \mathcal{A}_i . Since the points are written in homogeneous coordinates, we may form the line joining \bar{p} and \bar{a}_k as $w_k = \bar{p} \times \bar{a}_k$. Thus, the cross ratio $\text{Cr}(w_1, w_2, w_3, w_4)$ for ellipse \mathcal{A}_i is a projective invariant, and is the same for any choice of p on the ellipse. A second invariant may be computed

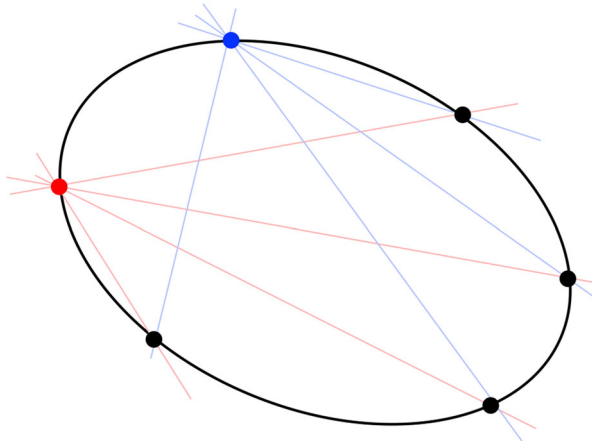


Fig. 17 Chasles’ theorem states that the cross-ratio formed by the family of red lines is the same as the cross-ratio of the family of blue lines

in exactly the same way by picking an arbitrary point on ellipse \mathcal{A}_j . This cross-ratio may also be rewritten in terms of the eigenvalues of the matrix $\mathbf{A}_i^{-1}\mathbf{A}_j$ [87].

Explicit computation of these cross ratios is not efficient, so an alternate technique is more appropriate—though the two alternate invariants are rational functions of the cross ratios.

A more conveniently computable form of the invariants for a pair of coplanar conics may be found by direct analysis of the matrices \mathbf{A}_i and \mathbf{A}_j describing the image conics. To see this, consider the pencil of conics $\lambda\mathbf{A}_i + \mu\mathbf{A}_j$ parameterized by the scalars λ and μ . Recall now that 3D crater \mathcal{C}_i projects to image ellipse \mathcal{A}_i according to the homography from Eq. 58. Since the craters are assumed coplanar their appearance in an image is related by a common homography,

$$\mathbf{H}_C^T (\lambda\mathbf{A}_i + \mu\mathbf{A}_j) \mathbf{H}_C = s_i\lambda\mathbf{C}_i + s_j\mu\mathbf{C}_j \tag{103}$$

Taking the determinant yields

$$|\mathbf{H}_C|^2 |\lambda\mathbf{A}_i + \mu\mathbf{A}_j| = |s_i\lambda\mathbf{C}_i + s_j\mu\mathbf{C}_j| \tag{104}$$

which may be expanded as

$$\begin{aligned} |\mathbf{H}_C|^2 |\lambda\mathbf{A}_i + \mu\mathbf{A}_j| &= |\mathbf{H}_C|^2 (\Theta_1\lambda^3 + \Theta_2\lambda^2\mu + \Theta_3\lambda\mu^2 + \Theta_4\mu^3) \\ &= \Theta'_1\lambda^3 + \Theta'_2\lambda^2\mu + \Theta'_3\lambda\mu^2 + \Theta'_4\mu^3 \\ &= |s_i\lambda\mathbf{C}_i + s_j\mu\mathbf{C}_j| \end{aligned} \tag{105}$$

Thus we find that

$$|\mathbf{H}_C|^2\Theta_1 = \Theta'_1 \quad |\mathbf{H}_C|^2\Theta_4 = \Theta'_4 \tag{106}$$

$$|\mathbf{H}_C|^2\Theta_2 = \Theta'_2 \quad |\mathbf{H}_C|^2\Theta_3 = \Theta'_3 \tag{107}$$

The coefficients $\Theta_1, \Theta_2, \Theta_3, \Theta_4$ are only *semi-invariants* since their numerical value changes with $|\mathbf{H}_C|$ and the arbitrary choices of s_i and s_j . These may easily be turned into *rational invariants* by considering appropriate ratios. To see this, first express the coefficients in terms of \mathbf{A}_i and \mathbf{A}_j ,

$$\Theta_1 = |\mathbf{A}_i| \quad \Theta_4 = |\mathbf{A}_j| \tag{108}$$

$$\Theta_2 = \text{Tr}[\mathbf{A}_i^* \mathbf{A}_j] = |\mathbf{A}_i| \text{Tr}[\mathbf{A}_i^{-1} \mathbf{A}_j] = \Theta_1 \text{Tr}[\mathbf{A}_i^{-1} \mathbf{A}_j] \tag{109}$$

$$\Theta_3 = \text{Tr}[\mathbf{A}_j^* \mathbf{A}_i] = |\mathbf{A}_j| \text{Tr}[\mathbf{A}_j^{-1} \mathbf{A}_i] = \Theta_4 \text{Tr}[\mathbf{A}_j^{-1} \mathbf{A}_i] \tag{110}$$

Likewise, we find that

$$\Theta'_1 = s_i^3 |\mathbf{C}_i| \quad \Theta'_4 = s_j^3 |\mathbf{C}_j| \tag{111}$$

$$\Theta'_2 = \frac{s_j}{s_i} \Theta'_1 \text{Tr}[\mathbf{C}_i^{-1} \mathbf{C}_j] \tag{112}$$

$$\Theta'_3 = \frac{s_i}{s_j} \Theta'_4 \text{Tr}[\mathbf{C}_j^{-1} \mathbf{C}_i] \tag{113}$$

As shown in [128], the simplest pair of independent rational invariants are

$$\frac{\Theta_1 \Theta_3}{\Theta_2^2} = \frac{\Theta'_1 \Theta'_3}{\Theta_2'^2} \quad \text{and} \quad \frac{\Theta_2 \Theta_4}{\Theta_3^2} = \frac{\Theta'_2 \Theta'_4}{\Theta_3'^2} \tag{114}$$

which we quickly verify produce a pair of non-trivial scalars that are not dependent on \mathbf{H}_C, s_i , or s_j (hence, they are the two fully generic invariants for a pair of conics):

$$\frac{|\mathbf{A}_j| \text{Tr}[\mathbf{A}_j^{-1} \mathbf{A}_i]}{|\mathbf{A}_i| \left(\text{Tr}[\mathbf{A}_i^{-1} \mathbf{A}_j]\right)^2} = \frac{\Theta_1 \Theta_3}{\Theta_2^2} = \frac{\Theta'_1 \Theta'_3}{\Theta_2'^2} = \frac{|\mathbf{C}_j| \text{Tr}[\mathbf{C}_j^{-1} \mathbf{C}_i]}{|\mathbf{C}_i| \left(\text{Tr}[\mathbf{C}_i^{-1} \mathbf{C}_j]\right)^2} \tag{115}$$

$$\frac{|\mathbf{A}_i| \text{Tr}[\mathbf{A}_i^{-1} \mathbf{A}_j]}{|\mathbf{A}_j| \left(\text{Tr}[\mathbf{A}_j^{-1} \mathbf{A}_i]\right)^2} = \frac{\Theta_2 \Theta_4}{\Theta_3^2} = \frac{\Theta'_2 \Theta'_4}{\Theta_3'^2} = \frac{|\mathbf{C}_i| \text{Tr}[\mathbf{C}_i^{-1} \mathbf{C}_j]}{|\mathbf{C}_j| \left(\text{Tr}[\mathbf{C}_j^{-1} \mathbf{C}_i]\right)^2} \tag{116}$$

Since the left-hand side of these expressions are independent of \mathbf{H}_C , the same value is computed for any arbitrary projection of the conics \mathcal{C}_i and \mathcal{C}_j . These results are in agreement with [115], which also follows [128].

Following the convention of [41], we now observe that the equations simplify considerably by choosing the scale such that $|\mathbf{A}_i| = |\mathbf{A}_j| = 1$, making the coefficients Θ_1 and Θ_4 trivial. Using this simplification, substitute (108) and (111) into Eq. 106 to find

$$|\mathbf{H}|^2 = s_i^3 = s_j^3 \tag{117}$$

which, in turn, leads to

$$\Theta_2 = \Theta'_2 \rightarrow \text{Tr} \left[\mathbf{A}_i^{-1} \mathbf{A}_j \right] = \text{Tr} \left[\mathbf{C}_i^{-1} \mathbf{C}_j \right] \tag{118}$$

$$\Theta_3 = \Theta'_3 \rightarrow \text{Tr} \left[\mathbf{A}_j^{-1} \mathbf{A}_i \right] = \text{Tr} \left[\mathbf{C}_j^{-1} \mathbf{C}_i \right] \tag{119}$$

Therefore, when conics are pre-scaled to $|\mathbf{A}_i| = |\mathbf{A}_j| = 1$, the two unique invariants are simply

$$I_{ij} = \Theta_2 = \text{Tr} \left[\mathbf{A}_i^{-1} \mathbf{A}_j \right] \tag{120}$$

$$I_{ji} = \Theta_3 = \text{Tr} \left[\mathbf{A}_j^{-1} \mathbf{A}_i \right] \tag{121}$$

We note that [41] follows a completely different approach to finding I_{ij} and I_{ji} , and even more geometrically-inspired derivations may be found in [96]. The specific approach shown here is chosen because of its extensibility to finding invariants of three or more coplanar conics.

These two invariants for a pair of coplanar conics were used by Cheng, et al., for crater identification in [18, 20]. We find, however, that better performance is often achieved by considering a triad of craters.

6.2.2 Triads of Coplanar Conics

It is established in Section 5.2 that there exist $5d - 8$ projective invariants for a d -tuple of coplanar conics ($d \geq 2$). Thus, for a triad of coplanar conics, there exists $15 - 8 = 7$ projective invariants. This fact has been known for some time [53, 116].

The procedure for finding these seven invariants follows the same framework used for a pair of coplanar conics. Therefore, consider the determinant of a *net* of three conics [128]. As observed in [116], this evaluates to

$$\begin{aligned} |\lambda \mathbf{A}_i + \mu \mathbf{A}_j + \sigma \mathbf{A}_k| &= \Theta_1 \lambda^3 + \Theta_2 \lambda^2 \mu + \Theta_3 \lambda \mu^2 + \Theta_4 \mu^3 + \Theta_5 \lambda^2 \sigma \\ &\quad + \Theta_6 \lambda \sigma^2 + \Theta_7 \sigma^3 + \Theta_8 \mu^2 \sigma + \Theta_9 \mu \sigma^2 + \Theta_{10} \lambda \mu \sigma \end{aligned} \tag{122}$$

We immediately see that the coefficients of the first nine terms correspond to the pair-wise combinations of the three conics, with results that follow directly from the two-conic case

$$\Theta_1 = |\mathbf{A}_i|, \quad \Theta_4 = |\mathbf{A}_j|, \quad \Theta_7 = |\mathbf{A}_k| \tag{123}$$

and that

$$\Theta_2 = \Theta_1 \text{Tr} \left[\mathbf{A}_i^{-1} \mathbf{A}_j \right], \quad \Theta_3 = \Theta_4 \text{Tr} \left[\mathbf{A}_j^{-1} \mathbf{A}_i \right] \tag{124}$$

$$\Theta_5 = \Theta_1 \text{Tr} \left[\mathbf{A}_i^{-1} \mathbf{A}_k \right], \quad \Theta_6 = \Theta_7 \text{Tr} \left[\mathbf{A}_k^{-1} \mathbf{A}_i \right] \tag{125}$$

$$\Theta_8 = \Theta_4 \text{Tr} \left[\mathbf{A}_j^{-1} \mathbf{A}_k \right], \quad \Theta_9 = \Theta_7 \text{Tr} \left[\mathbf{A}_k^{-1} \mathbf{A}_j \right] \tag{126}$$

Note that Θ_{10} is the only coefficient that simultaneously depends on all three of the conics, thus making it the only term unique to a triad (and not to a pair). This may be computed as

$$\Theta_{10} = \frac{1}{2} \text{Tr} \{ [(A_j + A_k)^* - (A_j - A_k)^*] A_i \} \tag{127}$$

where the reader is briefly reminded that A^* is the adjugate of A . Computation of this result (which, interestingly, does not appear in [116], [53], or any other reference discussing triads of coplanar conics) is tedious but straightforward, and is left as an exercise to the reader.

The ten coefficients of Eq. 122 may be used to define the seven unique invariants. As with the pair of coplanar conics, the simplest approach is to choose the arbitrary scale of the matrices A_i , A_j , and A_k such that $|A_i| = |A_j| = |A_k| = 1$. Thus the coefficients Θ_1 , Θ_4 , and Θ_7 become trivial, and the remaining seven coefficients become the unique invariants. That is, with $|A_i| = |A_j| = |A_k| = 1$, we find that

$$I_{ij} = \text{Tr} [A_i^{-1} A_j] \quad I_{ji} = \text{Tr} [A_j^{-1} A_i] \tag{128}$$

$$I_{ik} = \text{Tr} [A_i^{-1} A_k] \quad I_{ki} = \text{Tr} [A_k^{-1} A_i] \tag{129}$$

$$I_{jk} = \text{Tr} [A_j^{-1} A_k] \quad I_{kj} = \text{Tr} [A_k^{-1} A_j] \tag{130}$$

$$I_{ijk} = \text{Tr} \{ [(A_j + A_k)^* - (A_j - A_k)^*] A_i \} \tag{131}$$

Therefore, if three coplanar craters C_i , C_j , and C_k project into three image conics \mathcal{A}_i , \mathcal{A}_j , and \mathcal{A}_k (described by the matrices A_i , A_j , and A_k), then the seven scalar values in Eqs. 128 to 131 will remain exactly the same under perspective projection regardless of the camera position and attitude.

7 Creating a Crater Pattern Descriptor from Invariants

Having identified projective invariants that may be computed from the observed crater rim geometry in an image (three for a triad of non-coplanar conics on a quadric surface and seven for a triad of coplanar conics), we wish to use this information to construct a pattern descriptor. This descriptor must have a structure that can be consistently reproduced to allow for matching against a static index, which requires some care since the invariants from Section 6 depend on the ordering of the observed craters. We note that the sensitivity of invariants to ordering is not inherently a bad thing (after all, we must specifically match each individual crater to the catalog)—though it does result in a few different methods for building descriptors.

The index is built using descriptors for crater triads. We must choose, therefore, whether we want to impose ordering of the three craters within the triad before or after matching to the index. If we choose the former, then the invariants developed before may be used directly. If we choose the latter, we must either sort the projective invariants from Section 6 into a prescribed order or transform them into *projec-*

itive and permutation (p^2) invariants. These conventions each have their advantages, which we now discuss.

7.1 Crater Pattern Descriptors with Projective Invariants

Suppose we observe a triad of craters that we want to match to a precomputed index in an order-dependent fashion. Since the crater pattern must be observed from above (i.e., the camera must be looking downwards from above the lunar surface because it cannot be inside the Moon), there are three possible orderings of the craters. If we choose to arrange these in a clockwise fashion within the image (see Fig. 18) then the three possibilities are (1, 2, 3), (3, 1, 2), and (2, 3, 1). For a match to be successful, the observed descriptor must match the index descriptor. Therefore, if the descriptor entries are order dependent, then a match only occurs when the observation ordering matches index ordering.

There are at least two options for matching to an index with order-dependent descriptors. First, if each crater triad has a single index entry, the index must be searched three times (once with each possible observation ordering). Alternatively, the index could include three entries for each crater triad (one for each ordering) and the larger index is searched only once. The first method (one index per triad) performs best in the presence of measurement noise.

The pattern descriptors in this scenario are simply a concatenation of the invariants from Section 6. Thus, for a triad of conics on a quadric surface, we build a three-element descriptor:

$$J_{ijk}^T = [J_i \ J_j \ J_k] \tag{132}$$

where the individual elements are from Eqs. 94 to 96.

Likewise, for a triad of conics on a plane, we build a seven-element descriptor:

$$I_{ijk}^T = [I_{ij} \ I_{jk} \ I_{ki} \ I_{ji} \ I_{kj} \ I_{ik} \ I_{ijk}] \tag{133}$$

where the individual elements are from Eqs. 128 to 131.

7.2 Crater Pattern Descriptors with Sorted Projective Invariants

As before, suppose we observe a triad of craters. Now, instead of having an order-dependent descriptor, suppose that we want to match to a precomputed index in

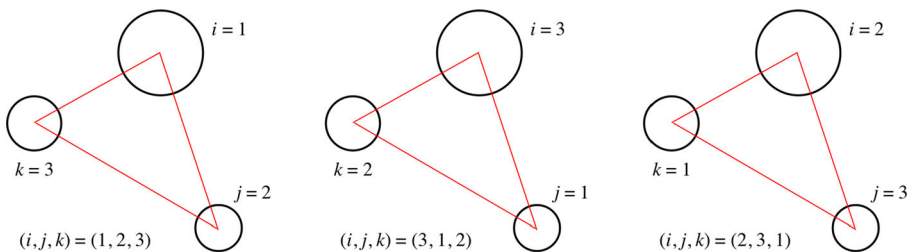


Fig. 18 Crater triad labels in an image are assigned in a clockwise fashion. There are three possible permutations of these labels, which may be interpreted as a cyclic permutation of the labels

an order-independent fashion. There are a variety of approaches one might use to achieve this objective. We will see, however, that some care is required to arrive at a robust solution.

A very common technique for achieving permutation invariance is to always permute the order of the descriptor elements such the smallest (or largest) element appears first. For example, one possible descriptor is $J'_{ijk} = [J'_i, J'_j, J'_k]$, where $J'_i \leq J'_j \leq J'_k$ is a rearrangement of (J_i, J_j, J_k) , i.e., we sort the numbers J_i, J_j, J_k from small to large and let the result be the descriptor. This arrangement was employed by Hanak [49, 50] to define the clockwise/counterclockwise sense of a crater pattern by ordering the legs of the crater triangle from shortest to longest. Similarly, for their lost-in-space algorithm, Maass, et al., [86] presort the entries in their index of crater triangles (which consists of two triangle interior angles) such that the smallest interior angle is first. Other examples of this idea abound within the space navigation community, ranging from crater pattern identification to star pattern identification.

Returning to the problem at hand, we know from our labeling convention that the crater pattern indices may only undergo a cyclic permutation. Thus, we may cyclically permute the entries until smallest valued invariant is in the first position—that is, where (J'_i, J'_j, J'_k) is a cyclic permutation of (J_i, J_j, J_k) with $J'_i = \min(J_i, J_j, J_k)$.

Thus, for a triad of conics on a quadric surface, we build a three-element descriptor:

$$J^T_{ijk} = [J'_i \ J'_j \ J'_k] \tag{134}$$

where we label observed crater i such that $J'_i = \min(J_i, J_j, J_k)$ and the rest of the primed invariants follow by cyclic permutation. The individual elements are from Eqs. 94 to 96.

Likewise, for a triad of conics on a plane, we build a seven-element descriptor:

$$I^T_{ijk} = [I'_{ij} \ I'_{jk} \ I'_{ki} \ I'_{ji} \ I'_{kj} \ I'_{ik} \ I'_{ijk}] \tag{135}$$

where we label observed crater i such that $I'_{ij} = \min(I_{ij}, I_{jk}, I_{ki})$ and the rest of the primed invariants follow by cyclic permutation of the crater labels i, j, k (and, to be clear, not by cyclic permutation of the elements of the descriptor vector). The individual elements are from Eqs. 128 to 131.

The usual argument for anchoring the descriptor element ordering with the smallest (or largest) element is that it is simple. We find in practice, however, that this ordering scheme is not optimally descriptive and is not very robust to noise. Indeed, it is not uncommon for noise to cause two similarly-valued invariants to switch position. If this happens to the value used to anchor the descriptor permutation then it renders the entire descriptor useless—even though the invariants themselves still describe the crater pattern well. Thus, we cycle through the sorted projective invariant descriptors exactly as we do for the unsorted case (Section 7.1). However, the pre-sorting works most of the time, and greatly reduces the trials required (on average) to find a match.

The main problem with this approach is how sorting populates the k -d search space. By sorting the descriptor to have the smallest element first, we preferentially populate the search space for smaller values along this dimension. This doesn't matter so long as the measurement noise is very small—but, when supplied with noisy measurements, this reduces the likelihood that the correct descriptor will correspond with the first nearest neighbor. The end result is that an index of sorted descriptors will yield fewer matches (as compared with Section 7.1) when the data is noisy. Thus, while it may provide speed-up for low noise cases, both matching performance and speed quickly degrade as noise increases.

7.3 Crater Pattern Descriptors with Projective and Permutation (p^2) Invariants

Our objective is to construct a descriptor that is *invariant* to the order in which the craters are observed. It should be no surprise, therefore, that invariant theory is (once again) a useful mathematical framework for achieving this objective. There is some precedence for this approach using the so-called projective and permutation (p^2) invariants. The p^2 invariants are discussed for a set of five points in [76] and for a pair of conics in [87]. An attempt was made to apply this to the problem of crater identification in [110], though this approach suffered from a number of mistakes (see Section 3.3). Within the context of this past work, we now introduce the first complete set of p^2 invariants for a triad of conics on a quadric surface or on a plane (e.g., p^2 invariants for a triad crater rims on the Moon).

We briefly remind the reader that the p^2 invariants introduced here are algebraic functions of the projective invariants from Section 6. Thus, the p^2 invariants do not represent any new (independent) information. They simply express the information we already have in a different way.

7.3.1 Non-Coplanar Crater Patterns (Three-Element Descriptor)

We first develop the p^2 invariants for a triad of conics lying on a quadric surface. We know there exists exactly three algebraically independent invariants for this case, and we are able to compute three such invariants J_i, J_j, J_k directly from an image using Eqs. 94 to 96. Our goal is to compute rational functions of J_i, J_j, J_k that are invariant to a cyclic permutation of the crater labels.

Consider, therefore, the triple of numbers $\{x, y, z\}$. The standard generators of the ring of polynomial invariants for the symmetric group, S_3 , of all coordinate permutations are the elementary symmetric functions (see [103, Chapter 4])

$$e_1 = x + y + z \quad (136)$$

$$e_2 = xy + yz + zx \quad (137)$$

$$e_3 = xyz \quad (138)$$

which are clearly invariant to permutations of $\{x, y, z\}$. To obtain the generators of the invariant ring for the cyclic group $\mathbb{Z}/3\mathbb{Z}$, which we may also view as the alternating group A_3 , we require one more invariant:

$$\Delta = (x - y)(y - z)(z - x) \quad (139)$$

(see for example [103, Example 4.24]). We observe that Δ is invariant under cyclic permutations, but that odd permutations change its sign. While these polynomials could certainly be used to construct p^2 invariants and build a pattern descriptor, we note that the polynomials are of different degree which often leads to undesirable numerical properties for indexing. Thus, we instead look for three rational functions, where the difference in the degrees of the numerator and denominator are the same for all three functions.

To do this, we begin by defining a polynomial map

$$F(x, y, z) = (F_1(x, y, z), F_2(x, y, z), F_3(x, y, z)) \tag{140}$$

where F_1, F_2, F_3 are continuous rational functions with the property that $F(x, y, z) = F(x', y', z')$ if and only if (x', y', z') is a cyclic permutation of (x, y, z) . Let the cyclic group $\mathbb{Z}/3\mathbb{Z}$ act on \mathbb{R}^3 by permuting the coordinates. Then we desire F_1, F_2, F_3 to be invariant functions on \mathbb{R}^3 .

Whenever a finite group acts by a linear transformation, there exists a particular coordinate change where the action of the group becomes diagonal. That is, in some coordinate system, the matrix describing the action of each group element is a diagonal matrix. This follows from two facts from the representation theory of finite groups (see [42, Chapter 1]). First, every representation of a finite group is a direct sum of irreducible representations. Second, every irreducible representation of a finite abelian group is 1-dimensional. For the specific problem under consideration here (the action of $\mathbb{Z}/3\mathbb{Z}$ on \mathbb{R}^3), we can make this very explicit: namely, the change of coordinates making the action diagonal is a discrete Fourier transform (DFT) (which involves complex numbers). To make these ideas explicit, let a, b, c be

$$a = x + y + z \tag{141}$$

$$b = x + \zeta y + \zeta^2 z \tag{142}$$

$$c = x + \zeta^2 y + \zeta z \tag{143}$$

where $\zeta = e^{2\pi i/3} = -\frac{-1+\sqrt{3}i}{2}$. Then the generator σ of $\mathbb{Z}/3\mathbb{Z}$ acts by the diagonal matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \zeta & 0 \\ 0 & 0 & \zeta^2 \end{bmatrix}$$

We define $F_1, F_2,$ and F_3 as rational functions of $a, b, c,$

$$F_1(x, y, z) = a = x + y + z \tag{144}$$

$$\begin{aligned} F_2(x, y, z) &= \frac{b^2}{c} + \frac{c^2}{b} \tag{145} \\ &= \frac{2(x^3 + y^3 + z^3) + 12xyz - 3(x^2y + y^2x + y^2z + z^2y + z^2x + x^2z)}{x^2 + y^2 + z^2 - (xy + yz + zx)} \end{aligned}$$

$$\begin{aligned} F_3(x, y, z) &= \frac{1}{i} \left(\frac{b^2}{c} - \frac{c^2}{b} \right) \tag{146} \\ &= \frac{-3\sqrt{3}(x - y)(y - z)(z - x)}{x^2 + y^2 + z^2 - (xy + yz + zx)} \end{aligned}$$

The denominator of F_2 and F_3 is equal to $\frac{1}{2}[(x - y)^2 + (y - z)^2 + (z - x)^2]$ and is equal to 0 if and only if $x = y = z$. We can also verify that

$$F_2^2 + F_3^2 = 2[(x - y)^2 + (y - z)^2 + (z - x)^2] = 2bc$$

This implies that if (x, y, z) is a sequence that converges to a point (t, t, t) , then $F_2(x, y, z)$ and $F_3(x, y, z)$ converge to 0. Consequently, we can extend F_2 and F_3 to continuous functions on all of \mathbb{R}^3 by defining them to be 0 whenever $x = y = z$.

Lemma 1 $F(x, y, z) = F(x', y', z')$ if and only if (x', y', z') is a cyclic permutation of (x, y, z) for all $(x, y, z), (x', y', z') \in \mathbb{R}^3$.

Proof Suppose now that $F(x, y, z) = F(x', y', z')$. Let a, b, c be as given in Eqs. 144–146 and define similarly $a' = x' + y' + z'$, etc. To begin, observe that if $F_2(x, y, z) = F_2(x', y', z') = F_3(x, y, z) = F_3(x', y', z') = 0$ then we must have $x = y = z$ and $x' = y' = z'$ and from $F_1(x, y, z) = 3x = 3x' = F_1(x', y', z')$ it follows that $(x, y, z) = (x', y', z')$. Now suppose that $F_2(x, y, z)^2 + F_3(x, y, z)^2$ (and, therefore, $F_2(x', y', z')^2 + F_3(x', y', z')^2$) is nonzero. Then b, b', c, c' are all nonzero. Since $F_2(x, y, z) = F_2(x', y', z')$ and $F_3(x, y, z) = F_3(x', y', z')$ we get

$$\frac{b^2}{c} = \frac{F_2(x, y, z) + iF_3(x, y, z)}{2} = \frac{F_2(x', y', z') + iF_3(x', y', z')}{2} = \frac{(b')^2}{c'}$$

and

$$2bc = F_2(x, y, z)^2 + F_3(x, y, z)^2 = F_2(x', y', z')^2 + F_3(x', y', z')^2 = 2b'c'$$

This implies that

$$b^3 = \frac{b^2}{c}(bc) = \frac{(b')^2}{c'}b'c' = (b')^3.$$

Therefore $b' = \zeta^s b$ for some $s \in \{0, 1, 2\}$. Taking complex conjugation, we get that $c' = \zeta^{-s}c$. We conclude that (a, b, c) and (a', b', c') lie in the same orbit $\mathbb{Z}/3\mathbb{Z}$, which means that (x', y', z') is a permutation of (x, y, z) . □

One can also verify that the field of rational invariants $\mathbb{R}(x, y, z)^{\mathbb{Z}/3\mathbb{Z}}$ is generated by F_1, F_2, F_3 .

7.3.2 Coplanar Crater Patterns (Seven-Element Descriptor)

The development of p^2 invariants for a triad of conics lying on a plane follows similar arguments as for conics on a quadric, though it requires more care due to additional relationships between the seven elements in the descriptor. Therefore, recall from Eqs. 128–131 that the seven invariants consists of three invariant pairs, $(I_{ij}, I_{ji}), (I_{jk}, I_{kj}), (I_{ki}, I_{ik})$, and a 7th invariant for the entire triad I_{ijk} . Because they're constructed from a common set of crater observations, the two invariants within any given pair always undergo a common permutation. We also observe that I_{ijk} is already symmetric and invariant to the ellipse label order.

Simply by applying the results of the previous section, we can find six $\mathbb{Z}/3\mathbb{Z}$ -invariant functions from $F_\ell(I_{ij}, I_{jk}, I_{ki})$ and $F_\ell(I_{ji}, I_{kj}, I_{ik})$. The descriptiveness of these six invariants, however, is not optimal because members of a common pair are not forced to undergo the same permutation. Therefore, we introduce two new invariants.

Again making use of a DFT, define a_ℓ, b_ℓ, c_ℓ as

$$a_\ell = x_\ell + y_\ell + z_\ell \tag{147}$$

$$b_\ell = x_\ell + \zeta y_\ell + \zeta^2 z_\ell \tag{148}$$

$$c_\ell = x_\ell + \zeta^2 y_\ell + \zeta z_\ell \tag{149}$$

for $\ell = 1, 2$. Note that $c_\ell = \bar{b}_\ell$ where x_ℓ, y_ℓ, z_ℓ are real. A generator of $\mathbb{Z}/3\mathbb{Z}$ acts on (x_1, y_1, z_1) and (x_2, y_2, z_2) by the same cyclic permutation of coordinates, and it acts on b_1, b_2 with a scalar ζ and on c_1, c_2 with a scalar ζ^2 . We see, therefore, that $b_1 c_2$ is invariant. We define $G_1(x_1, y_1, z_1, x_2, y_2, z_2)$ and $G_2(x_1, y_1, z_1, x_2, y_2, z_2)$ be the real and imaginary part of $b_1 c_2$, which we compute as

$$\begin{aligned} b_1 c_2 &= (x_1 + \zeta y_1 + \zeta^2 z_1)(x_2 + \zeta^2 y_2 + \zeta z_2) = \\ &= (x_1 x_2 + y_1 y_2 + z_1 z_2) + \zeta(x_1 z_2 + y_1 x_2 + z_1 y_2) + \zeta^2(x_1 y_2 + y_1 z_2 + z_1 x_2) \end{aligned}$$

We may isolate the real part to obtain G_1

$$\begin{aligned} G_1(x_1, y_1, z_1, x_2, y_2, z_2) &= \\ &= (x_1 x_2 + y_1 y_2 + z_1 z_2) - \frac{1}{2}(x_1 z_2 + y_1 x_2 + z_1 y_2 + x_1 y_2 + y_1 z_2 + z_1 x_2) \\ &= \frac{3}{2}(x_1 x_2 + y_1 y_2 + z_1 z_2) - \frac{1}{2}(x_1 + y_1 + z_1)(x_2 + y_2 + z_2) \end{aligned}$$

and the imaginary part to obtain G_2

$$\begin{aligned} G_2(x_1, y_1, z_1, x_2, y_2, z_2) &= \frac{\sqrt{3}}{2} [(x_1 z_2 + y_1 x_2 + z_1 y_2) - (x_1 y_2 + y_1 z_2 + z_1 x_2)] \\ &= -\frac{\sqrt{3}}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} \end{aligned}$$

Lemma 2 Suppose that $x_\ell, y_\ell, z_\ell, x'_\ell, y'_\ell, z'_\ell \in \mathbb{R}$ for $\ell = 1, 2$ such that

$$\begin{aligned} F(x_1, y_1, z_1) &= F(x'_1, y'_1, z'_1) \\ F(x_2, y_2, z_2) &= F(x'_2, y'_2, z'_2) \\ G(x_1, y_1, z_1, x_2, y_2, z_2) &= G(x'_1, y'_1, z'_1, x'_2, y'_2, z'_2) \end{aligned}$$

Then

$$(x_1, x_2), (y_1, y_2), (z_1, z_2)$$

is a cyclic permutation of

$$(x'_1, x'_2), (y'_1, y'_2), (z'_1, z'_2)$$

Proof By Lemma 1 we know that (x'_1, y'_1, z'_1) is a cyclic permutation of (x_1, y_1, z_1) and (x'_2, y'_2, z'_2) is a cyclic permutation of (x_2, y_2, z_2) , i.e., $b'_1 = \zeta^s b_1$ and $b'_2 = \zeta^t b_2$ for some $s, t \in \{0, 1, 2\}$. It is not yet clear that this is the same permutation, i.e., whether $s = t$. From $G(x_1, y_1, z_1, x_2, y_2, z_2) = G(x'_1, y'_1, z'_1, x'_2, y'_2, z'_2)$ follows that $b_1 \bar{b}_2 = b_1 c_2 = b'_1 c'_2 = b'_1 \bar{b}'_2$. On the other hand, $b_1 \bar{b}_2 = b'_1 \bar{b}'_2 = (\zeta^s b_1)(\zeta^{-t} \bar{b}_2) = \zeta^{s-t} b_1 \bar{b}_2$. So either $b_1 = 0, b_2 = 0$, or $s = t$. If $t = s$, then we are done. If $b_1 = 0$ then $x_1 = y_1 = z_1 = x'_1 = y'_1 = z'_1$ and we are also done because (x'_2, y'_2, z'_2) is a cyclic permutation of (x_2, y_2, z_2) . The case $b_2 = 0$ goes similarly. \square

It follows, therefore, that there are many approaches to choose a descriptor for a set of three coplanar craters. We summarize three obvious choices, with the third being the one we usually recommend.

The most obvious approach is to form a seven-element descriptor as

$$[F(I_{ij}, I_{jk}, I_{ki}), F(I_{ji}, I_{kj}, I_{ik}), I_{ijk}]$$

As was noted above, the descriptiveness of such a scheme is not optimal, because this descriptor remains the same when the values of I_{ij}, I_{jk}, I_{ki} cyclically rotate while the values of I_{ji}, I_{kj}, I_{ik} stay the same. This means that 3 different configurations might have the same descriptor.

A second option is to form a nine-element descriptor

$$[F(I_{ij}, I_{jk}, I_{ki}), F(I_{ji}, I_{kj}, I_{ik}), \tilde{G}(I_{ij}, I_{jk}, I_{ki}, I_{ji}, I_{kj}, I_{ik}), I_{ijk}]$$

This descriptor is more descriptive but uses more space. Here, we have replaced the invariant G with \tilde{G} , which we define as

$$\tilde{G}(x_1, y_1, z_1, x_2, y_2, z_2) = \frac{G(x_1, y_1, z_1, x_2, y_2, z_2)}{\sqrt[4]{\left((x_1 - y_1)^2 + (y_1 - z_1)^2 + (z_1 - x_1)^2\right)\left((x_2 - y_2)^2 + (y_2 - z_2)^2 + (z_2 - x_2)^2\right)}}$$

This is an essential step, since G scales quadratically with $(x_1, y_1, z_1, x_2, y_2, z_2)$ while F scales linearly. Using G instead of \tilde{G} results in a poorly scaled descriptor that complicates (and sometimes precludes) nearest neighbor searches with efficient data structures. Conversely, the function \tilde{G} scales linearly. We note that \tilde{G} is not a rational function because of the 4-th root, but it does extend to a continuous function on \mathbb{R}^6 by defining the function to be 0 whenever $x_1 = y_1 = z_1$ or $x_2 = y_2 = z_2$.

A third option (and the one we suggest) is to use the seven-element descriptor

$$[F(I_{ij}, I_{jk}, I_{ki}), F_1(I_{ji}, I_{kj}, I_{ik}), \tilde{G}(I_{ij}, I_{jk}, I_{ki}, I_{ji}, I_{kj}, I_{ik}), I_{ijk}]$$

For generic configurations, this descriptor is as good as the nine-element descriptor. The subfield of $L = \mathbb{C}(x_1, y_1, z_1, x_2, y_2, z_2)$ generated by $F_\ell(x_1, y_1, z_1)$, $\ell = 1, 2, 3$, $F_1(x_2, y_2, z_2)$ and $G_\ell(x_1, y_1, z_1, x_2, y_2, z_2)$, for $\ell = 1, 2$ is equal to $K = \mathbb{C}(a_1, a_2, b_1^3, b_1 c_1, b_1 c_2, c_1 b_2)$. We observe that $L = K(b_1)$ and b_1 has degree three over K . So the degree of the field extension L/K is at most three, K is contained in the fixed field $L^{\mathbb{Z}/3\mathbb{Z}}$ and the degree of the extension $L/L^{\mathbb{Z}/3\mathbb{Z}}$ is three. This

implies that $K = L^{\mathbb{Z}/3\mathbb{Z}}$. In other words, every $\mathbb{Z}/3\mathbb{Z}$ -invariant rational function in $I_{ij}, I_{jk}, I_{ki}, I_{ji}, I_{kj}, I_{ik}$ is a rational function in the first 6 elements of the descriptor.

However, in the degenerate case $I_{ij} = I_{jk} = I_{ki} = t$ for some fixed t we have $G(I_{ij}, I_{jk}, I_{ki}, I_{ji}, I_{kj}, I_{ik}) = 0$. In that case, the descriptor cannot distinguish between different configurations of the same pattern (e.g., a pattern and its mirror). This is to be expected, since the degenerate case corresponds to equally sized craters around an equilateral triangle. Here, while there may be enough information to uniquely identify the triad (i.e., match the triad descriptor to the database), there is not sufficient information to disambiguate the specific crater labels with the projected crater rims alone. No special action is required, since this ambiguity is naturally handled in the pattern verification process (Section 9).

7.4 Remarks on Choosing a Descriptor Convention

In the sections above we introduce descriptors based on the projective invariants, sorted projective invariants, and on the p^2 invariants. Which one is best is often application dependent.

Descriptors built directly on the projective invariants (Section 7.1) have better matching performance when presented with noisy data (see left-hand plot of Fig. 19), though they require the index be searched three times per observed pattern (one for each possible permutation).

Conversely, the descriptors built with the sorted invariants (Section 7.2) or p^2 invariants (Section 7.3) only require one index search per observed pattern, but are more sensitive to measurement noise. As a consequence the sorted invariant and p^2 invariant descriptors are faster for low noise situations, but slower for high noise situations (see right-hand plot of Fig. 19). This may seem counter-intuitive. The explanation, however, is straightforward. As measurement noise increases, the sorted invariant and p^2 invariant descriptors must attempt more triads (on average, as compared to the projective invariant descriptor from Section 7.1) before finding a correct nearest neighbor match. Thus, with large amounts of measurement noise, the projective invariant descriptor from Section 7.1 tends to find a match sooner despite needing three index searches per triad—and finding a match sooner results in a lower (faster) run-time.

8 Building a Global Crater Index

Global lunar crater identification requires multi-scale indexing and careful catalog curation. When close to the Moon, a spacecraft sees small craters that are nearly coplanar. Conversely, when far from the Moon, a spacecraft sees larger craters distributed over the lunar sphere (not coplanar). This immediately suggests at least two crater indexes be built—one for small patterns of nearly coplanar craters and another for larger patterns of non-coplanar craters. In practice, we find that the most efficient real-time performance is achieved by creating a single index for the planar case and

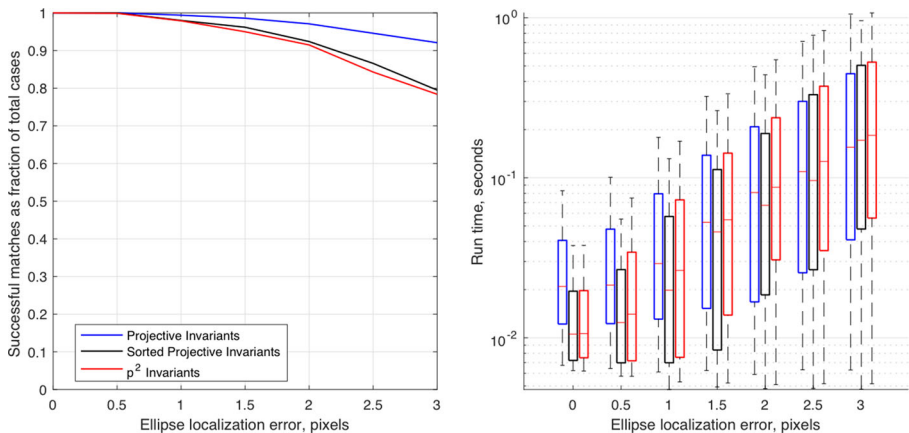


Fig. 19 A comparison of crater matching performance and execution speed for 1,000 random images of the Moon. These images correspond to a spacecraft at 600 km altitude placed randomly around the Moon (uniform distribution over the lunar sphere). Matching is performed using the three-element (non-coplanar) crater triad descriptor from Section 6.1.3. The left-hand plot shows the fraction of cases with a successful match. The balance of cases returned no match, and there were no cases where an incorrect match was returned. The right-hand plot shows execution time for the entire matching pipeline (conics as input, verified match as output) using the three different descriptors. These times correspond to non-optimized prototype code and should be taken to represent relative (rather than absolute) execution time

a hierarchy of indexes for the non-coplanar case. These indexes may be combined into a single large index or kept as separate indexes. We find the latter to often be the better choice.

The lunar crater database from Ref. [119] contains 1.3M craters over about 1–2 km in diameter. It is immediately obvious that consideration of all the $\binom{1.3M}{3} = 3.66 \times 10^{17}$ combinations of possible crater triads is not reasonable. This further motivates construction of a hierarchy of indexes with different scales to better manage the combinations of craters that could plausibly be observed at the same time. A visualization of the local and global crater indexes used in this work is shown in Fig. 20, where the index spaces are superimposed on the lunar crater density data from Ref. [119].

We propose the Hierarchical Equal Area isoLatitude Pixelization (HEALPix)⁴ framework [45] be used to subdivide the lunar surface into equal area regions, which we refer to as *surface pixels*. HEALPix was developed for subdividing the celestial sphere in support of science objectives for the Wilkinson Microwave Anisotropy Probe (WMAP) mission [10], and has since found widespread use for analysis of data from other NASA and ESA missions [e.g., Cosmic Background Explorer (COBE), Planck]. It is also used for managing the creation of star quadrilaterals in

⁴Software implementations of HEALPix in C++, FORTRAN, and Python are freely available online at <https://healpix.sourceforge.io>.

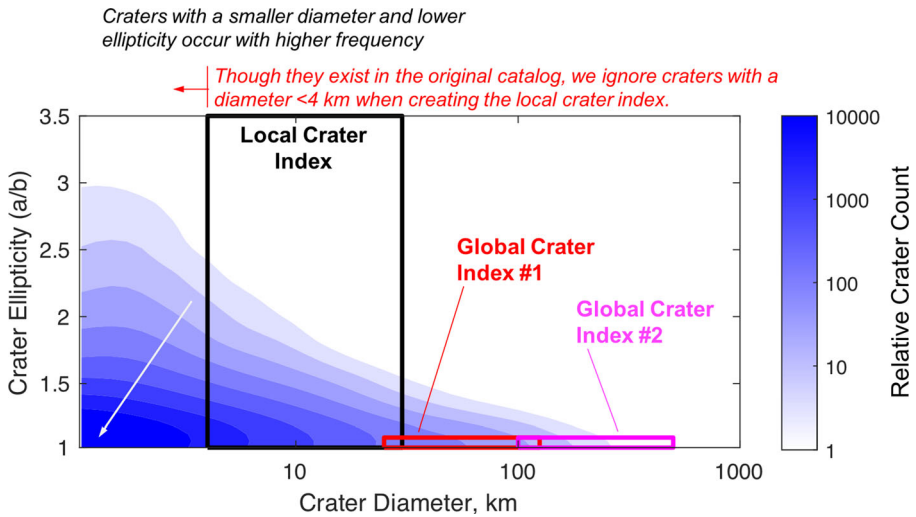


Fig. 20 Relative lunar crater density as a function of diameter and ellipticity. Areas of darker blue indicate more craters per unit area. Overlaid boxes show regions from which craters are drawn to for the local and global crater indexes

the present state-of-the-art for star identification, calibration, and alignment of astrometric images [71]. We believe our work to be the first application of HEALPix to the management of lunar surface features.

All indexes are built using the same fundamental approach, but at different scales. The procedure is as follows:

First, the lunar surface is tiled into $N_{pix} = 12(2^{2k})$ surface pixels of equal area. Next, a list is constructed for each pixel containing the catalog entries for craters within a specified size range (e.g., minimum/maximum diameter) and whose catalog fit was constructed using at least 90% of the rim's circumference.

Second, given a list of usable craters in each HEALPix surface pixel, we loop through all the pixels to create crater triads. At each pixel, we consider craters from the 3×3 HEALPix grid centered about the reference pixel. From these 9 surface pixels, all possible triads are formed where (1) craters do not intersect one another and (2) the triad center lies within the reference (center) surface pixel. Valid crater triads are arranged in a clockwise order and the scale-appropriate projective invariant descriptors are computed. To better illustrate this, Fig. 21 shows an example 3×3 HEALPix grid (region of support for surface pixel 6318) overlaid on an example Metric Camera image from Apollo 17. For further intuition, a global visualization of the HEALPix grid and the crater density (i.e., number of craters per HEALPix surface pixel) is shown in Fig. 22.

Third, once the triads belonging to all surface pixels have been computed, the results are stored in an efficiently searchable data structure. The authors have found excellent performance using either a k -d tree [11] or n -d k -vector [6], though other reasonable choices exist. We briefly remark that the 1-d k -vector [93, 94] has seen

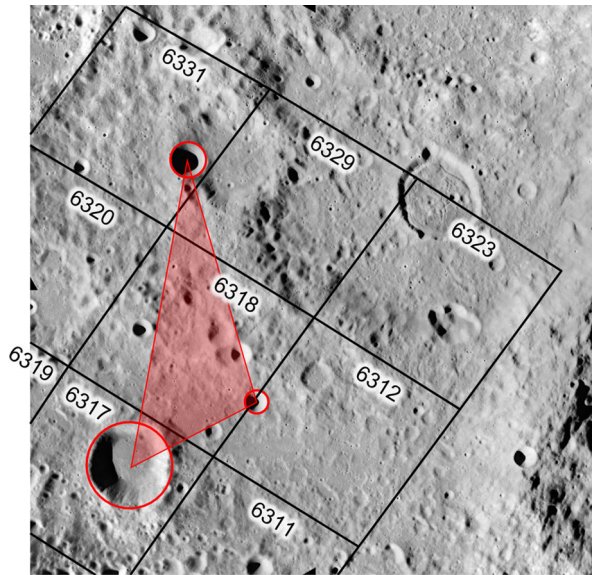


Fig. 21 Example lunar crater triad belonging to HEALPix 6318. Overlay is on Apollo Metric Camera image AS17-M-0684. (Credit for raw scans of Apollo flight film images: NASA/JSC/ASU. See [73, 121])

extensive use for star pattern matching in space applications [95, 134], and the n -d k -vector has been proposed for space applications as well [74]. The specific choice of data structure is not of primary concern in the present analysis so long as a reasonable selection is made.

8.1 Indexing Local Crater Patterns with Coplanar Invariants

To construct a global index of local-scale (small) crater patterns, we choose a HEALPix resolution of $k = 5$ for craters of diameter 4–30 km. This yields $N_{pix} = 12(2^{10}) = 12,288$ surface pixels, with each surface pixel having a surface area of approximately $3,086 \text{ km}^2$. There are 20,737 craters with catalog parameters supported by $> 90\%$ of the crater rim circumference and having a diameter between 4 km and 30 km. Rather than producing $\binom{20,737}{3} = 1.49 \times 10^{12}$ combinations, the HEALPix grouping strategy only produces the 4.8M crater triads that are nearly coplanar.

The local crater patterns are assumed coplanar and craters are allowed to follow an arbitrary elliptical shape. Thus, we construct a seven-element descriptor using an appropriate method from Section 7.

8.2 Indexing Global Crater Patterns with Non-Coplanar Invariants

For crater patterns that are regional or global in extent, we find it helpful to construct at least two indexes—though specific mission needs may require more or less. The extension to a single global index or many global indexes is trivial.

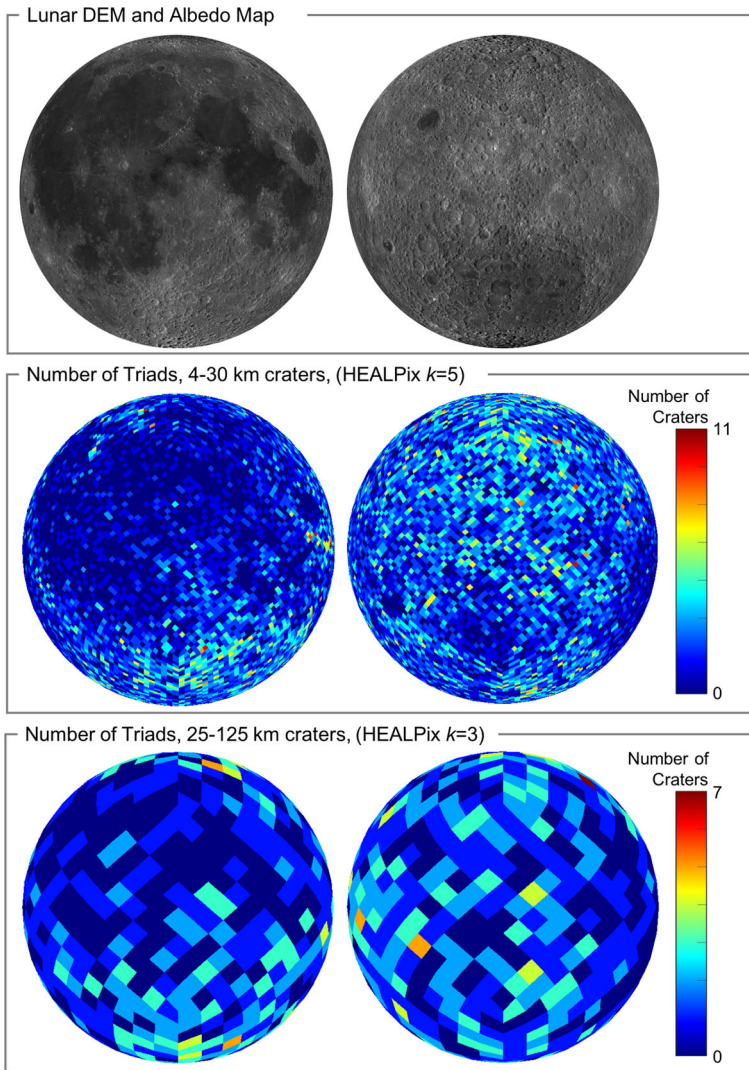


Fig. 22 Graphical depiction of the number of triads in each HEALPix surface pixel. Left column shows orthographic projection of lunar near side, while the right column shows the same for the lunar far side

The Moon is very nearly a sphere at the global level. Therefore, because the non-coplanar invariants require each pair of ellipses to lie on a common quadric surface, the craters must be nearly circular. The global lunar crater databases constructed here are built only from craters having an ellipticity of $a/b \leq 1.1$. Since the global patterns are assumed to lie on a nondegenerate quadric surface, we construct a three-element descriptor using an appropriate method from Section 7.

The first global crater index has a HEALPix resolution of $k = 3$ for craters of diameter 25–125 km. This yields $N_{pix} = 12(2^6) = 768$ surface pixels, with

each surface pixel having a surface area of approximately $49,390 \text{ km}^2$. The index is good for matching crater patterns at the regional level. There are 904 nearly circular craters ($a/b \leq 1.1$) with catalog parameters supported by $> 90\%$ of the crater rim circumference and having a diameter between 25 km and 125 km. Rather than producing $\binom{904}{3} = 1.23 \times 10^8$ combinations, the HEALPix grouping strategy only produces the 140,929 crater triads. The example crater triad in Fig. 16 (see Section 6.1) was recognized using this index.

The second global crater index has a HEALPix resolution of $k = 1$ for craters having a diameter over 100 km. This yields $N_{pix} = 12(2^2) = 48$ surface pixels, with each surface pixel having a surface area of approximately $790,300 \text{ km}^2$. The index is good for matching crater patterns at the global level, when nearly an entire hemisphere is visible in an image. There are 31 nearly circular craters ($a/b \leq 1.1$) with catalog parameters supported by $> 90\%$ of the crater rim circumference and having a diameter over 100 km. Rather than producing $\binom{31}{3} = 4,495$ combinations, the HEALPix grouping strategy only produces the 707 crater triads.

8.3 Remarks on the Utility of Crater Index Hierarchies

The primary purpose of having more than one index is to apply coplanar invariants at the local scale (allowing for arbitrary elliptical crater shape) and non-coplanar invariants at the regional/global scale (where curvature of the Moon makes craters lie in substantially different planes).

In practice, the orbital regime is often known ahead of time (e.g., LLO, cislunar), allowing us to only search crater patterns in the index of appropriate scale. If, however, we are truly “lost-in-space” and do not know the orbital regime *a priori*, we have found that all indexes may be queried and only the correct one will produce a match. Thus, the hierarchy of indexes allows us to exploit mission-specific information when it is available, but does not necessarily require this information.

While this work builds a truly global index for both local crater patterns and global crater patterns, it may sometimes be desirable to only include craters visible along a reference trajectory.

9 Matching Observed Crater Patterns to a Pre-Built Index

When supplied an image of the lunar surface, the objective is to recognize a pattern of observed craters using the image conics corresponding to the projection of the crater rims.

This is accomplished using a straightforward matching and verification procedure. First, a crater detection algorithm (CDA) produces an ellipse fit to a crater rims in an image. Second, we consider triads of these image conics, cycling through combinations using the Enhanced Pattern Shifting (EPS) method [5]. For each crater triad, we compute the appropriate descriptor (see Section 7) and query the corresponding pre-built index to find potential matches (e.g., best $N \geq 1$ matches). For each crater

match hypothesis, we compute the unknown camera position (see Section 9.1). After first checking to ensure the position estimate is not inside the Moon (as would happen if we accidentally match to a pattern on the wrong side of the Moon), we then reproject the expected crater rim locations in the image using Eq. 47. The match is verified by comparing the observed and expected crater rims using a new distance metric (see Section 9.2). Previous methods only compared crater centers and thus required 4–5 correct correspondences to verify a match. Comparing the craters rims is more informative, allowing for pattern match verification with only three craters (thus allowing crater identification above more sparsely cratered terrain).

If a pattern match hypothesis is verified by this procedure, we declare this the solution and terminate the search. If a pattern match hypothesis is not verified we consider another hypothesis for that triad or move to the next triad in the EPS sequence. If we reach the end of the EPS sequence with no hypothesis being verified, we declare that no match is possible and terminate the search.

The subsections that follow describe the mathematics for pose estimation from corresponding non-coplanar conics and a crater rim distance metric. Both of these algorithms are novel.

9.1 Pose from Non-Coplanar Conics in Correspondence

The usual approach for computing pose from matched craters only uses the coordinates of the crater center. Here, it is essential to remember that the center of an image ellipse does not generally produce a line-of-sight vector to the center of a 3D ellipse (or circle). Some past crater identification pipelines consider this effect (e.g., [110]), but many do not. Regardless, there are a variety of algorithms one may use to compute pose from corresponding 2D image and 3D model points [4, 77]. Unfortunately, however, using only the crater center points neglects the substantial amount of valuable navigation information contained within the shape of the projected crater rims. We aim to address this problem here.

Were the 3D conics strictly coplanar, there are a number of algorithms one could use to estimate pose from the projected image conics [67, 135, 141]. However, while we assume local crater patterns are coplanar for index building/matching, there is no reason to assume coplanarity for pose estimation or match verification. Furthermore, we require the ability to solve this problem for both local and global patterns—thus requiring a method for pose estimation from the projection of non-coplanar conics.

The literature discussing pose estimation from non-coplanar conics is limited. The most common approach is to look at a pair of ellipses in an image and construct a single-parameter family of possible poses from one of these two ellipses. This one parameter family is then numerically searched to find the best possible agreement with the second ellipse. This procedure was suggested in [85], which is generally cited as the solution to this problem.

The approach of [85], however, is not appropriate for spacecraft navigation as it (1) does not fully use the information content of every conic and (2) it only deals with a conic pair. Instead, we search for a solution that uses all the information from a

d -tuple (usually a triad) of image conics to solve for camera pose in the least squares sense.

Since the application at hand is lunar crater identification, we may simplify the problem by assuming the relative attitude is known. We consider this to be a reasonable assumption in practice since we have excellent knowledge of both the Moon’s attitude (from ephemeris data; e.g., SPICE kernels [1, 2]) and the spacecraft attitude (from star trackers [81, 82]). There are two observations that can now be made. First, we can think of no plausible failure mode (where recovery is still possible) in which the spacecraft has no knowledge of time (necessary for finding lunar attitude from SPICE kernels) or of inertial attitude. Second, we observe that this still formally qualifies as a lost-in-space problem since it assumes no knowledge of the spacecraft translational states (position or velocity). Moreover, the “known” attitude comes from the star tracker and there are many well-established lost-in-space star identification algorithms [118, 134]. Regardless, we presume the attitude transformation matrix T_C^M is known, such that the pose estimation problem only requires a solution for the selenographic camera location r_M .

Therefore, let us begin with consideration of the projection of a single conic as described by Eq. 58. Now, substitute for H_{C_i} from Eq. 50 and expand to find

$$H_{M_i}^T B_i H_{M_i} - H_{M_i}^T B_i r_M k^T - k r_M^T B_i H_{M_i} + r_M^T B_i r_M k k^T = s_i C_i \tag{150}$$

where

$$B_i = T_M^C K^T A_i K T_C^M \tag{151}$$

Recall here that A_i describes the measured image conic, C_i describes the 3D crater conic in the i th crater’s ENU frame, T_C^M is presumed known, and $k^T = [0 \ 0 \ 1]$. Thus, the only unknown in Eq. 150 is r_M .

Now, recalling (34) for H_{M_i} , rearrange the left-hand side into matrix form

$$\begin{bmatrix} S^T T_{E_i}^M B_i T_M^{E_i} S & S^T T_{E_i}^M B_i (p_{M_i} - r_M) \\ (p_{M_i} - r_M)^T B_i^T T_M^{E_i} S & (p_{M_i} - r_M)^T B_i (p_{M_i} - r_M) \end{bmatrix} = s_i C_i \tag{152}$$

where S is from Eq. 35. It is immediately evident that the upper-left 2×2 submatrix is independent of r_M , the lower-right element is quadratic in r_M , and the remaining terms are linear in r_M . Therefore, we first find the unknown scalar s_i using only the upper-left 2×2 submatrix. Using the Frobenius norm, compute \hat{s}_i as

$$\hat{s}_i = \arg \min_{s_i} \left\| S^T T_{E_i}^M B_i T_M^{E_i} S - s_i S^T C_i S \right\|_F^2 \tag{153}$$

which has the least squares solution

$$\hat{s}_i = \frac{\text{vec}(S^T C_i S)^T \text{vec}(S^T T_{E_i}^M B_i T_M^{E_i} S)}{\text{vec}(S^T C_i S)^T \text{vec}(S^T C_i S)} \tag{154}$$

With the scale s_i known, we can now take the upper-right 2×1 submatrix to form the linear system for all of the observed conics

$$\begin{bmatrix} S^T T_{E_1}^M B_1 \\ \vdots \\ S^T T_{E_n}^M B_n \end{bmatrix} r_M = \begin{bmatrix} S^T T_{E_1}^M B_i p_{M_1} - \hat{s}_i S^T C_1 k \\ \vdots \\ S^T T_{E_n}^M B_n p_{M_n} - \hat{s}_n S^T C_n k \end{bmatrix} \tag{155}$$

This may be solved for the selenographic camera location r_M in the least squares sense.

9.2 Comparing Two Image Conics

In order to verify a crater match hypothesis, we require a measure of the distance between two conics. For crater pattern verification, this distance of interest is usually between the image ellipse we expect from the projection of a crater’s rim (\mathcal{A}_i) and the image ellipse we measure of the same crater’s rim ($\tilde{\mathcal{A}}_i$). For the moment, however, we will briefly discuss how to compute the distance between two arbitrary ellipses: \mathcal{A}_i and \mathcal{A}_j .

Given two ellipses in an image, \mathcal{A}_i and \mathcal{A}_j , we seek a scalar distance metric $d(\mathcal{A}_i, \mathcal{A}_j)$ that satisfies the three usual axioms for a distance metric [32]

1. Minimality: $d(\mathcal{A}_i, \mathcal{A}_j) = 0$ iff $\mathcal{A}_i = \mathcal{A}_j$. That is, the distance between an ellipse and itself is zero.
2. Symmetry: $d(\mathcal{A}_i, \mathcal{A}_j) = d(\mathcal{A}_j, \mathcal{A}_i)$. That is, the distance from \mathcal{A}_i to \mathcal{A}_j is the same as the distance from \mathcal{A}_j to \mathcal{A}_i .
3. Triangle Inequality: $d(\mathcal{A}_i, \mathcal{A}_j) \leq d(\mathcal{A}_i, \mathcal{A}_k) + d(\mathcal{A}_k, \mathcal{A}_j)$

as well as a fourth axiom unique to this application

4. Similarity Invariance: $d(\mathcal{A}_i, \mathcal{A}_j) = d(S[\mathcal{A}_i], S[\mathcal{A}_j])$, where $S[\cdot]$ is a similarity transformation. That is, the distance between \mathcal{A}_i to \mathcal{A}_j should not change if the two ellipses undergo a common translation, rotation, or scaling in the image (i.e., undergo a common similarity transformation). See Fig. 23.

After reviewing the literature, we found numerous approaches for comparing ellipses, but all fail to meet one (or more) of the above four axioms. Thus, after a brief review of existing techniques, a novel method is proposed.

There exist various *ad hoc* methods based on the explicit ellipse parameters (a , b , x_c , y_c , and ψ) [30, 113] or by measuring the distance only at one specific point (e.g., via the Hausdorff distance [132, 136]), though none of these are similarity invariants. Moreover, these methods have changing geometric meaning as the two ellipses change shape and relative orientation. While not a violation of one of the axioms, it is not a desirable attribute.

Another approach is to apply a common normalization to the implicit representation of both conics (e.g., $\det(\mathcal{A}_i) = 1$) and then compute the Frobenius norm of their

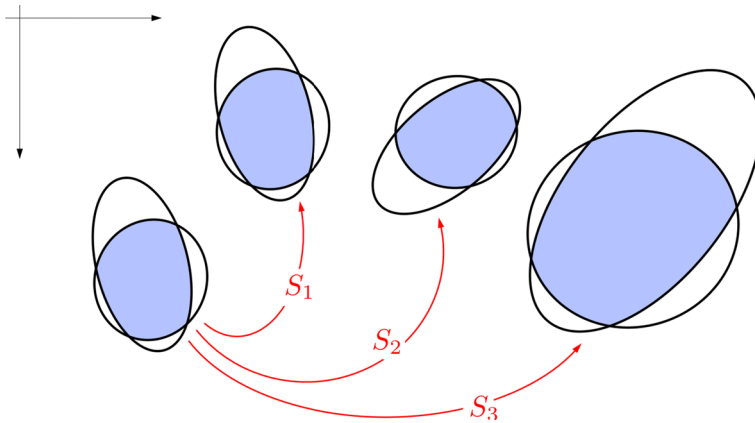


Fig. 23 Examples of an ellipse pair undergoing a simiarity transformation. The left-most ellipse pair may be translated (S_1); translated and rotated (S_2); or translated, rotated, and scaled (S_3). Under the similarity invariance axiom for an ellipse pair distance metric, all four of these ellipse pairs should produce the same value for distance since they all have the same relative geometry. Observe that all four of these ellipse pairs have the same Jaccard distance and Gaussian angle. The area of ellipse intersection (i.e., ellipse overlap) is shaded in light blue

difference [59]. There are various normalizations one could choose for the matrix A , as summarized in [66]. Nevertheless, such a comparison metric is given by

$$d_F(A_i, A_j) = \|A_i - A_j\|_F^2 \tag{156}$$

though the specific geometric meaning of this is not clear. This metric also fails to meet the similarity invariance axiom.

We also considered computing the distance as the line integral about one ellipse of the Euclidean distance to the other ellipse. While this admits an elegant and efficient solution with many interesting properties, it fails to meet both the symmetry and similarity invariance axioms. To meet the symmetry requirement, it would be possible to compute this metric in both directions and sum the result, though this is expensive. Regardless, a distance computed in this fashion will still fail to meet the similarity invariance axiom. The method was thus abandoned for the present application.

The deficiencies of the above methods motivate the need for another metric that satisfies all four distance axioms, has a geometrically consistent meaning, and possesses well-understood statistics. There are at least two such distance metrics: one using the Jaccard distance and another by interpreting the ellipse as a multivariate Gaussian.

9.2.1 Ellipse Pair Distance Metric with the Jaccard Index

The Jaccard index is the ratio of an intersection to a union. In this case, we take it to be the ratio of the areas of the ellipse intersection to the ellipse union,

$$J(A_i, A_j) = \frac{\text{Area}(A_i \cap A_j)}{\text{Area}(A_i \cup A_j)} \tag{157}$$

The Jaccard distance, which is well-known to satisfy all three of the classical distance metric axioms [79], is computed as

$$d_J(\mathcal{A}_i, \mathcal{A}_j) = 1 - J(\mathcal{A}_i, \mathcal{A}_j) = 1 - \frac{\text{Area}(\mathcal{A}_i \cap \mathcal{A}_j)}{\text{Area}(\mathcal{A}_i \cup \mathcal{A}_j)} \tag{158}$$

It is straightforward to see that this metric is also satisfies our similarity invariance axiom.

The only difficulty with the Jaccard distance is the quick computation of the intersection area. Although it is possible to analytically compute the area of ellipse overlap [60], doing so requires the separate consideration of nine different relative ellipse configurations. Such algorithm branching is generally undesirable. Fortunately, we recall that the ellipses in question represent regions of substantial extent in a digital image. Thus, the Jaccard distance may be approximated as the ratio of the number of pixels inside both ellipses to the number of pixels inside either ellipse. We may quickly count the pixels inside \mathcal{A}_i by finding those pixels that satisfy the inequality $\bar{\mathbf{u}}^T \mathbf{A}_i \bar{\mathbf{u}} < 0$, which is a simple and parallelizable computation. Once we count the pixels in this way, computing the Jaccard distance is trivial.

9.2.2 Ellipse Pair Distance Metric with the Gaussian Angle

It is possible to interpret the 2D ellipse \mathcal{A}_i as a bivariate Gaussian probability density function. If that interpretation is used for both \mathcal{A}_i and \mathcal{A}_j , the distance between two ellipses may be computed using any number of measures for comparing Gaussian PDFs. The authors of [144] follow this interpretation and consider the use of the Kullback-Leibler Divergence (KLD) and variants of the Wasserstein distance. The KLD, however, does not satisfy either the symmetry or triangle inequality axioms. The Wasserstein distance meets the first three axioms, but is not invariant under scaling (hence, fails the similarity invariance axiom—though modifications to the classical metrics may alleviate this shortcoming).

We introduce a different distance metric also based on the interpretation of \mathcal{A}_i as a bivariate Gaussian. Therefore, begin by subdividing the conic locus matrix \mathbf{A}_i describing the image ellipse \mathcal{A}_i as

$$\mathbf{A}_i \propto \begin{bmatrix} \mathbf{Y}_i & -\mathbf{Y}_i \mathbf{y}_i \\ -\mathbf{y}_i^T \mathbf{Y}_i^T & (\mathbf{y}_i^T \mathbf{Y}_i \mathbf{y}_i - 1) \end{bmatrix} \tag{159}$$

and it follows that that $\bar{\mathbf{u}}_i^T \mathbf{A}_i \bar{\mathbf{u}}_i = 0$ and that

$$(\mathbf{u}_i - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u}_i - \mathbf{y}_i) = 1 \tag{160}$$

With this particular scaling of \mathbf{A}_i , we immediately recognize the 2×2 submatrix \mathbf{Y}_i as the shape of the ellipse

$$\mathbf{Y}_i = \begin{bmatrix} \cos \psi_i & -\sin \psi_i \\ \sin \psi_i & \cos \psi_i \end{bmatrix} \begin{bmatrix} 1/a_i^2 & 0 \\ 0 & 1/b_i^2 \end{bmatrix} \begin{bmatrix} \cos \psi_i & \sin \psi_i \\ -\sin \psi_i & \cos \psi_i \end{bmatrix} \tag{161}$$

and $\mathbf{y}_i^T = [u_{c_i}, v_{c_i}]$ as the image coordinates of the ellipse center. If we interpret the ellipse as the 1σ isofrequency contour of a bivariate Gaussian distribution (i.e., the so-called 1σ error ellipse), the ellipse \mathcal{A}_i corresponds to the distribution $N(\mathbf{y}_i, \mathbf{Y}_i^{-1})$ with probability density

$$p_{\mathcal{A}_i}(\mathbf{u}) = \frac{|\mathbf{Y}_i|}{2\pi} \exp \left[-\frac{1}{2}(\mathbf{u} - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u} - \mathbf{y}_i) \right] \tag{162}$$

Therefore, given two ellipses \mathcal{A}_i and \mathcal{A}_j , we may compute the distance between these two ellipses as the ‘‘angle’’ between $p_{\mathcal{A}_i}(\mathbf{u})$ and $p_{\mathcal{A}_j}(\mathbf{u})$.⁵ Such an angle is computed as the inner product

$$\cos \theta = \frac{\int p_{\mathcal{A}_i}(\mathbf{u}) p_{\mathcal{A}_j}(\mathbf{u}) d\mathbf{u}}{\sqrt{\int p_{\mathcal{A}_i}^2(\mathbf{u}) d\mathbf{u} \int p_{\mathcal{A}_j}^2(\mathbf{u}) d\mathbf{u}}} \tag{163}$$

To compact integral notation in the equations that follow, we will no longer explicitly write the argument of integration. To proceed, we compute a distance metric as

$$d_{GA}(\mathcal{A}_i, \mathcal{A}_j) = \theta = \arccos \left[\frac{\int p_{\mathcal{A}_i} p_{\mathcal{A}_j}}{\sqrt{\int p_{\mathcal{A}_i}^2 \int p_{\mathcal{A}_j}^2}} \right] \tag{164}$$

From inspection, this clearly satisfies the minimality and symmetry conditions. Satisfaction of the triangle inequality follows from the Cauchy–Schwarz inequality.

We compute

$$\int p_{\mathcal{A}_i}^2 = \frac{|\mathbf{Y}_i|^2}{4\pi^2} \int \exp \left[-\frac{1}{2}(\mathbf{u} - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u} - \mathbf{y}_i) \right] = \frac{|\mathbf{Y}_i|^2}{4\pi^2} \cdot \frac{2\pi}{|\mathbf{2Y}_i|} = \frac{|\mathbf{Y}_i|}{8\pi} \tag{165}$$

To find the numerator, we first write

$$\int p_{\mathcal{A}_i} p_{\mathcal{A}_j} = \frac{|\mathbf{Y}_i| |\mathbf{Y}_j|}{4\pi^2} \int \exp \left[-\frac{1}{2} \left((\mathbf{u} - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u} - \mathbf{y}_i) + (\mathbf{u} - \mathbf{y}_j)^T \mathbf{Y}_j (\mathbf{u} - \mathbf{y}_j) \right) \right] \tag{166}$$

Now, considering the term inside the exponential,

$$(\mathbf{u} - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u} - \mathbf{y}_i) + (\mathbf{u} - \mathbf{y}_j)^T \mathbf{Y}_j (\mathbf{u} - \mathbf{y}_j) = (\mathbf{u} - \mathbf{z})^T (\mathbf{Y}_i + \mathbf{Y}_j) (\mathbf{u} - \mathbf{z}) + c \tag{167}$$

where

$$\mathbf{z} = (\mathbf{Y}_i + \mathbf{Y}_j)^{-1} (\mathbf{Y}_i \mathbf{y}_i + \mathbf{Y}_j \mathbf{y}_j) \tag{168}$$

$$c = (\mathbf{y}_i - \mathbf{y}_j)^T \mathbf{Y}_i (\mathbf{Y}_i + \mathbf{Y}_j)^{-1} \mathbf{Y}_j (\mathbf{y}_i - \mathbf{y}_j) \tag{169}$$

⁵See [107] for more on the idea of the ‘‘angle’’ between two functions.

Thus, noting that c is a constant, we rewrite the integral as

$$\int \exp \left[-\frac{1}{2} \left((\mathbf{u} - \mathbf{y}_i)^T \mathbf{Y}_i (\mathbf{u} - \mathbf{y}_i) + (\mathbf{u} - \mathbf{y}_j)^T \mathbf{Y}_j (\mathbf{u} - \mathbf{y}_j) \right) \right] \quad (170)$$

$$= \int \exp \left[-\frac{1}{2} (\mathbf{u} - \mathbf{z})^T (\mathbf{Y}_i + \mathbf{Y}_j) (\mathbf{u} - \mathbf{z}) + c \right] \quad (171)$$

$$= \exp \left[-\frac{c}{2} \right] \int \exp \left[-\frac{1}{2} (\mathbf{u} - \mathbf{z})^T (\mathbf{Y}_i + \mathbf{Y}_j) (\mathbf{u} - \mathbf{z}) \right] \quad (172)$$

$$= \exp \left[-\frac{c}{2} \right] \frac{2\pi}{|\mathbf{Y}_i + \mathbf{Y}_j|} \quad (173)$$

Consequently,

$$\begin{aligned} \int p_{\mathcal{A}_i} p_{\mathcal{A}_j} &= \frac{|\mathbf{Y}_i||\mathbf{Y}_j|}{2\pi|\mathbf{Y}_i + \mathbf{Y}_j|} \exp \left[-\frac{c}{2} \right] \quad (174) \\ &= \frac{|\mathbf{Y}_i||\mathbf{Y}_j|}{2\pi|\mathbf{Y}_i + \mathbf{Y}_j|} \exp \left[-\frac{1}{2} (\mathbf{y}_i - \mathbf{y}_j)^T \mathbf{Y}_i (\mathbf{Y}_i + \mathbf{Y}_j)^{-1} \mathbf{Y}_j (\mathbf{y}_i - \mathbf{y}_j) \right] \end{aligned}$$

We may therefore compute the distance metric d_{GA} by substitution of Eqs. 165 and 174 into Eq. 164,

$$d_{GA}(\mathcal{A}_i, \mathcal{A}_j) = \arccos \left\{ \frac{4\sqrt{|\mathbf{Y}_i||\mathbf{Y}_j|}}{|\mathbf{Y}_i + \mathbf{Y}_j|} \exp \left[-\frac{1}{2} (\mathbf{y}_i - \mathbf{y}_j)^T \mathbf{Y}_i (\mathbf{Y}_i + \mathbf{Y}_j)^{-1} \mathbf{Y}_j (\mathbf{y}_i - \mathbf{y}_j) \right] \right\} \quad (175)$$

In addition to satisfying the four required axioms, one of the great advantages of this particular distance metric is that it may be analytically computed from the parameters of the two image ellipses \mathcal{A}_i and \mathcal{A}_j .

9.2.3 A Fast Method for Assessing Ellipse Correspondence

We propose to use the Gaussian angle distance metric as the criteria for evaluating a potential crater correspondence. In the case where \mathcal{A}_j is a perturbed version of \mathcal{A}_i , we find that, to an excellent approximation,

$$\frac{d_{GA}^2}{\sigma^2} \sim \chi_4^2 \quad (176)$$

where

$$\sigma \approx \frac{0.85}{\sqrt{a_i b_i}} \sigma_{img} \quad (177)$$

The validity of this approximation is illustrated in Fig. 24 for both a nearly circular crater and a very elliptical crater.

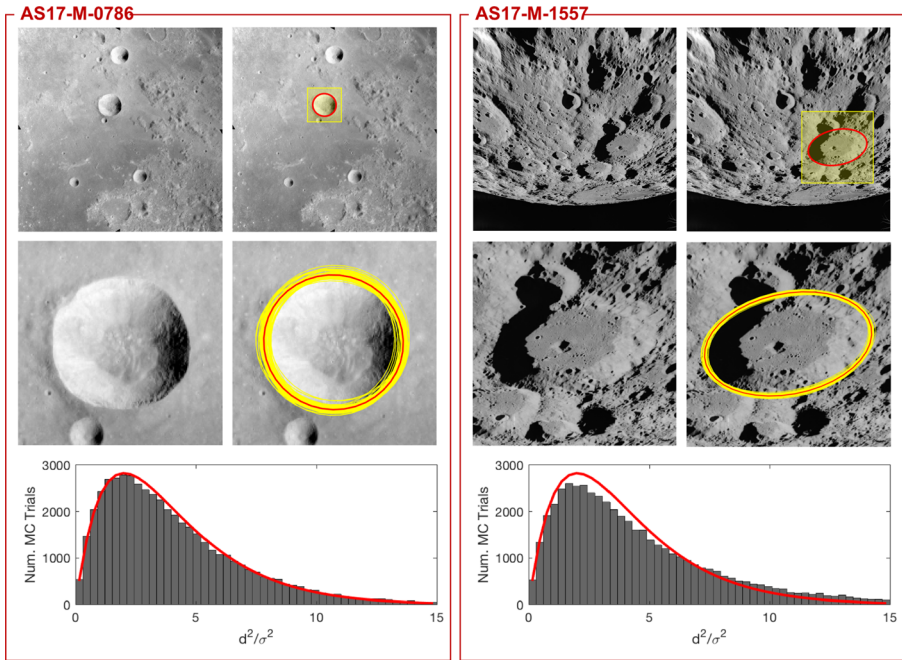


Fig. 24 Example Monte Carlo (MC) results (50,000 runs) for crater contour correspondence check using Metric Camera images from Apollo 17. The left frame shows results on image AS17-M-0786 collected on 12 December 1972 (17:31:56 UTC), while the right frame shows results on image AS17-M-1557 collected on 13 December 1972 (16:32:57 UTC). The top pair of images show the original Apollo image (left) and with the overlay of a red elliptical crater rim and yellow region of interest (right). The bottom pair of images is the zoomed-in region of interest by itself (left) and with an overlay of the red circular crater rim and 50 yellow example crater rims from the Monte Carlo (right). The perturbed limb fits (e.g. those plotted in yellow) were generated by perturbing the true limb’s ellipse parameters. The histograms at the bottom show the statistic $d^2/\sigma^2 \sim \chi_4^2$. The red overlay on the histogram is the analytic χ_4^2 probability density function (PDF). (Credit for raw scans of Apollo flight film images: NASA/JSC/ASU. See [73, 121])

Since the distance metric is observed to follow a χ_4^2 distribution, we may construct a statistically-informed criterion for accepting a crater match hypothesis. This is accomplished by evaluating the quantile function for the χ_4^2 distribution. For example, the 99th percentile for the χ_4^2 distribution occurs at 13.277, and would lead to an acceptance criteria of

$$\text{Hypothesis}(\mathcal{A}_i = \tilde{\mathcal{A}}_i) \leftarrow \begin{cases} \text{Accept } d_{GA_i}^2/\sigma^2 \leq 13.277 \\ \text{Reject otherwise} \end{cases} \quad (178)$$

The appropriate percentile for accepting or rejecting craters is application dependent and left as a design variable left to the analyst.

Finally, when selecting the threshold for accepting crater matches, the user may wish to consider that very elliptical craters have heavier tails (as compared to a

true χ_4^2 distribution) — thus increasing the likelihood of rejecting a valid crater assignment. This may be seen in the right-hand example of Fig. 24. Except for very elliptical craters, the deviation from the χ_4^2 distribution is modest and numerical experiments can help the analyst decide if this is important for a specific application.

10 Numerical Results

10.1 Sensitivity to Crater Rim Fit and Viewing Geometry

We performed a Monte Carlo analysis to evaluate the sensitivity of the proposed crater matching methodology to differing levels of measurement noise and under different viewing geometries.

Therefore, as an example, consider a spacecraft placed randomly (with a uniform distribution) around the lunar sphere, thus creating a situation where we randomly image with equal probability any part of the lunar globe. We assume a camera with a full field-of-view (FOV) of about 73.7×73.7 deg (same as the Apollo metric camera [38, 143]) and with a $2,000 \times 2,000$ pixel focal plane array (creating a 4.8 MPixel image).

We now consider two parametric scenarios: (1) a nadir pointing camera with varying errors in crater rim localization and (2) a fixed level of error in crater rim localization and increasing off-nadir pointing angle. For the first scenario (nadir pointing images), let the error in the ellipse parameters a , b , u_c , v_c all vary by σ_{img} and range from 0 to 3 pixel. Subpixel estimation of these parameters is not uncommon since we generally fit an ellipse to hundreds of pixel measurements around the crater circumference. Thus, for the second scenario (off-nadir pointing images), we assume an error of 0.5 pixel in the ellipse parameters.

Performance for each case is summarized with the root-sum-square (RSS) error in the estimated position of the camera. In all cases the estimated position errors appeared to be unbiased and are not explicitly recorded here for brevity.

10.1.1 Local Crater Matching Results

Suppose the randomly placed spacecraft has an altitude of 150 km above the lunar surface. At this altitude, the camera views primarily local crater patterns. In this case, results for the two Monte Carlo scenarios are shown in Tables 2 and 3.

From Table 2 we observe that matching performance is not appreciably affected by ellipse localization error until these errors reach about 2–3 pixels. Even then, the degradation in performance is gradual. The dropping percentage of correct matches occurs because the closest (nearest neighbor) index match to the computed invariants is no longer correct. We could delay the onset of this effect by considering additional neighbors beyond just the single best match, but this would increase run-time. The reader should also be warned that these results do not suggest that crater localization error always needs to be better than 2–3 pixels. What matters is not the absolute rim localization error, but rather the rim localization error as compared to the size of the

Table 2 Monte Carlo results (100 trials) for local crater matching using seven-element descriptor

Ellipse Fit Error	Correct Matches	Incorrect Matches	No Match Found	Less Than Three Craters	RSS Camera Position Error
0.0 pixel	100	0	0	0	2.5×10^{-6} m
0.5 pixel	96	0	1	3	116 m
1.0 pixel	96	0	4	0	285 m
1.5 pixel	94	0	4	2	428 m
2.0 pixel	91	0	7	2	620 m
2.5 pixel	93	0	7	0	696 m
3.0 pixel	83	0	15	2	923 m

A nadir-pointing camera is placed randomly around the lunar globe at an altitude of 150 km, with ellipse fit error varying from 0 to 3 pixel

image craters (i.e., a 2 pixel error on a crater with a 10 pixel diameter is significant, while a 2 pixel error on a crater with a 500 pixel diameter is not significant).

Test cases are counted in the “No Match Found” column when we see at least three craters but the algorithm produces no matches. This can happen for a number of reasons. First, the acceptance threshold for a crater match was set at the 99th percentile (as in Eq. 178), which would suggest that $1 - 0.99^3 \approx 3\%$ of the triads should fail to match due to measurement noise (this acceptance threshold can be adjusted based on the analyst’s preferences). When there are more than three craters in an image, the likelihood of this happening becomes rather small (but is never zero). The second reason for no match is that no combination of observed craters corresponds to an entry in the index. Given the FOV of this particular camera, it is possible to see more than nine HEALPix surface pixels at one time—thus creating the possibility of observing three craters that do not correspond to a triad within the index.

From Table 3 we observe that off-nadir pointing has no meaningful effect (at least up to 30 deg) on our ability to recognize a crater pattern.

Table 3 Monte Carlo results (100 trials) for local crater matching using seven-element descriptor

Off-Nadir Angle	Correct Matches	Incorrect Matches	No Match Found	Less Than Three Craters	RSS Camera Position Error
0 deg	98	0	1	1	140 m
10 deg	97	0	3	0	147 m
20 deg	99	0	1	0	134 m
30 deg	96	0	4	0	178 m

A camera is placed randomly around the lunar globe at an altitude of 150 km, with an off-nadir pointing varying from 0 to 30 deg. Crater fit error of $\sigma = 0.5$ pixel on ellipse parameters

Table 4 Monte Carlo results (100 trials) for global crater matching using three-element descriptor

Ellipse Fit Error	Correct Matches	Incorrect Matches	No Match Found	Less Than Three Craters	RSS Camera Position Error
0.0 pixel	100	0	0	0	1.1×10^{-6} m
0.5 pixel	100	0	0	0	485 m
1.0 pixel	100	0	0	0	1,294 m
1.5 pixel	99	0	1	0	1,790 m
2.0 pixel	99	0	1	0	2,605 m
2.5 pixel	96	0	4	0	3,549 m
3.0 pixel	95	0	5	0	4,500 m

A nadir-pointing camera is placed randomly around the lunar globe at an altitude of 600 km, with ellipse fit error varying from 0 to 3 pixel

10.1.2 Global Crater Matching Results

Suppose the randomly placed spacecraft has an altitude of 600 km above the lunar surface. At this altitude, the camera views primarily regional/global crater patterns. In this case, results for the two Monte Carlo scenarios are shown in Tables 4 and 5.

We observe the same basic trend for the global pattern as for the local patterns. Matching performance degrades as we approach localization error of around 2–3 pixels, which occurs at the roughly same level of noise only because the 600 km altitude choice. We once again observe very little affect on matching performance due to off-nadir pointing.

Note that the camera location is computed in exactly the same way for the local and global matching examples (both use the algorithm from Section 9.1). The global camera location errors are larger because the images are taken from a higher altitude. Regardless, errors are still less than 1 km for the local crater patterns and less than 5 km for the global crater patterns—even with rim localization errors much larger than should be expected in practice.

Table 5 Monte Carlo results (100 trials) for global crater matching using three-element descriptor

Off-Nadir Angle	Correct Matches	Incorrect Matches	No Match Found	Less Than Three Craters	RSS Camera Position Error
0 deg	100	0	0	0	549 m
10 deg	99	0	1	0	539 m
20 deg	99	0	1	0	652 m
30 deg	97	0	2	1	866 m

A camera is placed randomly around the lunar globe at an altitude of 600 km, with an off-nadir pointing varying from 0 to 30 deg. Crater fit error of $\sigma = 0.5$ pixel on ellipse parameters

10.2 Local Matching on Clementine Images

We demonstrate the seven-element local matching descriptor on real images collected by the Clementine spacecraft's UV/Visible (UV/Vis) camera. During its 71 days in lunar orbit in 1994 [104], the Clementine mission collected millions of images of the lunar surface with a variety of instruments. One of these instruments was the UV/Visible camera, a 4.2×5.6 deg FOV camera with a 288×384 pixel CCD focal plane array [58, 70]. Images of the lunar surface are available at a variety of ranges and viewpoints and serve as an excellent source of data for testing our proposed crater identification algorithms.

Some example crater identification results are shown in Fig. 25. These results show good matching performance, both for nearly nadir pointing images (top two rows) and for images from oblique viewpoints (bottom row). For each example image, the best fit ellipse parameters (as manually obtained from the image) were perturbed by a Gaussian distribution with a standard deviation of $\sigma_{img} = 0.5$ pixel. The noise causes a different crater pattern to be matched first for each run, resulting in different crater triads shown in Fig. 25. In each case, the craters with the white outline (and with centers connected by the thin yellow line) were correctly matched

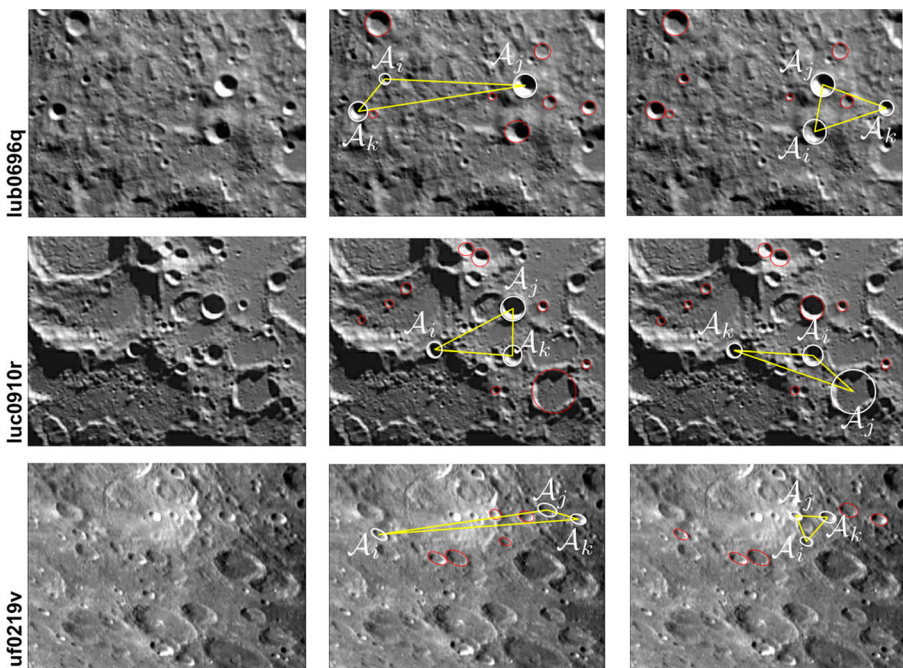


Fig. 25 Example crater identification on a sampling of images from the Clementine UV/Vis Camera. Each row shows results for a different Clementine image. Left frame is the raw image from [100]. Center and right frame show examples of successful (correct) crater triad matches in white, with centers connected by a yellow triangle to facilitate pattern visualization. Other craters (observed but not contributing to the matched triad) are shown in red. Matches were performed using the seven-element descriptor for local patterns

to the Robbins catalog [119] using the local crater index (HEALPix with $k = 5$) described in Section 8.

10.3 Global Matching on Synthetic Images on Reference Lunar Trajectory

We demonstrate the three-element global matching descriptor on synthetic images along a reference trajectory on synthetic images along a reference lunar trajectory.

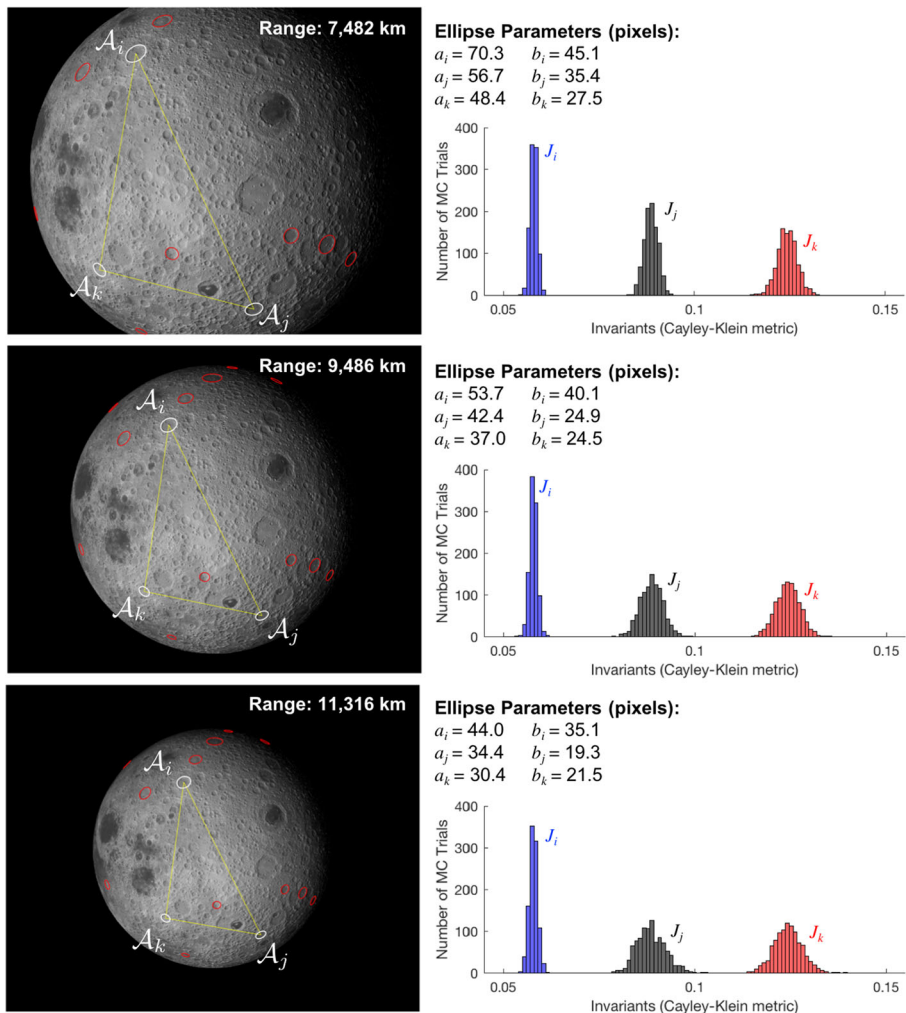


Fig. 26 Example synthetic images along a reference trajectory as the spacecraft moves away from the Moon (top to bottom). The craters used to compute invariants are outlined in white and a sampling of other craters (which are not used) are outlined in red. Monte Carlo results (1,000 trials) assume a $1-\sigma$ ellipse localization error of 1 pixel, leading to widening of invariant histograms as craters become smaller relative to the fixed error. Synthetic images of the Moon were produced using PANGU Planet Surface Simulation Software developed by the Space Technology Centre at the University of Dundee, Scotland [111]

The synthetic images were generated using PANGU, and have a resolution of $2,048 \times 2,592$ pixels with a 30 deg horizontal FOV. Three example images are shown in Fig. 26. For each image, the parameters of reference crater ellipse (a , b , u_c , and v_c) were then perturbed with uncorrelated Gaussian noise of $\sigma_{img} = 1.0$ pixel. This was repeated 1,000 times for each image (with each trial having different noise applied to the crater rim fits) to understand the sensitivity of the invariants to measurement noise. Histograms of the three invariants are also shown in Fig. 26. Since the measurement noise is fixed ($\sigma_{img} = 1.0$ pixel on a , b , u_c , and v_c), we see the histograms widen as the spacecraft gets farther away from the Moon (top to bottom in figure). This occurs because the size of the pattern shrinks relative to the fixed magnitude of ellipse localization error—hence the fixed error causes a greater perturbation of the crater rim geometry.

11 Conclusion

The identification of craters in a digital image of the lunar surface is a critical capability, both for terrain relative navigation (TRN) and for the registration of scientific images. In this work, we provide the first comprehensive and mathematically rigorous approach for image-based crater identification using concepts from invariant theory. We first show that crater rims tend to be elliptical in shape, suggesting that they are well modeled as a conic (e.g., circle, ellipse) lying on the lunar surface. We then show that there are no projective invariants for conics lying on the surface of an arbitrarily shaped celestial body, but there are invariants for regularly shaped bodies like the Moon. Specifically, for a d -tuple of conics, we show that there are $3d - 6$ algebraically independent projective invariants for conics lying on a nondegenerate quadric surface and there are $5d - 8$ algebraically independent projective invariants for conics lying on a plane.

With the number of algebraically independent projective invariants known, we then develop a practical means of computing these invariants from just the apparent crater rims in a digital image (i.e., from the perspective projection of the 3D conics). These invariants may be used to form a pattern descriptor that is always the same, regardless of camera viewpoint. Thus, these descriptors may be precomputed for known crater patterns and stored in a searchable index. We find that descriptors for a triad of craters provide good matching results.

When given an image with observed crater rims, a match hypothesis for the crater rim pattern may be found by a simple nearest neighbor search of the index. With the resulting image-to-map crater correspondence hypothesis, we show how to estimate the location of the camera in the least-squares sense using the three crater rims directly (instead of just the crater center coordinates). Finally, to verify the crater match and pose hypothesis, we reproject the expected crater rim contours into the images and compare these with the measured crater rims. This is accomplished with a new distance metric that compares the entire ellipse fit (instead of just the crater center coordinates). By verifying with a distance metric considering the full crater rim contours, we are able to verify a pattern using fewer craters than methods using only center coordinates.

These techniques are demonstrated in a variety of numerical experiments—both on synthetic and real images. For a camera placed randomly around the Moon with no *a priori* position knowledge, successful matches generally exceed 90% of the test cases. The remaining cases return “no match,” either because there were fewer than three craters in a particular image or because measurement noise corrupted the crater rim fits too much for unambiguous recognition. In no case was an incorrect match returned, a sign that the new distance metric can indeed verify match hypotheses with only three craters (instead of the usual ≥ 5 craters).

While this work represents a significant advance in methods for crater-based TRN, a great deal of work remains to be done. What follows are a few thoughts regarding the next steps:

1. While the repurposing of scientific crater catalogs for TRN is convenient, these catalogs were built with different objectives and are generally suboptimal for TRN applications. Thus, we suggest that the lunar exploration community would benefit greatly from a purpose-built TRN crater catalog. Such a catalog would be populated with only well-defined craters, with each having a shape and size consistent with what an automated CDA would produce.
2. We provide a complete set of independent projective invariants for crater patterns on the surface of the Moon in Section 6 (three invariants for a triad of conics on a quadric surface, seven invariants for a triad of conics on a plane). While there are no additional algebraically independent invariants, it is possible to construct other algebraically *dependent* invariants as functions of the ones provided in this work (e.g., the p^2 invariants in Section 7 are algebraic functions of the projective invariants from Section 6). There may exist other forms of these invariants that make lunar crater patterns more distinct or that have superior stability in the presence of measurement noise. Searching for such alternative formulations of the invariants from Section 6 is an interesting problem.
3. This work develops a global index of crater descriptors at three different scales using HEALPix. Most real missions likely do not need a global index—but, instead, just an index of craters along a reference trajectory. Although HEALPix may still be used to manage varying scales, there is a great deal of forward work in customizing the proof-of-concept outlined here to a mission-specific application. The best design for an actual in-flight index is not self evident and is the topic of ongoing work.
4. Not addressed in this work are crater detection algorithms (CDAs). Though many different CDAs exist, it is not clear which specific algorithms are best paired with the crater identification framework outlined here. Our crater identification algorithm is critically dependent on the localization of the best-fit ellipse to the crater rim, which is often not the metric used to design or evaluate CDAs. Thus, finding (or developing) an appropriate CDA is an obvious topic of follow-on work.

In conclusion, we find that crater-based TRN holds great promise for lost-in-space navigation near the Moon. We introduce a robust method for crater identification in

this manuscript, though we note that additional work is required to achieve a flight-capable system. Crater-based TRN is a rich field for future study.

Acknowledgements The authors thank Peter Sturm for insightful discussions related to the projection of conics that led to the technique presented in Section 9.1. The authors also thank Paul McKee and the NASA Johnson Space Center for assistance with production of the PANGU synthetic images. The following individuals provided valuable discussion that greatly improved the quality of this manuscript: Stuart Robbins, Joseph Mundy, Richard Hartley, Jeffrey Banks, James McCabe, Kristen Schell, Jacob Hinkle, Christopher D'Souza, Juno Woods, and Devin Renshaw. This work was partially supported by NASA award 80NSSC17M0027 (J.A.C.), NSF grant IIS 1837985 (H.D.), NSF grant DMS 2001460 (H.D.), and NASA Lunar Data Analysis Program grant 80NSSC17K0343 (R.W.).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Acton, C.: Ancillary data services of nasa's navigation and ancillary information facility. *Planet. Space Sci.* **44**(1), 65–70 (1996). [https://doi.org/10.1016/0032-0633\(95\)00107-7](https://doi.org/10.1016/0032-0633(95)00107-7)
2. Acton, C., Bachman, N., Semenov, B., Wright, E.: A look towards the future in the handling of space science mission geometry. *Planet. Space Sci.* **150**, 9–12 (2018). <https://doi.org/10.1016/j.pss.2017.02.013>
3. Adams, D., Criss, T., Shankar, U.: Passive optical terrain relative navigation using APLNav. In: *IEEE Aerospace Conference* (2008). <https://doi.org/10.1109/AERO.2008.4526303>
4. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 578–589 (2003). <https://doi.org/10.1109/TPAMI.2003.1195992>
5. Arnas, D., Fialho, M.A.A., Mortari, D.: Fast and robust kernel generators for star trackers. *Acta Astronaut.* **134**, 291–302 (2017). <https://doi.org/10.1016/j.actaastro.2017.02.016>
6. Arnas, D., Leake, C., Mortari, D.: The n-dimensional k-vector and its application to orthogonal range searching. *Appl. Math. Comput.* **372**, 125010 (2020). <https://doi.org/10.1016/j.amc.2019.125010>
7. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008). <https://doi.org/10.1016/j.cviu.2007.09.014>
8. Bellantoni, J.: Unidentified landmark navigation. *AIAA J.* **5**(8), 1478–1483 (1967). <https://doi.org/10.2514/3.4222>
9. Benedix, G., Norman, C., Bland, P., Towner, M., Paxman, J., Tan, T.: Automated detection of martian craters using a convolutional neural network. In: *49th Lunar and Planetary Science Conference* 2018, 2083 (2018)
10. Bennett, C.L.: et. al: The microwave anisotropy probe mission. *Astrophys. J.* **583**, 1–23 (2003). <https://doi.org/10.1086/345346>
11. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**(9), 509–517 (1975). <https://doi.org/10.1145/361002.361007>
12. Borkowski, K.M.: Accurate algorithms to transform geocentric to geodetic coordinates. *Bulletin Géodésique* **63**, 50–56 (1989). <https://doi.org/10.1007/BF02520228>
13. Bos, B.J., et al.: In-flight calibration and performance of the OSIRIS-REx touch and go camera system (TAGCAMS). *Space Sci. Rev.* **216**(71). <https://doi.org/10.1007/s11214-020-00682-x> (2020)
14. Botke, W., Love, S., Tytell, D., Glotch, T.: Interpreting the elliptical crater populations on mars, venus, and the moon. *Icarus* **145**, 108–121 (2000). <https://doi.org/10.1006/icar.1999.6323>

15. Bowring, B.R.: The accuracy of geodetic latitude and height equations. *Surv. Rev.* **28**(218), 202–206 (1985). <https://doi.org/10.1179/sre.1985.28.218.202>
16. Burns, J.B., Weiss, R.S., Riseman, E.M.: Geometric Invariance in Computer Vision, chap. The non-existence of general-case view-invariants, pp. 120–131. MIT Press, Cambridge (1992)
17. Chen, H.F., Belhumeur, P.N., Jacobs, D.W.: In search of illumination invariants. In: IEEE Conference on Computer Vision and Pattern Recognition (2000). <https://doi.org/10.1109/CVPR.2000.855827>
18. Cheng, Y., Ansar, A.: Landmark based position estimation for pinpoint landing on mars. In: IEEE International Conference on Robotics and Automation (2005). <https://doi.org/10.1109/ROBOT.2005.1570338>
19. Cheng, Y., Goguen, J., Johnson, A., Leger, C., Matthies, L., San Martin, M., Wilson, R.: The mars exploration rovers decent image motion estimation system. *IEEE Intell. Syst.* **19**(3), 13–21 (2004). <https://doi.org/10.1109/MIS.2004.18>
20. Cheng, Y., Miller, J.: Autonomous landmark based spacecraft navigation. In: AAS/AIAA Astrodynamics Specialist Conference, AAS 03-223 (2003)
21. Christian, J.: Accurate planetary limb localization for image-based spacecraft navigation. *J. Spacecr. Rocket.* **54**(3), 708–730 (2017). <https://doi.org/10.2514/1.A33692>
22. Christian, J.: Perspective projection of ellipses and ellipsoids with applications to spacecraft navigation. In: AAS/AIAA Astrodynamics Specialist Conference, AAS 17-800 (2017)
23. Christian, J.: Autonomous initial orbit determination with optical observations of unknown planetary landmarks. *J. Spacecr. Rocket.* **56**(1), 211–220 (2019). <https://doi.org/10.2514/1.A34259>
24. Christian, J., Benhacine, L., Hikes, J., D’Souza, C.: Geometric calibration of the orion optical navigation camera using star field images. *J. Astronaut. Sci.* **63**(4), 335–353 (2016). <https://doi.org/10.1007/s40295-016-0091-3>
25. Christian, J.A.: Optical navigation for a spacecraft in a planetary system. Ph.D. thesis, The University of Texas at Austin (2010)
26. Christian, J.A.: A tutorial on horizon-based optical navigation and attitude determination with space imaging systems. *IEEE Access*, 19819–19853. <https://doi.org/10.1109/ACCESS.2021.3051914> (2021)
27. Christian, J.A., Hong, L., McKee, P., Christensen, R., Crain, T.P.: Image-based lunar terrain relative navigation without a map: Measurements. *J. Spacecr. Rocket.* **58**(1), 164–181 (2021). <https://doi.org/10.2514/1.A34875>
28. Clemens, D.T., Jacobs, D.W.: Space and time bounds on indexing 3d models from 2d images. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(10), 1007–1017 (1991). <https://doi.org/10.1109/34.99235>
29. Collins, G.S., Elbeshausen, D., Davison, T.M., Robbins, S.J., Hynek, B.M.: The size-frequency distribution of elliptical impact craters. *Earth Planet. Sci. Lett.* **310**, 1–8 (2011). <https://doi.org/10.1016/j.epsl.2011.07.023>
30. Cuevas, E., González, M., Zaldívar, D., Pérez-Cisneros, M.: Multi-ellipses detection on images inspired by collective animal behavior. *Neural Comput. Appl.* **24**, 1019–1033 (2014). <https://doi.org/10.1007/s00521-012-1332-4>
31. Cui, P., Gao, X., Zhu, S., Shao, W.: Visual navigation using edge curve matching for pinpoint planetary landing. *Acta Astronaut.* **146**, 171–180 (2018). <https://doi.org/10.1016/j.actaastro.2018.02.033>
32. Cullinane, M.J.: Metric axioms and distance. *Math. Gaz.* **95**, 414–419 (2011). <https://doi.org/10.1017/S002557200003508>
33. DeLatte, D., Crites, S., Guttenberg, N., Yairi, T.: Automated crater detection algorithms from a machine learning perspective in the convolutional neural network era. *Adv. Space Res.* **64**(8), 1615–1628 (2019). <https://doi.org/10.1016/j.asr.2019.07.017>
34. Derksen, H., Kemper, G.: Computational Invariant Theory, Encyclopaedia of Mathematical Sciences, enlarged edn, vol. 130. Springer, Berlin (2015). <https://doi.org/10.1007/978-3-662-48422-7>. With two appendices by Vladimir L. Popov, and an addendum by Norbert A’Campo and Popov, Invariant Theory and Algebraic Transformation Groups, VIII
35. Dieudonné, J.A., Carrell, J.B.: Invariant theory, old and new, vol. 4. [https://doi.org/10.1016/0001-8708\(70\)90015-0](https://doi.org/10.1016/0001-8708(70)90015-0) (1970)
36. Dieudonné, J.A., Carrell, J.B.: Invariant Theory, Old and New. Academic Press, New York-London (1971)
37. Downes, L., Steiner, T., How, J.: Deep learning crater detection for lunar terrain relative navigation. In: AIAA SciTech Forum (2020). <https://doi.org/10.2514/6.2020-1838>
38. Edmundson, K., Alexandrov, O., Archinal, B., Becker, K., Kirk, R., Moratto, Z., Nefian, A., Richie, J., Robinson, M.: Photogrammetric processing of apollo 15 metric camera oblique images. In: The

- International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B4 (2016). <https://doi.org/10.5194/isprs-archives-XLI-B4-375-2016>
39. Elbeshhausen, D., Wünnemann, K., Collins, G.S.: The transition from circular to elliptical impact craters. *Journal of Geophysical Research: Planets* **118**, 2295–2309 (2013). <https://doi.org/10.1002/2013JE004477>
 40. Emami, E., Bebis, G., Nefian, A., Fong, T.: Automatic crater detection using convex grouping and convolutional neural networks. In: 11th International Symposium on Visual Computing (2015). https://doi.org/10.1007/978-3-319-27863-6_20
 41. Forsyth, D., Mundy, J., Zisserman, A., Coelho, C., Heller, A., Rothwell, C.: Invariant descriptors for 3-D object recognition and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(10), 971–991 (1991). <https://doi.org/10.1109/34.99233>
 42. Fulton, W., Harris, J.: Representation theory, Graduate Texts in Mathematics, vol. 129. Springer, New York (1991). <https://doi.org/10.1007/978-1-4612-0979-9>. A first course, Readings in Mathematics
 43. Gallier, J. *Geometric Methods and Applications*, 2nd edn. Springer, New York (2011)
 44. Gaskell, R.W., Barnouin-Jha, O.S., Scheers, D.J., et al.: Characterizing and navigating small bodies with imaging data. *Meteor. Planet. Sci.* **43**(6), 1049–1061 (2008). <https://doi.org/10.1111/j.1945-5100.2008.tb00692.x>
 45. Górski, K., Hivon, E., Banday, A., Wandelt, B., Hansen, F., Reinecke, M., Bartelmann, M.: Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *Astrophys. J.* **622**(2), 759–771 (2005). <https://doi.org/10.1086/427976>
 46. collab=Group, C.A.T.W.: Standard techniques for presentation and analysis of crater size-frequency data. *Icarus* **37**, 467–474 (1979). [https://doi.org/10.1016/0019-1035\(79\)90009-5](https://doi.org/10.1016/0019-1035(79)90009-5)
 47. Gupta, R., Hartley, R.I.: Linear pushbroom cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(9), 963–975 (1997). <https://doi.org/10.1109/34.615446>
 48. Haboush, W.J.: Reductive groups are geometrically reductive. *Ann. Math. (2)* **102**(1), 67–83 (1975). <https://doi.org/10.2307/1970974>
 49. Hanak, C., Crain, T., Bishop, R.: Crater identification algorithm for the lost in low lunar orbit scenario. In: AAS Guidance & Control Conference, AAS 10-052 (2008)
 50. Hanak, F.: Lost in low lunar orbit crater pattern detection and identification. Ph.D. thesis The University of Texas at Austin (2009)
 51. Hartley, R., Zisserman, A. *Multiple View Geometry*, 2nd edn. Cambridge University Press, Cambridge (2003)
 52. Head, J.W., Fassett, C.I., Kadish, S.J., Smith, D.E., Zuber, M.T., Neumann, G.A., Mazarico, E.: Global distribution of large lunar craters: Implications for resurfacing and impactor populations. *Science* **329**(5998), 1504–1507 (2010). <https://doi.org/10.1126/science.1195050>
 53. Heisterkamp, D.R., Bhattacharya, P.: Invariants of families of coplanar conics and their applications to object recognition. *J. Math. Imaging Vis.* **7**, 253–267 (1997). <https://doi.org/10.1023/A:1008230528693>
 54. Herrick, R.R., Schenk, P.M., Robbins, S.J.: Surveys of elliptical crater populations on the saturnian satellites, mercury, and mars. *Icarus* **220**(2), 297–304 (2012). <https://doi.org/10.1016/j.icarus.2012.05.027>
 55. Hiesinger, H.: Ages of mare basalts on the lunar nearside. *J. Geophys. Res. Planets* **105**, 29239–29275 (2000). <https://doi.org/10.1029/2000JE001244>
 56. Hilbert, D.: Ueber die Theorie der algebraischen Formen. *Math. Ann.* **36**(4), 473–534 (1890). <https://doi.org/10.1007/BF01208503>
 57. Hilbert, D.: Ueber die vollen Invariantensysteme. *Math. Ann.* **42**(3), 313–373 (1893). <https://doi.org/10.1007/BF01444162>
 58. Hillier, J.K., Buratti, B.J., Hill, K.: Multispectral photometry of the moon and absolute calibration of the clementine uv/vis camera. *Icarus* **141**, 205–225 (1999). <https://doi.org/10.1006/icar.1999.6184>
 59. Hosseinzadeh, M., Latif, Y., Pham, T., Suenderhauf, N., Reid, I.: Structure aware slam using quadrics and planes. In: Asian Conference on Computer Vision (2018). https://doi.org/10.1007/978-3-030-20893-6_26
 60. Hughes, G.B., Chraïbi, M.: Calculating ellipse overlap areas. *Comput. Vis. Sci.* **15**, 291–301 (2012). <https://doi.org/10.1007/s00791-013-0214-3>
 61. Humphreys, J.E.: *Linear Algebraic Groups*. Springer, New York (1975). Graduate Texts in Mathematics, No. 21
 62. Hungerford, T.W.: *Algebra*. Holt, Rinehart and Winston, Inc., New York-Montreal Que.-London (1974)

63. Jacobs, D.W.: What makes viewpoint-invariant properties perceptually salient? *J. Opt. Soc. Am. A* **20**(7), 1304–1320 (2003). <https://doi.org/10.1364/JOSAA.20.001304>
64. Johnson, A., Villaume, N., Umsted, C., Kourchians, A., Sternberg, D., Trawny, N., Cheng, Y., Geipel, E., Montgomery, J.: The Mars 2020 Lander Vision System Field Test. In: AAS Guidance, Navigation and Control Conference, AAS 20-105 (2020)
65. Johnson, R.A.: The conic as a space element. *Trans. Am. Math. Soc.* **15**, 335–368 (1914). <https://doi.org/10.2307/1988792>
66. Kanatani, K., Rangarajan, P.: Hyper least squares fitting of circles and ellipses. *Comput. Stat. Data Anal.* **55**, 2197–2208 (2011). <https://doi.org/10.1016/j.csda.2010.12.012>
67. Kannala, J., Salo, M., Heikkilä, J.: Algorithms for computing a planar homography from conics in correspondence. In: Proceedings of the British Machine Vision Conference (BVMC) (2006). <https://doi.org/10.5244/C.20.9>
68. Kariya, K., Ishida, T., Sawai, S., Fukuda, S.: Position and displacement estimation using crater-based line segments for pinpoint lunar landing. In: International Symposium on Space Flight Dynamics (2017)
69. Klear, M.: Pycda: An open-source library for automated crater detection. In: 9th Planetary Crater Consortium, 1806 (2018)
70. Kordas, J.F., Lewis, I.T., Priest, R.E., White, W.T., Nielsen, D.P., Park, H., Wilson, B.A., Shannon, M.J., Ledebuhr, A.G., Pleasance, L.D.: Uv/visible camera for the clementine mission. *Proc. SPIE*, 2478. <https://doi.org/10.1117/12.210943> (1995)
71. Lang, D., Hogg, D., Mierle, K., Blanton, M., Roweis, S.: Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *Astron. J.* **139**, 1782–1800 (2010). <https://doi.org/10.1088/0004-6256/139/5/1782>
72. Lang, S. *Algebra Graduate Texts in Mathematics*, 3rd edn., vol. 211. Springer, New York (2002). <https://doi.org/10.1007/978-1-4613-0041-0>
73. Lawrence, S.J., et al.: The apollo digital image archive: new research and data products. In: Proceedings of the NLSI Lunar Science Conference (2008)
74. Leake, C., Arnas, D., Mortari, D.: Non-dimensional star-identification. *Sensors* **20**, 2697 (2020). <https://doi.org/10.3390/s20092697>
75. Lefschetz, S.: *Algebraic Geometry*. Dover, Mineola (1953)
76. Lenz, R., Meer, P.: Point configuration invariants under simultaneous projective and permutation transformations. *Pattern Recogn.* **27**(11), 1523–1532 (1994). [https://doi.org/10.1016/0031-3203\(94\)90130-9](https://doi.org/10.1016/0031-3203(94)90130-9)
77. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* **81**, 155–166 (2009). <https://doi.org/10.1007/s11263-008-0152-6>
78. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: IEEE International Conference on Computer Vision (ICCV) (2011). <https://doi.org/10.1109/ICCV.2011.6126542>
79. Levandowski, M., Winter, D.: Distance between sets. *Nature* **234**, 34–35 (1971). <https://doi.org/10.1038/234034a0>
80. Levine, G.: A method of orbital navigation using optical sightings to unknown landmarks. *AIAA J.* **4**(11), 1928–1931 (1966). <https://doi.org/10.2514/3.3820>
81. Liebe, C.C.: Star trackers for attitude determination. *IEEE Aerosp. Electron. Syst. Mag.* **10**(6), 10–16 (1995). <https://doi.org/10.1109/62.387971>
82. Liebe, C.C.: Accuracy performance of star trackers — a tutorial. *IEEE Trans. Aerosp. Electron. Syst.* **38**(2), 587–599 (2002). <https://doi.org/10.1109/TAES.2002.1008988>
83. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004). <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
84. Lu, T., Hu, W., Liu, C., Yang, D.: Relative pose estimation of a lander using crater detection and matching. *Opt. Eng.* **55**(2), 023102 (2016). <https://doi.org/10.1117/1.OE.55.2.023102>
85. Ma, S.D.: Conics-based stereo, motion estimation, and pose determination. *Int. J. Comput. Vis.* **10**(1), 7–25 (1993). <https://doi.org/10.1007/BF01440844>
86. Maass, B., Woicke, S., Oliveria, W.M., Razgus, B., Krüger, H.: Crater navigation system for autonomous precision landing on the moon. *J. Guid. Control Dyn.* <https://doi.org/10.2514/1.G004850> (2020)
87. Maybank, S.J.: Geometric Invariance in Computer Vision, chap. The projection of two non-coplanar conics, pp. 105–119. MIT Press, Cambridge (1992)

88. McCabe, J., DeMars, K.: Anonymous feature-based terrain relative navigation. *J. Guid. Control Dyn.* **43**(3), 410–421 (2020). <https://doi.org/10.2514/1.G004423>
89. Melosh, H.J.: *Impact Cratering: A Geologic Process*, pp. 1–253. Oxford University Press, New York (1989)
90. Melosh, H.J.: *Planetary Surface Processes*, chap. The shapes of planets and moons, pp. 25–48. Cambridge, Cambridge University Press (2011). <https://doi.org/10.1017/CBO9780511977848.003>
91. Michikami, T., Hagermann, A., Morota, T., Haruyama, J., Hasegawa, S.: Oblique impact cratering experiments in brittle targets: Implications for elliptical craters on the moon. *Planet. Space Sci.* **135**, 27–36 (2017). <https://doi.org/10.1016/j.pss.2016.11.004>
92. Morel, J., Yu, G.: ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Imaging Sci.* **2**(2), 438–469 (2009). <https://doi.org/10.1137/080732730>
93. Mortari, D., Neta, B.: k-vector range searching techniques. In: *AAS/AIAA Spaceflight Mechanics Meeting*, AAS 00-128 (2000)
94. Mortari, D., Rogers, J.: A k-vector approach to sampling, interpolation, and approximation. *J. Astron. Sci.* **60**, 686–706 (2013). <https://doi.org/10.1007/s40295-015-0065-x>
95. Mortari, D., Samaan, M., Bruccoleri, C., Junkins, J.: The pyramid star identification technique. *Navigation* **51**(3), 171–183 (2004). <https://doi.org/10.1002/j.2161-4296.2004.tb00349.x>
96. Mundy, J.L., Kapur, D., Maybank, S.J., Gros, P., Quan, L.: *Geometric Invariance in Computer Vision*, chap. Geometric Interpretation of Joint Conic Invariants, pp. 77–86. MIT Press, Cambridge (1992)
97. Mundy, J.L., Zisserman, A.: *Geometric Invariance in Computer Vision*, chap. Introduction—towards a new framework for vision, pp. 1–39. MIT Press, Cambridge (1992)
98. Nagata, M.: On the 14-th problem of Hilbert. *Amer. J. Math.* **81**, 766–772 (1959). <https://doi.org/10.2307/2372927>
99. Nagata, M.: Invariants of a group in an affine ring. *J. Math. Kyoto Univ.* **3**, 369–377 (1963/64). <https://doi.org/10.1215/kjm/1250524787>
100. Naval Research Laboratory: CLEM1-LUN/EAR/SKY-ASTAR/BSTAR/UVVIS/HRES/LWIR/NIR-2-EDR-V1.0 NASA Planetary Data System (1995)
101. Neukum, G.: *Meteoritenbombardement und datierung planetarer oberflächen*. Habilitation Dissertation for Faculty Membership, Univ. of Munich (1983)
102. Neukum, G.: Cratering records in the inner solar system in relation to the lunar reference system. *Space Sci. Rev.* **96**, 55–86 (2001). <https://doi.org/10.1023/A:1011989004263>
103. Neusel, M.D.: *Invariant theory Student Mathematical Library*, vol. 36. American Mathematical Society, Providence (2007)
104. Nozette, S. et al.: The clementine mission to the moon: Scientific overview. *Science* **266**(5192), 1835–1839 (1994). <https://doi.org/10.1126/science.266.5192.1835>
105. Olds, R., May, A., Mario, C., Hamilton, R., Debrunner, C., Anderson, K.: The application of optical based feature tracking to OSIRIS-REX asteroid sample collection. In: *AAS Guidance, Navigation, & Control Conference*, AAS 15-124 (2015)
106. Olver, P.J.: *Classical invariant theory London Mathematical Society Student Texts*, vol. 44. Cambridge University Press, Cambridge (1999). <https://doi.org/10.1017/CBO9780511623660>
107. Olver, P.J., Shakiban, C. *Applied Linear Algebra*, 2nd edn. Springer, Cham (2018)
108. Owen, W.: Methods of optical navigation. In: *AAS/AIAA Space Flight Mechanics Meeting*, AAS 11-215 (2011)
109. Owen, W., Duxbury, T., Acton, C., Synnott, S., Riedel, J., Bhaskaran, S.: A brief history of optical navigation at JPL. In: *AAS Guidance and Control Conference*, AAS 08-053, Breckenridge, CO (2008)
110. Park, W., Jung, Y., Bang, H., Ahn, J.: Robust crater triangle matching algorithm for planetary landing navigation. *J. Guid. Control Dyn.* **42**(2), 402–410 (2019). <https://doi.org/10.2514/1.G003400>
111. Parkes, S., Dunstan, M., Martin, I., McCrum, M., Dubois-Matra, O.: Testing advanced navigation systems for planetary landers and rovers. In: *International Astronautical Congress*, IAC-09-A3.1.6 (2009)
112. Povilaitis, R.Z., Robinson, M.S., van der Bogert, C.H., Hiesinger, H., Meyer, H.M., Ostrach, L.R.: Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planet. Space Sci.* **162**, 41–51 (2018). <https://doi.org/10.1016/j.pss.2017.05.006>

113. Prasad, D.K., Leung, M.K.H.: Clustering of ellipses based on their distinctiveness: An aid to ellipse detection algorithms. In: 3rd International Conference on Computer Science and Information Technology (2010). <https://doi.org/10.1109/ICCSIT.2010.5564932>
114. Quan, L.: Algebraic and geometric invariant of a pair of non-coplanar conics in space. *J. Math. Imaging Vis.* **5**, 263–267 (1995). <https://doi.org/10.1007/BF01248375>
115. Quan, L., Gros, P., Mohr, R.: Invariants of a pair of conics revisited. *Image Vis. Comput.* **10**(5), 319–323 (1992). [https://doi.org/10.1016/0262-8856\(92\)90049-9](https://doi.org/10.1016/0262-8856(92)90049-9)
116. Quan, L., Veillon, F.: Joint invariants of a triplet of coplanar conics: Stability and discriminating power for object recognition. *Comput. Vis. Image Underst.* **70**(1), 111–119 (1998). <https://doi.org/10.1006/cviu.1998.0617>
117. Richter-Gebert, J.: *Perspective on Projective Geometry: A Guided Tour Through Real and Complex Geometry*. Springer, Berlin (2011)
118. Rijlaarsdam, D., Yous, H., Byrne, J., Oddenino, D., Furano, G., Moloney, D.: A survey of lost-in-space star identification algorithms since 2009. *Sensors* **20**, 2579 (2020). <https://doi.org/10.3390/s20092579>
119. Robbins, S.: A new global database of lunar impact craters > 1–2 km: 1. crater locations and sizes, comparisons with published databases, and global analysis. *J. Geophys. Res. Planets* **124**, 871–892 (2018). <https://doi.org/10.1029/2018JE005592>
120. Robbins, S.J., Riggs, J.D., Weaver, B.P., Bierhaus, E.B., Chapman, C.R., Kirchoff, M.R., Singer, K.N., Gaddis, L.R.: Revised recommended methods for analyzing crater size-frequency distributions. *Meteor. Planet. Sci.* **53**(4), 891–931 (2018). <https://doi.org/10.1111/maps.12990>
121. Robinson, M.S. et al.: The apollo digital image archive. In: *Proceedings of the Lunar and Planetary Science Conference (LPSC) XXXIX* (2008)
122. Rosenlicht, M.: Some basic theorems on algebraic groups. *Amer. J. Math.* **78**, 401–443 (1956). <https://doi.org/10.2307/2372523>
123. Rothwell, C.A., Zisserman, A., Forsyth, D.A., Mundy, J.L.: Planar object recognition using projective shape representation. *Int. J. Comput. Vis.* **16**, 57–99 (1995). <https://doi.org/10.1007/BF01428193>
124. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: *IEEE International Conference on Computer Vision (ICCV)* (2011). <https://doi.org/10.1109/ICCV.2011.6126544>
125. Salamunićcar, G., Lončarić, S.: Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on mola data. *Adv. Space Res.* **42**(1), 6–19 (2008). <https://doi.org/10.1016/j.asr.2007.04.028>
126. Salamunićcar, G., Lončarić, S., Mazarico, E.: Lu60645gt and ma132843gt catalogues of lunar and martian impact craters developed using a crater shape-based interpolation crater detection algorithm for topography data. *Planet. Space Sci.* **60**, 236–247 (2012). <https://doi.org/10.1016/j.pss.2011.09.003>
127. Samaan, M.A., Mortari, D., Junkins, J.L.: Nondimensional star identification for uncalibrated star cameras. *J. Astronaut. Sci.* **54**(1), 95–111 (2006). <https://doi.org/10.1007/BF03256478>
128. Semple, J.G., Kneebone, G.T.: *Algebraic Projective Geometry*, pp. 191–196. Oxford University Press, Oxford (1952)
129. Shiu, Y.C.: Ahmad, S.: 3d location of circular and spherical features by monocular model-based vision. In: *IEEE International Conference on Systems, Man and Cybernetics* (1989). <https://doi.org/10.1109/ICSMC.1989.71362>
130. Shoemaker, E.M.: *Physics and Astronomy of the Moon*, chap. Interpretation of Lunar Craters, pp. 283–360. Academic Press, New York (1962)
131. Silburt, A., Alo-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., Menou, K.: Lunar crater identification via deep learning. *Icarus* **317**, 27–38 (2019). <https://doi.org/10.1016/j.icarus.2018.06.022>
132. Solarna, D., Gotelli, A., Le Moigne, J., Moser, G., Serpico, S.: Crater detection and registration of planetary images through marked point processes, multiscale decomposition, and region-based analysis. *IEEE Geosci. Remote Sens. Lett.* <https://doi.org/10.1109/TGRS.2020.2970908> (2020)
133. Speyerer, E.: Quantifying crater production and regolith overturn on the moon with temporal imaging. *Nature* **538**, 215–218 (2016). <https://doi.org/10.1038/nature19829>
134. Spratling, B., Moratiri, D.: A survey on star identification algorithms. *Algorithms* **2**, 93–107 (2009). <https://doi.org/10.3390/a2010093>

135. Sugimoto, A.: A linear algorithm for computing the homography from conics in correspondence. *J. Math. Imaging Vis.* **13**, 115–130 (2000). <https://doi.org/10.1023/A:1026571913893>
136. Świrski, L., Bulling, A., Dodgson, N.: Robust real-time pupil tracking in highly off-axis images. In: *Symposium on Eye Tracking Research and Applications* (2012). <https://doi.org/10.1145/2168556.2168585>
137. Toda, F., Schlee, F.: Autonomous orbital navigation by optical tracking of unknown landmarks. *J. Spacecr. Rocket.* **4**(12), 1644–1648 (1966). <https://doi.org/10.2514/3.29147>
138. Troglia, G., Le Moigne, J., Atli Benediktsson, J., Moser, G., Serpico, S.: Automatic extraction of ellipsoidal features for planetary image registration. *IEEE Geosci. Remote Sens. Lett.* **9**(1), 95–99 (2012). <https://doi.org/10.1109/LGRS.2011.2161263>
139. Wang, H., Jiang, J., Zhang, G.: Crateridnet: An end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images. *Remote Sens.* **10**, 1067 (2018). <https://doi.org/10.3390/rs10071067>
140. Wang, J., Cheng, W., Zhou, C.: A chang'e-1 global catalog of lunar impact craters. *Planet. Space Sci.* **112**, 42–45 (2015). <https://doi.org/10.1016/j.pss.2015.04.012>
141. Wang, W., Zhang, X., Qi, Y., Cheng, W.: An algorithm of pose estimation based on conic correspondences. In: *IEEE International Conference on Robotics and Biomimetics* (2017). <https://doi.org/10.1109/ROBIO.2017.8324436>
142. Woicke, S., Moreno Gonzalez, A.S., El-Hajj, I., Mes, J.W.F., Henkel, M., Autar, R.S.D., Klavers, R.A.: Comparison of crater-detection algorithms for terrain-relative navigation. In: *AIAA Guidance, Navigation, and Control Conference* (2020). <https://doi.org/10.2514/6.2018-1601>
143. Wu, S., Moore, H.: Experimental photogrammetry of lunar images. In: *Geological Survey Professional Paper 1046-D*. U.S. Government Printing Office (1980). <https://doi.org/10.3133/pp1046D>
144. Yang, S., Baum, M., Granström, K.: Metrics for performance evaluation of elliptic extended object tracking methods. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (2016). <https://doi.org/10.1109/MFI.2016.7849541>
145. Yu, M., Cui, H., Tian, Y.: A new approach based on crater detection and matching for visual navigation in planetary landing. *Adv. Space Res.* **53**, 1810–1821 (2014). <https://doi.org/10.1016/j.asr.2013.04.011>
146. Zanetti, R.: Rotations, transformations, left quaternions, right quaternions? *J. Astronaut. Sci.* **66**, 361–381 (2019). <https://doi.org/10.1007/s40295-018-00151-2>
147. Zisserman, A., Forsyth, D., Mundy, J., Rothwell, C., Liu, J., Pillow, N.: 3d object recognition using invariance. *Artif. Intell.* **78**, 239–288 (1995). [https://doi.org/10.1016/0004-3702\(95\)00023-2](https://doi.org/10.1016/0004-3702(95)00023-2)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.