

# Optimization Methods for Asteroid Lightcurve Inversion

## I. Shape Determination

M. Kaasalainen and J. Torppa

*Observatory, University of Helsinki, P.O. Box 14, FIN-00014 Helsinki, Finland*

E-mail: [kaselain@gstar.astro.helsinki.fi](mailto:kaselain@gstar.astro.helsinki.fi)

Received December 22, 1999; revised October 18, 2000

**We have developed new methods for determining the shapes (and albedo distributions), rotation periods, and pole directions (as well as other parameters, e.g., those of the scattering law) of asteroids from their lightcurves. This paper concentrates on shape determination. The recovered shapes are general and not based on modifications of any prior shape model. We produce test lightcurves of various nonconvex bodies with a fast ray-tracing algorithm. We show that the use of positive definite quantities effectively removes the apparent ill-posedness of the problem. Our methods can obtain convex hulls even for strongly nonconvex objects; major concavities can also be resolved.** © 2001 Academic Press

**Key Words:** asteroids; photometry; surfaces, asteroids.

### 1. INTRODUCTION

Photometric lightcurves are and will remain a major source of information about asteroids, whose sizes and distances allow disk-resolved images to be obtained only for a limited number of targets. Hence lightcurve inversion will always be an important tool in the study of the small objects of our Solar System. Despite the large number of observed lightcurves and their analyses with various methods, the main question has remained unanswered: exactly how much information can we obtain from lightcurves?

In the order of increasing complexity, we assign the properties and quantities one can hope to recover to the following groups: (1) a convex representation of a nonconvex original body (with other parameters fixed), (2) the sidereal period and the pole direction (as well as other parameters, e.g., those of the scattering law) together with the shape solution, and (3) a nonconvex description of the original object. This paper concentrates on items (1) and (3), while a separate paper will be dedicated to item (2).

In Kaasalainen *et al.* (1992a,b) (hereafter KLLB and KLL), the inversion of the lightcurves of convex objects was studied mostly from the theoretical point of view. It turned out that, contrary to the general preconception, the object's shape can be derived from sufficiently extensive good-quality data—in an ideal case, it is possible even to separate the shape effects from those caused by albedo variegation. Also, two sets of real obser-

vational data were analyzed to provide shape solutions for two asteroids, and some of the restrictions imposed upon inversion in practice were discussed.

Several other authors have addressed the problem of lightcurve inversion; various schemes are presented in, e.g., Magnusson *et al.* (1989, 1996), Cellino *et al.* (1989), Barucci *et al.* (1992), and Michalowski (1996), only to mention a few. A typical approach has been to minimize the number of free parameters by placing, in addition to convexity, severe constraints on the shape: many models describe, e.g., simple modifications to the traditional triaxial ellipsoid. The rationale behind this is that the number and quality of observations are seldom believed to facilitate a more detailed method. We aim to show in this paper that it is quite possible to obtain inversion shapes that are significantly better than the triaxial ellipsoid, i.e., the general dimensions of the object.

Some probe and radar images of asteroids have been obtained during the past decade (see, e.g., Thomas *et al.* (1994, 1996, 1999) and Hudson and Ostro (1994, 1995, 1999)). The most conspicuous feature is the existence of substantial nonconvexities: there are globally significant valleys and indentations as well as large local craters on the surfaces. The object may even be a double asteroid of the rubble-pile type (as the radar images of Castalia and Toutatis suggest). It also seems that the most important cause of brightness variations is indeed the shape while the albedo effects are usually relatively small (probably due to the thick dust layers). The larger the solar phase angle, the more pronounced the shape effects are.

The new observations indicate that it is very important to study (a) how useful and descriptive convex inversion is in general, and (b) whether nonconvex inversion is feasible. Even strongly nonconvex bodies can often be reasonably well represented by their convex hulls; however, it has so far been very unclear whether convex inversion can really produce something akin to the convex hull or whether the result might be something else (better or worse depictive). Neither does one know how large the nonconvexities typically need to be to cause problems in inversion, or what could be regarded as a reliable indicator of nonconvexity (a poor fit or an inconsistent shape solution could be due

to other causes). In any case, with nonconvex objects we have to give up the uniqueness theorems proved for convex bodies.

The aim of this paper is to address the above questions by using simulated lightcurves of objects whose shapes range from almost convex to strongly nonconvex. We use practical inversion methods that are quite different from that presented in KLLB—although clearly best (and, in fact, almost the only possible one) for analytical studies, the latter is not very well suited for nonconvex shapes and/or observations of limited quantity and quality. In this way we seek to establish the range of applicability of convex inversion. We also give some answers to the still-open question of the direct problem: how do the lightcurves of a nonconvex body differ from those of its convex hull?

Although we refer to KLLB and KLL at some points, reading those papers is in no way necessary for understanding the present approach—they just provide more information for a reader interested in the theoretical details of the problem. This presentation is completely self-contained.

Not many nonconvex inversion schemes have been presented so far. One approach is the statistical one by Muinonen and Lagerros (1998) that, rather than obtaining a detailed shape solution, aims to give an idea of the extremity and general class of the nonconvex shape by employing Gaussian random spheres (Muinonen 1998). Another method by Mottola and Lahulla (2000) seeks to determine the basic dimensions of a binary target when both components are described by triaxial ellipsoids. We show that it is possible to build a general nonconvex inversion scheme using suitable optimization algorithms and parametrization.

In Section 2, we briefly discuss the numerical procedures involved in computing synthetic lightcurves. Section 3 describes the two methods we use in convex inversion, while nonconvex inversion is discussed in Section 4. Section 5 sums up and briefly introduces the topics on which we hope to report in the near future.

## 2. LIGHTCURVES OF NONCONVEX OBJECTS AND THEIR CONVEX HULLS

The lightcurves of convex bodies can be calculated (semi) analytically, whereas those produced by nonconvex objects must be computed numerically using a ray-tracing code. Since second-order scattering is negligible for low albedos, our code only needs to check which parts of the surface are visible to both the Earth and the Sun. For this end, the surface must be given as a polyhedron with triangles as facets. If the surface is given as a continuous function, it is triangulated with some standard discretization method such as the octant procedure (Appendix A) used by, e.g., Muinonen and Lagerros (1998). Triangulation can be applied separately to each object of a group; thus, e.g., binary objects can be handled with the same code.

Once it is known that a surface patch  $ds$  is both visible and illuminated, its contribution  $dL$  to the total brightness is given

by (omitting trivial scale factors such as the squares of distances)

$$dL = S(\mu, \mu_0)\varpi ds, \quad (1)$$

where  $S$  and  $\varpi$  are the scattering law (in a simple form here; more arguments can naturally be included) and albedo;  $\mu = \mathbf{E} \cdot \mathbf{n}$  and  $\mu_0 = \mathbf{E}_0 \cdot \mathbf{n}$ , where  $\mathbf{E}$  and  $\mathbf{E}_0$  are, respectively, unit vectors toward the observer (Earth) and the Sun, and  $\mathbf{n}$  is the surface unit normal. Lambert law, for example, is  $S_L = \mu\mu_0$ , while Lommel–Seeliger law is  $S_{LS} = S_L/(\mu + \mu_0)$ .

The ray-tracing procedure is quite simple. First one checks which vertices are above each facet’s local horizon and which facets connected to these vertices are facing this facet. These facets are the possible local blockers of light, and finding (and labeling) them first as well as precomputing their positions with respect to the local horizon makes the algorithm very fast. (The facets for which no vertices appear above the local horizon belong to the convex hull.) After the precomputation has been done, the actual lightcurve computation consists of only a few elementary geometric operations (such as projections along the viewing/illumination directions) per facet. The possible visibility and illumination of each facet is checked in the same way as in the convex case; i.e., both  $\mu$  and  $\mu_0$  must be positive. Those of the possibly visible and illuminated facets that do not belong to the convex hull must be checked further: if their centroids are blocked by any local blocker-facet, their contribution to the total brightness is omitted (only those possible blockers that do not belong to the group of possibly illuminated and visible facets need be included in the checking).

If any facet represents a large portion of the total surface area, checking only the shadowing of the centroids does not necessarily lead to an accurate result. In this case we place a number of test points on each facet (in a hexagonal mesh with small random perturbations) and check the shadowing for each of these points separately. This procedure defines how large a part of the facet is blocked; only a few tens of points are needed to give an accurate result. As a rule, the simple centroid check is quite accurate if there are hundreds of facets.

For the scattering law we chose a combination of Lommel–Seeliger and Lambert laws; this is quite sufficient for examining the shape effects. For each nonconvex shape we also computed the corresponding convex hull (as a polyhedron) from the vertices of the triangles on the surface. Such a procedure is easy to write by systematically comparing each point with the rest to find planes such that all the other points lie on one side of them; this gift-wrapping principle results in an  $N^2$ -algorithm. For completeness, it is described in Appendix B. Other, more efficient ( $N \log N$ ) methods are available in literature and on the Internet; some of them, however, are very complex or not completely reliable— $N^2$  is foolproof and not very much slower in absolute time when  $N$  is less than, say, 1000. The convex hulls typically contain large planar parts forming bridges over the valleys of the original shapes; such facets have a key role in convex inversion.

The direct problem of computing lightcurves produced by arbitrary shapes, observing geometries, and scattering laws is an interesting one in its own right. We plan to investigate a large sample of different shapes (especially Gaussian ones) in a future paper to find out, both qualitatively and quantitatively (statistically), what kind of phase- and lightcurve phenomena are characteristic of certain shapes or shape classes.

### 3. CONVEX INVERSION

Theoretical aspects of the lightcurve inversion of convex objects have been studied quite exhaustively (KLLB). If, however, an object is nonconvex, how well does inversion perform under the convexity assumption fare, and what is the best inversion method? Since this problem cannot be investigated analytically, we must study the simulated lightcurves of various irregular bodies.

The convex inverse problem can be cast in the form

$$\mathbf{L} = \mathbf{A}\mathbf{g}, \quad (2)$$

where  $\mathbf{L}$  is the vector of the observed brightnesses, related through the matrix  $\mathbf{A}$  to the vector  $\mathbf{g}$  that contains the parameters to be solved (the variances of the observations can be taken into account in  $\mathbf{A}$  and  $\mathbf{L}$  in the usual manner and are thus not explicitly written out here). This vector can describe the object's curvature function (also called Gaussian surface density), which determines the shape uniquely; it may also represent the albedo distribution or the product of the two. We usually assume the first case; others are extensions either explicitly stated or implicitly clear. Any more complicated expressions including parameters of the scattering law in  $\mathbf{g}$  are not relevant to our considerations here. The parameters in  $\mathbf{g}$  are either the areas of the facets of a convex polyhedron or the coefficients of a spherical harmonics series. In both cases the problem is ill-posed: without prior constraints, it is simply impossible to find a meaningful solution for  $\mathbf{g}$  from the deceptively simple-looking linear equation (2) if there are slightest errors in the observations.

The two basic choices for  $\mathbf{g}$ —the facets of a polyhedron or smooth functions—are rather complementary. While the latter allows a very low number of parameters, the former is the best choice for a large parameter set: a functional series would have to be extremely long to describe a large planar part or a sharp feature on the surface as well as facets.

In KLLB and KLL a regularization technique (statistical inversion) was used for obtaining an acceptable solution, i.e., a feasible curvature function. The main problem is that if  $\mathbf{g}$  is a set of coefficients of a spherical harmonics series, there is no practical way of guaranteeing that the curvature function is everywhere positive (as it must be if it is to describe a real shape). Thus the regularization was performed by forcing the solution to fluctuate as little as possible in the hope that this would produce a meaningful result.

#### 3.1. Polyhedra and Conjugate Gradients

If  $\mathbf{g}$  describes the areas of the facets of a polyhedron, the positivity constraint is easy to check: we must have  $g_j \geq 0$  for all  $j$ . Also,

$$\sum_j \mathbf{n}_j g_j = \mathbf{0}, \quad (3)$$

where the unit vector  $\mathbf{n}_j$  is the chosen surface outward normal of the facet  $j$  (we typically use the facet normals of a sphere or a triaxial ellipsoid triangulated in the standard manner). These constraints are necessary and sufficient for  $\mathbf{g}$  to describe a convex polyhedron (see KLLB). If the constraint (3) is omitted,  $\mathbf{g}$  is taken to include albedo variegation.

The matrix  $\mathbf{A}$  is obtained directly from (1),

$$A_{ij} = S_j(\mu^{(ij)}, \mu_0^{(ij)})\varpi_j, \quad (4)$$

where  $S_j$  and  $\varpi_j$  are the scattering law and albedo at the facet  $j$ ;  $\mu^{(ij)} = \mathbf{E}_i \cdot \mathbf{n}_j$  and  $\mu_0^{(ij)} = \mathbf{E}_{0i} \cdot \mathbf{n}_j$  for the observation  $i$  (in the asteroid's frame of reference). If either  $\mu^{(ij)}$  or  $\mu_0^{(ij)}$  is less than or equal to 0,  $A_{ij}$  vanishes, of course.

The standard solution of (2) by minimizing the square norm

$$\chi^2 = \|\mathbf{L} - \mathbf{A}\mathbf{g}\|^2 \quad (5)$$

using least-squares normal equations or singular-value decomposition would usually produce negative  $g_j$  values. Any regularization scheme, such as maximum entropy method or an application of statistical inversion, would introduce an a priori element representing conditions that are not necessarily true or knowledge we do not really possess. All we really know is that the  $g_j$  must be positive: other than that, they can vary arbitrarily if we relax the convexity constraint to be able to obtain an indication of albedo variation.

The easiest way to guarantee positivity is to represent each  $g_j$  exponentially, the optimization parameter being now the exponent  $a_j$ :

$$g_j = \exp(a_j). \quad (6)$$

The values of  $a_j$  are not constrained, so this is much more practicable than using penalty or barrier functions to keep  $g_j$  (as actual parameters) positive. Also, since the surfaces of constant  $\chi^2$  are convex surfaces (hyperellipsoids) in  $\mathbf{g}$ -space, there is one and only one vector  $\mathbf{g}$  with  $g_j \geq 0$  for all  $j$  that minimizes  $\chi^2$ . Because the exponential function is monotonous, only one vector  $\mathbf{a}$  corresponds to that  $\mathbf{g}$ ; thus the smallest  $\chi^2$  solution is unique in the exponential formalism.

Using the exponential form renders the optimization problem nonlinear. However, any optimization procedure will “flow” toward the smallest  $\chi^2$  solution and ultimately find it because there is only one global and local minimum. Since the number

of fitted parameters must be large (of order 1000) to make sure that the result does not depend on the directions of the surface normals, we use the conjugate gradient method (see, e.g., Press *et al.* (1994)) for minimizing  $\chi^2$ . Once the areas of the facets are known, the vertices of the facets can be obtained by Minkowski minimization (KLLB, Lamberg 1993). This is another nonlinear optimization problem that can be solved with standard methods; we briefly outline the procedure in Appendix C.

In practice, the observed brightnesses are usually several times smaller at large solar phase angles than near opposition. Therefore it is advantageous to replace the standard  $\chi^2$  of (5) by a renormalized  $\chi_{\text{ren}}^2$

$$\chi_{\text{ren}}^2 = \sum_i \left\| \frac{\mathbf{L}^{(i)} - A^{(i)} \mathbf{g}}{\bar{L}^{(i)}} \right\|^2, \quad (7)$$

where the index  $i$  refers to each lightcurve sequence separately, and  $\bar{L}^{(i)}$  is the mean brightness of the  $i$ th lightcurve. By redefining  $\mathbf{L}$  and  $A$ , this form thus normalizes each lightcurve to oscillate around unity, giving each observing geometry equal weights. Such ‘‘democratization’’ is better than one produced by working in magnitude space since the latter would not provide a simple  $\chi^2$  function.

### 3.2. Smooth Functions and Levenberg–Marquardt Method

If we represent an object’s curvature function as an exponential spherical harmonics series, the positivity constraint is automatically fulfilled. Again, we lose linearity (and the corresponding analytic properties discussed in KLLB), but gain the possibility of stable inversion without regularization. Thus,

$$G(\vartheta, \psi) = \exp\left(\sum_{lm} a_{lm} Y_l^m(\vartheta, \psi)\right), \quad (8)$$

where  $(\vartheta, \psi)$  are the spherical coordinates of the surface normal. If albedo variegation is absorbed into  $G$ , the observed brightness is

$$L(\mathbf{E}, \mathbf{E}_0) = \int \int_{A_+} S G(\vartheta, \psi) d\sigma, \quad (9)$$

where the integration region  $A_+$  on the unit sphere of normal directions (i.e., the Gaussian image sphere) is the part on which  $\mu, \mu_0 \geq 0$ ,  $S$  is the scattering law, and  $d\sigma$  is the surface element of the sphere.

The numerical integration of (9) is easy to perform as a sum over a triangulated unit sphere, using the facets to approximate  $d\sigma$ ; with a dense enough triangulation, the error is negligible and no ‘‘fancier’’ methods with rotations and quadratures for double integration need be used. The triangulation returns (9) to (2) and (4), with

$$g_j = G(\vartheta_j, \psi_j) \Delta\sigma_j, \quad (10)$$

where  $\Delta\sigma_j$  is the area of the sphere facet corresponding to  $(\vartheta_j, \psi_j)$ .

Since we are minimizing a nonlinear least-squares function (5) and the number of the coefficients  $a_{lm}$  to be solved for is not large (typically from, say, 40 to 100), it is advantageous to use the Levenberg–Marquardt optimization scheme (Press *et al.* 1994). The initial guess for iteration can be, e.g., a suitable triaxial ellipsoid; the logarithm of its curvature function (given in KLL) can be fitted by the chosen number of coefficients using linear least squares.

Once the curvature function  $G$  has been obtained, it can be discretized to represent the areas of the facets of a convex polyhedron (or rather, a  $\mathbf{g}$  that may include albedo variegation). The easiest choice is to use the  $g_j$  already given by (10); should one not want to have an evenly distributed set of normal directions, any other system can be used as well. After this step, we proceed as in the polyhedron method.

One advantage of this scheme is that, while the behavior of  $\chi^2$  in the parameter space cannot be described as easily as in the polyhedron case, the optimization procedure usually converges very efficiently toward the correct solution even with a poor initial guess. Thus, one can use this approach to obtain a fast initial solution that can then be enhanced with the polyhedron method. Another advantage is that the robust convergence is usually retained when any fixed parameters (e.g., period, pole, or scattering law) are changed to free ones. This scheme is thus a powerful all-purpose tool for lightcurve inversion.

### 3.3. Albedo Variegation

We can quite safely assume that if the sum of the facet vectors (3) is very small, the lightcurve features are in all probability caused by the shape. A strongly nonzero residual vector indicates albedo variegation. In this case the stability of the solution must first be checked. For this end, minimization can be rerun using the convexity constraint. The exponential form renders also the constraint (3) nonlinear and thus does not allow explicit projection of the search directions into the convex subspace where (3) always holds (as a linear form would); however, convexity can be enforced in practice by adding the square of the length of the vector sum to  $\chi^2$  as a regularization function. This is equivalent to modifying the original  $\chi^2$  of (5) by adding three zero elements to  $\mathbf{L}$  and three new rows to  $A$ . By changing the weighting factor of these rows one can easily see whether the  $\chi^2$  of the unregularized solution is really much lower than that of a regularized one. Light regularization may sometimes be necessary to filter out artificial albedo indications especially in pole directions.

Once the true size of the residual nonconvexity vector (3) has been established, one encounters the fundamental property of the inverse problem: in all realistic cases albedo effects are in principle quantitatively inseparable from shape effects. Therefore we must use some prior constraints to obtain a plausible result. If we denote the area and the albedo of facet  $j$  by, respectively,  $s_j$  and  $\varpi_j$ , the new objective function for the simultaneous

optimization of shape and albedo is given by

$$\chi_{\text{sep}}^2 = \sum_j (g_j - s_j \varpi_j)^2 + \lambda_s \sum_{i=1}^3 \left[ \sum_j n_j^{(i)} s_j \right]^2 + \lambda_{\varpi} f(\varpi), \quad (11)$$

where  $g_j$  are the facet values obtained from lightcurves,  $f(\varpi)$  is some regularization function for albedo, and  $\lambda_s$  and  $\lambda_{\varpi}$  are regularization weights. It is better to use the form (11) rather than (5) augmented by the  $\lambda_s$  and  $\lambda_{\varpi}$  terms since the obtained  $g_j$  already contain all the shape/albedo information that can be extracted from the observational data.  $\lambda_s$  and  $\lambda_{\varpi}$  are to be kept very small as (11) must be strongly dominated by the first term.

The function (11) can be minimized just as before, with the exception that now we have twice as many parameters. If the function series form is used for facet areas and albedos, a good smoothing  $f(\varpi)$  is  $f(\varpi) = \int \int_{S^2} |\nabla \varpi|^2 d\sigma$ , where integration over the unit sphere  $S^2$  is done as with (9); this function retains the  $\chi^2$  form used in the Levenberg–Marquardt routine. Another possibility, suitable also for separate facet values, is to use  $f(\varpi) = \sum_j \sum_i (\varpi_{ij}/\varpi_j - 1)^2$ , where  $\varpi_{ij}$  are the albedo values of the facets adjacent to facet  $j$ . The adjacency relations are not known at this stage, but a very good approximation is to use those of the octant triangulation. The purpose of both these choices of  $f(\varpi)$  is to minimize the albedo differences between facets close to each other.

Since the albedo values should be restricted to some given interval  $[a, b]$ , it is practical to write  $\varpi$  in the form

$$\varpi = a + (b - a) \frac{\exp(c)}{\exp(c) + 1}, \quad (12)$$

where  $c$  is the optimization parameter/function.

In the above manner we get a solution that fits the lightcurves, is convex, and describes the albedo asymmetry rather than the actual distribution. All albedo symmetries are absorbed into the shape solution, so the albedo “map” is realistic only if there is a single prominent albedo spot on the surface. An important fact is that the shape resulting from this separation is close to the correct one so long as the albedo markings on the surface are not very bright and/or extensive. The reason for this is the strong stability of the Minkowski problem: the shape of a convex body may change very little even if the areas of the separate facets change a lot. It is precisely this property that makes convex inversion so robust; it is also the reason why it is much safer to attribute brightness changes to shape rather than albedo.

Minkowski stability also means that we need not fulfill the convexity constraint very accurately to obtain the convex shape. A small nonzero residual in (3) is easily fixed by adding a facet of corresponding size such that the new (3) vanishes: this is done to make sure that Minkowski minimization proceeds successfully. The small new facet is completely dark, but its existence does not affect the overall shape.

### 3.4. Relative Brightness in Inversion

The explicit assumption in any approach based on (5) or (7) is that all the factors contributing to the absolute brightness of the object as a scale factor for a single observing geometry are known (either explicitly or in a form that allows the determination of the values of the corresponding functional parameters). If this does not hold, one can still use relative brightnesses. In this case we minimize

$$\chi_{\text{rel}}^2 = \sum_i \left\| \frac{\mathbf{L}^{(i)}}{\bar{L}^{(i)}} - \frac{A^{(i)} \mathbf{g}}{\langle A^{(i)} \mathbf{g} \rangle} \right\|^2, \quad (13)$$

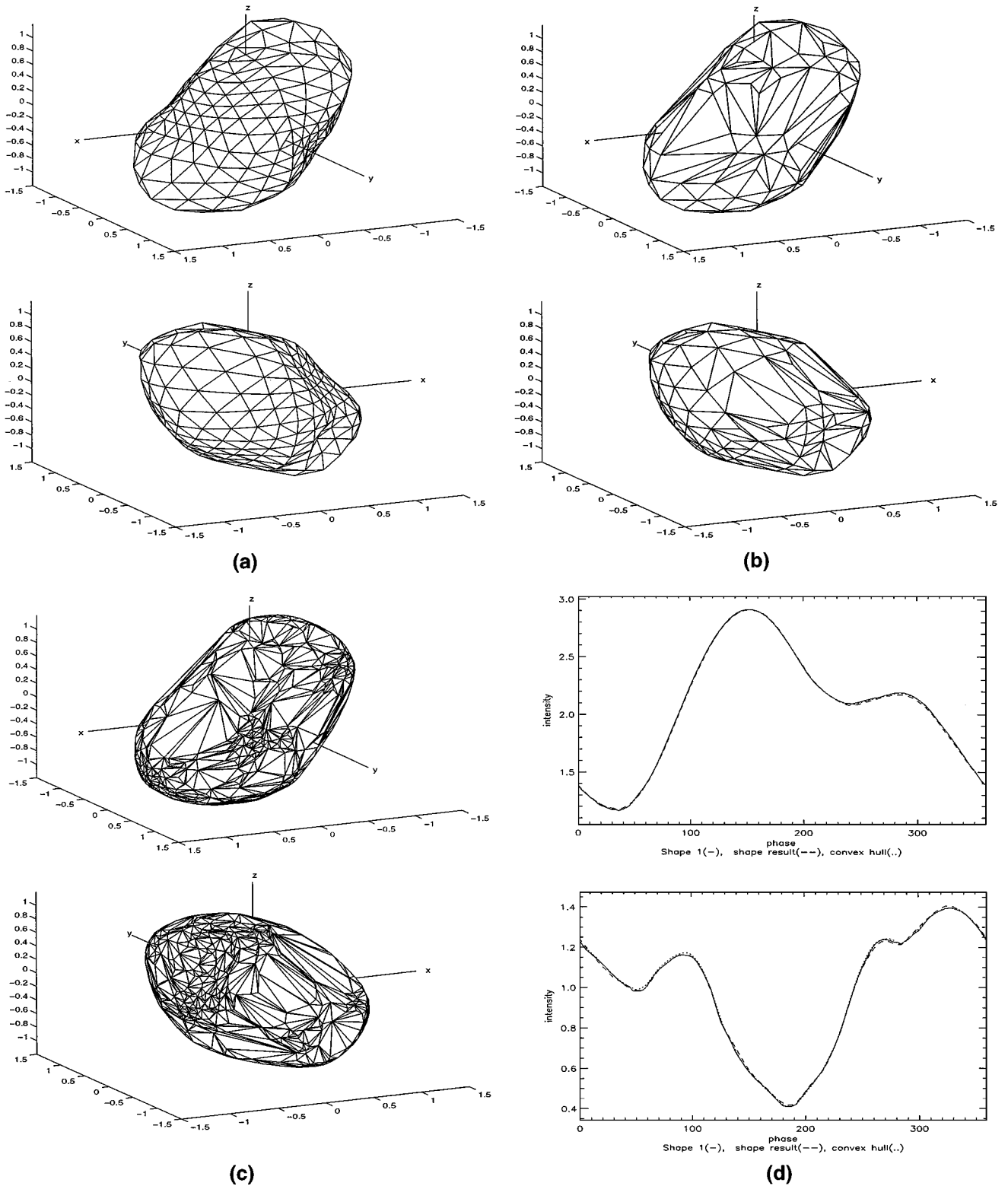
where  $\langle A^{(i)} \mathbf{g} \rangle$  is the mean brightness of the  $i$ th model lightcurve fitting the  $i$ th observed lightcurve. Thus both the observed and the model lightcurves are renormalized to mean brightnesses of unity. This is similar to leaving the scale factor for each lightcurve as a free parameter to be determined; the form (13) is more advantageous especially for the smooth function scheme as it discards all scale factors and thus keeps the number of free parameters as low as possible. The coefficient  $a_{00}$  in (8) is a scale factor as well, so it can be left out of the parameter set.

A possible problem with the form (13) (or fitting separate scale factors) is that, e.g., the smoothness of the solar phase function of the scattering law is not used as a constraint in inversion. Allowing arbitrary scale factors for each lightcurve improves the fit but may in principle remove the solution from the correct one. However, we have not found the use of (13) problematic in practice (if relative photometry is included in the data set it is, of course, the only option). Test simulations indicate that the solution based on relative brightnesses is for all practical purposes equivalent to that obtained from absolute photometry. This is quite understandable: it is primarily the shapes of the lightcurves that are strongly connected with the pole, the period, and the shape of the asteroid. The absolute brightnesses are principally connected with the scattering properties; thus it is actually useful to employ (13) to decouple one set of parameters from the rest as much as possible.

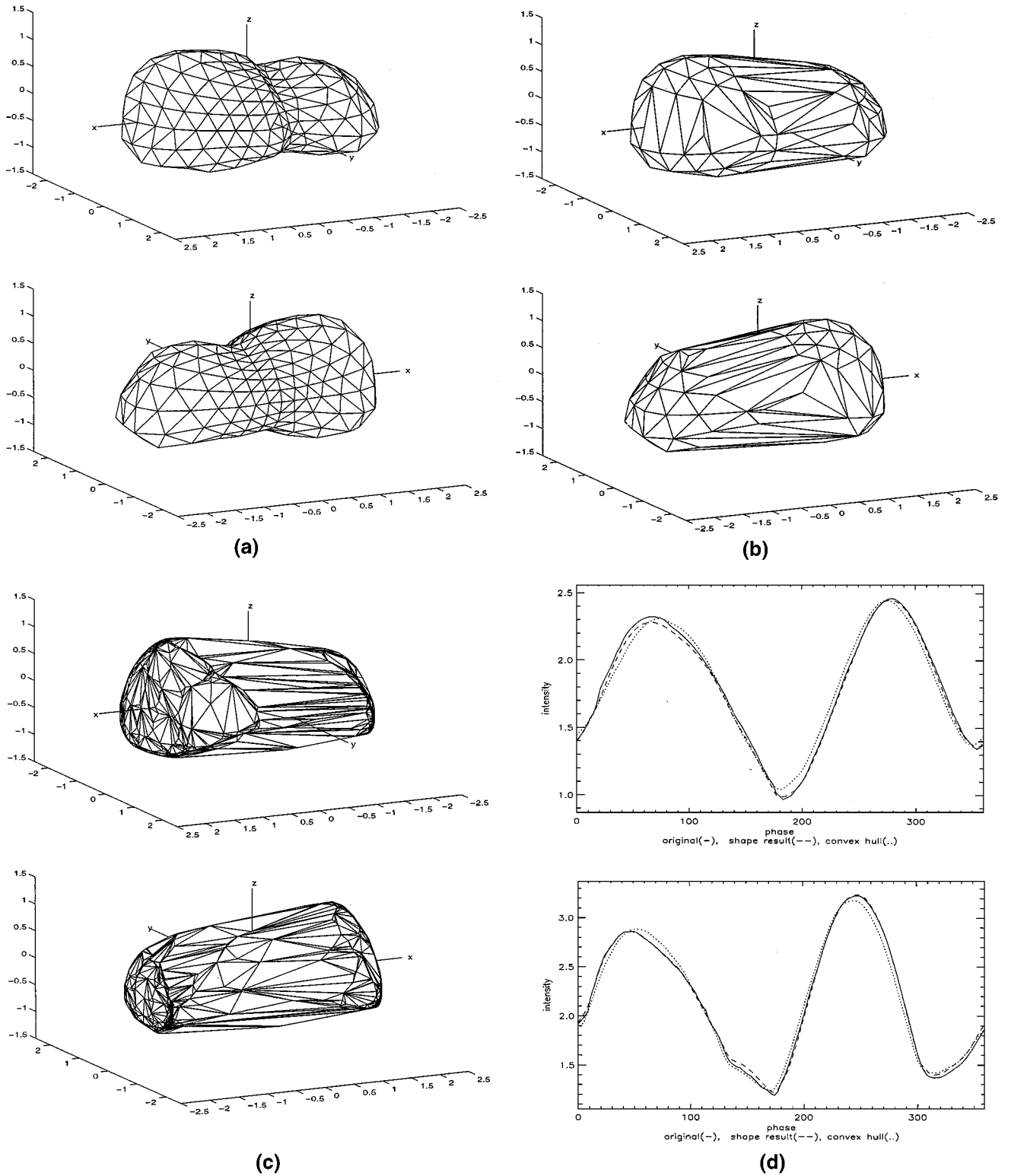
It is often necessary to use convexity regularization in connection with (13). This is caused by the fact that since albedo markings at poles affect the shape of a lightcurve very little, the inversion procedure may happily adopt too large facet values at pole directions to obtain a tiny improvement in  $\chi_{\text{rel}}^2$ . This instability in pole directions can be controlled with very moderate regularization; in fact, it is sufficient to include only the  $z$ -component term of (3) to be minimized with  $\chi_{\text{rel}}^2$ .

### 3.5. Numerical Simulations

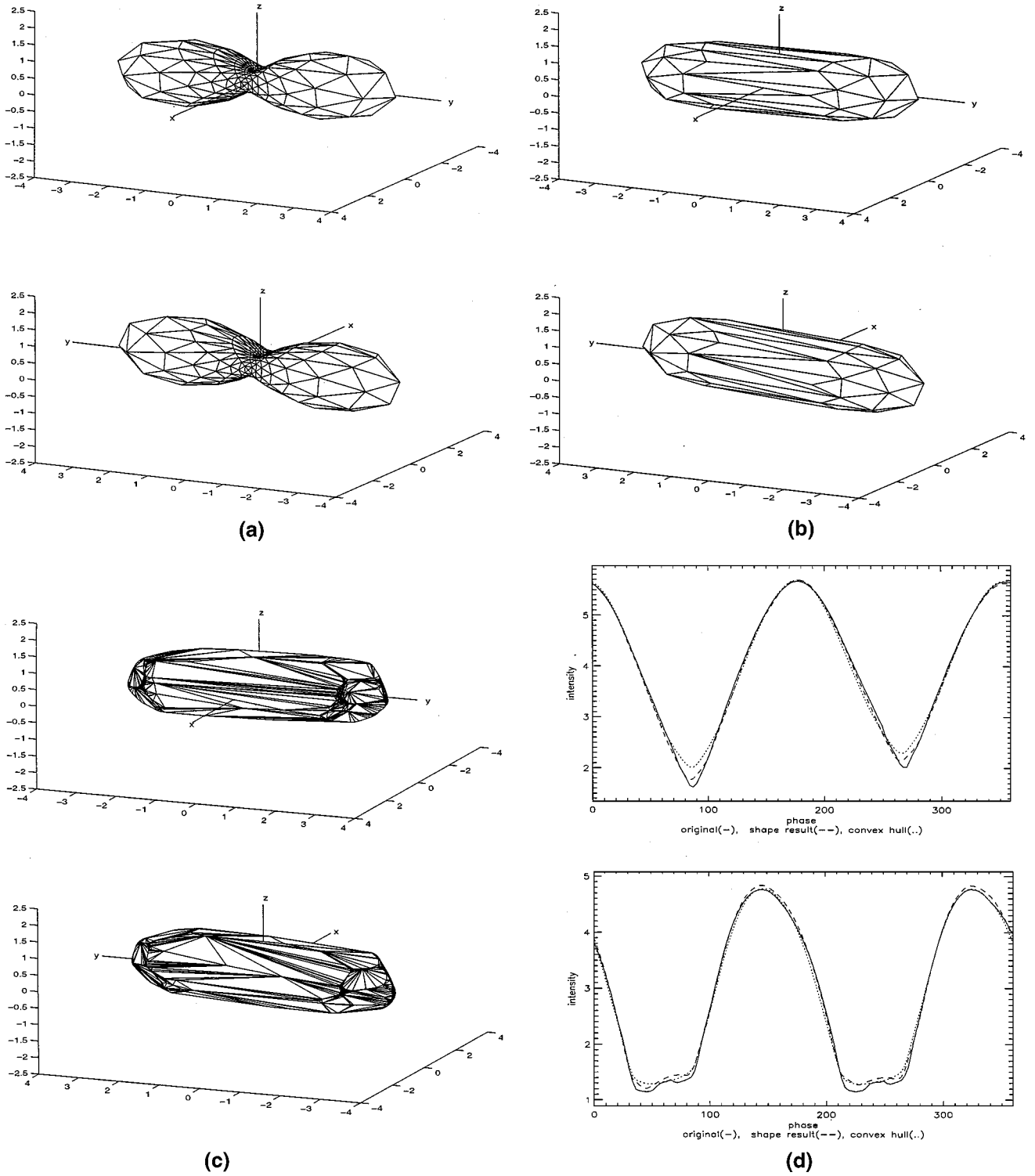
We present four examples of increasing nonconvexity. The lightcurves were generated by: **1**, an irregular shape (tilted to produce interesting lightcurves); **2**, a shape based on the radar images of Asteroid Castalia (Hudson and Ostro 1994); **3**, a peanut-shaped object to represent a strongly nonconvex yet regular shape); and **4**, a binary object. Figures 1–4 correspond to



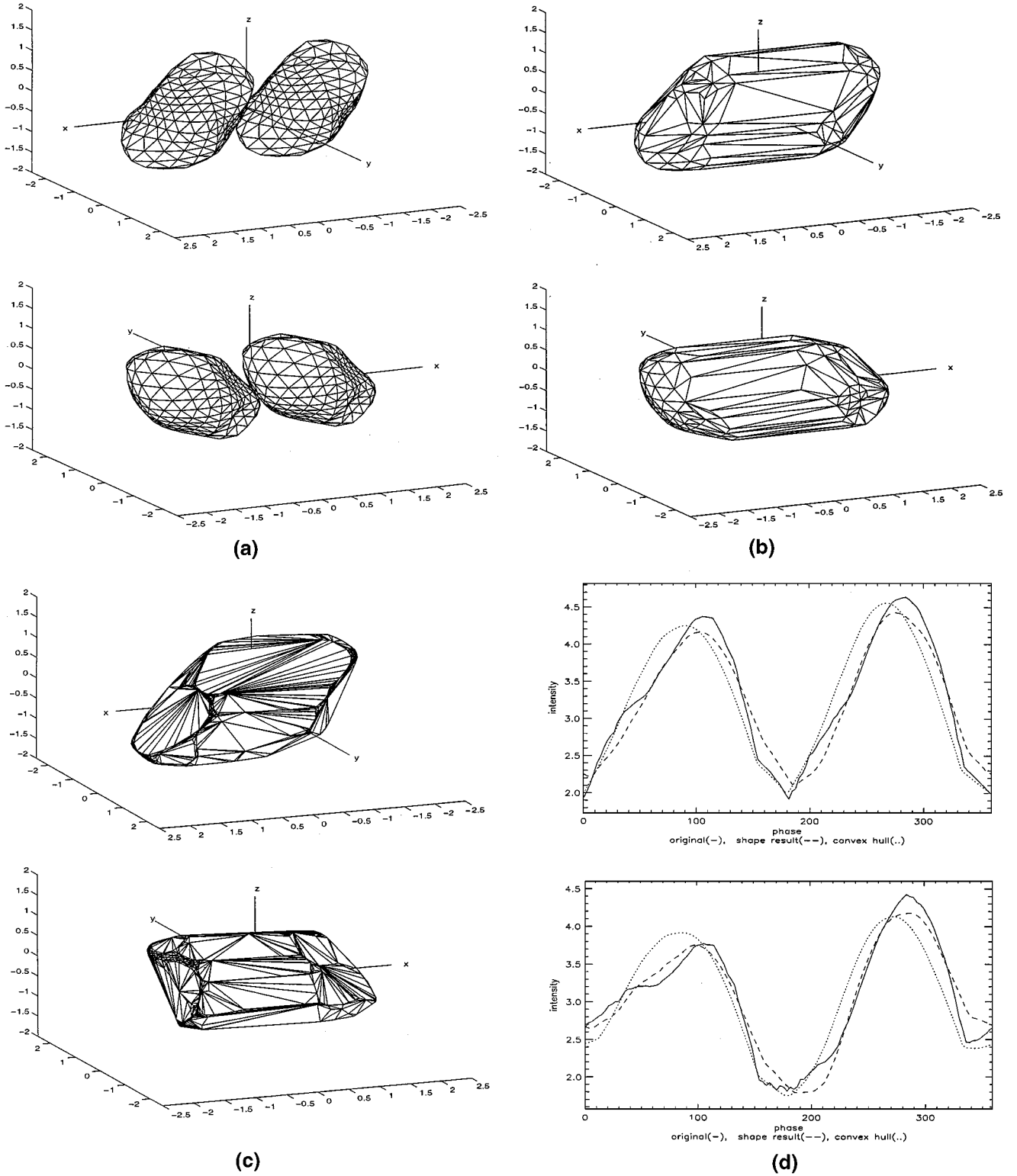
**FIG. 1.** Shape 1. (a) The original shape shown from two directions, (b) corresponding views of the convex hull, (c) the convex shape solution obtained by polyhedron inversion, and (d) lightcurves produced by the shapes (a) (solid line), (b) (dotted line), and (c) (dashed line) in two observing geometries ( $\alpha = 27^\circ$  and  $\alpha = 80^\circ$ ).



**FIG. 2.** Shape 2. (a) The original shape shown from two directions, (b) corresponding views of the convex hull, (c) the convex shape solution obtained by polyhedron inversion, and (d) lightcurves produced by the shapes (a) (solid line), (b) (dotted line), and (c) (dashed line) in two observing geometries ( $\alpha = 53^\circ$  for both; one geometry viewed and illuminated from the equator, the other from different sides of the equator).



**FIG. 3.** Shape 3. (a) The original shape shown from two directions, (b) corresponding views of the convex hull, (c) the convex shape solution obtained by polyhedron inversion, (d) lightcurves produced by the shapes (a) (solid line), (b) (dotted line), and (c) (dashed line) in two observing geometries ( $\alpha = 27^\circ$  and  $\alpha = 53^\circ$ ).



**FIG. 4.** Shape 4. (a) The original shape shown from two directions, (b) corresponding views of the convex hull, (c) the convex shape solution obtained by polyhedron inversion, (d) lightcurves produced by the shapes (a) (solid line), (b) (dotted line), and (c) (dashed line) in two observing geometries ( $\alpha = 27^\circ$  for both; one geometry viewed and illuminated from the equator, the other from different sides of the equator).

these objects; the original shapes are shown in part (a) of each figure, and their convex hulls in part (b). The shape solutions obtained by polyhedron inversion are shown in part (c).

Not very many input lightcurves are needed for inversion. Ten curves were used here for each case: they suffice so long as they cover a wide range of observing geometries and there are sufficiently many lightcurve points. The main requirement for the data sets is that they consist of lightcurves that exhibit “characteristic” features caused by the shape. In practice this means that observing geometries must reach large solar phase angles  $\alpha$ ; due to the lack of shadowing effects, shape information in observations made at small solar phases is often restricted to the general dimensions of the target. For a more detailed solution, there should be at least a few lightcurves with  $\alpha$  greater than, say,  $20^\circ$ . The density of lightcurve points in rotational phase does not seem to be crucial as such: if the observing geometries produce characteristic lightcurves, it does not necessarily matter if there are 20 curves with 50 points in each, or 10 curves with a 100 points in each.

The scattering law used in all computations was a combination of Lommel–Seeliger and Lambert laws with equal weights. Albedo was first held constant. Lightcurves produced at two observing geometries by the model shapes, their convex hulls, and the inversion shapes are shown, respectively, as solid, dotted, and dashed lines in part (d) of Figs. 1–4. We chose to show lightcurves at large  $\alpha$  to emphasize their role. To enable the comparison of pure shape effects, no simulated errors are shown in the input lightcurves. Since the projected area of the convex hull is larger than that of the original surface, the lightcurves of the former were rescaled by finding a scale coefficient such that the lightcurves of the two shapes are as similar as possible (this corresponds to a simple shrinking of the convex hull.) Such a coefficient is the one that minimizes  $\|\mathbf{L} - c\mathbf{L}_{\text{ch}}\|^2$ , where ch denotes convex hull. Thus

$$c = \frac{\sum_i L_i L_{\text{ch},i}}{\sum_i L_{\text{ch},i}^2}, \quad (14)$$

computed from all lightcurve data (not for each lightcurve separately).

As can be seen, the first three cases yielded shape results very close to the convex hulls of the original bodies, and even the shape solution of case **4** is similar to the convex hull. In all cases, the shape solution fitted the input lightcurves better than the convex hull: it is quite interesting to see how the convex solution is often able to reproduce even some fine details of the input lightcurves (while the convex hull smooths them out). Thus, even though the best fitting convex shape is not the exact convex hull, it is very close to the latter. It is actually quite surprising how similar the lightcurves of even a strongly nonconvex body and its convex hull are. In the case of shape **3**, for example, the regularity and symmetricity of the body seem to contribute to the lightcurve much more than the otherwise large concavity in the middle. These and other simulations indicate that the range

of applicability of convex inversion is not seriously limited by the nonconvexity of the target body.

The fact that the inversion result is very similar to the convex hull is quite important. The convex hull is a convex shape that actually describes a (reasonably ordinary) nonconvex body well since the locations of the concavities are seen in the convex hull as large planar sections. A smooth, rounded convex surface would not provide any information about the possible concavities. In this sense the polyhedron approach is better than the smooth function one: the latter produced very good inversion shapes in all cases as well, but the large planar areas were not so clearly defined.

The polyhedral model is defined by a set of surface normal directions. We found that when the number of parameters (i.e., the number of facets) is too low, of order a few hundred, the result clearly depends on the choice of the normal directions. When the directions were those of a triangulated ellipsoid, the best result was obtained with the axis ratios that best described the overall dimensions of the target. Such a choice of the surface normals uses a priori information that we, in general, do not have. It turned out that the number of parameters should be of order 1000 (corresponding to evenly distributed surface normals) to make the result independent of the exact choice of the normal directions. Densely spaced normal directions are required especially to enable the minimization procedure to find the possible large facets in the convex hull: such a facet cannot be found if there is no surface normal directions in the model that closely corresponds to it. If there are substantially fewer than 1000 data points, one can “create” more by adding points to the “safe” parts of lightcurves by interpolation. This is equivalent to constraining the shape of the lightcurve and is thus quite legal.

Realistic data noise does not alter the result appreciably. In fact, we have found that even considerable noise (from 5 to 10%), while of course affecting the resolution (the large facets were not retained as solid areas but were split into smaller ones and rounded a little), does not cause a need for regularization. Thus the positivity constraint indeed removes the ill-posedness: the unique solution is also stable; i.e., there is only one convex shape that corresponds to a specific set of lightcurves (if the set is a good one). The problem always exhibits strong artificial ill-posedness since infinitely many different unrealistic solutions produce similar sets of lightcurves. Filtering such solutions out removes stability problems.

No albedo indication was obtained for surfaces with constant albedo; i.e., optimization converged to virtually convex shapes. The ratio of the size of the residual nonconvexity (3) to the total surface area varied between 0.001 and 0.007 for the four shapes. This is surprisingly small for shapes **3** and **4** considering the scale of the concavities. It thus seems that even strong nonconvexities are usually not mistaken for albedo variegation (if there are various observing geometries). Adding realistic albedo spots onto the surfaces posed no difficulties for the inversion procedure: the obtained shapes remained practically the same, and the strengths and locations of the albedo asymmetries were correct.

No number of simulations can prove the efficacy of the albedo procedure conclusively, so we plan to analyze asteroids with probable albedo markings (such as 4 Vesta and 51 Nemausa) in the near future.

The sizes of the models for shapes **1**, **2**, and **3** agree well with the sizes of the convex hulls of the original bodies. The result for shape **4**, however, has one dimension slightly too small. This stems from the fact that, due to the strong nonconvexity, the projected area of the surface is significantly smaller than that of its convex hull when viewed from the direction perpendicular to this dimension. Thus, if the concavities in the original shape are large, the inversion procedure tends to contract the corresponding directions of the shape result. Even if there were clear evidence (based on the locations of the large facets) that some specific dimension should be expanded to obtain the convex hull, the amount of expansion would not be clear since the depths of the concavities are not known.

#### 4. NONCONVEX INVERSION

This problem is a very demanding one, the main complications being that all uniqueness theorems are lost and the parameter space is usually plagued by local minima. There are not very many methods available; after some experimenting, we have found it best to use a short functional series describing the locations of the vertices of a triangulated surface. One possibility is

$$r(\theta, \varphi) = \exp\left(\sum_{lm} c_{lm} Y_l^m(\theta, \varphi)\right), \quad (15)$$

where  $(\theta, \varphi)$  are spherical coordinates. Another one is, for example,

$$\rho(x, \phi) = \exp\left(\sum_{jk} c_{jk} x^j e^{ik\phi}\right), \quad (16)$$

where  $\rho$  is the cylindrical radius as a function of the cylindrical coordinates  $(x, \phi)$  the cylinder lies horizontally in the  $xy$  plane). Using a previous solution from convex inversion, the  $x$  axis must be chosen (by rotating the coordinate system) such that it coincides with the long axis of the body.  $\rho$  is set to 0 at the two endpoints  $x_-, x_+$ , whose  $x$  values can be obtained from the convex inversion result as well (this form thus contains the implicit assumption that such a long axis can be drawn through the body); one of the endpoints can be adjusted during the iteration. The function (16) is only valid in some given interval  $[x_1, x_2]$ , where  $x_- < x_1 < x_2 < x_+$ , and the intervals  $[x_-, x_1]$ ,  $[x_2, x_+]$  correspond to the first and last rows of the triangulation mesh.

We fix a set of directions along which we optimize the  $c_{lm}$ , giving the radii of the vertices of a nonconvex polyhedron approximating the shape; the facet connections between the vertices are also fixed. The directions and connections are easiest to form by standard triangulation; any other scheme can be used if one has a preconception of the object's shape. The

exponential form is again useful for ensuring positivity, and it is better to use the  $c_{lm}$  as parameters rather than the individual radii themselves. If the latter were used, the number of parameters would be much larger and we would need to regulate the correlations between adjacent radii to prevent porcupine-shaped solutions.

The trial lightcurves are now computed with the ray-tracing program described earlier; the resulting  $\chi^2$  can be minimized with Levenberg–Marquadt since local derivatives with respect to shape parameters do exist even though they are not continuous. If the length of the radius vector at a given vertex is denoted by  $r$  and the corresponding unit vector by  $\hat{\mathbf{r}}$ , we have  $\partial A/\partial r = \hat{\mathbf{r}} \cdot (\mathbf{d} \times \mathbf{n})/2$  and  $\partial \mu_{(0)}/\partial r = [\hat{\mathbf{r}} \cdot (\mathbf{d} \times \mathbf{E}_{(0)})/2 - \mu_{(0)} \partial A/\partial r]/A$ , where  $A$  is the area of a seen and illuminated triangle whose corner the vertex is, and  $\mathbf{d}$  is the vector corresponding to the side of the triangle opposite the vertex (pointing in the positive rotation direction). This approach converges very fast despite the discontinuity of the optimization gradient caused by the polyhedron approximation. The initial guess should be a good one (e.g., the series fitted to a convex inversion result). Genetic algorithms can also be used (in which case the gradient is not needed); they are considerably slower but can be employed for global scanning of parameter space.

It is often necessary to employ smoothness regularization to suppress unrealistic surface fluctuation and the formation of

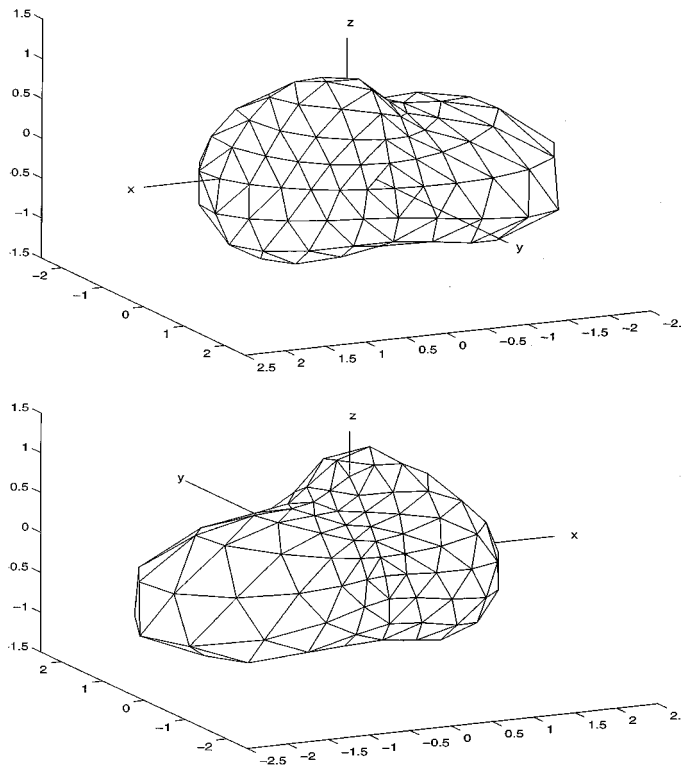


FIG. 5. Nonconvex shape solution for shape **2**, obtained using the function (15) and the vertex directions of a triangulated sphere.

artificial features. A useful method is to minimize the area “sunk below” the convex hull of the current result, i.e., to encourage convexity. The regularization term consists of the sum of the areas of the facets not in the convex hull, each multiplied by the average “height” of the vertices of possible blockers above the local horizon.

As in convex inversion, the observing geometries must be suitable and at large phase angles to emphasize the effects of nonconvexity as much as possible. A feasible albedo distribution can be found in the same way as in convex inversion. In Fig. 5 we show a nonconvex inversion result for object **2**; we adjusted the radii (15) when the vertex directions were simply those of a triangulated sphere, i.e., no a priori assumptions were used. The same sets of lightcurves were used as in convex inversion. The series was truncated at order and degree four since the effects of detailed nonconvexities are certainly drowned in the noise—this is why convex inversion actually offers better spatial resolution than nonconvex inversion. Nonconvex features in inversion results are typically more qualitative than quantitative: the existence of, say, valleys is indicated, but the depths of the valleys are not very precise (in the same manner that convex inversion only finds the possible locations of the valleys).

## 5. CONCLUSIONS AND DISCUSSION

We have found that although the shape part of the lightcurve inversion problem can in theory be cast in a linear form (which is characteristic of many imaging and other inverse problems), the best methods for posing and solving the problem are nonlinear. Nonlinearity is caused by the fact that some solved quantities always need to be positive if they are to describe a convex shape (and/or a real albedo distribution), and this positivity is automatically enforced by using exponential forms. What is more, the positivity constraint is quite sufficient by itself for removing the apparent ill-posedness of the problem. No particular regularization methods are necessary: the problem is stabilized simply by demanding that the result of inversion be restricted to feasible shapes. This implies that no two real convex shapes (or albedo distributions) can differ very much from each other and yet produce similar sets of lightcurves, if the sets include various observing geometries especially at large solar phase angles.

The main distinction between all possible methods is that either smooth functions or polyhedra can be employed; the latter describe sharp features in lightcurves (and on surfaces) better, while the former enable fast inversion and provide a good initial guess for other optimization methods. Smooth functions are also very useful when solving for, e.g., rotation parameters. Basically, one should solve for the shape simultaneously with the period and pole. Preliminary results indicate that adding the latter as free parameters to the Levenberg–Marquardt optimization procedure constitutes a very efficient technique even with significant noise and a poor initial guess. Additional parameters, for example, those of the solar phase function of the scattering law, can also be left adjustable. It thus appears that convex inversion

is a very powerful tool in determining not just the shape but any other free parameters as well.

Nonconvex inversion, although not as robust and stable as the convex approach, is apparently possible in the sense that a conception of the largest nonconvexities can be formed. Its success largely depends on the observing geometries, the initial guess, and the appropriateness of the chosen system of vertex directions and functions describing the surface. A contact binary, for example, may sometimes be better described by moving vertices along the radius directions of a horizontal cylindrical coordinate system rather than in the directions corresponding to spherical coordinates.

In a forthcoming paper, we shall investigate the above possibilities and apply our methods to real asteroid data.

## APPENDIX A: OCTANT TRIANGULATION

The surface is divided into eight octants (according to the coordinate axes), each of which is divided into  $N$  horizontal rows usually with equal  $(\pi/2N)$  spacing in polar angle. The first rows from the poles toward the equator have no azimuthal points between the octant lines; then, for every row closer to the equator, there is one azimuthal point more than in the previous row. The points on each row are evenly spaced in azimuthal angle. The facets are formed by joining each azimuthal point of a row to two points in the neighboring rows so that a regular mesh of triangles is created. In every triangle row closer to the equator there are two more triangles than in the previous one. The number of facets is  $8N^2$ , and the number of vertices  $4N^2 + 2$ .

As a result of triangulation, there are always only four facets at both ends of each coordinate axis (all other vertices are surrounded by six facets); thus the polar facets are about the same size as the equatorial ones for a roughly spherical body. If the body is very strongly elongated the equatorial facets will be larger, in which case one can use some other polar angle spacing when constructing the rows. A suitable number of rows is typically from 8 to 10.

## APPENDIX B: CONVEX HULL ALGORITHM

1. Find one point of the convex hull, for example, the point that has the largest/smallest value of  $x$ ,  $y$ , or  $z$ . This will be point  $a$ .
2. Find one of the points that are adjacent to point  $a$  in the convex hull. It is easiest to pick the one whose angular distance from the origin, as seen from point  $a$ , is the largest. Call this point  $b$ . Now the edge between points  $a$  and point  $b$  forms the first base line.
3. A new point of the convex hull is found by taking a reference point  $c$  and then starting to compare each of the input points with the plane formed by the base line and point  $c$ . If the compared input point  $d$  is above this plane,  $d$  becomes the new test point  $c$  and comparing continues from it. When all the input points have been gone through, the point that was left as point  $c$  belongs to the convex hull. Mathematically the comparing can be done as follows: whenever  $\mathbf{r}_{ad} \cdot (\mathbf{r}_{ac} \times \mathbf{r}_{ab}) > 0$ , point  $d$  is on the other side of the plane  $(\mathbf{r}_{ac}, \mathbf{r}_{ab})$  from the current  $c$  (and all the previous input points).
4. Replace old  $b$  with current  $c$ . Take line  $ab$  as the new base line.
5. Go to step 3 if point  $b$  is not the one that formed the first base line with  $a$ .
6. Now, having found all the vertices around point  $a$ , find a new center point  $a$ . This is the first point that is already included in the convex hull, but has not yet been a center point  $a$ . If there is no such point left, the convex hull has been found  $\rightarrow$  go to step 8.
7. The new  $b$  is any of the points connected to the new  $a$  in the convex hull. Now  $a$  and  $b$  form again the first base line. Go to step 3.
8. From the lists of vertices around each point, remove those points that do not define new planes, i.e., those that lie in the plane defined by the adjacent vertices.

The hull obtained the way described above consists of the lists of vertices connected to each vertex. From these, one can immediately form the circuits of vertices that define the facets.

## APPENDIX C: MINKOWSKI MINIMIZATION

As explained in KLLB and Lamberg (1993), the reconstruction of the convex polyhedron corresponding to given facet areas  $\mathbf{g}$  and surface normals can be expressed as a constrained minimization problem where  $\mathbf{l}$ , the distances of the facet planes from the origin, are to be solved for. The object function is the inner product  $\langle \mathbf{l}, \mathbf{g} \rangle$  in  $R^n$ -space, while the constraint function is  $V(\mathbf{l})$ , the volume of the polyhedron computed from  $\mathbf{l}$ . In practice, the equivalent procedure of maximizing  $V(\mathbf{l})$  while staying on the hyperplane  $\langle \mathbf{l}, \mathbf{g} \rangle = \text{constant}$  is computationally more efficient as the constraint function is now linear.

The volume  $V$  is given by

$$V = \frac{1}{3} \sum_{j=1}^n l_j A_j(\mathbf{l}), \quad (17)$$

where  $A_j$  is the area of a facet as computed from  $\mathbf{l}$ . The gradient of (17) is simply  $\mathbf{A}$ , and its projection onto the constraint plane is

$$\mathbf{f} = \mathbf{A} - \frac{\langle \mathbf{A}, \mathbf{g} \rangle}{\langle \mathbf{g}, \mathbf{g} \rangle} \mathbf{g}. \quad (18)$$

Using the projected gradient  $\mathbf{f}$  instead of the original  $\mathbf{A}$  in methods that employ gradient information to determine the direction of the iteration step ensures that the constraint is fulfilled. The starting point of the iteration must, of course, satisfy the constraint: a possible choice is, e.g., to set each  $l_j$  to  $\langle \mathbf{g}, \mathbf{g} \rangle / \sum g_j$ .

Constructing the polyhedron from  $\mathbf{l}$  is easiest via the so-called dual transform. This transform maps a plane with the surface unit normal  $\mathbf{n}$  and distance  $l$  from the origin into a point given by the radius vector

$$\mathbf{r} = \mathbf{n}/l, \quad (19)$$

and vice versa; i.e., a point given by  $\mathbf{r}$  is mapped into a plane with corresponding  $\mathbf{n}$  and  $l$ . The transform is well defined when  $l > 0$ . It thus maps the facets of a convex polyhedron into vertices and vice versa. The important point is that adjacency information is retained; i.e., the vertices of a facet become the facets surrounding a vertex, facets adjacent to each other become connected vertices, etc. Thus, given the vector  $\mathbf{l}$  and the corresponding surface normals, one first performs the dual transformation and finds the convex hull of the points obtained (using, e.g., the procedure of Appendix B). The facets of the convex hull in dual space are then transformed into vertices in home space; the vertices belonging to a facet are obtained from the information on the facets surrounding a vertex of the convex hull in dual space. This completes the construction of the convex polyhedron corresponding to  $\mathbf{l}$ ; the computation of  $A_j(\mathbf{l})$  is simple geometry.

It is useful to shift the centroid of the polyhedron to the origin at each iteration step; i.e., the elements  $l_i$  of the present vector  $\mathbf{l}$  are changed to  $l_i - \mathbf{n}_i \cdot \mathbf{r}_c$ , where  $\mathbf{r}_c$  is the computed position of the centroid. If the final vertices are to correspond to the facet areas given at the beginning of the iteration, each vertex coordinate must be scaled with the factor  $\sqrt{|\mathbf{A}|/|\mathbf{g}|}$  at the end of the iteration (if the initial guess for  $\mathbf{l}$  is the one given above). The direction and size of the iteration step can be determined using standard methods such as conjugate gradients; when implementing line minimization, any trial steps (in, e.g., bracketing the minimum) leading to negative values for any  $l_i$  must be contracted back to the positive region by, e.g., bisection. Lamberg (1993) avoids explicit line minimization by estimating suitable step sizes for the steepest descent method.

## ACKNOWLEDGMENTS

It is a pleasure to thank Karri Muinonen for suggesting an efficient triangulation scheme, as well as for several other valuable comments, and Jukka Piironen for many interesting discussions and useful remarks.

## REFERENCES

- Barucci, M. A., A. Cellino, C. De Sanctis, M. Fulchignoni, K. Lumme, V. Zappala, and P. Magnusson 1992. Ground-based Gaspra modelling: Comparison with the first Galileo image. *Astron. Astrophys.* **266**, 385–394.
- Cellino A., V. Zappala, and P. Farinella 1989. Asteroid shapes and lightcurve morphology. *Icarus* **78**, 298–310.
- Hudson, R. S., and S. J. Ostro 1994. Shape of Asteroid 4769 Castalia (1989 PB) from inversion of radar images. *Science* **263**, 904–943.
- Hudson, R. S., and S. J. Ostro 1995. Shape and non-principal axis spin state of Asteroid 4179. *Science* **270**, 84–86.
- Hudson, R. S., and S. J. Ostro 1999. Physical model of Asteroid 1620 Geographos from radar and optical data. *Icarus* **140**, 369–378.
- Kaasalainen, M., L. Lamberg, K. Lumme, and E. Bowell 1992a. Interpretation of lightcurves of atmosphereless bodies. I. General theory and new inversion schemes. *Astron. Astrophys.* **259**, 318–332.
- Kaasalainen, M., L. Lamberg, and K. Lumme 1992b. Interpretation of lightcurves of atmosphereless bodies. II. Practical aspects of inversion. *Astron. Astrophys.* **259**, 333–340.
- Lamberg, L. 1993. *On the Minkowski Problem and the Lightcurve Operator*. Academia Scientiarum Fennica, Series A, I. Mathematica dissertationes 87, University of Helsinki.
- Magnusson, P., M. A. Barucci, J. Drummond, K. Lumme, S. J. Ostro, J. Surdej, R. C. Taylor, and V. Zappala 1989. Determination of pole orientations and shapes of asteroids. In *Asteroids II* (R. P. Binzel, T. Gehrels, and M. S. Matthews, Eds.), pp. 67–97. Univ. of Arizona press, Tucson.
- Magnusson, P., and 46 colleagues 1996. Photometric observations and modeling of Asteroid 1620 Geographos. *Icarus* **123**, 227–244.
- Michalowski, T. 1996. A new model of the asteroid 532 Herculina. *Astron. Astrophys.* **309**, 970–978.
- Mottola, S., and F. Lahulla 2000. Mutual eclipse events in asteroid binary system, 1996 FG3: Observations and a numerical model. *Icarus* **146**, 556–567.
- Muinonen, K. 1998. Introducing the Gaussian shape hypothesis for asteroids and comets. *Astron. Astrophys.* **332**, 1087–1098.
- Muinonen, K., and J. S. V. Lagerros 1998. Inversion of shape statistics for small Solar System bodies. *Astron. Astrophys.* **333**, 753–761.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling 1994. *Numerical Recipes in Fortran*. Cambridge Univ. Press, Cambridge, UK.
- Thomas, P. C., M. J. S. Belton, B. Carcich, C. R. Chapman, M. E. Davies, R. Sullivan, and J. Veverka 1996. The shape of Ida. *Icarus* **120**, 20–32.
- Thomas P. C., J. Veverka, D. Simonelli, P. Helfenstein, B. Carcich, M. J. S. Belton, M. E. Davies, and C. Chapman 1994. The shape of Gaspra. *Icarus* **107**, 23–36.
- Thomas, P. C., and 11 colleagues 1999. Mathilde: Size, shape and geology. *Icarus* **140**, 17–27.