The Astronomical Journal, 157:42 (20pp), 2019 February https://doi.org/10.3847/1538-3881/aaf565

Shadow Imaging of Transiting Objects

Emily Sandford and David Kipping
Department of Astronomy, Columbia University, 550 W 120th St. New York, NY 10027, USA; esandford@astro.columbia.edu
Received 2018 September 20; revised 2018 November 14; accepted 2018 November 30; published 2019 January 11

Abstract

We consider the problem of inferring the shape of a transiting object's silhouette from its light curve alone, without assuming a physical model for the object. We model the object as a grid of pixels which transits a star; each pixel has an opacity, ranging from transparent to opaque, which we infer from the light curve. We explore three interesting degeneracies inherent to this problem, in which markedly different transiting shapes can produce identical light curves: (i) the "flip" degeneracy, by which two pixels transiting at the same impact parameter on opposite sides of the star's horizontal midplane generate the same light curve; (ii) the "arc" degeneracy, by which opacity can be redistributed along the semicircular arc of pixels which undergoes ingress or egress at the same time without consequence to the light curve; and (iii) the "stretch" degeneracy, by which a wide shape moving fast can produce the same light curve as a narrow shape moving more slowly. By understanding these degeneracies and adopting some additional assumptions, we are able to numerically recover informative shadow images of transiting objects, and we explore a number of different algorithmic approaches to this problem. We apply our methods to real data, including the TRAPPIST-1c/e/f triple transit and two dips of Boyajian's Star. We provide `Python` code to calculate the transit light curve of any grid and, conversely, infer the image grid which generates any light curve in the software package accompanying this paper, EightBitTransit (https://github.com/esandford/EightBitTransit).

Key words: methods: analytical – methods: statistics – planetary systems – planets and satellites: general

1. Introduction

Transit light curves are rich in information. If we assume a physical model for a transiting object—usually, a spherical body in a Keplerian orbit about a host star—we may then infer the parameters of this model, including physical properties of the transiter, its orbit, and the host star, from the light curve.

However, anomalous transit-like events, such as those observed in the star KIC 8462852 (Boyajian et al. 2016), resist this type of analysis, because their physical cause—and consequently the appropriate model—is not apparent. In this paper, we consider the general problem of inferring the transiting shape, or shadow image, that generated a particular light curve. We wish to infer this image from the light curve alone, with as few additional assumptions as possible.

A number of problems related to shadow imaging have been studied before. The inverse problem, of how to calculate the light curve of an arbitrary transiting shape, has been tangentially addressed by several numerical transit-light-curve-calculating codes. Generally, however, these assume some parametric model for the transiting object: in the case of `BATMAN` (Kreidberg 2015), the transiting object must be a spherical planet; of `LUNA` (Kipping 2011), a spherical planet accompanied by a spherical moon; of `PyTranSpot` (Juvan et al. 2018), circular starspots projected on a stellar surface; and of the Universal Transit Modeller (Deeg 2009), a planet with moons or rings.

Meanwhile, significant advances have been made in another problem closely related to shadow imaging: eclipse mapping, which attempts to reconstruct the two-dimensional surface features of an exoplanet undergoing secondary eclipse from the light reflected off the planet's surface as it disappears and reappears from behind a star. Majeau et al. (2012) and de Wit et al. (2012) were the first to demonstrate this method, on hot Jupiter HD 189733b. Kawahara & Fujii (2011) extended this theory to surface mapping of exoplanets in face-on orbits using scattered light, and Farr et al. (2018) recently released the `exocartographer` code to carry out surface mapping in a fully Bayesian framework with robust uncertainty estimation. Berdyugina & Kuhn (2017) showed that next-generation coronagraphic telescopes will be able, using these techniques, to map the surface of a handful of nearby planets, including Proxima b.

Analogous two-dimensional mapping methods have been successfully applied to the problem of starspot inversion, or the deduction of the pattern of starspots responsible for time variations in the spectrum or light curve of a star. Goncharskii et al. (1982) were among the first to attempt starspot inversion, aiming to explain spectral variations in Ap stars by inferring the pattern of chemical inhomogeneities on the surface that would generate them. Vogt & Penrod (1983) introduced Doppler imaging to infer maps of starspots on rapidly rotating stars from time series spectra, and Vogt et al. (1987) refined the technique by introducing maximum entropy regularization as a means of choosing from a set of degenerate solutions to the same observations. Piskunov et al. (1990) compared the maximum entropy method, which prefers a solution with the minimum spatial correlation between points on the star's surface, to an alternative constraint, Tikhonov regularization, which prefers the smoothest possible pattern of starspots that matches the observations. Similar techniques, with varying choices of regularization, have been applied to stellar light curves by, e.g., Lanza et al. (1998).

In this work, we build upon these techniques to develop a mathematical and numerical treatment of shadow imaging which has a similar geometric setup to and is subject to similar degeneracies as eclipse mapping and starspot inversion. In Section 2, we investigate, analytically, the degeneracies inherent to the light curve imaging problem. We explain how discretizing the problem—modeling the transiting object as a

grid of pixels of fixed opacity, rather than as a smooth, continuous image—allows us to make progress in the problem despite these degeneracies. In Section 3, we define the pixel grid model, which can be used to represent any transiting object and explain how to calculate its light curve. In Section 4, we consider how, starting from a transit light curve, we may infer the pixel grid image which generated it, and we discuss the results of this inference in a number of test cases. In Section 5, we consider the results of light curve inversion in the real cases of the TRAPPIST-1c/e/f triple transit and the anomalous transits observed in Boyajian's Star. We conclude in Section 6.

2. Transit Degeneracies

Calculating the light curve of a transiting object is an act of projection. It begins with a three-dimensional object in space, projected against the sky to make a two-dimensional image. At a few discrete points in time, as this image crosses a star, the starlight that the image does not block is summed, and the sums are strung together to make a light curve: a one-dimensional time series.

Deducing the image that generated a particular light curve, therefore, is a problem of inferring two-dimensional data from one-dimensional data. As such, we do not expect to find a unique solution to match each light curve. Vogt et al. (1987), Piskunov et al. (1990), Majeau et al. (2012), and de Wit et al. (2012) note similar degeneracies in starspot inversion and eclipse mapping.

We begin by examining mathematically the degeneracies inherent to the problem of inferring the shape that generated a particular light curve. We operate under the assumptions, discussed further in Section 3, that the occulting shape is unchanging in time and moving at a constant velocity across the star; that the star is spherical and of uniform brightness; and that the observed light curve is well sampled in time.

2.1. The Flip Degeneracy

The first important degeneracy in the shadow imaging problem results from the reflection symmetry of the star about its horizontal midplane. An opaque shape that transits at an impact parameter b above the midplane produces the same light curve as a "flipped" shape that transits below the midplane.

In planetary transit modeling, this degeneracy can be ignored, because the sign of the impact parameter $b = \cos i \left(\frac{a}{R_*}\right)\left(\frac{1-e^2}{1+e\sin\omega}\right)$ is a function of the inclination angle i of the planet's orbital plane and does not describe any inherent property of the transiting planet. However, if we wish to model more general transiting shapes, we must consider the full space of flip-degenerate solutions.

To express the degree of flip degeneracy in a given shadow imaging problem, we consider an image made up of a grid of opaque and transparent pixels, N rows by M columns. (See Figure 1 for an example.) Although there are 2^{NM} unique permutations of opaque and transparent pixels arranged in this grid, each of these permutations does not yield a unique light curve, and in general, a light curve cannot be inverted to produce a unique pixel grid shadow image.

We can express the degree of degeneracy by calculating the number of unique light curves, $U_{\rm LC}$, possible for this N-row by M-column grid,

$$U_{\rm LC} = \begin{cases} (2 \times 3^{\frac{(N-1)}{2}})^M, & N \text{ odd} \\ (3^{\frac{N}{2}})^M, & N \text{ even.} \end{cases} \quad (1)$$

For intuition, consider first the even-N case. In each of the M columns, there are $\frac{N}{2}$ pixels above the midplane; each of these has a counterpart below the midplane with the same impact parameter. There are four possible opacity states for this pair of

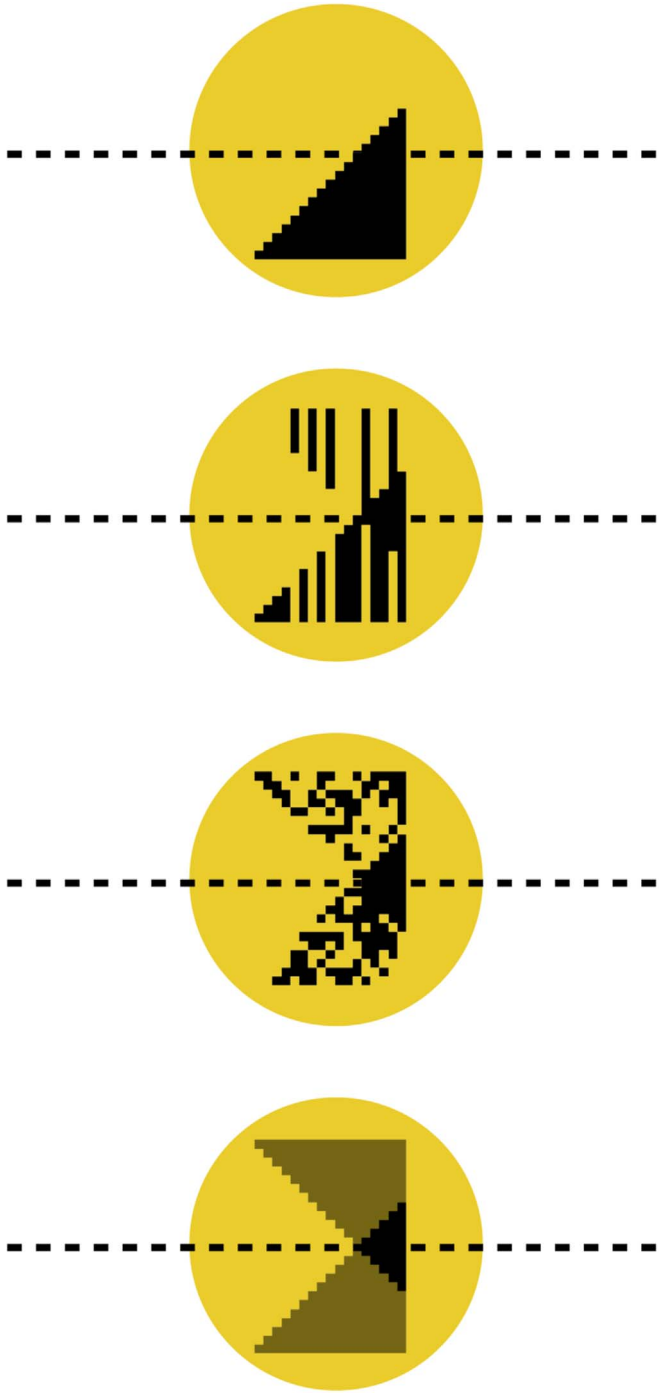

Figure 1. Four transiting binary-opacity pixel images which generate the same light curve. The bottom pixel image (opaque black pixels have $\tau = 1$; semitransparent gray pixels have $\tau = 0.5$) is the average of the full set of flip-degenerate solutions which match this image's light curve.

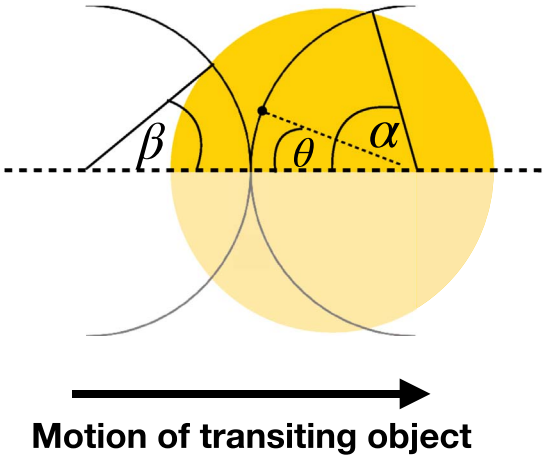

Figure 2. A pair of arcs which generates the same light curve as a single opaque point transiting along the horizontal midplane of the star. For this shape to generate a perfect boxlike transit, the arcs must be infinitely thin and cannot be of uniform opacity; rather, opacity must be distributed symmetrically along them as a function of θ.

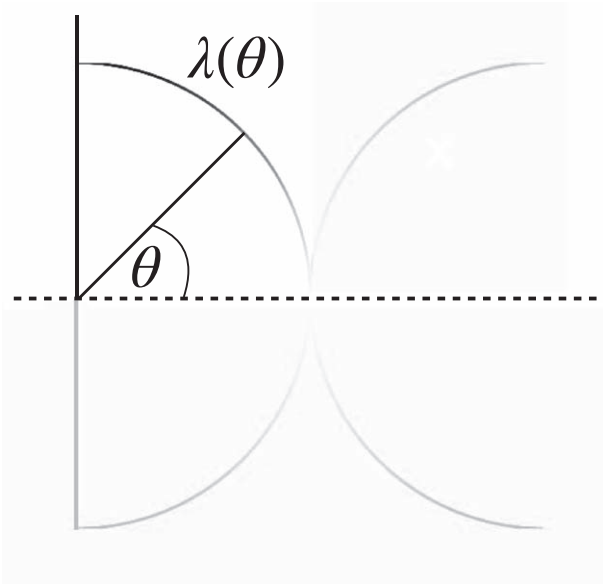

Figure 3. θ represents the angle from the horizontal midplane of either arc to any point along it. $\lambda(\theta)$ represents the opacity of the arc at θ. We wish to solve for $\lambda(\theta)$ such that the arc pair can produce a flat-bottomed transit.

pixels: 00, 01, 10, or 11. However, because of the flip degeneracy, the 01 and 10 cases produce the same light curve, so only three arrangements are unique—hence, three unique opacity values for each degenerate pixel pair, raised to the power of the total number of pixels above the midplane.

The arithmetic in the odd-N case is the same, except that the middle pixel row has no across-midplane counterpart. Each pixel in that row may only take on an opacity of 0 or 1.

In the case of a square, 3×3 pixel grid, there are $2^9 = 512$ unique permutations of opaque and transparent pixels, but only 216 unique light curves.

As a result, the binary-opacity pixel grid solution to any given light curve inversion is not unique, *unless* the light curve was, in truth, generated by a grid of binary-opacity pixels ($\tau = 0$ or 1) that is symmetrical about its horizontal midplane. Physically, we would only expect such a situation in the case of a perfectly spherical planet, or perhaps a planet–moon system or ringed planet, transiting a star at an impact parameter of 0.

In general, therefore, the inverted pixel grid which generates a light curve is not unique. Figure 1 shows four transiting pixel images which generate identical light curves. Starting with the pixel image at the top and flipping any pixel about the horizontal midplane leaves the light curve unchanged. The pixel image in the bottom panel is the average of the full set of flip-degenerate solutions.

We hope, therefore, to recover shadow images analogous to this bottom panel, which represent a kind of "superposition" of the full set of flip-degenerate solutions to a particular light curve.

2.2. The Arc Degeneracy

There is, however, another degeneracy inherent to the shadow imaging problem by which the set of physically allowable images matching any particular light curve becomes infinitely large. This degeneracy allows a transiting pair of semicircular arcs to generate the same light curve as a single opaque point, and we term it the "arc" degeneracy.

Figure 2 illustrates the geometry of the pair of arcs which generates the same light curve as an infinitesimally small opaque point transiting exactly along the horizontal midplane of the star. Consider this shape to transit from left to right across the star: because the right-hand arc traces the shape of the stellar limb, the entire right-hand arc will ingress upon the star at the same moment, yielding the same vertical ingress feature in the light curve that we would expect from an infinitesimally small transiting planet. (A correspondingly sharp egress feature in the light curve happens when the left-hand arc egresses all at once sometime later.)

After the moment of ingress, the top- and bottommost edges of the right-hand arc immediately egress again. However, this egress is balanced by the ingress of the middle of the left-hand arc. If opacity is appropriately distributed along each arc, then the ingress of the left-hand arc and the egress of the right-hand arc may balance exactly. Here, we derive the functional form of the opacity distribution along the arc to allow this exact balance.

Let α (see Figure 2) denote the angle between the horizontal midplane of the star and the point of intersection between the stellar limb and the right-hand arc (which ingresses first). At the moment of ingress, $\alpha = \frac{\pi}{2}$; at the moment of egress, $\alpha = 0$. Let β denote the angle corresponding to the point of intersection on the left-hand arc, and let β range from 0 at ingress to $\frac{\pi}{2}$ at egress.

Let θ represent an angle measured from the horizontal midplane of either arc to its outermost point, and let $\lambda(\theta)$ represent the opacity along the arc as a function of this angle. Figure 3 illustrates this setup. Note that $\lambda(\theta)$ cannot be constant, because, for example, during some small time interval dt immediately after the moment of ingress, the length of the right-hand arc which egresses is greater than the length of the left-hand arc which ingresses.

Let T be the duration of the transit of the pair of arcs (in other words, the interval between the moment of ingress and the moment of egress). Let the moment of ingress happen at $t = 0$, and let us define a dimensionless time coordinate $\kappa = \frac{t}{T}$ to parameterize the progress of the transit. At ingress then, $\kappa = 0$, and at egress, $\kappa = 1$.

Following these definitions, we may write

$$\cos\alpha = \kappa, \tag{2}$$

$$\cos\beta = 1 - \kappa. \tag{3}$$

The total opacity $L(\kappa)$ transiting the star at a particular moment κ is equal to

$$L(\kappa) = \int_0^{\alpha(\kappa)} \lambda(\theta)d\theta + \int_0^{\beta(\kappa)} \lambda(\theta)d\theta. \tag{4}$$

For the transit to be flat-bottomed, we require that $L(\kappa)$ be constant, or that $\frac{dL}{d\kappa} = 0$. Differentiating both sides of Equation (4) by κ, we obtain

$$\frac{dL}{d\kappa} = \lambda(\alpha)\frac{d\alpha}{d\kappa} + \lambda(\beta)\frac{d\beta}{d\kappa}, \tag{5}$$

because λ is time-independent and therefore independent of κ.

Setting this expression equal to 0 and substituting, we obtain

$$0 = \lambda(\alpha)\frac{1}{\sqrt{1-\kappa^2}} + \lambda(\beta)\frac{1}{\sqrt{1-(1-\kappa)^2}}, \tag{6}$$

or

$$\frac{\lambda(\alpha)}{\lambda(\beta)} = -\frac{\sqrt{1-\kappa^2}}{\sqrt{1-(1-\kappa)^2}}. \tag{7}$$

By the definitions of α and β, we may write

$$\frac{\lambda(\alpha)}{\lambda(\beta)} = -\frac{\sqrt{1-\cos^2\alpha}}{\sqrt{1-\cos^2\beta}} = \frac{\sin\alpha}{\sin\beta}. \tag{8}$$

By inspection then,

$$\lambda(\theta) \propto \sin\theta, \tag{9}$$

whose sign we choose to be positive because physically meaningful opacities are between 0 and 1.

The overall normalization of $\lambda(\theta)$ sets the transit depth of the arcs' light curve. Figure 4 shows a transiting arc pair with $\lambda(\theta) = \sin\theta$.

We note that there are two other solutions to $\lambda(\theta)$ that satisfy the condition $\frac{dL}{d\kappa} = 0$. The first is the trivial solution, $\lambda(\theta) = 0$. The second is a Dirac delta function at $\theta = 0$,

$$\lambda(\theta) \propto \delta(\theta = 0), \tag{10}$$

where again the overall normalization sets the transit depth.

For intuition, the two nontrivial solutions to $\lambda(\theta)$ given by Equations (9) and (10) represent two extremes: the least and most compact arrangements of opacity, respectively, that produce the same flat-bottomed, boxlike transit. Any linear combination of these solutions also satisfies $\frac{dL}{d\kappa} = 0$ and generates a boxlike transit.

The above derivation demonstrates that a pair of arcs of variable opacity can match the transit shape of an infinitesimal point of opacity transiting along the horizontal midplane of the star, at impact parameter $b = 0$. The same logic applies to an infinitesimal point at an arbitrary impact parameter b. Figure 5 illustrates the geometry of this situation.

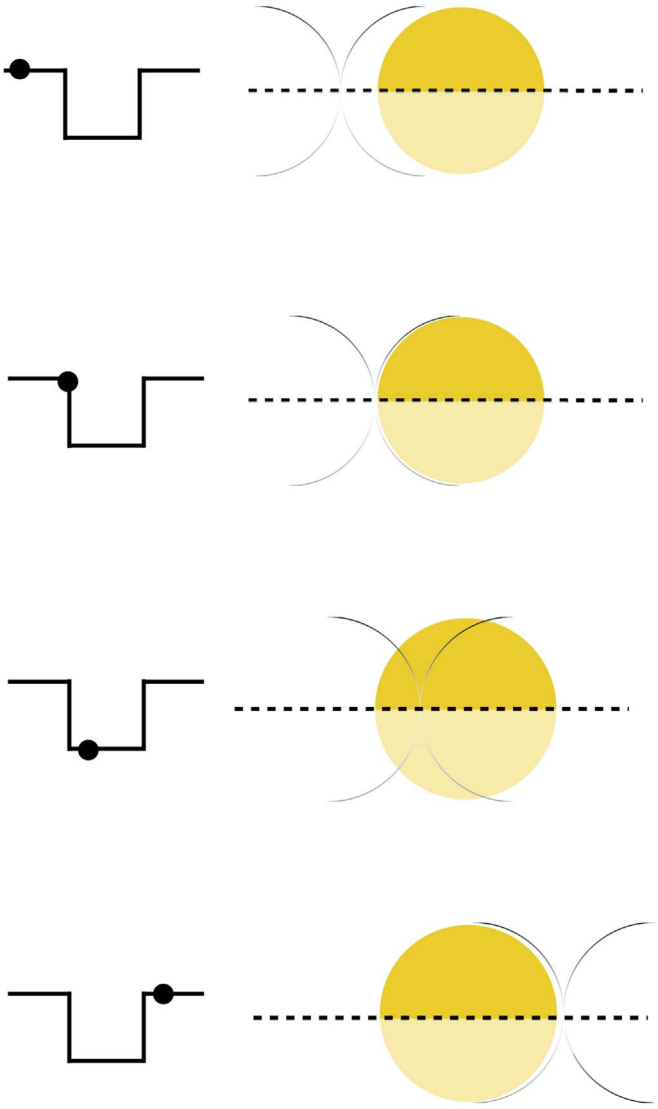

Figure 4. A transiting arc pair with opacity distributed as $\lambda(\theta) = \sin\theta$. This shape generates a boxlike transit light curve. The circles in the left-hand panels mark the time along the transit at which the right-hand panels occur.

Mathematically, a change in the impact parameter b means that the limits of integration in Equation (4) change,

$$L(\kappa) = \int_{\arcsin b}^{\alpha(\kappa)} \lambda(\theta)d\theta + \int_{\arcsin b}^{\beta(\kappa)} \lambda(\theta)d\theta. \tag{11}$$

Since b is constant, the subsequent steps and resulting solutions for $\lambda(\theta)$ do not change, except that the delta function solution is localized at $\theta = \arcsin b$.

We note finally that the arc degeneracy technically only operates for an occulter transiting a uniformly bright star: if the star is limb-darkened, then there is no (unchanging) arc arrangement which can maintain the perfect opacity ingress–egress balance described by Equation (6). However, in practice, the limited time resolution of light curve observations leaves room for significant arc-degenerate behavior in shadow images recovered from real transit data (see Section 5 below).

2.3. The Stretch Degeneracy

A third degeneracy inherent to light curve imaging results from the "scale-free" nature of the problem and allows a wide

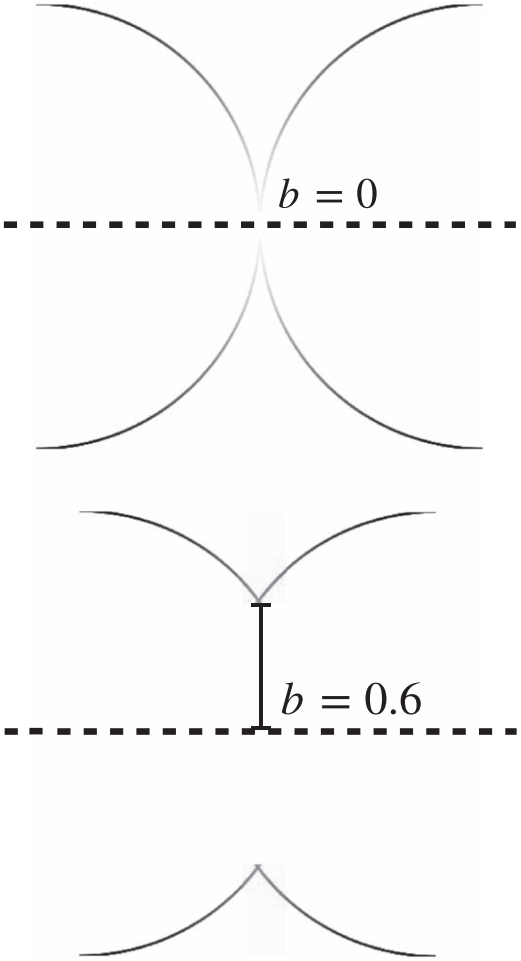

Figure 5. A pair of truncated arcs, as illustrated in the lower panel, can match the transit shape of an infinitesimal opaque point transiting at an arbitrary impact parameter.

image moving at high velocity to generate the same light curve, within an arbitrarily small measurement uncertainty, as a narrower image moving at lower velocity. We term this degeneracy the "stretch" degeneracy.

The stretch degeneracy is mathematically simpler than the arc and flip degeneracies. Two occulters with the same transit duration T both obey

$$T = \frac{W}{v}, \tag{12}$$

where W is the width of the occulter and v is its velocity. The right-hand side of this equation can be multiplied by the same constant in the numerator and denominator without consequence to T. In other words, a "stretched" image traveling fast can generate a transit event of the same duration as a narrow image traveling slowly.

Figure 6 illustrates the stretch degeneracy for a simple, low-resolution circular occulter. Note in particular two features of the "stretched" image: first, that it is semiopaque rather than fully opaque like the unstretched image, and second, that its edges are less opaque than its middle. The semiopacity of the stretched image is necessary in order to match the transit depth of the unstretched image: because the stretched image is wider, it occults more of the stellar surface, so it must let some light through or it will produce a much deeper transit than the

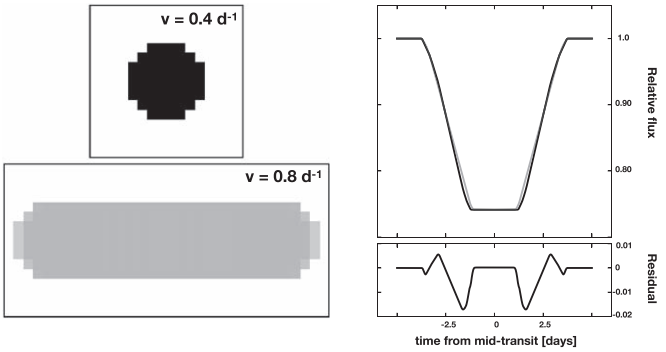

Figure 6. Two transiting images with light curves that differ by $\mathcal{O}(1\%)$. The lower image transits at a velocity twice that of the upper image. Note that the left- and rightmost edges of the lower image are slightly less opaque than the middle, an adjustment made to better match the ingress and egress shape of the upper image's light curve. At higher resolution for the lower image, an even better match to the upper image's light curve could be found.

unstretched image. Meanwhile, the lightened edges of the stretched image are necessary to better match the ingress and egress shape of the unstretched image's light curve; with arbitrarily high image resolution, it is possible to match the unstretched image's light curve to arbitrary accuracy.

In practice, the stretch degeneracy is the least important of the three nontrivial degeneracies we explore in this section, because a fast-transiting, stretch-degenerate image can only match a narrow, slower image's light curve if the image resolution is high enough, as suggested by the example in Figure 6. For real data, image resolution is constrained by the number of observed data points, which causes us to prefer the narrowest, slowest possible image which can match an observed light curve (see Section 4.2 for further discussion).

2.4. Trivial Degeneracies

Finally, we note two trivial degeneracies which do not affect the inference of a shadow image. The first relates to the arbitrary sign of the velocity of the transiter; an image which transits left to right across the star generates the same light curve as the same image, horizontally mirrored, transiting right to left across the star at the same velocity. We choose positive v to indicate that the image transits left to right (see Section 3.2 below).

The second trivial degeneracy relates to a time translation of the entire transit event. As we discuss in Section 3.2, we must choose a "reference time," analogous to a transit midpoint time, along a light curve in order to recover a shadow image; shifting this reference time forward or backward along the light curve results in a shadow image which is shifted right or left, respectively (given our choice of v direction above).

3. Model: Generating a Light Curve from a Discretized Image

By the arguments of Section 2, a given light curve may be generated by infinitely many images. To constrain the solution set, we therefore conclude that it is necessary to impose further constraints on the shadow image. (Starspot inversion requires an analogous constraint—popular choices include the maximum entropy principle, which chooses the solution with minimum spatial correlation between points on the stellar surface, and Tikhonov regularization, which chooses the

smoothest solution, or the solution with the minimum spatial derivative.)

In this section, we define a forward model for generating a light curve, sampled at discrete time intervals, from a pixelated image. This simulated light curve can be compared to observations of a real transit event. After we establish this forward model, we will investigate the inverse problem, of how to infer a pixelated image from an observed light curve, in the next section. We return to the question of degeneracies in Section 4.3.

3.1. Discretizing the Image

Pixelating, or discretizing, the shadow image is motivated by recognizing that real light curves are themselves discrete time series. A light curve is not infinitely resolved in time, and therefore we should not attempt to recover a shadow image that is infinitely resolved spatially. Similarly, each flux measurement in a light curve has an associated uncertainty; we should not attempt to recover a shadow image with pixel elements too small to be definitively detected within that uncertainty (see Section 4.2 below for further discussion).

Furthermore, discretizing the pixel image enables us to investigate two physical variants of the shadow imaging problem:

1. What if the transiting object which generated the light curve is a solid body and therefore our shadow image should only admit of completely transparent (opacity $\tau = 0$) or completely opaque ($\tau = 1$) pixels?
2. What if the transiting object is dusty or translucent or is a solid body smaller than the pixel scale and our shadow image can contain pixels of intermediate opacity ($0 \leqslant \tau \leqslant 1$)?

These two variations of the shadow imaging problem have different constraints on the pixel opacities and require different mathematical approaches to inversion. In case (1), discretizing the pixel image is necessary to divide it up into opaque and transparent elements. In case (2), discretizing the pixel image enables us to set up the light curve inversion problem as a single-matrix equation (Equation (35) below) and to explore both analytic and numerical approaches to solving this equation. (Similar mathematical formulations exist for both starspot inversion (Vogt et al. 1987) and eclipse mapping, e.g., Berdyugina & Kuhn 2017.)

The same forward model, or procedure for generating a light curve from a pixelated image, can be used in both cases, so we begin there. How do we calculate the light curve of a pixelated image grid transiting a star?

3.2. Grid Definitions and Positions

We consider a pixel grid of N rows and M columns transiting a star. We normalize the physical scale of the problem such that the radius of the star is unity.

The grid lives in the X–Y sky-projected plane, with the observer at $Z = +\infty$. The grid moves laterally along the X axis, with $dX/dt > 0$, and does not translate up or down (i.e., $dY/dt = 0$). We illustrate this setup in Figure 7.

We treat the grid as moving at a constant lateral velocity $v \equiv dX/dt$, where $dv/dt \equiv 0$. This is a reasonable approximation over the timescale of a transit, unless the object resides in a very tight orbit or the object is near the pericenter in a highly eccentric orbit. We define positive v to mean that the grid transits from left to right across the star, such that the rightmost column of pixels ingresses first.

We define the vertical position of the grid such that the top of the highest row of pixels falls at $Y = 1$ and the bottom of the lowest row of pixels falls at $Y = -1$. In this way, the grid perfectly overlaps with the star in the vertical dimension.

This definition sets the size of each pixel to have a width w of

$$w = 2/N. \tag{13}$$

We emphasize that every pixel has the same square shape with this dimension. For $N = 1$ then, $w = 2$ and is thus equal to the diameter of the star.

To refer to individual pixels, we adopt the index notation $i \in [1, N]$ to denote the row and $j \in [1, M]$ to denote the column. To calculate the amount of stellar flux the grid blocks at each discrete time step t_k of the transit observation, we must first calculate the X and Y positions of each grid pixel i, j at each time step.

We may write the Y position of the center of pixel i, j as

$$Y_{i,j} = 1 - (w/2) - (i - 1)w, \tag{14}$$

where setting $i = 1$ recovers $Y_{1,j} = 1 - (w/2)$ and setting $i = N$ recovers $Y_{N,j} = -1 + (w/2)$. The Y positions of the grid pixels are constant.

For the X positions of the pixels, which evolve in time, we first define a reference X position for each pixel at a reference time $t = t_{\rm ref}$ as

$$X_{i,j}^{\rm ref} \equiv X_{i,j}[t = t_{\rm ref}]. \tag{15}$$

We choose the reference time such that the grid is centered on the star at $t = t_{\rm ref}$. Therefore:

$$X_{i,j}^{\rm ref} = (j - j_{\rm mid})w, \tag{16}$$

where

$$j_{\rm mid} = 1 + (M - 1)/2. \tag{17}$$

We may now write the time-evolving X position of the center of any pixel as

$$X_{i,j}(t_k) = X_{i,j}^{\rm ref} + (t_k - t_{\rm ref})v, \tag{18}$$

where t_k is the kth time index and $k \in [1, K]$. Practically speaking, $t_{\rm ref}$ is analogous to the transit midtime fitted in conventional transit models.

We may use the above equation for the time-evolving $X_{i,j}$ to solve for the time at which the grid makes first and last contact with the star, $t_{\rm enter}$ and $t_{\rm exit}$. The grid moves from left to right across the star, so at first contact, the Mth column of pixels has an X position equal to $-1 - (w/2)$, and for the last contact the first column of pixels has an X position equal to $1 + (w/2)$, giving

$$t_{\rm enter} = t_{\rm ref} - \frac{1 + Mw/2}{v}, \tag{19}$$

$$t_{\rm exit} = t_{\rm ref} + \frac{1 + Mw/2}{v}. \tag{20}$$

We assign a time-independent opacity $\tau_{i,j}$ to each pixel. $\tau_{i,j}$ is a binary value equal to zero or unity—in other words, we construct a grid of perfectly transparent pixels ($\tau_{i,j} = 0$) and opaque pixels ($\tau_{i,j} = 1$).

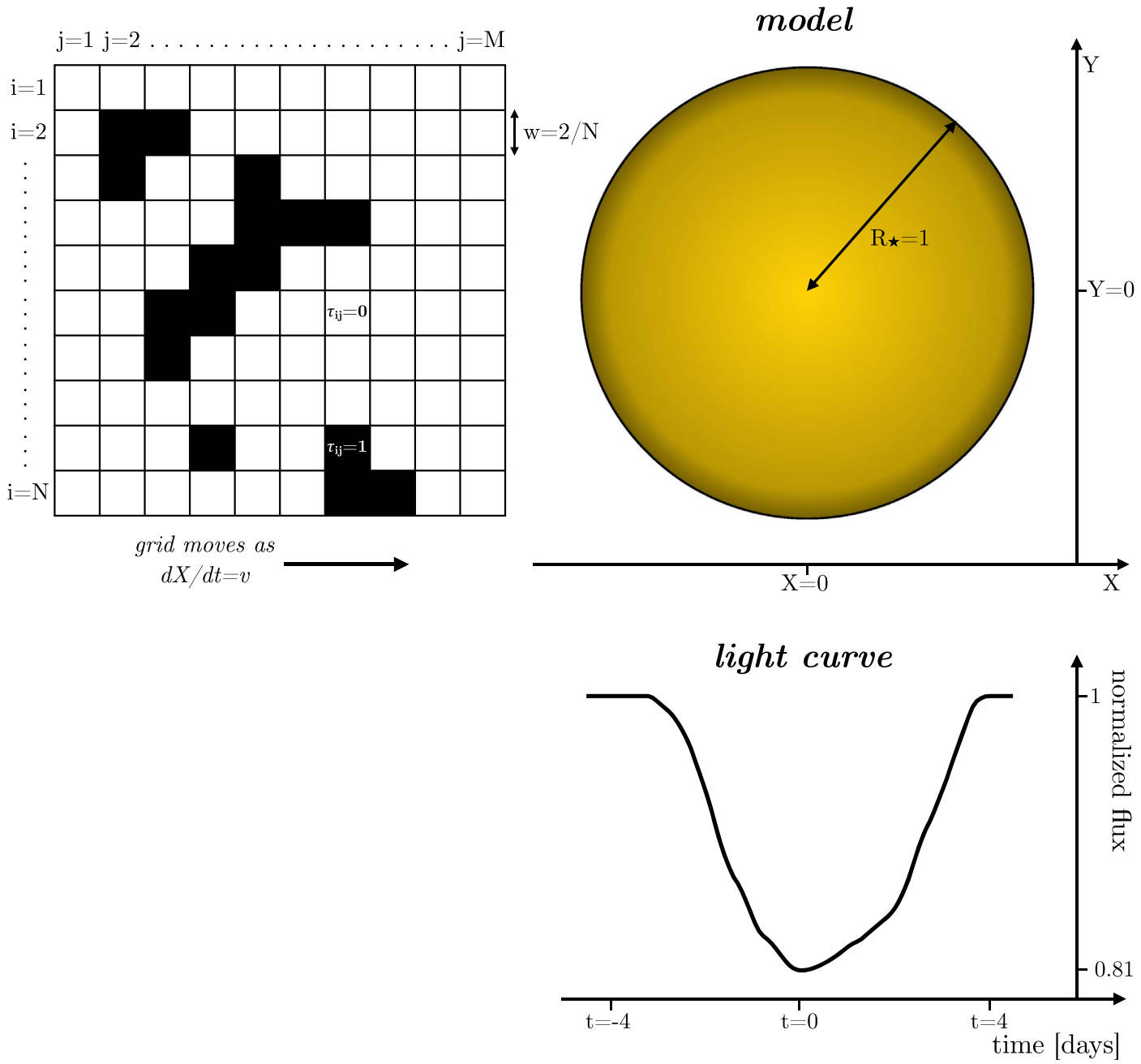

Figure 7. (Top panels) Illustration of an $N = 10$ by $M = 10$ binary-opacity grid model with 16 opaque pixels. The star itself is not pixelated; rather, the pixelated grid transits across the star, and the exact area of overlap between each square pixel and the star is evaluated at each discrete time step in order to generate a light curve. (Bottom panel) The light curve generated when this grid transits across a uniformly bright star at $v = 0.4$ days^{-1}, $t_{\rm ref} = 0$ days.

In total then, our model has MN opacity parameters, which are binary-valued (case 1) or real numbers between 0 and 1, inclusive (case 2), and two auxiliary parameters, $t_{\rm ref}$ and v, which are real-valued.

3.3. Computing the Light Curve of a Pixel

As pixel i, j transits the star, it occludes a fractional area $A_{i,j}(t_k)$ of the stellar disk at time t_k; $A = 0$ for pixels which do not overlap the star, and $A = \frac{w^2}{\pi}$ for pixels which overlap completely (because we choose $R = 1$, the area of the entire stellar disk is equal to π).

If we assume that the stellar disk is uniformly bright (i.e., there is no limb darkening), we may then compute the light curve $\boldsymbol{F}(t)$ of the transiting grid by recognizing that the fractional flux *blocked* by the grid at each time step t_k is equal to the fractional area of the star occulted by nontransparent pixels ($\tau_{i,\,j} > 0$), in proportion to their opacity. Therefore, the *unocculted* flux at time t_k is given by:

$$F(t_k) = 1 - \sum_{i=1}^{N} \sum_{j=1}^{M} \tau_{i,j} A_{i,j}(t_k). \tag{21}$$

This is the equation for the transit light curve, normalized such that $F = 1$ out of transit.

We emphasize that while opacities $\tau < 0$ and $\tau > 1$ are mathematically permissible in this equation, they are unphysical: $\tau < 0$ would represent a transiting pixel brighter than the

stellar surface it occulted, and $\tau > 0$ would describe a pixel that blocked more than its proper area's worth of starlight.

We compute the area of overlap $A_{i,j}(t_k)$ of pixel i, j at time step t_k from the (X, Y) position of the pixel's center at t_k, given by Equations (14) and (18), and the pixel's width, given by Equation (13). When a pixel partially overlaps the star, we approximate its overlap area as either a triangle, a trapezoid, or a square missing a triangular corner. We choose the appropriate overlap shape by computing the number of intersection points between the edge of the star and the sides of the pixel and also noting whether the center of the pixel falls inside or outside the star. We then correct this approximation by using the length of the chord between intersection points to calculate the area of the sliver of occluded star yet unaccounted for by this approximation.

For a limb-darkened star, we must also account for the position of each opaque pixel relative to the stellar limb at each time step in order to determine how much flux it occludes. We adopt the small-planet approximation of Mandel & Agol (2002), in which it is assumed that the star's surface brightness is constant across a pixel. In other words, we treat the pixel as occulting a thin, uniform-surface-brightness, annular slice of the stellar disk, where the radius of the annulus is the distance from the center of the stellar disk to the center of the pixel and the annulus is just wide enough to encompass the pixel.

We denote the area of this annulus as $A_{\rm annulus}$ and its emitted flux as $F_{\rm annulus}$. As a rule of thumb, this small-planet approximation is only appropriate for $w \lesssim 0.2$ (i.e., $N > 10$), which corresponds roughly to an occulter-to-star ratio-of-radii of 0.1, for a circular occulter of the same area as the pixel. The exact ratio-of-radii at which the small-planet approximation becomes inappropriate depends on the impact parameter of the pixel, the size of the pixel, the limb darkening profile of the star, and the bandpass of the observations, so there is no general exact cutoff.

To calculate the light curve in the limb-darkened case, we must renormalize Equation (21): the fractional flux occulted by an opaque pixel is no longer equal to the fractional area of the stellar disk occluded by the pixel, but rather to:

$$\bar{A}_{i,j}(t_k) = \frac{A_{i,j}(t_k)}{A_{\rm annulus}} \frac{F_{\rm annulus}}{F_\star}, \tag{22}$$

where $F_\star$ is equal to the flux of the entire unocculted limb-darkened star relative to the non–limb-darkened star (which must be calculated given a choice of limb darkening coefficients). We note that this equation reduces to $\bar{A}_{i,j}(t_k) = A_{i,j}(t_k)$ in the case of uniform limb darkening.

The value of the light curve at t_k is then given by:

$$F(t_k) = 1 - \sum_{i=1}^{N} \sum_{j=1}^{M} \tau_{i,j} \bar{A}_{i,j}(t_k). \tag{23}$$

We provide `Python` code to calculate the transit light curve of any grid in the case of uniform, linear, quadratic, or four-parameter nonlinear limb darkening in the software package accompanying this paper, `EightBitTransit`.

4. Fitting: Shadow-imaging a Pixel Grid from a Light Curve

In this section, we describe how we use the forward model described above to solve the inverse problem, shadow imaging. We observe a light curve $\boldsymbol{F}$ made up of discrete flux measurements $F_k \equiv F(t_k)$ over K points in time: what pixelated image generated that light curve?

To illustrate the complexity of this problem, we begin with an order-of-magnitude estimation of the number of arrangements of pixels in a binary-valued shadow image (case (1)). There are 2^{NM} unique permutations of transparent and opaque pixels for an N by M grid and $\mathcal{O}[3^{NM/2}]$ unique light curves that can result (by the flip degeneracy, discussed in Section 2.1). For a 10 by 10 grid then, there are $\mathcal{O}[10^{30}]$ unique permutations of the binary pixel opacities; accounting for the flip degeneracy, if one wished to find the binary pixel opacity arrangement of just the top half of a 10 by 10 grid to best match a particular light curve, one would have to evaluate $\mathcal{O}[10^{24}]$ possibilities.

A full parameter search is therefore not practically feasible. The largest square grid which could be reasonably fully searched is 5 by 5, for which there are 33.6 million full-grid permutations and 1.9 million half-grid permutations (by Equation (1)). We must therefore infer the pixel opacities from the light curve, not attempt to guess them.

To infer a pixel grid from a light curve $\boldsymbol{F}$, we must first select the grid parameters: the dimensions N and M, the velocity v, and the reference time $t_{\rm ref}$. Given these choices, we may calculate the areas of overlap of each grid pixel at each light-curve time step and the corresponding $\bar{A}_{i,j}(t_k)$ for any choice of limb darkening law. All that remains is to solve Equation (23) for the opacities of the grid pixels, $\tau_{i,\ j}$, subject to the constraints of either case (1) ($\tau_{i,\ j} = 0$ or 1) or case (2) ($0 \leqslant \tau_{i,j} \leqslant 1$).

4.1. Mathematical Setup

To be exact, we note that $\boldsymbol{F}$ is a column vector of length K, of which each scalar entry $F_k \equiv F(t_k)$ is given by Equation (23). Let us "unravel" the double sum in Equation (23) by defining a new index l, such that

$$l[i, j] = j + (i - 1)M. \tag{24}$$

Given that i ranges from 1 to N, and j from 1 to M, l ranges from 1 to MN.

We may then rewrite Equation (23) as

$$F_k = 1 - \sum_{l=1}^{L} \tau_l \bar{A}_l(t_k), \tag{25}$$

where we define $L \equiv MN$. If we further define $\bar{A}_{k,l} \equiv \bar{A}_l(t_k)$, then

$$F_k = 1 - \sum_{l=1}^{L} \tau_l \bar{A}_{k,l}. \tag{26}$$

Let us now rewrite $\bar{A}_{k,l}$ in matrix form:

$$\bar{\mathsf{A}} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,L} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K,1} & a_{K,2} & \cdots & a_{K,L} \end{pmatrix}. \tag{27}$$

$\bar{\mathsf{A}}$ is a matrix of shape K by L, where the kth row encodes the state of overlap of the entire pixel grid at time step k and the lth column encodes the overlap state of pixel l across all time steps.

Similarly, we may "unravel" the opacity matrix τ into a column vector $\boldsymbol{\tau}$ of length L:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_L \end{pmatrix}. \tag{28}$$

We may now re-express Equation (23) in matrix form:

$$\boldsymbol{F} = 1 - \bar{\mathsf{A}}\boldsymbol{\tau}, \tag{29}$$

where

$$\boldsymbol{F} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_k \end{pmatrix} \tag{30}$$

and 1 is a column vector of ones, equal in length to $\boldsymbol{F}$.

If we define a vector $\boldsymbol{R} = 1 - \boldsymbol{F}$, we may rearrange this equation to read

$$\bar{\mathsf{A}}\boldsymbol{\tau} = \boldsymbol{R}. \tag{31}$$

If $\bar{\mathsf{A}}$ were invertible, then our work would be done: we could solve Equation (31) directly for the vector of pixel opacities $\boldsymbol{\tau}$. However, because of the flip degeneracy, pixel i, j has the same area of overlap at every time step as pixel $(N + 1 - i), j$, and as a result, $\bar{\mathsf{A}}$ always has repeated columns. By the invertible matrix theorem, a matrix with repeated columns is not invertible.

We may proceed by recognizing that $\bar{\mathsf{A}}$ and $\boldsymbol{\tau}$, given that they describe the entire pixel grid, contain redundant information. We need only solve for the opacities of one half of the pixels (we choose the top half, for convenience). We define a new index

$$L_{\rm half} \begin{cases} \frac{(N-1)M}{2} + M, & N \text{ odd} \\ \frac{NM}{2}, & N \text{ even.} \end{cases} \tag{32}$$

We define a new area-of-overlap matrix $\bar{\mathsf{A}}_{\rm half}$, which represents the left half (columns 1 through $L_{\rm half}$, inclusive) of $\bar{\mathsf{A}}$, and a new opacity vector $\boldsymbol{\tau}_{\rm half}$, which represents the corresponding top half of $\boldsymbol{\tau}$. We may then write:

$$\bar{\mathsf{A}}_{\rm half}\boldsymbol{\tau}_{\rm half} = \boldsymbol{R}. \tag{33}$$

Given that, in general, $K \neq L_{\rm half}$, we may multiply both sides of this equation by $\bar{\mathsf{A}}_{\rm half}^{\rm T}$ to yield

$$\bar{\mathsf{A}}_{\rm half}^{\rm T}\bar{\mathsf{A}}_{\rm half}\boldsymbol{\tau}_{\rm half} = \bar{\mathsf{A}}_{\rm half}^{\rm T}\boldsymbol{R} \tag{34}$$

so that both sides of the equation are column vectors of length $L_{\rm half}$ and $\bar{\mathsf{A}}_{\rm half}^{\rm T}\bar{\mathsf{A}}_{\rm half}$ is a square matrix.

For notational simplicity, let $\mathsf{B} \equiv \bar{\mathsf{A}}_{\rm half}^{\rm T}\bar{\mathsf{A}}_{\rm half}$, and let $\boldsymbol{C} \equiv \bar{\mathsf{A}}_{\rm half}^{\rm T}\boldsymbol{R}$, such that

$$\mathsf{B}\boldsymbol{\tau}_{\rm half} = \boldsymbol{C}. \tag{35}$$

We have therefore reduced our shadow imaging problem to the problem of solving a system of linear equations for the entries of the column vector $\boldsymbol{\tau}_{\rm half}$. These entries, reshaped into the matrix τ, correspond to the opacities of the pixels making up the top half of the grid, which define the image.

In the sections below, we elaborate on the two steps of shadow imaging: first, selecting the grid parameters and second, solving Equation (35) for the pixel opacities subject to our chosen physical constraints.

4.2. Constraining the Grid Parameters

In general, the auxiliary parameters $t_{\rm ref}$ and v can be set to reasonable approximations of their "true" values, and the pixel image will slightly shift and stretch, respectively, relative to the "truth," without disturbance to its principal morphology. This means we can proceed by fixing these terms and optimizing the opacities $\boldsymbol{\tau}$ only. We may then, depending on the success of the solution $\boldsymbol{\tau}$, perform further iterations, varying the grid parameters each time, to reach an optimal grid with optimal auxiliary parameters. We discuss here some constraints of the grid parameters which allow us to estimate their values initially.

The first constraint we consider is that the number of pixel elements should not exceed the number of data points obtained during the transit event of interest. For regularly sampled data, such as those of *Kepler*, we may write the sampling constraint as

$$NM \leqslant \frac{t_{\rm event}}{t_{\rm cadence}}, \tag{36}$$

where $t_{\rm event}$ is the timescale of the event we wish to image and $t_{\rm cadence}$ is the cadence of the time series, i.e., the interval between successive observations.

The second constraint we consider is that a pixel should not be too small to detect individually. In other words, the transit depth of a single opaque pixel should not be smaller than the uncertainties on the flux measurements. In principle, smaller pixels could be resolved over repeated transit observations, but this approximation again aids in selecting a unique initial grid size from which to begin optimizing the grid opacities.

Mathematically, we can express the precision constraint as:

$$\frac{w^2}{\pi} \gtrsim \sigma \tag{37}$$

where σ is the typical photometric uncertainty. Combining Equation (13) with this constraint gives

$$N \lesssim \sqrt{\frac{4}{\pi\sigma}}. \tag{38}$$

For reference, using a 60 ppm uncertainty, this yields $N \lesssim 146$. (In practice, we are usually limited to much smaller values of N by the number of data points in the observed light curve.)

The third constraint we consider is the size of M. Our grid must be wide enough to create a total duration sufficient to explain the event timescale $t_{\rm event}$. We require that $t_{\rm exit} - t_{\rm enter} \geqslant t_{\rm event}$, or

$$\frac{2 + 2(M/N)}{v} \geqslant t_{\rm event}. \tag{39}$$

Similarly, we consider that a single pixel needs to be able to traverse the entire disk of the star within the event timescale. The actual duration of a single pixel's transit will depend on the pixel's latitude Y, but to simplify things, we consider an equatorial pixel of infinitesimal size and use an approximate symbol for the inequality, to give

$$v \gtrsim 2/t_{\rm event}. \tag{40}$$

Together, these expressions constrain the velocity to the range

$$2/t_{\rm event} \lesssim v \leqslant 4/t_{\rm event}. \tag{41}$$

As a general strategy then, we choose a grid velocity v equal to $2/t_{\rm event}$, and $t_{\rm ref}$ to correspond to the minimum of the observed light curve. To choose N and M, we recognize that, for a chosen N, we may solve for M such that the grid continuously overlaps the star, by rearrangement of Equation (20). We can then adjust N to accommodate the constraint that NM should be less than the number of observed data points. Once the grid dimensions have been chosen, we re-execute the inversion for different velocities until the fit ceases to improve.

Because of the resolution constraint, we prefer the slowest grid velocity v which returns a reasonable fit to the observed light curve, because this slow velocity corresponds to the highest image resolution N. This is, in a sense, an image prior which prefers narrow, slow images to their fast, stretch-degenerate counterparts.

4.3. Solving for the Pixel Opacities

Once we have reasonable first estimates for $t_{\rm ref}$, v, N, and M and have chosen a limb darkening law to describe the stellar disk, we may use Equation (22) to solve for $\bar{A}_{i,j}(t_k)$ for each grid pixel at each light-curve time step. At this stage of shadow imaging, it is helpful to think of the grid pixels as containers for as-yet-to-be-determined opacity: each transits the star in a definite way according to the grid parameters, so $\bar{A}_{i,j}(t_k)$ and hence $\bar{\mathsf{A}}$ are well defined, but its opacity is not yet known.

For a chosen $t_{\rm ref}$, v, N, and M, we restrict our attention to the observed light-curve data points that satisfy $t_{\rm enter} < t < t_{\rm exit}$. In other words, we consider only the points in time during which the grid partially overlaps the star, because the transiting grid cannot influence points outside this range.

To determine the opacities, we must solve Equation (35) for the entries of the opacity vector $\boldsymbol{\tau}$. Given that this matrix equation is linear, in principle it can be directly, analytically solved.

However, direct solution of Equation (35) cannot accommodate constraints on the pixel opacities. Namely, there is no way to restrict the entries of $\boldsymbol{\tau}$ to the physically meaningful range [0,1] (case (1)), let alone to the binary values (0 or 1; case (2)). Mathematically, introducing these constraints transforms the problem into a nonlinear optimization problem, which is not susceptible to solution by a linear matrix equation. We furthermore find that transforming the opacity variables through a logistic function, which maps the real numbers to the range [0,1], results in numerical instabilities in our attempts to solve Equation (35) both directly and iteratively (e.g., with the Simultaneous Algebraic Reconstruction Technique [SART]; see Section 4.3.2 below).

Furthermore, we find that choosing grid parameters $t_{\rm ref}$, v, N, and M that deviate even slightly from the true values leads to a completely nonsensical recovered $\boldsymbol{\tau}$. Direct analytic solution is therefore not robust enough to apply to a light curve of unknown origin, where our initial guesses for the grid parameters are unlikely to be so accurate.

We therefore explore less exact but significantly more robust algorithmic approaches to solving for $\boldsymbol{\tau}$. Below, we discuss each of these algorithms in turn. The first two address case (1), where pixels may take on intermediate opacities, and the latter three address the more restrictive case (2), where pixels are constrained to be binary-valued.

In Figures 8 and 9, we compare their performance in recovering a number of known test grids from noiseless light curves. In these recovery tests, the parameters N, M, v, and $t_{\rm ref}$ are assumed to be known. Eight of the test grids are binary-valued, and three (the low-resolution planet–moon, planet-ring, and comet test grids) include intermediate-opacity pixels.

In Figures 8 and 9, we have chosen to generate our test light curves with a uniformly bright star, i.e., without limb darkening. We make this choice because non–limb-darkened light curves are sharper and less rounded than limb-darkened light curves, and the inversions result in correspondingly sharper image grids, among which the differences between the images generated by our four recovery algorithms stand out most clearly.

We find that introducing realistic limb darkening results in recovered images very similar to those shown in Figures 8 and 9, with two notable qualitative differences: First, for the limb-darkened case, opacity tends to be pushed farther out toward the top and bottom edges of the recovered image. This effect is most obvious in the arc combinatorics images. Second, the recovered images appear blurrier, which makes intuitive sense given the more rounded features of a limb-darkened transit event compared to those of a non–limb-darkened transit.

4.3.1. Arc Averaging

The first algorithmic approach we explore relies on the time derivative of Equation (31). At each time step dt, the overlap state of the grid changes; we can express the change in overlap area as the matrix $d\bar{\mathsf{A}}/dt$, calculated at each time step. Most of the entries of this matrix will be equal to 0, because only the pixels overlapping the stellar limb at that time step will have nonzero change in overlap area.

Meanwhile, at each dt, we can calculate the net change in the observed light curve, $d\boldsymbol{R}/dt$. Two effects can contribute to nonzero $d\boldsymbol{R}/dt$ at a particular time step: (i) one or more pixels with nonzero opacity overlapping the stellar limb at that time step and (ii) in the case of non-uniform limb darkening, one or more pixels with nonzero opacity overlapping any part of the star. For the low-resolution grid inversions that are possible given the time resolution of currently available transit data (see, e.g., Sections 5.1 and 5.2), effect (i) is much larger than effect (ii). Additionally, the stellar intensity profile changes most steeply near the limb, so effect (ii) is most prominent for limb pixels anyway.

Therefore, for the "arc-averaged" algorithm, we take the naive approach of calculating the average $d\boldsymbol{R}/dt$ per pixel which overlaps the limb at that time step. Then, we endow each limb pixel with that average opacity, weighted by $\frac{1}{\sin\theta_{\rm pixel}} = \frac{R_*}{b_{\rm pixel}} = \frac{1}{b_{\rm pixel}}$ to mitigate the effects of the arc degeneracy.

We do the above arc averaging independently for each time step dt, then average the results over all time steps to compute the final grid. Finally, we renormalize the pixel grid to match the transit depth of the observed light curve. (Renormalization is necessary because the arc averaging algorithm only exploits information from the derivative of the light curve, not from the light curve itself.)

Arc-averaged pixel solutions, because they exploit the arc degeneracy, exhibit semicircular arc–like features. They are also horizontally symmetrical as a result of the flip degeneracy. Overall, they are smoother and more dispersed than their true pixel grid counterparts, with smoother light curves, because the averaging step precludes sharp, isolated islands of opacity. The

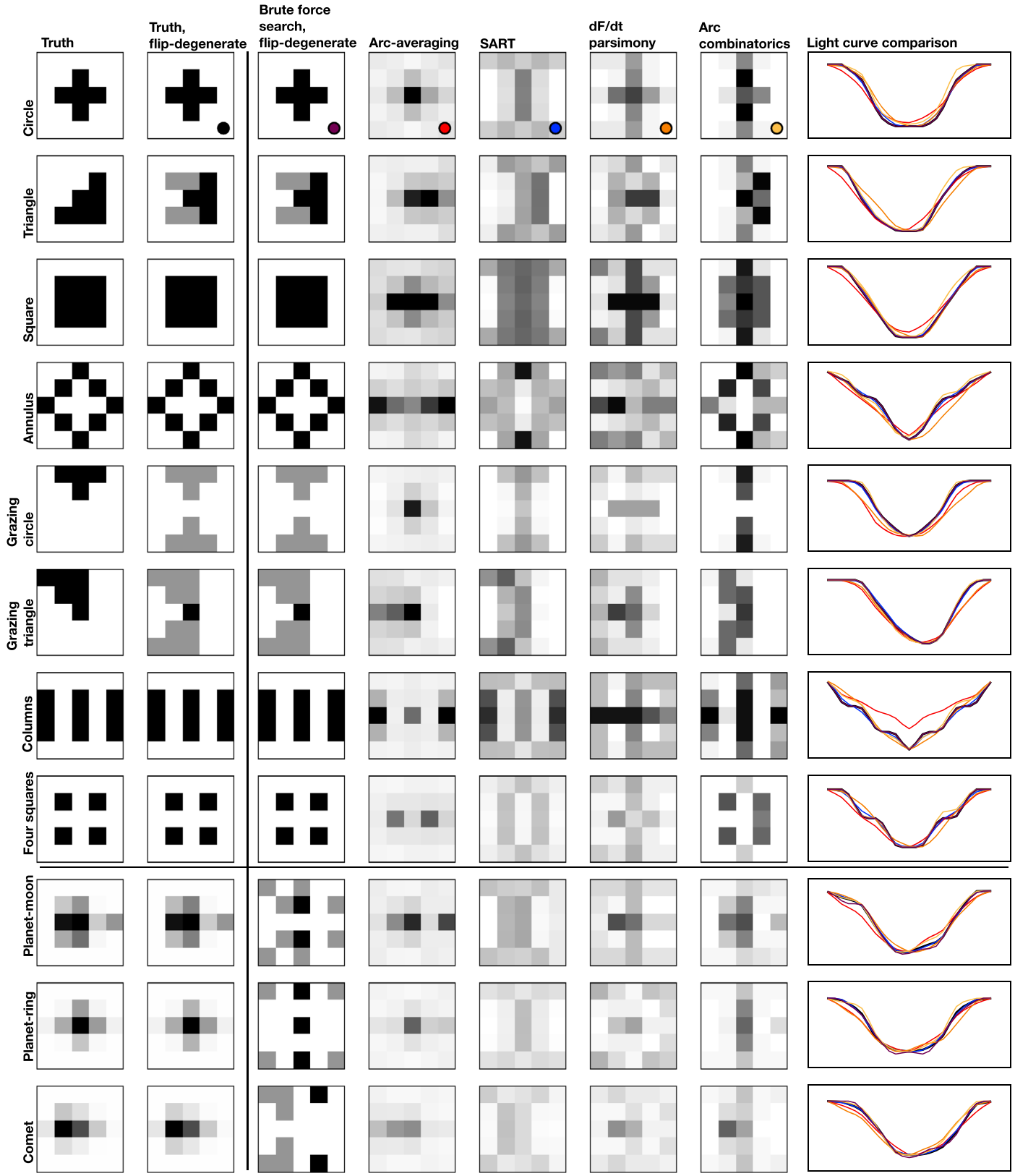

Figure 8. Performance of several light-curve inversion algorithms on 11 known 5 by 5 pixel test grids. The leftmost two columns represent the true input grid; the subsequent columns represent the grid recovered by each inversion algorithm given only the (noiseless) true light curve as input. The eight test grids above the horizontal line are pure binary grids (i.e., pixel opacities are either 0 or 1); the three below have intermediate, semiopaque pixels. Each algorithm is initialized with correct grid parameters N, M, $t_{\rm ref}$, and v, and the light curves are generated with a uniform limb darkening law. The brute-force search algorithm performs the best—i.e., returns the light curve with the lowest rms error compared to the true image's light curve—in every pure binary test case, but SART performs best on the semiopaque test cases.

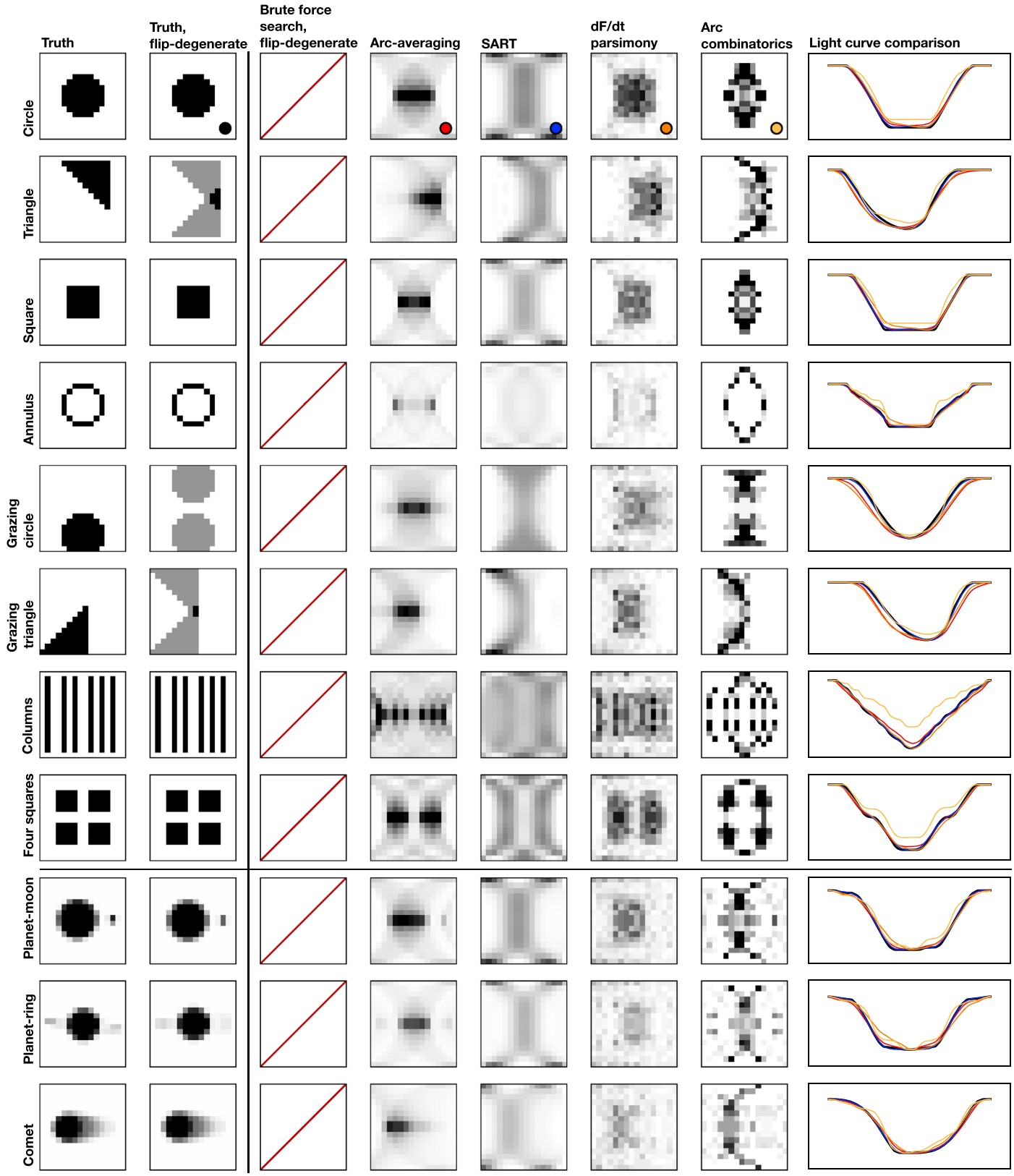

Figure 9. Performance of several light-curve inversion algorithms on 11 known 16 by 16 pixel test grids, which are too large to allow for a brute-force permutation search. The two leftmost columns represent the true input grid; the subsequent columns represent the grid recovered by each inversion algorithm given only the (noiseless) true light curve as input. The eight test grids above the horizontal line are pure binary grids (i.e., pixel opacities are either 0 or 1); the three below have intermediate, semiopaque pixels. Each algorithm is initialized with correct grid parameters N, M, $t_{\rm ref}$, and v, and the light curves are generated with a uniform limb darkening law. SART performs the best—i.e., returns the light curve with the lowest rms error compared to the true image's light curve—in every test case; arc averaging is second best in every case except the offset circle, for which arc combinatorics does better.

impact parameter weighting causes opacity to be concentrated at the midplane of the grid.

As shown in Figures 8 and 9, the light curves of arc-averaged solutions match observed light curves well, particularly for large, centrally concentrated test shapes. The worst matches are for grazing shapes (see, e.g., the 16 by 16 pixel grazing circle), because the $\frac{1}{b_{\rm pixel}}$ weighting pushes opacity strongly toward the grid midplane and away from the top and bottom of the grid. The arc-averaged light curves also tend to be more rounded than the observed light curve, meaning that the arc averaging approach struggles to reproduce sharp light-curve features. This is sensible because, by design, it produces solutions where opacity is distributed continuously along overlapping arcs rather than confined to discrete islands.

We also note that because arc averaging can easily accommodate pixel opacities between 0 and 1, it can be applied to semiopaque pixel grids, like the low-resolution planet–moon, planet-ring, and comet test grids.

4.3.2. SART

The next algorithm we test is called SART (Andersen & Kak 1984). SART was originally developed for medical computed tomographic imaging. Specifically, SART reconstructs a 2D image from projections of X-rays through the body—this is directly analogous to our shadow imaging problem, where "projections" of the pixelated image against the stellar disk are the individual data points in the light curve.

SART operates iteratively upon an initial guess for the opacity vector $\boldsymbol{\tau}_{\rm half}$, which encodes the opacities of the pixels of the top half of the image grid. Beginning from this initial guess, it computes subsequent corrective updates to the individual entries of $\boldsymbol{\tau}_{\rm half}$.

The $(q+1)$th iteration of τ_l, the opacity of pixel l, is given by

$$\tau_l^{q+1} = \tau_l^q + \frac{\sum_{k=1}^{L_{\rm half}} \left(B_{kl} \frac{C_k - \sum_{\lambda=1}^{L_{\rm half}} (B_{k\lambda}\, \tau_\lambda^q)}{\sum_{\lambda=1}^{L_{\rm half}} B_{k\lambda}} \right)}{\sum_{k=1}^{L_{\rm half}} B_{kl}}. \quad (42)$$

The scalar τ_l^q is the lth entry of $\boldsymbol{\tau}_{\rm half}$ at iteration q, representing the opacity of pixel l; the scalar B_{kl} is the kth-row, lth-column entry of B; and the scalar C_k is the kth entry of $\boldsymbol{C}$. B and $\boldsymbol{C}$ are defined in Equation (35).

For intuition, the update term in Equation (42) is equal to the average correction to pixel opacity τ_l over all rows and all columns of B. (Hence, the sum in the denominator is over all rows of column vector $\boldsymbol{B}_l$, and the sum in the numerator term's denominator is over all columns of row vector $\boldsymbol{B}_k$.) The numerator, specifically, is the average value over all pixels in the grid of a sort of "residual" between the observed light curve and the model. This residual is equal to C_k minus the scalar projection of $\boldsymbol{\tau}_{\rm half}^q$ along $\boldsymbol{B}_k$. In effect, these two averages allow for a correction to the opacities, which is averaged over all time steps of the light curve and all pixels in the grid.

By running the SART algorithm for a large number of iterations (usually $\sim 10^4$ for a 16 by 16 pixel grid), we achieve good convergence to the observed light curve for a number of test cases. The rms error between the light curve of the input image and the light curve of the SART solution declines monotonically over the SART iterations, indicating that SART achieves a progressively better fit to the light curve as it proceeds.

We find that starting from an initial guess of all $\tau_l = 0.5$ works well, because the step-by-step updates to $\boldsymbol{\tau}$ are generally small, so the algorithm does not wander far into unphysical parameter space (i.e., $\tau_l < 0$ or $\tau_l > 1$). In the event that the resulting SART solution does have slightly unphysical opacities, we redistribute the excess positive or negative opacity uniformly among the pixels whose centers fall within a distance of $w/2$ of the arc pair that intersects at the unphysical pixel. This redistribution renders the SART solution fully physical without drastically changing its light curve. Because SART exploits information in the light curve, not just its derivative, it is not necessary to renormalize the SART solution's pixel opacities.

SART solutions exhibit horizontal symmetry as a result of the flip degeneracy and semicircular arc–like features as a result of the arc degeneracy. Like the arc averaging algorithm, SART tends to smear out sharp features in the true input image along arcs, resulting in pixel grid solutions which are smoother, with more dispersed opacity than the true image. (SART solutions are even smoother than the corresponding arc-averaged solutions.) As as a result, SART fails, for example, to match the sharply flat-bottomed transits of the 16 by 16 pixel circle and square test grid light curves (Figure 9), producing slightly rounded light-curve shapes instead. On the other hand, because SART allows the pixel opacities to take any continuous value between 0 and 1, it can accurately reproduce the light curves of nonbinary test grids, like the planet–moon, planet-ring, and comet test grids.

4.3.3. Brute-force Search

The next three algorithms we explore attempt to invert light curves subject to the constraint of binary pixel opacities: in other words, we attempt to recover grids with pixel opacities of 0 (completely transparent) or 1 (completely opaque). We begin with the simplest, a brute-force search of every possible arrangement of binary pixel opacities.

As discussed in Section 2.1, by the flip degeneracy, a grid of N by M opaque and transparent pixels can generate $\mathcal{O}[3^{NM/2}]$ unique light curves. Correspondingly, one would have to evaluate $\mathcal{O}[3^{NM/2}]$ permutations of transparent and opaque pixels to find the grid that matches a given light curve best. The largest square grid for which such a full search is feasible is 5 by 5 pixels, which has 1.9 million associated pixel arrangements with unique light curves (for comparison, a 6 by 6 grid has $\sim$390 million).

In Figure 8, we illustrate the results of a brute-force full-grid search for noiseless test light curves generated by a number of 5 by 5 known input grids. The brute-force algorithm returns the pixel arrangement which, when transiting the star, generates the light curve with the lowest rms error compared to the truth.

Unsurprisingly, when the input grid is truly binary, i.e., made up of completely opaque and completely transparent pixels, the full search converges to the best possible solution every time. However, when the input grid includes semi-opaque pixels, as in the low-resolution planet–moon, planet-ring, and comet test cases, the brute-force search struggles; the lowest-rms solution does not necessarily bear any resemblance to the input grid, even though its light curve matches the true light curve well. This is a testament to the complex and multimodal likelihood landscape of the pixel opacities and also an illustration of why conventional

nonlinear optimization methods cannot solve the light-curve inversion problem. (We note here that we have also investigated both a genetic algorithm and a downhill simplex algorithm (Nelder & Mead 1965), but without success—both methods tend to reach local optima and stall, and as illustrated here, locally optimal grids are not necessarily morphologically similar to the true grid.)

Brute-force search solutions are not presented in Figure 9, because these grids are far too large to be exhaustively permuted.

4.3.4. Parsimonious Opacity Assignment

The next two algorithms we test rely, like arc averaging, on the time derivative of Equation (31). However, instead of averaging the ingress or egress opacity over all of the limb pixels at each time step, we attempt to parcel it out in units of 0.5 opacity (to accommodate the flip degeneracy). We note that consequently, these two algorithms do not work well for inverting shallow transits observed with small time sampling (i.e., few light-curve data points), because in such cases, the grid will be low-resolution and the transit depth of a single pixel's worth of opacity can be greater than the observed transit depth. There will then be no good match to the light curve.

First, we explore the "parsimonious" approach, which assigns opacity to as few pixels as possible in order to accommodate the change in the light curve. This algorithm is motivated by compactness—is it possible to match the light curve with as few "on" pixels as possible?

The parsimonious approach assigns opacity first to the pixel with the largest change in overlap area $d\bar{\mathrm{A}}/dt$, then steps through successively "less influential" pixels until the entire change in the light curve has been accounted for. As with the arc averaging approach, it is necessary to average the results over all time steps dt, then renormalize the resulting pixel grid to match the observed transit depth; the pixel grid solutions presented in Figures 8 and 9 therefore have some pixel opacities between 0 and 1.

In practice, this algorithm generates pixel grids which are strongly concentrated at the stellar midplane, because these middle pixels undergo the greatest change in overlap area at fixed dt during their ingress and egress. Correspondingly, it fails to reproduce high-impact-parameter features in the input grids and is especially poor at matching the light curves of grazing shapes, like grazing circles and grazing triangles (Figure 9). Overall, it is the least successful of the four algorithms.

4.3.5. Arc Combinatorics

Finally, we consider an algorithm which attempts to assign units of 0.5 opacity to the best *combination* of limb pixels at every time step in order to match the observed light curve. At every dt, the algorithm calculates the number of "spaces" on the stellar limb, s, that could accommodate a unit of 0.5 opacity. This is equal to twice the number of limb pixels of the appropriate "sign": if the light curve is decreasing at dt, we need only consider the limb pixels which are undergoing ingress and vice versa.

Next, it calculates the number of 0.5 opacity units n that need to be accommodated. This is equal to the change in the light curve, $d\boldsymbol{R}/dt$, divided by the mean overlap area of the limb pixels at that time step, multiplied by 2 (because we wish to distribute opacity in units of 0.5, not 1).

The number of ways to arrange n opacity units over s spaces is then $\binom{s}{n}$. The algorithm explores each combination and chooses the one which matches the vector $d\boldsymbol{R}/dt$ best. Finally, as with arc averaging and the parsimonious approach, the resulting grid is averaged over all time steps and renormalized to match the observed transit depth (so once again, the pixel grid solutions presented in Figures 8 and 9 have some pixel opacities between 0 and 1).

The arc combinatorics approach is able to match certain vertically sharp features in the input images, such as the 16 by 16 pixel annulus and column test cases (Figure 9). It can also accommodate narrow features at high impact parameters; to see this, compare the parsimonious and arc combinatorics solutions to the 16 by 16 pixel four-square test case.

Because of the arc degeneracy, however, the arc combinatorics algorithm tends to prefer solutions where opacity is pushed too far toward the top and bottom edges of the grid (e.g., the 16 by 16 pixel circle and square test grids, Figure 9). (This is the opposite problem of the parsimonious algorithm.) It also struggles to capture the nuances of semiopaque test grids, like the planet–moon, planet-ring, and comet test grids. Finally, we note that the computational cost of this algorithm scales poorly with increasing grid resolution (i.e., increasing s), because the algorithm needs to evaluate $\binom{s}{n}$ opacity arrangement possibilities.

5. Real Data

In this section, we discuss the performance of shadow imaging in two real test cases: first, the light curve of the triple transit of TRAPPIST-1c, 1e, and 1f (Gillon et al. 2017) and second, two unexplained transit-like events observed in KIC 8462852, or Boyajian's Star (Boyajian et al. 2016).

5.1. TRAPPIST-1c/e/f Triple Transit

We begin with the TRAPPIST-1c/e/f triple transit, for which the expected shadow image is known. We hope to recover an image of three transiting planets, analogous to the diagram presented in Gillon et al. (2017, Extended Data Figure 1).

In attempting to invert this light curve, we have useful prior information beyond the expected image. First, because of the repeated transit observations and N-body dynamical simulations presented in Gillon et al. (2017), the periods and eccentricities, respectively, of planets c, e, and f are well constrained. This enables us to calculate the Keplerian orbital velocities of c, e, and f, which we can use as the v of our transiting grid. (We note that because these three orbital velocities are different, the pixel image we are attempting to recover changes during the transit, so we will only be able to recover an approximate image for any single choice of v.)

Second, because the physical behavior of this system is so well understood and the other properties of these planets (R_{p}/R_{*}, b, a/R_{*}) are so well constrained by transit modeling, we can generate an extremely finely time-sampled model light curve, based on a BATMAN model (Kreidberg 2015), of this triple transit, which matches the observed light curve. We can use this high-resolution light curve to test the effects of grid resolution on the success of shadow imaging: when the light

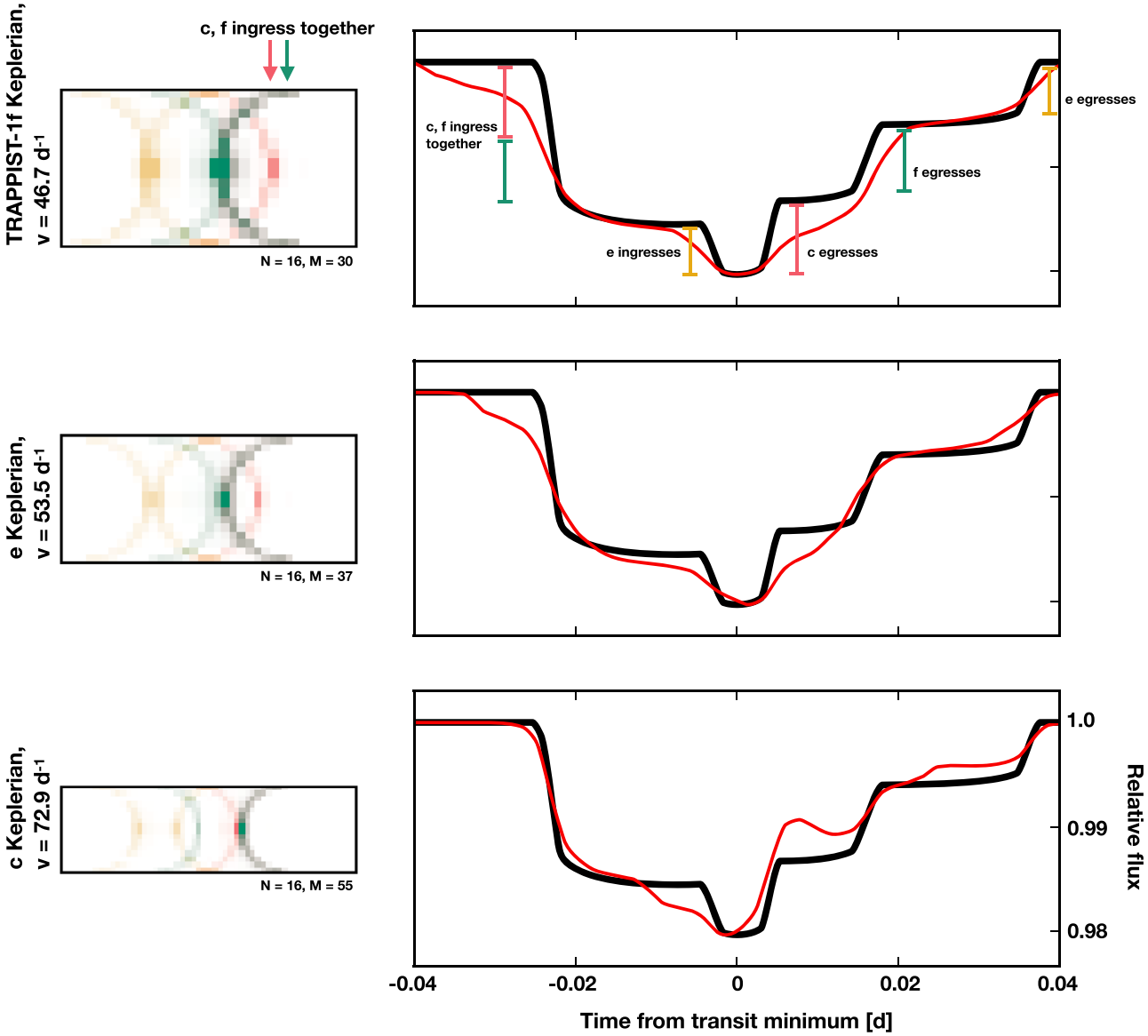

Figure 10. Three inversions of a BATMAN-modeled high-resolution TRAPPIST-1c/e/f triple transit light curve, conducted with the arc averaging algorithm, with the grid v equal to the Keplerian velocity of planets c (bottom), e (middle), and f (top). These images transit the star moving left to right, so the features at the right-hand side of the image influence the light curve first. The BATMAN-modeled light curve (black) and arc-averaged shadow image light curve (blue) are compared in the right-hand panels. We have added color to the shadow images to indicate the positions of planets c (pink), e (yellow), and f (green). (Note that c and f ingress together, so their ingress arc is green + pink = gray.)

curve is finely sampled, we can recover a much higher-resolution grid than when the light curve is sparsely sampled. Finally, we can adopt the same approach to determining the quadratic limb darkening coefficients for TRAPPIST-1 as Gillon et al. (2017) did in their analysis, interpolating TRAPPIST-1's stellar properties from the tables of Claret et al. (2012).

In Figure 10, we present three inversions of the BATMAN-modeled high-resolution TRAPPIST-1c/e/f triple transit light curve, conducted with the grid v equal to the Keplerian velocity of planets c, e, and f, respectively. We choose $N = 16$ because it is a high enough resolution that pixel width $w \lessapprox R_p/R_*$ for planet e, the smallest planet ($(R_p/R_*)^2 = 0.52$, according to the transit modeling of Gillon et al. 2017). We show the results of the arc averaging algorithm here, because it produces the cleanest and most interpretable shadow images, although results from the other three algorithms are qualitatively similar.

In the shadow images, which transit the star moving left to right (i.e., the pixels at the right-hand edge of the image transit first), clear ingress and egress arcs for each planet are visible, in the expected order: first, planets c and f ingress together; then, e ingresses; c egresses; f egresses; and e egresses.

The three planets move at three different velocities to produce the light curve, but the grid moves as a unit, so none of the three inversions perfectly matches the light-curve model. When the velocity is correct for a particular planet, that planet's image is a pair of arcs whose points of intersection fall at the planet's impact parameter as measured by Gillon et al. (2017), demonstrating that shadow imaging of that planet is successful within the constraints of the arc and flip degeneracies.

When the grid v is slower than the planet's velocity, the planet's ingress and egress arcs are spaced too closely together; this effect is most visible for planet c in the top panel, where the grid moved at planet f's velocity. In the light curve, the overlapping arcs manifest themselves in a too early dip, caused by c's *egress* arc entering too quickly, and in a too deep transit depth between the egresses of c and f, caused by c's *ingress* arc remaining in front of the star for too long.

Conversely, when the grid velocity is faster than the planet's, the planet's arcs are too widely separated; this effect is most visible for planet f in the bottom panel. This time, the light curve is too shallow between the egresses of planets c and f, because f's *ingress* arc egresses too soon.

We next investigate what happens if we attempt to invert the observed Gillon et al. (2017) light curve of this triple transit, which is noisy and much more coarsely time-sampled, rather than a high-resolution BATMAN-modeled light curve. Additionally, we ask what happens if we attempt to recover a shadow image without knowing the true velocity of the transiting object: what happens if we use the guidelines presented in Section 4.2 instead?

We invert the observed TRAPPIST-1 triple transit light curve at a range of velocities: the slowest is 31.9 day^{-1}, corresponding to 2 divided by the entire triple-transit event duration (in accordance with the guidelines presented in Section 4.2), and the fastest is 135.9 day^{-1}, corresponding to 4 divided by the duration of planet c's transit by itself. At each velocity, we choose the maximum grid resolution N that, when combined with v to solve for M, allows the transiting grid to partially overlap the star at all time steps of the light curve while still maintaining NM at less than the number of observed data points. Accordingly, the resolution N decreases as v increases, because M increases with v to maintain full light-curve coverage.

In Figure 11, we present the results of these inversions. There are a number of interesting features about these results. We note, first of all, that SART is consistently the most successful inversion algorithm—this is true across the range of tested grid velocities. Furthermore, the SART shadow image consistently resembles the expected shadow image illustrated in Figure 10, even at low image resolutions. Arc combinatorics is somewhat successful at matching the observed light curve at the slowest tested velocity (corresponding to the highest grid resolution) but fails otherwise.

The other algorithms fail consistently across the range of tested velocities. For arc parsimony and arc combinatorics, this results because these algorithms assign binary opacities (0 or 1) to individual pixels, rather than assigning continuous opacities. (We note that the shadow images presented in Figure 11 do not have binary opacities because the final step of both the arc parsimony and arc combinatorics algorithms is to average the binary shadow images produced at each time step dt and renormalize the average to match the observed transit depth.)

When the pixel resolution of the grid is too low, a single pixel's transit depth can exceed the transit depth of a shallow event like the TRAPPIST-1c/e/f triple transit (maximum transit depth $\sim$2%). As a result, the smallest unit of opacity that the arc parsimony or arc combinatorics algorithm can assign is too deep, and these algorithms cannot reproduce the observed light curve. Instead, they tend to assign opacity to pixels along the top and bottom of the image grid, which have the smallest impact on the light curve. This is especially visible in the high-v arc combinatorics panels in Figure 11.

Meanwhile, the arc averaging algorithm also fails to match the observed light curve, regardless of velocity. This is because the arc averaging algorithm, unlike SART, is not robust to noise in the light curve; noise is tantamount to light-curve fluctuations at much higher frequency than can be accommodated by the grid velocity. While SART is able to average out high-frequency noise over many corrective iterations, arc averaging calculates only one arc arrangement per time step dt; if these arrangements are wildly different for neighboring time steps, as they will be for noisy light curves, arc averaging fails.

From these investigations, we conclude that SART is the most robust light-curve imaging algorithm. In particular, light curves with large measurement uncertainties and/or shallow transit depths should only be inverted with SART.

5.2. Boyajian's Star

Next, we proceed to a light curve with an unknown generative shadow image: that of KIC 8462852, Boyajian's Star (Boyajian et al. 2016). This star exhibits aperiodic, deep transit events of unknown origin; hypotheses to explain these events include a family of transiting comets (Bodman & Quillen 2016; Boyajian et al. 2016); circumstellar rings (Katz 2017); an intervening occulter not orbiting Boyajian's Star directly, such as a structure in the interstellar medium or an object with a dusty disk (Wright & Sigurdsson 2016); circumstellar debris following the star's earlier engulfment of a planet (Metzger et al. 2017); and alien megastructures orbiting the star (Wright et al. 2016).

We focus on the two deepest dimming events observed in the Boyajian's Star light curve during the *Kepler* mission, which Boyajian et al. (2016) named Dip 5 and Dip 8. Because these events are aperiodic, the appropriate grid velocity v is not obvious; furthermore, in both dips, the light curve smoothly tapers to a sharp point, so the "beginning" and "ending" points of the event are not obvious. Correspondingly, we start from a velocity $v = 2/t_{\text{event, max}}$, where $t_{\text{event, max}}$ is a wide window around the deepest part of the transit, outside of which the flux of the star has essentially returned to 1. (These time ranges are plotted in Figures 12 and 13.) We then test several other velocities doubled from this starting point. We interpolate quadratic limb darkening coefficients for Boyajian's Star from Sing (2010).

The inverted images for Dips 5 and 8 are presented in Figures 12 and 13, respectively. There are a number of interesting features in these images.

First, there is a circular ring–like feature, of the same radius as the star, that appears generally in images from all four algorithms when the grid velocity v is too slow. (See Dip 5, $v \leqslant 1.6$ days, and Dip 8, $v \leqslant 1.2$ days, for examples.) This happens because when the grid velocity is too slow, the grid struggles to produce narrow features in the light curve; the rate of change of the state of the grid overlap is simply too slow. Under this constraint, the circular ring is the grid pattern that matches a narrow light-curve feature best, in the sense that it generates the narrowest possible V-shaped light-curve feature.

For intuition, consider a copy of the circular ring with the addition of some opaque interior pixels: at some time steps, these interior pixels will be entirely contained within the stellar disk, and their effect on the light curve during these time steps will be constant. In other words, their transit will be

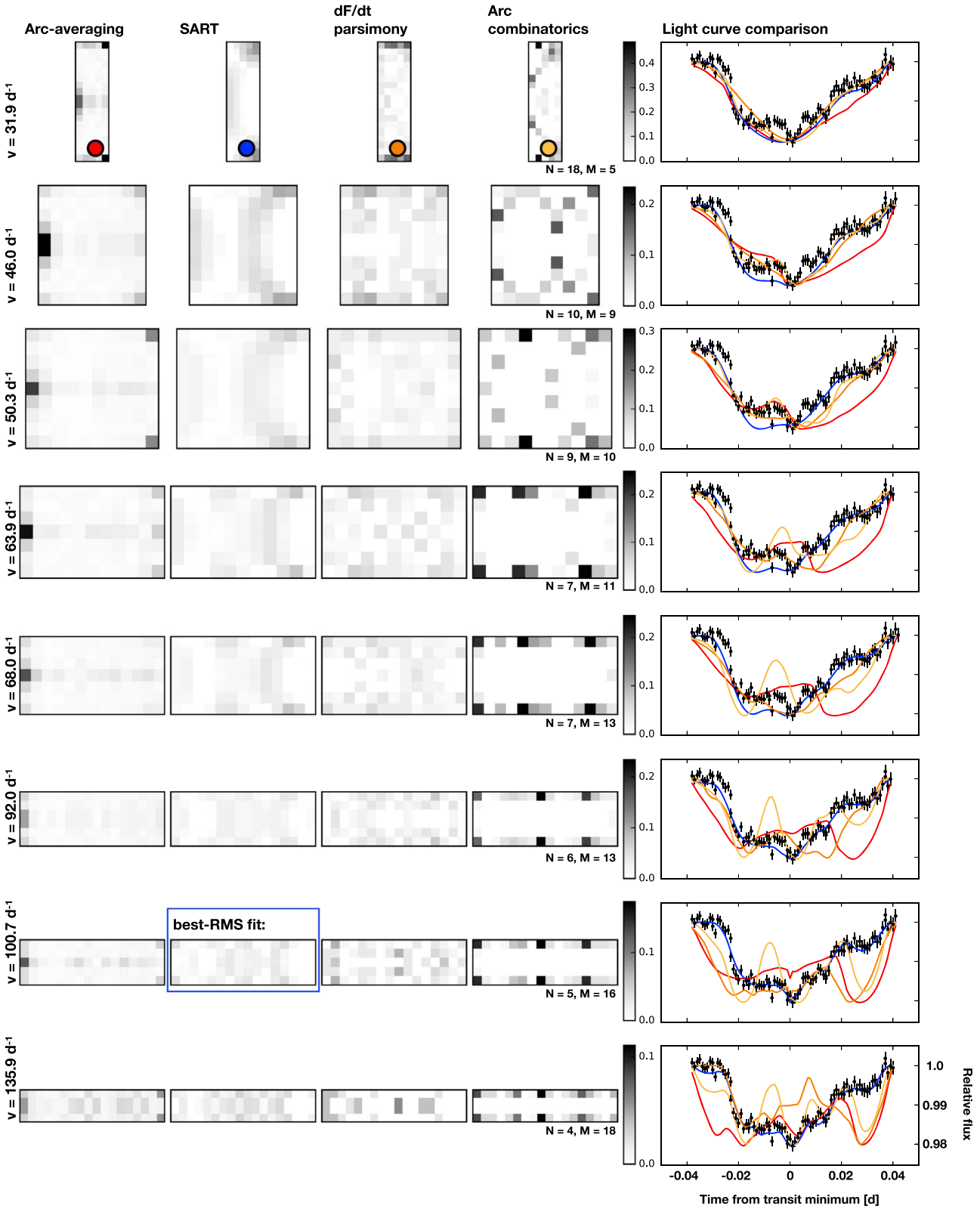

Figure 11. Performance of several light-curve inversion algorithms on the observed TRAPPIST-1c/e/f triple transit light curve. The test velocities and corresponding grid resolutions are chosen according to the guidelines set out in Section 4.2. The shadow image whose light curve has the lowest rms error compared to the observed light curve is the SART inversion at $v = 100.7$ day^{-1}, marked by the blue box. Arc combinatorics performs best, by rms, at the two lowest tested velocities, but SART performs best at all the others.

flat-bottomed. This is not the case for the ring, whose overlap state changes at every time step of the transit; the ring is the "opposite" of a flat-bottomed semicircular arc pair in this sense.

Velocities which produce a ringed shadow image are therefore too slow. This is also obvious from the light curve of the shadow image, which is wider than the observed transit event.

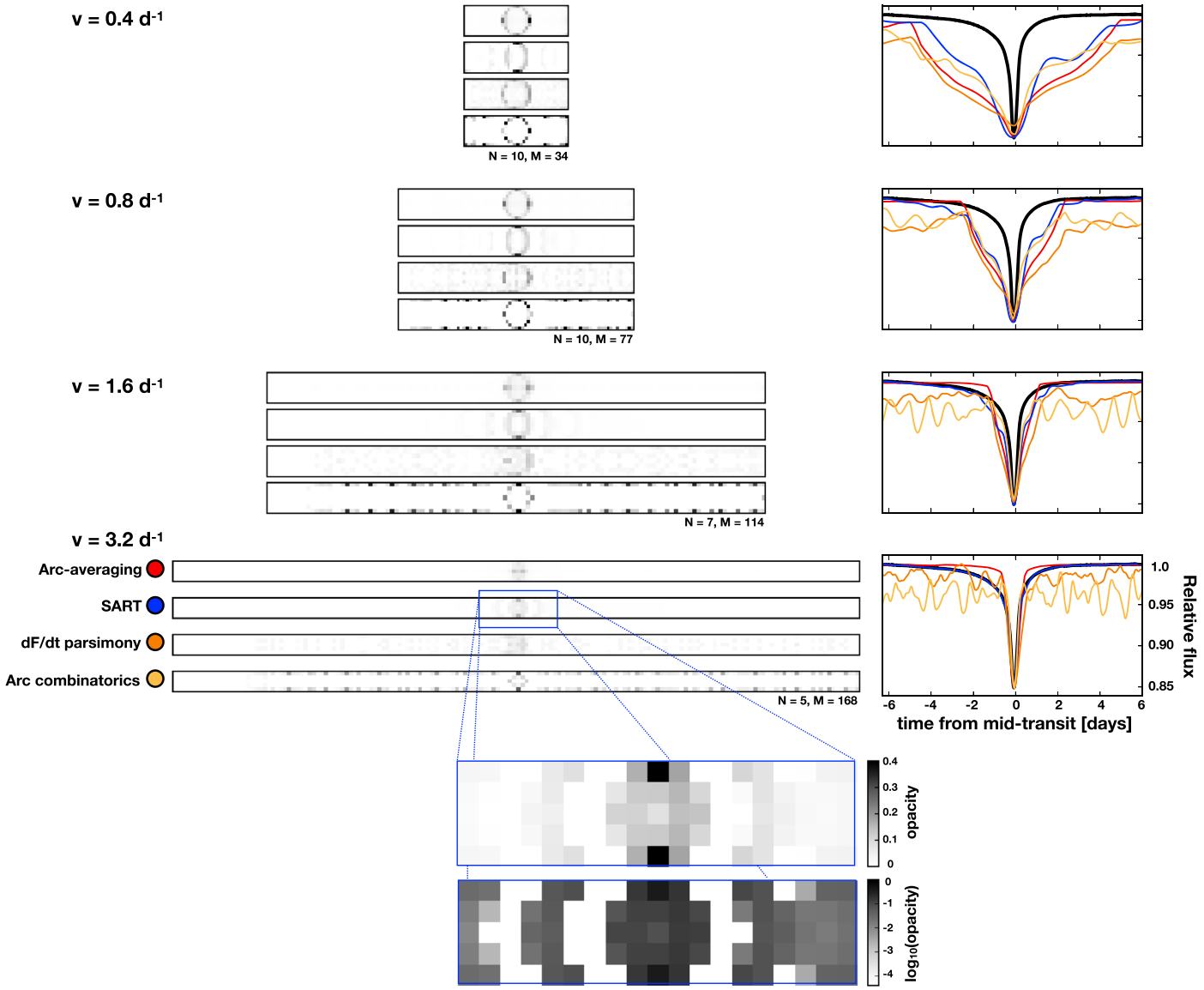

Figure 12. Performance of several light-curve inversion algorithms on Dip 5 of Boyajian's Star. Inset: a zoomed-in view of the central SART shadow image, with both linear and logarithmic color scaling to represent opacity. SART performs best, by rms, at all four choices of v.

When v is fast enough, we find that all four algorithms produce qualitatively similar shadow images for both Dip 5 and Dip 8 and furthermore that the shadow images of the two dips are similar to each other. As in the case of the TRAPPIST-1 triple transit, the arc parsimony and arc combinatorics algorithms generate "noisy" shadow images where the light curve is shallow, because they cannot assign opacity in units smaller than one fully opaque pixel. For Dip 5, SART is clearly the best match to the shadow image; arc averaging produces a light curve which is too narrow, likely because the near-transparent pixels farther from the center which would have produced the "wings" of the transit event have been averaged away in the last combination-and-normalization step of the algorithm. Meanwhile, for Dip 8, the shadow images from both arc averaging and SART match the light curve well.

We strongly caution that there is no straightforward way to interpret these images, for two reasons. First, these images are subject to both the flip and arc degeneracies; second, the grid resolution is low ($N = 5$ for Dip 5; $N = 6$ for Dip 8) because we are limited by the 30 minute cadence of *Kepler* observations, and technically this resolution is so low that the small-planet limb darkening approximation used to calculate the light curves of these shadow images is inappropriate. Nevertheless, these limitations should only affect the distribution of opacity among the semiopaque pixels in the shadow images of Figures 12 and 13; pixels which are fully transparent in the shadow images should remain so, even if we were to obtain a much higher-resolution time series of these events.

We therefore note that the "gaps" of near-zero opacity (i.e., nearly transparent regions) which symmetrically frame the opaque transiting blob at the center of the shadow images in Figures 12 and 13 suggest that structured occulters are responsible for Dips 5 and 8 of the light curve of Boyajian's Star. The gap structure appears to be necessary to produce a shadow image which matches the ingress and egress shape of both Dip 5 and Dip 8; we note, for example, that this gap structure is missing in the arc-averaged shadow image of Dip 5 at $v = 3.2\text{day}^{-1}$, and the ingress and egress features of the corresponding light curve are too sharp to match the observed light curve.

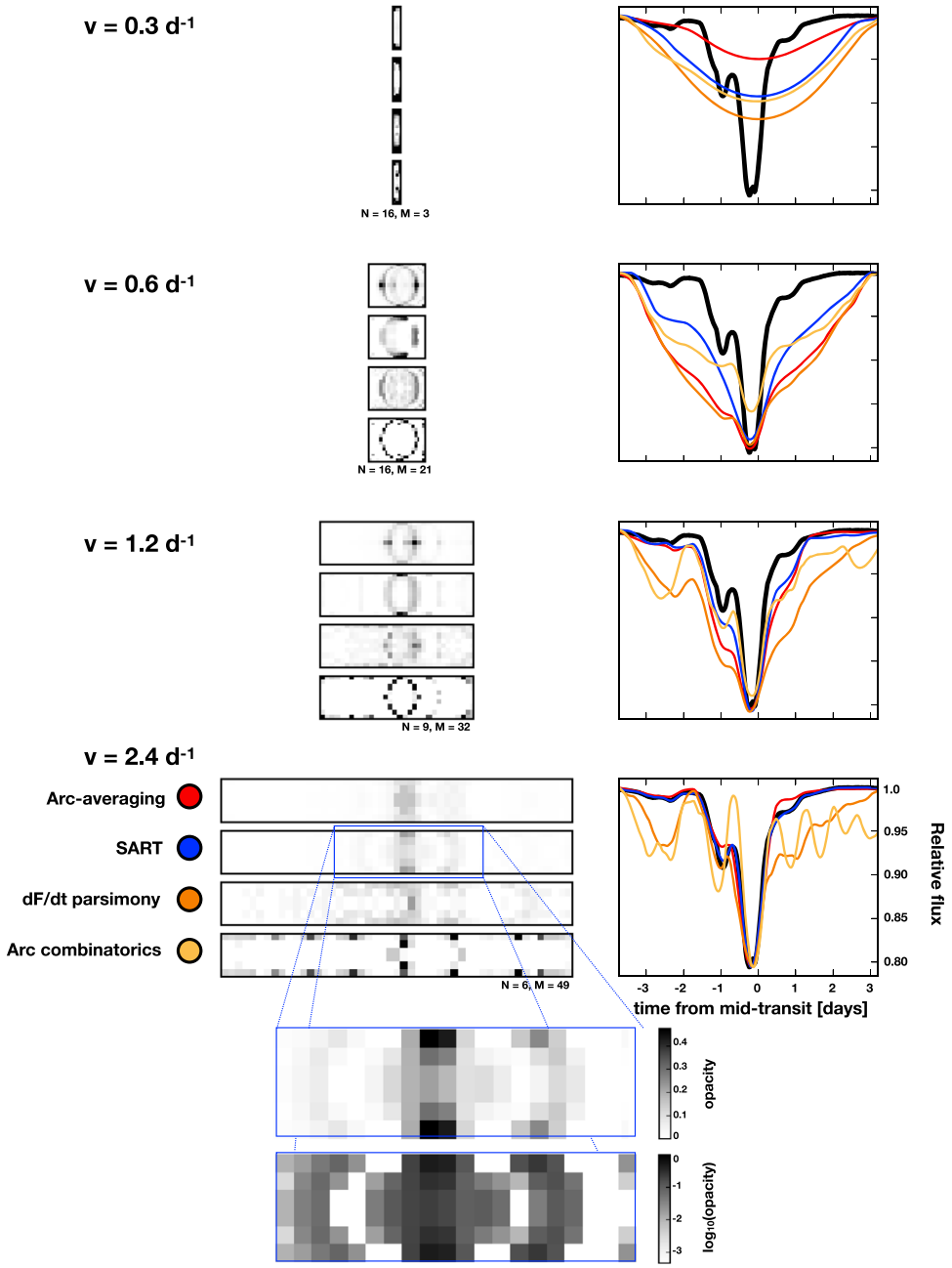

Figure 13. Performance of several light-curve inversion algorithms on Dip 8 of Boyajian's Star. Inset: a zoomed-in view of the central SART shadow image, with both linear and logarithmic color scaling to represent opacity. SART performs best, by rms, at all four choices of v.

6. Conclusions

Here, we have developed a mathematical and computational framework to address the problem of shadow imaging, or inferring the shape of a transiting object from its light curve alone. We find that this problem, which amounts to reconstructing a two-dimensional map from a one-dimensional time series, is degenerate, like the analogous problems of eclipse mapping and starspot inversion. In particular, by the flip degeneracy, shadow images are horizontally symmetrical; by the arc degeneracy, any infinitesimal opaque point in a shadow image can be replaced by a pair of intersecting semicircular arcs without consequence to the light curve; and by the stretch degeneracy, a wide image transiting at high velocity can produce the same light curve as a narrow image transiting slowly, given high enough pixel resolution.

In spite of these degeneracies, we are able to recover informative shadow images by adopting additional assumptions in algorithmic approaches to inverting light curves. We investigate four algorithms with different underlying assumptions. The first is arc averaging, which assumes that opacity should be distributed along arcs in inverse proportion to the $\sin\theta$ opacity distribution characteristic of the arc degeneracy. The second is SART, an iterative approach which assumes that opacity should be distributed so as to minimize the rms averaged over all time steps of the light curve and all pixels in the grid. The third is arc parsimony, which assumes that opacity should be distributed to as few individual opaque pixels

as possible. The fourth is arc combinatorics, which assumes that opacity should be assigned to the best combination of individual opaque pixels to match the light curve. More broadly, the first two algorithms require only that the grid opacities be physical (i.e., restricted to the range [0,1]), while the latter two algorithms operate under the more restrictive assumption that the grid pixel opacities ought to be binary-valued. The less restrictive case can accommodate pixel images of dusty, translucent, or solid objects smaller than the pixel scale, while the more restrictive case is in principle more appropriate for recovering an image of a solid body which is larger than the pixel scale.

Overall, we conclude that SART is the approach which is most robust to our choices of grid resolution and velocity, most robust to noise in the observed light curve, and best able to accommodate shallow transit events. The only downside of SART is that, because it is an iterative optimization method, it is not parallelizable. For grids of the size investigated here ($N \leqslant 16$), it is of perfectly manageable computational cost.

We evaluate the performance of the four algorithms in a number of test cases and find that we can recover informative shadow images for both binary- and continuous-valued opacity grids. We also apply them to real transit events—first, the triple transit of TRAPPIST-1c, 1e, and 1f, for which the true shadow image is known. We recover a shadow image of TRAPPIST-1c, 1e, and 1f which matches our expectations, subject to the constraint that our model grid transits the star at a single velocity, while the real TRAPPIST-1 planets all move individually.

We also apply our techniques to two of the dips observed in Boyajian's Star, for which the true shadow image is unknown. We recover images which are self-consistent in the sense that the results from all four algorithms are qualitatively similar; also, the shadow images of Dip 5 and Dip 7 resemble each other. Transparent gaps in the shadow images of both events suggest that both dips were caused by structured occulters. However, we caution that these shadow images are difficult to interpret: they are subject to both the flip and arc degeneracies, and they are limited in resolution by the cadence of the original *Kepler* observations. In the future, for successful shadow imaging of events like these, large time sampling of the light curve is key.

An important next step in shadow imaging will be to expand the framework presented here to encompass a true *inference* of shadow images: in other words, to recover, given a transit light curve, a distribution of images which could have generated it, complete with uncertainties on the pixel opacities. Such a distribution would meaningfully represent the full set of degenerate solutions that could generate a particular observed set of uncertain flux measurements in a way that a single image cannot.

Accounting for measurement uncertainties is certainly possible within the work presented here; one could, for example, draw repeated "realizations" of a particular light curve given the uncertainties on the individual flux measurements, then invert each realization to recover a single shadow image. The deeper question is how to take what is currently a deterministic retrieval procedure—one light curve, inverted with any of our algorithms, yields exactly one reproducible shadow image—and build in a way to account for the physical degeneracies of the problem, particularly the arc degeneracy, such that one light curve can generate an ensemble of possible shadow images.

In principle, one could also attempt to engineer such an ensemble from a single shadow image by perturbing opacity along arcs. We find that in practice, because of the complex overlapping pattern of the ingress and egress arcs, it is very difficult to perturb opacities and maintain a good fit to the observed light curve. In other words, the arc structure renders the pixel opacities strongly and nontrivially correlated. It remains nevertheless an interesting avenue for future work.

To accompany this work, we present the software package `EightBitTransit`, implemented in `Python`, which is able to calculate the light curves of arbitrary pixel arrangements and to recover shadow images from an input light curve, given the user's choice of grid parameters and inversion algorithm. This software package is available at https://github.com/esandford/EightBitTransit.

The authors thank the referee for a thorough and thoughtful review and members of the Cool Worlds lab for useful discussions. E.S. thanks Zephyr Penoyre for his help building the mathematical formalism of the arc degeneracy and for many conversations throughout the project. E.S. thanks Moiya McTier for test-driving the `EightBitTransit` installation instructions.

ORCID iDs

Emily Sandford https://orcid.org/0000-0003-0822-1839
David Kipping https://orcid.org/0000-0002-4365-7366

References

Andersen, A., & Kak, A. 1984, UltIm, 6, 81
Berdyugina, S. V., & Kuhn, J. R. 2017, arXiv:1711.00185
Bodman, E. H. L., & Quillen, A. 2016, ApJL, 819, L34
Boyajian, T. S., LaCourse, D. M., Rappaport, S. A., et al. 2016, MNRAS, 457, 3988
Claret, A., Hauschildt, P. H., & Witte, S. 2012, A&A, 546, A14
Deeg, H. 2009, in IAU Symp. 253, Transiting Planets (Cambridge: Cambridge Univ. Press), 388
de Wit, J., Gillon, M., Demory, B. O., & Seager, S. 2012, A&A, 548, A128
Farr, B., Farr, W. M., Cowan, N. B., Haggard, H. M., & Robinson, T. 2018, AJ, 156, 146
Gillon, M., Triaud, A. H. M. J., Demory, B.-O., et al. 2017, Natur, 542, 456
Goncharskii, A. V., Stepanov, V. V., Khokhlova, V. L., & Yagola, A. G. 1982, SvA, 26, 690
Juvan, I. G., Lendl, M., Cubillos, P. E., et al. 2018, A&A, 610, A15
Katz, J. I. 2017, MNRAS, 471, 3680
Kawahara, H., & Fujii, Y. 2011, ApJL, 739, L62
Kipping, D. M. 2011, MNRAS, 416, 689
Kreidberg, L. 2015, PASP, 127, 1161
Lanza, A. F., Catalano, S., Cutispoto, G., Pagano, I., & Rodono, M. 1998, A&A, 332, 541
Majeau, C., Agol, E., & Cowan, N. B. 2012, ApJL, 747, L20
Mandel, K., & Agol, E. 2002, ApJL, 580, L171
Metzger, B. D., Shen, K. J., & Stone, N. 2017, MNRAS, 468, 4399
Nelder, J. A., & Mead, R. 1965, CompJ, 7, 308
Piskunov, N. E., Tuominen, I., & Vilhu, O. 1990, A&A, 230, 363
Sing, D. K. 2010, A&A, 510, A21
Vogt, S. S., & Penrod, G. D. 1983, PASP, 95, 565
Vogt, S. S., Penrod, G. D., & Hatzes, A. P. 1987, ApJ, 321, 496
Wright, J. T., Cartier, K. M. S., Zhao, M., Jontof-Hutter, D., & Ford, E. B. 2016, ApJ, 816, 17
Wright, J. T., & Sigurdsson, S. 2016, ApJL, 829, L3