

SDSS Image Processing I: The Deblender

Robert Lupton¹

¹*Princeton University Observatory, Princeton, NJ 08544, U.S.A.*

ABSTRACT

I present a new image deblending algorithm which is being applied to the Sloan Digital Sky Survey (SDSS) imaging data; at high Galactic latitude, approximately 25% of objects between 17th and 23rd are deblended children. No assumptions are made about the properties of the ‘child’ objects (although a symmetry ansatz is used to extract them); it is able to handle data where more than one band is available; it conserves flux; it runs robustly with no human intervention; and it is reasonably fast.

1. Introduction

Gallia est omnis divisa in partes tres.

Caesar; De Bello Gallico

Disentangling overlapping objects in images of the sky is a classical problem with a history that is probably almost as long as that of astronomical image processing itself. Whenever accurate measurements are needed of anything more complex than isolated objects, neighbouring objects must be accounted for. This is for example necessary when constructing a colour-magnitude diagram of the inner part of a globular cluster or looking for the signature of gravitational microlensing against our Galactic bar, but it is also needed when doing surveys of galaxies (where, if not allowed for, foreground stars lead an overestimate of a galaxy’s flux, a bias which is a function of Galactic latitude) or estimating the luminosity function of rich clusters of galaxies.

The special case of overlapping stars (‘crowded field photometry’) has been addressed by a number of authors with considerable success (Tody (1981); Buonanno et al. (1983); Stetson (1987); Lupton and Gunn (1986); and Schechter et al. (1993)), but a rigorous solution in the general case is unfortunately impossible. A CCD image that looks like the superposition of five galaxies, e.g. Stephan’s Quintet (Stephan 1877) may well be just that, but without extra information (e.g. redshifts) we cannot be sure that it isn’t simply a messy blobby irregular galaxy that happens to have five peaks — or even a large elliptical galaxy that’s being viewed through a particularly perverse dust cloud.

Despite this difficulty, a number of authors have written codes to disentangle images that look like superposed galaxies and stars. Almost all crowded field photometry packages and these general *deblenders* proceed by making assumptions about the objects being deblended. For example,

crowded field codes assume that everything in the image is some multiple of the local point spread function (PSF).

In this paper I shall refer to the initial blended image as the *parent* and to the deblended components as *children*.

The codes *Cosmos* (Beard et al. 1990) and *SExtractor* (Bertin and Arnouts 1996) both find isophote levels which just split objects into two (or more) children; this splitting is repeated until no further segmentation occurs. They then fit a bivariate Gaussian (i.e. four free parameters in addition to the centre: an intensity, a major and minor axis, and a position angle) to the profile of each child. Pixels with intensities above the critical intensity which just separates the two children are simply assigned to the corresponding component, while pixels below this level are assigned to the child whose Gaussian model has the largest value at that point. A general discussion of this approach is given by Irwin (1985).

FOCAS (Jarvin and Tyson (1981); Valdes (1982); also *SKICAT* Weir et al. (1995)) uses a similar algorithm to define components, but divides the parent's flux between them based on the flux measured in each component above the splitting threshold. Some *FOCAS* users (Tyson 2000), prefer to use a model-fitting approach to apportion the flux between the components (Tyson 1998).

In a study of nearby rich clusters, Lauer (1986, 1988) successfully decomposed a number of clusters into a central galaxy and a number of cluster members by assuming that all the isophotes of each galaxy were self-similar ellipses (in fact a focus of his study was looking for cases for which this assumption was violated).

Doi et al. (1995) exploit the fact that most galaxies have an at least approximate center of symmetry to separate two or three overlapping objects; White and Keel (1992) use a similar argument to estimate the optical depth of dust in the disk of a spiral galaxy superimposed on an elliptical.

Rather than attempting to separate a blended image into its components, Ratnatunga and Newell (1983) and Dressler and Gunn (1992) use statistical techniques (e.g. using a median rather than a mean profile) to reject the contamination of a primary object by smaller secondary ones, for example stars superimposed on galaxies.

This paper is one of a series describing the Sloan Digital Sky Survey (York et al. 2000) image processing pipeline Lupton et al. (2001, 2005). The SDSS imaging data is taken nearly synchronously in five bands, and is expected to cover about πsr (about 10 Tby of raw images). The imminent arrival of enormous quantities of high-quality multi-band CCD data, coupled with the exponential growth of computing power, prompted the development of a new deblender.

The *Photo* algorithm makes use of the positions of peaks in the blended image, assumptions about the constituent objects' symmetry, and a very simple statistical procedure to deblend arbitrarily complex blends of stars and galaxies. It is able to handle data where more than one band is available (in the case of the SDSS, five bands); no assumptions are made about the properties

of the ‘child’ objects (although an symmetry ansatz is used to extract them); it conserves flux; it runs robustly with no human intervention; and it is reasonably fast.

The next section, 2, presents the basic algorithm in an idealised one-dimensional case. Section 3 applies this to 2-dimensional multi-band data, and shows some examples from the SDSS. Section 4 attempts to ask how well the algorithm performs, based on an analysis of SDSS data. Although the basic idea is very simple, there are a myriad of complexities that face an implementor and some of them are discussed in section 5.

2. The Basic Algorithm

Let us start by considering a 1-dimensional problem with data in only a single band, for example the object in the lower left panel of figure 1. Practical difficulties, for example how to find all the peaks in an object or how to handle unassigned flux, are discussed in section 5.

The procedure is simple: first find all the peaks in the blend; in this case there are three. These peaks define the children. Second, for each child construct a *template* T_r by comparing the intensities of pairs of pixels symmetrically placed about the peak, and replacing both by the *lower* of the two. The templates for my toy problem are shown in the lower right panel of figure 1. Given these templates we may write

$$I_i \approx \sum_{r=1}^n w_r T_{r,i}; \quad (1)$$

$$E \equiv \sum_i (I_i - \sum_{r=1}^n w_r T_{r,i})^2 \quad (2)$$

where I is the blended image, i runs over the pixels in the image, and r runs over the n children. Third, solve for the weights w_r by mimimising E . Fourth, for every pixel in the image assign flux to the r^{th} child C_r as

$$C_{r,i} = I_i \frac{w_r T_{r,i}}{\sum_{r=1}^n w_r T_{r,i}} \quad (3)$$

The children are are shown in the upper left panel of figure 1, and the differences between the truth and these estimates are shown in the upper right; note that (in contrast with the T_r) these children display no particular symmetry.

Having estimated the pixel intensities for each child, I then proceed to measure their properties exactly as if they were isolated objects. For the case presented above, the errors in their total fluxes are 1.6% for ‘galaxy1’, -6% for ‘galaxy2’, and 3% for ‘star’.

This approach has similarities with many of the approaches discussed in the introduction. Defining children by their peaks is central to all general deblenders which I am aware of (although it is not essential; for example if all blends were known to be binary stars a brute force fit of two PSFs wouldn’t require that the stars satisfied a ‘Rayleigh’ criterion and had distinct peaks); the construction of templates by choosing the minimum of two pixel values is related to the statistical approaches of Ratnatunga and Newell (1983) and Dressler and Gunn (1992); and the comparison

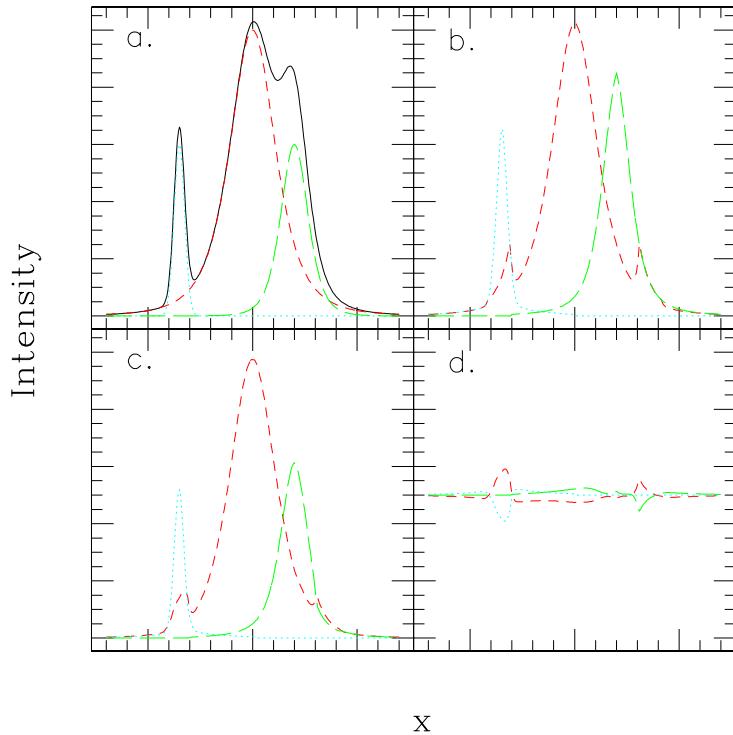


Fig. 1.— A simple 1-dimensional example of deblending. Panel a shows the blended object (solid) and the three components that make it up; from left to right they are referred to as ‘star’, ‘galaxy1’, and ‘galaxy2’. Panel b shows the corresponding *templates*; see text for details. Panel c shows the three deblended children; and panel d shows the difference between the three input components and the output of the deblender. The vertical scale is the same in all panels, but panel d has been shifted upwards to show negative as well as positive residuals.

of pixels placed symmetrically about the peak is inspired by Doi et al. (1995) and White and Keel (1992).

The generalisation to 2-dimensional multi-band data is immediate. The children are defined by the union of the distinct peaks found in each band, and the same set of centres is used in all bands. The templates are defined independently in the different bands (see section 5 for more discussion of this point), and the symmetry assumed is that the child has a two-fold rotational axis through its centre.

The result of applying this algorithm is a set of images of each child, one in each band. The sum of all the children is equal to the parent. The method is motivated by the realisation that most astronomical objects are symmetrical, have non-negative fluxes everywhere, and are at least somewhat optically thin, but the children born of this algorithm need *not* have any particular symmetry; the only place that symmetry is enforced is in the construction of the templates.

3. Application to SDSS data

The SDSS imaging data is taken with a large camera (Gunn et al 1998) containing 30 2048×2048 CCDs, arranged in 6 dewars each containing 5 CCDs with r , i , u , z , and g filters. Data is taken in TDI (Time Delay Integrate) mode, with the telescope scanning on great circles at the siderial rate; an object takes about five minutes to pass over all five of a dewar's CCDs. The plate scale is about 0.4 arcsec/pixel, and the limiting point-source 5σ magnitude in g is about 23.

The resulting data is thus a set of $30 \times 2048 \times n$ pixel strips, where n is about 1.35×10^5 per hour of data. For convenience in archiving and processing we cut the strip into 2048×1361 pixel *frames*; a set of five (u , g , r , i , and z) frames covering the same part of the sky is referred to as a *field*. The image processing is carried out field by field. To minimise the number of objects split between fields, 128 lines are added to each field from the succeeding one, so that actual processing is carried out on a region of 2048×1489 pixels.

For more details of the survey see York et al. (2000); other aspects of the image processing pipeline '*Photo*' are given by Lupton et al. (2001).

In what follows I assume that all the bands have been shifted to the same coordinate system.

Neglecting various unimportant details, objects are detected in each band by passing a PSF filter over the image and thresholding at 5σ ; only a single pixel need be over this threshold for an object to be detected. The objects defined by pixels at or above threshold are then ‘grown’ more-or-less isotropically by an amount approximately equal to the radius of the seeing disk.

An object in the output catalog is defined to be the union of all the pixels detected in at least one band. In some cases this leads to two objects that were distinct in (say) u being merged together based on data in (e.g.) r . This definition of an object is liberal; in some cases the members

of a blend are quite well-separated on the sky, but it seems safer to deblend well separated objects than to run the risk of not processing others and later regretting the decision.

All the detected pixels are saved in an ‘atlas image’; examples of such images are given in the figures which show the results of deblending. The careful reader will notice that the boundaries of the atlas images are in fact aligned on 4-pixel boundaries; see Lupton et al. (2001) for details.

Along with each object, *Photo* keeps a list of the peaks (in each band) that were detected within it. For the simple case of an isolated bright star this is simply a list of five measured positions and their errors, one in each band. For a high redshift quasar near a moderate redshift galaxy, this might be a single peak in g, r, and i; and two peaks in z.

An example application is given in figure 2. This was chosen from a (failed) search for supernovae in early SDSS data rather than as a particularly successful application of the method.

A rather more ambitious example in NGC991 (figure 3), which was chosen as a face-on spiral without reference to how well the deblender had performed. Residuals are visible near the bright star to the left of the nucleus and at the top, and many of the bright HII regions have been removed.

Another example is shown in figure 4, the core of the richness class III cluster Abell 1882 at a redshift of 0.14. The colour-magnitude diagram for all galaxies within 200 arcsec of the centre of the cluster is shown in figure 5 where children are shown as open symbols and isolated objects are shown filled. There is no obvious tendency for the deblended galaxies to be more scattered than the isolated ones; in particular the bright end of the tight E-S0 ridgeline is entirely composed of children.

15-20% of objects are blended in a typical high latitude run. The deblender is responsible for about 8% of the CPU time needed to process the data, or 12% of the time spent in measuring the properties of objects.

4. How Well does this Algorithm Work?

It is not easy to assess the performance of a deblender. One way would be to make an extensive series of simulations; an alternative is to use the properties of real astrophysical objects.

Figure 6 show two-colour diagrams for a small portion of the SDSS Data Release 3 (DR3; Abazajian et al. (2005)); only objects detected in all three bands are shown, and moving objects are rejected (see section 5). The magnitudes used in constructing these figures are ‘cmodel’ magnitudes, based upon fitting a model galaxy, convolved with the locally determined PSF, to the r data and using the same model (but the band’s PSF) in all five bands (Lupton et al. 2001).

The top panels show all objects classified as galaxies; the bottom panels show objects classified as stars. The left-hand panels show all isolated objects; the right-hand panels show all objects which were extracted as children from blends. In plotting the children, I selected those which had been

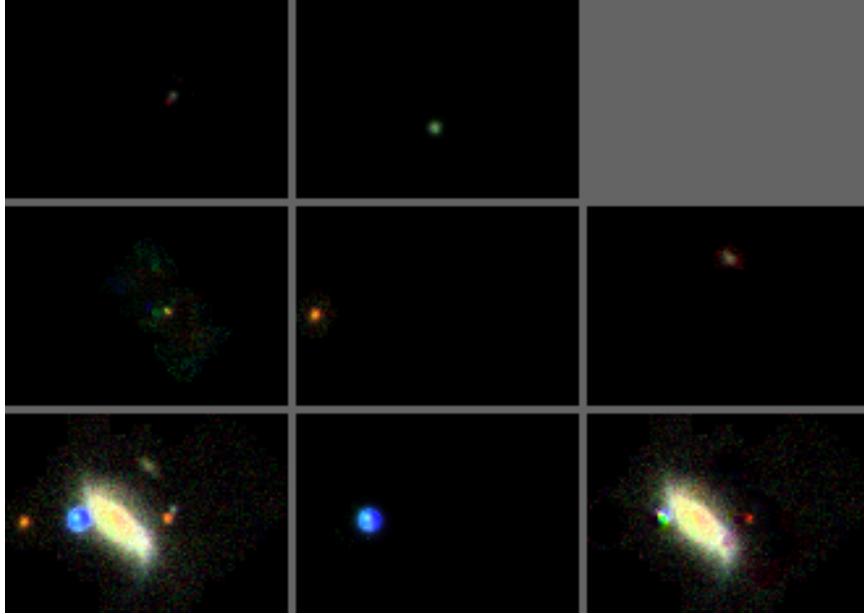


Fig. 2.— The results of the SDSS deblender applied to the object SDSS 756-3-356:1359,1051 (Lupton et al. 2001). The image is shown using a `linear 0:100` stretch (see below), clipping the bright star heavily. Figure 2b shows the same object with a `asinh:Q8 0:40` stretch to bring out the finer details of the object, and also the flaws in the deblended images.

Within each figure, the bottom left panel shows a g-r-i composite of the parent, prepared using the prescription of Lupton et al. (2004). The remaining panels each present a g-r-i composite of one of the children in approximate order of peak intensity. As discussed in the text, pixels outside the objects' atlas images are black.

The stretches employed in this paper are `linear a:b` (a linear stretch from `a` to `b`) and `asinh:Qn a:b` (an inverse hyperbolic stretch with softening parameter `n`; the linear part of the transformation is equivalent to a `linear a:b` stretch).

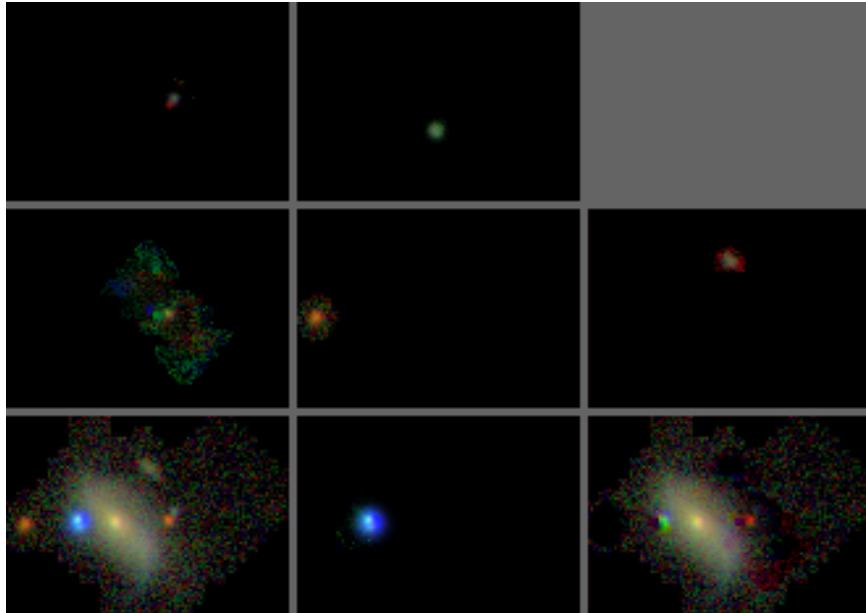


Fig. 2b.

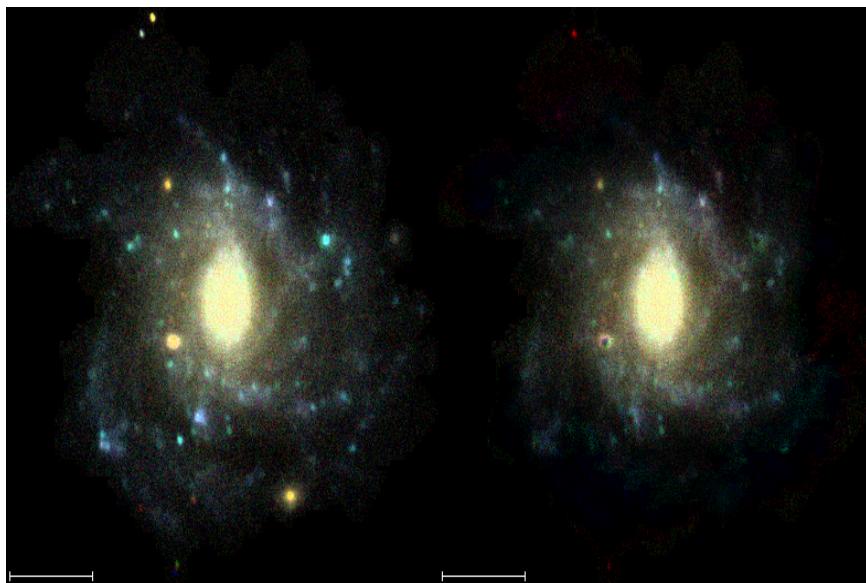


Fig. 3.— The results of the SDSS deblender applied to NGC 991. The image is shown using a linear 0:100 stretch (see the caption to Fig. 2). Only the parent and the child corresponding to NGC991 are shown; the scalebar is 30 arcsec long.

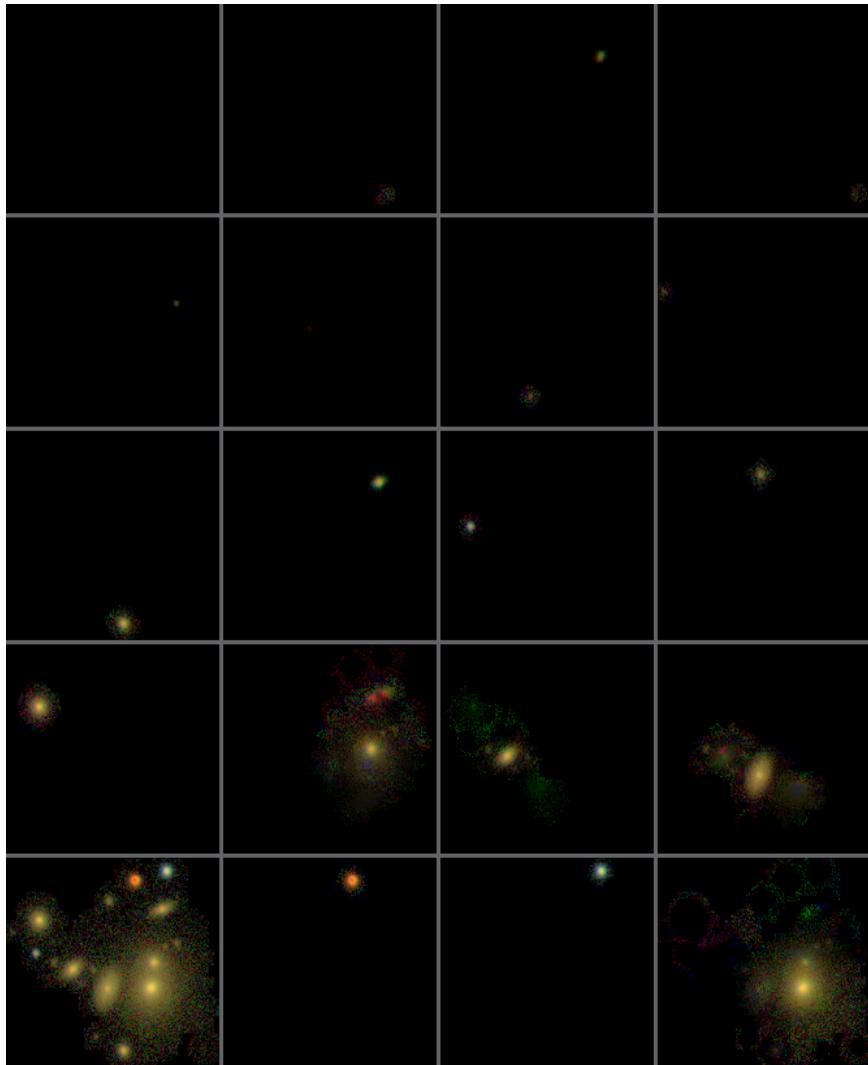


Fig. 4.— The results of the SDSS deblender applied to the object SDSS 752-3-468:582,471, shown using an `asinh:Q8 0:40` stretch (see the caption to Fig. 2)

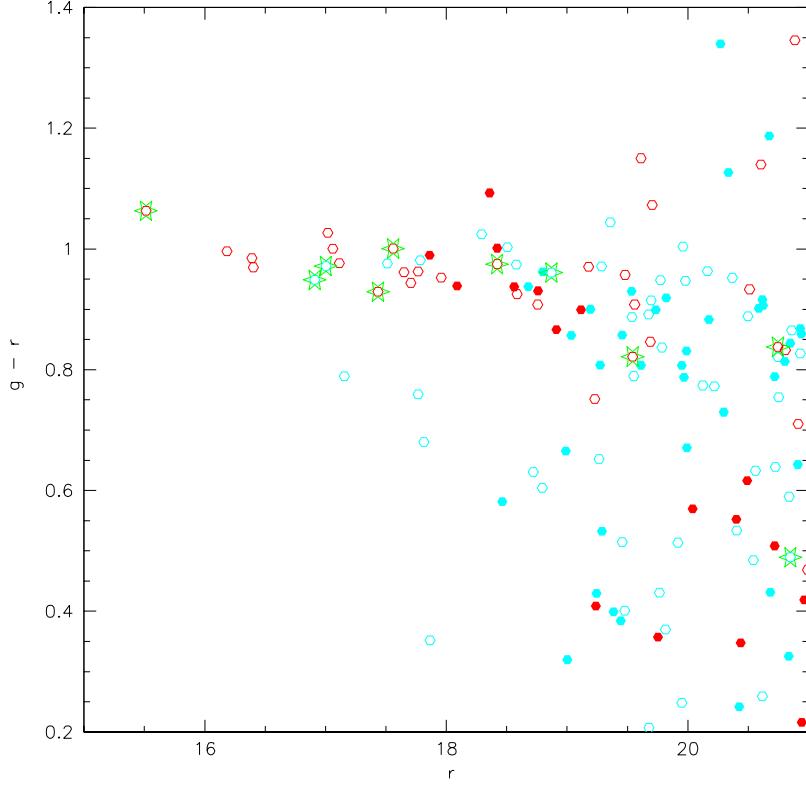


Fig. 5.— The colour-magnitude diagram of galaxies within 200 arcsec of the centre of the cluster core shown in Fig. 4; red symbols indicate objects classified as ellipticals (i.e. having deVaucouleurs profiles), and cyan symbols are exponentials. Children are shown as open symbols, while isolated objects are shown filled. The children of the object shown in Fig. 4 are outlined in green.

The scatter in isolated objects' colours is similar to the children's; for example, the 'E/S0' ridgeline (at $g - r \approx 1.1$) is comparably tight in the two symbol types.

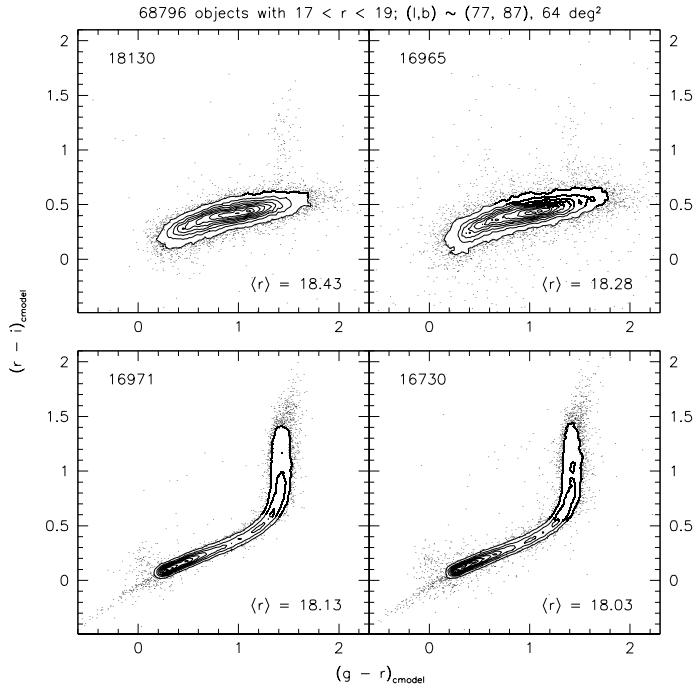


Fig. 6.— Two-colour ($g-r$, $r-i$) diagrams for 68796 objects selected from about 64deg^2 of DR3 (Abazajian et al. 2005) at $(l,b) \sim (77^\circ, 87^\circ)$, and for objects with $17 < r < 19$ (cmodel magnitudes; (Lupton et al. 2001)). The top two panels show all objects classified as galaxies; the bottom panels show objects classified as stars. The left-hand panels show all isolated objects; the right-hand panels show all objects which were extracted as children from blends. There is rather more scatter in the ‘child’ panels, but the fraction of outliers is still small.

The cloud of point-sources at $(g-r) \sim 0$, $(r-i) \sim 0.2$ are quasars, and the scattering of points in the region $(g-r) \sim 0.5$, $(r-i) \sim 1$ are mostly red-dwarf — white-dwarf binaries (Smolcic et al. 2004). In the top, galaxy, panel, the prominent feature at $(g-r) \sim 1.4$ are close binaries that the deblender failed to split.

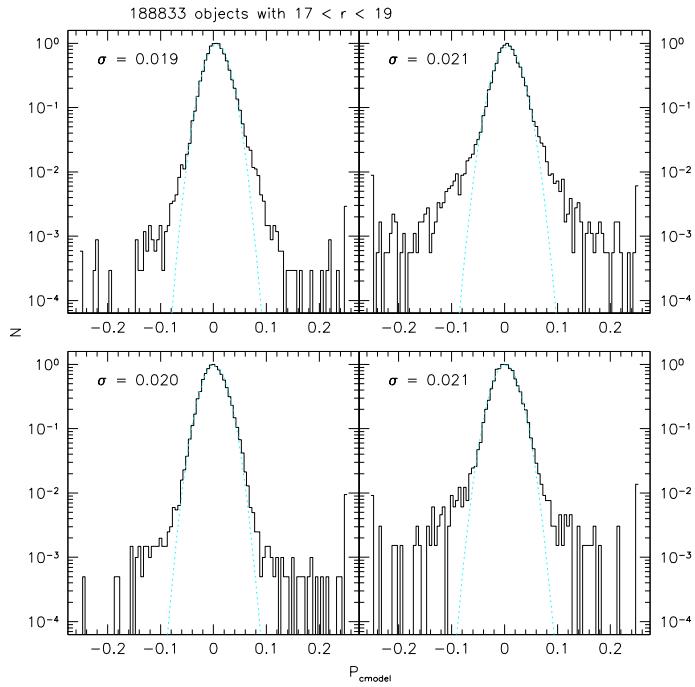


Fig. 7.— Cross-sections of the red part of the stellar locus ($0.27 < g - r < 0.81$); P is the ‘perpendicular colour’ resulting from rotating our coordinate system to lie along and parallel to the locus (cf. Ivezić (2004)). The bottom two panels show the data of Fig. 6, while the top two panels are equivalent, but for a 16deg^2 patch at $(l, b) \sim (190^\circ, 30^\circ)$ where the stellar density is about 6 times higher (although only about 1 object per 17^2 arcsec^2 to 21st magnitude). The dotted lines are clipped Gaussian fits to the data, and the quoted σ s are the corresponding widths. The children show slightly enhanced tails relative to the isolated objects, especially in the higher density field.

found as peaks in all three of g, r, and i, had formal errors of less than 0.5 magnitudes in each band. I additionally rejected all objects whose centres were severely compromised by interpolated pixels (due to e.g. cosmic rays or bleed trails of bright stars).

The ‘child’ panels are only slightly more scattered than the ‘isolated’ ones; this is quantified in Fig. 7 which shows the cross-section of the red part of the stellar locus for fields at $b \sim 30^\circ$ and $b \sim 90^\circ$.

5. Inevitable Real-Life Complexities

The algorithm presented in section 2 is simple but unfortunately the real world is not; in this section I discuss some of the tricks and compromises used in the SDSS implementation. Consumers of the SDSS catalogues would do well to be aware of some of these details, while others will probably only be of interest to those planning to code their own deblender.

First of all, there are offsets between the different bands’ coordinate systems which complicate the book-keeping; an alternative would be to repixel all data to the same pixel grid, taking due account of field distortions.

Each object carries a set of bitfield *flags* to record details about how its processing proceeded.

Before starting to measure object parameters, I replace all detected pixels by Gaussian noise with the same variance as the sky. Then as the turn comes for an object (parent or child) to be processed, it alone is inserted into the field. This means that the low-signal-to-noise outer parts of extended objects are measured twice.¹ The resulting errors are unimportant in processing SDSS data as the sky-estimation algorithm (Lupton et al. 2001) already over-subtracts sky from such large objects. If this problem were resolved, a more sophisticated approach would be needed.

5.1. Peaks

One essential input to the deblender is the list of peaks in the five SDSS bands. As each band is processed by the object finder, each detected object is searched for peaks — a non-trivial process as the code must deal with ‘plateaus’ of many pixels with exactly the same value which may or may not be adjacent to a still higher pixel. The resulting raw peak list is then ‘culled’, rejecting peaks that are joined to an adjoining higher one by a saddle point or valley less than 3 sky σ below the peak. Additionally, peaks may not be closer than some specified limit (currently 1 pixel, 0.4 arcsec), and the multiple spurious peaks within the interpolated cores of saturated stars are merged together. Note that this culling concerns peaks all of which are detected in the *same* band.

¹Specifically, portions of objects that cannot be detected at the 5σ level after smoothing with a Gaussian filter which has a FWHM four times larger than that of the PSF

The resulting lists of peaks in each band are then compared, and peaks that are closer than some number of positional σ are merged together. Unfortunately there are systematic errors in the relative astrometry, due to e.g. anomalous refraction (Pier et al. 2003), and the peak lists must be trimmed based on some limit in pixels rather than purely on σ . The criterion adopted for merging two peaks i and j is

$$(|c_i - c_j| - \sigma_{c,i} - \sigma_{c,j})^2 + (|r_i - r_j| - \sigma_{r,i} - \sigma_{r,j})^2 < \Delta^2$$

where r is the row-position, c the column-position, and dr and σ_r and σ_c their estimated errors. The adopted value of Δ is 2 pixels (0.8 arcseconds).

Throughout this process the actual measured position in each band of each surviving peak is preserved; we shall make use of these positions when we come to consider moving objects.

5.2. Preliminaries to Deblending

5.2.1. Objects at the Edge of the Frame

The deblender depends on comparing pairs of pixels to construct templates and also, as we'll soon see, on smoothing the image. Handling objects that lie close to the edge of the frame in any band would greatly complicate the code, so I simply refuse to deblend such objects, setting a bit `NODEBLEND` in a flag to record this failure, and another (`EDGE`) to indicate the reason² As the SDSS data from a given set of CCDs overlaps with the neighbouring scan by about 60arcsec, and a similar amount is added in the scan direction when processing a field, it might seem that `EDGES` would lead to problems only with objects whose diameter exceeds an arcminute, as smaller objects will be successfully processed as part of some other field. Objects this large can be processed by hand, which we will in any case have to do with e.g. large Messier galaxies. However, many significantly smaller objects are also lost due to the `EDGE` bit, due to the formation of complex blends containing a number of objects, one of which was often a severely saturated star with diffraction spikes and large scattering wings. Accordingly, blends that would be declared `EDGE` on *all* fields are trimmed until they just fit in a field, a `DEBLENDED_AT_EDGE` bit is set, and the deblender is allowed to proceed; other `EDGE` objects are not deblended.

5.2.2. Moving Objects

The next complexity to face is due to moving objects such as asteroids. The r' and g' data are taken about 5 minutes apart, and main belt asteroids move far enough during this interval that unless treated specially they will be deblended into one very red and one very blue object. Near

²These bits, and many others, are discussed in detail by Lupton et al. (2001)

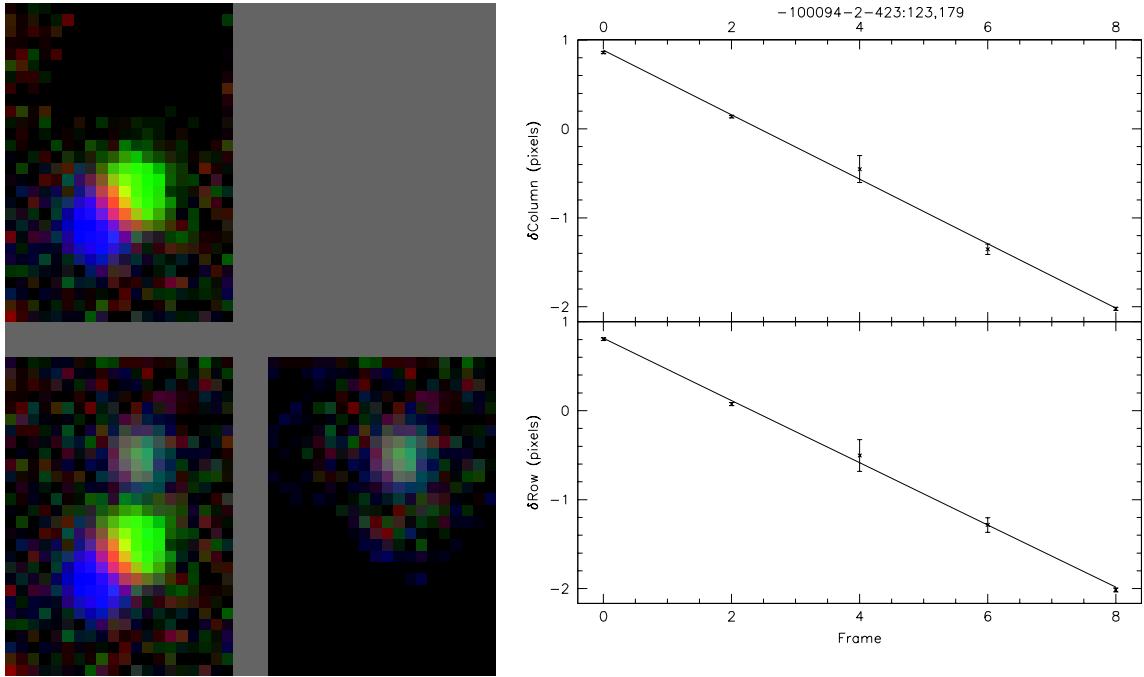


Fig. 8.— The results of the SDSS deblender applied to the object SDSS 94-2-423:123,179; the display is made with an `asinh:Q8 0:40` stretch (see the caption to Fig. 2); the right hand panel shows the row- and column-positions as a function of frame number, clearly showing the object's linear motion.

The brighter object is a main-belt asteroid, with a proper motion of about 41mas/minute; the fainter object is a star.

Earth objects are extended in each individual 55s exposure, but most asteroids are indistinguishable from PSFs. Accordingly, I search the peaks list for peaks that appear in only one band. If a given peak is detected in at least two bands it cannot be due to a moving object (as both peaks are, by definition, in the same place), but if two peaks are merely consistent that doesn't tell us that they don't belong to an asteroid. When all the single-band peaks have been found, if there are detections in at least three bands, and if there's at most one isolated peak in each band, I fit for two components of proper motion; if the χ^2 is acceptable and the resulting motion is sufficiently different from zero I decide that I've found something moving. The moving peaks are used to construct a special child; if any bands are not detected the missing centres are constructed using the velocity fit, and the deblender proceeds. If all peaks are accounted for the deblender is done; otherwise it continues to generate templates for the remaining peaks. For example, in the relatively common case of an asteroid moving past a star (8), the results of the deblender will be two children, although the parent may have up to 6 distinct peaks. During this process suitable flag bits are set to keep track of what happened.

Blend with two or more moving objects aren't handled by this approach. Blends with two bone-fide moving objects are rare but the deblender can become confused by combinations of (stationary) stars and asteroids, leading to serious incompleteness in asteroid catalogs produced from SDSS data. The problem is that when an asteroid is found moving past a brighter star, the *star* may be thought to be moving with a small but formally significant velocity; in this case, the deblender doesn't try to associate the various peaks belonging to the asteroid, resulting in its appearance as a number of separate objects each with very wrong (and very exciting) colours. It is possible to prevent this in particular cases by fiddling with thresholds, but this is not a robust procedure suited to the automatic processing of tens of millions of objects. The approach actually adopted is discussed in section 5.4.

5.3. Templates

As the reader doubtless realised while reading about templates in section 2, the minimum of two noisy values is biased. I therefore add a multiple of the sky noise to the template; I assume that the noise is Gaussian, so the correction applied is 0.5641895835σ . Note that prior to running the deblender all detected objects were replaced by noise, so this debiasing is correct even when one of the pairs of pixels lies within some other object.

Not all blends are composed entirely of galaxies, and the algorithm described in section 2 doesn't work well when stars are superimposed on extended sources. I therefore examine each child, trying to decide whether it's consistent with being a PSF. The decision is based on fitting a model of the PSF, background, and background gradient to the region around the peak; see Lupton et al. (2005) for details. If it is well described by a PSF, I use an approximation to the PSF as the template (using the proper PSF in each band), subtract a simple model of the PSF, and set `DEBLENDED_AS_PSF` to warn the user. These PSF-subtracted images are only used while constructing

templates for other peaks. Note that the final children are not constrained to be identical to their PSF templates.

There are a couple of special cases: Saturated peaks are never treated as PSFs, as the star’s scattering wings and diffraction spikes dominate the profile; and the central peak of a blend is never treated as a PSF even if it appears stellar as it is all too likely to be a pointlike or nearly-pointlike galactic nucleus. It’s tempting to think that the brightest peak is the one to be treated specially, but unfortunately the brightest peak in a complex blend is usually a foreground, often saturated, star.

To find the peak most likely to be the centre of an extended source (if it is indeed extended; the same procedure does no harm if it is not), I median smooth the entire blend with an $n \times m$ filter and find the position of the highest peak in the resulting smoothed image. I then look for the brightest of the original peaks within a box with sides $2n/3, 2m/3$ centered at this point and declare it to be the ‘central peak’. The smoothing filter is taken to be about 1/8 the size of the blended object, but not less than 16×16 pixel — 4.8×4.8 arcsec — unless the object itself is smaller than this.

I must now generate templates for non-DEBLENDED_AS_PSF peaks. If the peak isn’t detected in a given band I don’t generate a template directly from the image (see below). When objects are moving, there’s some danger that the estimated peak position in some band may lie outside the frame; caveat processor. Fortunately this is rare, and almost always caused by wild estimates of the peak’s velocity.

With the templates for a given peak in hand, I smooth each in turn with the appropriate PSF and run an object finder on the smoothed template. If more than one object is found, I keep the one that contains the original peak. If the template isn’t detected in *any* band, the corresponding peak is discarded as a possible child, otherwise templates for bands without peaks are generated as the average of templates in the good bands; portions of templates that lie outside the original detected region are set to zero.

The ansatz used in inventing this deblender is that all children are defined by a single peak in the parent object; it therefore makes no sense for a template to be multipeaked. If a template *does* contain more than one peak, it’s usually caused by an blended object containing a set of three roughly equal peaks lie approximately equi-spaced along a straight line; the end peaks present no difficulties, but the central one’s template looks very like the initial blend. In this case, I find the level t of the saddle point joining it to the main peak and set all pixels above t within the subsidiary peak to t . This isn’t ideal — it’d probably be better to run a deblender on the template — but it isn’t clear that the resulting added complexity could be justified. An alternative approach is discussed in section 6.

With all the templates in hand we’re ready to actually deblend, but in practice I add another step. Some objects were declared DEBLENDED_AS_PSF — but what are the consequences if they are actually extended, and no other template is non-zero in their vicinity? The central part of their

flux (within the footprint of the PSF) will be correctly assigned, but the rest will be in limbo, to be assigned using a rather naïve heuristic described below. Rather than allow this, I reconsider all `DEBLENDDED_AS_PSF` children, and if no other templates are able to provide the background level found when fitting a PSF the `DEBLENDDED_AS_PSF` bit is set, the PSF subtracted at that point is reinstated, and a regular symmetrized template calculated.

It would probably be better to complete the deblending of the object, run a star/galaxy classifier on each child, and then rerun the deblender with the `DEBLENDDED_AS_PSF` question settled, but I felt that this would be too expensive.

Section 6 discusses the possible use as templates of model fits to the children; such an approach would naturally include a robust star/galaxy classification step.

5.4. The Peephole-Optimiser

A peephole-optimiser in a compiler looks for short sequences of instructions that can be *locally* improved.³ In the context of the SDSS deblender, the optimiser looks for deblends that are clearly unsatisfactory, and that can be improved by a making minor modifications to the template list.

The only optimisation currently implemented is to look for groups of children that clearly belong to a single moving objects (cf. 5.2.2); for example a child with only detections in r and i, near an object with detections in z and g, and with the 4 peak positions consistent with linear motion. This is more easily done at this stage as the process of template construction has carefully whittled down the list of children to a set that appear to make sense as stationary objects; of course none of the considerations of the previous section knew of the possibility of moving objects, and, indeed, proceeded essentially independently in each band.

This step works well; in retrospect it seems possible that the entire careful treatment of moving objects in section 5.2.2 could have been avoided.

5.5. Solving for the Children

We have to solve for the w_r in

$$I_i = \sum_{r=1}^n w_r T_{r,i} + \epsilon_i$$

(equation 2), leading to the Normal Equations

$$\mathbf{A}\mathbf{w} = \mathbf{b}$$

³A classic example is replacing unsigned integer multiplications by powers of two with a left-shift by an appropriate number of bits.

with

$$A_{rs} = \sum_i T_{r,i} T_{s,i} \quad (4)$$

and

$$b_r = \sum_i I_i T_{r,i} \quad (5)$$

These equations use the same weight for all pixels as this gives relatively higher weight to the centers of objects. Equation 2 is not a statistical model, so there is no reason to weight by the pixel variance. Furthermore, all templates are normalised to have a peak value of unity, which makes the interpretation of the w_i (and \mathbf{A} 's eigenvalues; see below) simpler.

The usual way to solve non-trivial least-squares problems is by Singular Value Decomposition (SVD; e.g. Press et al. (1992)), but this involves $r \times n$ matrices, where r is the number of points in the model, and n the number of free parameters.

An alternative procedure which seems to preserve most of the advantages of SVD but at much lower computational cost is to solve the Normal Equations by means of an eigenvector decomposition. The only disadvantage that I'm aware of is that this approach effectively solves the square of the SVD problem, thereby squaring the condition number of the matrices involved. Although least-squares problems are plagued by singular matrices, this is usually due to ill-posed problems rather than delicate numerical cancellations, and squaring the condition number doesn't appear to lead to significant difficulties.

Let us write the eigenvector decomposition of \mathbf{A} as $\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$ where \mathbf{R} is an orthogonal matrix, and $\mathbf{\Lambda}$ is diagonal, with its non-zero elements equal to \mathbf{A} 's eigenvalues. We may then solve for \mathbf{w} as

$$\mathbf{w} = \mathbf{R}^T \mathbf{\Lambda}^{-1} \mathbf{R} \mathbf{b}$$

So far we have gained nothing except complexity and computational cost from using an eigenvalue decomposition. As for SVD, we gain because we can handle cases where \mathbf{A} is nearly singular as this is reflected in near-zero eigenvalues. If the smallest eigenvalue is too small, I find the template which is most nearly represented by the corresponding eigenvector and reject it; this procedure is repeated until \mathbf{A} is reliably invertable. There is of course no need to return to the pixel data to do this, I merely have to reject the proper elements from \mathbf{A} and \mathbf{b} .

What does it mean if an eigenvalue is too small? It corresponds to the case that the dimensionality of \mathbf{A} is lower than the number of templates, i.e. two templates represent essentially the same object.

If any of the w_r are negative, I set them to zero. It can also happen that the product $w_r T_r$ can lead to a child that wouldn't have been detected in any band if isolated; if this happens I reject that child and solve for \mathbf{w} anew. For objects superimposed on other objects, I make the detectability condition somewhat tighter: if the per-pixel flux in the parent is greater than ten

times the detection threshold, I require that the child be detected at least three times that threshold. This has the beneficial effect of suppressing noise peaks and minor surface brightness fluctuations within large objects. It would be easy to invent a more sophisticated approach.

The next stage in rejecting templates is to ask if any of them are too nearly parallel (in a space with dimension equal to the number of pixels in the blend), i.e. if the inner product of any two templates is too close to unity. Such pairs of templates represent essentially the same object, and this step is especially important for successfully deblending large late-type face-on spirals. More precisely, for each pair of templates i and j , I calculate

$$\cos_{rs} \equiv \sum_c A_{rs} / \left(\sum_c A_{rr} \sum_c A_{ss} \right)^{1/2}$$

where the A_{rs} are defined in Eq. 4 and the sum over c represents a sum over all bands, and then look for the smallest value of \cos_{rs} (with $r \neq s$). If this exceeds a critical value (I adopt 0.5) either child r or s is rejected; if only one is `DEBLENDED_AS_PSF` it is the unlucky one, otherwise I choose the one with the smaller peak value. It is necessary to transfer book-keeping information from the rejected to the retained peak, as the object finder may have detected only one of them — after all, they are degenerate.

5.6. Constructing the Children

For each pixel in the blend, if only one template is non-zero the corresponding child is given all the flux. Otherwise, I calculate the sum given in equation 3, namely

$$\mathcal{S} \equiv \sum_{r=1}^n w_r T_{r,i} \quad (6)$$

If \mathcal{S} is large enough (I use > 10) I simply apply equation 3; otherwise I calculate \mathcal{S}' , replacing the templates T_r in equation 6 by their smoothed equivalents. If \mathcal{S}' is large enough (I use > 1) the flux is split between the components using the obvious generalisation of 3; if it isn't, no flux is assigned to children at this stage (see below).

After going through every pixel in the blend there may still be unassigned flux, from low S/N areas of the parent or from regions where the only non-zero template didn't make it into the final list of children. In order to assign this flux to the children, I go through the list of children, and calculate for each child $\mathcal{F}_{r^2} \equiv \sum_i (I_i r_i^2)$ where r_i is the distance from a pixel to the object's center. This is an estimate of (total flux)*(scale size)².

For each pixel with unassigned flux, I give it all to the child with the largest value of \mathcal{F}_{r^2}/R^2 where R is the distance from the pixel to the child. There is a flag bit, `DEBLEND_UNASSIGNED_FLUX`, associated with objects for which a substantial amount of flux was parcelled out in this manner; in routine SDSS processing this bit is set if more than $> 5\%$ of the Petrosian flux is unassigned.

An extreme example of a deblend with a lot of unassigned flux is the low surface brightness galaxy shown in figure 9; the deblender has not done an especially good job.

5.7. Errors in Deblended Children

It is not sufficient to create children, we also need to know the errors associated with the process. At present the SDSS tackles this by adjusting two terms in the error budget; the background level, and the error in the background level.

A child’s ‘sky’ level is taken to be the true sky at its centre plus the flux at that point due to all its siblings, and the error in the sky is taken to be the root-mean-square of the pixels underlying the child. This latter term is clearly an over-estimate — it assumes that the deblender failed to remove extraneous flux, and it would be better to at least remove a simply fit to the subtracted background before estimating the rms fluctuation.

6. Future Directions

A number of improvements and extensions to the algorithm presented in this paper come to mind although they were not implemented in the SDSS data processing system.

6.1. High-density Stellar Fields

The *Photo* deblender was primarily designed to handle high-Galactic latitude fields where the bulk of objects are extended (the SDSS project was originally designed to only image fields with $|b| > 30^\circ$). Unfortunately, we have now imaged large areas of the sky near the Galactic plane — how should we reduce these data? As discussed in section 5.3, peaks that look like point sources are treated specially. If *all* peaks were *a priori* known to be point-like, we would have reinvented a classic crowded-field PSF code. In practice, if we included a prior probability of an object’s being a point-source as well as the likelihood that the object was well fitted by a PSF, we could implement a code that smoothly switched behaviour to follow the properties of the data.

6.2. Consistent Deblending for a Stack of Many Images

The algorithm described in this paper handles sets of images taken in different colours consistently. What should we do if we have a set of images taken at different times with the same filter? One approach would be to only analyse the difference images (Alard and Lupton 1998), and to somehow add together the images and perform a static analysis on the resulting deep image. There are difficulties associated with both of these paths (Is the astrometry in the difference image really

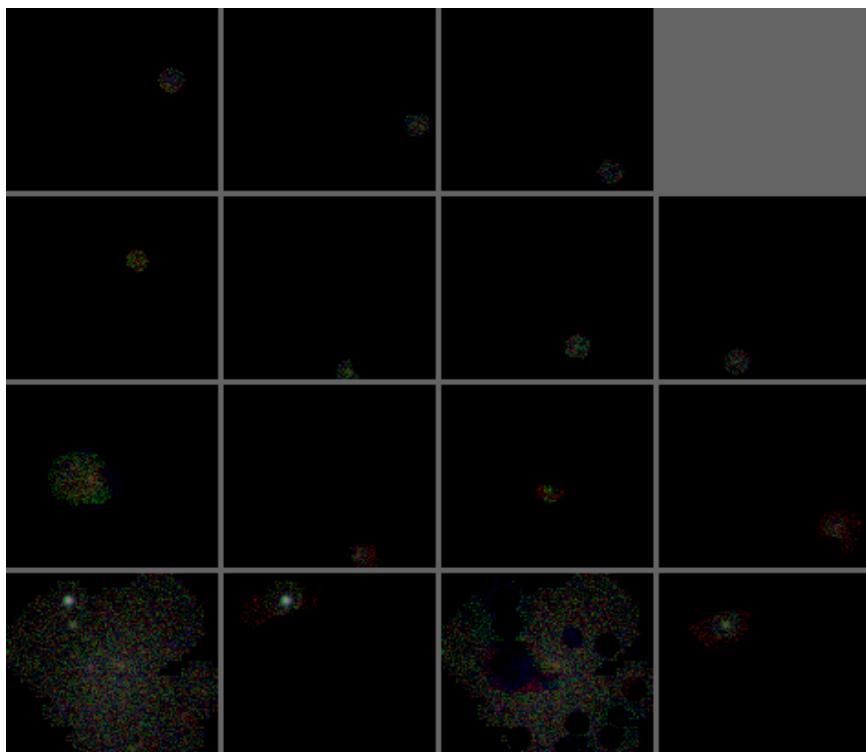


Fig. 9.— The results of the SDSS deblender applied to the object SDSS 752-3-327:769,1770; the display uses an `asinh:Q8 0:40` stretch (see the caption to Fig. 2).

as good as it would have been on the [uncrowded] input images? how should we weight the images going into the stack?), and so for some purposes we would like to analyse the ensemble of images individually. We need to ensure that the deblender behaved consistently in each image, rather than sometimes splitting a close pair of stars, and sometimes treating them as a single compact galaxy.

Let us ignore the problem of moving objects for now. We could use the *Photo* deblender on each input image by anointing a single set of ‘correct’ templates and running the flux-assignment step separately for each exposure. How should these templates be determined? One approach would be to run the *Photo* deblender on the coadded image, save the final template set, and use it for each of the input images. There are a couple of obvious problems with this approach: how would we allow for variable seeing, and do we really want to carry around a pixel-based description of each child? The former is probably not too serious (remember that the children are not identical to the templates), but the latter is a real concern.

An approach that would seem to be worth pursuing is to replace each template derived from the coadded image by a best-fit model of the child (either a PSF or a simple PSF-convolved galaxy model⁴) convolved with the PSF. This solves both of the problems raised above, and would also alleviate the difficulties raised in section 5.3 of multi-peaked templates.

RHL thanks Jim Gunn for many helpful conversations, and Michael Strauss for helpful comments on an early version of this manuscript.

This research has made use of NASA’s Astrophysics Data System Abstract Service

Funding for the creation and distribution of the SDSS Archive has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Aeronautics and Space Administration, the National Science Foundation, the U.S. Department of Energy, the Japanese Monbukagakusho, and the Max Planck Society. The SDSS Web site is <http://www.sdss.org/>.

The SDSS is managed by the Astrophysical Research Consortium (ARC) for the Participating Institutions. The Participating Institutions are The University of Chicago, Fermilab, the Institute for Advanced Study, the Japan Participation Group, The Johns Hopkins University, Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, University of Pittsburgh, Princeton University, the United States Naval Observatory, and the University of Washington.

REFERENCES

Alard, C, and Lupton, R. H. 1998 AJ 503 325

⁴An elliptical Sersic (Sersic 1968) model with 4 free parameters (n , r_e , a/b , ϕ) may be sufficient.

- Abazajian, K. et al. AJ 2005 129 1755
- Buonanno R, Buscema G, Corsi C.E., Ferraro I, Iannicola G, AA 1983 126 278
- Beard S. M., MacGillivray H. T., and Thanisch P. F., MNRAS 1990 247 311
- Bertin E. and Arnouts S., AAS 1996 117 393
- Caesar, C. J., De Bello Gallico c. 51 B.C.
- Doi M., Fukugita M., and Okamura S., ApJS 1995 97 59
- Dressler A. and Gunn J. E., ApJS 1992 78 1
- Fan, X. et al. 1999, AJ 118 1
- Gunn, J. E. et al. 1988, AJ, 116, 3040
- Irwin M. J. MNRAS 1985 214 575
- Ivezić Ž. et al. AN 2004 325, No. 6-8, 583
- Jarvin J. F. and Tyson J. A. AJ 1981 86 476
- Lauer T R., ApJ 1986 311 34
- Lauer T. R., ApJ 1988 325 39
- Lupton R. H., and Gunn J. E., AJ 1986 91 317
- Lupton R. H. et al., 2001 ‘The SDSS Imaging Pipelines’ in ASP Conf. Set. 238, Astronomical Data Analysis Software and Systems X.
- Lupton R. H., Ivezić Z., Gunn J. E., etc., 2005 in prep.
- Lupton Robert H., et al., PASP 2004 116 133
- Pier, J. R., et al., AJ 2003 125 1559
- William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes: the Art of Scientific Computing. 1992, Cambridge University Press.
- Ratnatunga K. U. and Newell E. B., AJ 1983 89 176
- Schechter P. L., Mateo M., Saha A., PASP 1993 105 1342
- Sersic, J.L. 1968, *Atlas de Galaxias Australes, Observatory Astronomico, Cordoba.*
- Smolcic, V., et al., ApJ 2004 615L 141

Stephan M. E., C. R. Acad. Sci. Paris 1877 84 641

Stetson P. B., PASP 1987 99 191

Tody D Photometry Program User's Guide, 1981 (Tucson KPNO)

Tyson J. A. , Kochanski G. P., and Dell'Antonio I. P. ApJLett 2000 498 L107

Tyson J. A., 2000. Private communication.

Valdes F., Focas User's Manual 1982 (Tucson, NOAO)

Weir N., Fayyad U. M., Djorgovski S. G., and Roden J., PASP 1995 107 11243

White III, R. E. and Keel W. C. Nature 1992 359 129

York et al., 2000 AJ 120 1579