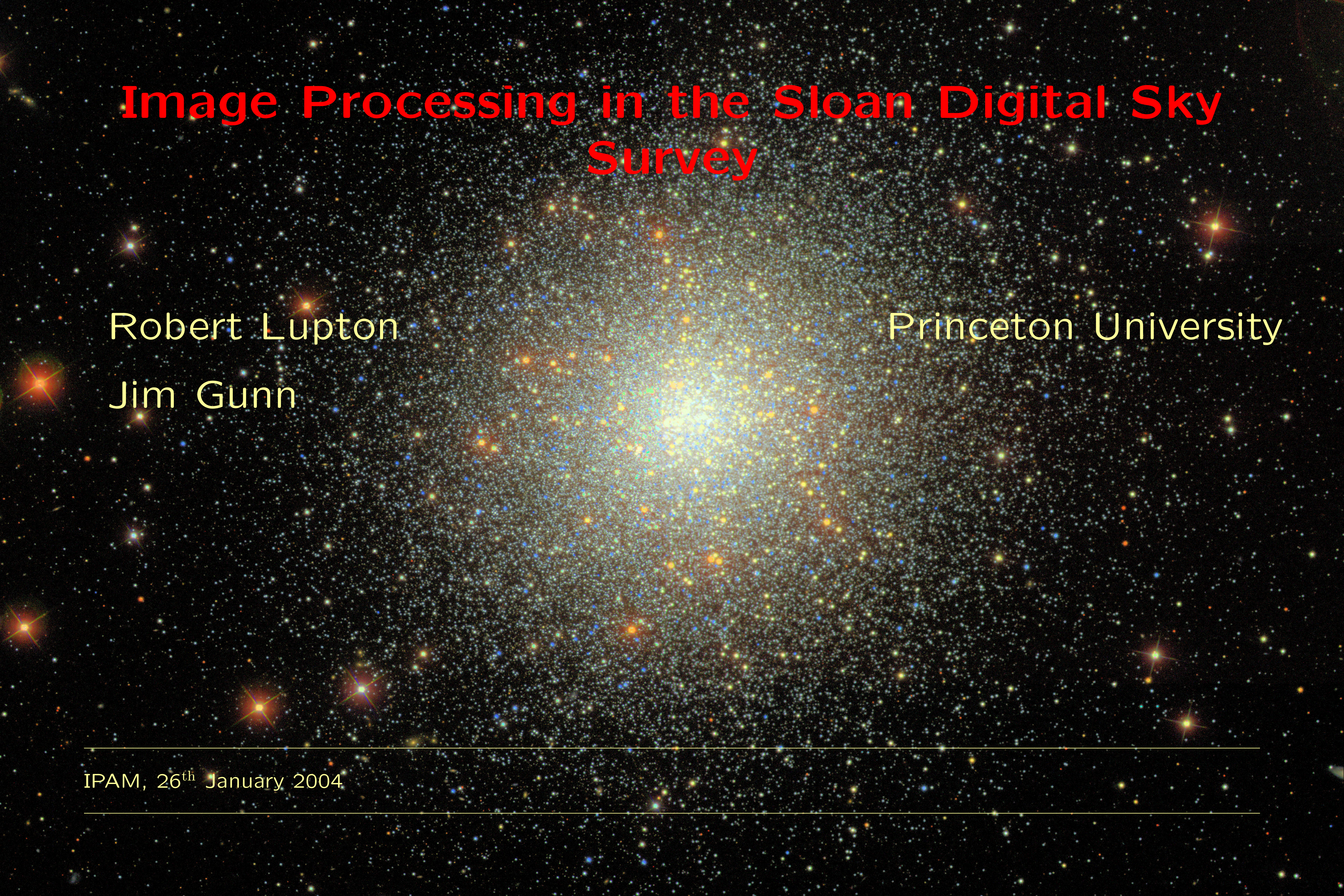


Image Processing in the Sloan Digital Sky Survey

Robert Lupton

Jim Gunn

Princeton University

IPAM, 26$^{\mathrm{th}}$ January 2004

Outline

Outline

Introduction

What is the SDSS?

Outline

Introduction

What is the SDSS?

Deblending

How should we handle overlapping objects, while not shredding NGC galaxies into multiple pieces?

Outline

Introduction

What is the SDSS?

Deblending

How should we handle overlapping objects, while not shredding NGC galaxies into multiple pieces?

Galaxy Fluxes

How should we measure galaxies' fluxes?

Outline

Introduction

What is the SDSS?

Deblending

How should we handle overlapping objects, while not shredding NGC galaxies into multiple pieces?

Galaxy Fluxes

How should we measure galaxies' fluxes?

Coloured Images

How should we make coloured pictures, and should we bother?

What is the Sloan Digital Sky Survey?

What is the Sloan Digital Sky Survey?

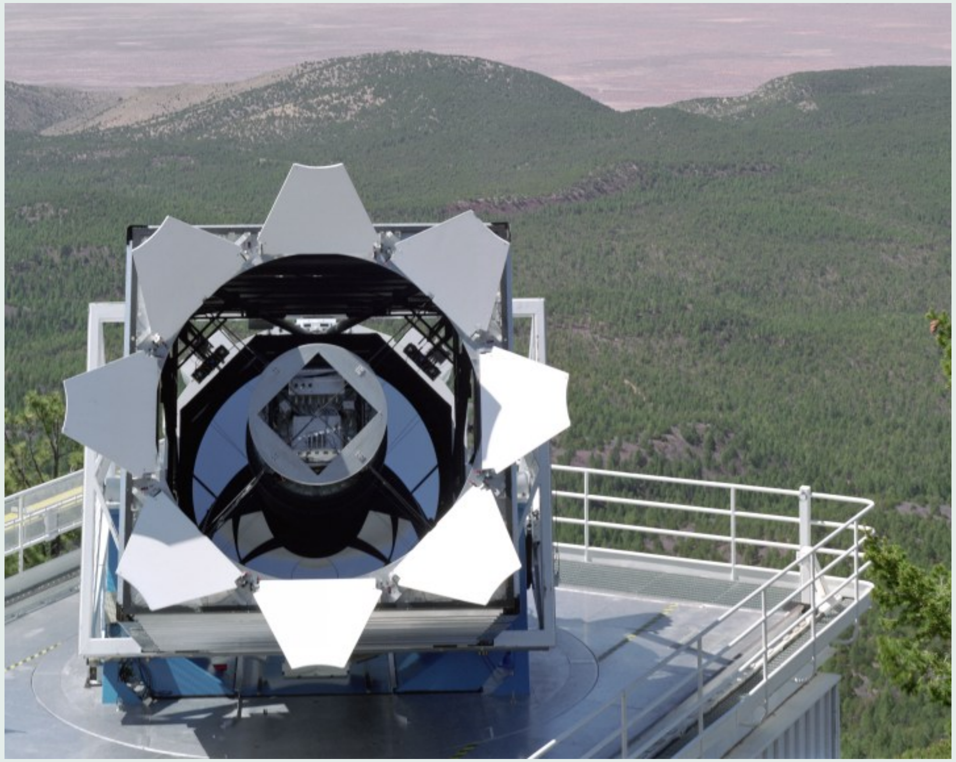
- A telescope (with a 2.5m diameter primary mirror) at Apache Point, New Mexico

What is the Sloan Digital Sky Survey?

- A telescope (with a 2.5m diameter primary mirror) at Apache Point, New Mexico

What is the Sloan Digital Sky Survey?

- A telescope (with a 2.5m diameter primary mirror) at Apache Point, New Mexico



What is the Sloan Digital Sky Survey?

- A telescope (with a 2.5m diameter primary mirror) at Apache Point, New Mexico
- A camera containing
  - 30 2048×2048 photometric CCDs;
    u g r i z filters
  - 24 2048×400 astrometric and focus CCDs
  - Lots of Electronics, Quartz, Liquid Nitrogen, and Vacuum

What is the Sloan Digital Sky Survey?

- A telescope (with a 2.5m diameter primary mirror) at Apache Point, New Mexico
- A camera containing
  - 30 2048×2048 photometric CCDs; u g r i z filters
  - 24 2048×400 astrometric and focus CCDs
  - Lots of Electronics, Quartz, Liquid Nitrogen, and Vacuum
- Lots of software

Deblending Overlapping Images

The problem of deblending stars is well defined; the image is made up of a set of δ-functions convolved with a known PSF, ϕ:

The problem of deblending stars is well defined; the image is made up of a set of δ-functions convolved with a known PSF, ϕ:

$$I = S + \sum_r F_r \delta(\mathbf{x} - \mathbf{x_r}) \otimes \phi + n$$

(I : observed intensity; S : sky level; δ: delta-function; F_r: flux in r^{th} star; ϕ : PSF; n : noise)

All that we have to do is solve a minimisation problem in $3r$ unknowns.

The problem of deblending stars is well defined; the image is made up of a set of δ-functions convolved with a known PSF, ϕ:

$$I = S + \sum_r F_r \delta(\mathbf{x} - \mathbf{x_r}) \otimes \phi + n$$

(I : observed intensity; S : sky level; δ: delta-function; F_r: flux in r^{th} star; ϕ : PSF; n : noise)

All that we have to do is solve a minimisation problem in $3r$ unknowns.

Writing efficient, robust, accurate code may not be trivial.


Galaxies are harder.

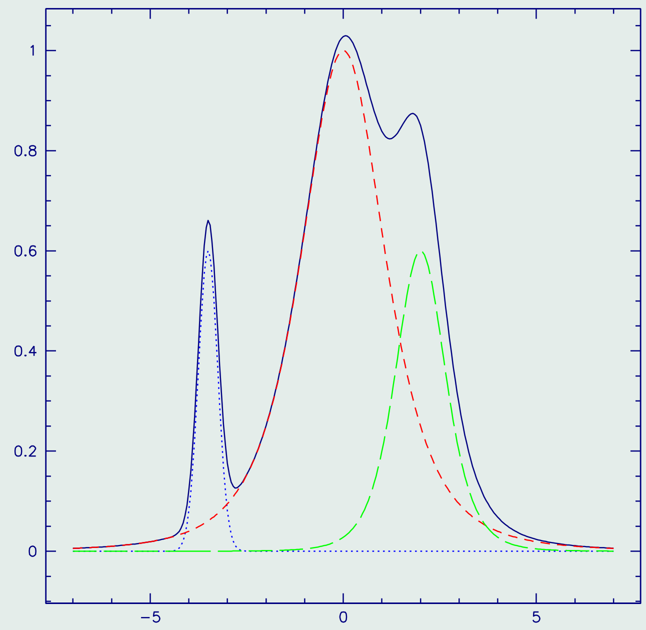
Galaxies are harder.

Something that looks like the superposition of three galaxies 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 three peaks — or even a large elliptical galaxy that's being viewed through a particularily perverse dust cloud.

A 1-D Toy Problem

A 1-D Toy Problem

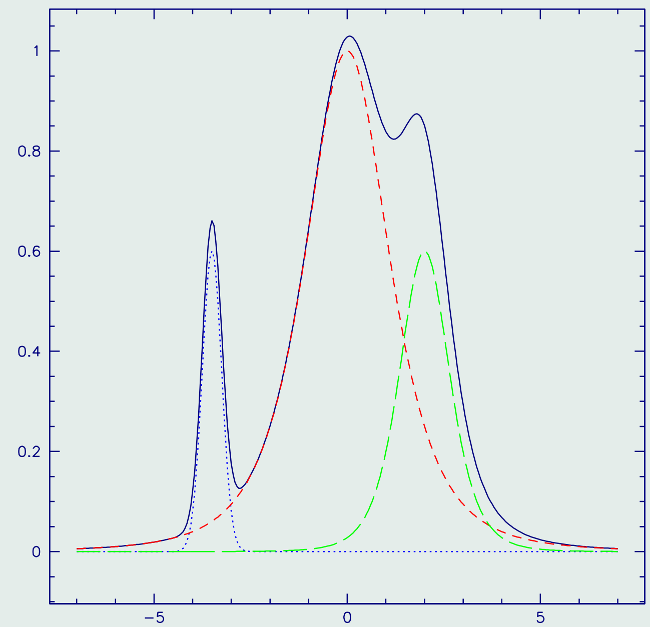
Let us consider a simple 1-dimensional problem, a ‘star’ and two ‘galaxies’.



The procedure is simple:

The procedure is simple:

- Find all the peaks in the image I. Each is associated with a 'child' object.

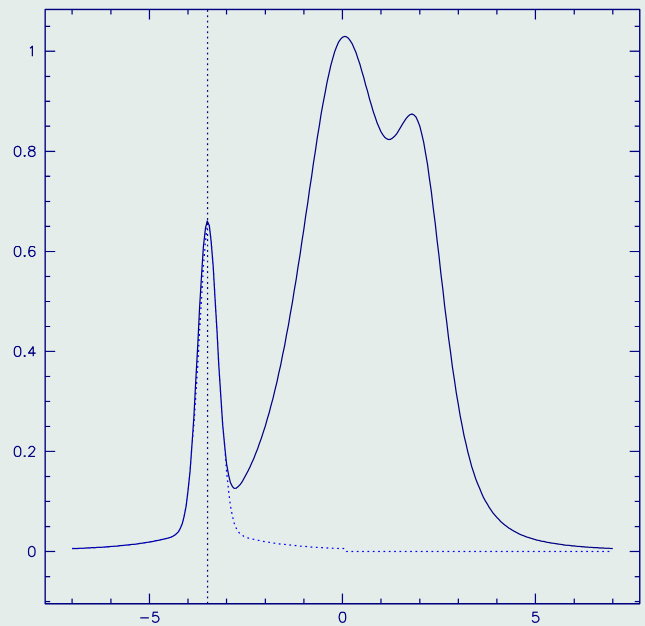


The procedure is simple:

- Find all the peaks in the image I. Each is associated with a 'child' object.
- Define a 'template' T_r from each peak. This is the image formed by comparing pairs of pixels symmetrically placed about the peak of the r^{th} object, and replacing both by the *lower* of the two.

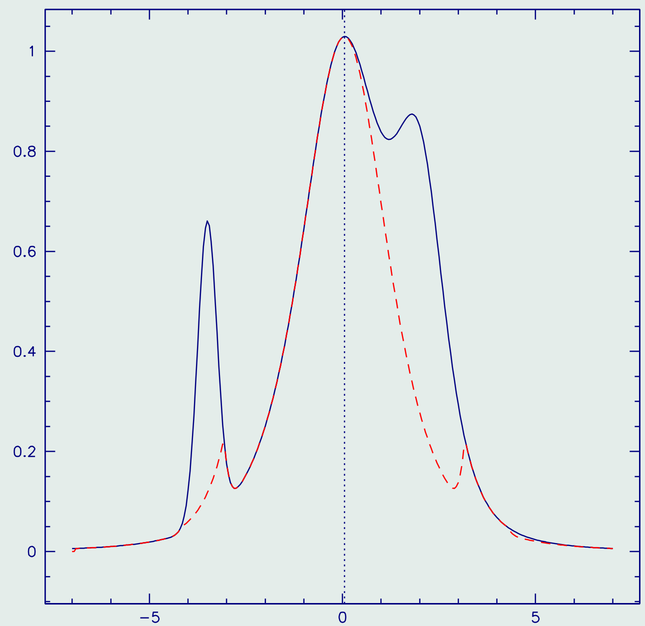
The procedure is simple:

- Find all the peaks in the image I. Each is associated with a 'child' object.
- Define a 'template' T_r from each peak. This is the image formed by comparing pairs of pixels symmetrically placed about the peak of the $r^{\rm th}$ object, and replacing both by the *lower* of the two.



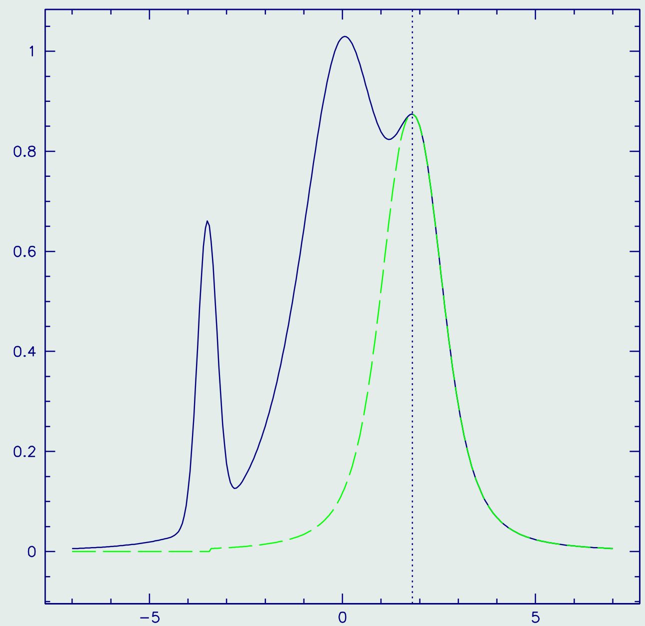
The procedure is simple:

- Find all the peaks in the image I. Each is associated with a 'child' object.
- Define a 'template' T_r from each peak. This is the image formed by comparing pairs of pixels symmetrically placed about the peak of the $r^{\rm th}$ object, and replacing both by the *lower* of the two.



The procedure is simple:

- Find all the peaks in the image I. Each is associated with a 'child' object.
- Define a 'template' T_r from each peak. This is the image formed by comparing pairs of pixels symmetrically placed about the peak of the r^{th} object, and replacing both by the *lower* of the two.



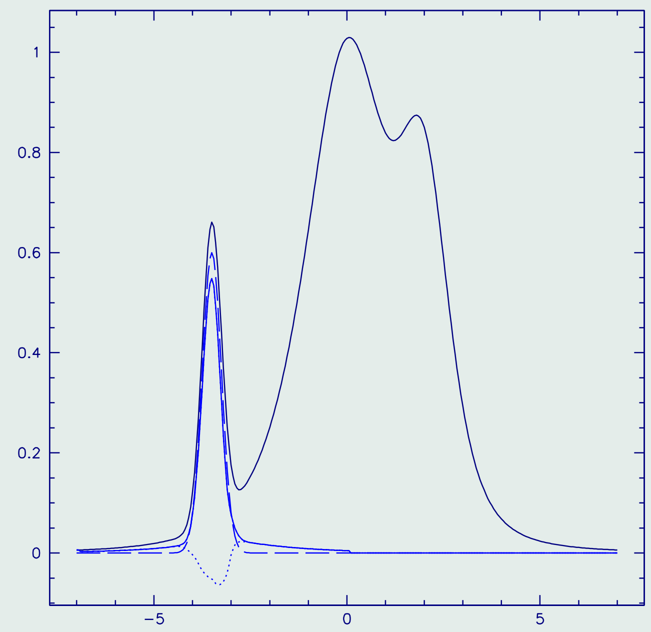
- Assume that we can write $I = \sum_r w_r T_r$, and solve for the weights in a least-squares sense.

- Assume that we can write $I = \sum_r w_r T_r$, and solve for the weights in a least-squares sense.
- For each pixel with intensity I_i, share the flux between the children:

$$C_{r,i} = \frac{w_r T_r}{\sum w_r T_r} I_i$$

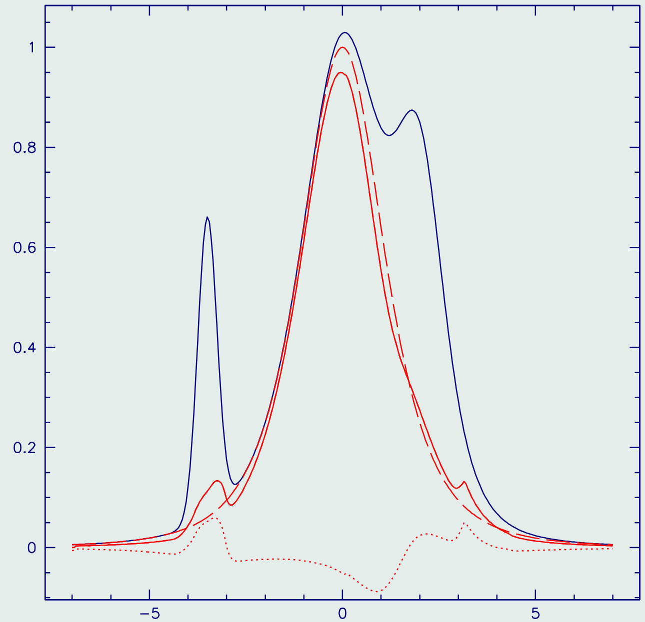
- Assume that we can write $I = \sum_r w_r T_r$, and solve for the weights in a least-squares sense.
- For each pixel with intensity I_i, share the flux between the children:

$$C_{r,i} = \frac{w_r T_r}{\sum w_r T_r} I_i$$



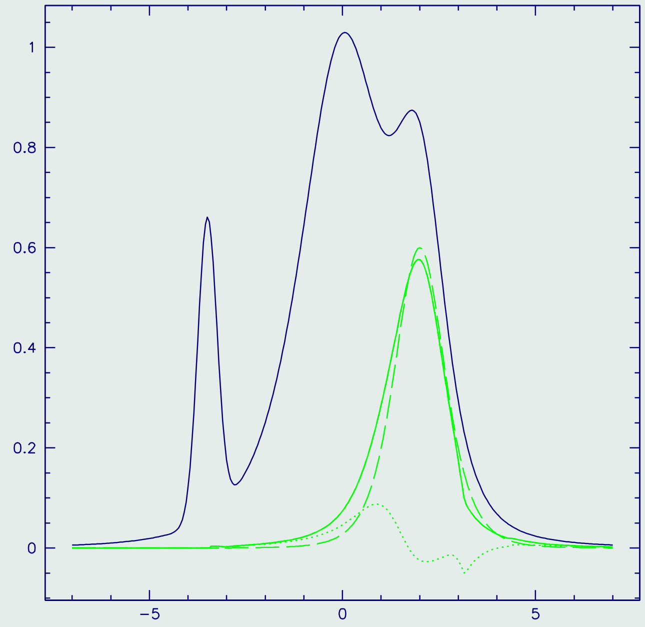
- Assume that we can write $I = \sum_r w_r T_r$, and solve for the weights in a least-squares sense.
- For each pixel with intensity I_i, share the flux between the children:

$$C_{r,i} = \frac{w_r T_r}{\sum w_r T_r} I_i$$



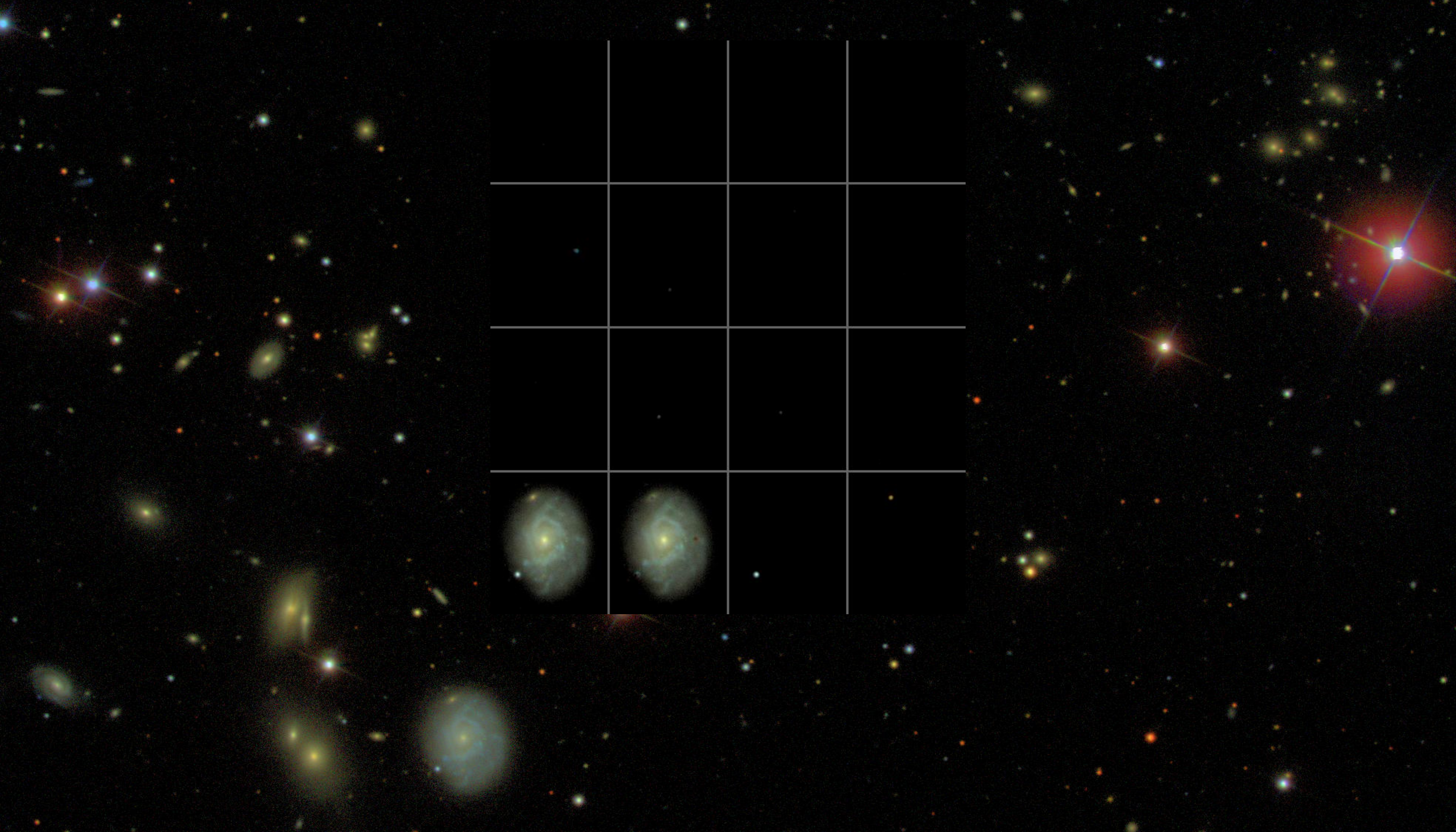
- Assume that we can write $I = \sum_r w_r T_r$, and solve for the weights in a least-squares sense.
- For each pixel with intensity I_i, share the flux between the children:

$$C_{r,i} = \frac{w_r T_r}{\sum w_r T_r} I_i$$

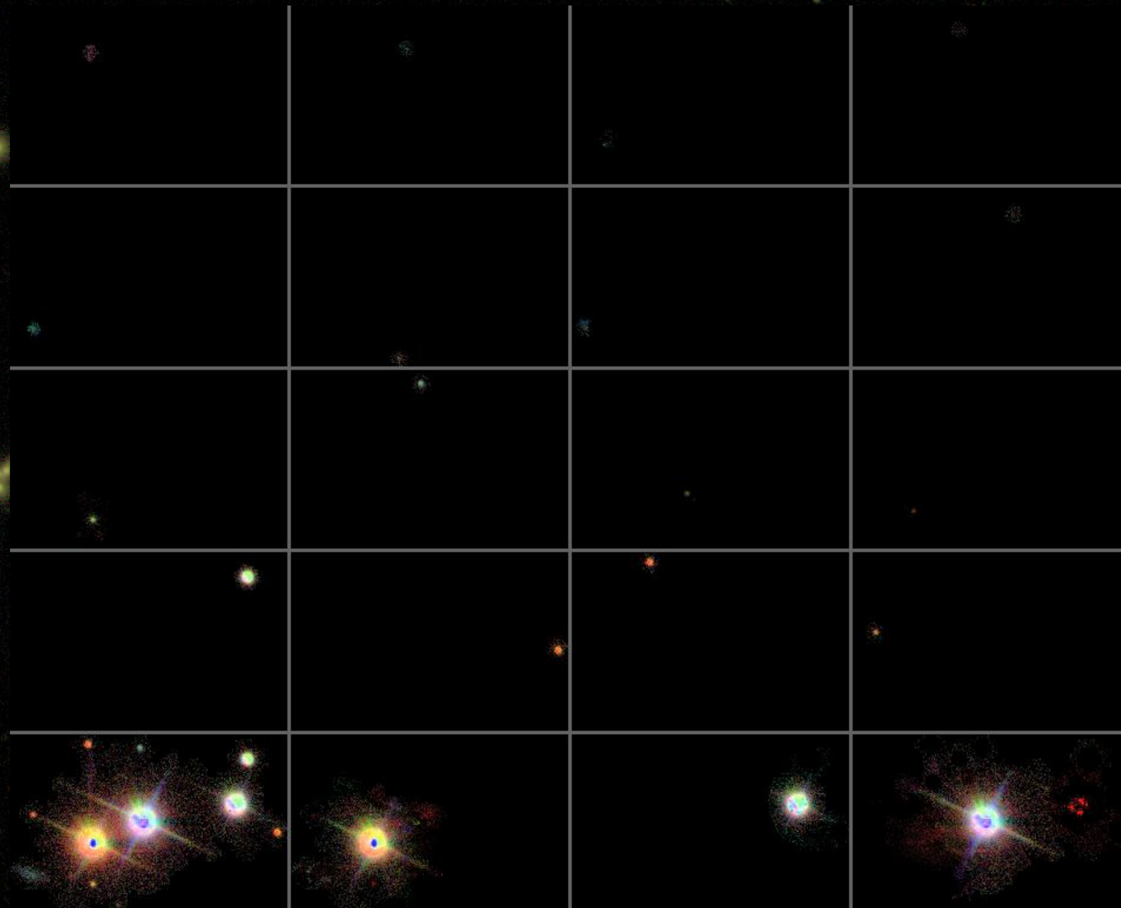


Does this work in practice?

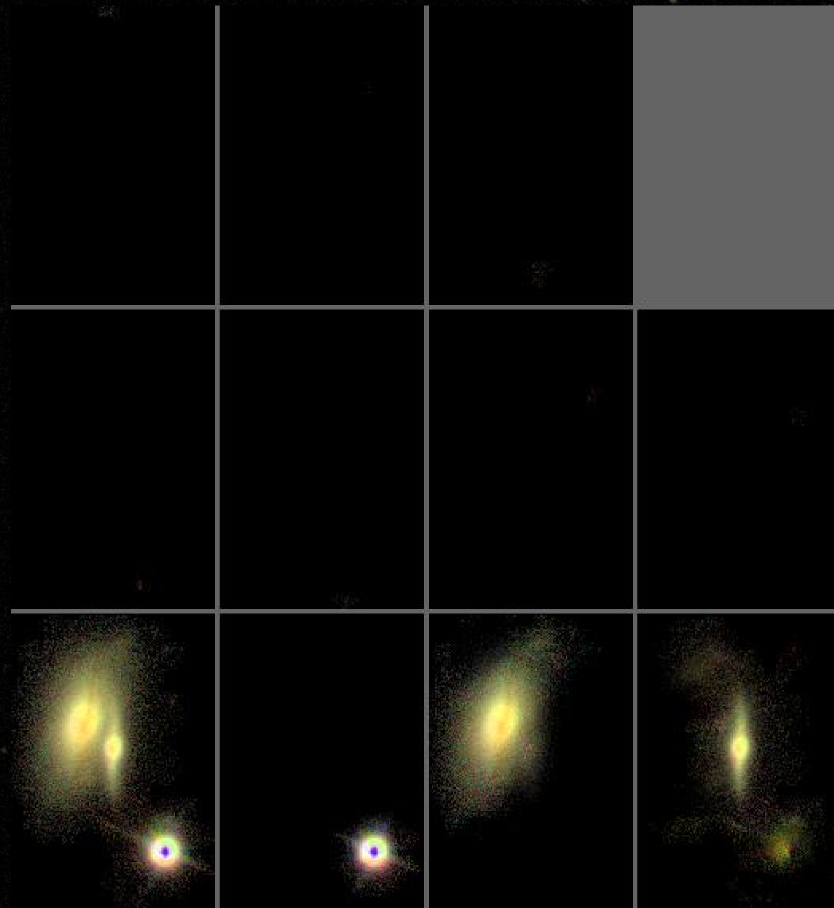
Does this work in practice?



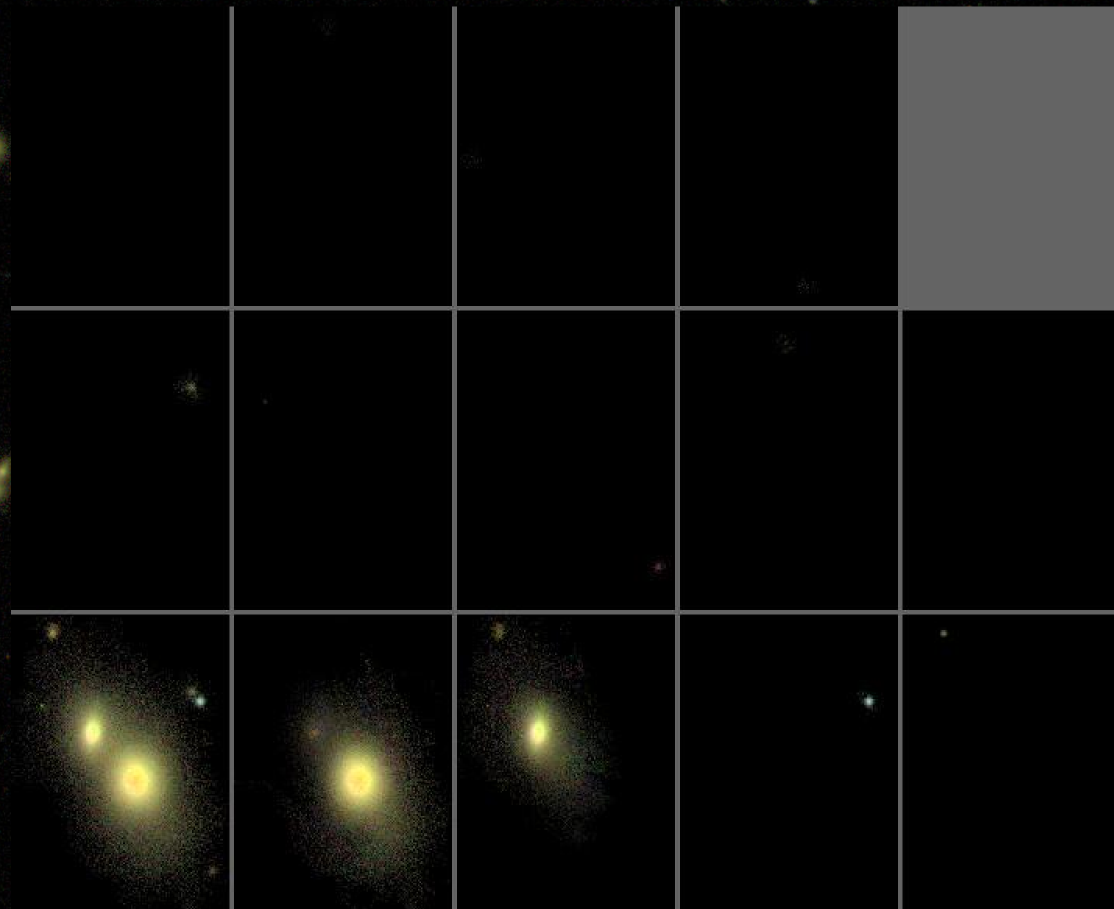
Does this work in practice?



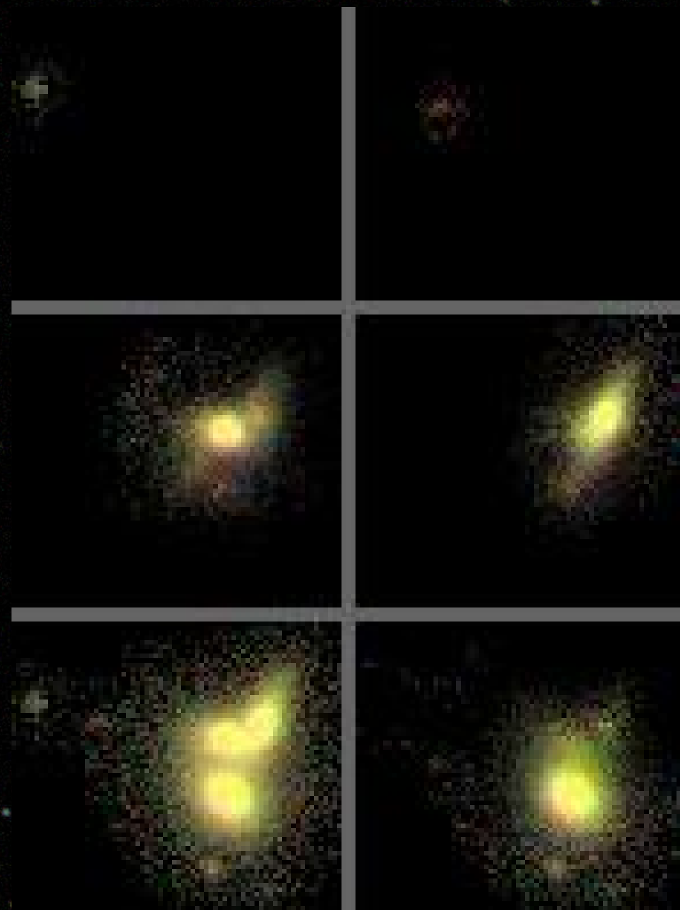
Does this work in practice?



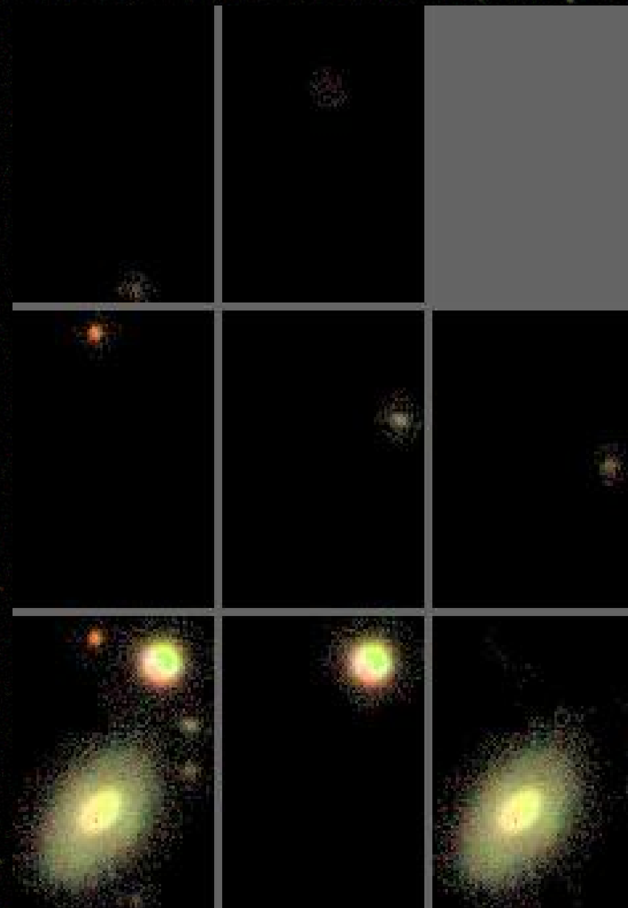
Does this work in practice?



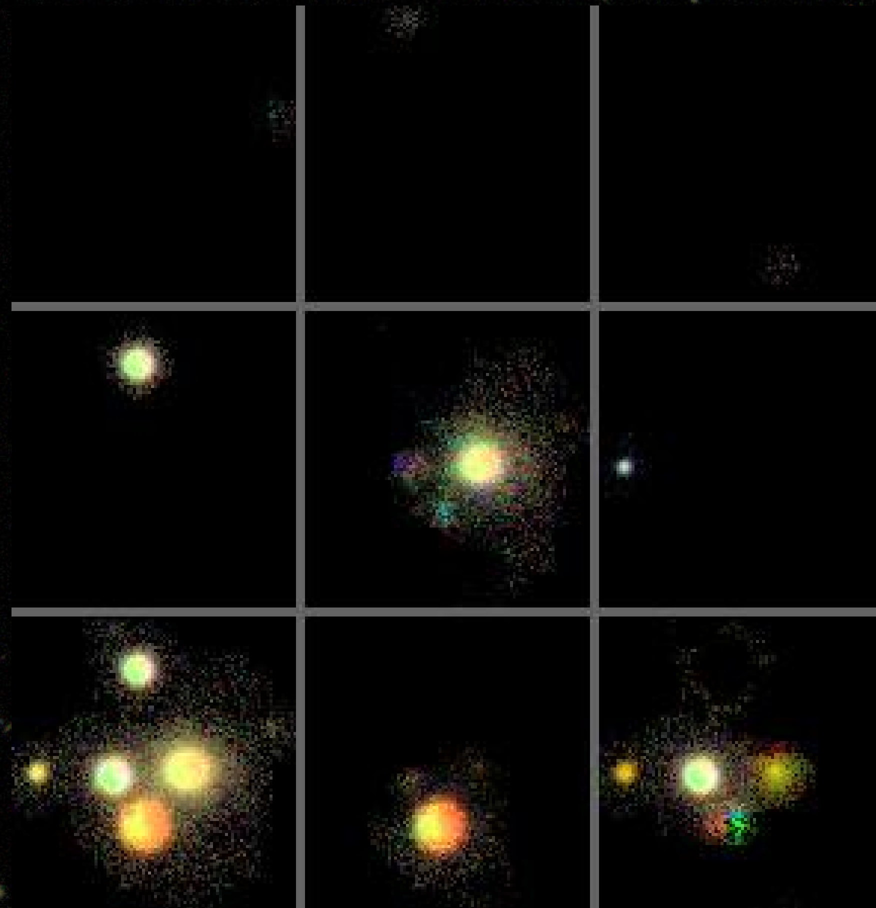
Does this work in practice?



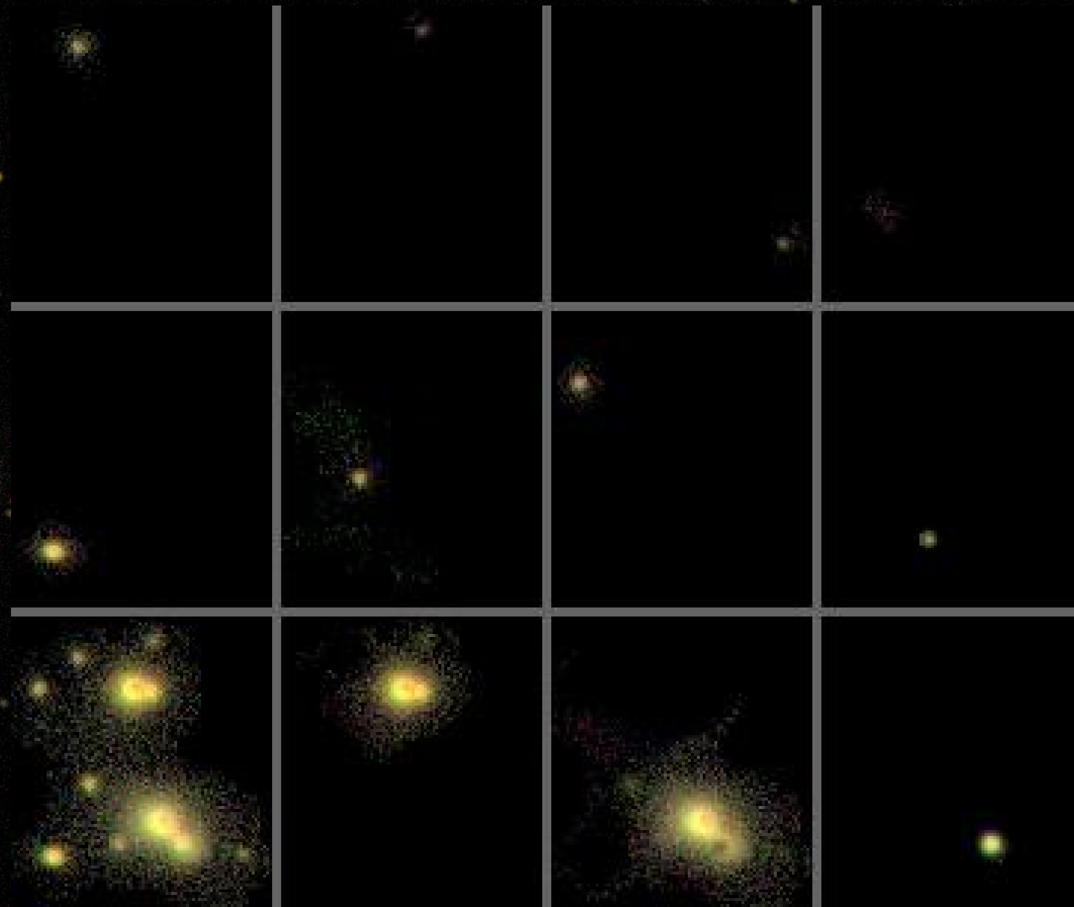
Does this work in practice?



Does this work in practice?



Does this work in practice?



Have you told us everything?

Have you told us everything?

For NGC galaxies, the fractions of various outcomes were:

very good	210	49.4%
good	146	34.4%
fair	50	11.8%
bad	7	1.6%
shredded	12	2.8%

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks
  - Matching peaks in different bands

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks
  - Matching peaks in different bands
- Star/Galaxy separation

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks
  - Matching peaks in different bands
- Star/Galaxy separation
- Moving objects

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks
  - Matching peaks in different bands
- Star/Galaxy separation
- Moving objects
- Degenerate templates

This algorithm is simple but unfortunately the real sky isn't.

Complexities include:

- SDSS takes data in 5 bands, and they need to be handled consistently
- The algorithm is based on peaks. This is tricky:
  - Finding close pairs of peaks
  - Rejecting non-significant peaks
  - Matching peaks in different bands
- Star/Galaxy separation
- Moving objects
- Degenerate templates
- Multi-peaked template

How should we measure Objects' Brightnesses?

How should we measure Objects' Brightnesses?

Stellar Photometry

How should we measure Objects' Brightnesses?

Stellar Photometry

Measuring an isolated star's brightness is reasonably easy; we can write down a model and find the ML estimate of the total flux:

$$I = S + F\delta \otimes \phi + n$$

(I : observed intensity; S : sky level; δ: delta-function; F: flux in star; ϕ : PSF; n : noise)

How should we measure Objects' Brightnesses?

Stellar Photometry

Measuring an isolated star's brightness is reasonably easy; we can write down a model and find the ML estimate of the total flux:

$$I = S + F\delta \otimes \phi + n$$

(I : observed intensity; S : sky level; δ: delta-function; F: flux in star; ϕ : PSF; n : noise)

As an alternative, we can measure the total flux contained within some aperture of radius R; as R is increased the details of the PSF matter less, but the noise in the measurements increase, and so does the importance of accurate an measurement of the sky level, S.

Sky Determination

What is meant by the sky level? It's the *mean* value of the contribution of everything that we haven't explicitly allowed for: atmospheric emission, zodiacal dust, CCD dark current, *and* unresolved sources.

Sky Determination

What is meant by the sky level? It's the *mean* value of the contribution of everything that we haven't explicitly allowed for: atmospheric emission, zodiacal dust, CCD dark current, *and* unresolved sources.

It isn't clear how to measure this, and simply making a robust estimate of the mode or median of the intensity distribution is certainly not correct — although it may be good enough.

Galaxy Photometry

Unfortunately, Galaxies belong to no such 1-parameter family; the simplest even plausibly appropriate model would be:

$$I = S + F[f_D D(Ie_D, re_D, ab_D, \alpha_D) + \\ (1 - f_D) E(Ie_E, re_E, ab_E, \alpha_E)] \otimes \phi + n$$

where D and E are a pure deVaucouleurs bulge and a pure exponential disk respectively.

Galaxy Photometry

Unfortunately, Galaxies belong to no such 1-parameter family; the simplest even plausibly appropriate model would be:

$$I = S + F[f_D D(Ie_D, re_D, ab_D, \alpha_D) + \\ (1 - f_D)E(Ie_E, re_E, ab_E, \alpha_E)] \otimes \phi + n$$

where D and E are a pure deVaucouleurs bulge and a pure exponential disk respectively.

Rather than face this difficulty, astronomers have traditionally defined a number of less-efficient measures that include some well-understood fraction of the total flux.

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

An aperture measure; dependent on $(1+z)^4$ dimming and (weakly) on seeing

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

An aperture measure; dependent on $(1+z)^4$ dimming and (weakly) on seeing

Kron flux

The flux enclosed within a aperture of radius αr_K where

$r_K \equiv \frac{\int_{\in A} rI \, 2\pi r \, dr}{\int_{\in A} I \, 2\pi r \, dr}$ for some choice of A

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

An aperture measure; dependent on $(1+z)^4$ dimming and (weakly) on seeing

Kron flux

The flux enclosed within a aperture of radius αr_K where

$r_K \equiv \frac{\int_{\in A} r I \, 2\pi r \, dr}{\int_{\in A} I \, 2\pi r \, dr}$ for some choice of A

An aperture measure; dependent on seeing and A (the noise diverges as $A \to \infty$)

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

An aperture measure; dependent on $(1+z)^4$ dimming and (weakly) on seeing

Kron flux

The flux enclosed within a aperture of radius αr_K where

$r_K \equiv \frac{\int_{\in A} r I \, 2\pi r \, dr}{\int_{\in A} I \, 2\pi r \, dr}$ for some choice of A

An aperture measure; dependent on seeing and A (the noise diverges as $A \to \infty$)

Petrosian flux

The flux enclosed within a aperture of radius $f_2 r_P$ where

$$f_1 = \frac{\int_0^{R_P} I \, 2\pi r \, dr}{\pi {R_P}^2}$$

Isophotal flux

The flux enclosed within a aperture defined by $I > I_0$.

An aperture measure; dependent on $(1+z)^4$ dimming and (weakly) on seeing

Kron flux

The flux enclosed within a aperture of radius αr_K where

$r_K \equiv \frac{\int_{\in A} rI\,2\pi r\,dr}{\int_{\in A} I\,2\pi r\,dr}$ for some choice of A

An aperture measure; dependent on seeing and A (the noise diverges as $A \to \infty$)

Petrosian flux

The flux enclosed within a aperture of radius $f_2 r_P$ where

$$f_1 = \frac{\int_0^{R_P} I\,2\pi r\,dr}{\pi {R_P}^2}$$

An aperture measure; weakly dependent on seeing.

But are Model Fluxes Totally Impractical?

But are Model Fluxes Totally Impractical?

I claimed that the simplest plausible model was:

$$I = S + F[f_D D(Ie_D, re_D, ab_D, \alpha_D) + (1 - f_D) E(Ie_E, re_E, ab_E, \alpha_E)] \otimes \phi + n$$

But what if I ignore this claim in the interest of computational efficiency?

But are Model Fluxes Totally Impractical?

I claimed that the simplest plausible model was:

$$I = S + F[f_D D(Ie_D, re_D, ab_D, \alpha_D) + (1 - f_D) E(Ie_E, re_E, ab_E, \alpha_E)] \otimes \phi + n$$

But what if I ignore this claim in the interest of computational efficiency?

The *entire* per-band processing for an object in the SDSS takes about 15ms on a 1GHz processor.

But are Model Fluxes Totally Impractical?

I claimed that the simplest plausible model was:

$$I = S + F[f_D D(Ie_D, re_D, ab_D, \alpha_D) + \\ (1 - f_D) E(Ie_E, re_E, ab_E, \alpha_E)] \otimes \phi + n$$

But what if I ignore this claim in the interest of computational efficiency?

The *entire* per-band processing for an object in the SDSS takes about 15ms on a 1GHz processor.

So let us consider a one-component model:

$$I = S + F[M(Ie_M, re_M, ab_M, \alpha_M)] \otimes \phi + n$$

$M \equiv D$ (deVaucouleurs profile: $I \sim exp(-r^{-1/4})$) or E (exponential: $I \sim exp(-r)$)

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;
- convolving with the PSF at that point in the frame;

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;
- convolving with the PSF at that point in the frame;
- and finally determining the value of χ^2 for that model by summing over all the pixels in the object.

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;
- convolving with the PSF at that point in the frame;
- and finally determining the value of χ^2 for that model by summing over all the pixels in the object.

Fitting these models is a straightforward χ^2 minimisation problem, which I solve using the standard Levenberg-Marquardt algorithm.

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;
- convolving with the PSF at that point in the frame;
- and finally determining the value of χ^2 for that model by summing over all the pixels in the object.

Fitting these models is a straightforward χ^2 minimisation problem, which I solve using the standard Levenberg-Marquardt algorithm.

A naïve implementation is impractically slow; but tricks can be invented (basically, pre-compute *everything*).

Solving for the model parameters is a three-dimensional non-linear optimisation problem. Each function evaluation requires

- building a model galaxy;
- convolving with the PSF at that point in the frame;
- and finally determining the value of χ^2 for that model by summing over all the pixels in the object.

Fitting these models is a straightforward χ^2 minimisation problem, which I solve using the standard Levenberg-Marquardt algorithm.

A naïve implementation is impractically slow; but tricks can be invented (basically, pre-compute *everything*).

Using a continuum method for data given on a grid may not be a smart idea. I'll take suggestions from the audience.

Model Magnitudes

Once I know $(I_0, r_e, a/b)$ for a model of a given class (exponential or deVaucouleurs) I can easily calculate the total flux.

Model Magnitudes

Once I know $(I_0, r_e, a/b)$ for a model of a given class (exponential or deVaucouleurs) I can easily calculate the total flux.

The meaning of the 'total flux' isn't entirely clear; for deVaucouleurs profiles ten percent of the flux can be so far out that the enclosed signal-to-noise is less than unity.

Model Magnitudes

Once I know $(I_0, r_e, a/b)$ for a model of a given class (exponential or deVaucouleurs) I can easily calculate the total flux.

The meaning of the 'total flux' isn't entirely clear; for deVaucouleurs profiles ten percent of the flux can be so far out that the enclosed signal-to-noise is less than unity.

If the object is a star I'd like the 'model' magnitude to equal that measured in any other way, which implies that I need an aperture correction.

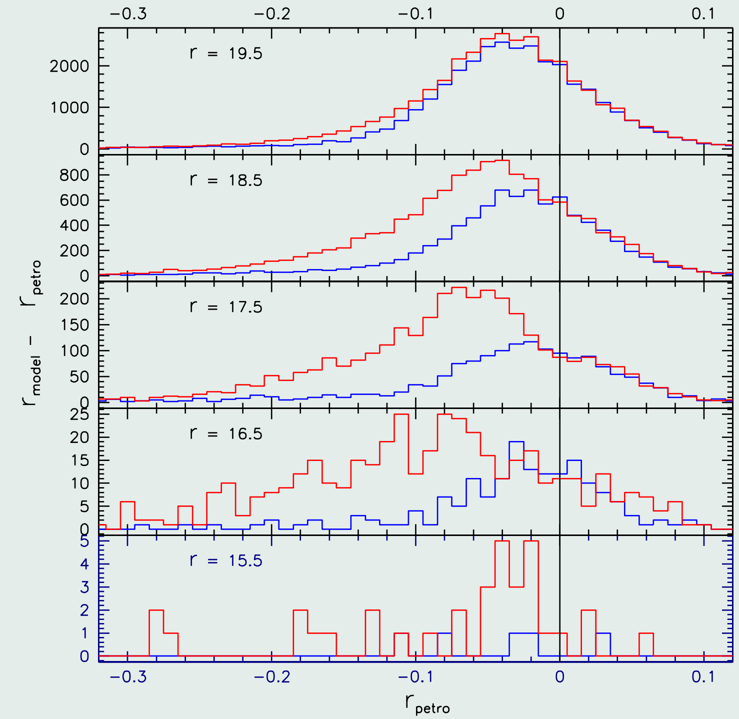
Model Magnitudes

Once I know $(I_0, r_e, a/b)$ for a model of a given class (exponential or deVaucouleurs) I can easily calculate the total flux.

The meaning of the 'total flux' isn't entirely clear; for deVaucouleurs profiles ten percent of the flux can be so far out that the enclosed signal-to-noise is less than unity.

If the object is a star I'd like the 'model' magnitude to equal that measured in any other way, which implies that I need an aperture correction.

I calculate this by fitting our galaxy models to the known (KL) PSF at various places across the field, and estimating an appropriate aperture correction. These are (now) small $(1.016 + -0.007)$.



The comparison between Petrosian and Model magnitudes; red objects have $u - r < 2.2$ (Strateva et al.).

But do Galaxies really have pure deVaucouleurs or Exponential profiles?

But do Galaxies really have pure deVaucouleurs or Exponential profiles?

No

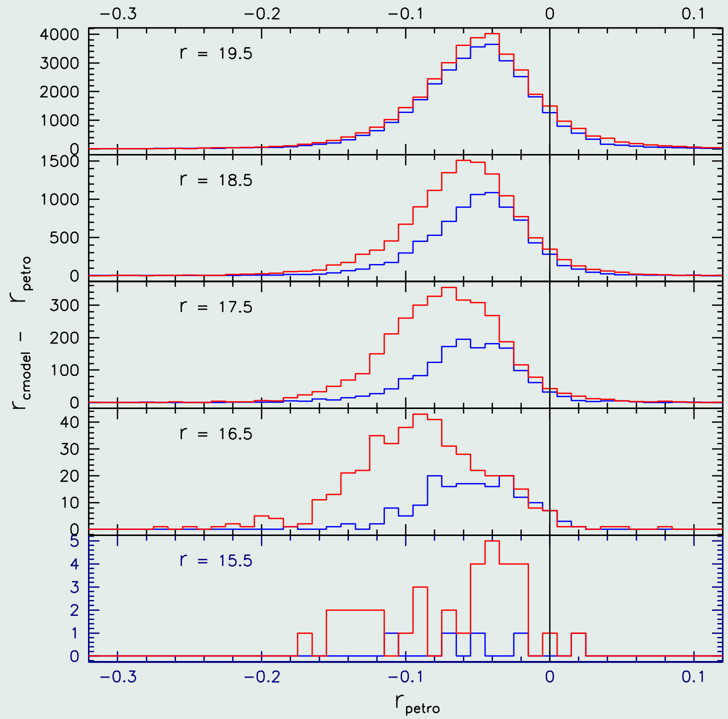
But do Galaxies really have pure deVaucouleurs or Exponential profiles?

...but I have less than 15ms available

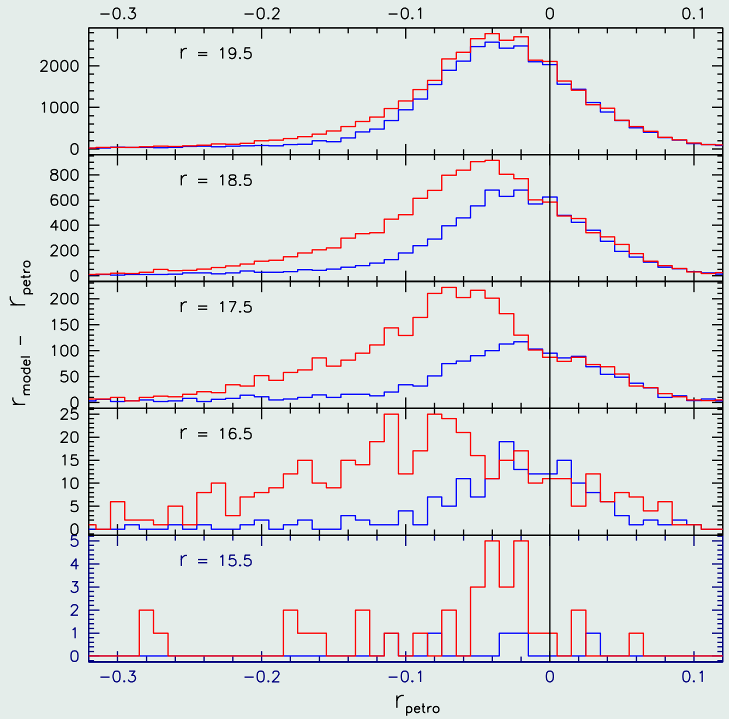
But do Galaxies really have pure deVaucouleurs or Exponential profiles?

So fit a *linear* combination of the best (non-linear) deV and exp models; I refer to this as a *composite-model (cmodel) magnitude:*

$$F_{\mathit{cmodel}} = f_{deV} F_{\mathit{deV}} + (1 - f_{exp}) F_{\mathit{exp}}$$



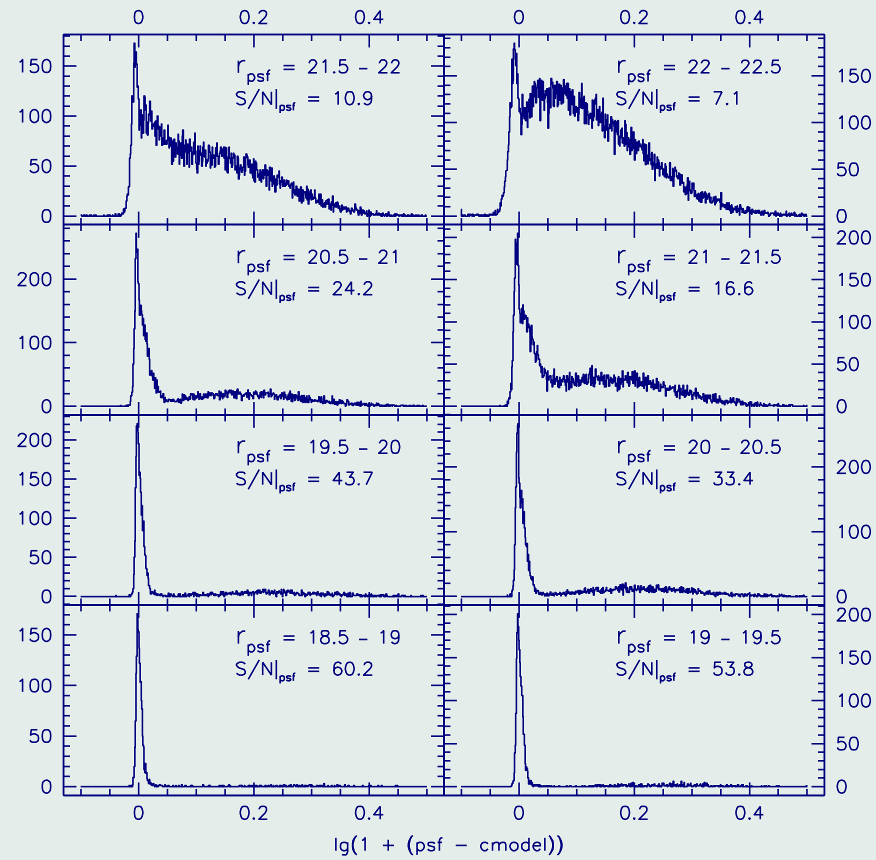
The comparison between Petrosian and Composite-Model magnitudes; red objects have $u - r < 2.2$ (Strateva et al.).



The comparison between Petrosian and Model magnitudes; red objects have $u - r < 2.2$ (Strateva et al.).

We can now ask, 'Is this f_{deV} good for anything else?'

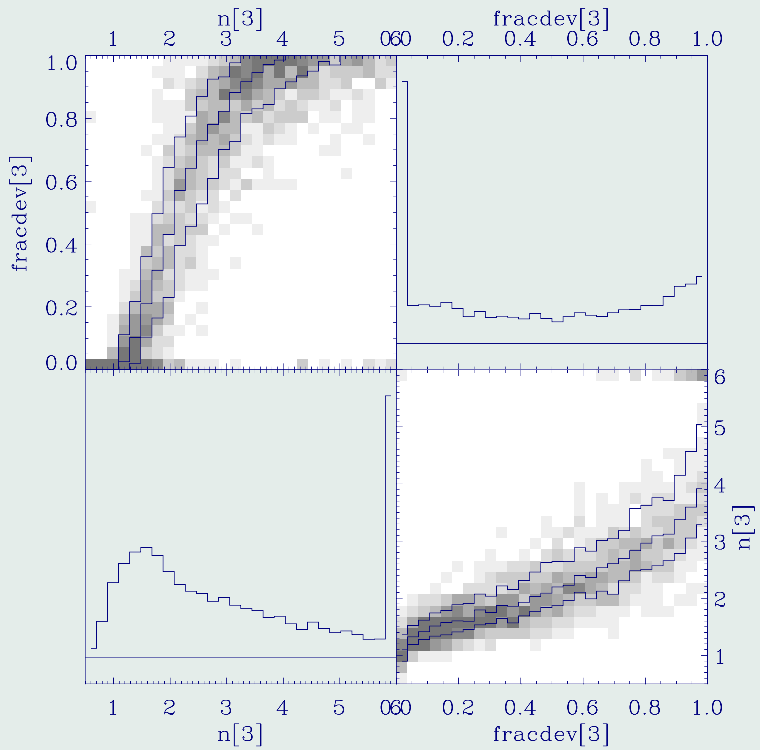
We can now ask, 'Is this f_{deV} good for anything else?'



Star-Galaxy Separation

How about structural parameters?

How about structural parameters?



Galaxies with $r < 18$; values shown are for i band.

'fracdev' is what I have called f_D; n is the Sersic index.

(Plot courtesy of Michael Blanton)

How Should we Represent Multi-Colour CCD Data?

How Should we Represent Multi-Colour CCD Data?

Preparing RGB Images from CCD Data

I want a mapping to the range $[0, 1]$ *for each of three colors red (R), green (G), and blue (B).*

Preparing RGB Images from CCD Data

I want a mapping to the range $[0, 1]$ *for each of three colors red (R), green (G), and blue (B).*

The usual algorithm is:

$$R = f(r);\ G = f(g);\ B = f(b)$$

where

$$f(x) = \begin{cases} 0, & x < m; \\ F(x-m)/F(M-m) & m \le x \le M; \\ 1 & M < x. \end{cases}$$

and m *is the minimum value to display, and* M *the maximum.*

Preparing RGB Images from CCD Data

I want a mapping to the range $[0,1]$ *for each of three colors red (R), green (G), and blue (B).*

The usual algorithm is:

$$R = f(r);\ G = f(g);\ B = f(b)$$

where

$$f(x) = \begin{cases} 0, & x < m; \\ F(x-m)/F(M-m) & m \le x \le M; \\ 1 & M < x. \end{cases}$$

and m *is the minimum value to display, and* M *the maximum.*

Note that there is no unique mapping from $(r/g, i/g)$ *to* $(R/G, I/G)$*.*

An (preferable) alternative is to define $I \equiv (r+g+b)/3$*, and set*

$$\begin{aligned} R &= r * f(I)/I \\ G &= g * f(I)/I \\ B &= b * f(I)/I \end{aligned}$$

An (preferable) alternative is to define $I \equiv (r+g+b)/3$*, and set*

$$R = r * f(I)/I$$
$$G = g * f(I)/I$$
$$B = b * f(I)/I$$

Note that now $r/g \equiv R/G$ *and* $i/g \equiv I/G$*.*

An (preferable) alternative is to define $I \equiv (r+g+b)/3$*, and set*

$$\begin{aligned} R &= r * f(I)/I \\ G &= g * f(I)/I \\ B &= b * f(I)/I \end{aligned}$$

Note that now $r/g \equiv R/G$ *and* $i/g \equiv I/G$.

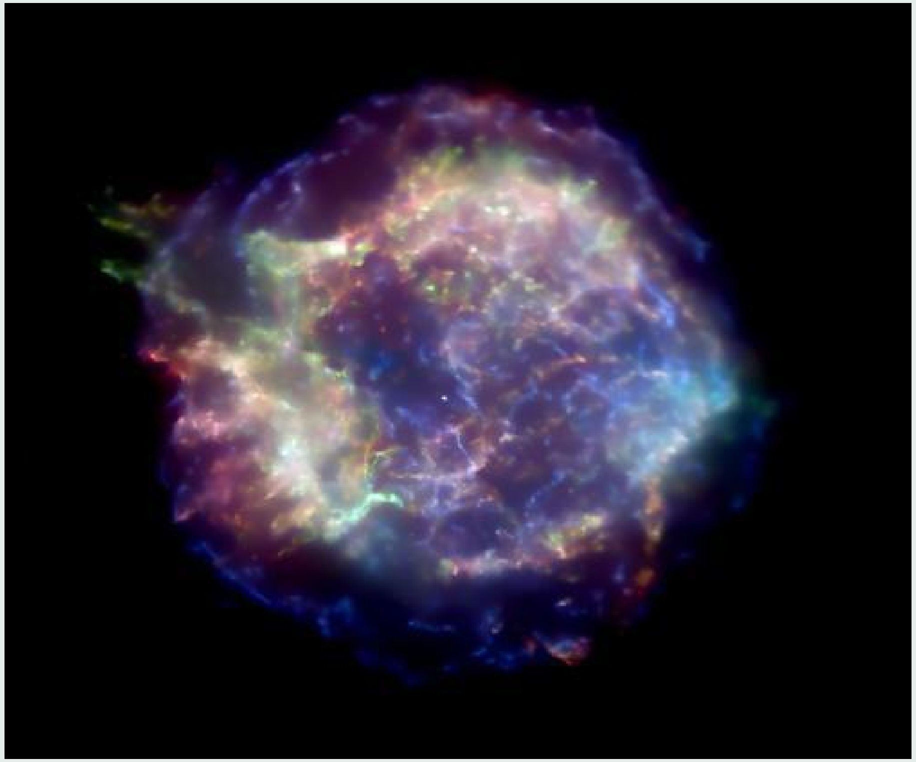
Additionally, it is possible to choose a more flexible functional form for F*; I like to take* $f(x) = asinh(\alpha Q(x-m))/Q$*, which allows the user to first set* $Q \to 0$ *and choose the linear stretch* α*, and then adjust* Q *to bring out brighter features.*

Let us apply this X-Ray data from Chandra

R: 0.3–1.55 keV; G: 1.55–3.34 keV; B: 3.34–10.0 keV

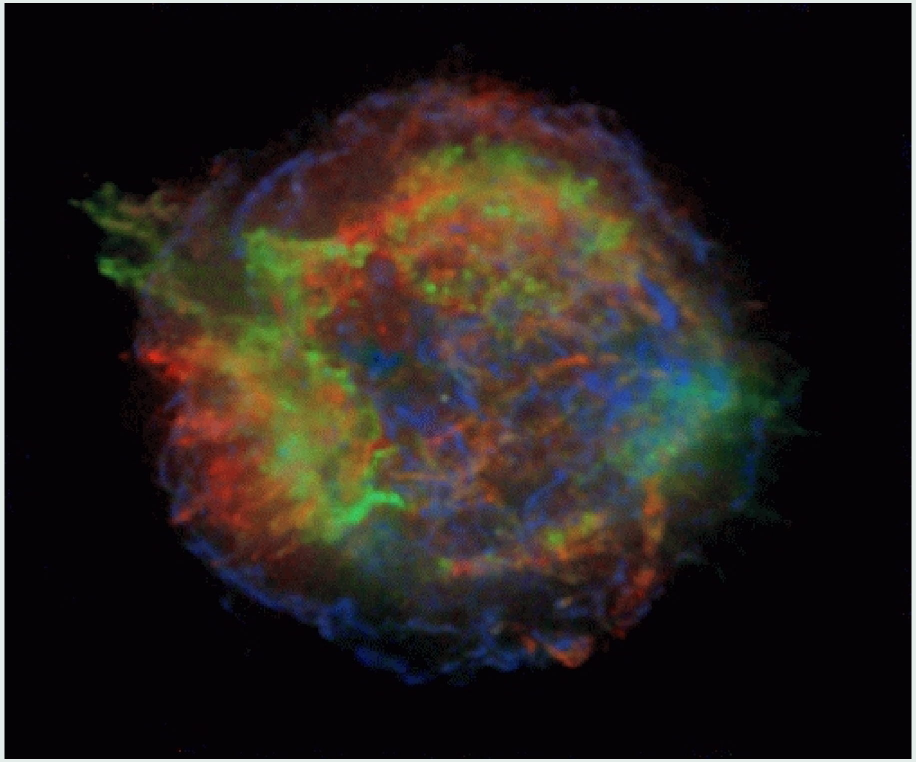
Let us apply this X-Ray data from Chandra

R: 0.3–1.55 keV; G: 1.55–3.34 keV; B: 3.34–10.0 keV



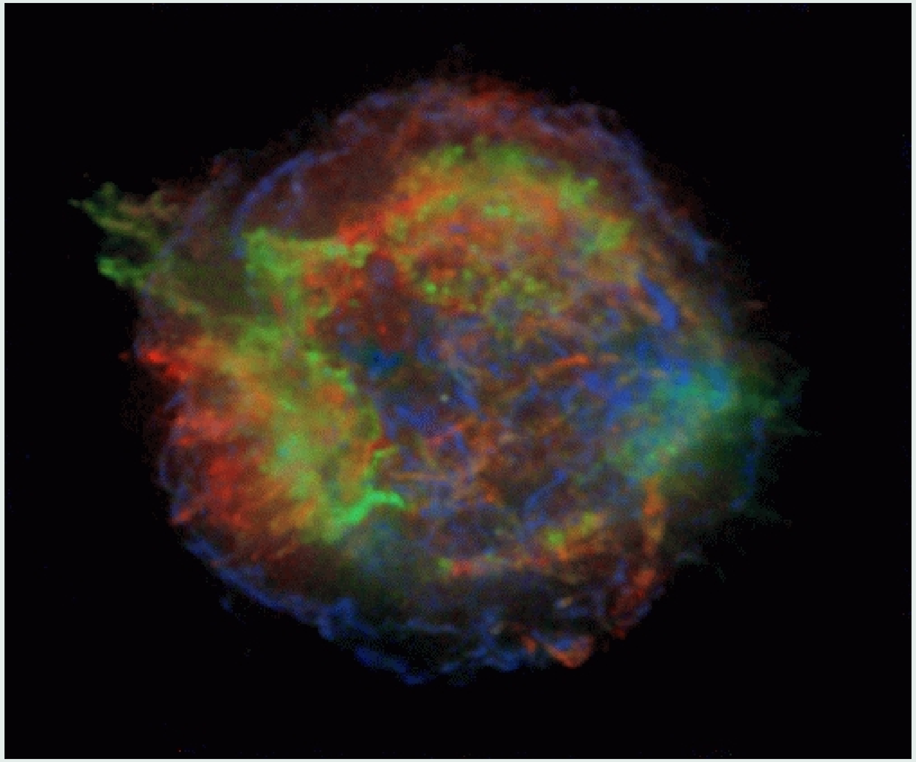
Let us apply this X-Ray data from Chandra

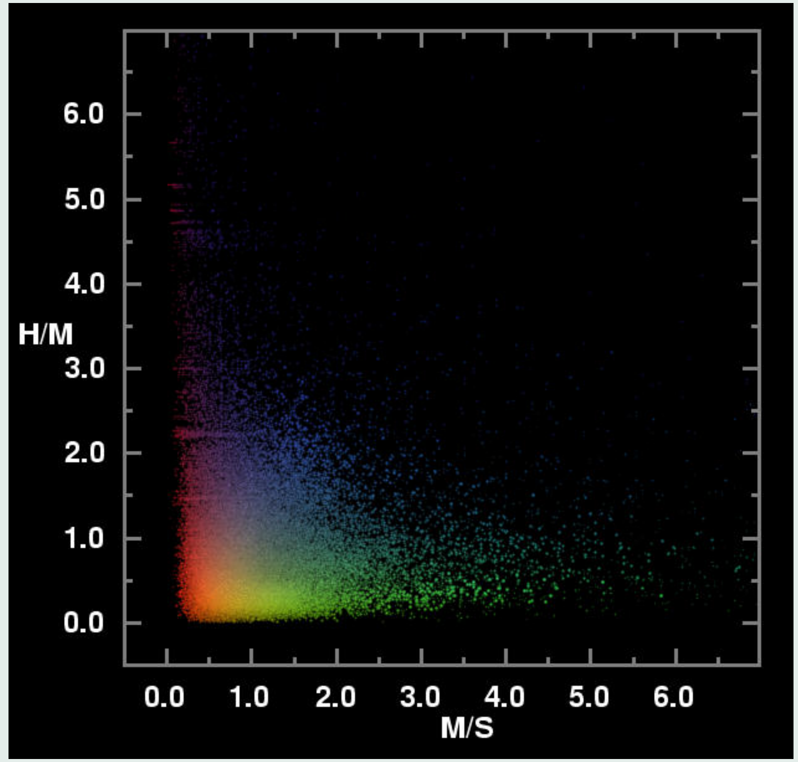
R: 0.3–1.55 keV; G: 1.55–3.34 keV; B: 3.34–10.0 keV



Let us apply this X-Ray data from Chandra

R: 0.3–1.55 keV; G: 1.55–3.34 keV; B: 3.34–10.0 keV



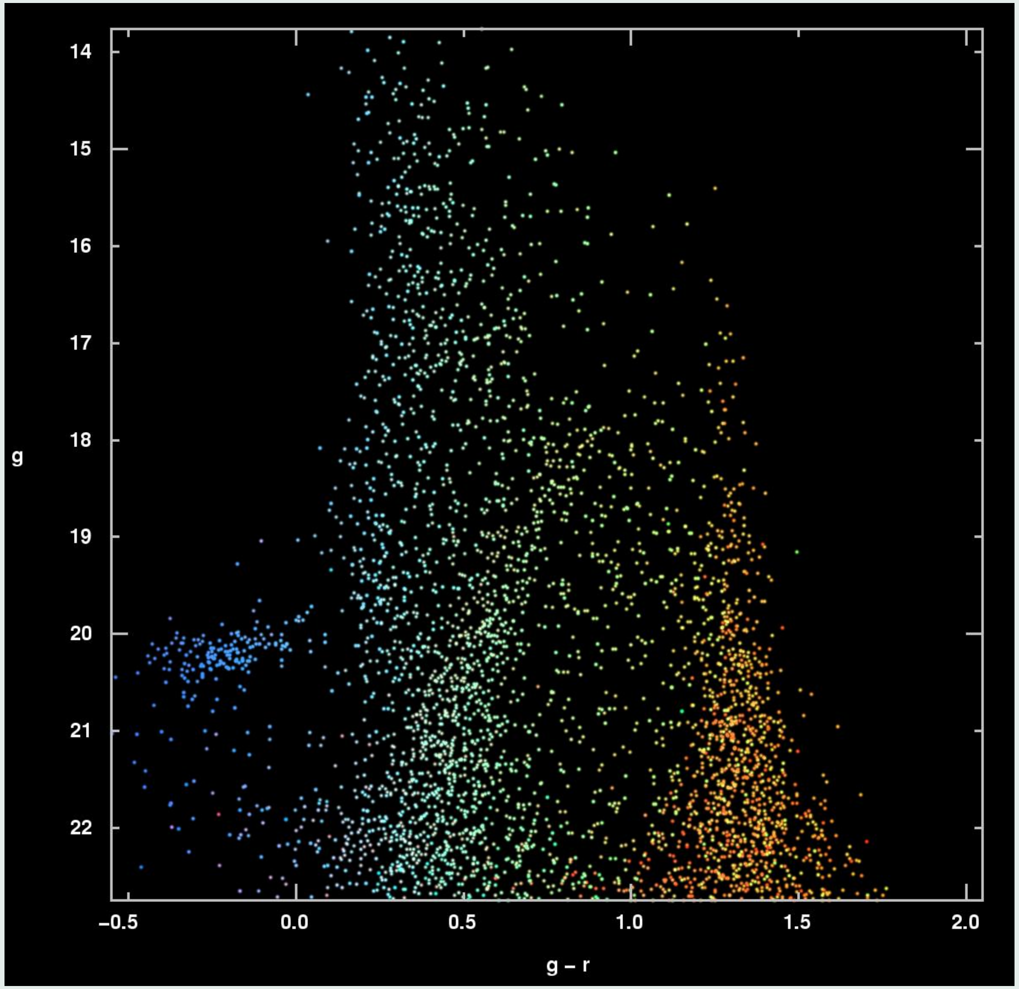


What about SDSS imaging of NGC2419?

What about SDSS imaging of NGC2419?

What about SDSS imaging of NGC2419?

What about SDSS imaging of NGC2419?



Conclusions

Conclusions

Deblending

A peak-based deblender with a symmetry ansatz and various tricks seems to perform well for a large range of astronomical objects

Conclusions

Deblending

A peak-based deblender with a symmetry ansatz and various tricks seems to perform well for a large range of astronomical objects

Galaxy Fluxes

It is possible to use model fits that more-or-less reproduce Petrosian magnitudes for bright galaxies, and are efficient at faint magnitudes.

Conclusions

Deblending

A peak-based deblender with a symmetry ansatz and various tricks seems to perform well for a large range of astronomical objects

Galaxy Fluxes

It is possible to use model fits that more-or-less reproduce Petrosian magnitudes for bright galaxies, and are efficient at faint magnitudes.

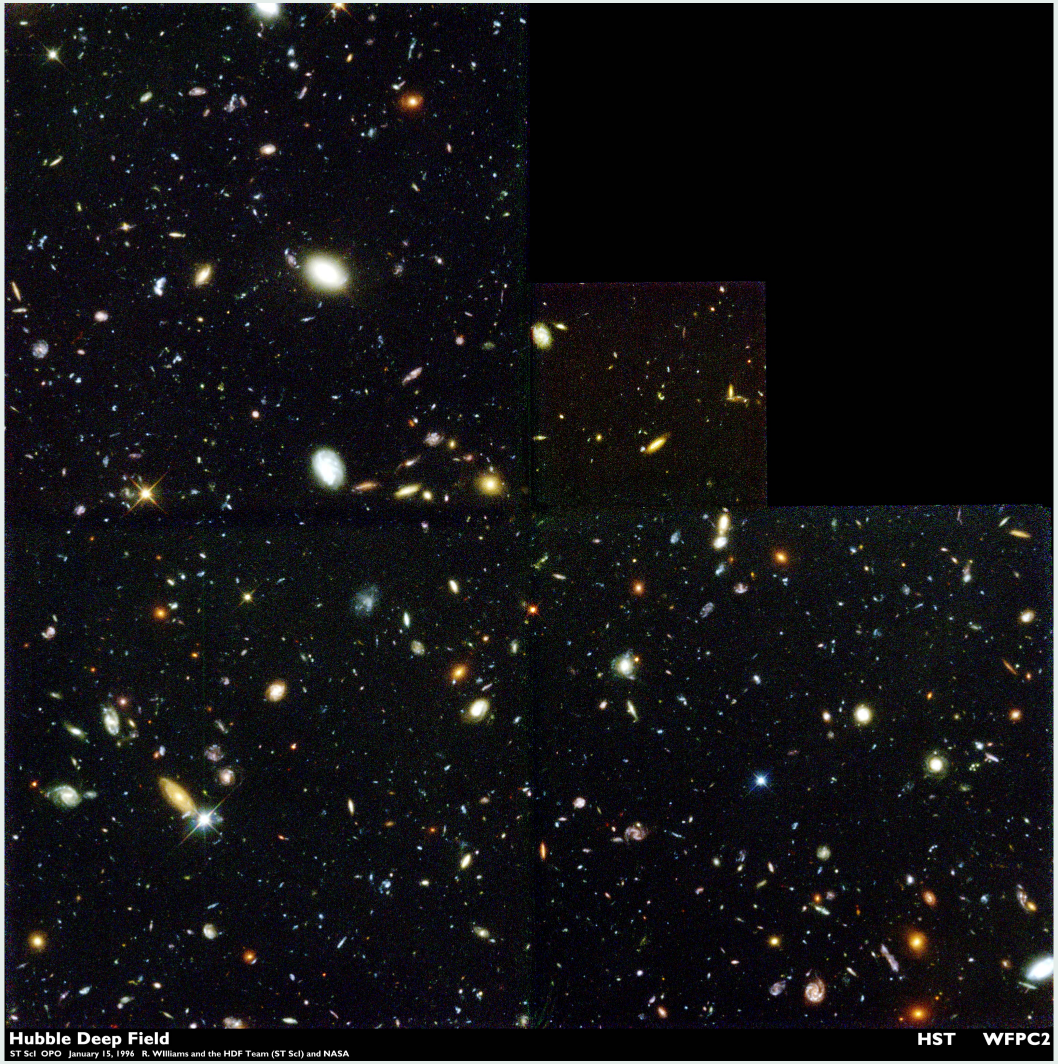
Coloured Images

There is way to uniquely map flux ratios to perceived colours; this is valuable.

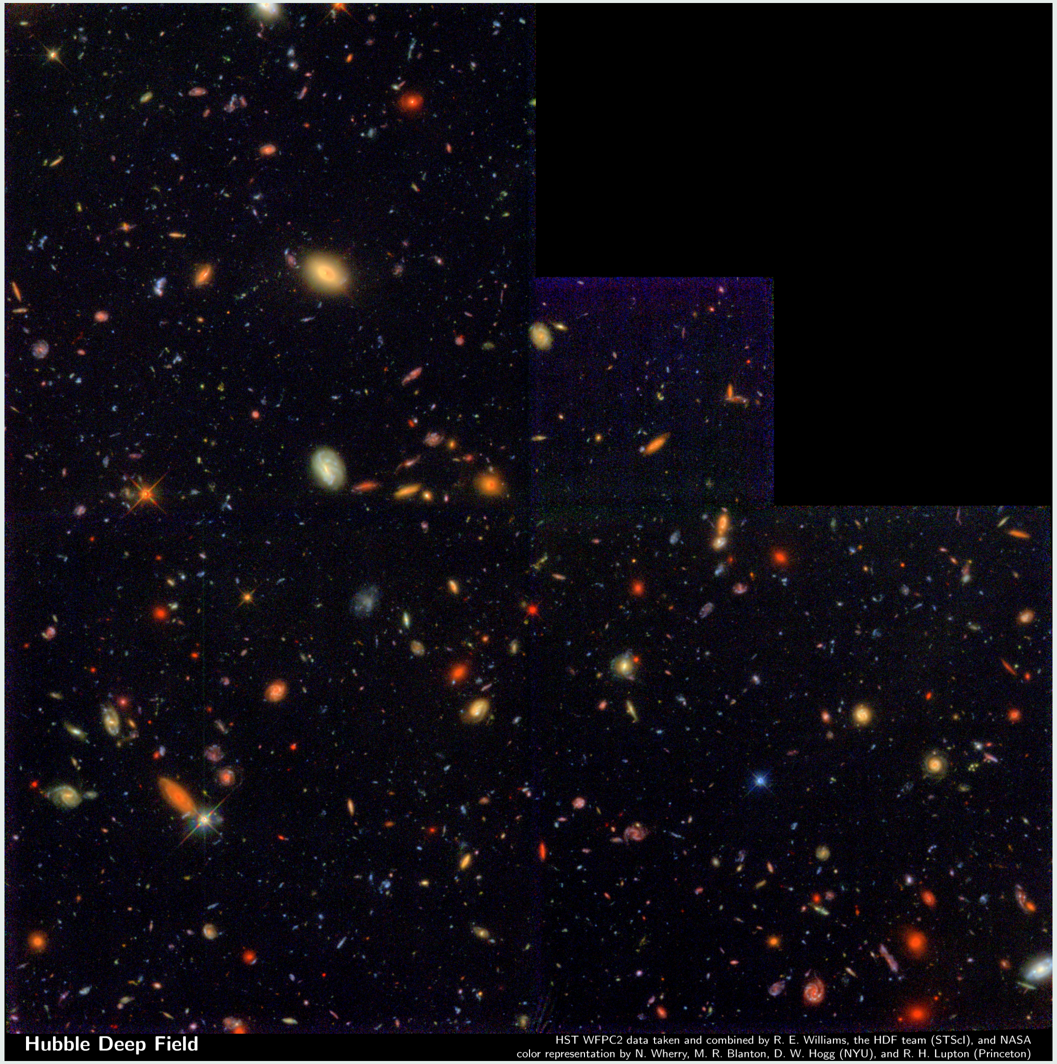

The End

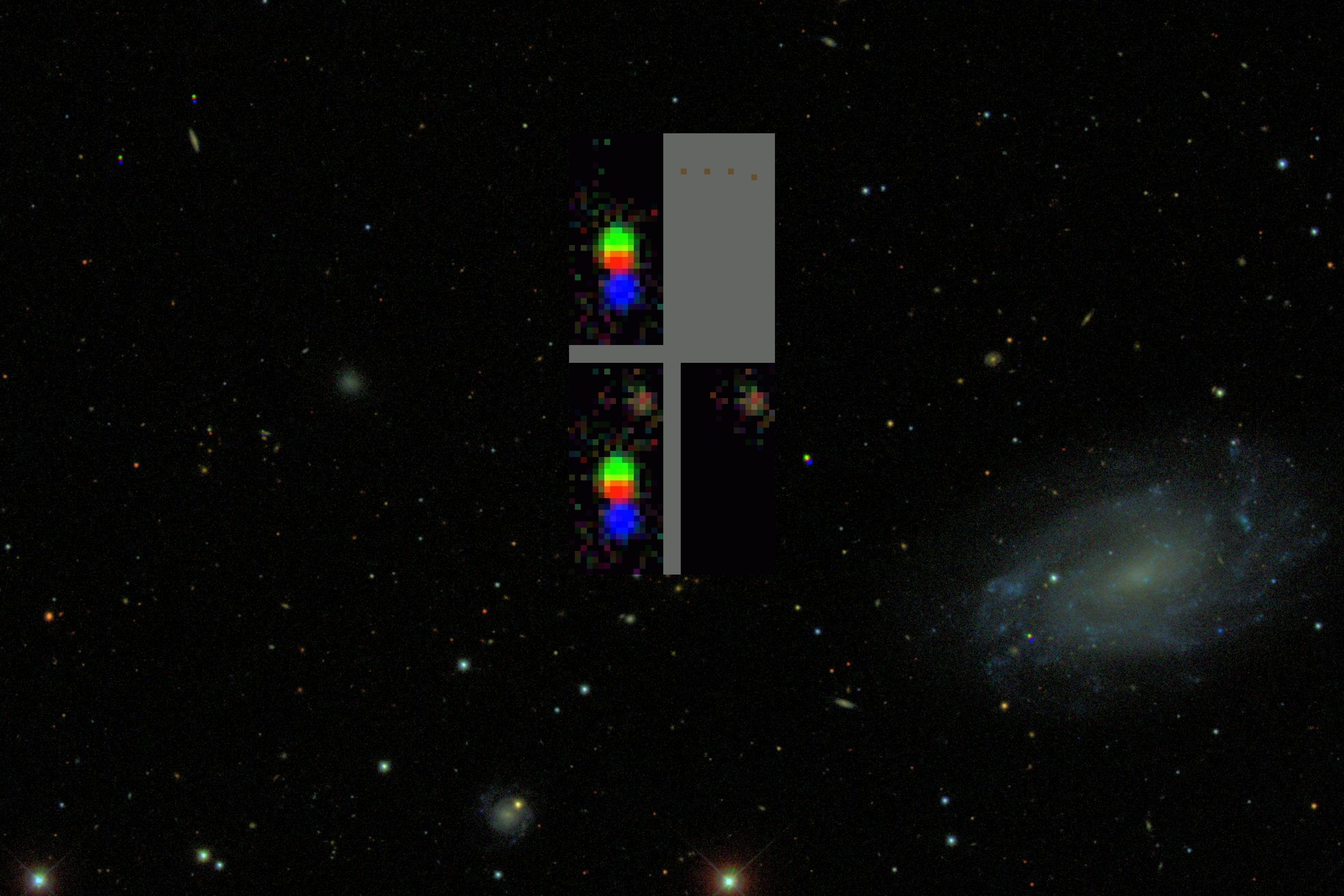
What does the Hubble Deep Field Look like in Colour?

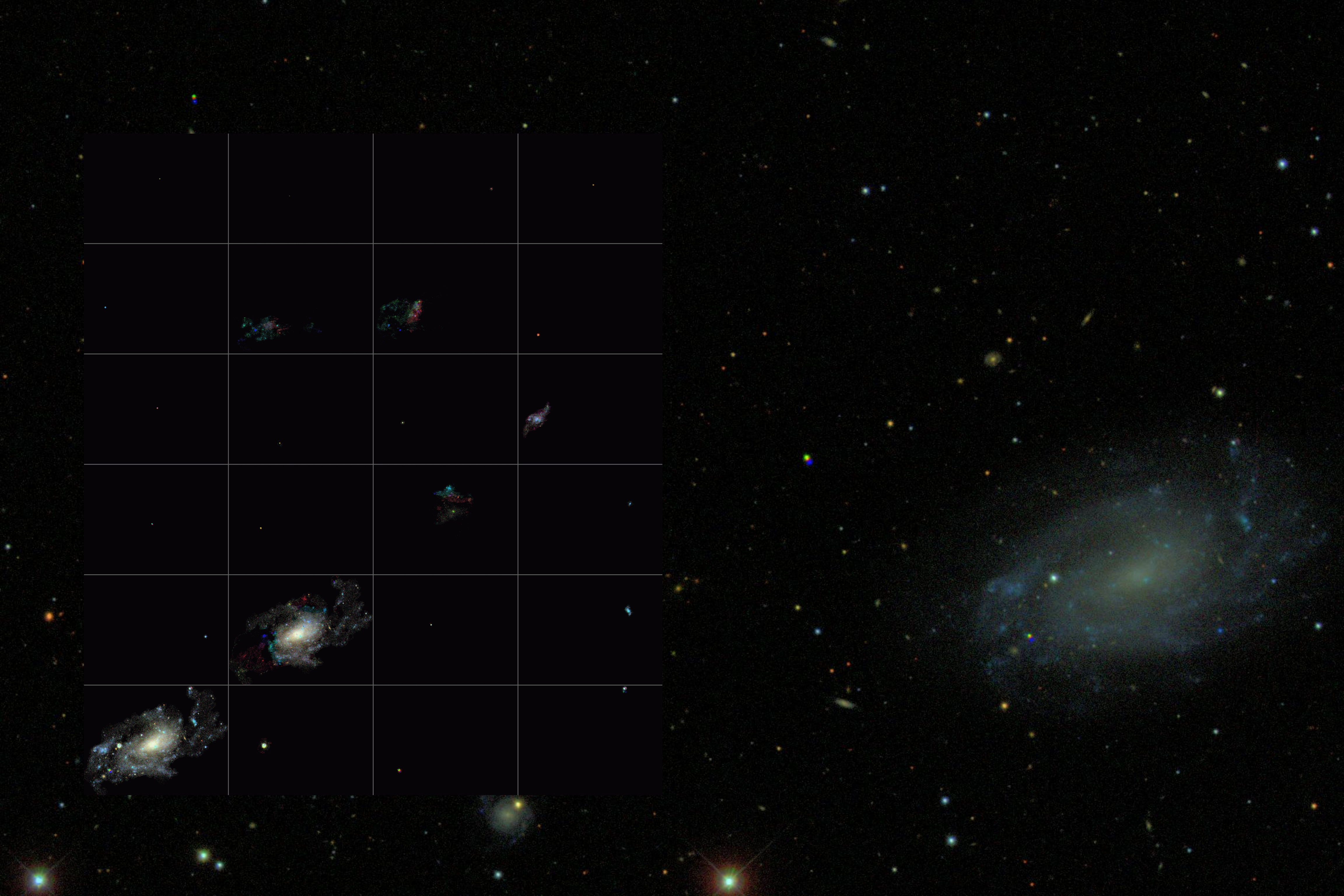
What does the Hubble Deep Field Look like in Colour?

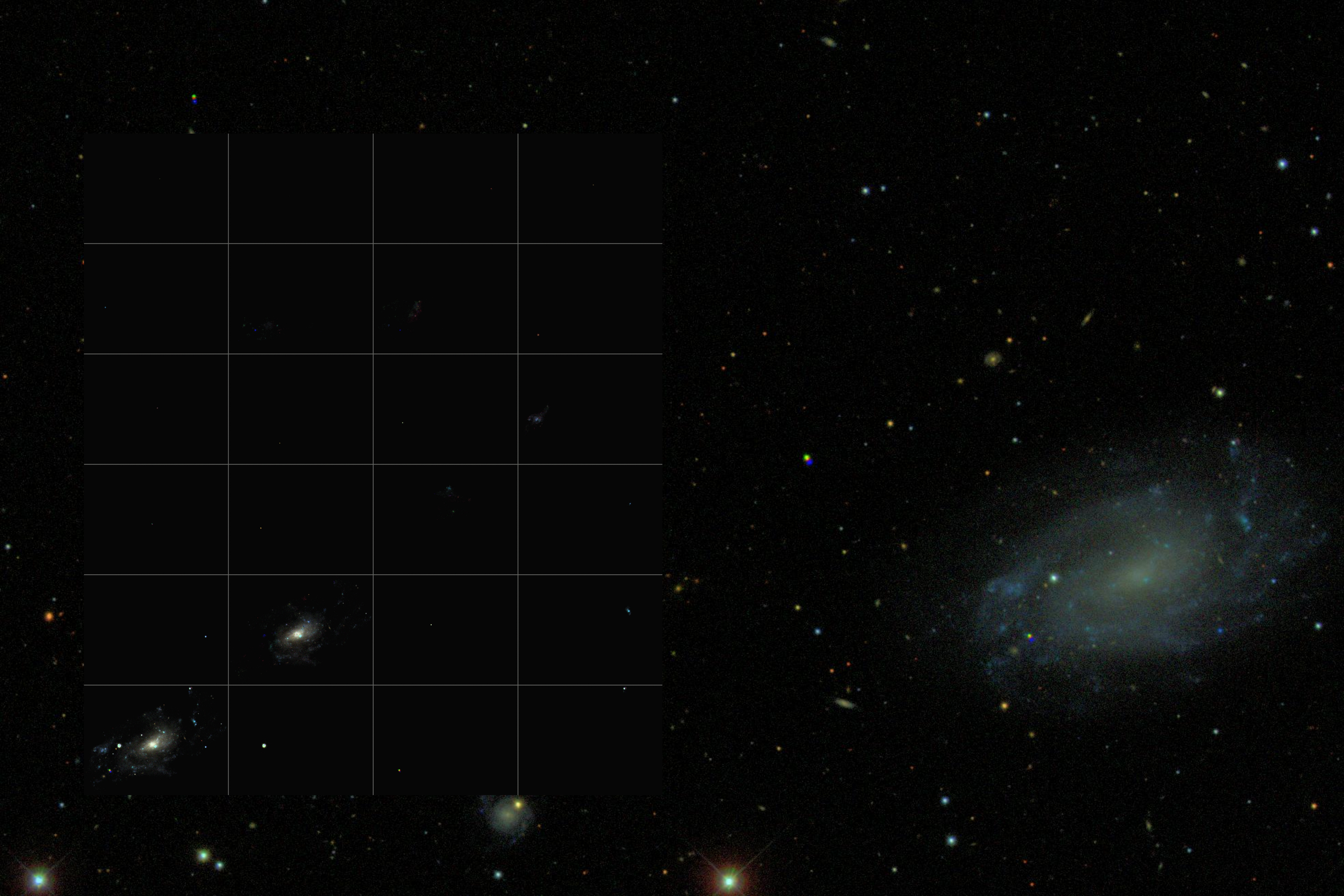


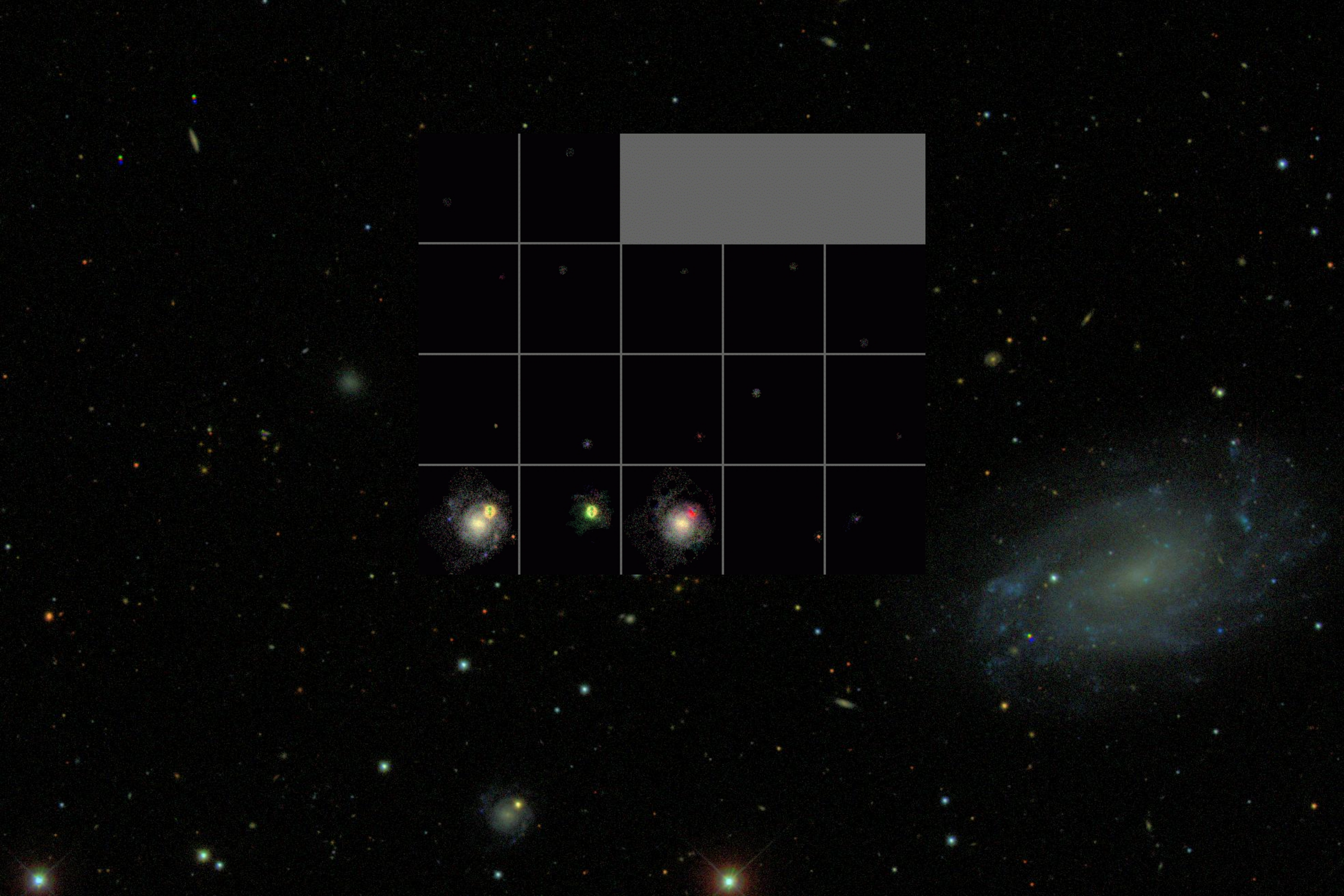
What does the Hubble Deep Field Look like in Colour?



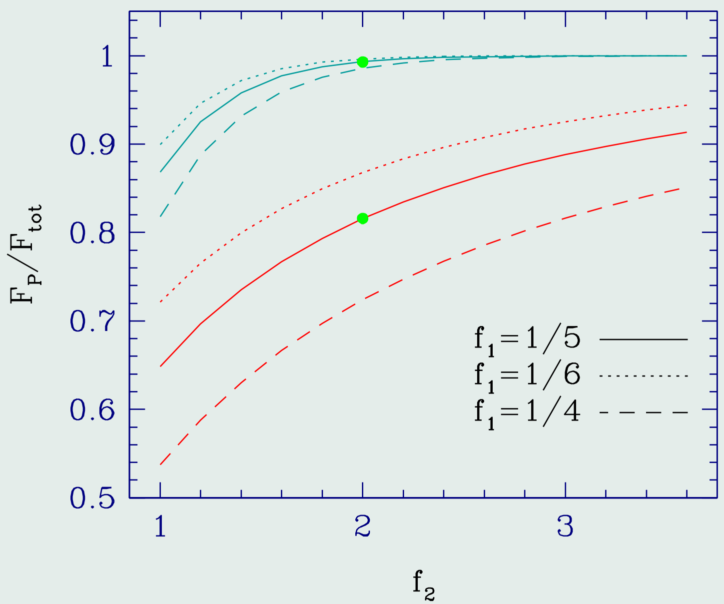






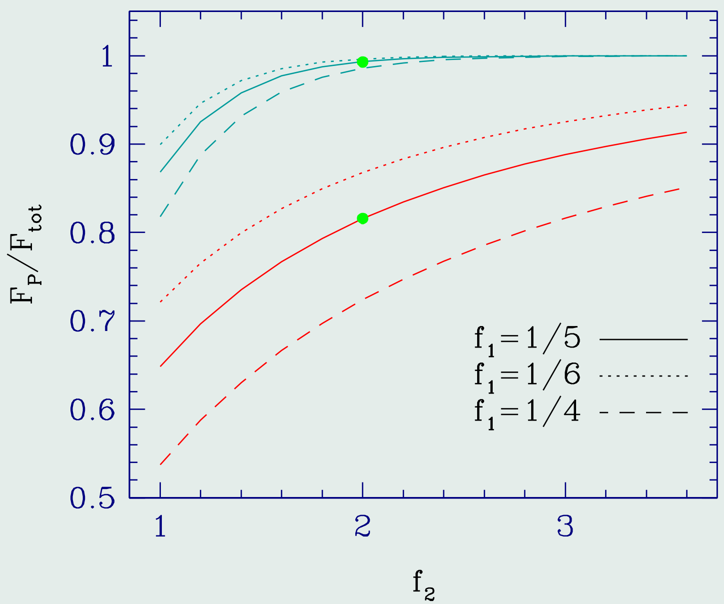


Petrosian v. Total Fluxes



$$f_1 \equiv \frac{\int_0^{R_P} I\, 2\pi r\, dr}{\pi {R_P}^2};\; F_P \equiv \int_0^{f_2 R_P} I\, 2\pi r\, dr.$$

Petrosian v. Total Fluxes



$$f_1 \equiv \frac{\int_0^{R_P} I\,2\pi r\,dr}{\pi {R_P}^2};\ F_P \equiv \int_0^{f_2 R_P} I\,2\pi r\,dr.$$

I actually use deV models truncated $8r_e$; this reduces the flux by 0.080 magnitudes (I truncate the exp models at $4r_e$ which reduces the flux by 0.018 magnitudes)

Speeding up the LM Function Evaluation

Speeding up the LM Function Evaluation

- *Rather that work directly in pixel space, I fit the models to the extracted 'cell profile'.*

Speeding up the LM Function Evaluation

- *Rather that work directly in pixel space, I fit the models to the extracted 'cell profile'.*
- *The models are symmetrical, so I only need consider the average of pairs of cells placed symmetrically about the object's centre.*

Speeding up the LM Function Evaluation

- *Rather that work directly in pixel space, I fit the models to the extracted 'cell profile'.*
- *The models are symmetrical, so I only need consider the average of pairs of cells placed symmetrically about the object's centre.*
- *I model the PSF as a sum of Gaussians and a residual table R:*

$$PSF = \alpha N(0, \sigma^2) + \beta \left(N(0, \tau^2) + bN(0, (c\tau)^2) \right) + R$$

where b and c are fixed (I adopt 0.1 and 3 respectively).

- *I precompute galaxy models of each type for a range of* $(r_e, a/b, \phi)$*, convolve each with a set of PSFs of the forms* $N(0, \sigma^2)$ *and* $N(0, \tau^2) + bN(0, (c\tau)^2)$ *for a set of values of* σ *and* τ*, extract their profiles, and save the results to disk.*

- *I precompute galaxy models of each type for a range of* $(r_e, a/b, \phi)$*, convolve each with a set of PSFs of the forms* $N(0,\sigma^2)$ *and* $N(0,\tau^2) + bN(0,(c\tau)^2)$ *for a set of values of* σ *and* τ*, extract their profiles, and save the results to disk.*
- *I save the pre-extracted model profiles as Fourier series in which only the* $cos(2r\theta)$ *terms are non-zero.*

  With this Fourier expansion in hand, the profiles are a smooth function of ϕ*, and I can therefore use standard efficient techniques to solve for* ϕ *given* $(r_e, a/b)$*; this essentially reduces the dimensionality of the non-linear optimisation from three to two.*

PSF representations

I represent the PSF at a point with a KL expansion (Lupton et al.; ADASS X). I need the best representation of that KL PSF as a sum of Gaussians, where the σ *and* τ *are restricted to the values present in the pre-computed model tables:* $PSF_{KL} = PSF_{table} + R$.

PSF representations

I represent the PSF at a point with a KL expansion (Lupton et al.; ADASS X). I need the best representation of that KL PSF as a sum of Gaussians, where the σ *and* τ *are restricted to the values present in the pre-computed model tables:* $\mathit{PSF}_{KL} = \mathit{PSF}_{\mathit{table}} + R$.

We may then write

$$\begin{aligned} \mathit{model} &= \quad \mathit{model}_0 \otimes \mathit{PSF}_{KL} \\ &\approx \mathit{model}_0 \otimes \mathit{PSF}_{\mathit{table}} + R \end{aligned}$$

where model_0 *is the model galaxy above the atmosphere and model is that model after convolution with the PSF.*

The fit is regularised with a term dependent on the difference between the width of the true (actually KL) PSF, and the best represention in terms of sums of Gaussians