Difference Imaging: Algorithms, Problems, and some Possible Solutions

Robert Lupton

2016-04-12

Measuring Variability

Given two images of a patch of sky I_1 and I_2 with PSFs ϕ_1 and ϕ_2 how should I measure a star (or SNe Ia)'s light curve (*i.e.* the fluxes A_1 and A_2)?

Measuring Variability

Given two images of a patch of sky I_1 and I_2 with PSFs ϕ_1 and ϕ_2 how should I measure a star (or SNe Ia)'s light curve (*i.e.* the fluxes A_1 and A_2)?
The first thing to try is to measure the flux in each image and subtract. If I know the PSFs I can write down a model

$$I_i = A\phi(\boldsymbol{x}_i) + \epsilon_i$$

and make an optimal measurement of each flux

$$A_r = \frac{\sum_i I_{r,i}\, \phi_{r,i}/\sigma_{r,i}^2}{\sum_i \phi_{r,i}^2/\sigma_{r,i}^2}$$

where $r = 1, 2$ and i runs over the pixels.

Measuring Variability

Given two images of a patch of sky I_1 and I_2 with PSFs ϕ_1 and ϕ_2 how should I measure a star (or SNe Ia)'s light curve (*i.e.* the fluxes A_1 and A_2)?
The first thing to try is to measure the flux in each image and subtract. If I know the PSFs I can write down a model

$$I_i = A\phi(\boldsymbol{x}_i) + \epsilon_i$$

and make an optimal measurement of each flux

$$A_r = \frac{\sum_i I_{r,i}\,\phi_{r,i}/\sigma_{r,i}^2}{\sum_i \phi_{r,i}^2/\sigma_{r,i}^2}$$

where $r = 1, 2$ and i runs over the pixels. For faint sources the noise is dominated by the sky noise, and we find

$$A_1 - A_2 = \frac{\sum_i I_{1,i}\,\phi_{1,i}}{\sum_i \phi_{1,i}^2} - \frac{\sum_i I_{2,i}\,\phi_{2,i}}{\sum_i \phi_{2,i}^2}$$

Measuring Variability

Given two images of a patch of sky I_1 and I_2 with PSFs ϕ_1 and ϕ_2 how should I measure a star (or SNe Ia)'s light curve (*i.e.* the fluxes A_1 and A_2)?

The first thing to try is to measure the flux in each image and subtract. If I know the PSFs I can write down a model

$$I_i = A\phi(\boldsymbol{x}_i) + \epsilon_i$$

and make an optimal measurement of each flux

$$A_r = \frac{\sum_i I_{r,i}\, \phi_{r,i}/\sigma^2_{r,i}}{\sum_i \phi^2_{r,i}/\sigma^2_{r,i}}$$

where $r = 1, 2$ and i runs over the pixels. For faint sources the noise is dominated by the sky noise, and we find

$$A_1 - A_2 = \frac{\sum_i I_{1,i}\, \phi_{1,i}}{\sum_i \phi^2_{1,i}} - \frac{\sum_i I_{2,i}\, \phi_{2,i}}{\sum_i \phi^2_{2,i}}$$

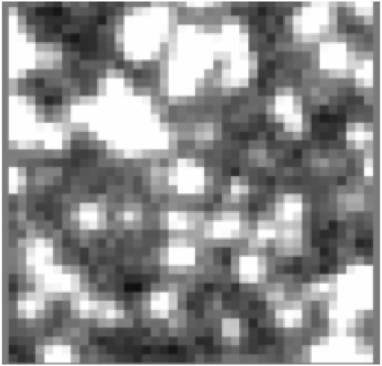
If the images are complicated (*e.g.* the Galactic centre or M31) these measurements may not be very good; in fact, the expressions for A_r were only optimal for isolated objects.

M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*

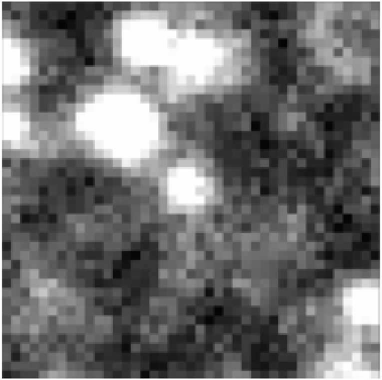
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



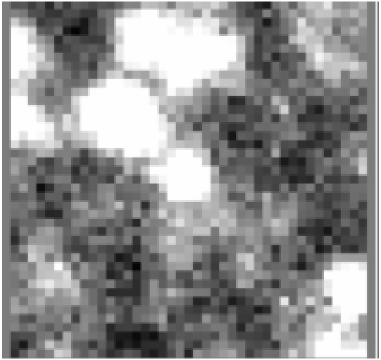
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



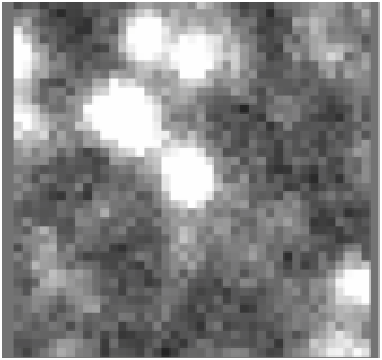
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



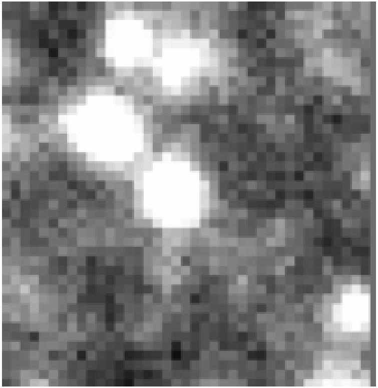
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



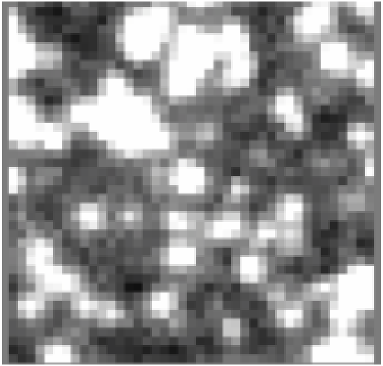
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



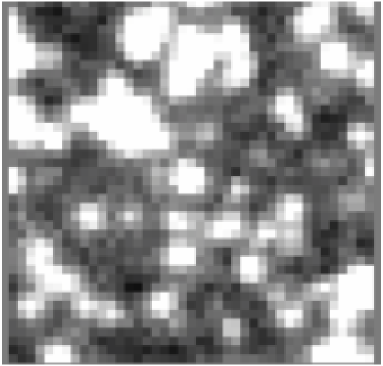
M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



M31

Takada Masahiro and Niikura Hiroko used HSC to image M31 for an entire night, *e.g.*



The "isolated star" condition isn't satisfied.

Classical Image Subtraction

If the two PSFs were the same we could solve this problem by calculating $I_1 - I_2$ directly:

$$A_1 - A_2 = \frac{\sum_i (I_{1,i} - I_{2,i})\, \phi_i}{\sum_i \phi_i^2}$$

Classical Image Subtraction

If the two PSFs were the same we could solve this problem by calculating $I_1 - I_2$ directly:

$$A_1 - A_2 = \frac{\sum_i (I_{1,i} - I_{2,i})\,\phi_i}{\sum_i \phi_i^2}$$

If $\phi_1 \neq \phi_2$ many people (*e.g.* Davis and Phillips) have proposed correcting I_2 in Fourier space:

$$I_2'(k) = I_2(k)\frac{\phi_1(k)}{\phi_2(k)} \equiv I_2(k)\kappa(k)$$

and then proceeding as before.

Classical Image Subtraction

If the two PSFs were the same we could solve this problem by calculating $I_1 - I_2$ directly:

$$A_1 - A_2 = \frac{\sum_i (I_{1,i} - I_{2,i})\,\phi_i}{\sum_i \phi_i^2}$$

If $\phi_1 \neq \phi_2$ many people (*e.g.* Davis and Phillips) have proposed correcting I_2 in Fourier space:

$$I_2'(k) = I_2(k)\frac{\phi_1(k)}{\phi_2(k)} \equiv I_2(k)\kappa(k)$$

and then proceeding as before. This turns out to be problematic as you need to know the PSFs very well.

Classical Image Subtraction

If the two PSFs were the same we could solve this problem by calculating $I_1 - I_2$ directly:

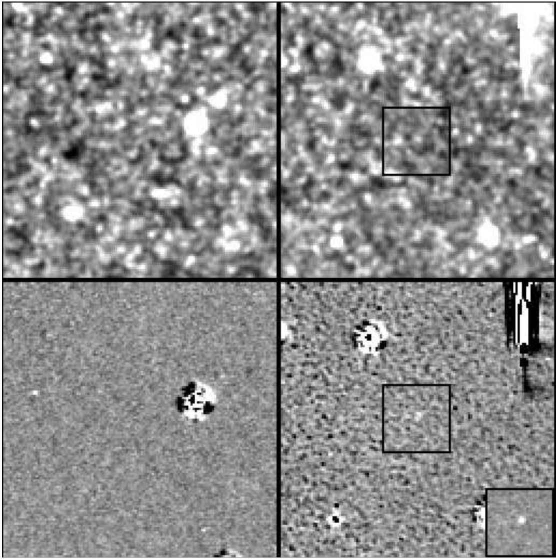
$$A_1 - A_2 = \frac{\sum_i (I_{1,i} - I_{2,i})\,\phi_i}{\sum_i \phi_i^2}$$

If $\phi_1 \neq \phi_2$ many people (e.g. Davis and Phillips) have proposed correcting I_2 in Fourier space:

$$I_2'(k) = I_2(k)\frac{\phi_1(k)}{\phi_2(k)} \equiv I_2(k)\kappa(k)$$

and then proceeding as before. This turns out to be problematic as you need to know the PSFs very well. Problems are revealed in the residual image $I_1 - I_2'$.

Residuals in M31 (using a spatially-constant PSF model)



(Tomaney and Crotts, 1996)

Cross-Convolution

Another obvious approach (Gal-Yam) is to construct the difference image as

$$\phi_2 \otimes I_1 - \phi_1 \otimes I_2$$

but this sacrifices resolution.

Alard and Lupton

Christophe Alard and I proposed a way to circumvent the need to know ϕ by choosing the kernel κ to minimise the residuals in the subtracted image.

Alard and Lupton

Christophe Alard and I proposed a way to circumvent the need to know ϕ by choosing the kernel κ to minimise the residuals in the subtracted image.
If we write

$$I_2' = \kappa \otimes I_2$$

and expand

$$\kappa = \sum_r a_r B^r$$

we may minimise

$$\left| \frac{I_1 - \sum_r a_r \left(B^r \otimes I_2 \right)}{\sigma} \right|^2$$

by solving linear equations for a_r.

Alard and Lupton

Christophe Alard and I proposed a way to circumvent the need to know ϕ by choosing the kernel κ to minimise the residuals in the subtracted image.
If we write

$$I_2' = \kappa \otimes I_2$$

and expand

$$\kappa = \sum_r a_r B^r$$

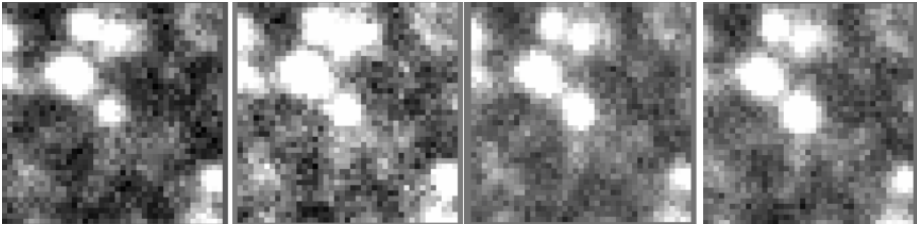
we may minimise

$$\left| \frac{I_1 - \sum_r a_r \left(B^r \otimes I_2 \right)}{\sigma} \right|^2$$

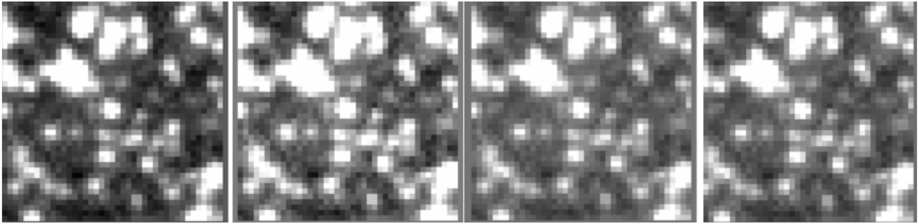
by solving linear equations for a_r.

The choice of B^r is arbitrary. We originally used Gauss-Hermite functions, but people have also investigated using δ-function (pixel) bases.

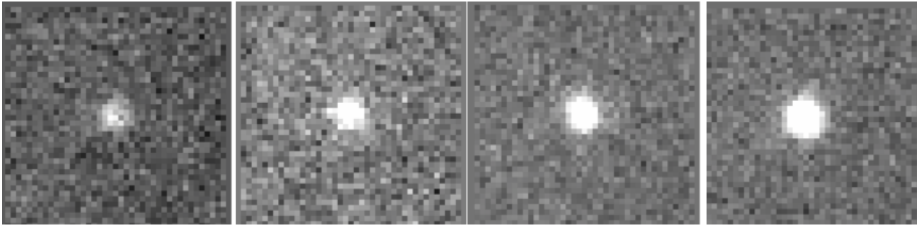
M31

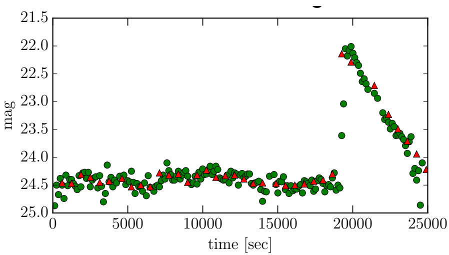


M31



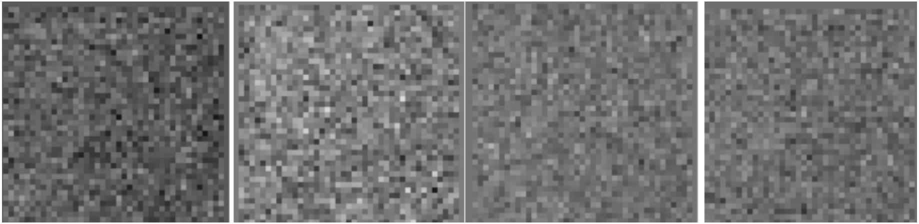
M31

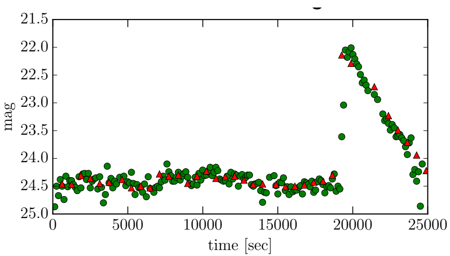




21.5
22.0
22.5
23.0
23.5
24.0
24.5
25.0
mag
0
5000
10000
15000
20000
25000
time [sec]

M31



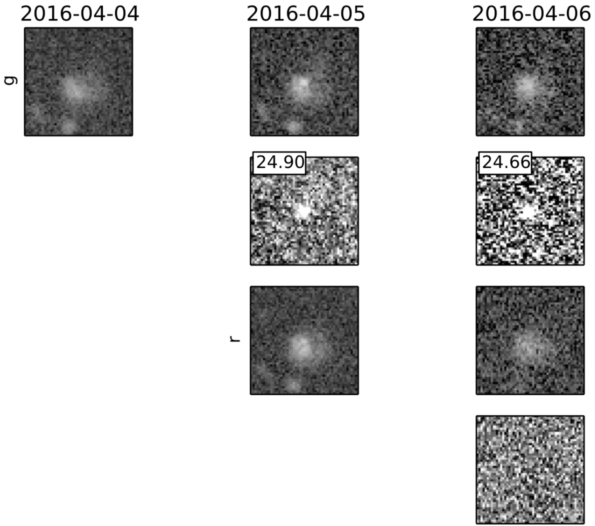

mag
time [sec]
21.5
22.0
22.5
23.0
23.5
24.0
24.5
25.0
0
5000
10000
15000
20000
25000

MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:

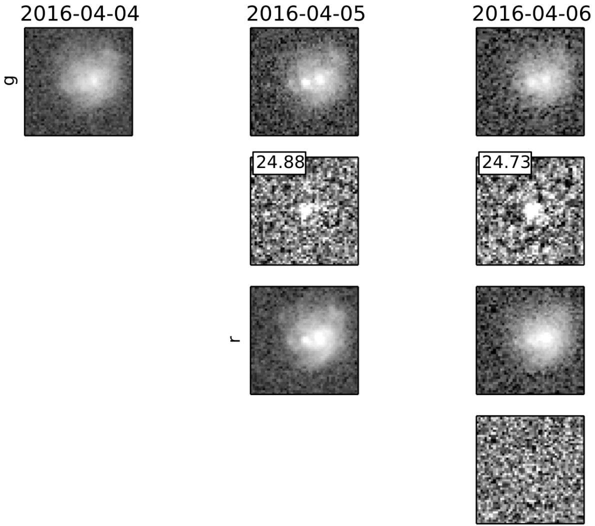
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



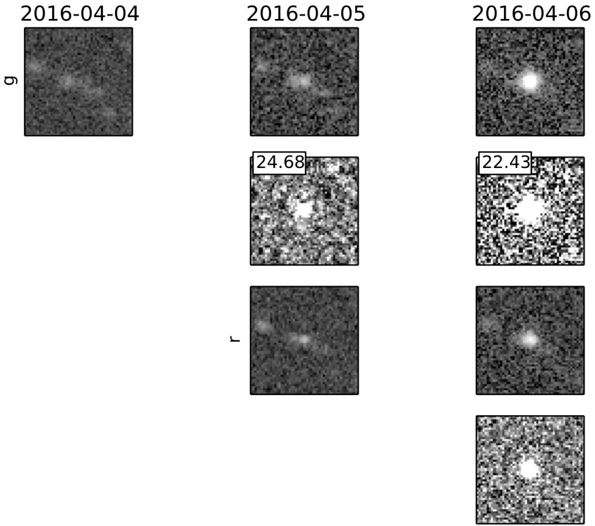
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



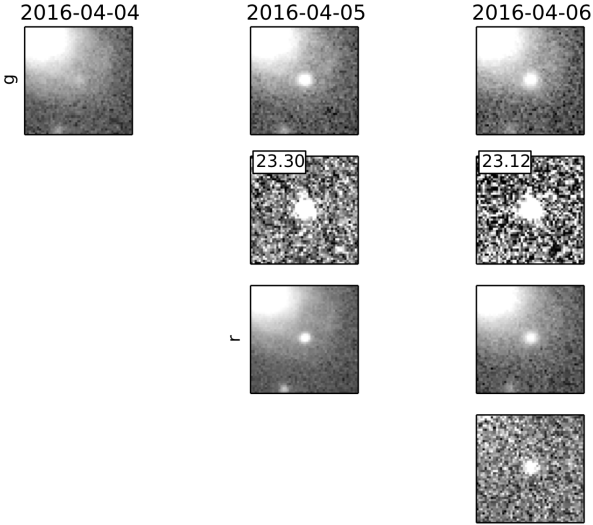
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



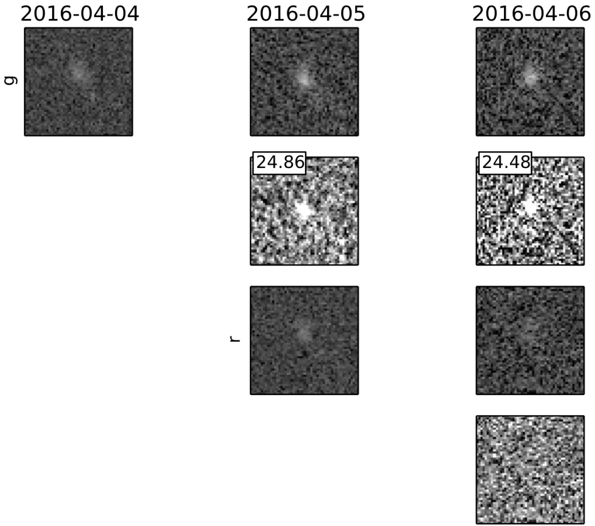
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



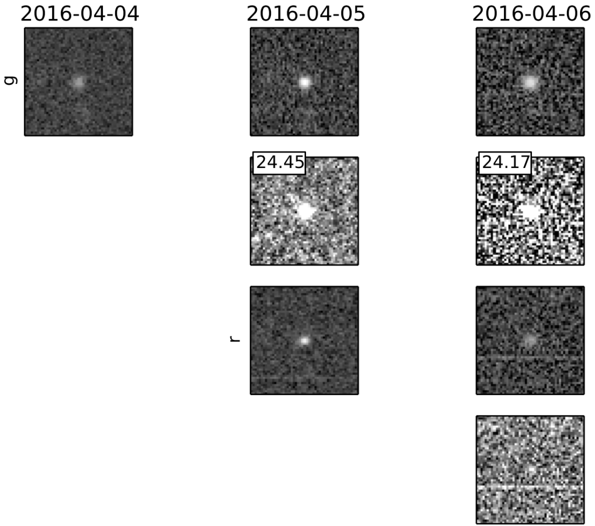
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



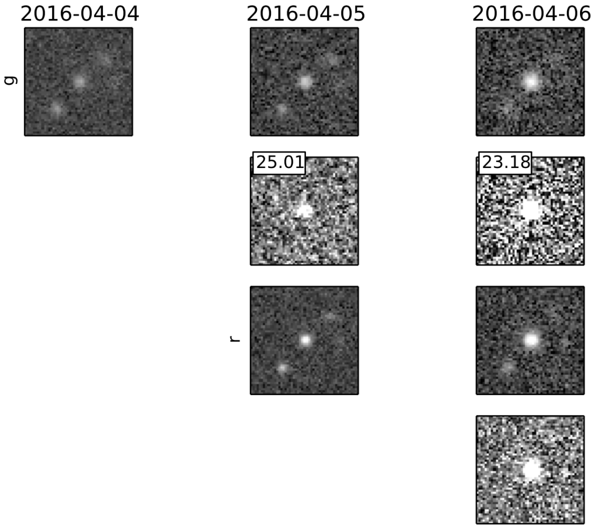
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



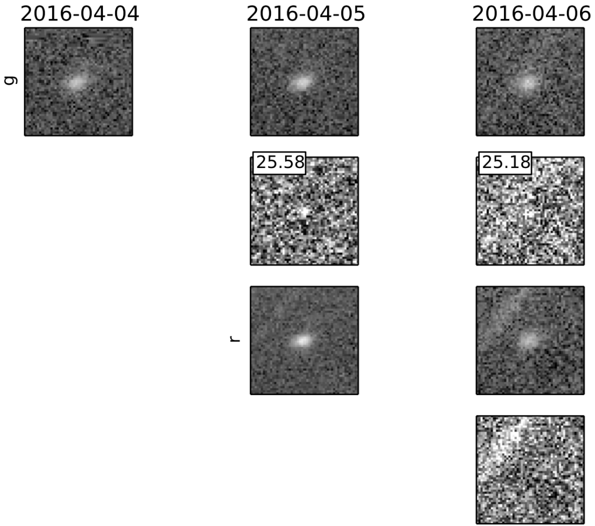
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



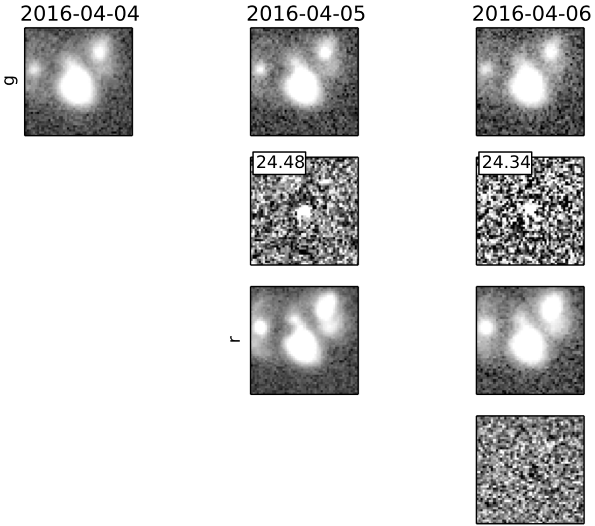
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



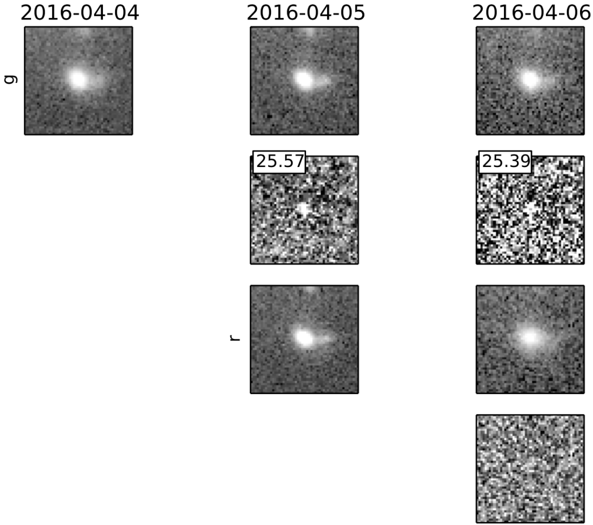
MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



MUSSES

Jiang Jing, Doi Mamuru, and Yasuda Naoki have been looking for early SNe:



Problems with Alard and Lupton

Problems with the Alard and Lupton Algorithm

A&L is optimal for the problem it was designed to solve, but...

Problems with the Alard and Lupton Algorithm

A&L is optimal for the problem it was designed to solve, but…

- Where does the template I_2 come from?

Problems with the Alard and Lupton Algorithm

A&L is optimal for the problem it was designed to solve, but…

- Where does the template I_2 come from?

- What should we do if our camera has no atmospheric dispersion corrector (ADC) (*e.g.* DECam, LSST)?

Problems with the Alard and Lupton Algorithm

A&L is optimal for the problem it was designed to solve, but…

- Where does the template I_2 come from?
- What should we do if our camera has no atmospheric dispersion corrector (ADC) (*e.g.* DECam, LSST)?
- What should we do if the data is sharper than the template?

Problems with the Alard and Lupton Algorithm

A&L is optimal for the problem it was designed to solve, but…

- Where does the template I_2 come from?
- What should we do if our camera has no atmospheric dispersion corrector (ADC) (*e.g.* DECam, LSST)?
- What should we do if the data is sharper than the template?
- What is the consequence of noise in the template?

Where Does The Template I_2 Come From?

We didn't attempt to solve this problem, but fortunately Nick Kaiser did 15 years ago.

Where Does The Template I_2 Come From?

We didn't attempt to solve this problem, but fortunately Nick Kaiser did 15 years ago. Unfortunately he didn't publish his result except as "PSDC-002-01[01]-00".

Where Does The Template I_2 Come From?

We didn't attempt to solve this problem, but fortunately Nick Kaiser did 15 years ago. Unfortunately he didn't publish his result except as "PSDC-002-01[01]-00".

If I have a set of n realisations of an image I_r with known PSFs ϕ_r, what is the best estimate for the true image above the atmosphere, T?

Where Does The Template I_2 Come From?

We didn't attempt to solve this problem, but fortunately Nick Kaiser did 15 years ago. Unfortunately he didn't publish his result except as "PSDC-002-01[01]-00".

If I have a set of n realisations of an image I_r with known PSFs ϕ_r, what is the best estimate for the true image above the atmosphere, T?

We know that

$$I_{r,i} = (T \otimes \phi_r)_i + \epsilon_{r,i}$$

and let's assume that ϵ_r is an $N(0, \sigma_r^2)$ variable (*i.e.* we only care about faint objects)

Where Does The Template I_2 Come From?

We didn't attempt to solve this problem, but fortunately Nick Kaiser did 15 years ago. Unfortunately he didn't publish his result except as "PSDC-002-01[01]-00".

If I have a set of n realisations of an image I_r with known PSFs ϕ_r, what is the best estimate for the true image above the atmosphere, T?

We know that

$$I_{r,i} = (T \otimes \phi_r)_i + \epsilon_{r,i}$$

and let's assume that ϵ_r is an $N(0, \sigma_r^2)$ variable (*i.e.* we only care about faint objects)

We may estimate each Fourier mode independently using an ML estimator:

$$\ln \mathcal{L} \propto \sum_r \frac{\left(I_r(k) - T(k)\phi_r(k)\right)^2}{\sigma_r^2}$$

(assuming for clarity of presentation that the PSF is symmetric, and thus $\phi^* = \phi$) *i.e.*

$$\hat{T}(k) = \frac{\sum_r I_r(k)\phi_r(k)/\sigma_r^2}{\sum_r \phi_r(k)^2/\sigma_r^2}$$

with variance

$$\mathrm{Var}(\hat{T}(k)) = \frac{1}{\sum_r \phi_r(k)^2/\sigma_r^2}$$

An Optimal Template

If we'd like a template with uncorrelated noise we need to flatten the noise, resulting in

$$\hat{T}'(k) = \frac{\sum_r I_r(k)\phi_r(k)/\sigma_r^2}{\sqrt{\sum_r \phi_r(k)^2/\sigma_r^2}\sqrt{\sum_r 1/\sigma_r^2}}$$

Bad Things in Images

The assumption that ϵ_r is an $N(0, \sigma_r^2)$ states that the only things in our images are objects (with variable seeing) and noise.

Bad Things in Images

The assumption that ϵ_r is an $N(0, \sigma_r^2)$ states that the only things in our images are objects (with variable seeing) and noise. In reality there are:

- asteroids
- satellites
- cosmic rays
- diffraction spikes
- …

Bad Things in Images

The assumption that ϵ_r is an $N(0, \sigma_r^2)$ states that the only things in our images are objects (with variable seeing) and noise. In reality there are:

- asteroids
- satellites
- cosmic rays
- diffraction spikes
- …

People like to deal with these by taking a stack of images and taking the median or using a $5 - \sigma$ clip.

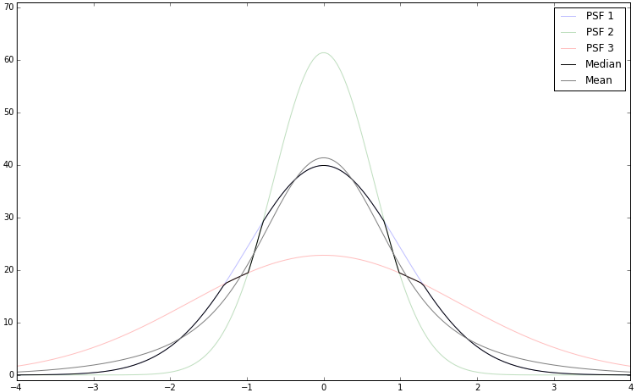
Bad Things in Images

The assumption that ϵ_r is an $N(0, \sigma_r^2)$ states that the only things in our images are objects (with variable seeing) and noise. In reality there are:

- asteroids
- satellites
- cosmic rays
- diffraction spikes
- …

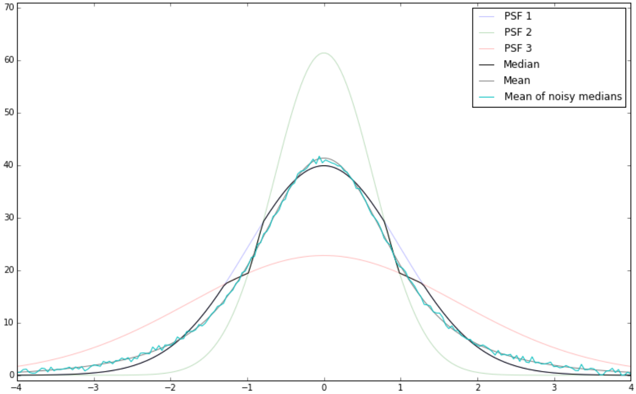
People like to deal with these by taking a stack of images and taking the median or using a $5 - \sigma$ clip. This is not a good idea; the problem is not statistical.

Building Templates with Clipping



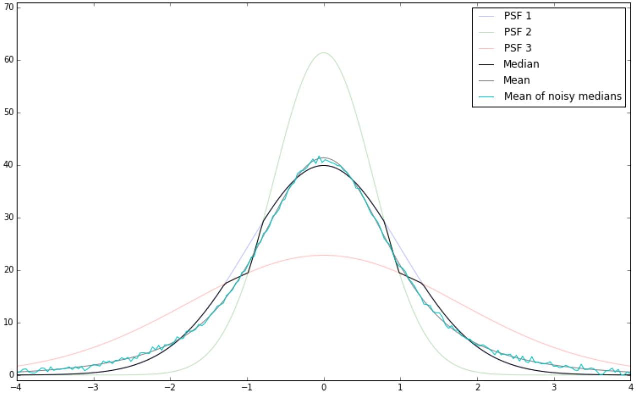
Three realisations of the image of a star with three different PSFs.

Building Templates with Clipping



Three realisations of the image of a star with three different PSFs.

Building Templates with Clipping



Three realisations of the image of a star with three different PSFs.
As far as I know, your only options are PSF-matched image or cunning exploitation of difference images while building templates -- which is circular.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon. If our science image and template image are taken at different airmasses, the difference image will be a dipole.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon. If our science image and template image are taken at different airmasses, the difference image will be a dipole. This isn't good. We subtracted the two images to convert a complicated image into a simple one, but now we've made a complicated image of dipoles.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon. If our science image and template image are taken at different airmasses, the difference image will be a dipole. This isn't good. We subtracted the two images to convert a complicated image into a simple one, but now we've made a complicated image of dipoles.

I don't have code to handle this (yet), but it looks as if we can use a set of images taken at different airmasses to estimate each pixel's SED.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon. If our science image and template image are taken at different airmasses, the difference image will be a dipole. This isn't good. We subtracted the two images to convert a complicated image into a simple one, but now we've made a complicated image of dipoles.

I don't have code to handle this (yet), but it looks as if we can use a set of images taken at different airmasses to estimate each pixel's SED. I predict that Yasuda-san will be suspicious of this approach; we'll see how well it works.

What should we do if our camera has no ADC?

This is a bit unpleasant. The atmosphere refracts blue light more than red, so a red/blue pair of stars that overlap at zenith become separated as we point towards the horizon. If our science image and template image are taken at different airmasses, the difference image will be a dipole. This isn't good. We subtracted the two images to convert a complicated image into a simple one, but now we've made a complicated image of dipoles.

I don't have code to handle this (yet), but it looks as if we can use a set of images taken at different airmasses to estimate each pixel's SED. I predict that Yasuda-san will be suspicious of this approach; we'll see how well it works.

Another popular way to get dipoles is by having bad astrometry; this is probably the biggest technical problem that current transient surveys face.

What Should We Do If The Data Is Sharper Than The Template?

After we've found our difference image *D*, how should we measure the flux?

What Should We Do If The Data Is Sharper Than The Template?

After we've found our difference image D, how should we measure the flux? That's easy; I already told you the answer.

$$A = \frac{\sum_i D_i \, \phi_{1,i}}{\sum_i \phi_{1,i}^2}$$

i.e. find the value of $D \otimes \phi$ and divide by $\sum \phi^2$.

What Should We Do If The Data Is Sharper Than The Template?

After we've found our difference image D, how should we measure the flux? That's easy; I already told you the answer.

$$A = \frac{\sum_i D_i\, \phi_{1,i}}{\sum_i \phi_{1,i}^2}$$

i.e. find the value of $D \otimes \phi$ and divide by $\sum \phi^2$.
We can find this from I_1 directly; our A&L problem becomes to find the kernel κ' which minimises

$$\left| \frac{I_1 \otimes \phi_1 - \kappa' \otimes I_2}{\sigma} \right|^2$$

What Should We Do If The Data Is Sharper Than The Template?

After we've found our difference image D, how should we measure the flux? That's easy; I already told you the answer.

$$A = \frac{\sum_i D_i \, \phi_{1,i}}{\sum_i \phi_{1,i}^2}$$

i.e. find the value of $D \otimes \phi$ and divide by $\sum \phi^2$.
We can find this from I_1 directly; our A&L problem becomes to find the kernel κ' which minimises

$$\left| \frac{I_1 \otimes \phi_1 - \kappa' \otimes I_2}{\sigma} \right|^2$$

Providing our frame I_1 is at least $\sqrt{2}$ wider than the template I_2 we won't have a problem.

What Should We Do If The Data Is Sharper Than The Template?

After we've found our difference image D, how should we measure the flux? That's easy; I already told you the answer.

$$A = \frac{\sum_i D_i \, \phi_{1,i}}{\sum_i \phi_{1,i}^2}$$

i.e. find the value of $D \otimes \phi$ and divide by $\sum \phi^2$.
We can find this from I_1 directly; our A&L problem becomes to find the kernel κ' which minimises

$$\left| \frac{I_1 \otimes \phi_1 - \kappa' \otimes I_2}{\sigma} \right|^2$$

Providing our frame I_1 is at least $\sqrt{2}$ wider than the template I_2 we won't have a problem.

Cross-Convolution doesn't suffer from the problem of over-sharp templates.

What Is The Consequence Of Noise In The Template?

This one is harder.

What Is The Consequence Of Noise In The Template?

This one is harder. If the template has noise, then $\kappa \otimes I_2$ will have correlated noise which complicates the analysis (for example, those optimal estimates of fluxes were written assuming diagonal covariance estimates). This is not fatal, but it is a nuisance.

What Is The Consequence Of Noise In The Template?

This one is harder. If the template has noise, then $\kappa \otimes I_2$ will have correlated noise which complicates the analysis (for example, those optimal estimates of fluxes were written assuming diagonal covariance estimates). This is not fatal, but it is a nuisance. I knew this.

What Is The Consequence Of Noise In The Template?

This one is harder. If the template has noise, then $\kappa \otimes I_2$ will have correlated noise which complicates the analysis (for example, those optimal estimates of fluxes were written assuming diagonal covariance estimates). This is not fatal, but it is a nuisance. I knew this.

More interestingly if the template is noisy, A&L is no longer optimal.

What Is The Consequence Of Noise In The Template?

This one is harder. If the template has noise, then $\kappa \otimes I_2$ will have correlated noise which complicates the analysis (for example, those optimal estimates of fluxes were written assuming diagonal covariance estimates). This is not fatal, but it is a nuisance. I knew this.

More interestingly if the template is noisy, A&L is no longer optimal. I hadn't realised this until I read a paper by Barak Zackay, Eran Ofek and Avishay Gal-Yam.

Subtracting Two Noisy Images

Let as adopt an A&L approach and write

$$D = I_1 - \kappa \otimes I_2$$

with the Gaussian homoschedastic (faint-object) assumption.

Subtracting Two Noisy Images

Let as adopt an A&L approach and write

$$D = I_1 - \kappa \otimes I_2$$

with the Gaussian homoschedastic (faint-object) assumption. Our model is that the difference image is D convolved with the PSF ϕ_1, so taking a Fourier transform and constructing the log-likelihood gives

$$\ln \mathcal{L} \sim \sum_k \frac{(I_1(k) - \kappa(k) I_2(k) - D(k)\phi_1(k))^2}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}$$

and the MLE for $D(k)$ is

$$\hat{D}(k) = \frac{I_1(k) - \kappa(k) I_2(k)}{\phi_1(k)}$$

with variance

$$\mathrm{Var}(\hat{D}(k)) = \frac{\sigma_1^2 + \kappa^2(k)\sigma_2^2}{\phi_1^2(k)}$$

Subtracting Two Noisy Images

That variance diverges at large k -- not surprising, as we're estimating a deconvolved scene D. As in Kaiser's analysis (and as emphasised by Zackay *et al.*) we can construct an uncorrelated image by whitening the noise, resulting in

$$\hat{D}(k) = (I_1(k) - \kappa(k) I_2(k)) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}}$$

with PSF

$$\phi_D(k) = \phi_1(k) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}}$$

Subtracting Two Noisy Images

That variance diverges at large k -- not surprising, as we're estimating a deconvolved scene D. As in Kaiser's analysis (and as emphasised by Zackay *et al.*) we can construct an uncorrelated image by whitening the noise, resulting in

$$\hat{D}(k) = (I_1(k) - \kappa(k) I_2(k)) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}}$$

with PSF

$$\phi_D(k) = \phi_1(k) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}}$$

i.e. we can estimate κ by standard methods, and then correct it for the noise in the template.

`Proper Image Subtraction'

Zackay *et al.* carry out what amounts to this calculation, assuming that ϕ_1 and ϕ_2 are known and that therefore $\kappa(k) = \phi_1(k)/\phi_2(k)$. If we substitute this into our equation for $\hat{D}$ and ϕ_D we find

$$D(k) = (\phi_2(k) I_1(k) - \phi_1(k) I_2(k)) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \phi_2^2(k) + \sigma_2^2 \phi_1^2(k)}}$$

$$\phi_D(k) = \phi_1(k)\phi_2(k) \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \phi_2^2(k) + \sigma_2^2 \phi_1^2(k)}}$$

which are Zackay *et al.*'s equations 13 and 14, except that I've multiplied D by $(\sigma_1^2 + \sigma_2^2)^{1/2}$)

`Proper Image Subtraction'

Zackay *et al.* carry out what amounts to this calculation, assuming that ϕ_1 and ϕ_2 are known and that therefore $\kappa(k) = \phi_1(k)/\phi_2(k)$. If we substitute this into our equation for $\hat{D}$ and ϕ_D we find

$$D(k) = (\phi_2(k)I_1(k) - \phi_1(k)I_2(k))\sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2\phi_2^2(k) + \sigma_2^2\phi_1^2(k)}}$$

$$\phi_D(k) = \phi_1(k)\phi_2(k)\sqrt{\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2\phi_2^2(k) + \sigma_2^2\phi_1^2(k)}}$$

which are Zackay *et al.*'s equations 13 and 14, except that I've multiplied D by $(\sigma_1^2 + \sigma_2^2)^{1/2}$)

One interesting feature of these equations is that they are symmetric in I_1 and I_2 and are thus able to handle better seeing in the science image than in the template. In this sense they are an optimal version of cross-correlation methods.

Sensitivity To Noise Levels

If the template is noise free ($\sigma_2 = 0$), we recover

$$\hat{D}(k) = I_1(k) - \kappa(k) I_2(k)$$
$$\phi_D(k) = \phi_1(k)$$

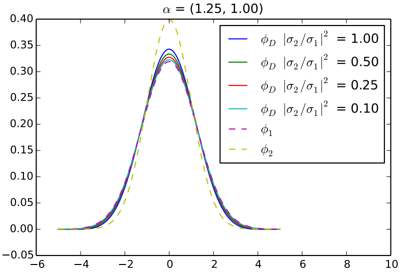
which are just the standard equations for difference imaging.

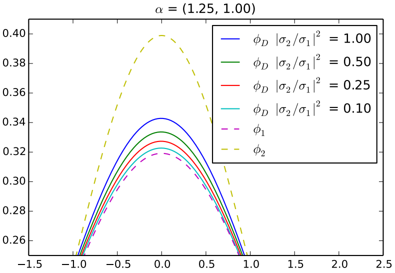
Sensitivity To Noise Levels

If the template is noise free ($\sigma_2 = 0$), we recover

$$\hat{D}(k) = I_1(k) - \kappa(k) I_2(k)$$
$$\phi_D(k) = \phi_1(k)$$

which are just the standard equations for difference imaging.
Numerically, once the S/N in the template is more than *c.* twice the science image the results are similar to the noise-free case





Likelihood Images

Given $\hat{D}$ and ϕ_D we can calculate the $\ln \mathcal{L}$ image $\hat{D} \otimes \phi_D$; the result is

$$\ln \mathcal{L} \sim \sum_k \phi_1(k) \frac{I_1(k) - \kappa(k) I_2(k)}{\sigma_1^2 + \kappa^2(k)\sigma_2^2}$$

(which is equivalent to equation 12 of the ZOGY paper).
In the noiseless template limit ($\sigma_2 \to 0$) this becomes

$$\ln \mathcal{L} \sim \sum_k \phi_1(k) \left(I_1(k) - \kappa(k) I_2(k)\right)$$

which is (unsurprisingly) precisely my pre-convolution proposal.