

# CCP4 interest in diffraction integration software

Harvard Medical School

David Waterman

CCP4

July 2012



**Funding:** CCP4 (DGW), EU FP7 BioStruct-X & Diamond Light Source for the rest

**Colleagues:** Graeme Winter, Gwyndaf Evans and Alun Ashton at Diamond

**Thinking:** Andrew Leslie, Phil Evans, Gleb Bourenkov, Garib Murshudov

**Materials:** Andrew Leslie for slides on Mosflm

**Collaboration:** Nick Sauter's group (NIH grant R01GM095887 for synchrotron data processing)



- Data reduction with CCP4 now
- The Mosflm legacy
- Beyond Mosflm → DIALS
- Error estimation: personal interest in a new framework for integration software



# Data reduction with CCP4 now



Mosflm/iMosflm

The screenshot shows the CCP4 Integration window for file hg\_001.mtz. The interface is divided into several sections:

- Navigation:** Images, Indexing, Strategy, Cell Refinement, Integration, History.
- Integration Parameters:**

Parameter	Value	Fix
Beam x	119.51	<input type="checkbox"/>
Beam y	120.02	<input type="checkbox"/>
Distance	249.50	<input type="checkbox"/>
Y-scale	1.0003	<input type="checkbox"/>
Tilt	0.02	<input type="checkbox"/>
Twist	0.11	<input type="checkbox"/>
Tangential offset	-0.040	<input type="checkbox"/>
Radial offset	-0.040	<input type="checkbox"/>
RMS residual	0.043	
RMS res. (central)	0.031	
RMS res. (weighted)	0.540	
- Integration Parameters (continued):**

Parameter	Value	Fix
w(x)	-0.04	
w(y)	-0.11	
w(z)	0.02	
a	58.43	<input checked="" type="checkbox"/>
b	58.43	<input checked="" type="checkbox"/>
c	156.01	<input checked="" type="checkbox"/>
$\alpha$	90.00	<input checked="" type="checkbox"/>
$\beta$	90.00	<input checked="" type="checkbox"/>
$\gamma$	120.00	<input checked="" type="checkbox"/>
Mosaicity	1.024	<input type="checkbox"/>
- Integration Statistics:**

Parameter	Full	Partial
$\langle I \rangle$ (prf)	0.00	10.70
$\langle I \rangle$ (sum)	0.00	10.30
Reflections	0	728
$\langle I \rangle$ - HR (prf)	0.00	4.80
$\langle I \rangle$ - HR (sum)	0.00	4.80
Reflections HR	0	133
- Plots:** Three plots showing the variation of parameters (Twist, Tilt, and  $\langle I \rangle$ ) across different image numbers. The Twist plot shows a yellow line fluctuating around 0.11. The Tilt plot shows a red line fluctuating around 0.02. The  $\langle I \rangle$  plot shows a yellow line for partial reflections and a red line for full reflections, both showing a general downward trend as the image number increases.
- Image Grid:** A grid of 15 images, with the central image (Image 10) highlighted. The grid is arranged in a 3x5 pattern.
- Block Selection:** A list of blocks, with Block 1 selected.
- Profile fits:** A bar chart showing the profile fits for full reflections (red bars) and partial reflections (yellow bars) across different image numbers. The x-axis represents the resolution in Angstroms (Å), and the y-axis represents the intensity.

[2 Warnings]

# Data reduction with CCP4 now



Mosflm/iMosflm

- Mosflm is a *very* successful package
- Widely available (through CCP4)
- Typical user experience is *via* the excellent GUI, iMosflm
- This encourages inspection of the images, to identify
  - poor spot shapes
  - anisotropic diffraction
  - multiple lattices
  - very high mosaicity (increase threshold in indexing)
  - incorrect direct beam position (no. 1 cause of indexing failure)
  - shadows
  - ice spots or rings



- The software is 3 decades old, derived from the Cambridge MOSCO system (Nyborg and Wonacott), developed further at Imperial College and then at the LMB
- Written in FORTRAN 5 for a Data General Nova 3/12 computer with 32 kWords of memory and a 10 Mbyte disk drive. Handswitches provided control of program operation at run time.
- The limited memory dictated that processing involved running a series of separate programs, further divided into overlays, with communication provided by external files



The drivers for development were technological advances, and to solve specific problems

- Arrival of VAX computers (virtual memory!)
- Image plates
- More “automated” processing for higher throughput
- CCD detectors
- Improve user-friendliness (GUIs) and software “intelligence”
- Fine  $\phi$  slicing
- Pilatus detectors

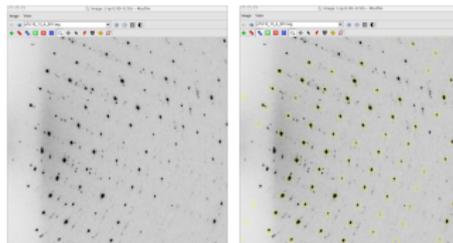


- The implementation has been overly influenced by hardware limitations
- Development has always been an adaptation of the original specification without changing the basic framework
- Particular issues include:
  - The instrumental correction factor for intensity error estimates
  - A non-general coordinate system (the “Cambridge frame”)
  - A confusing change of coordinate system in refinement after autoindexing, for use of REFI code <sup>1</sup>
  - Crystal orientation parameters “mopping up” unmodelled obliquity of the angle between beam and rotation axis
  - Unphysical parameters such as detector TILT and TWIST
  - Three separate parameter refinement routines
  - Strictly 2D integration

---

<sup>1</sup>though this is invisible to the user

- Development of Mosflm continues. Where it can adapt, it will
- Current developments:
  - better handling of Pilatus data (big improvement recently)
  - multiple lattices
  - parallel processing
  - generalised goniometry



New software is desired to provide:

- A physically realistic model of the experiment
- 3D parameterised profile fitting
- Rapid processing fully utilising modern computer hardware, keeping pace with high data acquisition rates
- Challenging cases, e.g. deconvolution of overlaps, handling of highly mosaic crystals
- A modular, extensible architecture, suitable for implementation within pipelines as well as interactively



# Error estimation



personal interest in a new framework for integration software

One particular deficiency of current integration programs is poor modelling of the error of integrated intensities

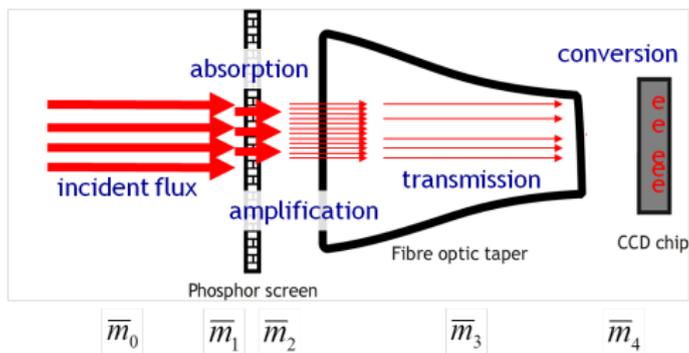
- Non-Poisson statistics of CCD detectors
- Correlations between pixels
- Problems with learned profiles



# Error estimation

## Non-Poisson detector response

- Real detectors have a DQE < 1, which implies Poisson statistics underestimate the true errors
- It is possible to derive a better estimate for a simulated CCD detector based on the physical processes involved in detection



$$\bar{m} = \bar{m}_0 \bar{m}_1 \bar{m}_2 \bar{m}_3 \bar{m}_4$$

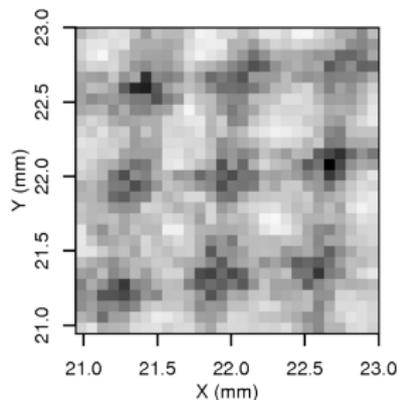
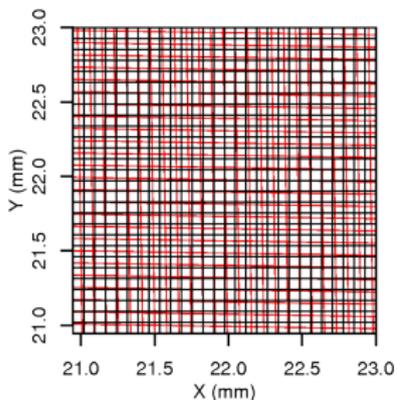
$$\sigma_m^2 = \bar{m}_0 \bar{m}_1 (\bar{m}_2 \bar{m}_3)^2 \left( 1 + \frac{\sigma_{m_2}^2}{\bar{m}_2^2} + \frac{1 - \bar{m}_3}{\bar{m}_2 \bar{m}_3} \right)$$

# Error estimation

## Non uniform correlation patterns



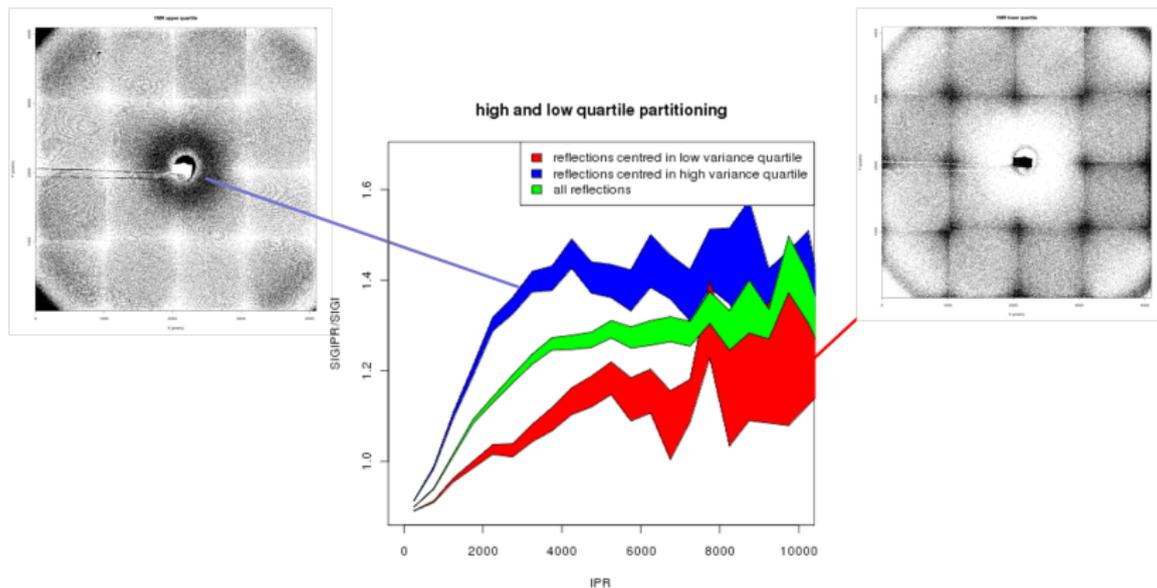
The distortion correction of CCD detectors introduces a Moiré pattern visible on flat field scattering images



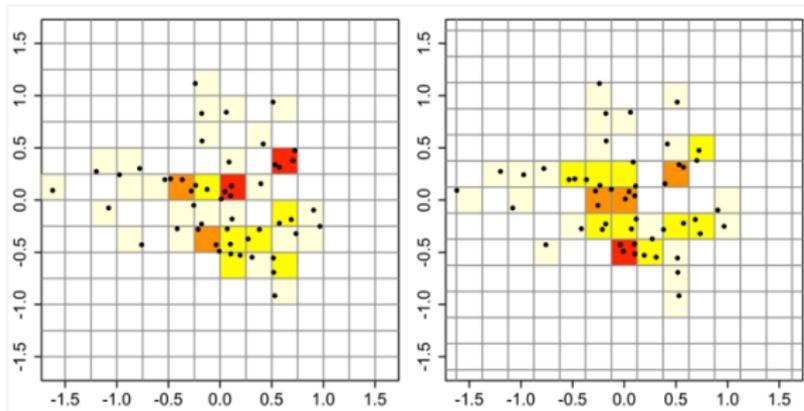
# Error estimation

## Non uniform correlation patterns

This leaves a detectable signature in profile fitting error estimates



- Pixellating detectors do not sample a signal, they average in bins (C. Nave)
- The appearance of histograms depends on the anchor points



- Rather than learned profiles we'd prefer to use a parameterised model to construct profiles
- This addresses the problem of profile anchor points
- It also provides a way to account for errors properly even in the presence of correlations



The end

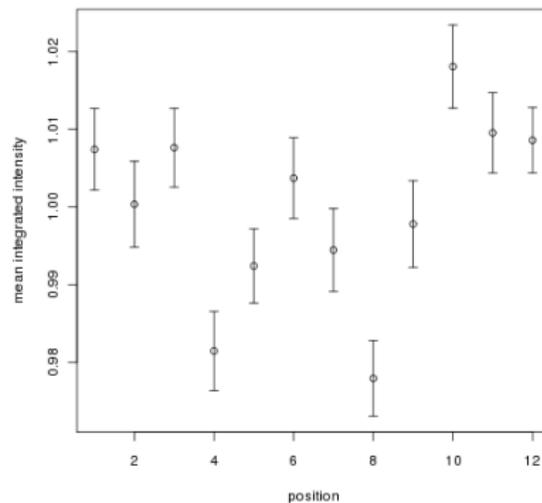
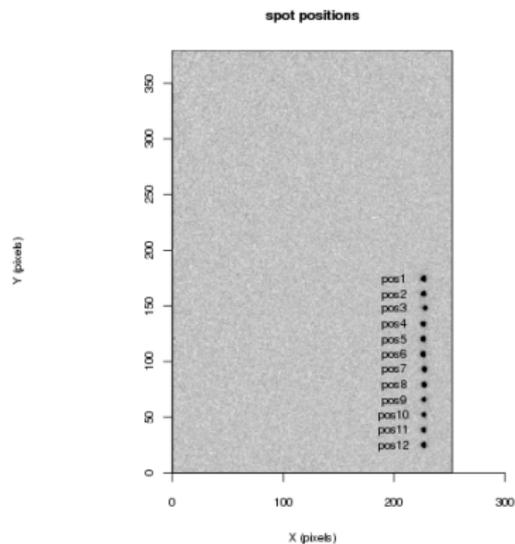


# Spatial noise

with James Holton



Position-dependent detector response causes a significant systematic error in the measured intensity



# Spatial noise

with James Holton



The spatial period of variation appears to be on the scale of a pixel. Could the cause be phosphor inhomogeneities?

