XPCS Simulations

The try to tame the monster

Julian Becker, DESY (FS-DS)

AGIPD Meeting, 8.3.2011

Outline

- What is XPCS, what can be done with it?
- Detector demands of the community
- Simulating XPCS
- Data evaluation
- Results
- Outlook on the next generation of simulations

What is XPCS

- X-ray Photon Correlation Spectroscopy
- Extension of PCS with optical light to opaque samples and smaller length scales
- Extremely successful in biological applications, esp. the study of membrane proteins
- Allows to extract protein size, concentration and interaction dynamics

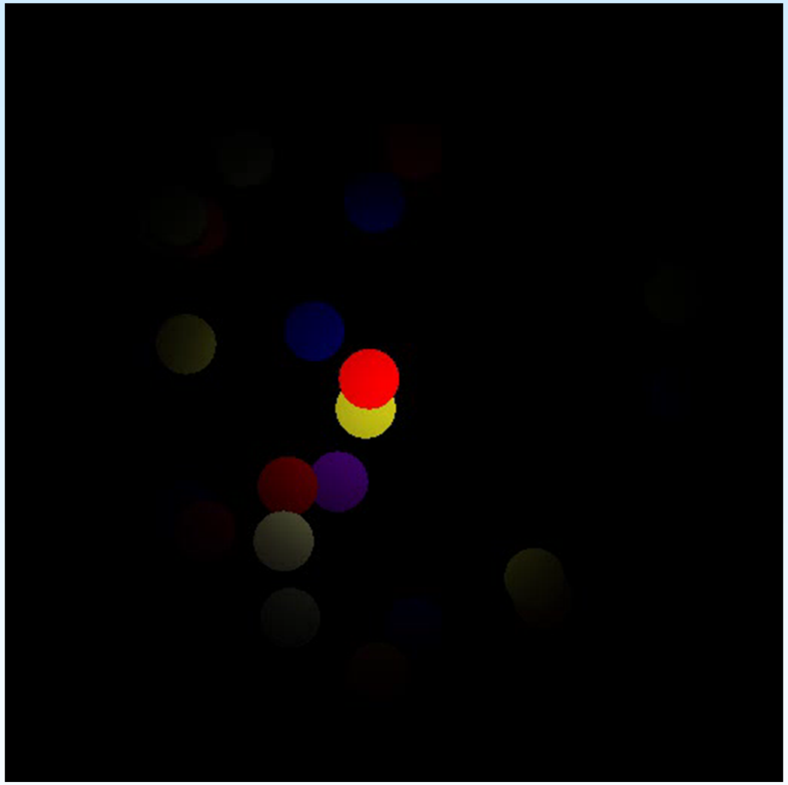
What is XPCS

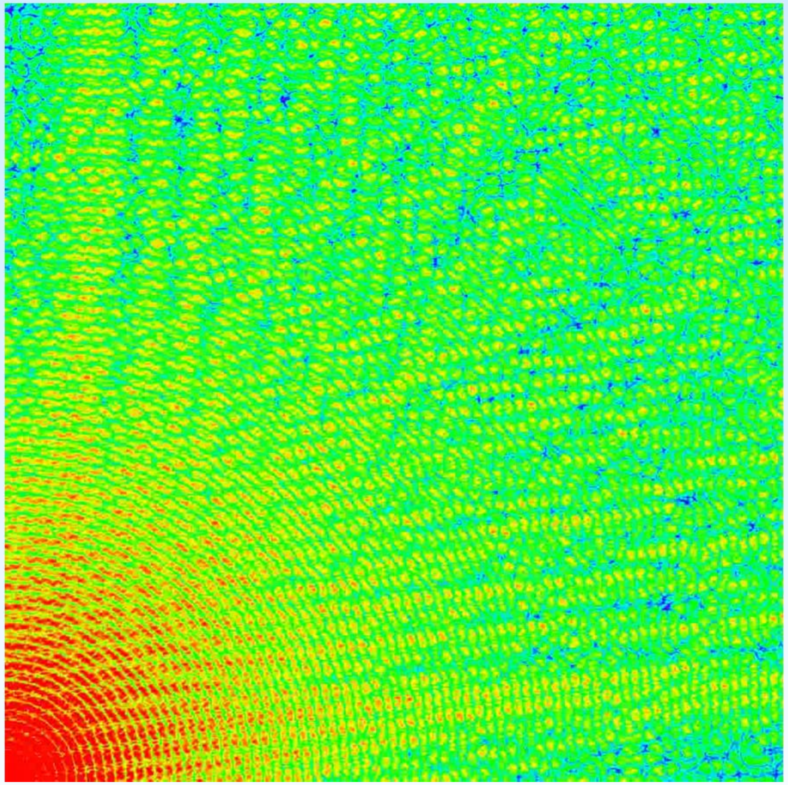
XPCS is a young technique but has already shown the potential to impact several areas of statistical physics and provide access to a variety of important dynamic phenomena.

Among them are the time-dependence of equilibrium critical fluctuations and the low frequency dynamics in disordered hard (e.g. non-equilibrium dynamics in phase separating alloys or glasses) and soft condensed matter materials, in particular complex fluids (e.g. hydrodynamic modes in concentrated colloidal suspensions, capillary mode dynamics in liquids and layer-fluctuations in membranes, equilibrium dynamics in polymer systems).

G. Grubel, F. Zontone, Correlation spectroscopy with coherent X-rays, J. Alloys and Compounds, DOI: 10.1016/S0925-8388(03)00555-3.

What XPCS does

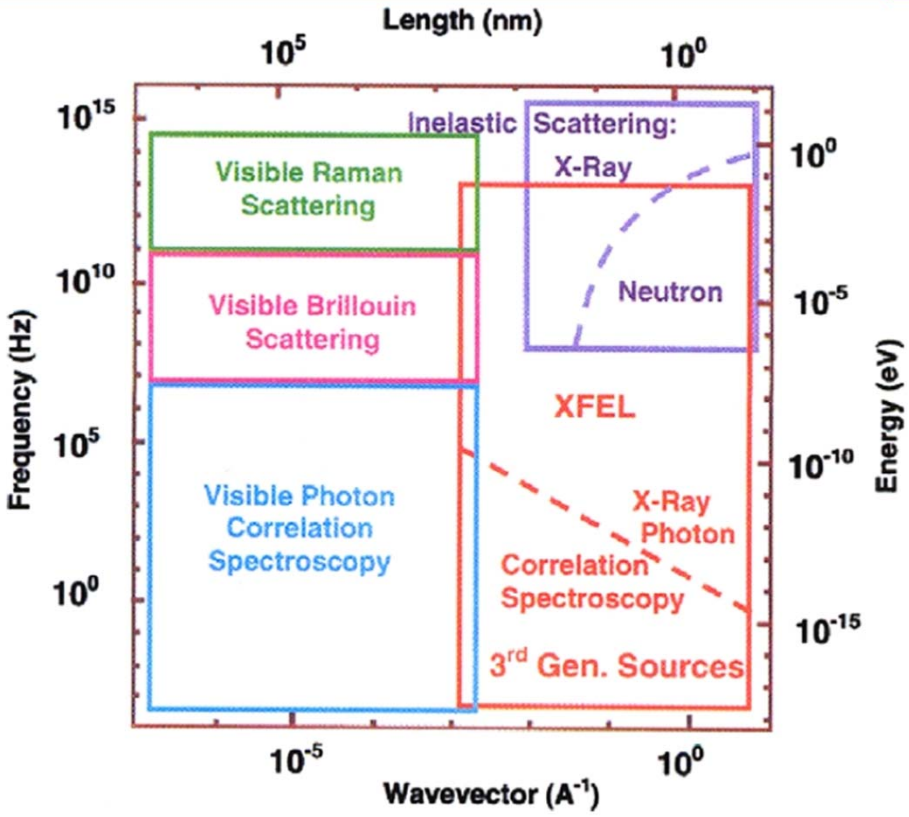




- Investigation of fluctuations in diffraction images
- Scientific case XPCS@XFEL: molecular dynamics in fluids, charge & spin dynamics in crystalline materials, atomic diffusion, phonons, pump-probe XPCS

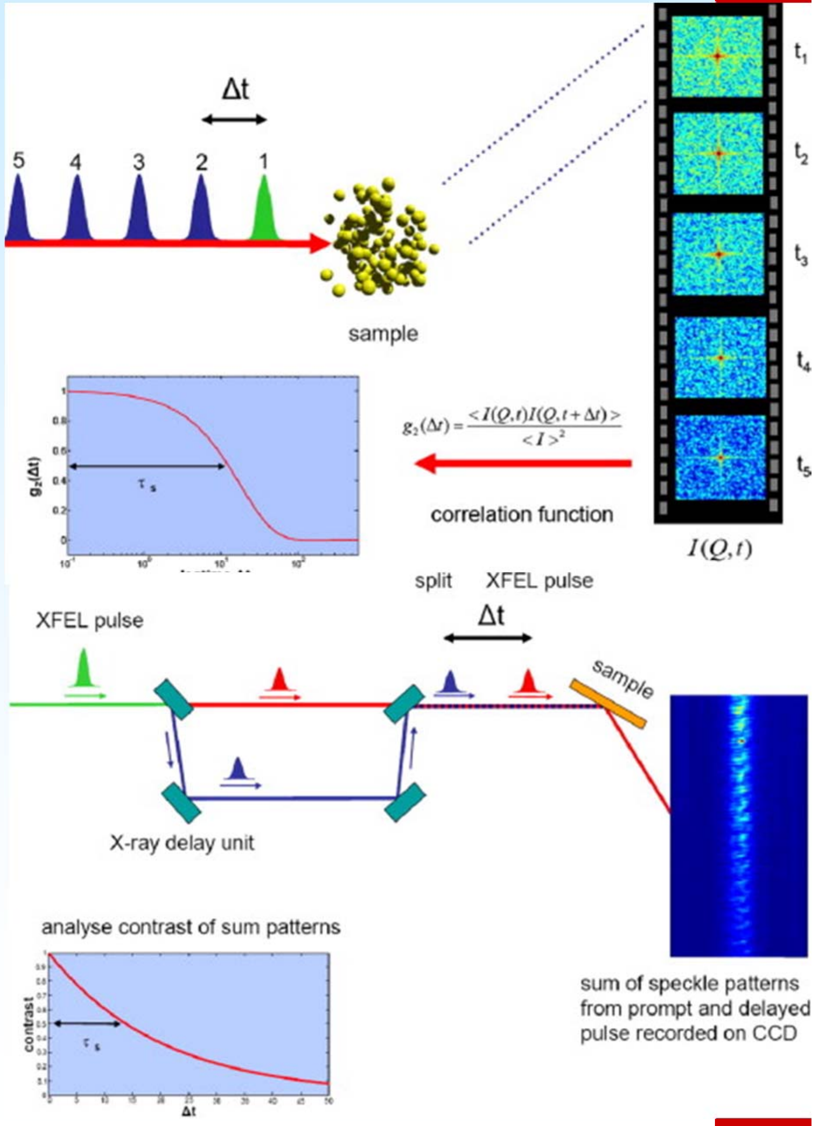
Some possible results from investigations with XPCS

- Insight into interaction type at probed length/time scales
- Determination of associated time constants
- Determination of anisotropies
- Investigation of phase transitions (esp. glassy states)
- Determination of rare symmetries (XCCA)
- … and much more



Different ways of XPCS@XFEL

- Choice of technique governed by investigated time scale τ
  - 0.22 µs < τ << 0.6 ms
    - intensity autocorrelation function (g2)
    - problems for low intensities, cannot correlate ‘0’ to anything
    - ‘slow’ time scale -> large particle movement -> low Q region -> SAXS
  - τ << 10 ns
    - use split pulse technique
    - problems for low intensities, offset value ~1/<I>
    - ‘fast’ time scale -> small particle movement -> large Q region -> WAXS
- For very low intensities (<I> -> 0) photon statistics have to be analyzed



Constrains for performing XPCS

- X-rays must be coherent
- Sample must survive multiple XFEL shots (undisturbed)
- Noise intensity must be smaller than signal intensity
- Speckle size ≈ O(effective pixelsize)

$$\sigma_{speckle} \approx \frac{\lambda L}{\sigma_{ill}} \propto \frac{L}{E\sigma_{ill}}$$

$\sigma_{speckle}$: speckle size

λ, E: wavelength, energy of the x-rays

L: distance between sample and detector

σ_{ill}: size of the illuminated area of the sample

Detector demands for XPCS

- Many (10^8 or more!) small pixels (100µm or less, 10µm preferred)
- Fast readout (record every frame)
- Large number of frames (some reduction possible with log spacing)
- Additional in-pixel logic to do multi-tau correlation on chip (not for XCCA)

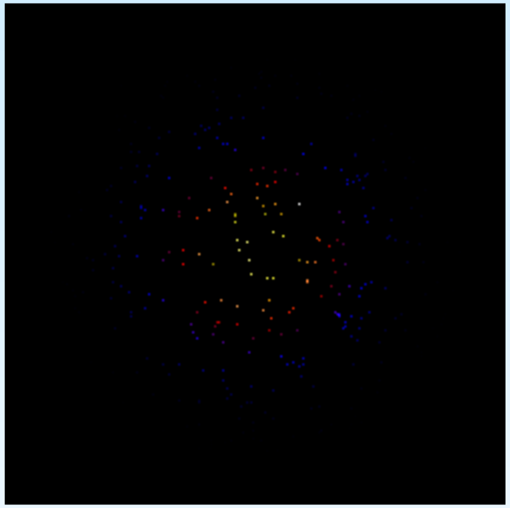
$$R_{sn} \propto C < I_{pix} > \sqrt{N_b N_f N_{pix}}$$

How to simulate XPCS

- Take a simple test system and generate a series of diffraction patterns
- Simulate detector response as function of relevant parameters
- Evaluate simulated detector images with established and foreseen techniques
- Quantify and compare results

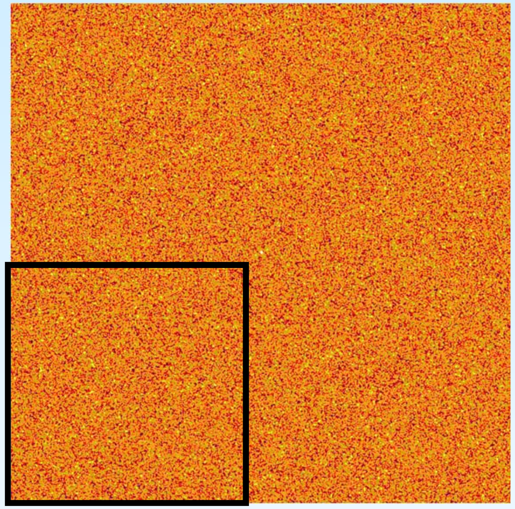
XPCS-Simulations

Real space



FFT

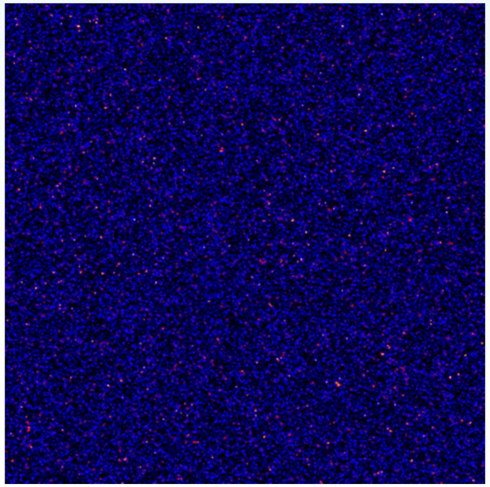
‘Diffraction image‘



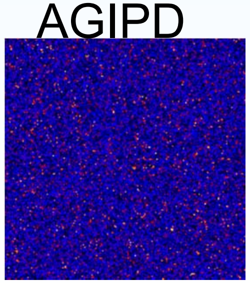
Detector response

(HORUS)

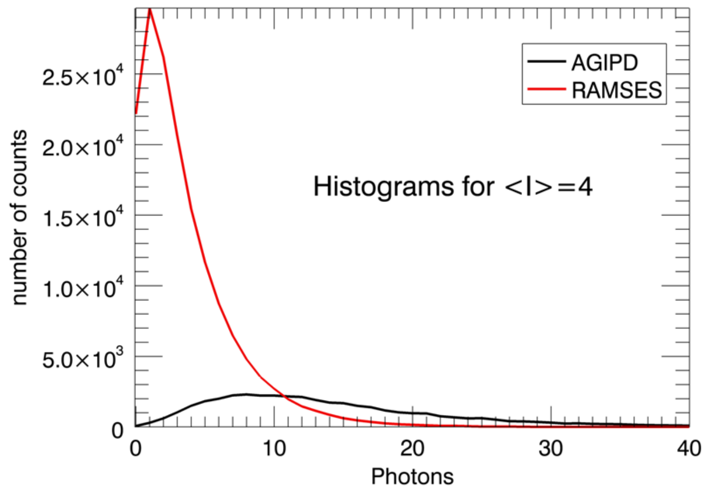
100µm AGIPD (RAMSES)



AGIPD



Evaluation

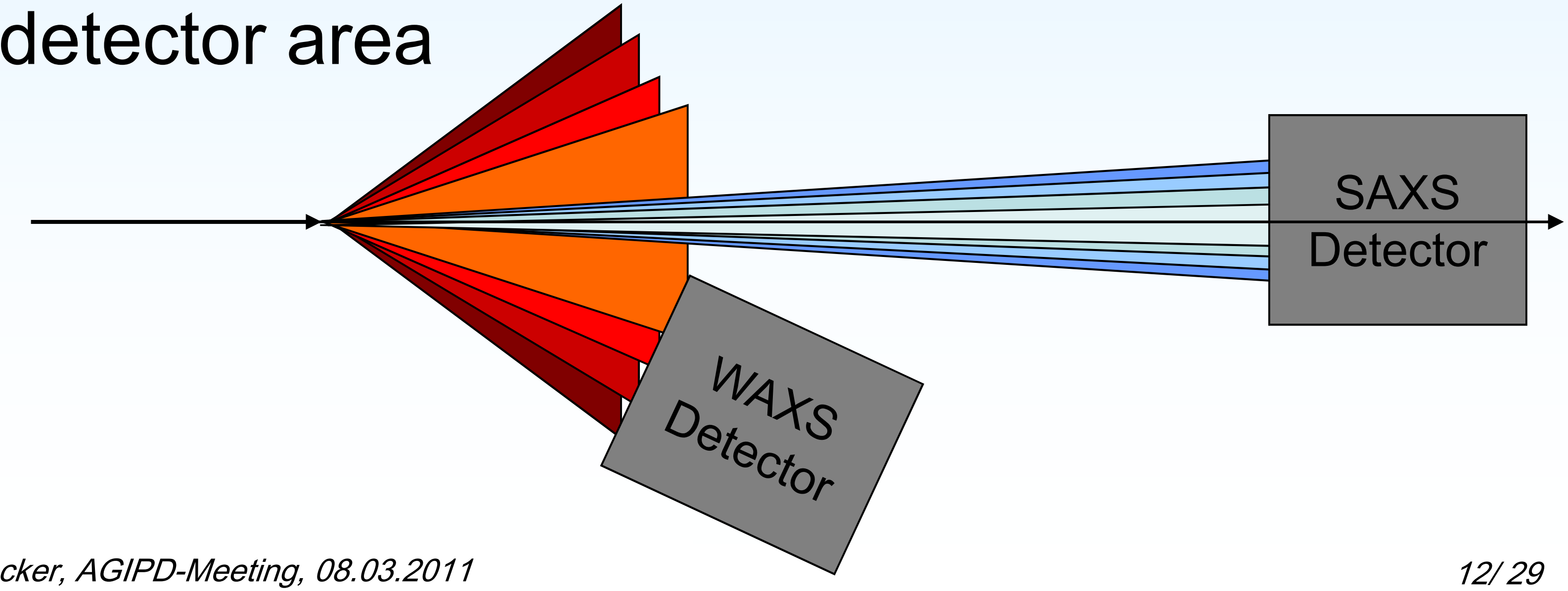


Detector Systems

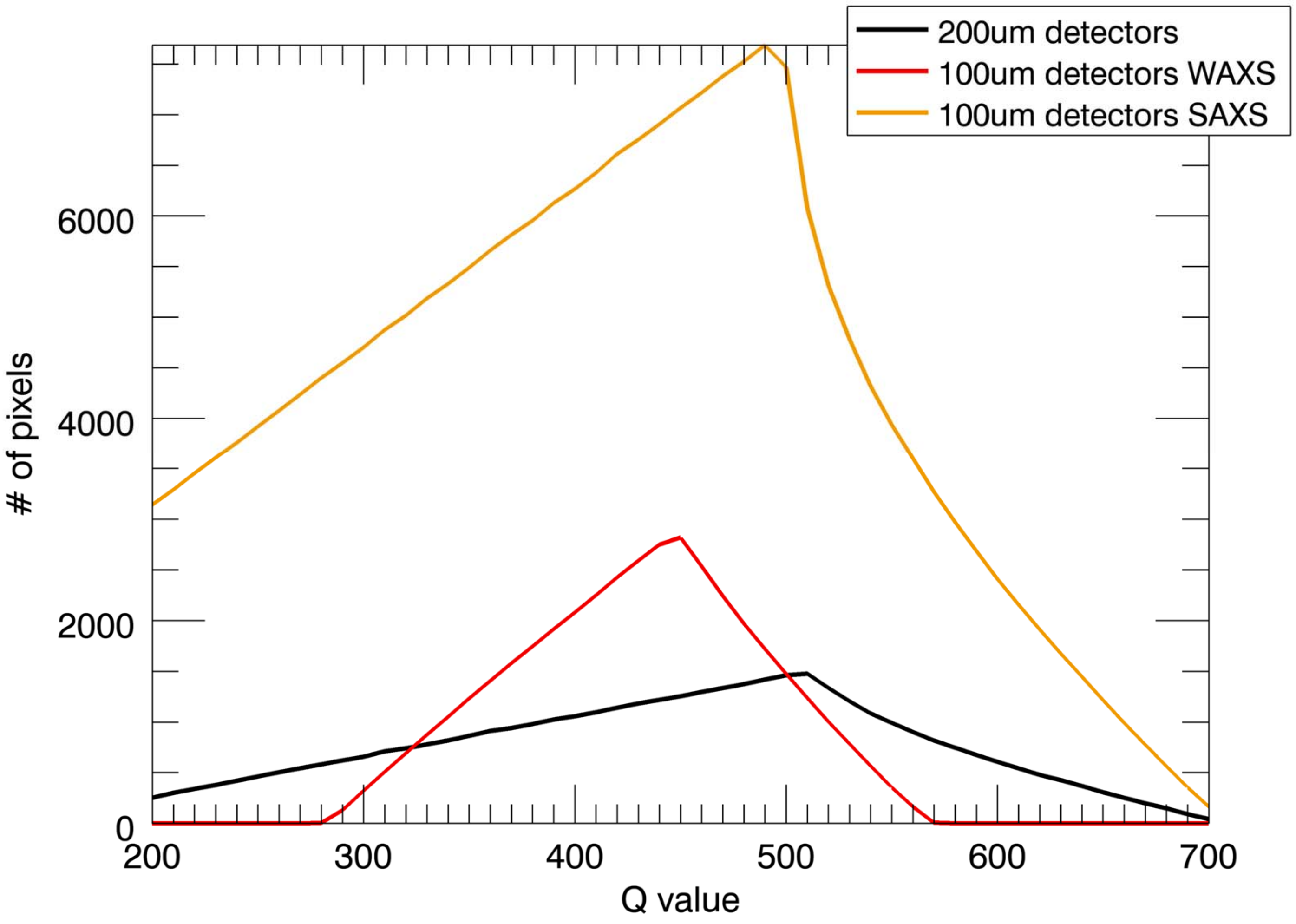
- ***Ideal***: 100µm / 200µm pixel size (no charge sharing, QE=1, no noise counts)
- ***AGIPD***: 200µm pixel size
- ***MAAT***: Modified AGIPD using Aperturing Technique, 200µm pixels apertured to 100µm
- ***RAMSES***: Reduced AMplitude SEnsing System, AGIPD with 100µm pixel size
- WAXS/SAXS configuration for 100µm systems

Detector Geometries

- SAXS: interesting Q region fits on detector area -> limiting factor: pixel density
- WAXS: only small part of the interesting Q region can be sampled -> limiting factor: detector area



Pixel distribution



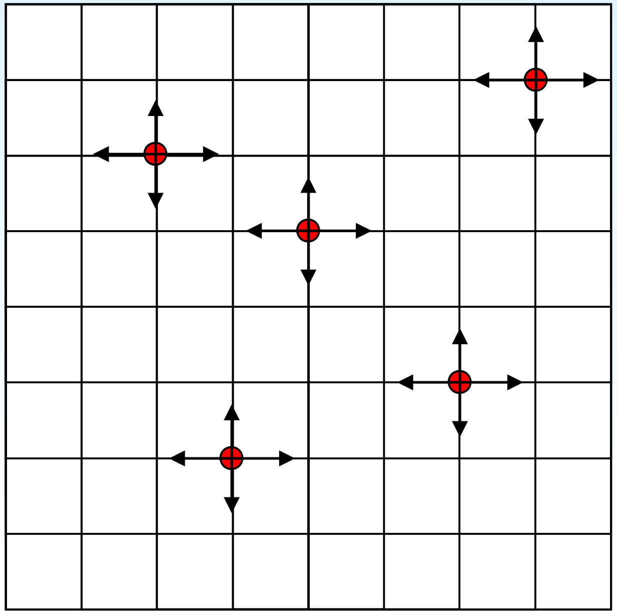
Evaluations as function of Q

Radial symmetry in Q-space allows averaging over pixels with similar Q (5)

Detector is a square, thus the number of pixels as function of Q shows a distinctive shark fin shape

Simple real space system

- Points hopping on a 2D grid by 1 position in each dimension (jump-diffusion)
- Absence of structure factor due to delta-like points
- Gaussian ‘illumination function’ producing Gaussian speckles with 4σ=2 pixels
- Oversampled ‘Diffraction’ image by Fourier transform (non-integer values)



Simulated noise sources

- Photon quantization noise (convert e.g. 1.2 γ to an integer number)
- 10% rms (uncompensated) intensity fluctuations
  - – Probably more at low intensities (inherent non-Gaussian SASE fluctuations)
  - – Probably less at high intensities (can be corrected for)
- Incoherent background noise (e.g from higher harmonics, sample fluorescence, residual gas scatter, etc.): completely random, probability of 1/100 (Poisson distributed) per 100µm pixel

Parameter space

- 13 different intensities (4e-4 to 40)
- 7 detector systems
- 4 sets of noise contribution
- 300 images per set
- 5 repetitions

=> O(10^6) simulations / evaluations

Data evaluation

- Calculate intensity autocorrelation function (g2) per pixel
- Sequential mode (constant Δt between frames)

$$g2(n,k) = \frac{1}{F-k}\frac{1}{<I_n>^2}\sum_{i=1}^{F-k} I_n(t_i)I_n(t_i + k\Delta t)$$

n: number of the individual pixel, identifying Q vector Q(n)

k: integer number, identifying lag time τ=kΔt

F: number of acquired frames

<I_n>: average (over all frames) pixel value

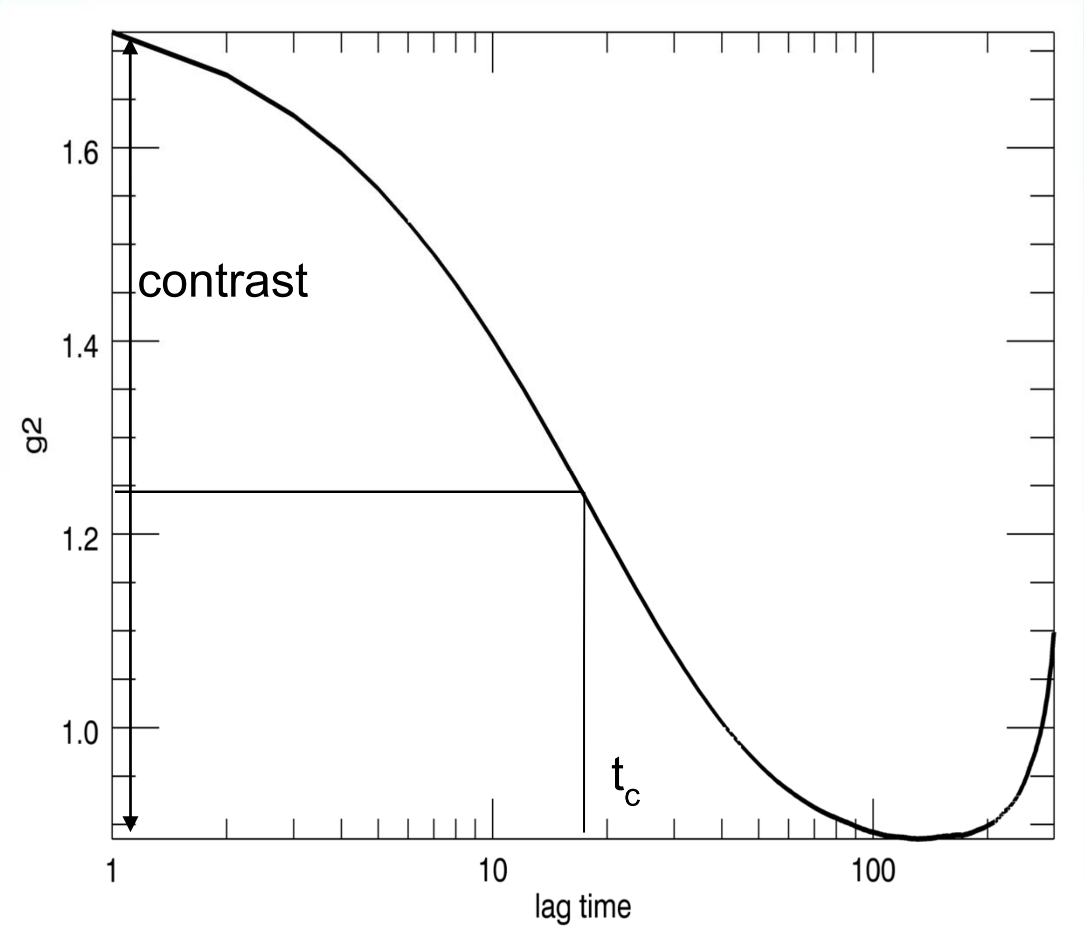
Data evaluation

- Average g2 values with identical Q (azimutal average)
- Fit exponential decay to resulting g2* function
- Extract fit parameters as function of Q
- Calculate average fit parameters and (relative) errors

$$g2^*(Q,\tau) = S(Q)\left(C(Q)e^{-\frac{\tau}{\tau_c(Q)}} + 1 \right)$$

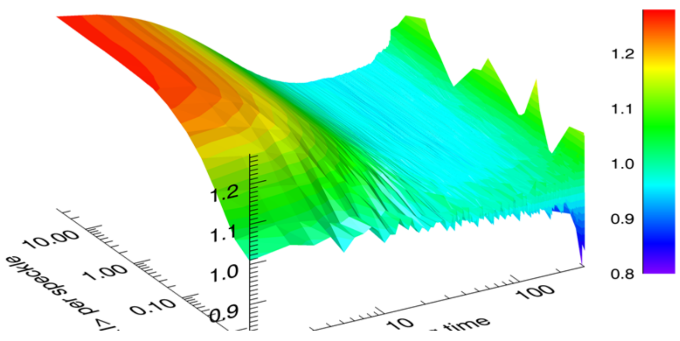
$$\Gamma(Q) = {}^{1}\!/_{\tau_c(Q)} \propto Q^2$$

G2 function

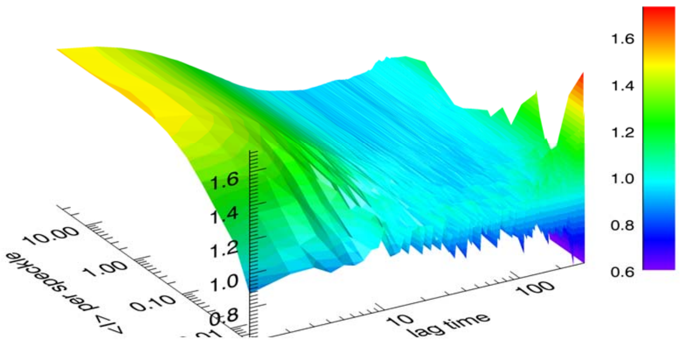


- Decaying (ideally) from contrast+1 to 1 with decay time t_c
- Artifacts toward large lag times are reduced by more frames (100x - 1000x t_c)
- Functional form determined by particle interactions

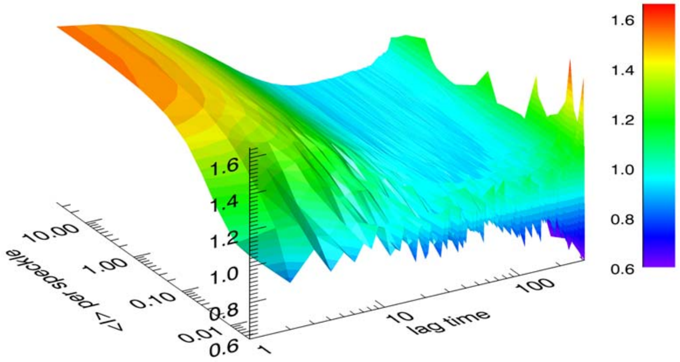
G2 function, AGIPD, incoherent noise and intensity fluctuations



G2 function, RAMSES WAXS, incoherent noise and intensity fluctuations



G2 function, MAAT, incoherent noise and intensity fluctuations



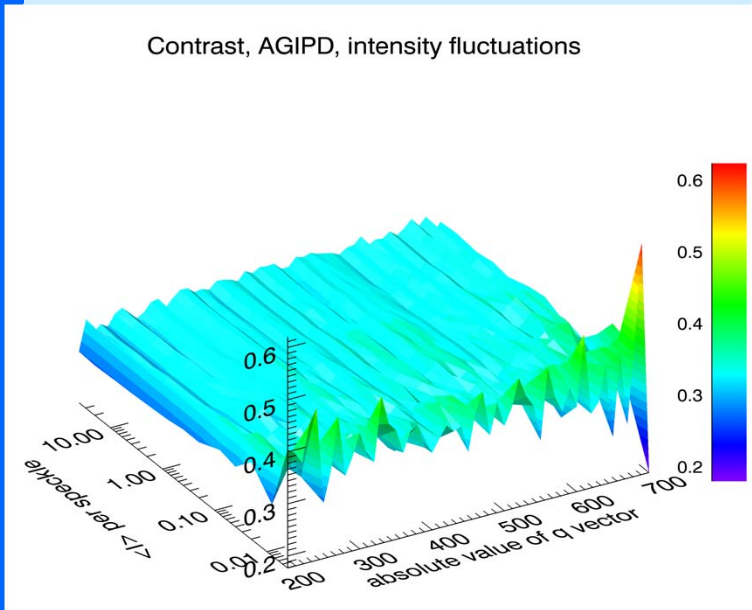
G2 at Q=500

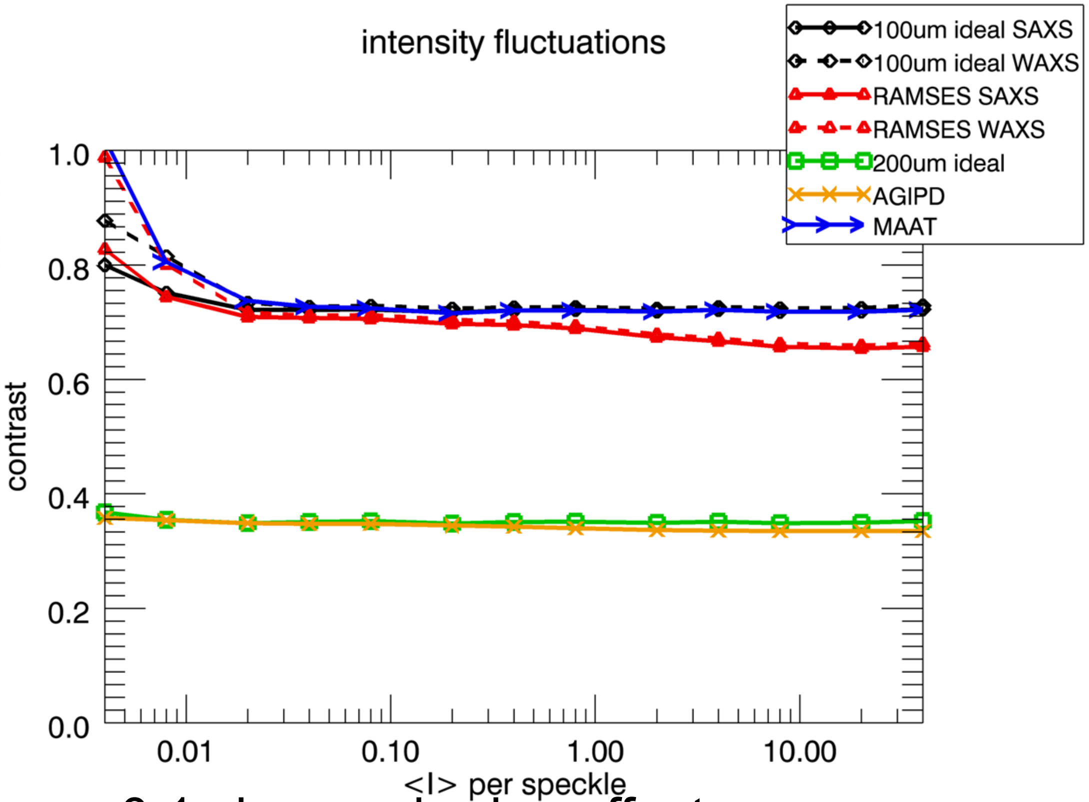
Basic data set to be fitted

- For AGIPD contrast is low, but lowest noise
- RAMSES in WAXS shows higher contrast and higher noise
- For MAAT contrast is as high as for RAMSES with similar noise

In the following slides only the results of the fit will be shown

Contrast with fluctuations



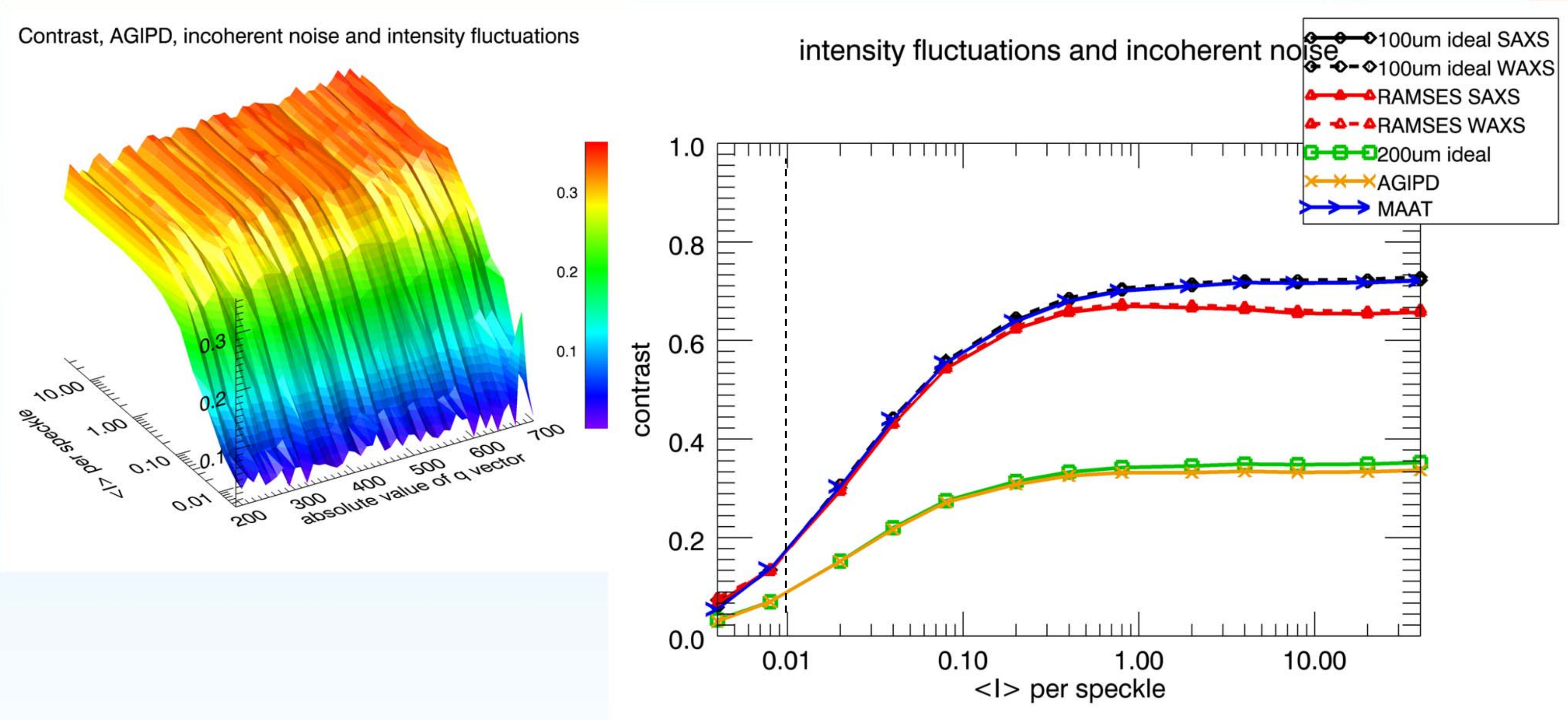


At average intensities above 0.1 charge sharing effects decrease the contrast, less strong for bigger pixels

At very low intensities the number of pixels/frames/bunches is not high enough for reliable results

MAAT yields contrast of an ideal 100um system

Contrast with noise

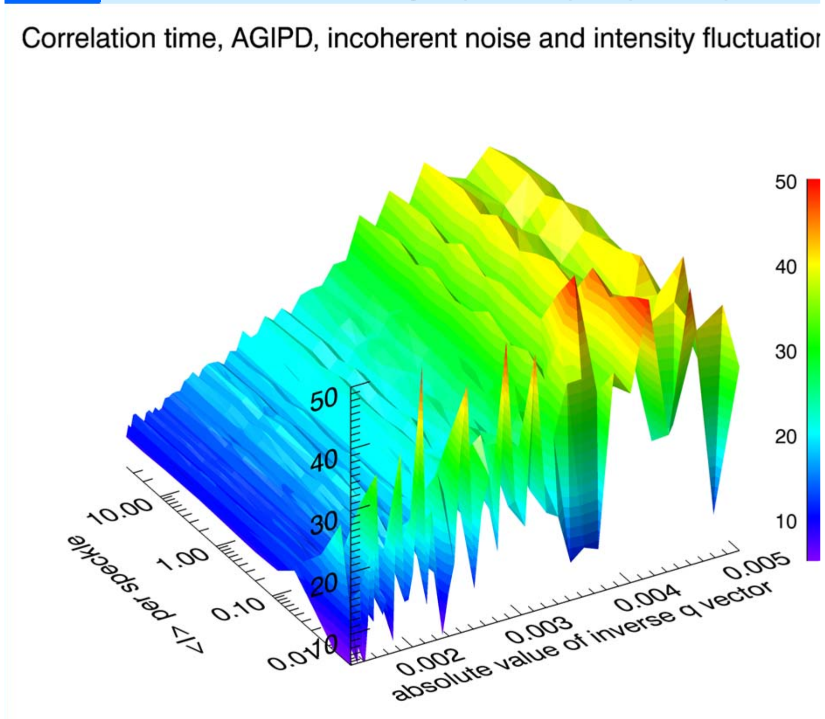


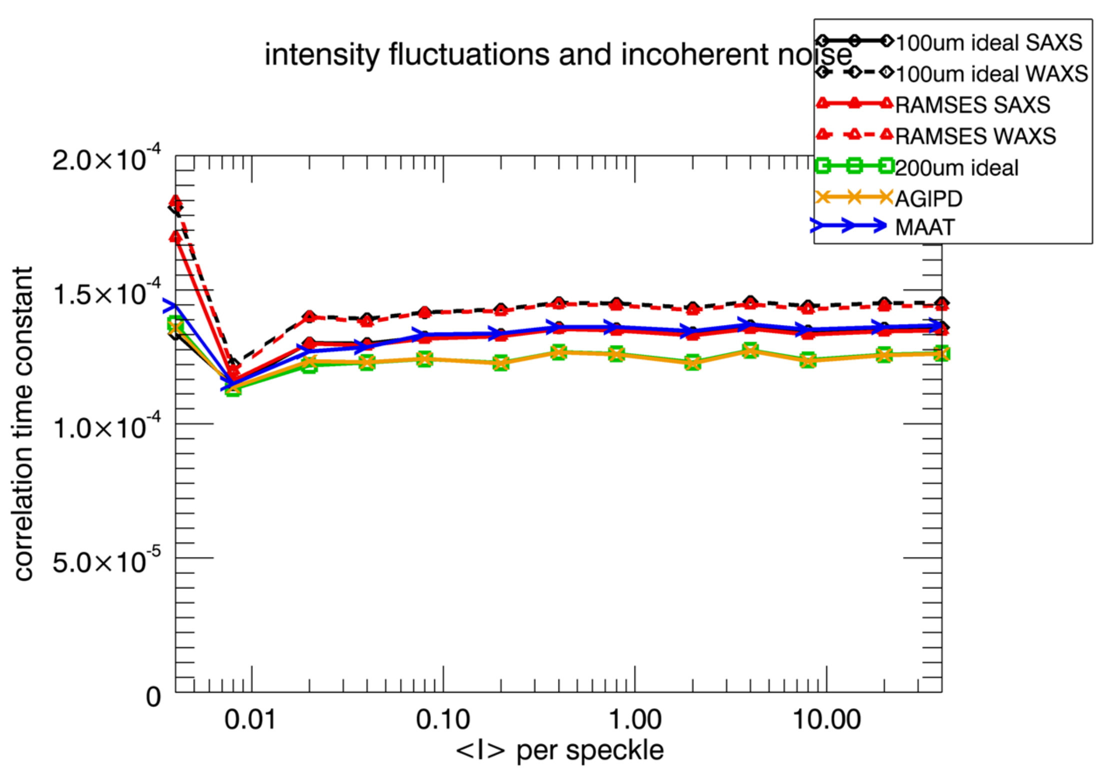
Charge sharing independent of noise

Contrast significantly decreases around the average intensity of the incoherent noise ($<I_{noise}>=0.01$)

MAAT still yields contrast of an ideal 100um system

Correlation time





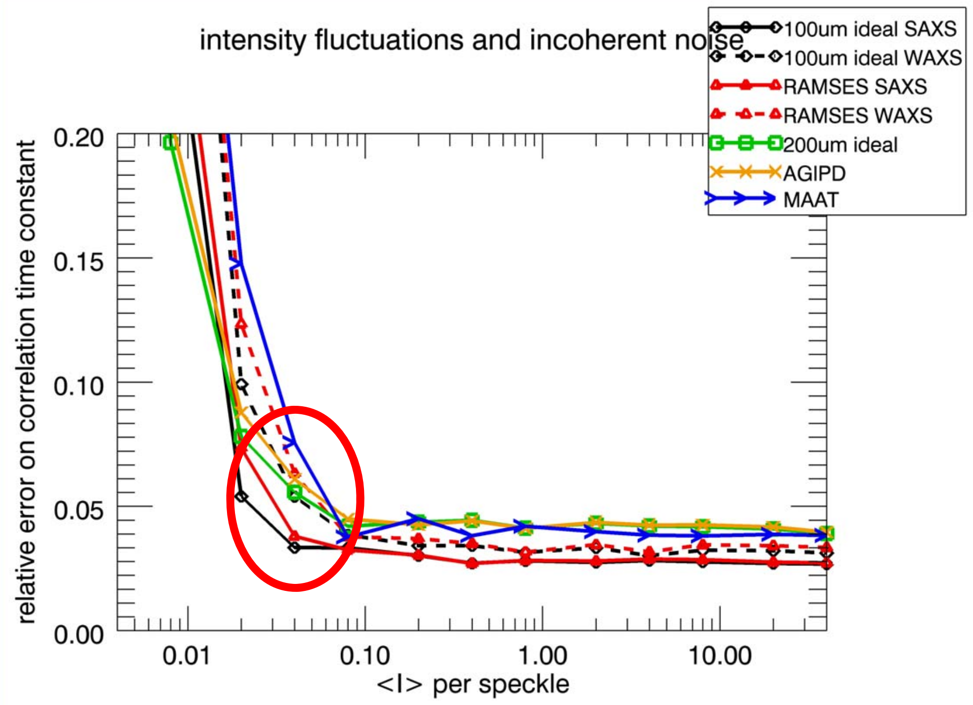
$\Gamma(Q)=(1/t_c)$ should be proportional to Q^2 for small Q and show distinct deviations from this when Q is in the region of the inverse lattice size

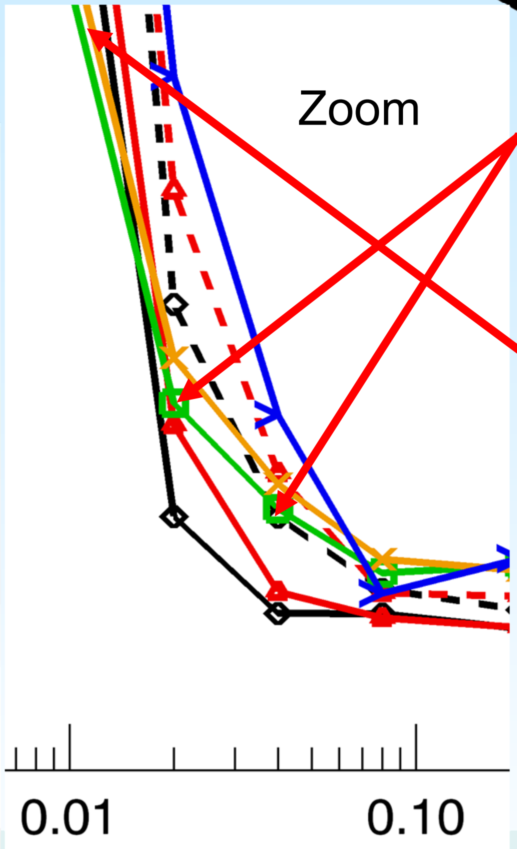
Correlation time is linear in 1/Q (crude approximation for this case)

Slightly different slope for different systems (due to crude approximation)

Error on correlation time with incoherent noise

Optimum Q range for each system





Green/Yellow below WAXS lines

Even below SAXS lines!

Bigger pixels win! (LPD?)

Intensities below <I>=0.01 require more images/bunches/pixels (seen from contrast)

Crossing behavior -> statistical effect: cannot correlate 0 photons to anything, higher fraction of non-zero pixels for larger pixel size

At low intensities MAAT (blue) as good/bad as 100µm systems in WAXS geometry, minor advantage over AGIPD (due to absence of charge sharing) at high intensities

Split pulse and other evaluation techniques

- Data for split pulse technique has been calculated and evaluated
  - calculation of 5 images each at 300 different Δt
  - not enough statistics to evaluate performance
  - even at high intensities
- Evaluation using photon statistics (# of 0’s, 1’s, 2’s, etc.) underway

Summary: XPCS

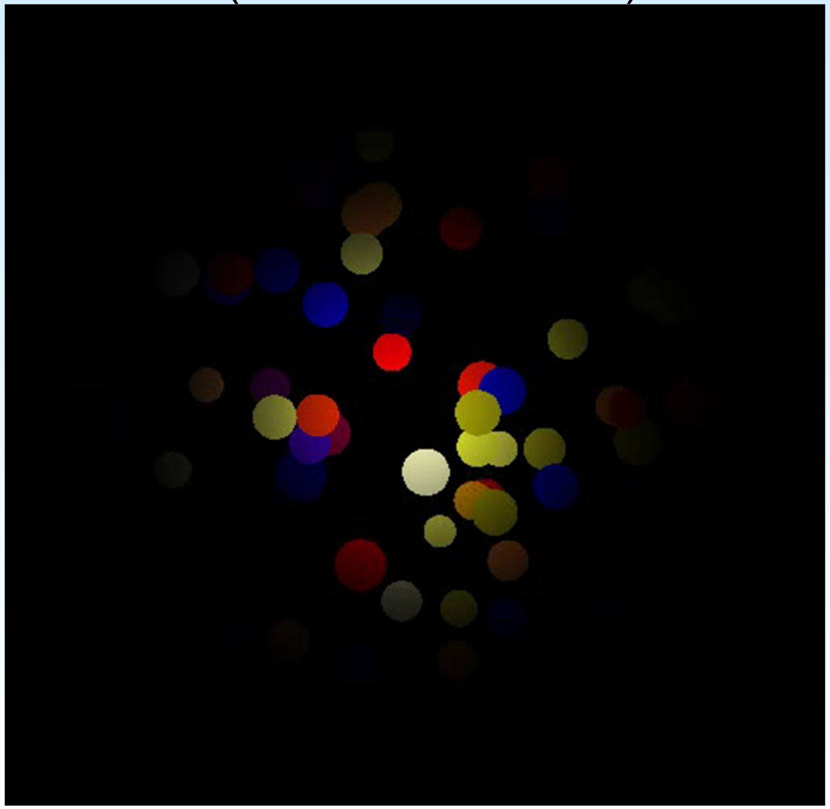
- Whole simulation chain set-up and tested
- Extraction of parameters allows comparison of different systems
- At high intensities (SAXS, lim. by pixel density):
  - MAAT yields higher contrast compared to AGIPD
    - smaller speckles
    - less focused x-rays
    - less beam damage
    - can cope with high intensities
  - RAMSES shows superior performance
    - amplitude limitation
- At low intensities (WAXS, lim. by pixel number):
  - AGIPD outperformes other systems
    - larger area (Q-space) coverage
    - better statistics due to higher non-zero probability
  - RAMSES and MAAT show equal performance

Next Generation XPCS simulations

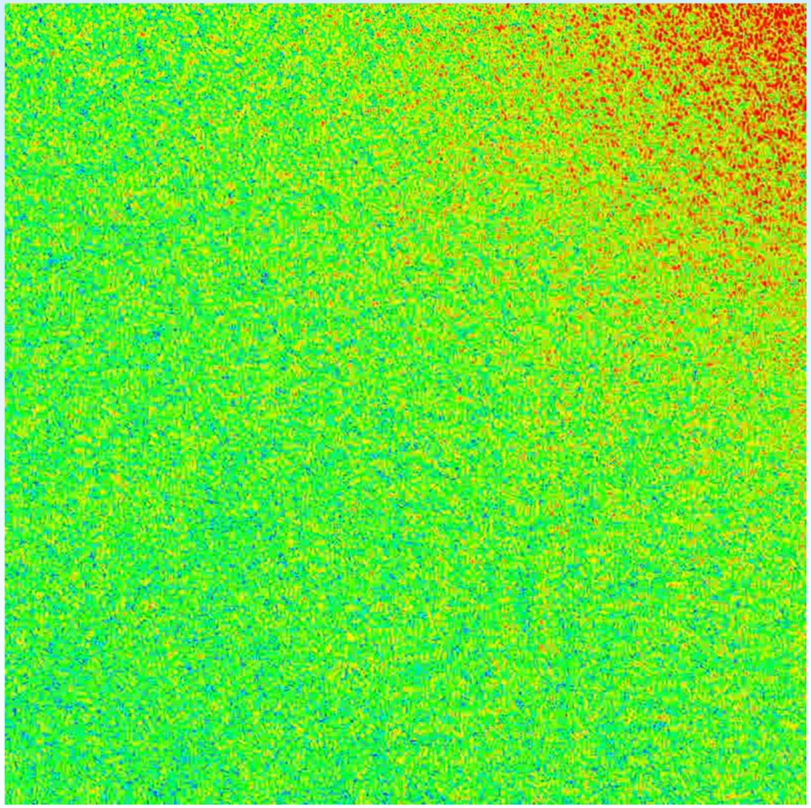
- Simulate a more realistic system
  - Charge stabilized colloids
    - 3D Diffusion
    - 3D Volume -> path length difference
    - Repulsive screened Coulomb force (Yukawa potential)
    - Finite extend of particles -> Structure factor
  - Based on PhD Thesis of Fabian Westermeier
- Concentrate on interesting region of phase space (high intensities take long to calculate)
- Calculate enough statistics to evaluate split pulse technique

Next generation XPCS simulations

Real space
(z axis color coded)



Detector plane
($\log_{10}$(intensity) color coded)



- All simulations in arbitrary units -> normalization constants
- Need to find right parameters to simulate a realistic system

Thank you for your attention

CDI simulations

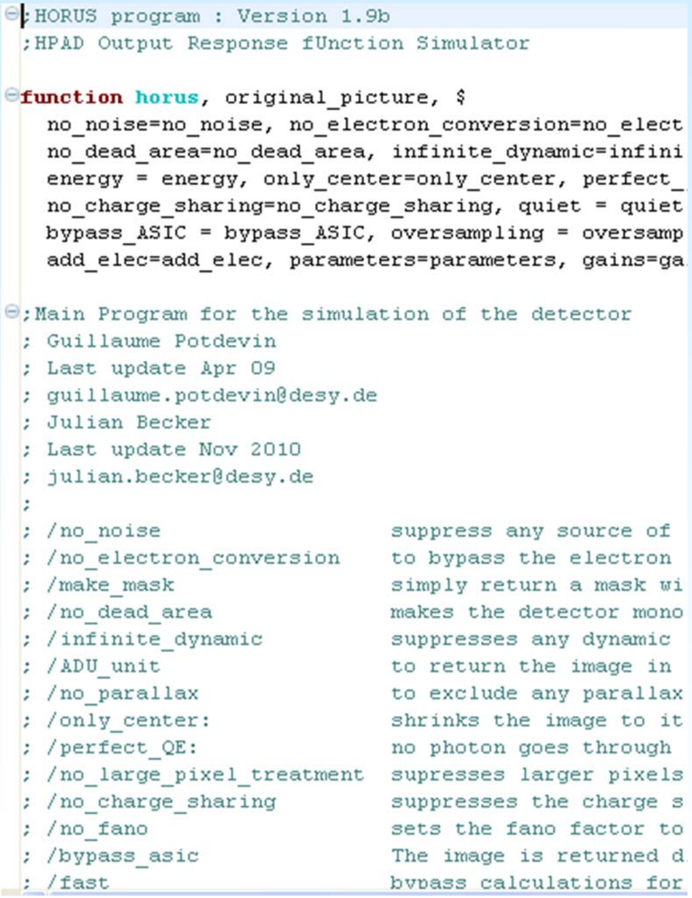
- In principle all tools to calculate CDI are there
- Proper input systems are needed (Lysosyme?)
- Reconstruction algorithms need to be implemented and some automation added
- No progress so far due to lack of knowledge (and time)
- Next big topic on the list

Outline

- HORUS
  - What's that?
  - Usage and simulation steps
  - Approximations, limitations, etc.
- XPCS simulations
  - XPCS what's that?
  - Simple test system
  - Parameter space
  - Data evaluation
  - Results
- Outlook on next generation of simulations

What is HORUS?

- HORUS stands for:
  Hpad **O**utput **R**esponse
  f**U**nction **S**imulator
- Collection of IDL routines
- Designed to evaluate influences of certain design choices for AGIPD
- Expanded to allow simulations of photon counting detectors (Medipix3) by D. Pennicard



```
;HORUS program : Version 1.9b
;HPAD Output Response fUnction Simulator

function horus, original_picture, $
  no_noise=no_noise, no_electron_conversion=no_elect
  no_dead_area=no_dead_area, infinite_dynamic=infini
  energy = energy, only_center=only_center, perfect_
  no_charge_sharing=no_charge_sharing, quiet = quiet
  bypass_ASIC = bypass_ASIC, oversampling = oversamp
  add_elec=add_elec, parameters=parameters, gains=ga

;Main Program for the simulation of the detector
; Guillaume Potdevin
; Last update Apr 09
; guillaume.potdevin@desy.de
; Julian Becker
; Last update Nov 2010
; julian.becker@desy.de
;
; /no_noise                   suppress any source of
; /no_electron_conversion     to bypass the electron
; /make_mask                  simply return a mask wi
; /no_dead_area               makes the detector mono
; /infinite_dynamic           suppresses any dynamic
; /ADU_unit                   to return the image in
; /no_parallax                to exclude any parallax
; /only_center:               shrinks the image to it
; /perfect_QE:                no photon goes through
; /no_large_pixel_treatment   supresses larger pixels
; /no_charge_sharing          suppresses the charge s
; /no_fano                    sets the fano factor to
; /bypass_asic                The image is returned d
; /fast                       bypass calculations for
```

What HORUS is not:

- Full Scale Monte Carlo Simulation
  - Pseudo analytical treatment of charge transport
  - Simplifying assumptions on sensor geometry
  - No simulation of surrounding material (Bumps/ASIC/Module mechanics)
- Tested with real detectors
  - Results might be slightly off, but tendencies should be right
- Bug free
  - Most major bugs are fixed
  - Works as designed, passed many consistency checks
  - …but you never find the last one

Design of HORUS

- HORUS is designed as a transparent end-to-end simulation tool:
  - Needs ‚input image‘ containing the number of photons in each pixel
  - Provides an output image, i.e. the number of detected photons in each pixel
  - Simulation parameters/behavior can by adjusted by the user
  - Additional functionality with special options/workarounds

The HORUS-GUI

Options | Images | Histograms

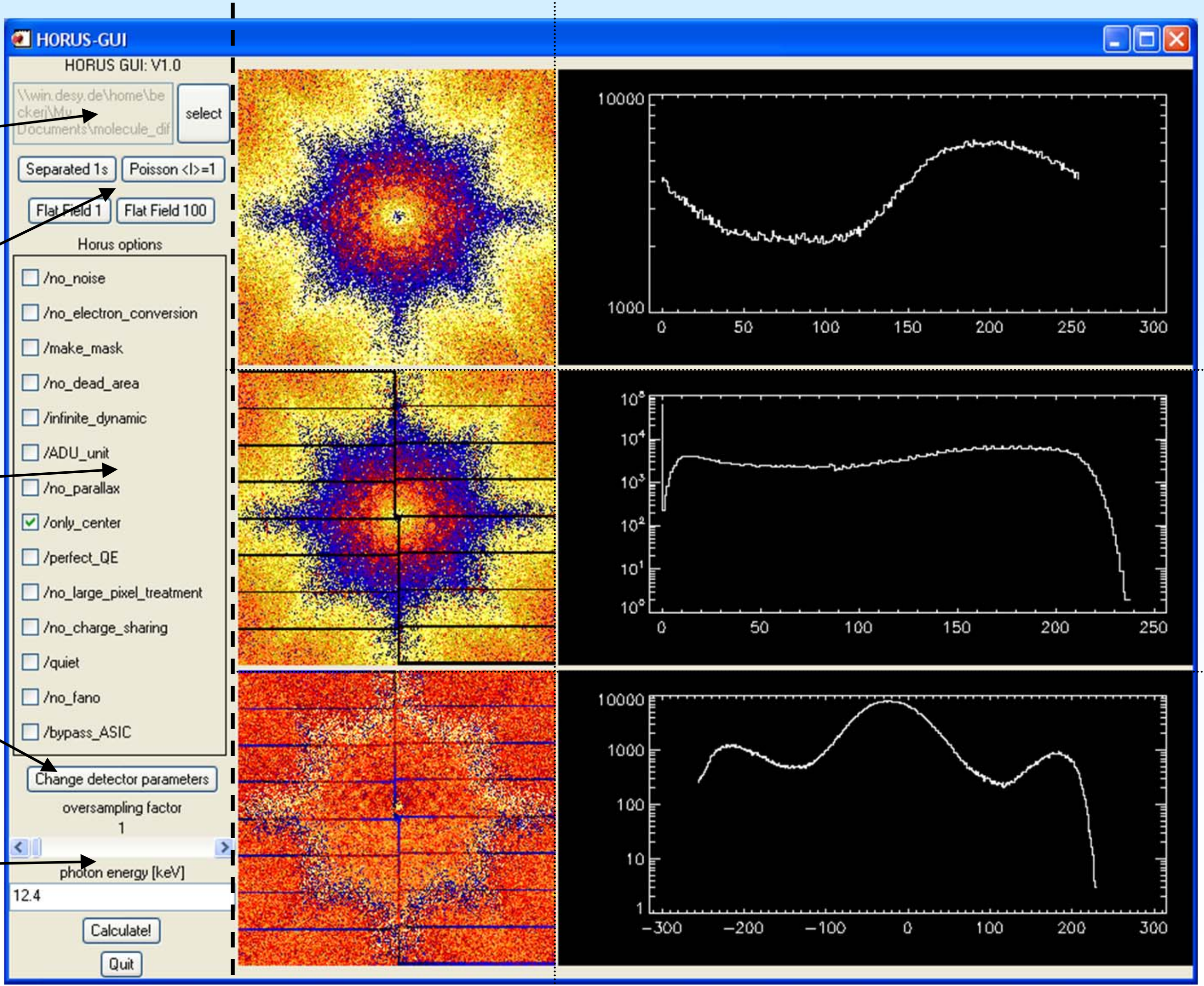
Input image selector

Generate standard patterns

Selector for simulation parameters

Change of detector parameters

Additional options



HORUS-GUI

HORUS GUI: V1.0

\\win.desy.de\home\beckeri\My Documents\molecule_dif

select

Separated 1s | Poisson <I>=1

Flat Field 1 | Flat Field 100

Horus options

- ☐ /no_noise
- ☐ /no_electron_conversion
- ☐ /make_mask
- ☐ /no_dead_area
- ☐ /infinite_dynamic
- ☐ /ADU_unit
- ☐ /no_parallax
- ☑ /only_center
- ☐ /perfect_QE
- ☐ /no_large_pixel_treatment
- ☐ /no_charge_sharing
- ☐ /quiet
- ☐ /no_fano
- ☐ /bypass_ASIC

Change detector parameters

oversampling factor: 1

photon energy [keV]: 12.4

Calculate!

Quit

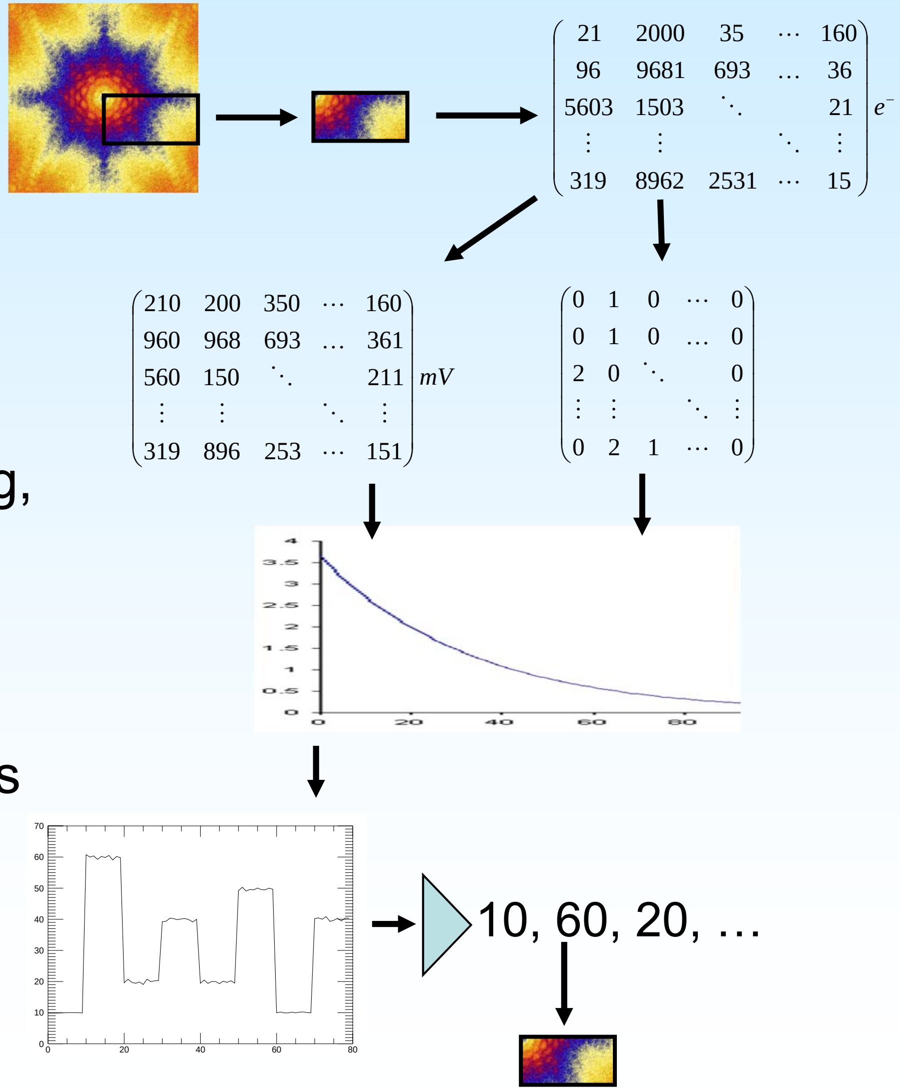
Input

Output

Difference

Simulation steps

- Input image processing, module definition
- Photon conversion -> generates input charge
- Amplification, gain switching, CDS simulation
- Treatment of storage cells
- ADC of voltage signals
- Requantization of ADC units
- Construction of the output image



$$\begin{pmatrix} 21 & 2000 & 35 & \cdots & 160 \\ 96 & 9681 & 693 & \dots & 36 \\ 5603 & 1503 & \ddots & & 21 \\ \vdots & \vdots & & \ddots & \vdots \\ 319 & 8962 & 2531 & \cdots & 15 \end{pmatrix} e^-$$

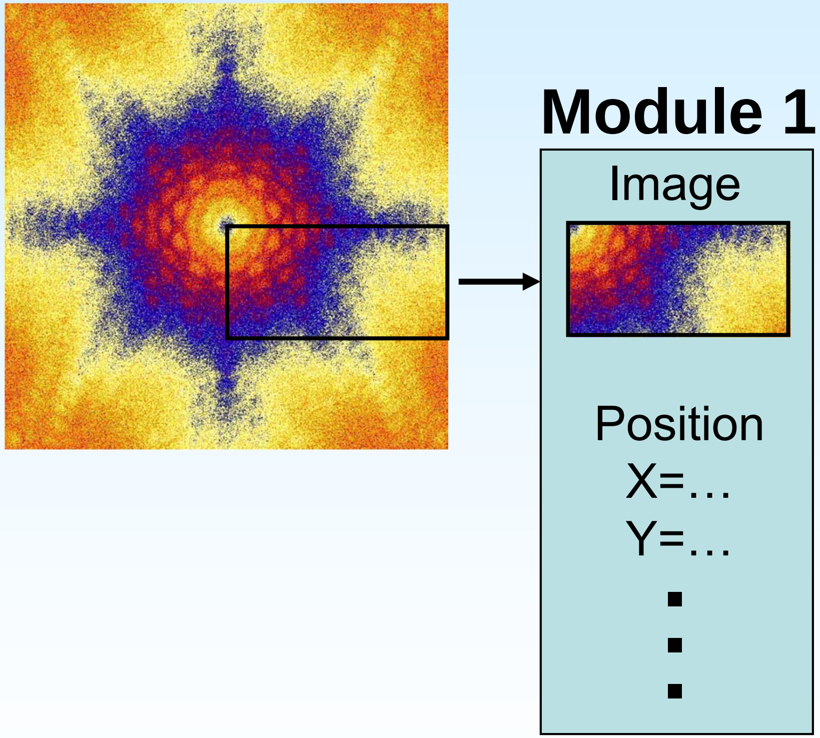
$$\begin{pmatrix} 210 & 200 & 350 & \cdots & 160 \\ 960 & 968 & 693 & \dots & 361 \\ 560 & 150 & \ddots & & 211 \\ \vdots & \vdots & & \ddots & \vdots \\ 319 & 896 & 253 & \cdots & 151 \end{pmatrix} mV$$

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 2 & 0 & \ddots & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 2 & 1 & \cdots & 0 \end{pmatrix}$$

10, 60, 20, …

Module definition

- Simulations are performed on a per module basis
- Input image is sliced into pieces
- Each module is an IDL struct carrying the image information of the current simulation step and additional information, like position, gains etc.
- Number of pixels/ASIC, ASICs/module and their arrangement is user definable



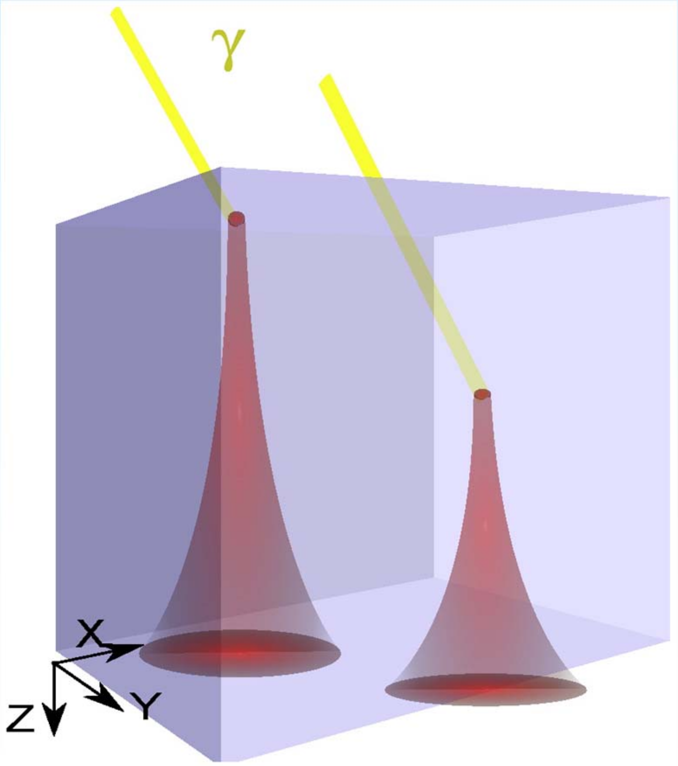
Module 1

Image

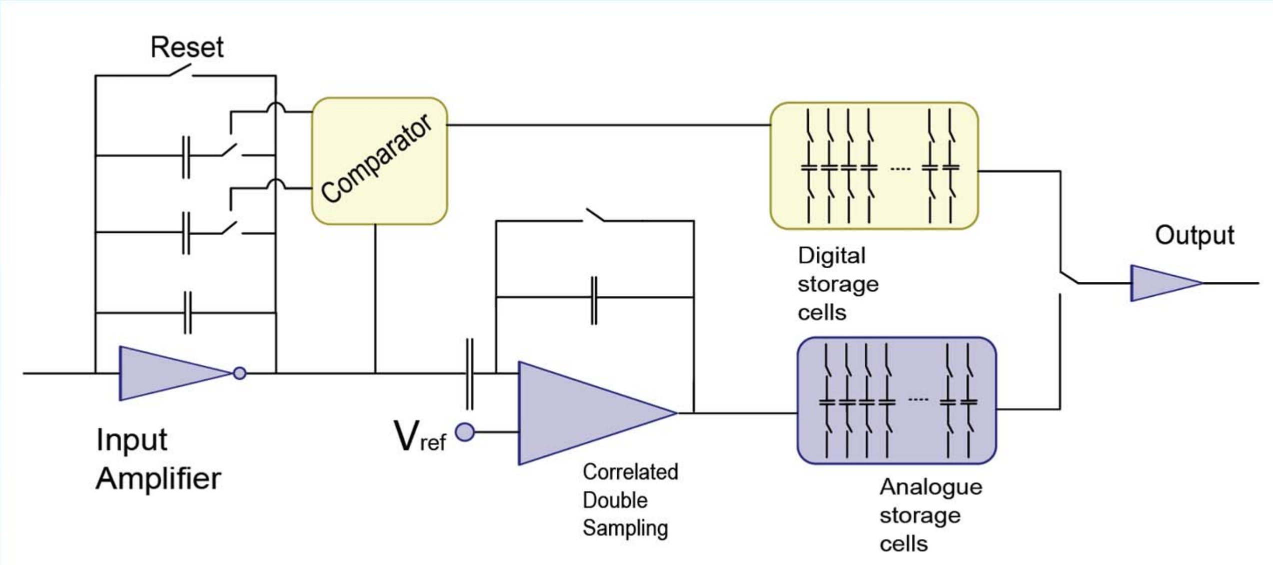
Position
X=…
Y=…
▪
▪
▪

Photon conversion

- Each photon is treated separately (MC approach)
- Absorption probability taken into account (Quantum efficiency, entry window as dead layer)
- Parallax effect is modeled
- Dispersion in actual e,h pairs created is modeled taking Fano factor into account
- Charge sharing is treated independently for each photon. Either as a depth dependent Gaussian or by a user-provided Cross-Coupling Matrix (allows to model CCE<1, non-uniform charge sharing etc.)



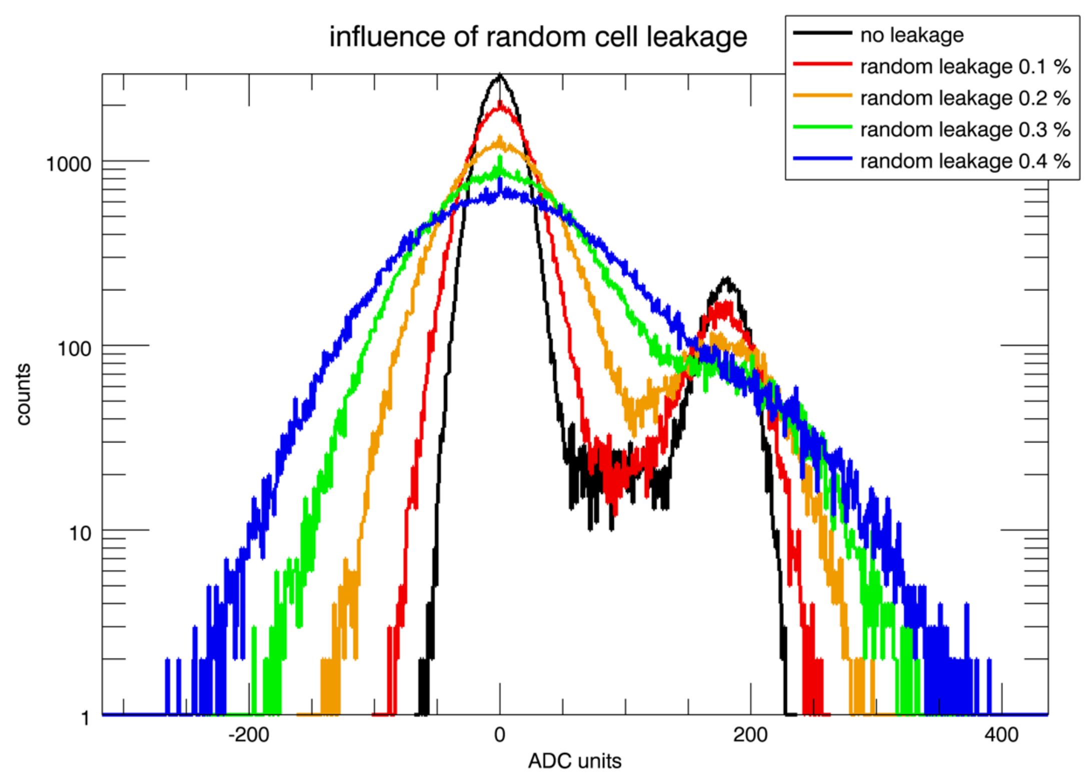
Amplification, gain switching, CDS simulation



- Noise sampled randomly according to ENC of current gain stage
- Charge injected by gain switching can be added (no data yet)
- Switching thresholds, gains and noise can be set by the user
- Fixed gain operation can be simulated by setting thresholds correspondingly
- Saturation behavior is unknown, implemented simple clipping to maximum allowed value, but code is prepared for different models

Treatment of storage cells

- Fixed leakage (can be corrected for): each cell the same each time it is read
- Random leakage (can not be corrected for): each cell different each time it is read
- Leakage parameters can be set by the user
- Code ready to handle more elaborate models

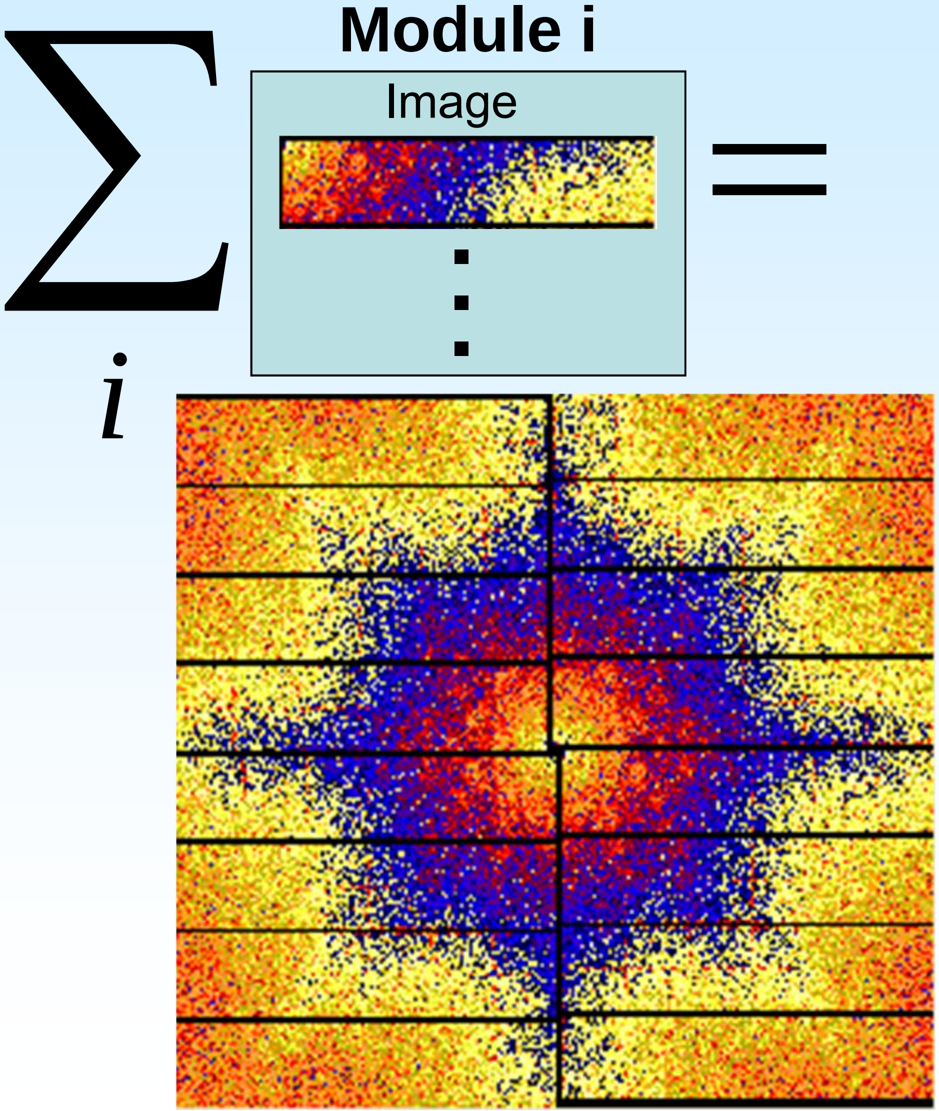


ADC of voltage signals and requantization

- ADC simulation
  - Each pixel is treated separately (MC approach)
  - Range of ADC taken into account (14 bit)
  - Noise of ADC taken into account (4.6 LSB)
  - Noise can be modified by the user
- Requantization
  - Gain stage taken into account
  - Values below 0 (due to noise) are clipped to 0

Construction of the output image

- Image data of each module is assembled into one large image
- Certain options allow to return different images (e.g. ADUs, Gains, input electrons, etc.)



Special features

- Can be used in an iterative way to calculate images for polychromatic sources (although inefficient)
- Treats parallax for any distance between detector and sample (assuming point like scattering source)
- Requantizied image, ADUs and gains are returned simultaneously

"It's Not a Bug, It's a Feature!"

Computer Wit and Wisdom

David Lubar

Limitations

- No treatment of plasma effects (happen with >10^3 γ in a pixel) so far (although some ideas are present)
- No non-centered photon sources (although there is a workaround for this)
- Fluorescence of Si not accounted for (code exists from Medipix simulations by David)
- So far limited to silicon as sensor material
- No backscatter/fluorescence from parts behind sensor (Bumps/ASIC) (but result of a MC-simulation can be fed into HORUS)

Summary: HORUS

- Working horse of detector simulations
- Many improvements: speed, less bugs, features, etc.
- Point and click interface to investigate behavior
- Less hard coded constrains
- No whole scale MC code (e.g. no fluorescence, Comton-scattering)
- Some open issues (eg. Plasma effect)