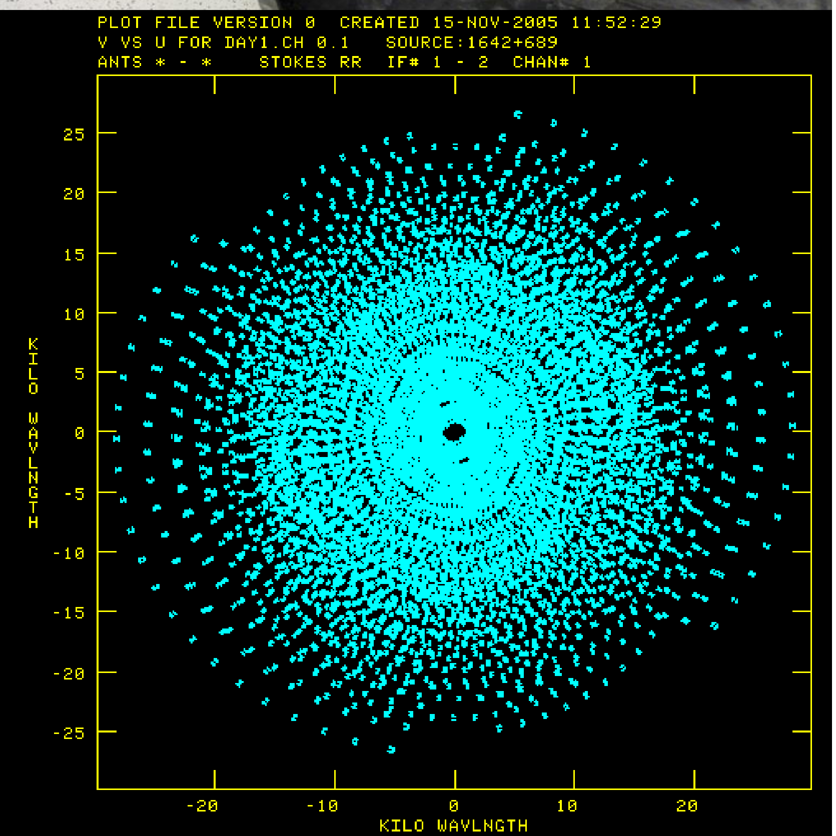


Cross correlators

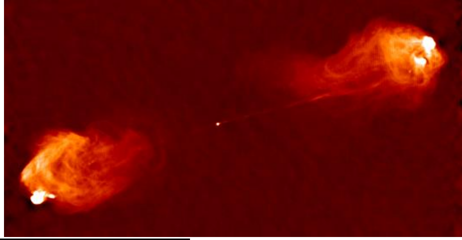
for radio astronomy

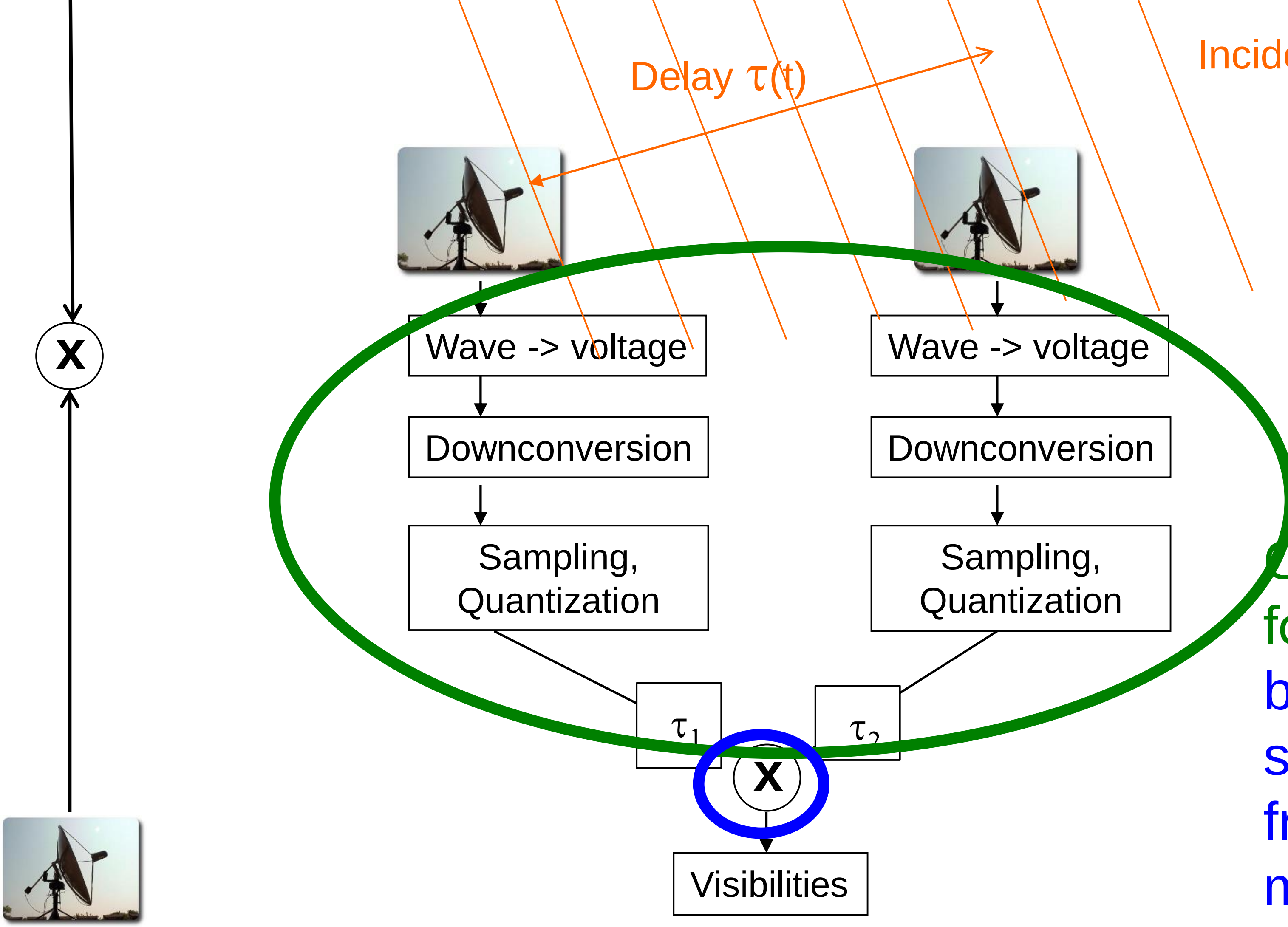
Adam Deller

June 1, 2016

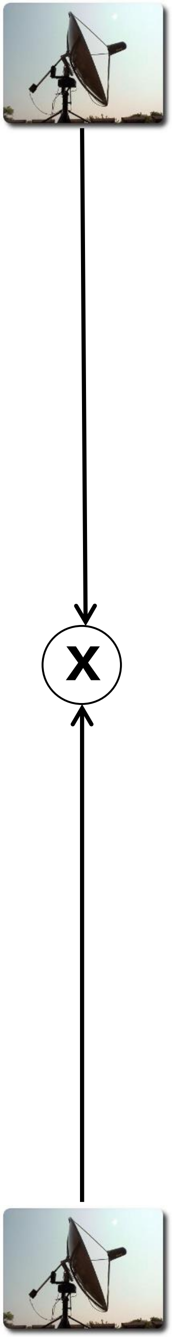
ASTRON

What is a correlator?



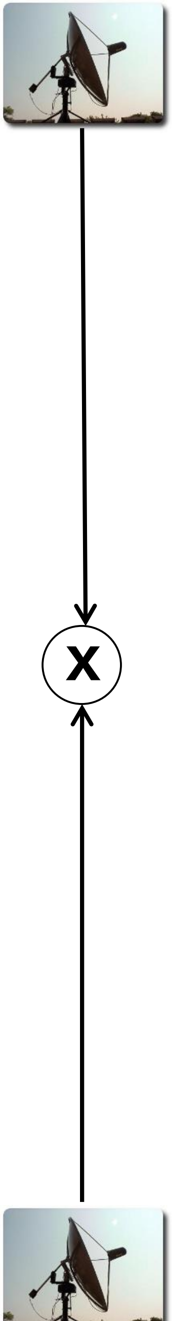


Compensate for this…
before/while splitting by frequency and multiplying

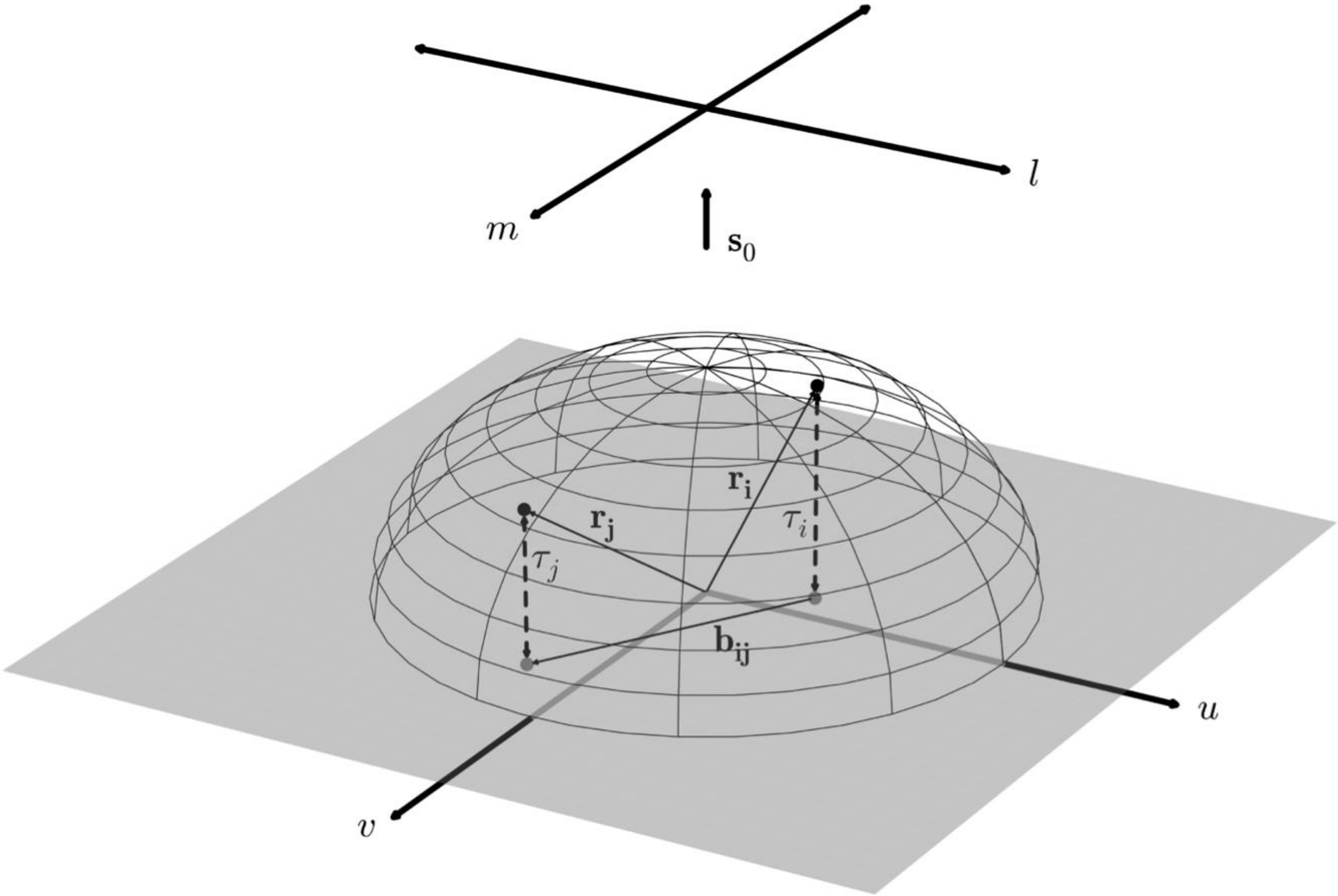


Why correlators matter to YOU

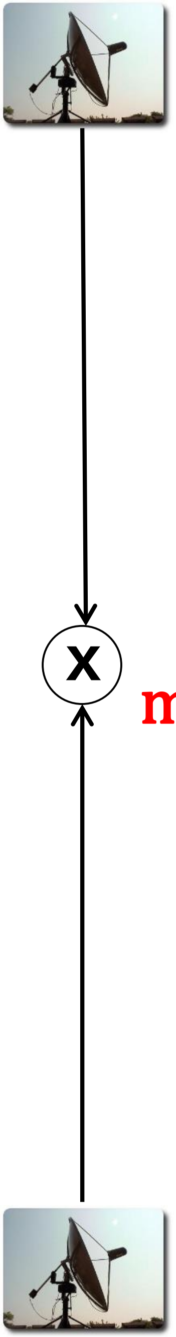
1. You want to design radio astronomy instrumentation / algorithms.
2. To help you propose the right observations and identify problems in data or images.

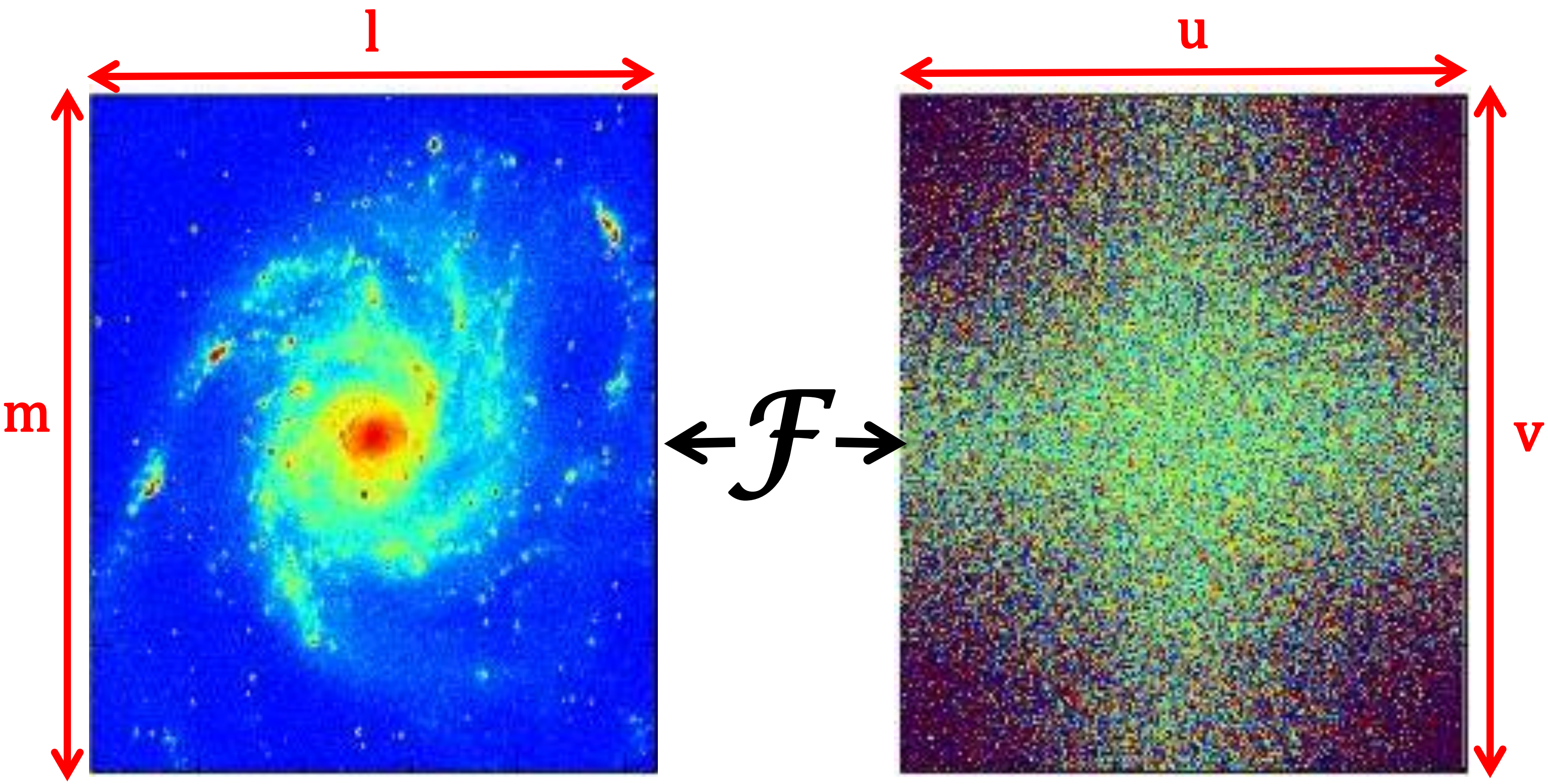


Correlators and Interferometry



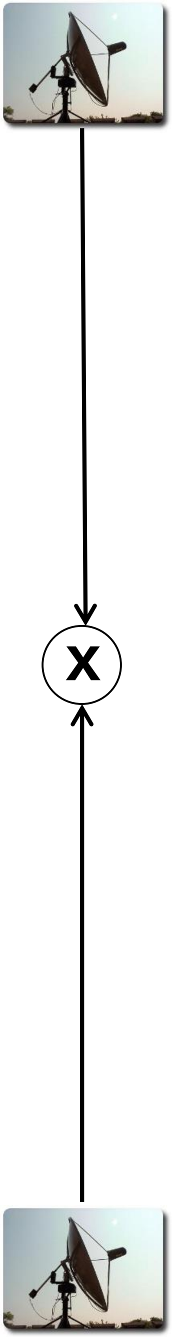
Correlators and Interferometry





Sky brightness at frequency ν_0

Visibilities (real component shown, unit is $\lambda_0 = c / \nu_0$)

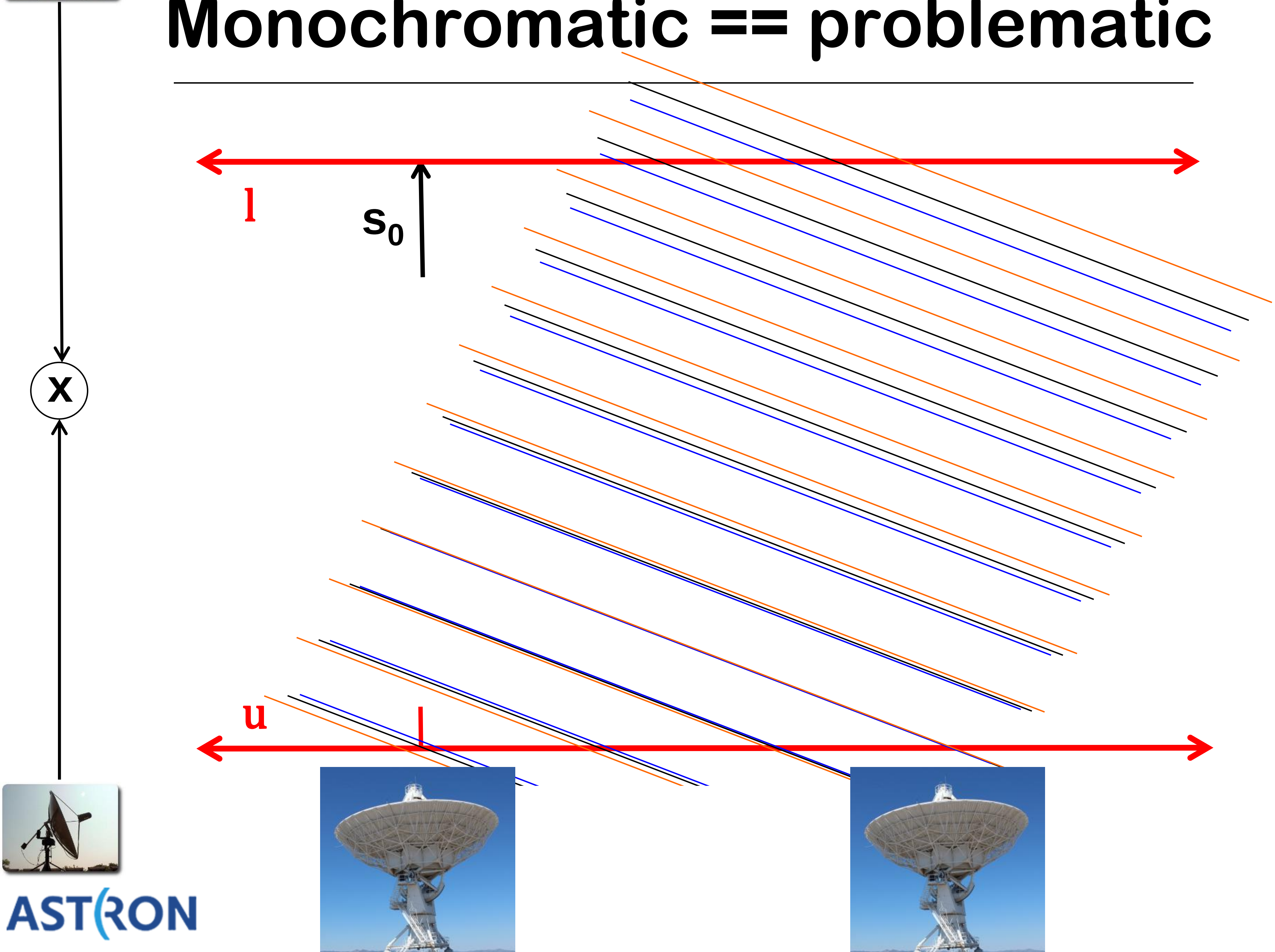


Monochromatic == problematic

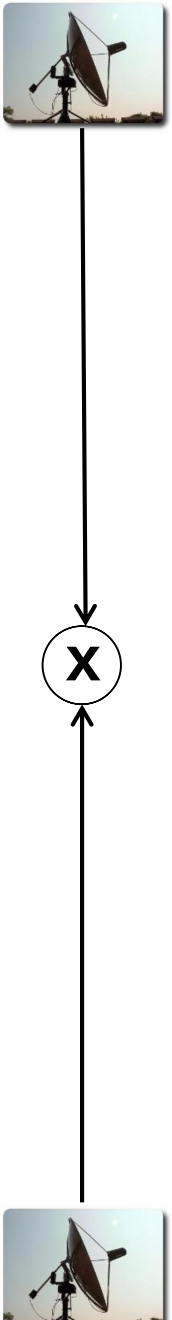
- No truly monochromatic radiation
- Fortunately, "fairly narrow" band of $\Delta\nu$ (*quasi-monochromatic*) can suffice:
  - $u \times l + v \times m$ is supposed be constant, and our problem is that both u and v are $\propto \nu$
  - if $\Delta u \times l \ll 1$ and $\Delta v \times m \ll 1$ then we're ok
  - Correlator needs to slice at least this fine
  - Real world viewpoint: different frequency components stay "in phase" as wavefront propagates from one antenna to the next

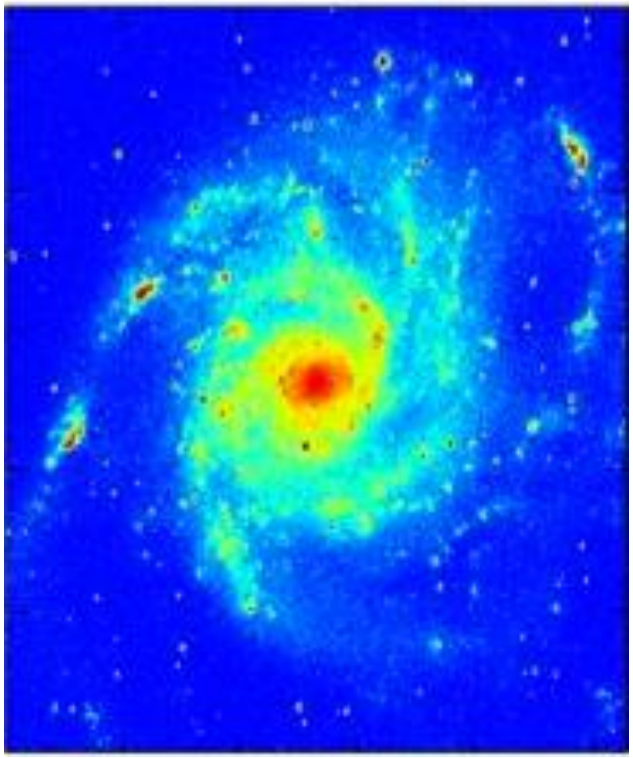
ASTRON

Monochromatic == problematic

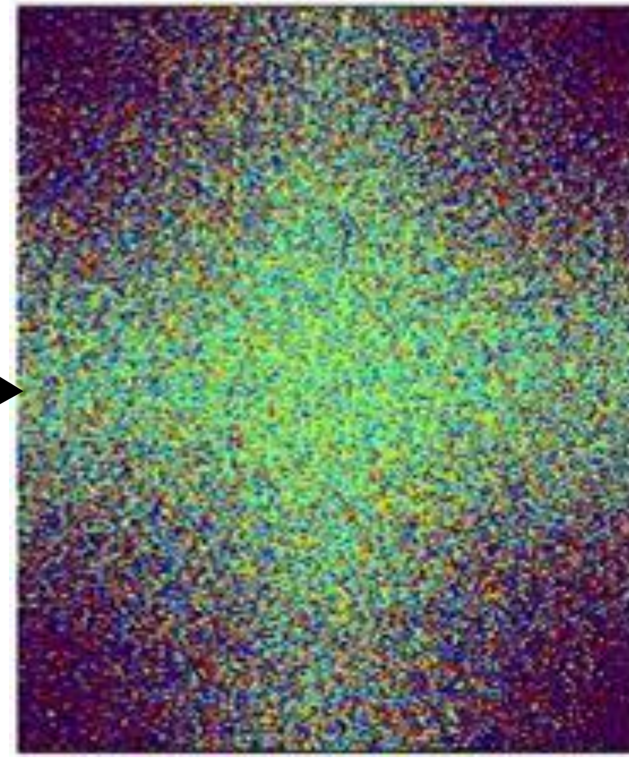


Correlators and Interferometry





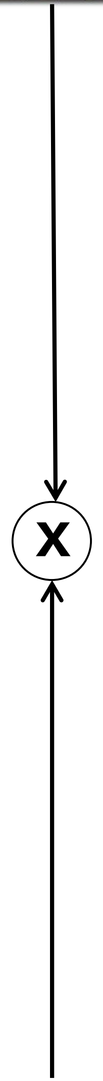
$\leftarrow \mathcal{F} \rightarrow$

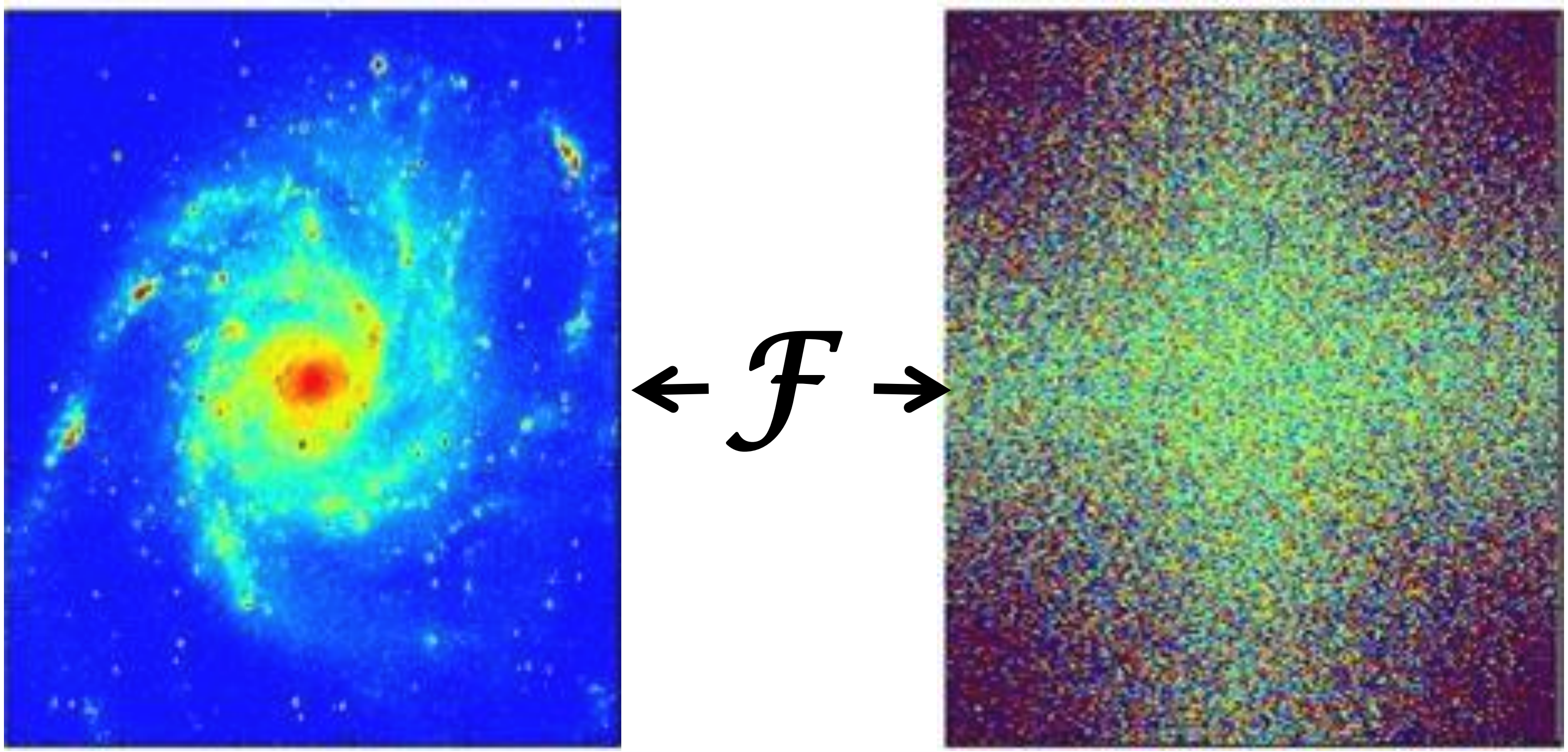


Sky brightness at frequency ν_0

Visibilities (real component shown, unit is $\lambda_0 = c / \nu_0$)

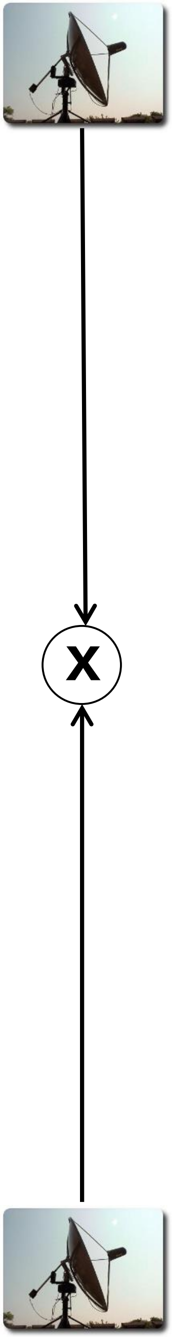
Correlators and Interferometry





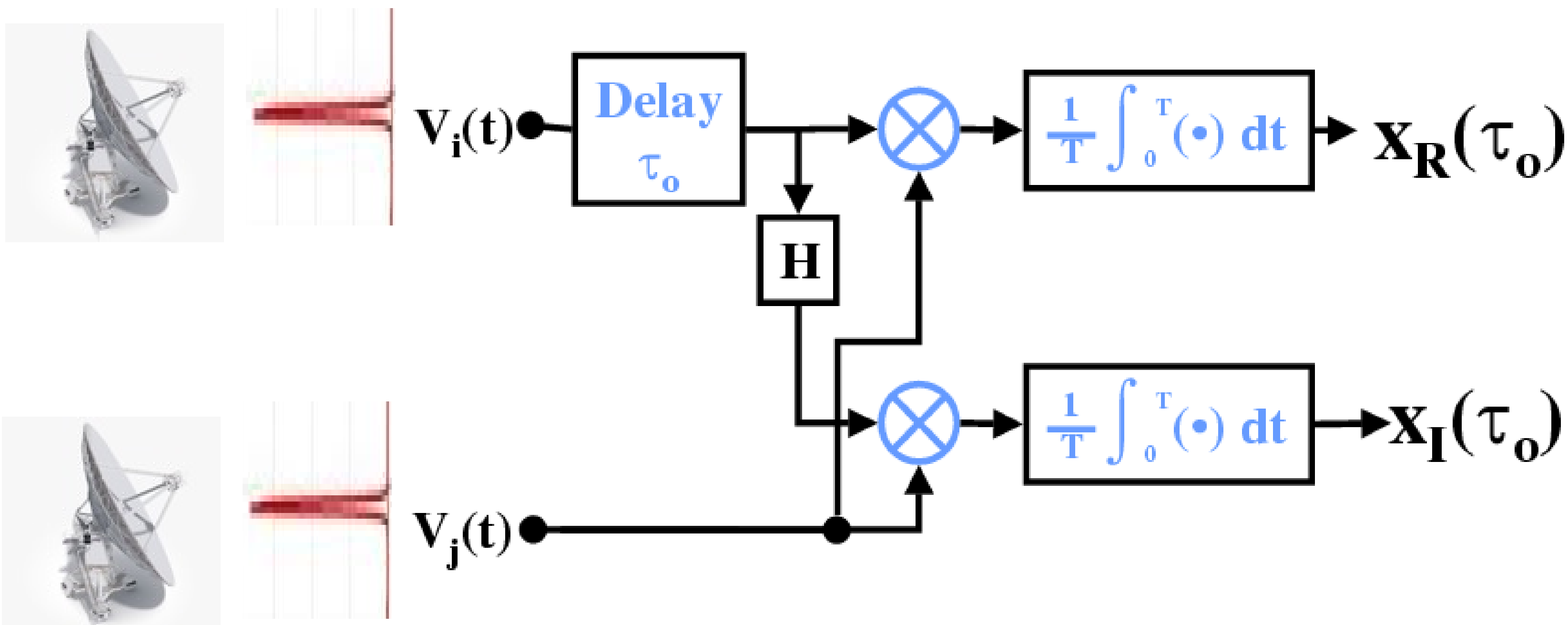
Sky brightness at frequency ν□ $= \nu_0 + \delta\nu$

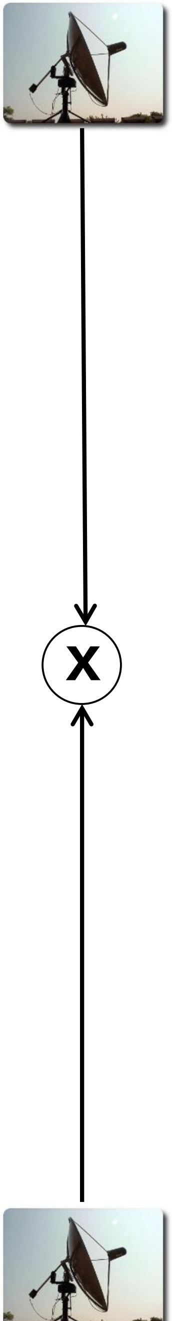
Visibilities (real component shown, unit is λ' = *c* / ν□)



A “dumb” correlator

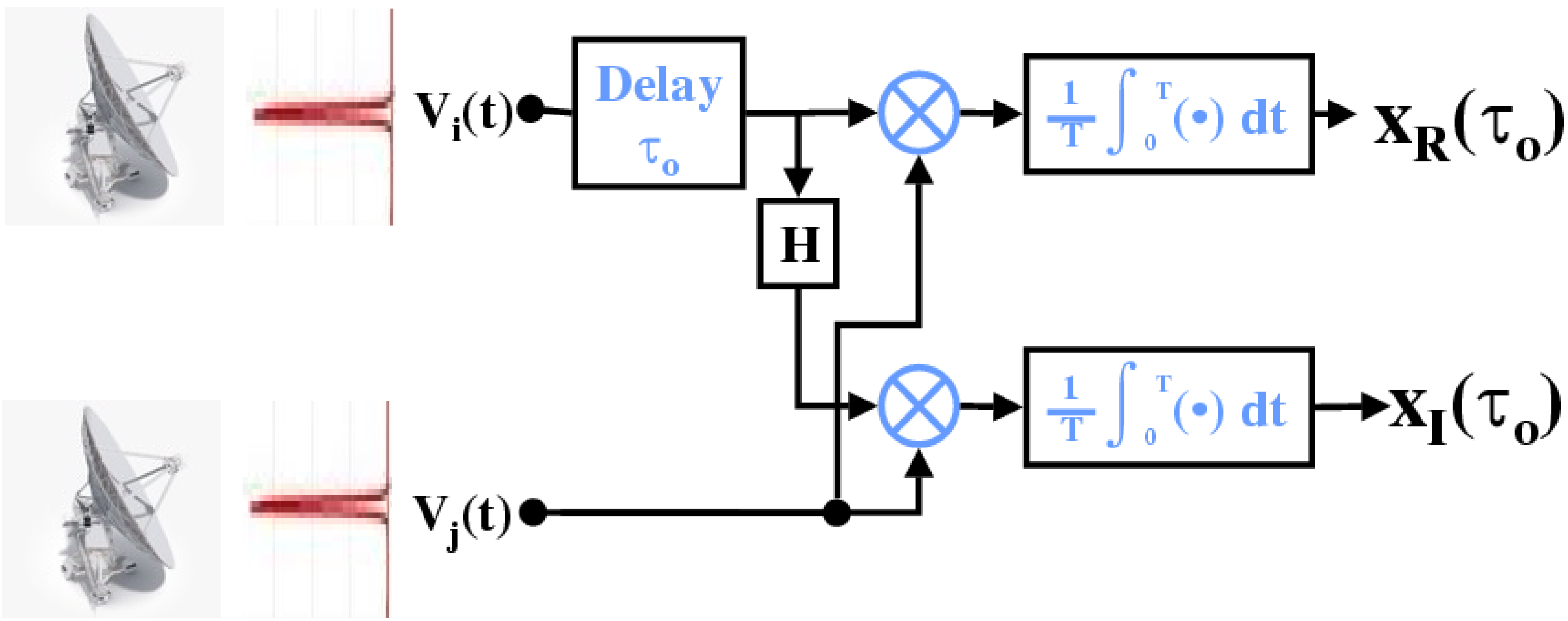
- Use many analog filters to make many narrow channels; correlate each one separately with a standard complex correlator:



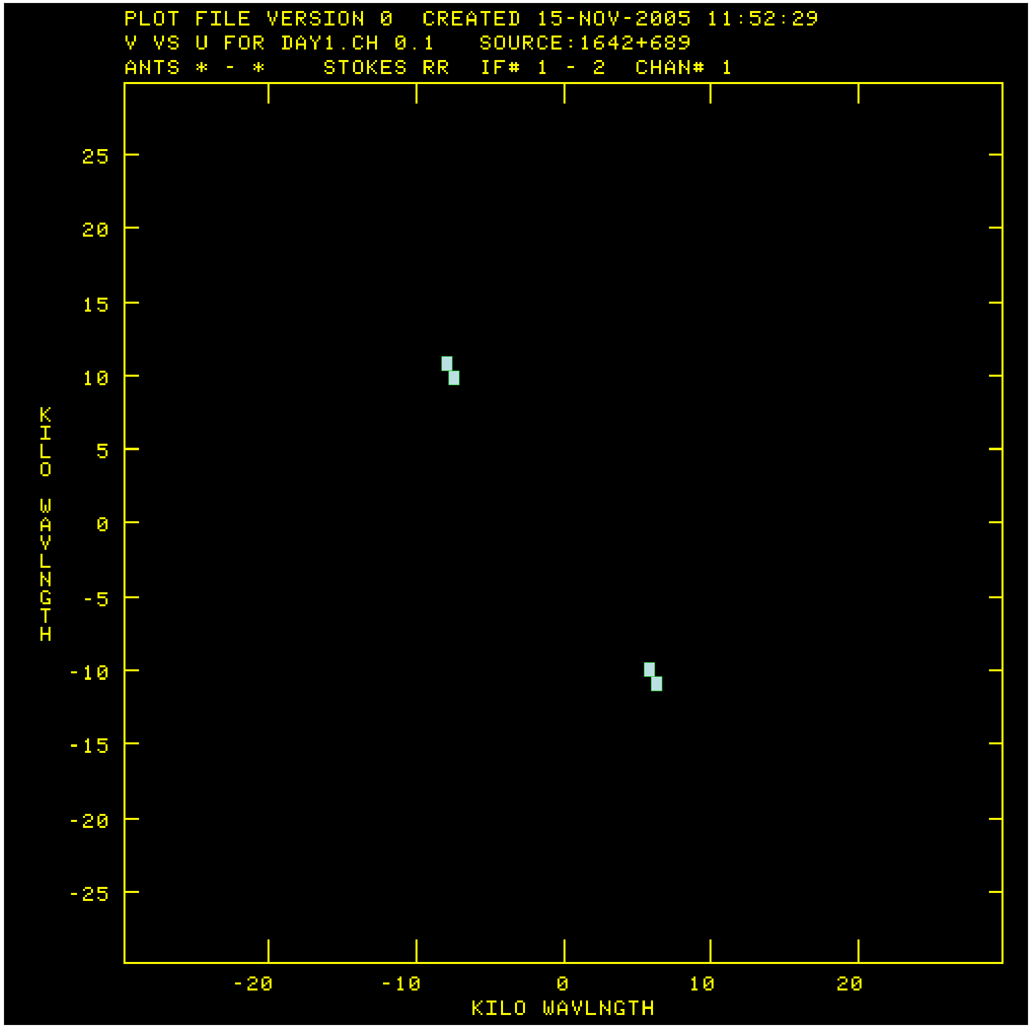


A "dumb" correlator

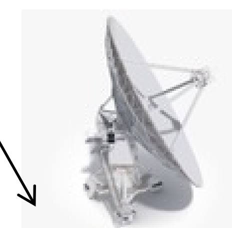
- Use many analog filters to make many narrow channels; correlate each one separately with a standard complex correlator:



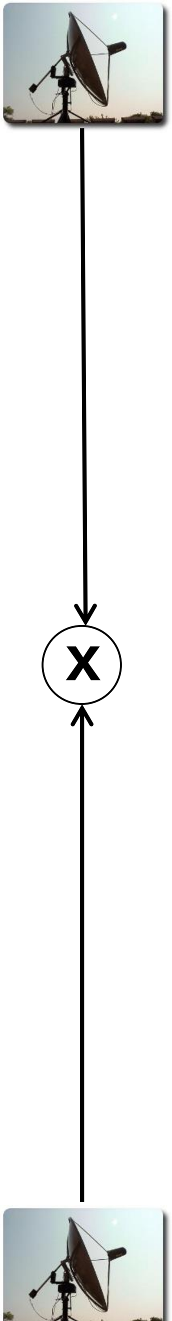
The output

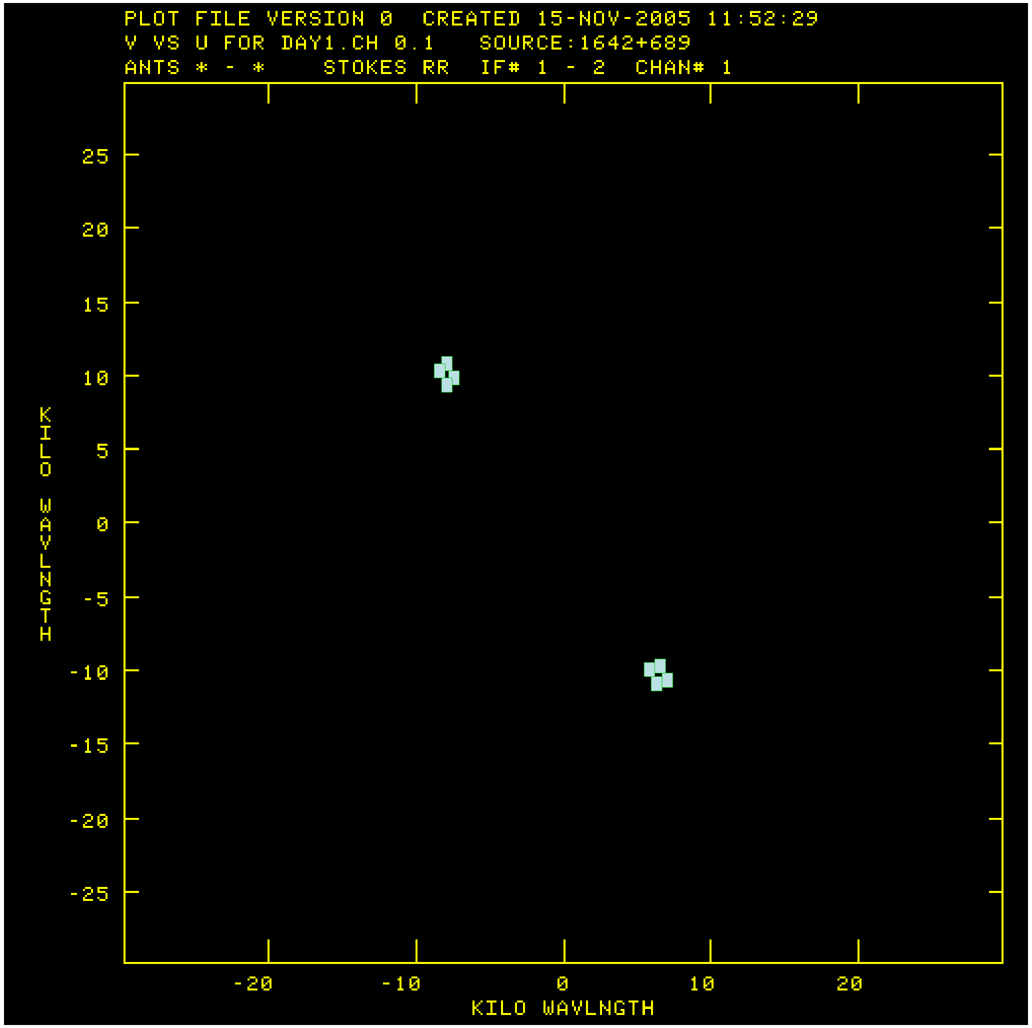


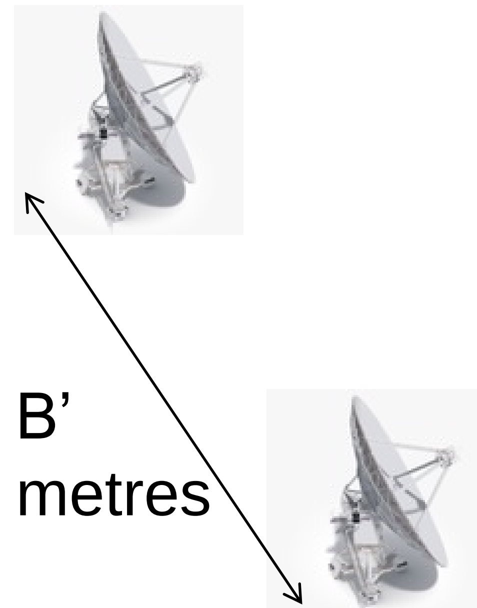
B metres

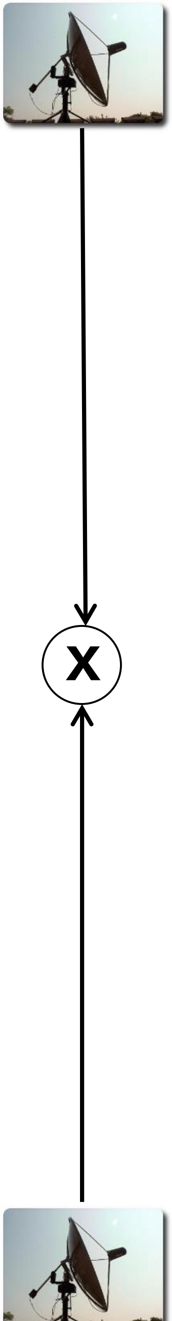


The output


X

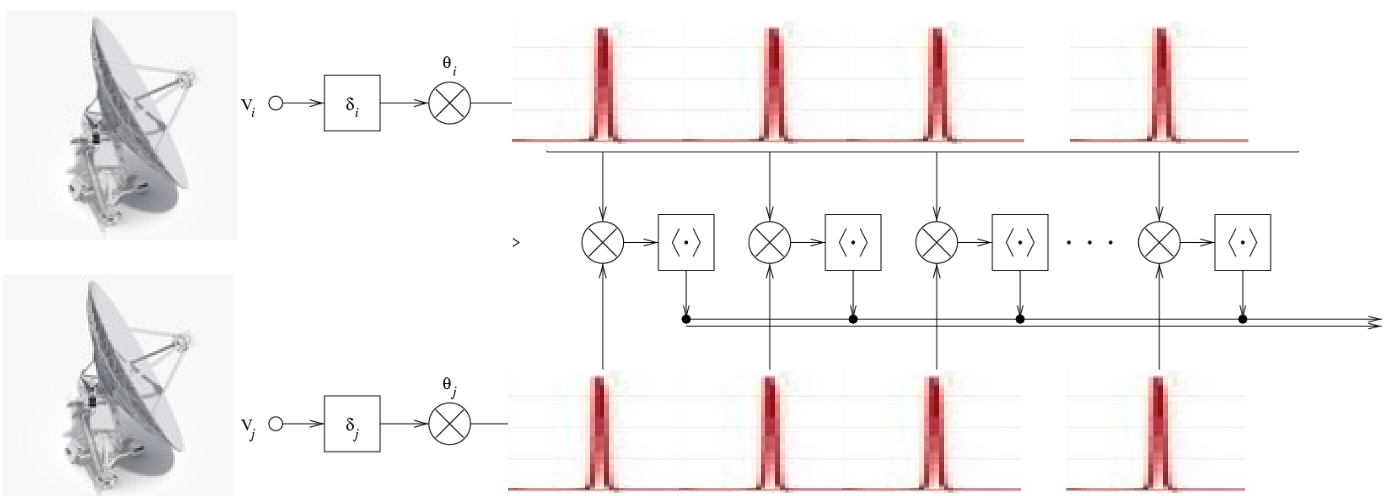

PLOT FILE VERSION 0 CREATED 15-NOV-2005 11:52:29
V VS U FOR DAY1.CH 0.1   SOURCE:1642+689
ANTS * - *   STOKES RR IF# 1 - 2 CHAN# 1
KILO WAVLNGTH
KILO WAVLNGTH
25
20
15
10
5
0
-5
-10
-15
-20
-25
-20
-10
0
10
20

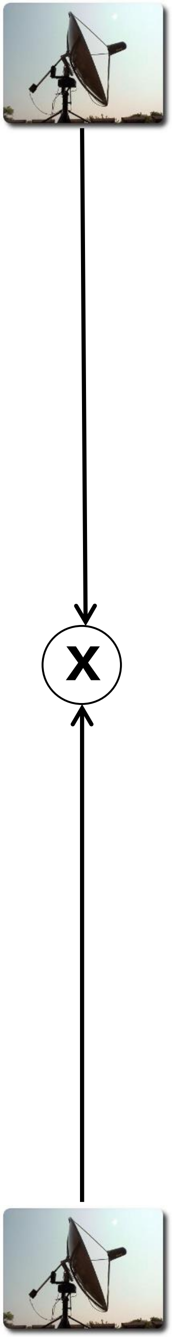

B’
metres



Making it feasible

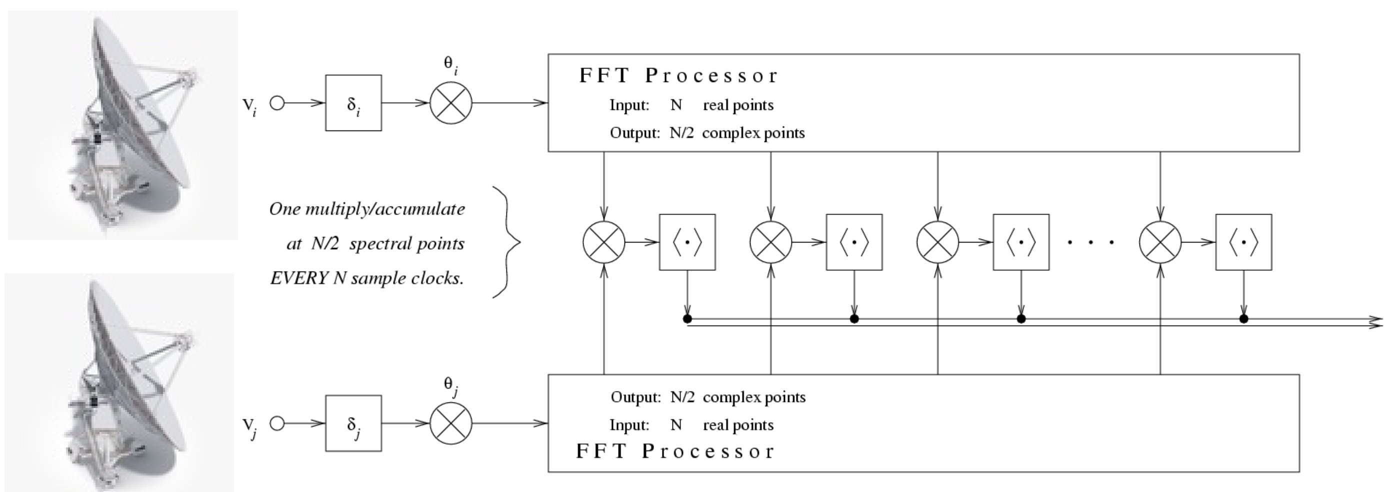
- Analog filters are costly & unstable; expensive and poor performance

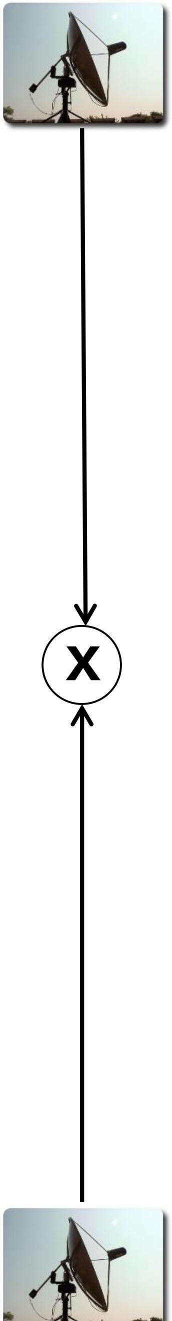




Making it feasible

- Analog filters are costly & unstable; expensive and poor performance
- Fortunately, we can (and do) digitize the signal – meaning we can use a digital substitute: **digital filterbank**

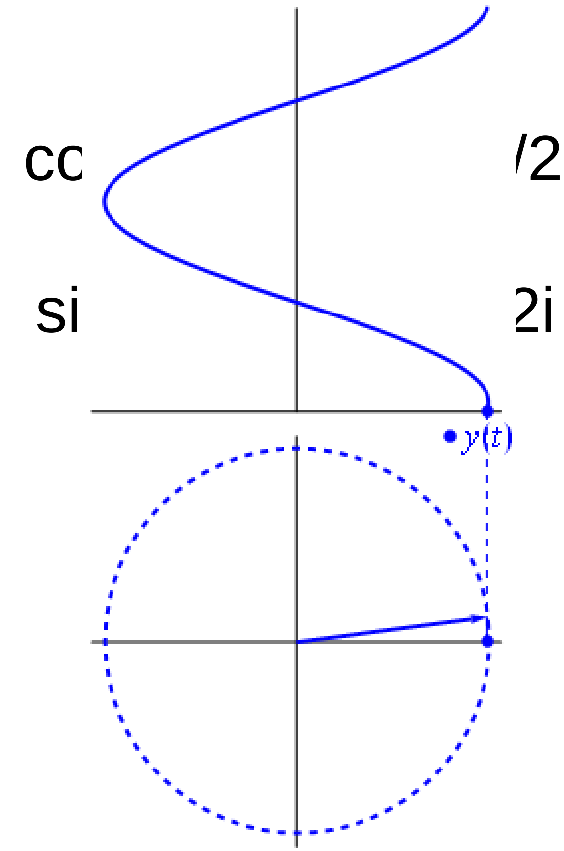




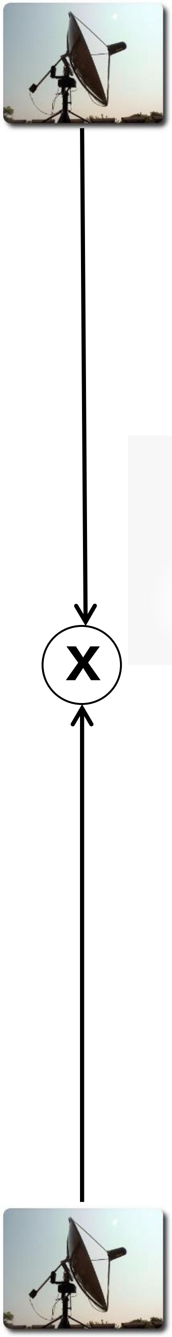
The advantage of going digital

- Stable, cheap filters
- Produces complex output: use a 1 complex multiplier rather than 2 real multipliers and a phase shift

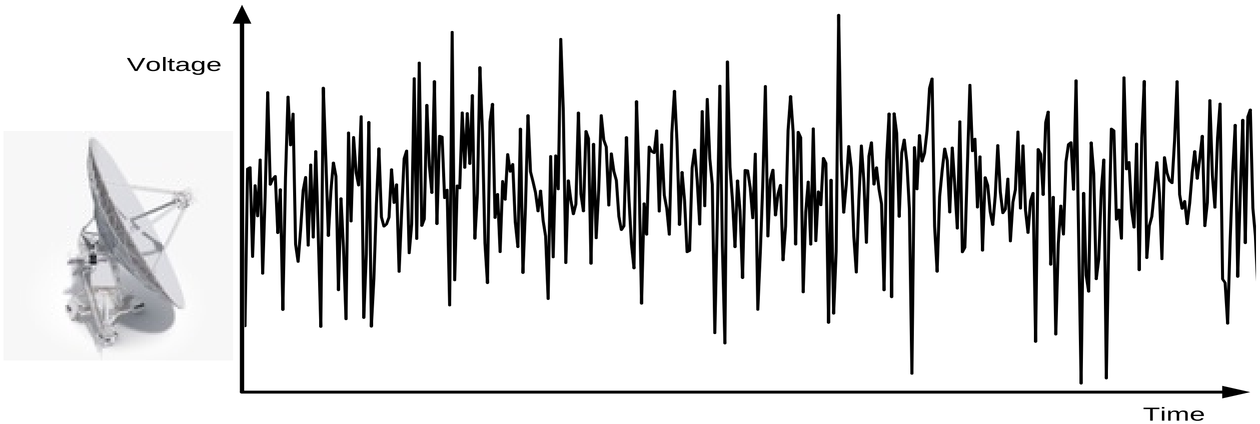
$e^{i\phi} = \cos\phi + i \sin\phi$



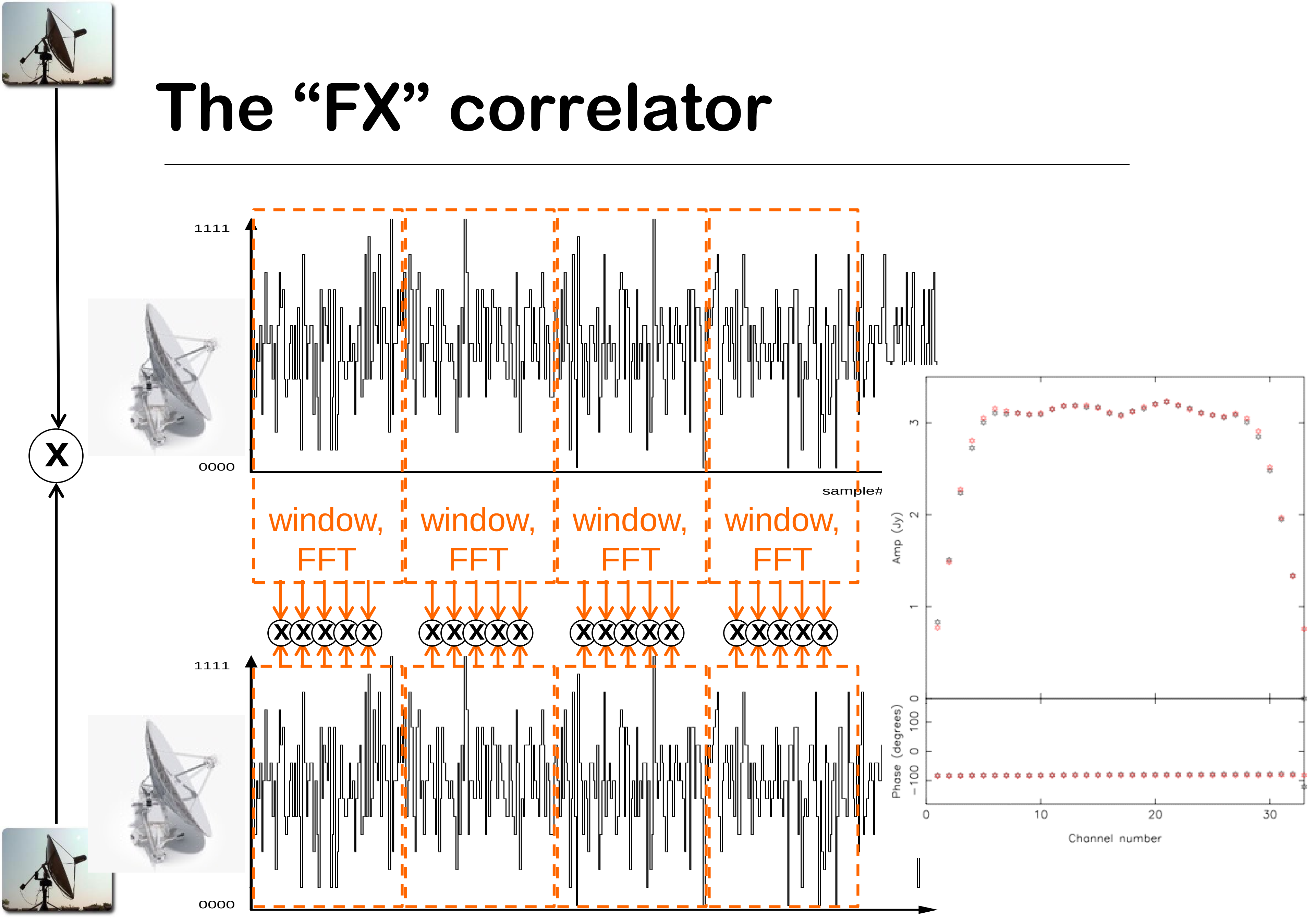
Animation from http://en.wikipedia.org/wiki/File:Unfasor.gif



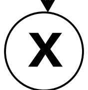
The "FX" correlator

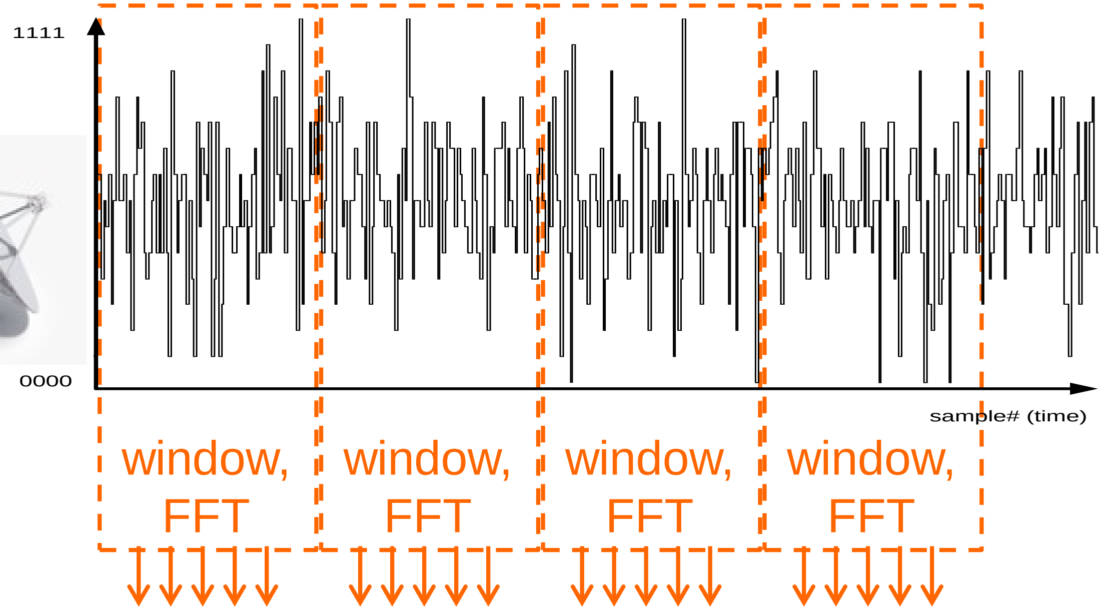


The "FX" correlator



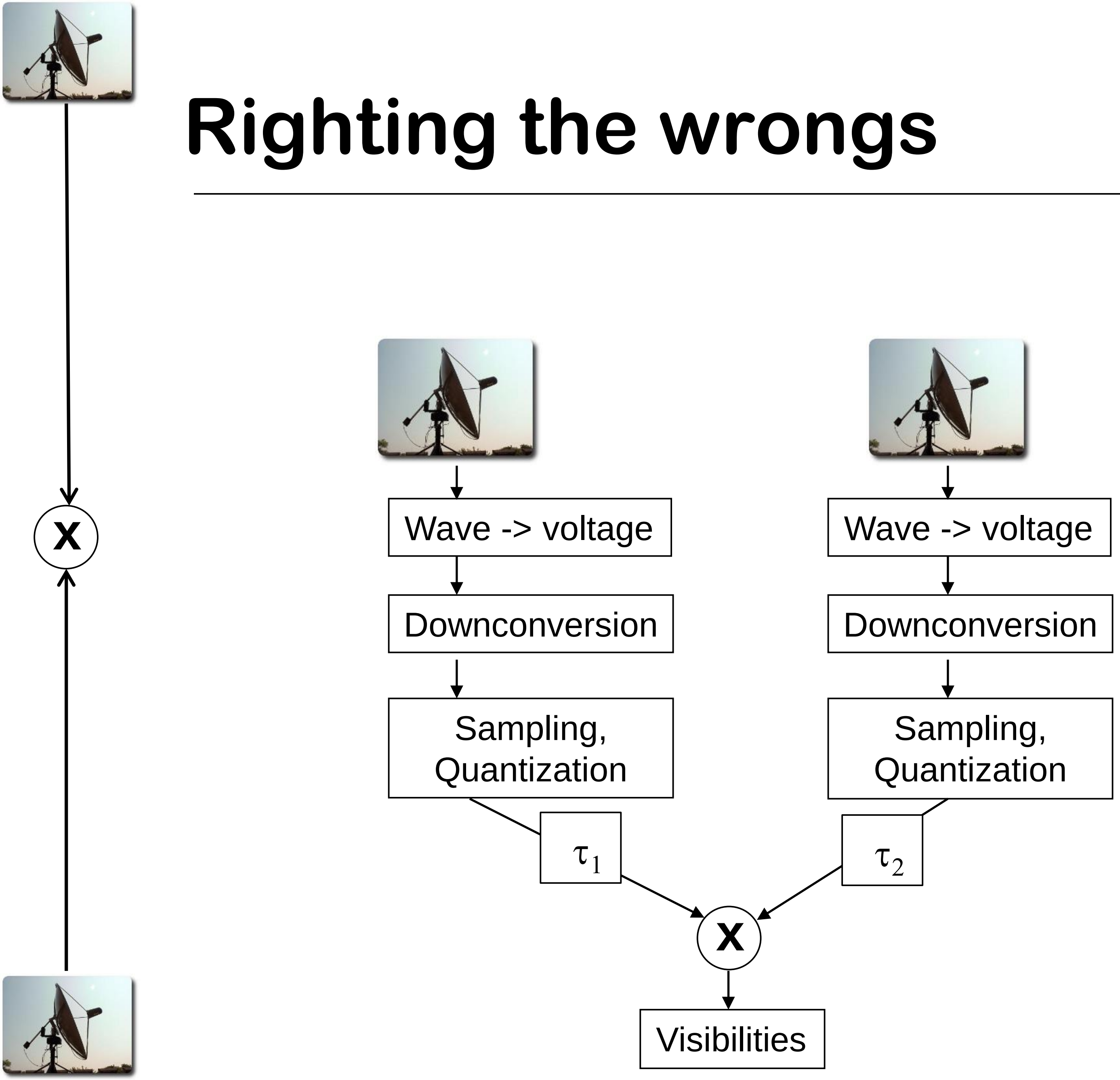
The "FX" correlator



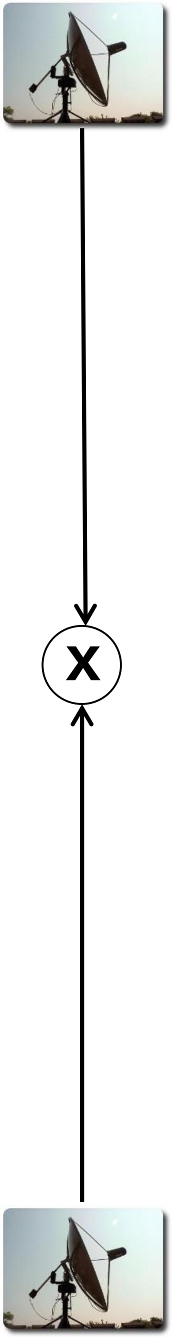


- Since this architecture consists of a Fourier transform (F) followed by cross-multiplication (X), we dub this the "FX" correlator

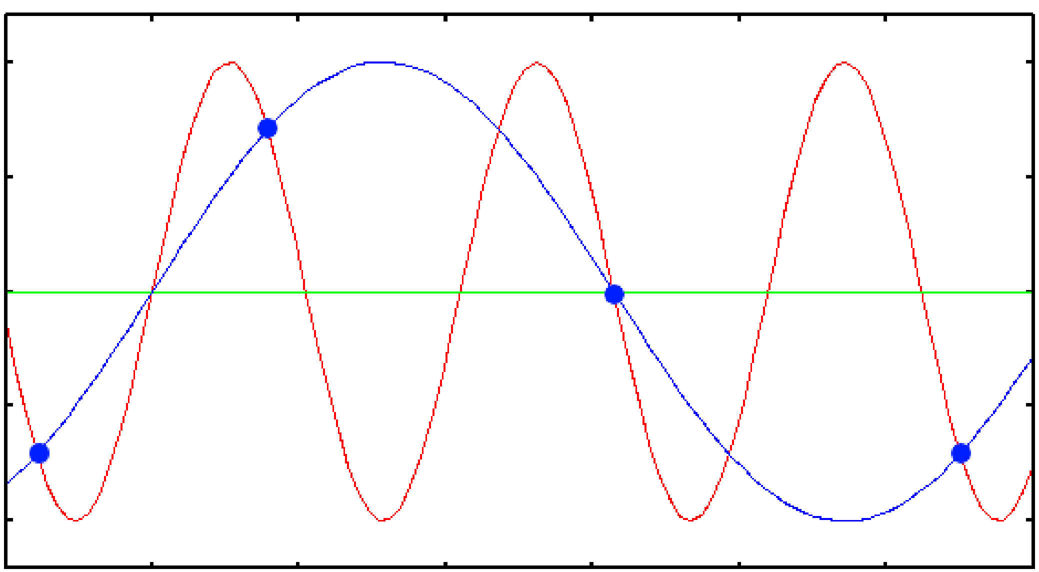
Righting the wrongs

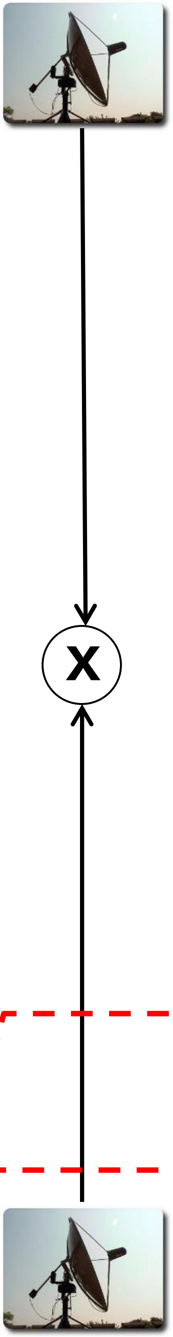


Sampling



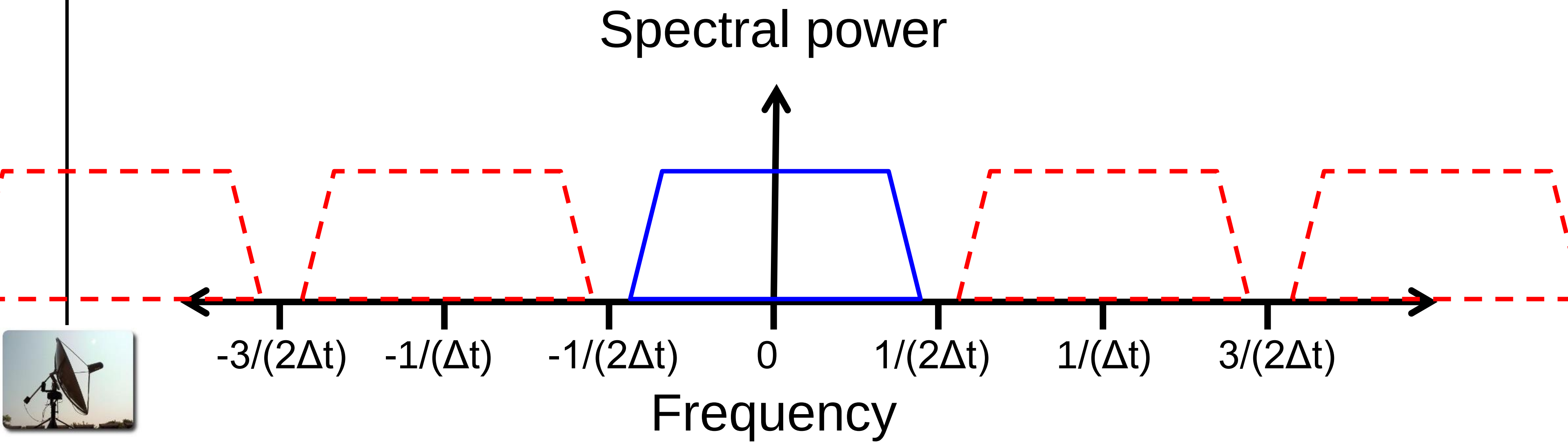
- Nyquist-Shannon sampling theorem:
  - real-valued signal is sampled every Δt sec
  - Original signal can be reconstructed perfectly so long as contains no power at frequencies ≥ 1 / (2 Δt) Hz (*band-limited*)





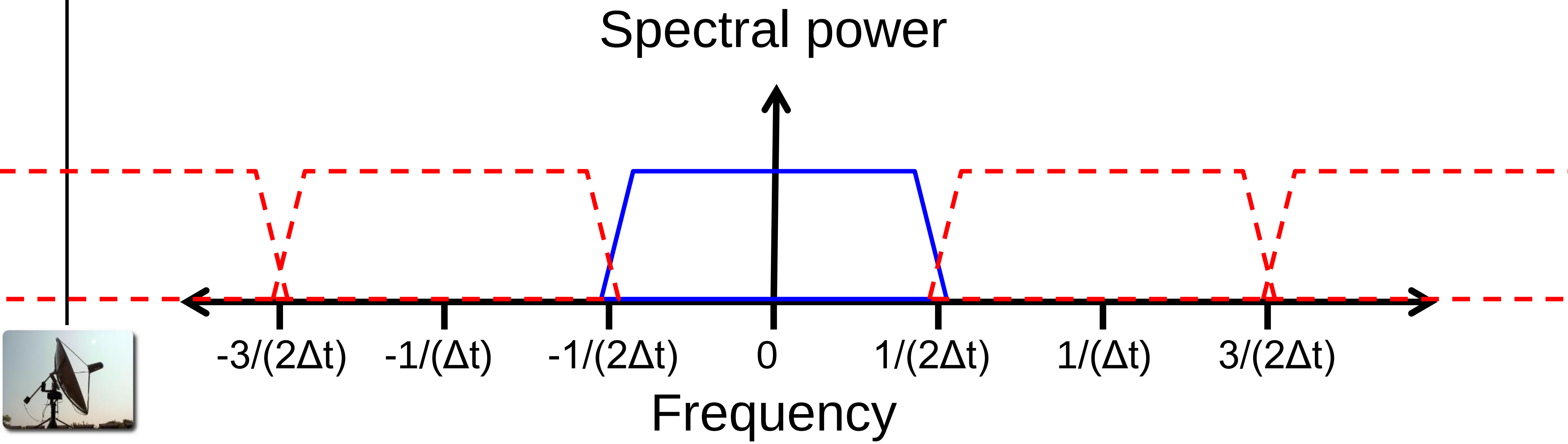
Sampling

- Nyquist-Shannon sampling theorem:
  - real-valued signal is sampled every Δt sec
  - Original signal can be reconstructed perfectly so long as contains no power at frequencies ≥ 1 / (2 Δt) Hz (*band-limited*)



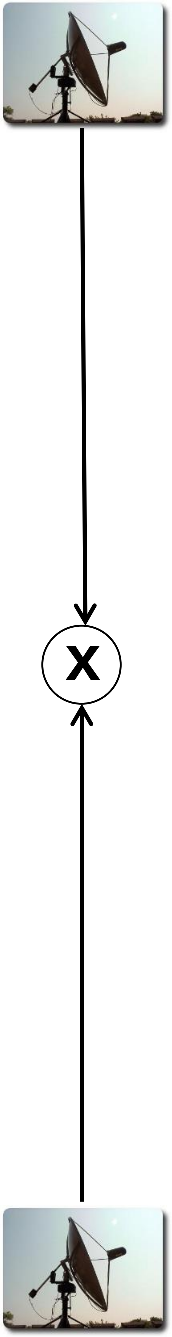
Sampling

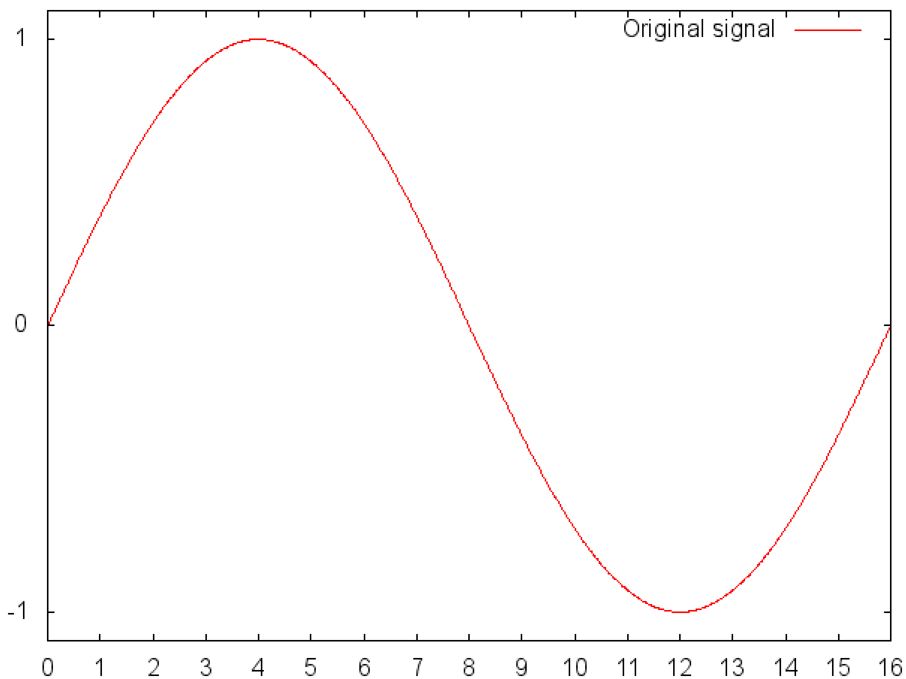
- Nyquist-Shannon sampling theorem:
  - real-valued signal is sampled every Δt sec
  - Original signal can be reconstructed perfectly so long as contains no power at frequencies ≥ 1 / (2 Δt) Hz (*band-limited*)



Quantization

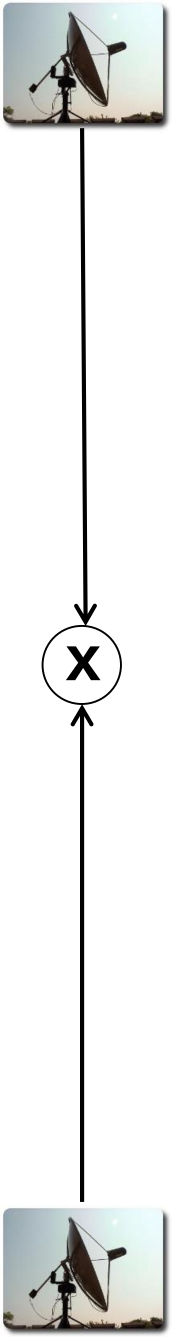
- When correlation is low (almost always) even very coarse quantization is ok!



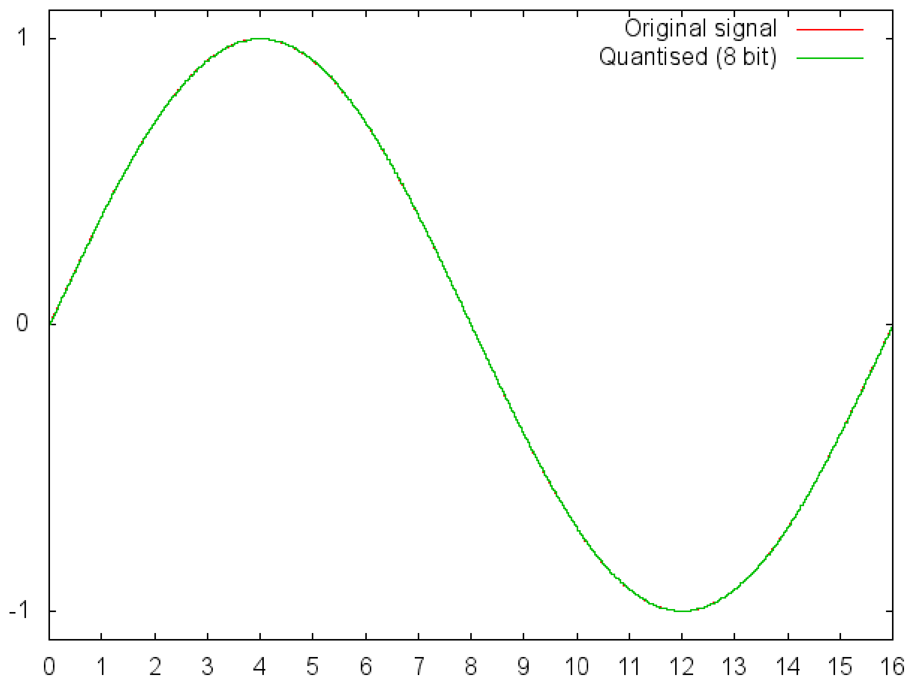


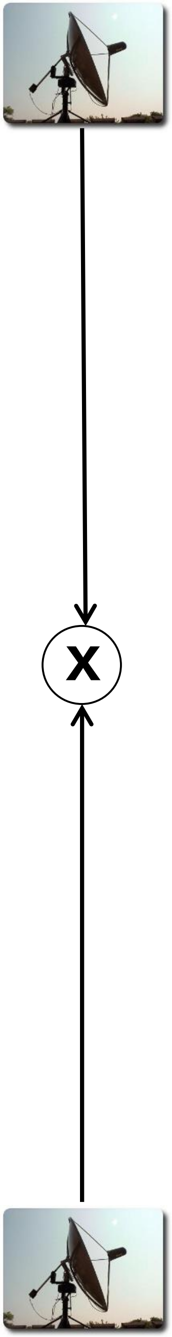
Sensitivity loss:

Quantization



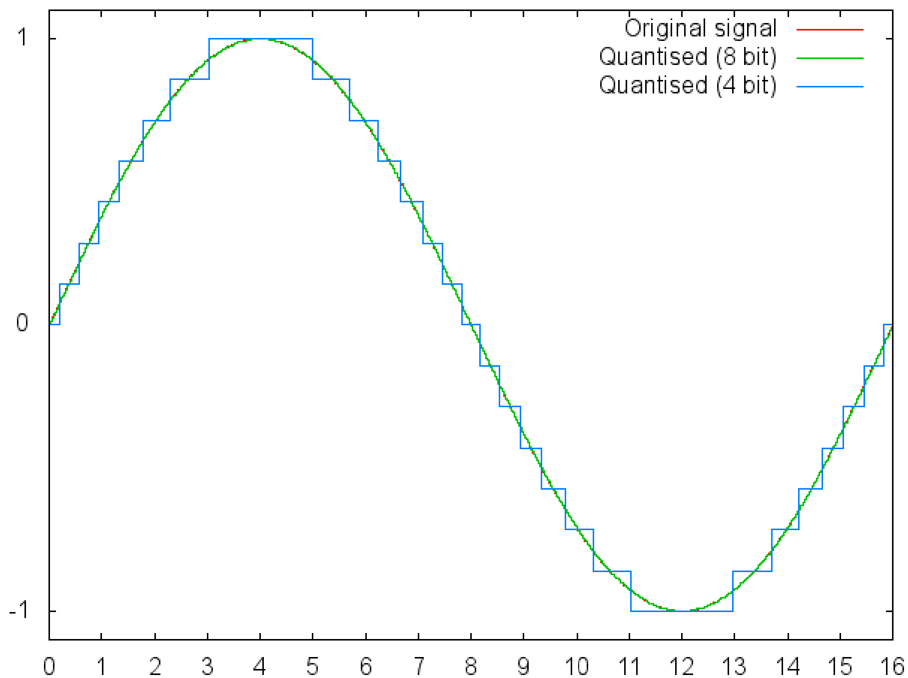
- When correlation is low (almost always) even very coarse quantization is ok!



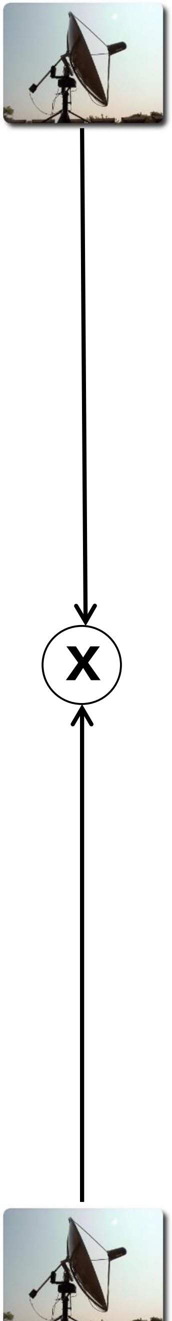


Quantization

- When correlation is low (almost always) even very coarse quantization is ok!

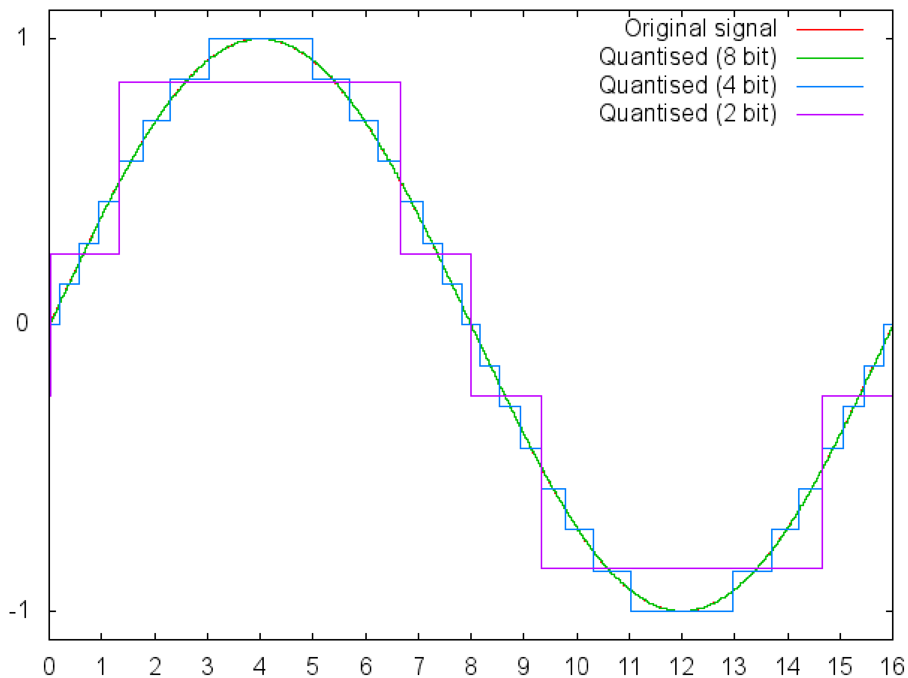


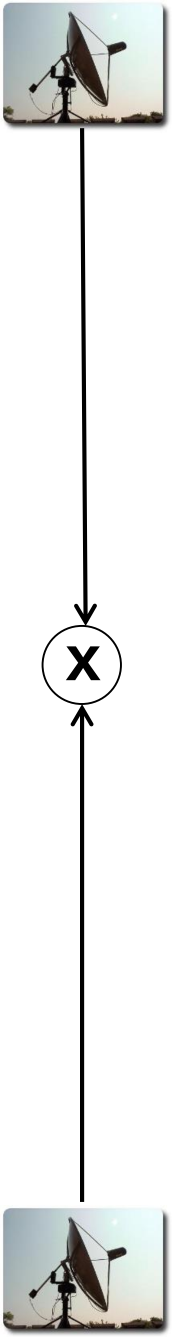
ASTRON



Quantization

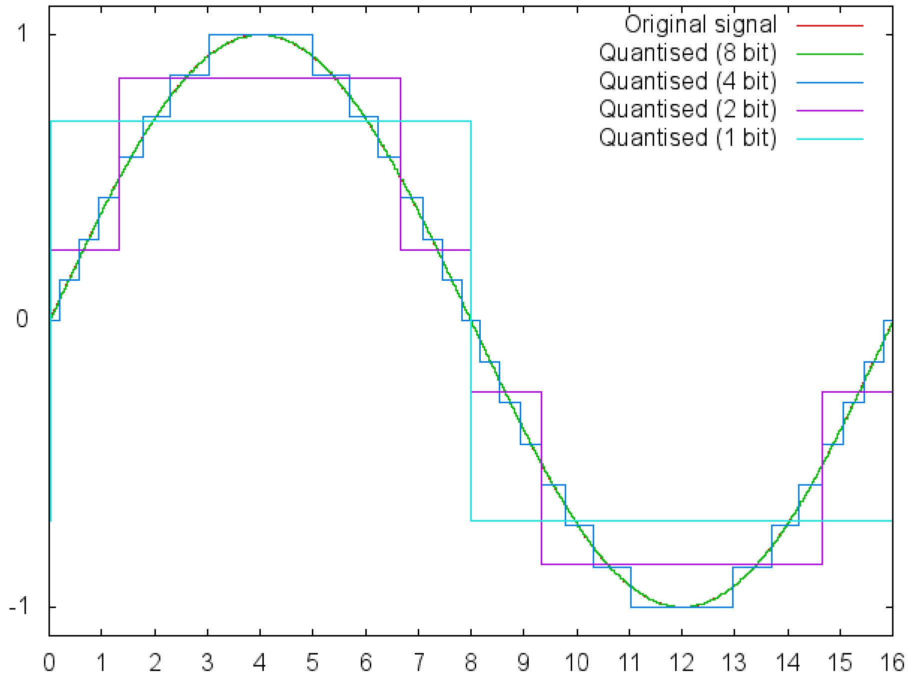
- When correlation is low (almost always) even very coarse quantization is ok!





Quantization

- When correlation is low (almost always) even very coarse quantization is ok!

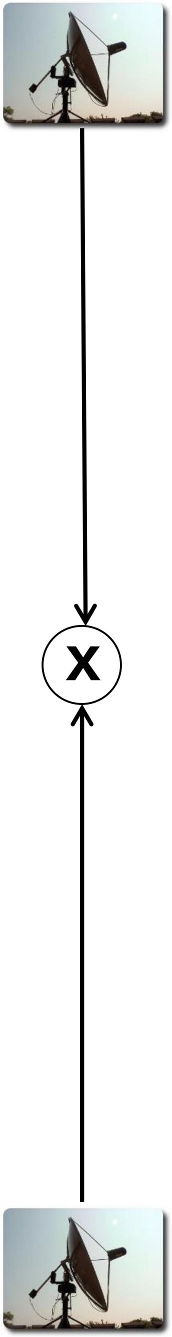


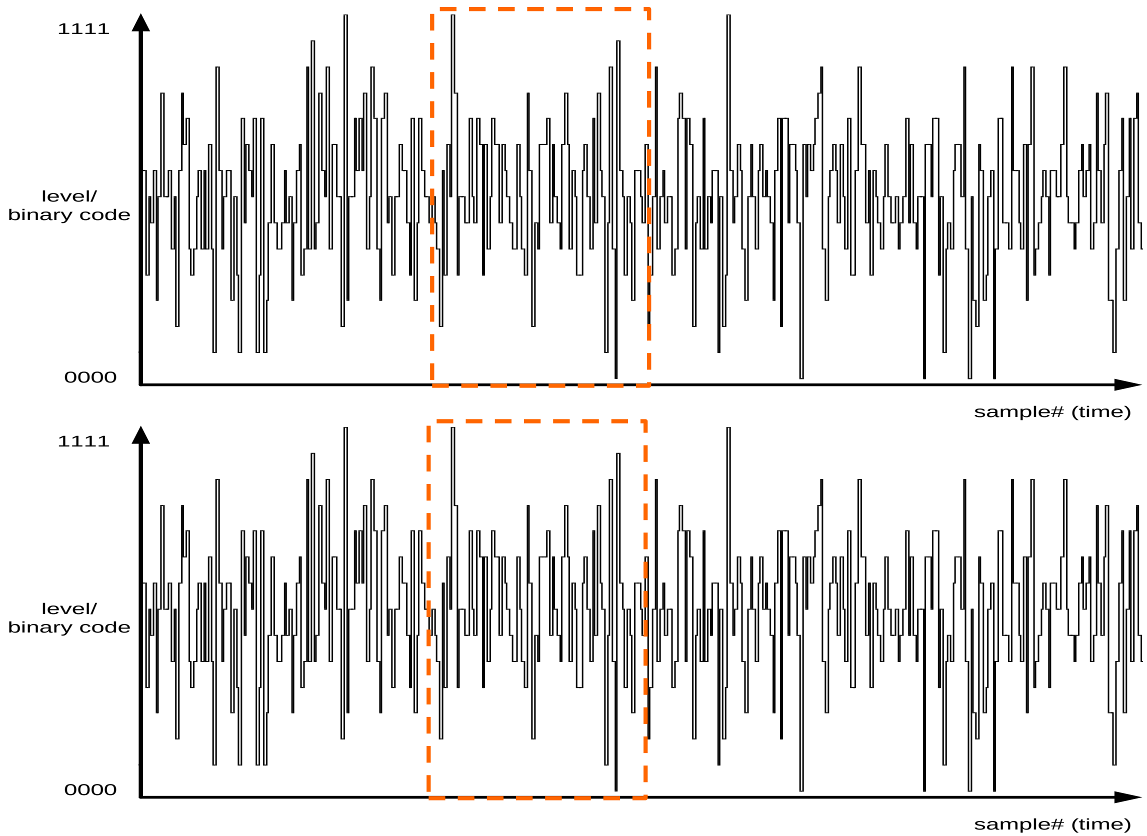
Sensitivity loss:
8 bit: 0.1%
4 bit: 1.3%
2 bit: 12%
1 bit: 36%

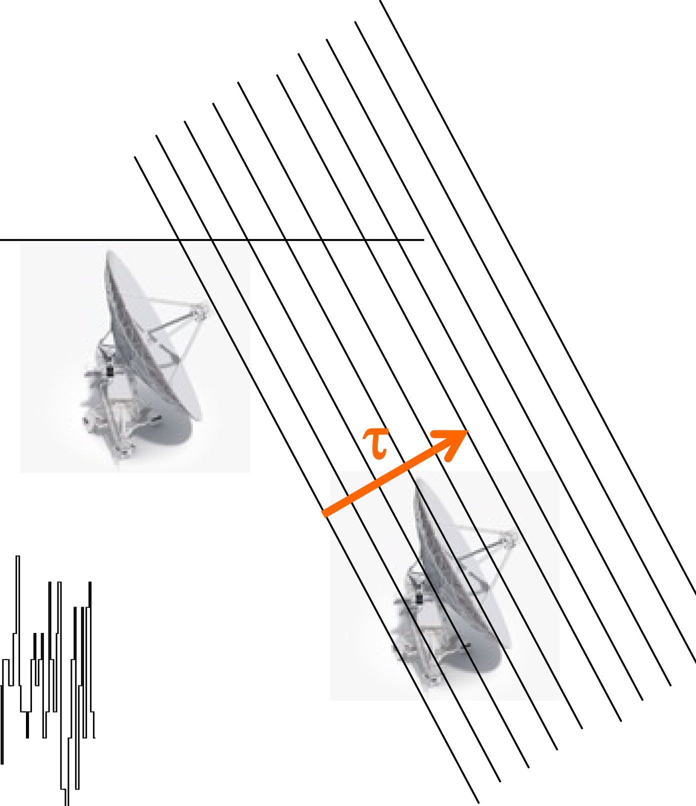
Correct visibility amplitudes for this sensitivity loss (done after correlation, exact correction depends on correlation level)

Delay compensation

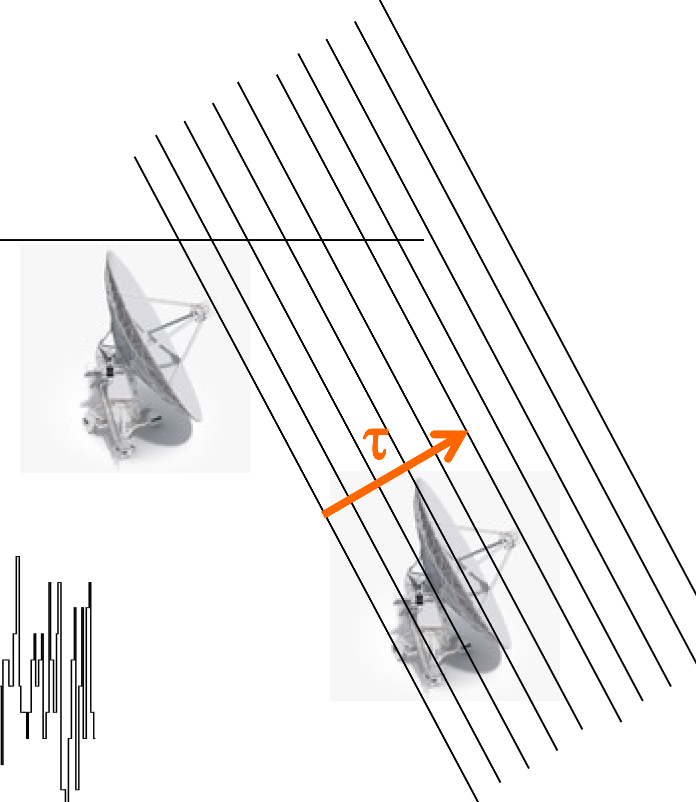
- Delay to the nearest sample is easy:



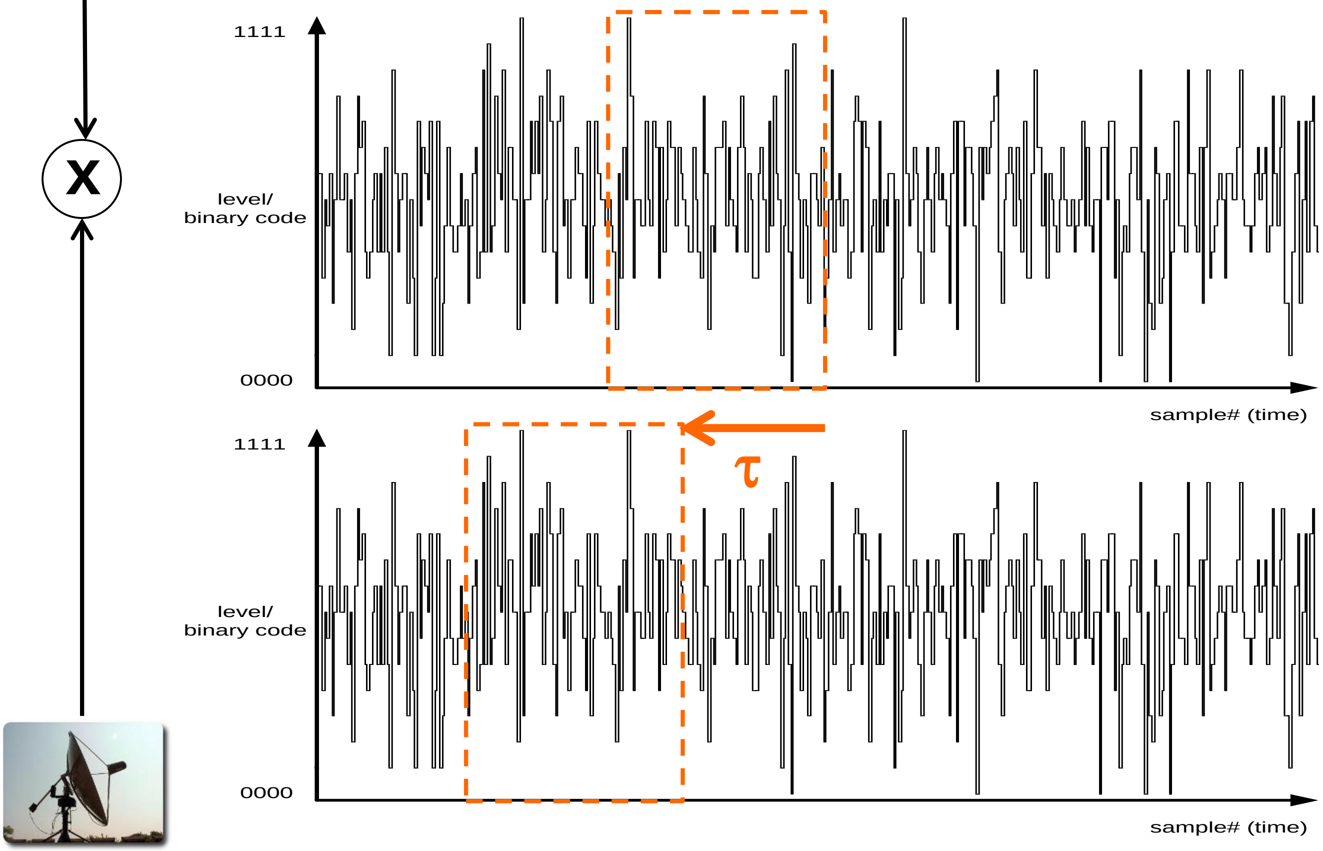




Delay compensation

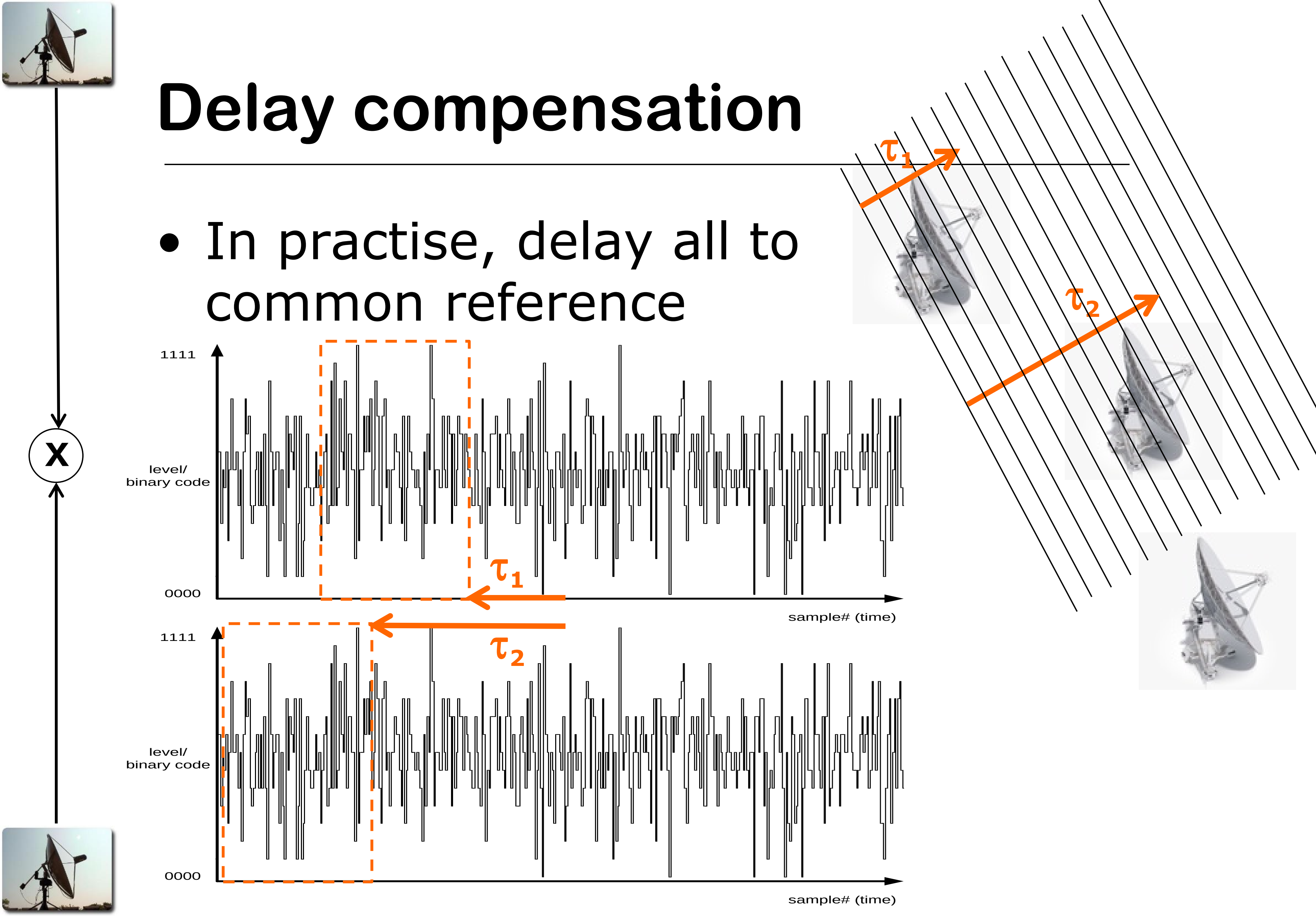


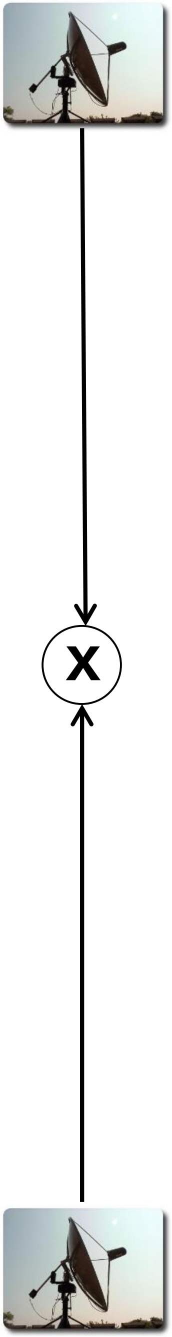
- Delay to the nearest sample is easy:



Delay compensation

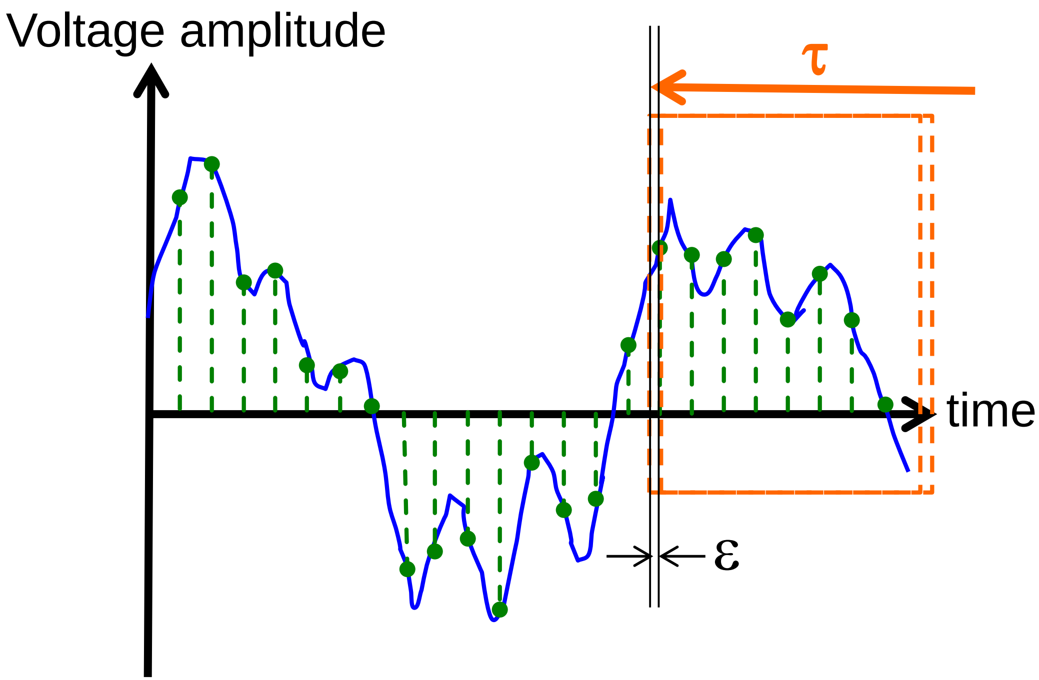
- In practise, delay all to common reference





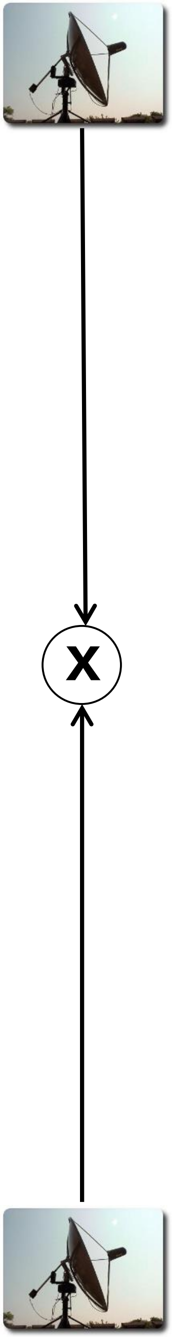
Fractional-sample correction

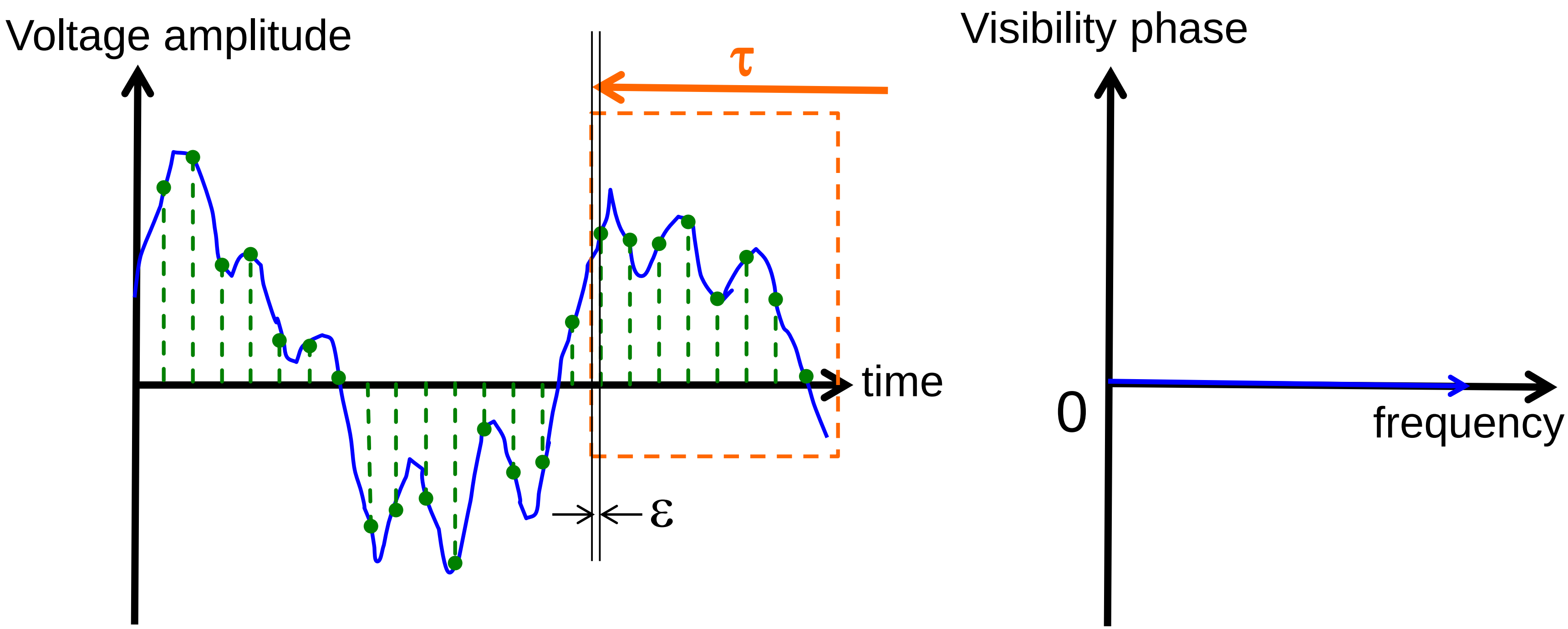
- Sampling prevents perfect alignment of datastreams; always a small error



Fractional-sample correction

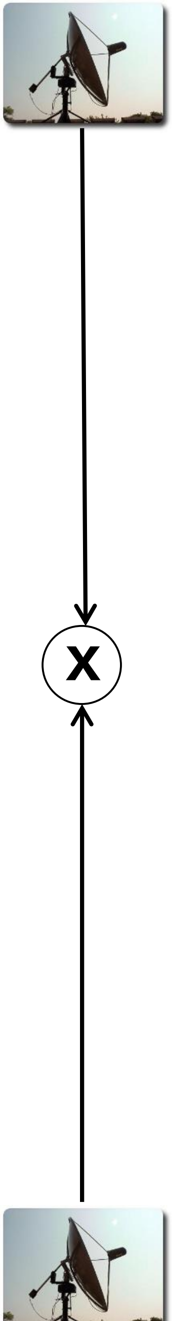
- Sampling prevents perfect alignment of datastreams; always a small error

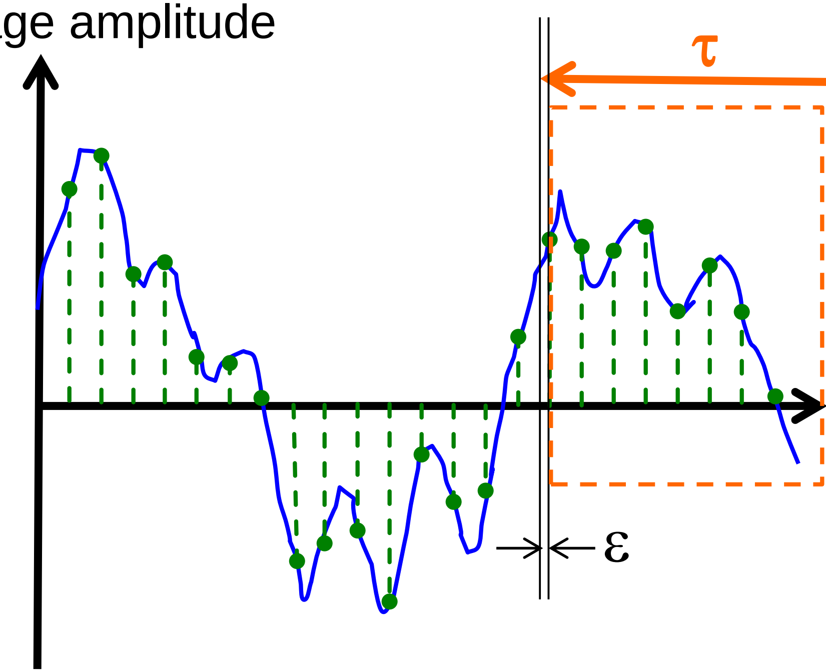


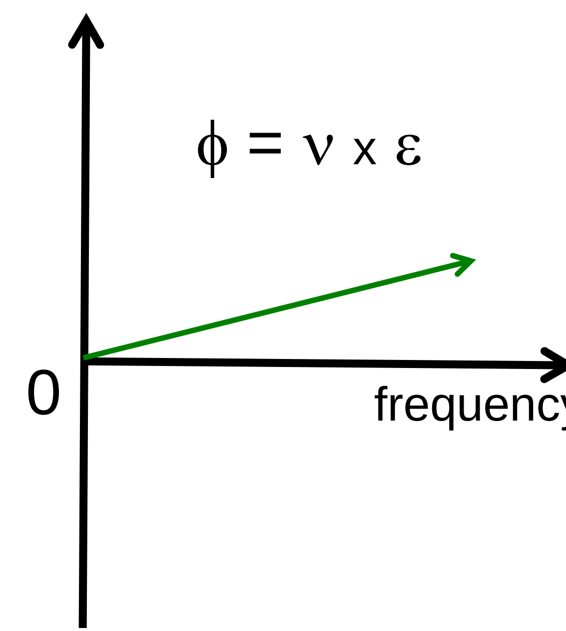


Fractional-sample correction

- Sampling prevents perfect alignment of datastreams; always a small error

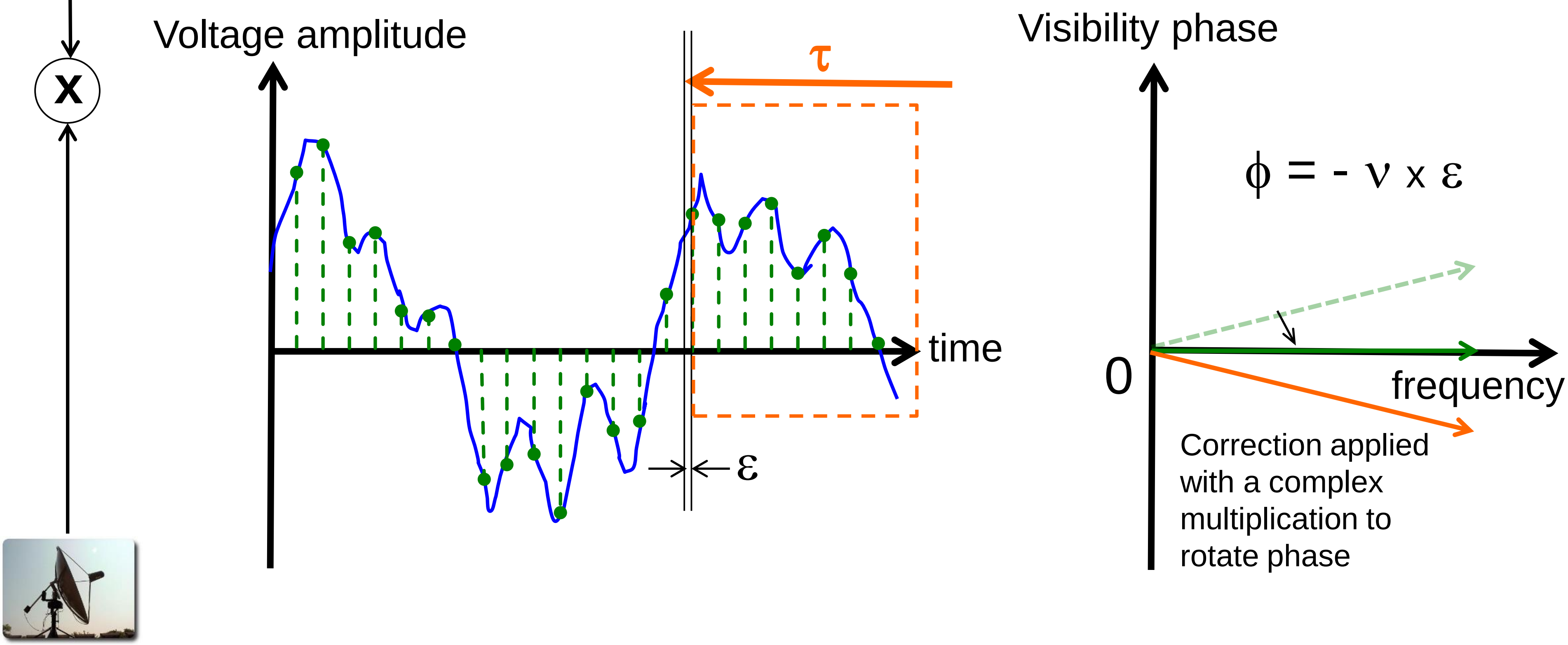




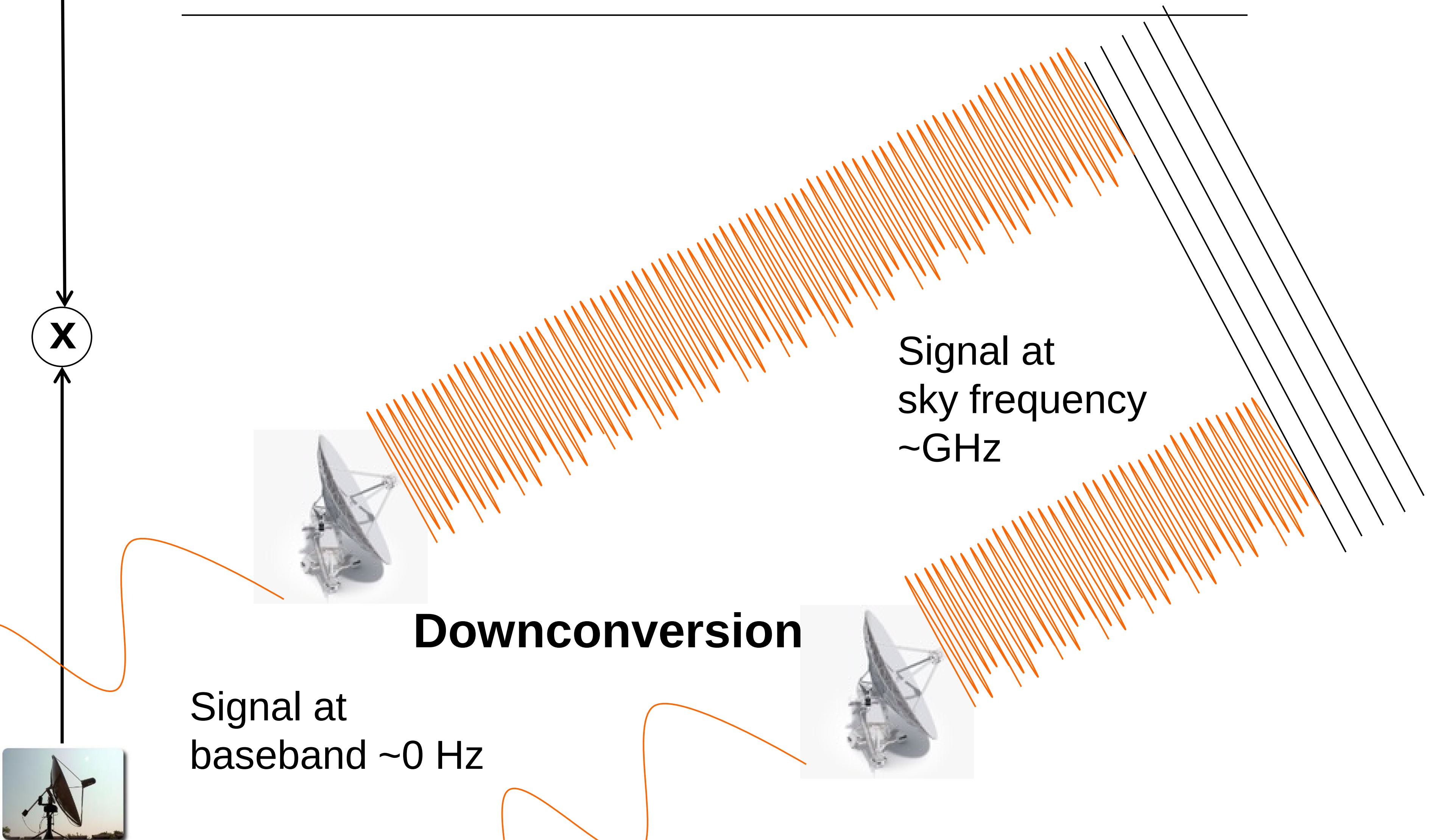


Fractional-sample correction

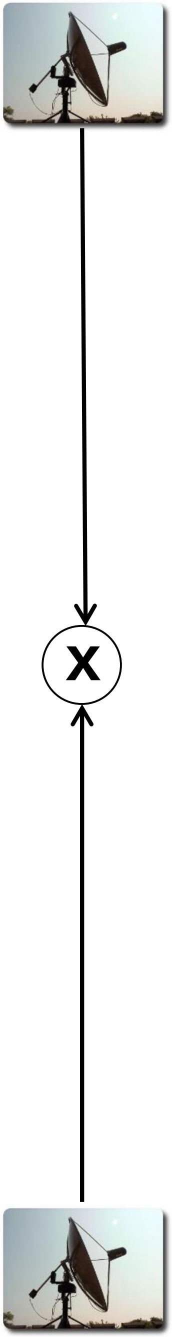
- Sampling prevents perfect alignment of datastreams; always a small error



Fringe rotation

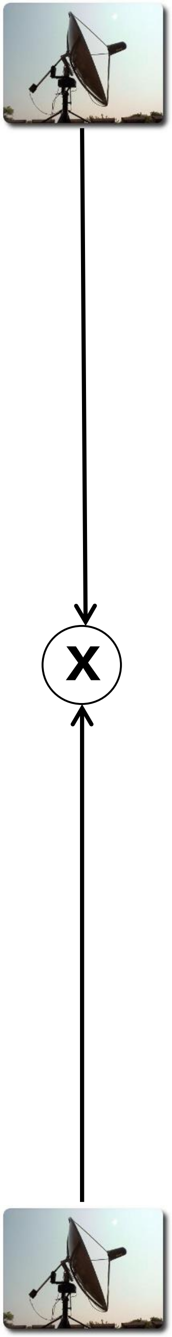


Fringe rotation



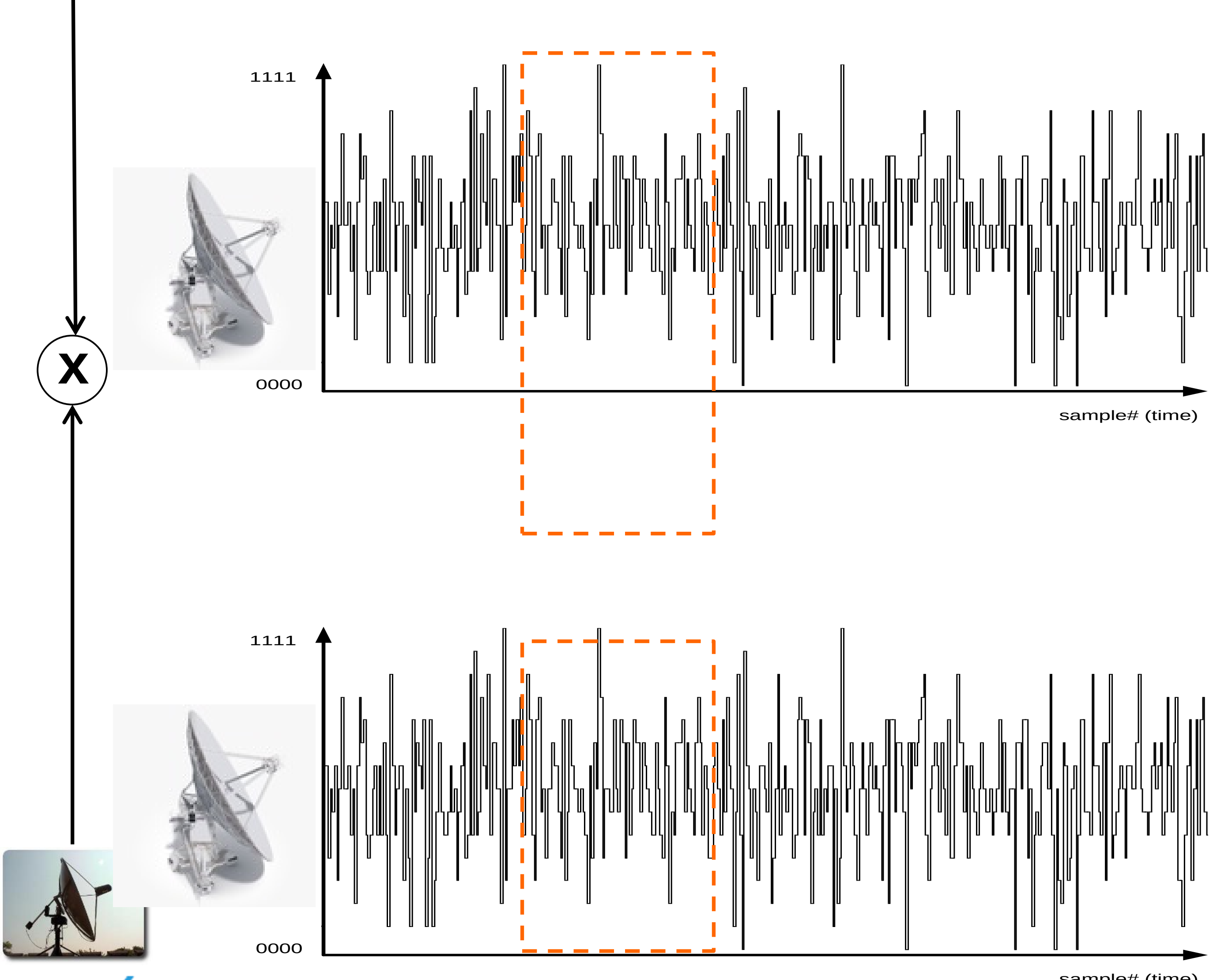
- Implementation: rotate phase using complex multiplier
- $\Delta\phi = 2\pi \nu_{lo} \tau_g$ ν_{lo} = local oscillator frequency, τ_g = applied delay
- Update rate of $\Delta\phi$ depends on how fast τ_g changes:
  - If τ_g is changing fast, update every sample in the time domain
  - For shorter baseline / low frequency instruments, single value per FFT

Alternate implementation

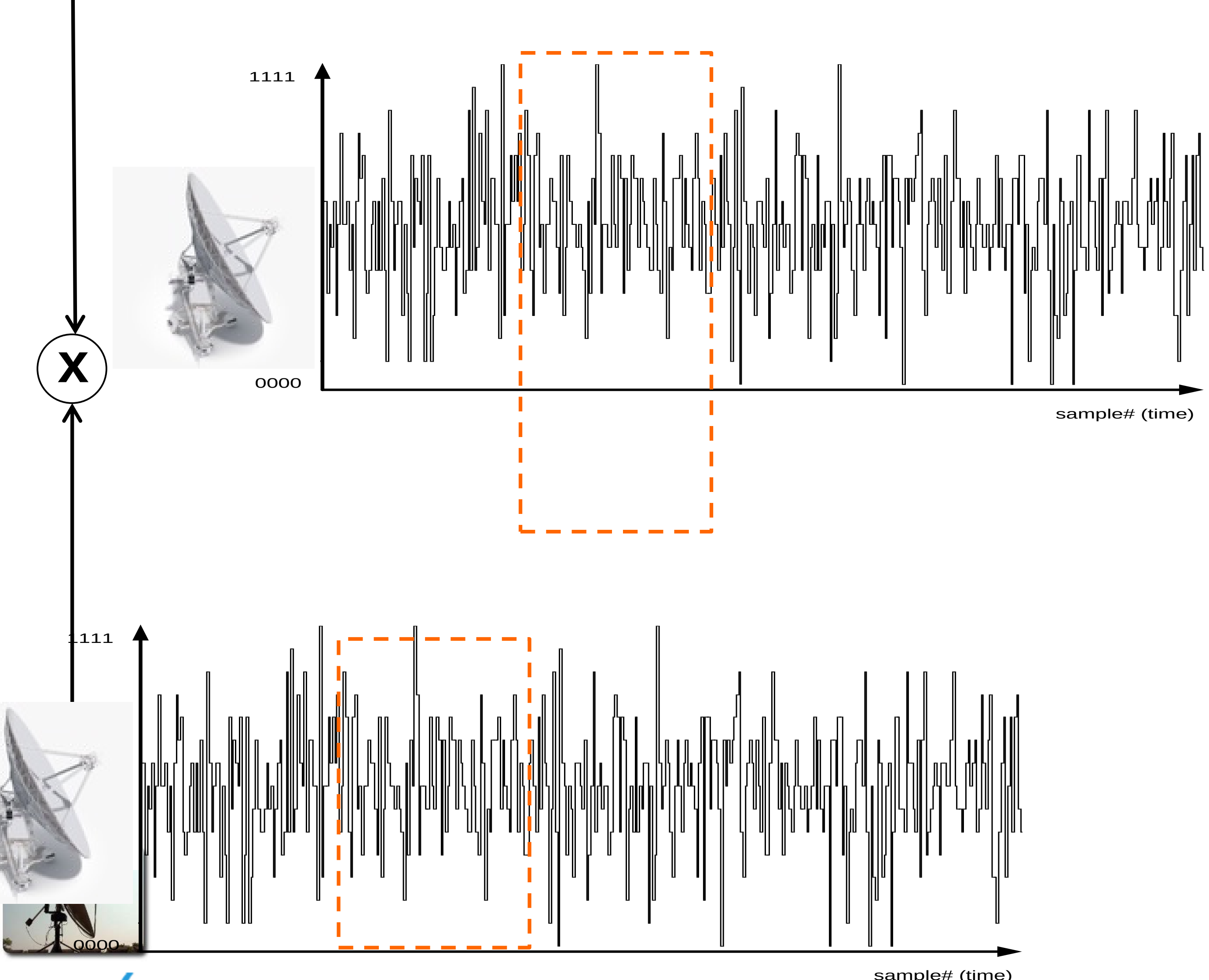


- We have shown how to build a practical FX correlator, which first Fourier transforms and then multiplies
- Convolution theorem: **Multiplication** in the frequency domain is equivalent to **convolution** in the time domain
- It is mathematically equivalent to convolve the two signals in the time domain and then Fourier transform

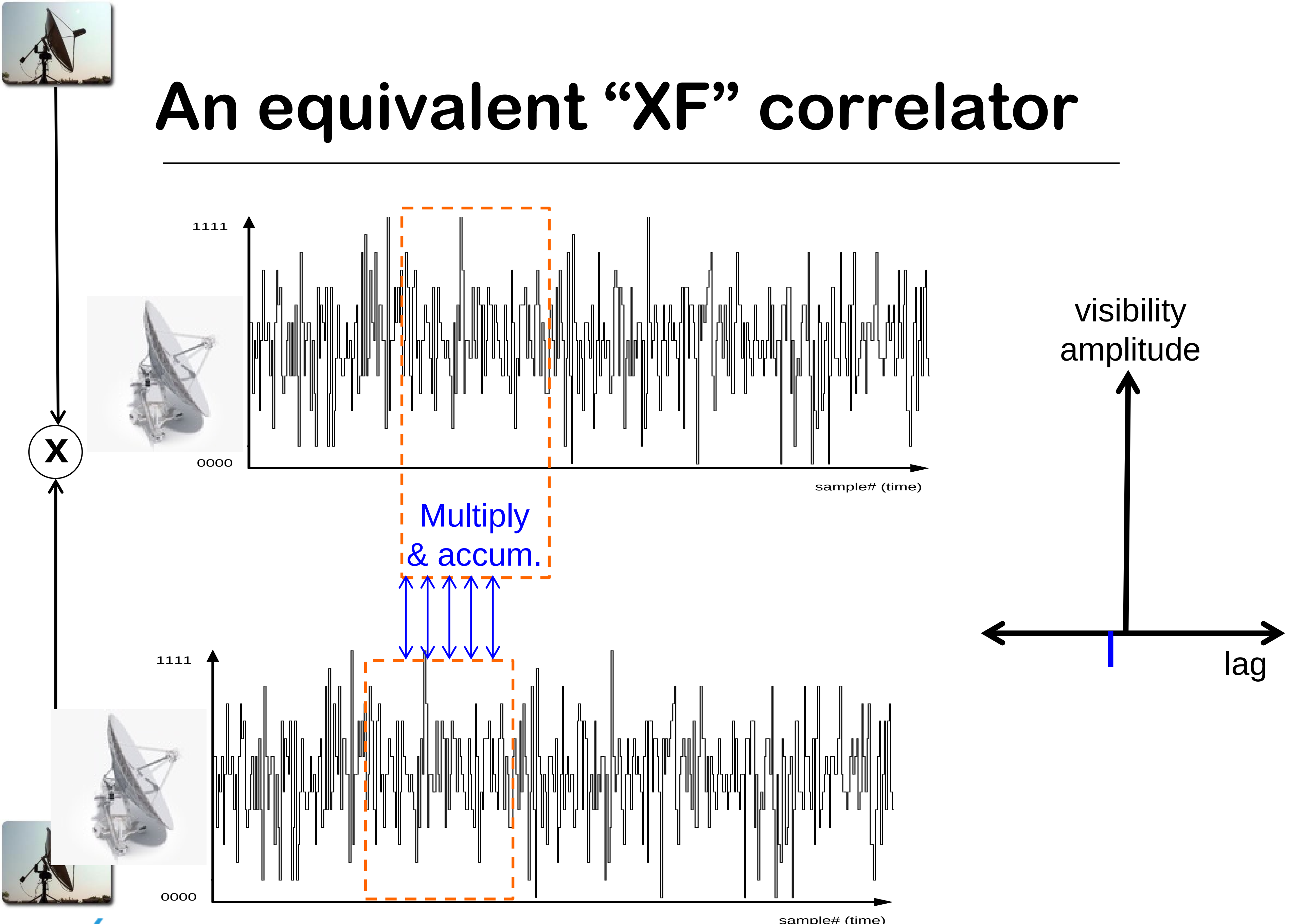
An equivalent "XF" correlator

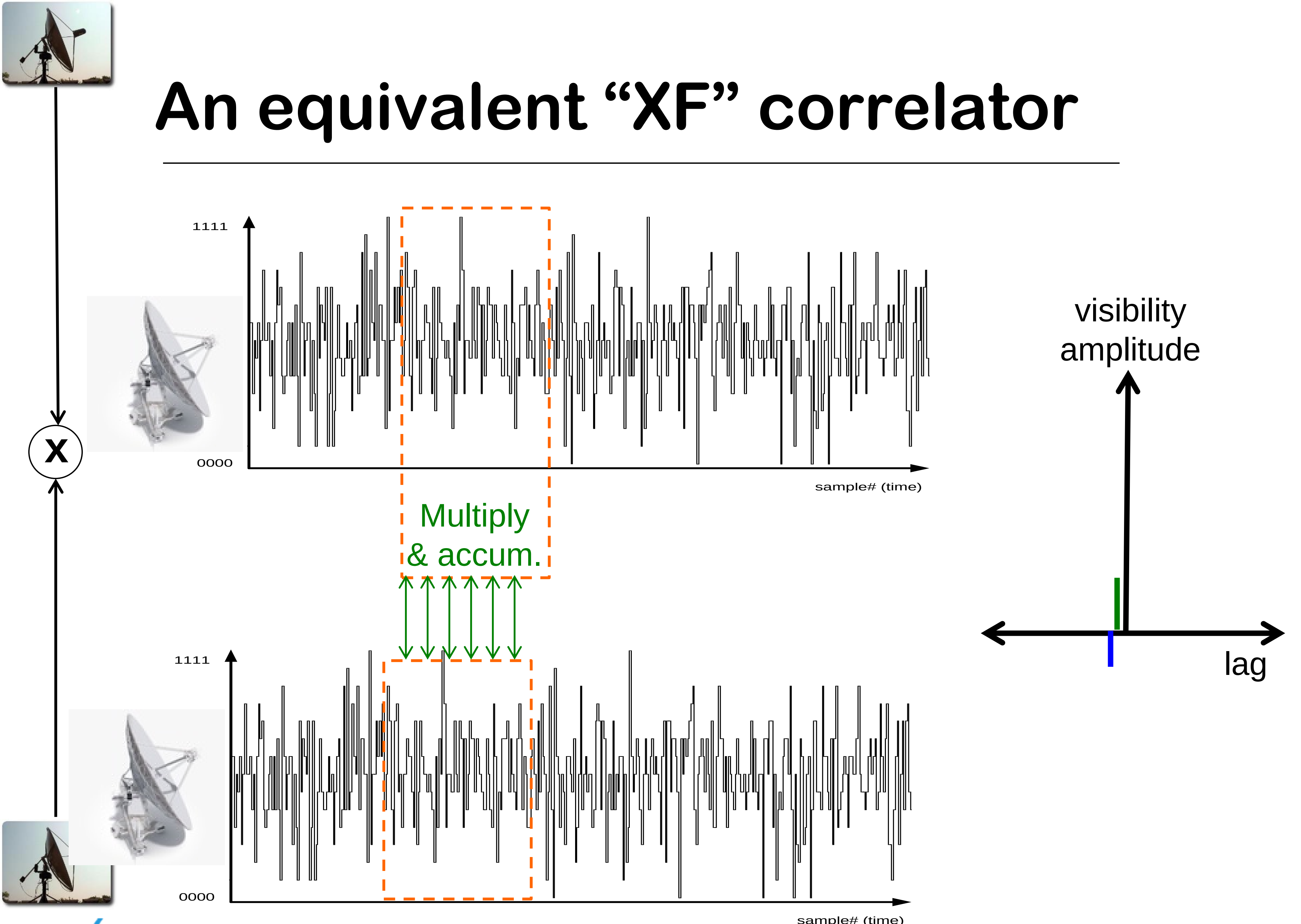
An equivalent "XF" correlator



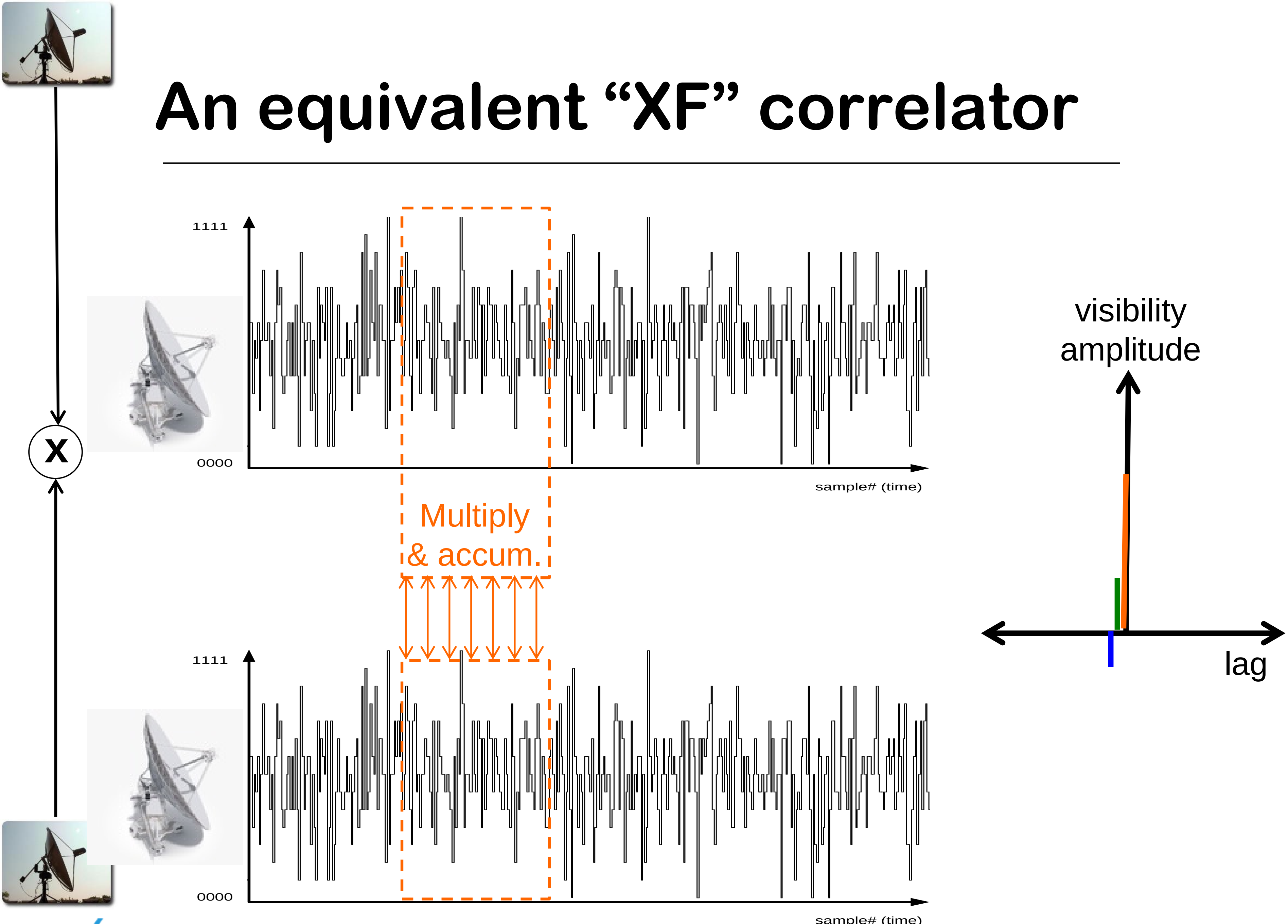
An equivalent "XF" correlator



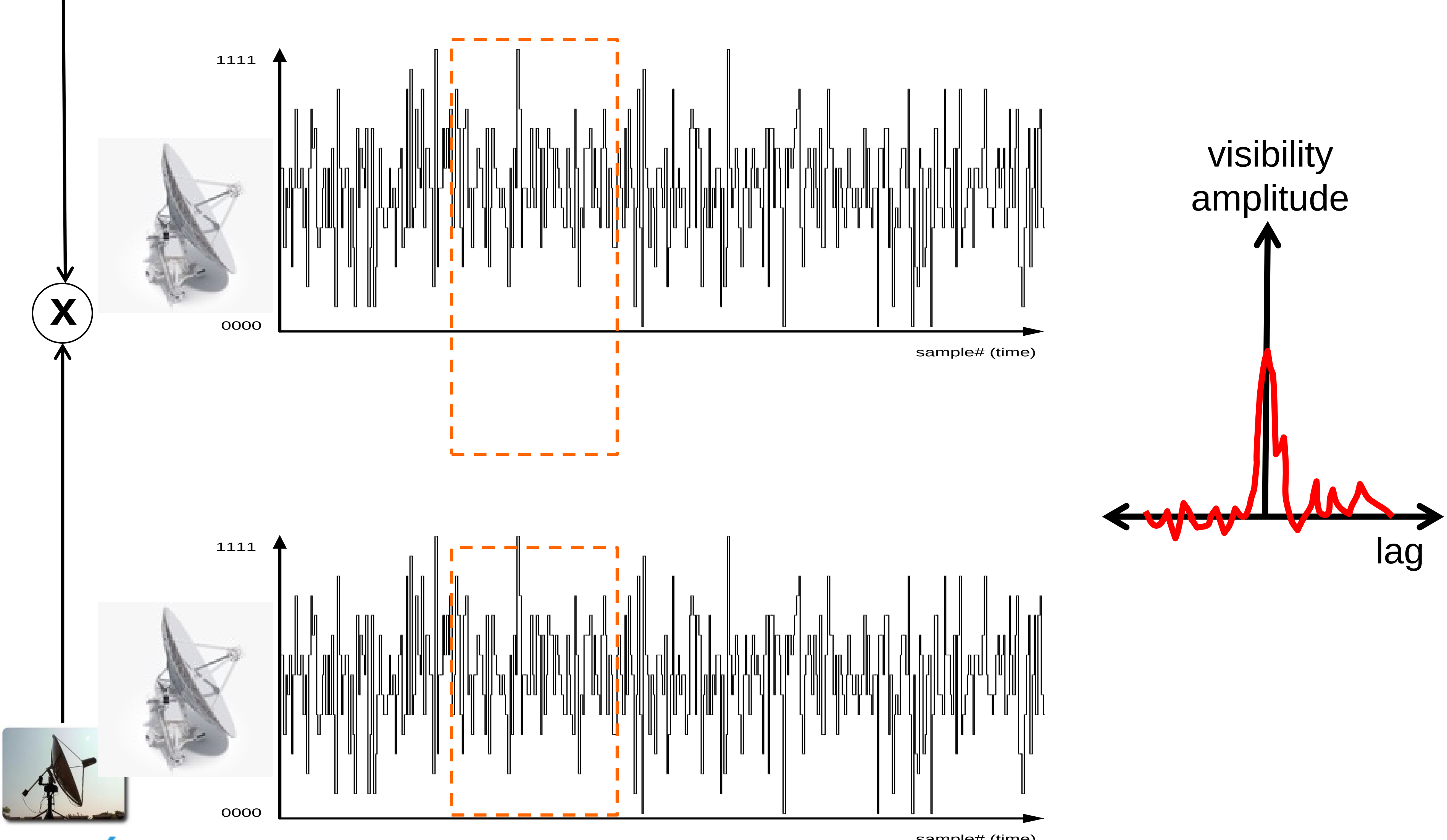
An equivalent “XF” correlator



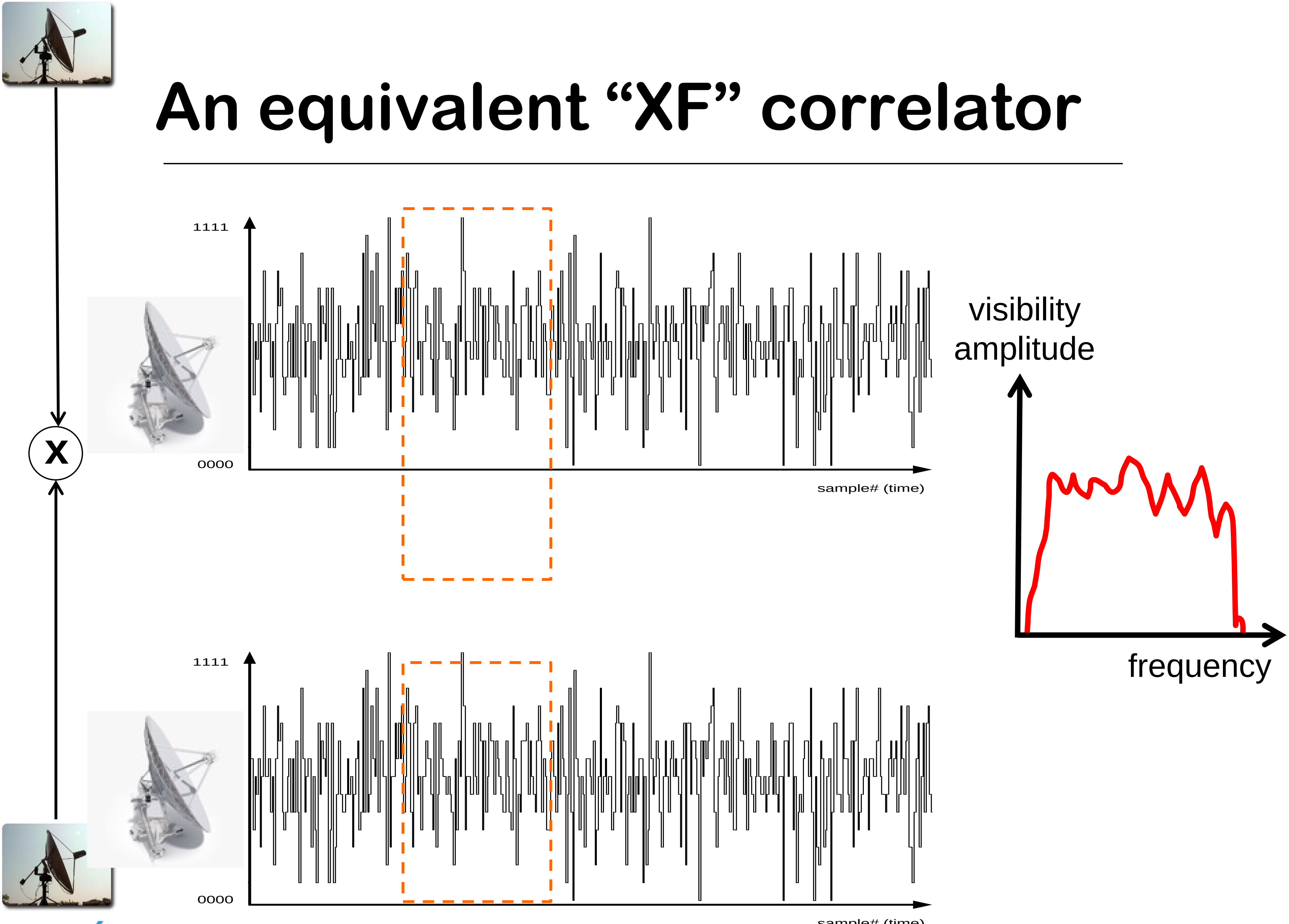
An equivalent “XF” correlator



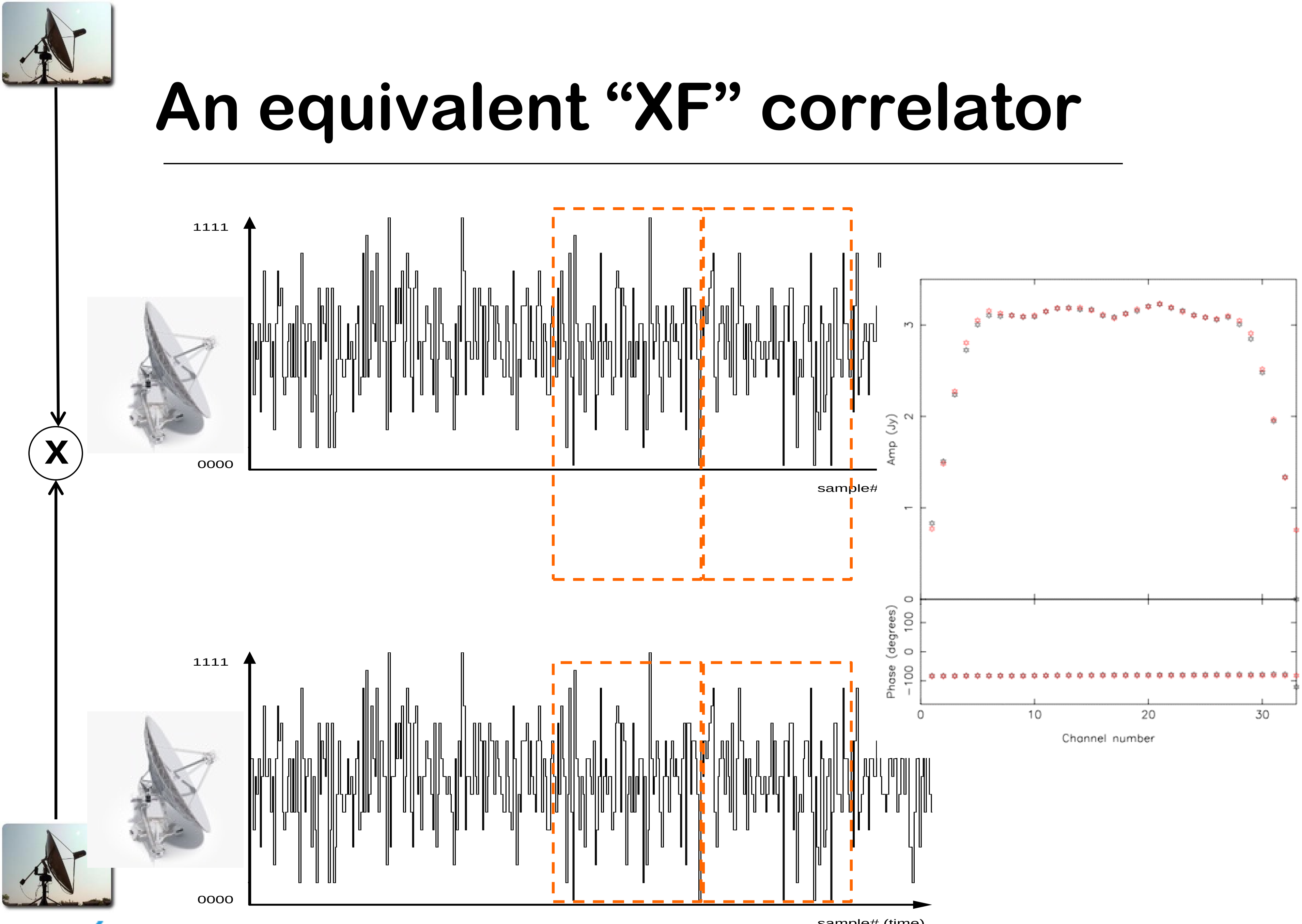
An equivalent "XF" correlator



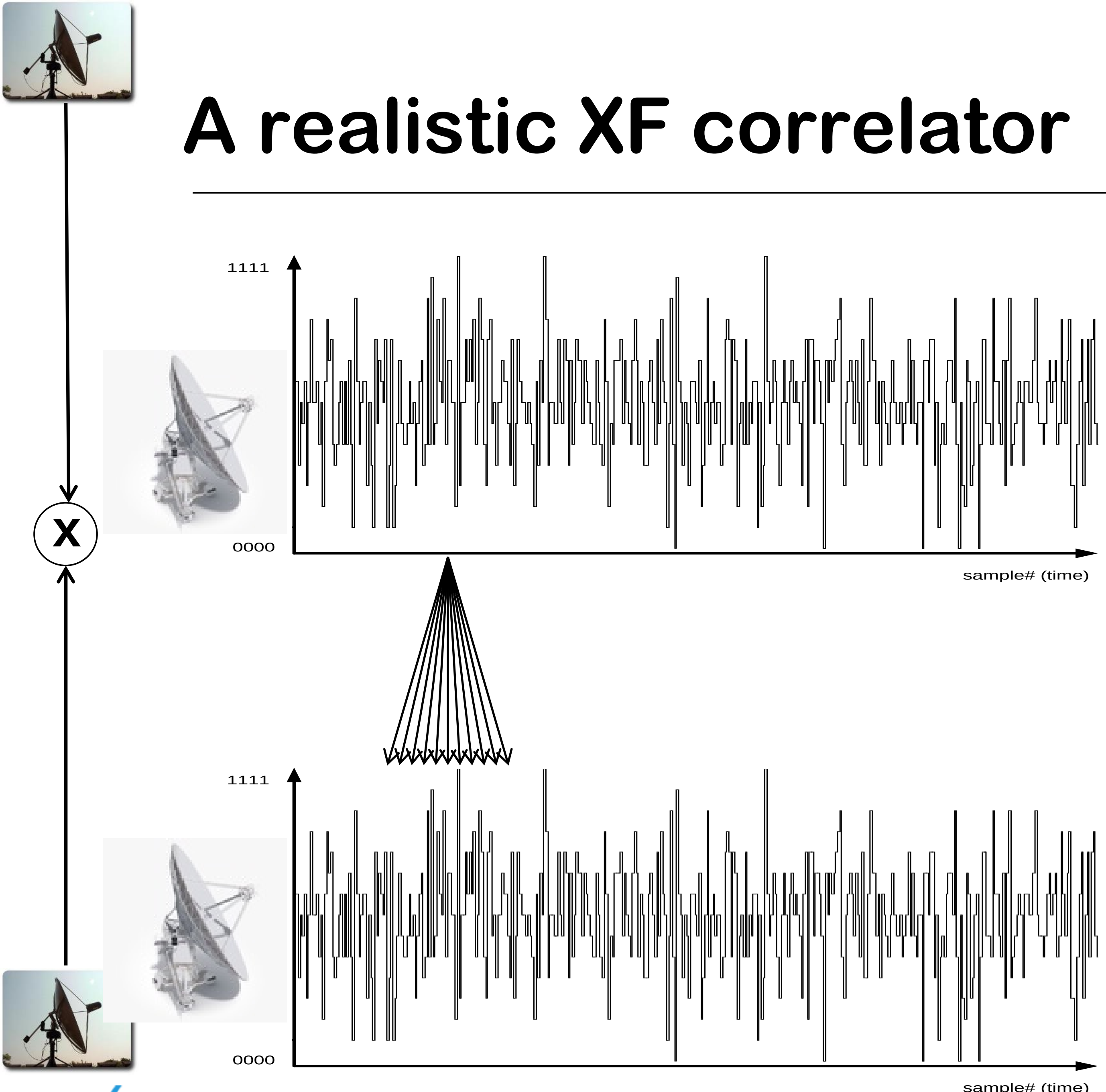
An equivalent “XF” correlator



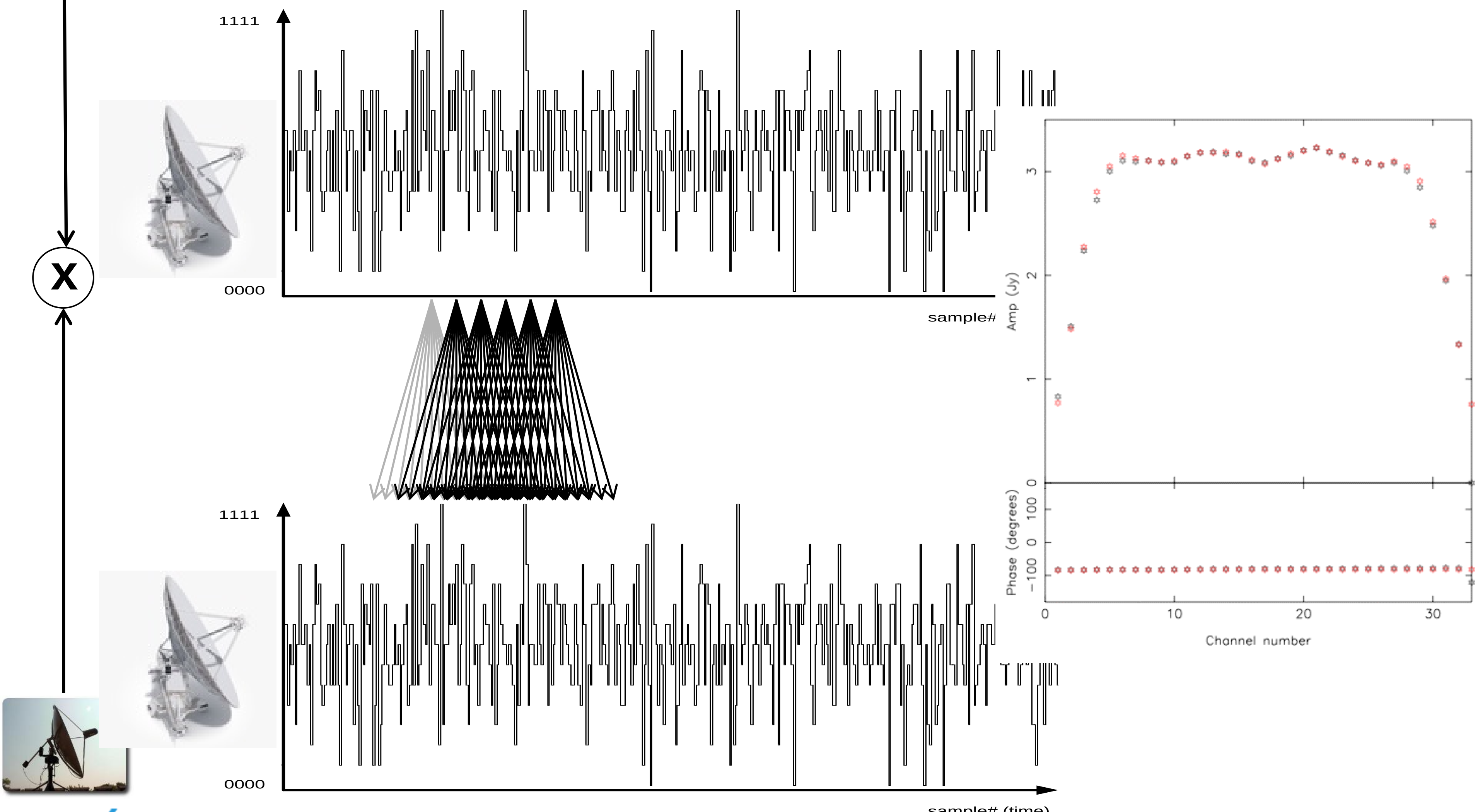
An equivalent "XF" correlator



A realistic XF correlator

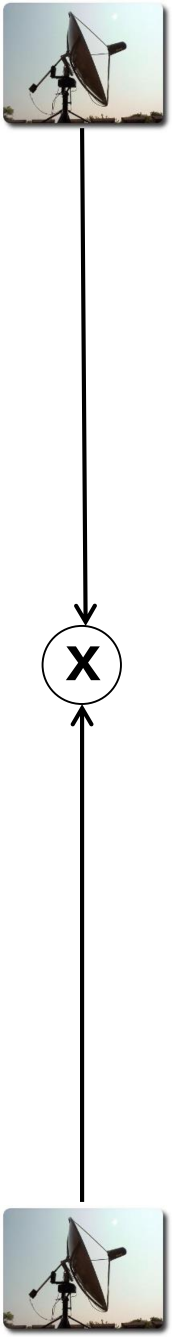


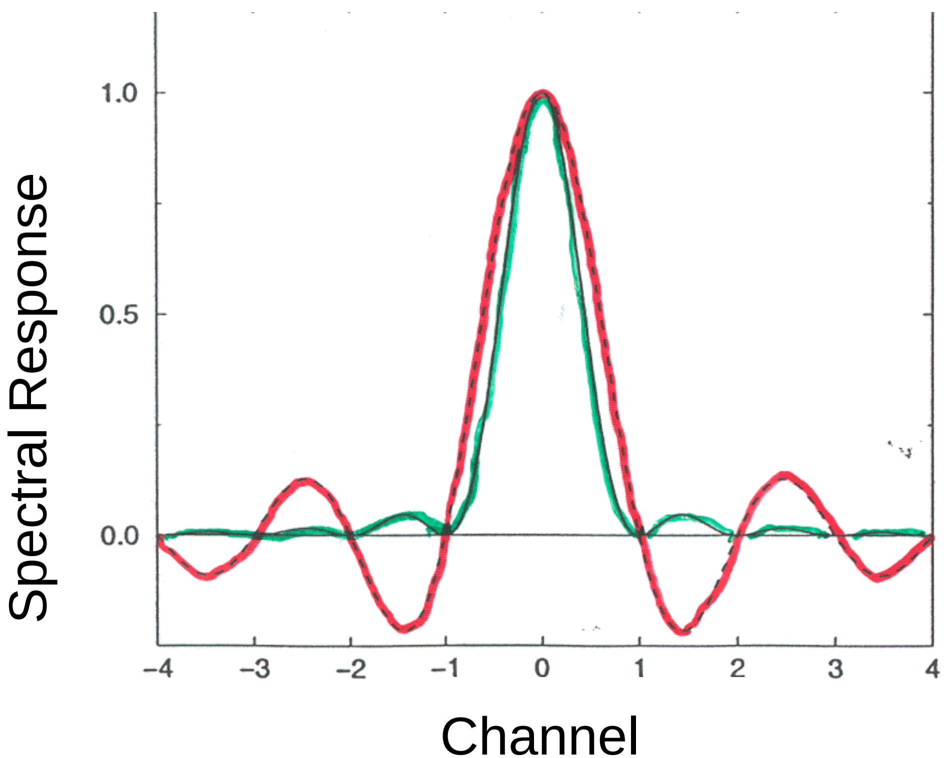
A realistic XF correlator

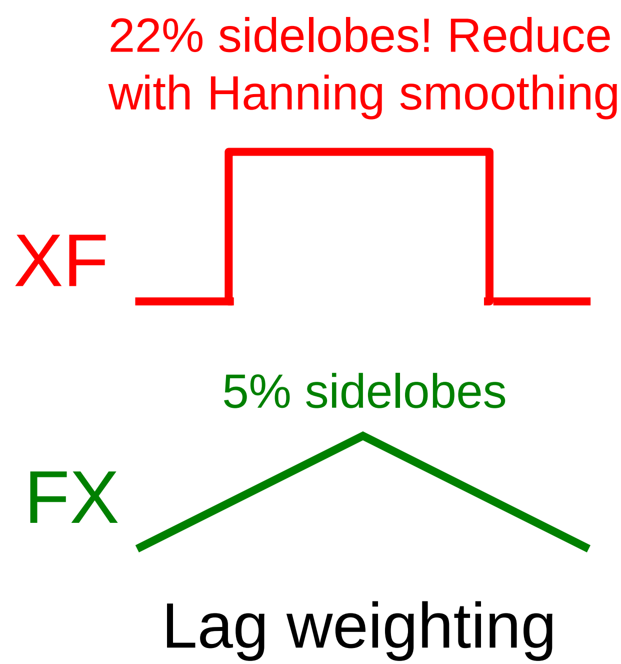


XF vs FX

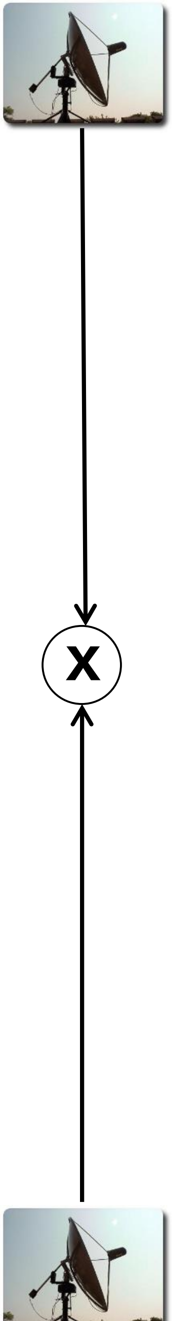
- Different windowing in time domain gives different spectral response





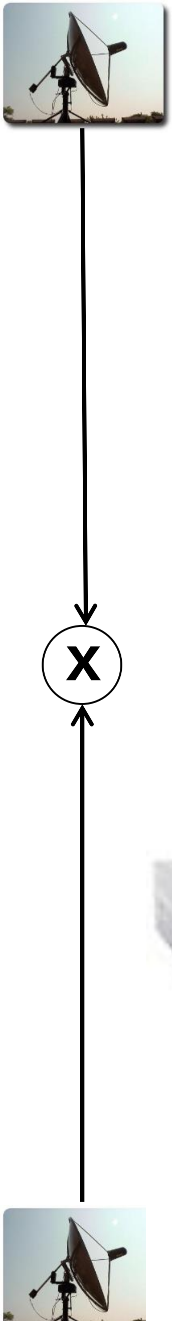


ASTRON



XF vs FX: which is better?

- Advantages and disadvantages to both
  - FX many fewer operations overall
  - XF can make use of very efficient low-precision integer multipliers up-front
  - FX: access to frequency domain at short timescale allows neat tricks and higher precision correction of delay effects
  - But issues with simple implementation of FX for very high spectral resolution
  - Modern correlators mostly FX-style, but use digital filterbank rather than a simple FFT

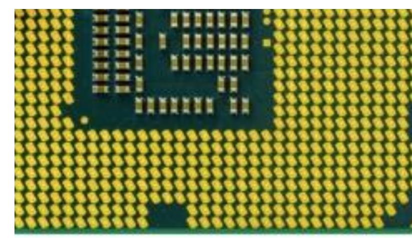


Correlator platforms

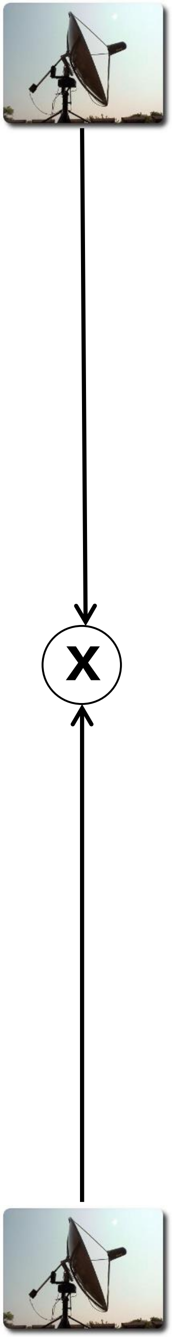
```
status = vectorFFT_CtoC_cf32(complexunpacked, fftd, pFFTSpecC, fftbuffer);
if(status != vecNoErr)
  csevere << startl << "Error doing the FFT!!!" << endl;

     ...

status = vectorAddProduct_cf32(vis1, vis2, &(scratchspace->threadcrosscorrs[resul
```



Correlators on CPUs



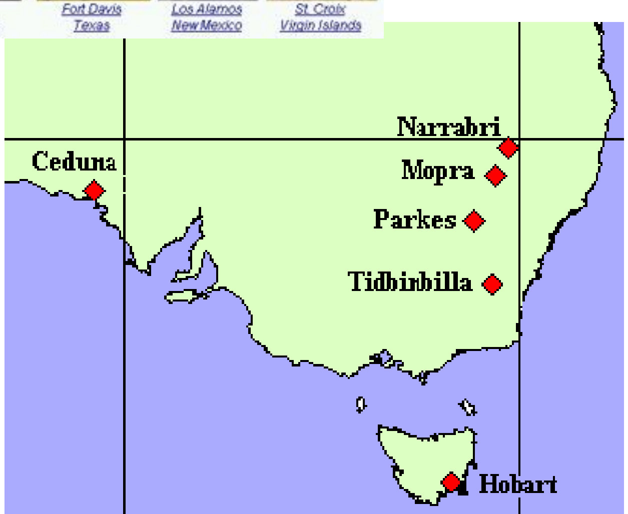
- Many positive points:
  - Can implement in "normal" code (e.g., C++); maintainable, many skilled coders
  - Development effort transferrable across generations of hardware
  - Incremental development is trivial
  - Natively good at floating point (good for FX), no cost to do high precision
- One major disadvantage:
  - CPUs not optimised for correlation; big system like JVLA would take **many** CPUs.

Correlators on CPUs

The Very Long Baseline Array, 10 stations

GMRT, India, 30 stations

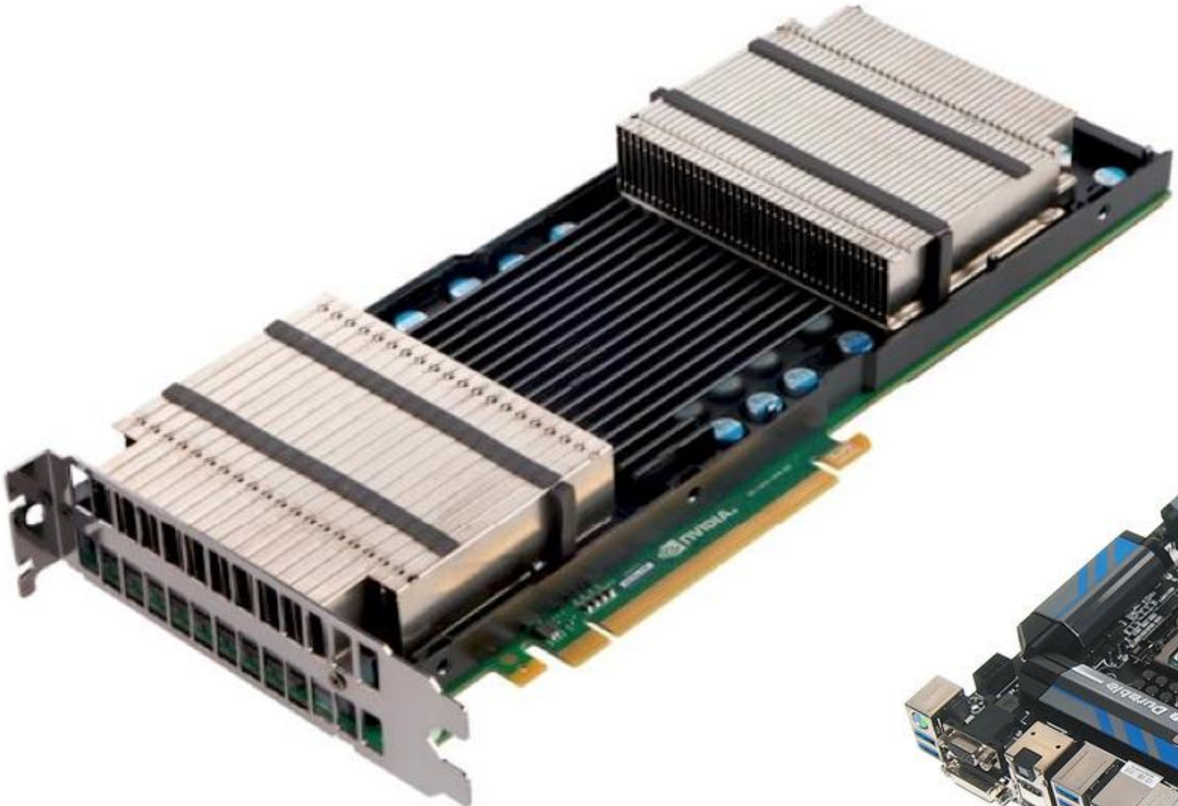
The Long Baseline Array, Australia, ~6 stations



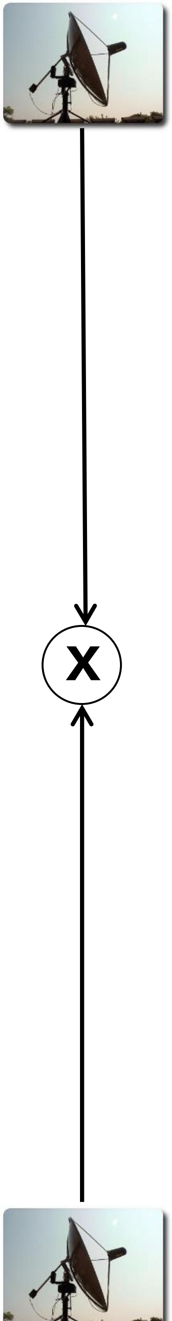
The European VLBI Network, ~20 stations

ASTRON

Correlators on GPUs



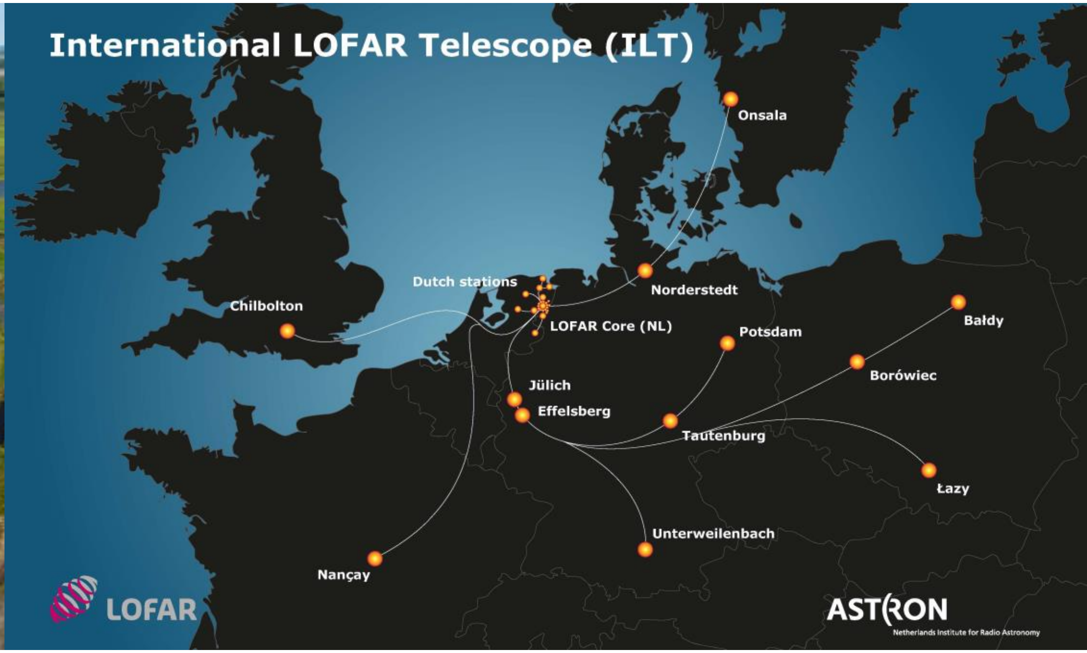
Like CPUs, GPUs are mounted on a standard motherboard



Correlators on GPUs

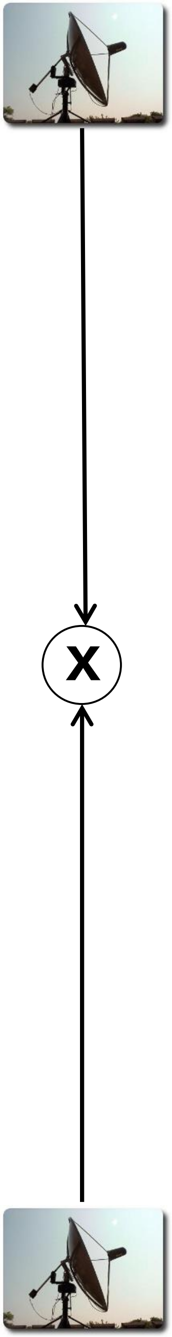
- Advantages:
  - More powerful and more efficient than CPUs
  - Also good at floating point
- Disadvantages:
  - Writing code is more difficult (GPUs are more specialized, less flexible: need to carefully manage data transfers)
  - Fewer trained GPU programmers available
  - Transfer-ability of code across hardware generations not yet reliable (but close)

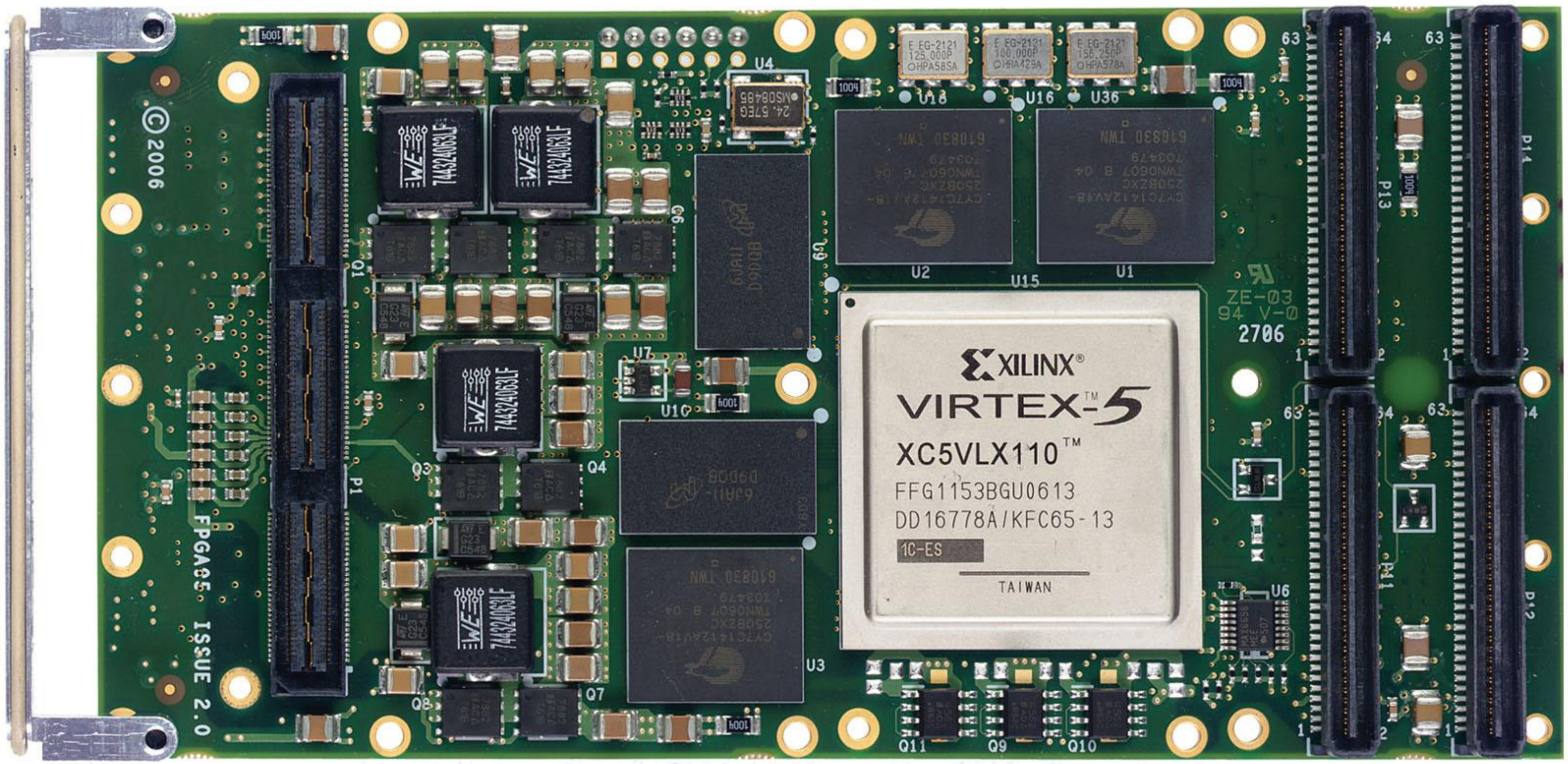
Correlators on GPUs



The Low Frequency Array (LOFAR), 73 stations

Correlators on FPGAs

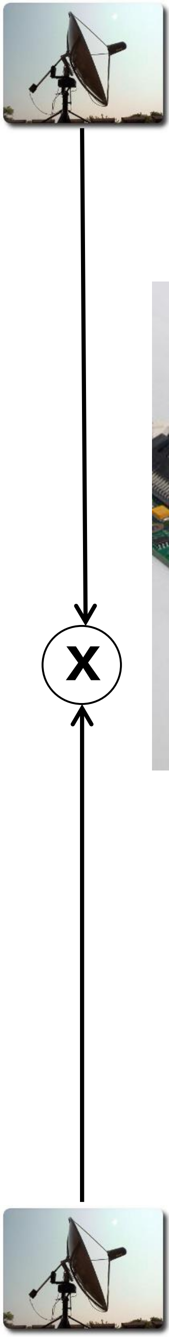




Correlators on FPGAs

- Advantages:
  - More efficient than CPUs or GPUs, particularly for integer multiplication
- Disadvantages:
  - Programming is harder again (especially debugging), yet fewer trained people
  - Transfer-ability across hardware generations even more limited
  - Synchronous (clocked) system, less robust to perturbations c.f. CPUs/GPUs

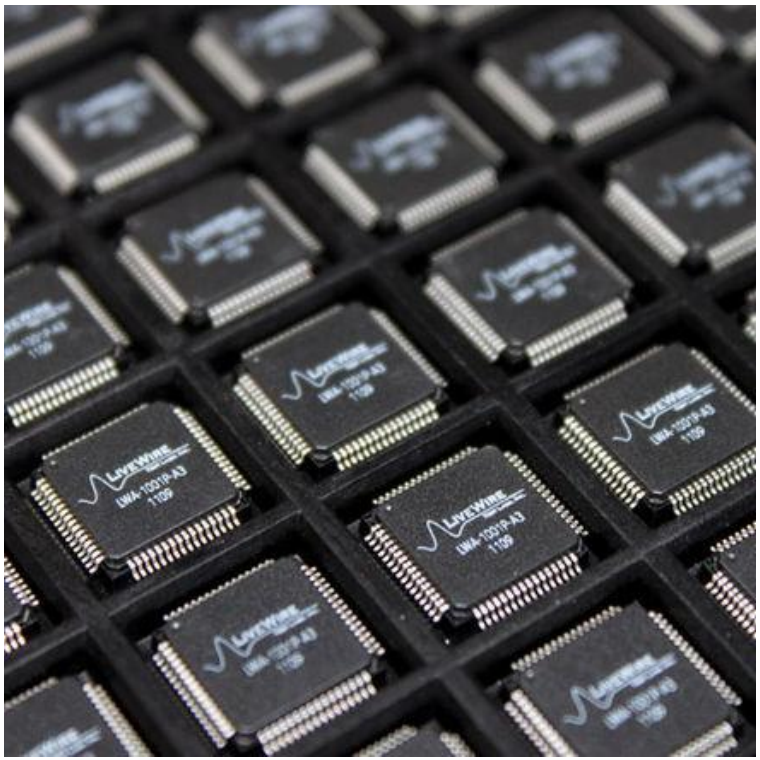
Correlators on FPGAs

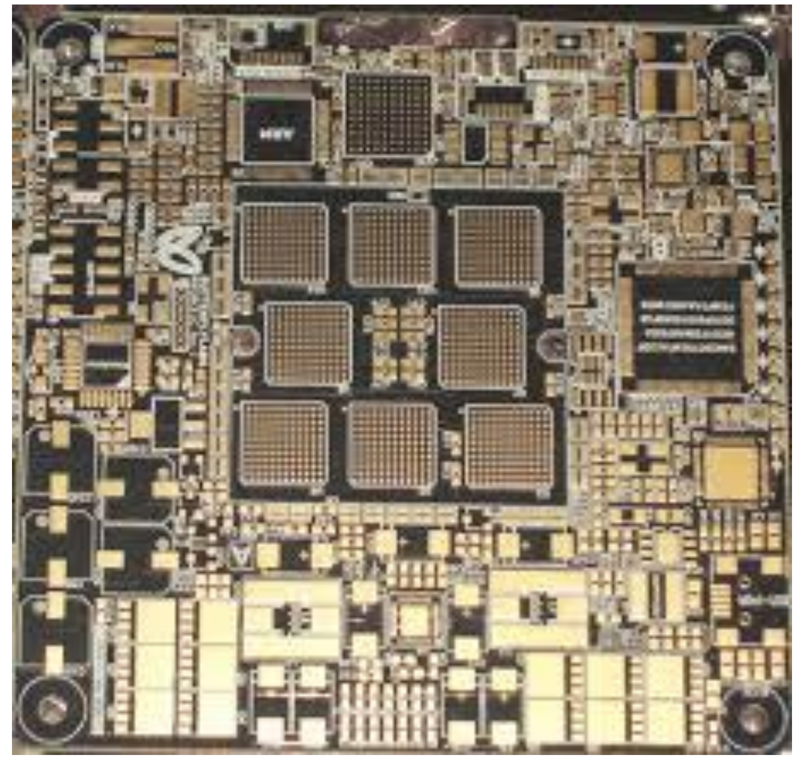


"Roach" reconfigurable FPGA board used for correlation

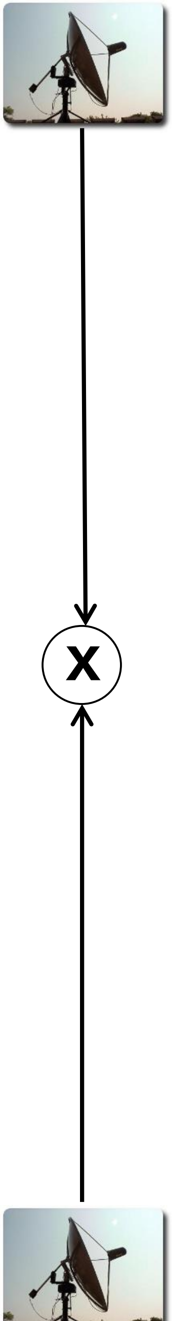
MeerKAT, 64 dishes, under construction

Correlators on ASICs



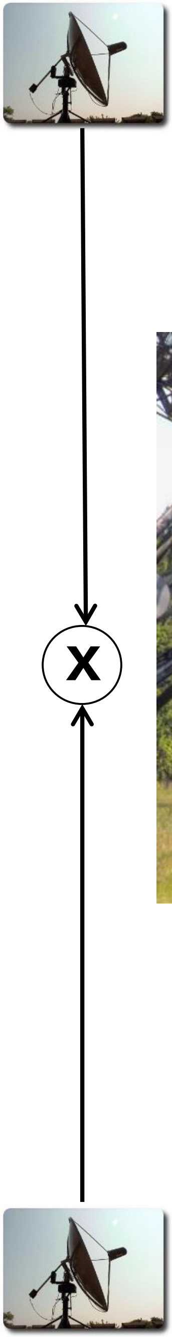


As with FPGAs, ASICs are mounted on boards



Correlators on ASICs

- Advantages:
  - Highest possible efficiency, low per-unit cost
- Disadvantages:
  - Highest development cost (time and manufacturing setup)
  - Specialized knowledge required
  - Can’t be changed / very difficult to upgrade during lifetime



Correlators on ASICs

The Westerbork Synthesis Radio Telescope, Netherlands

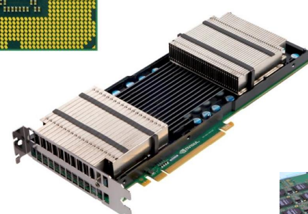
The Very Large Array, New Mexico

Correlator platform overview

Development effort required

Reuse-ability

CPU



GPU

FPGA

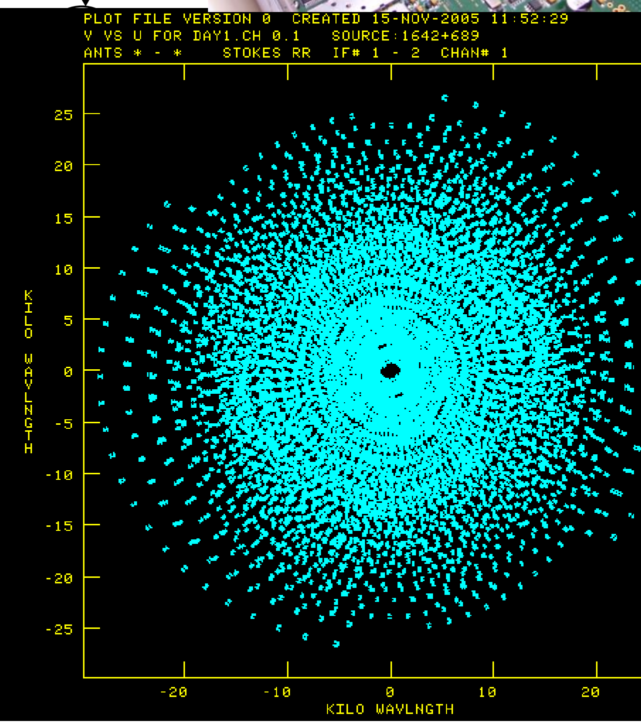
ASIC

Correlator capacity per hardware $$

X

ASTRON

The end



PLOT FILE VERSION 0 CREATED 15-NOV-2005 11:52:29
V VS U FOR DAY1.CH 0.1   SOURCE:1642+689
ANTS * - *   STOKES RR IF# 1 - 2 CHAN# 1
25
20
15
10
5
0
-5
-10
-15
-20
-25
KILO WAVLNGTH
-20
-10
0
10
20
KILO WAVLNGTH