Development of EPICS Embedded Image Processing System

Takashi Obina, Jun-ichi Odagiri, Ryota Takai

KEK, Accelerator Laboratory

Outline

- Requirements for Image Processing
- Our Solution
- Driver/Device Support
- Display
- Application
- Conclusion

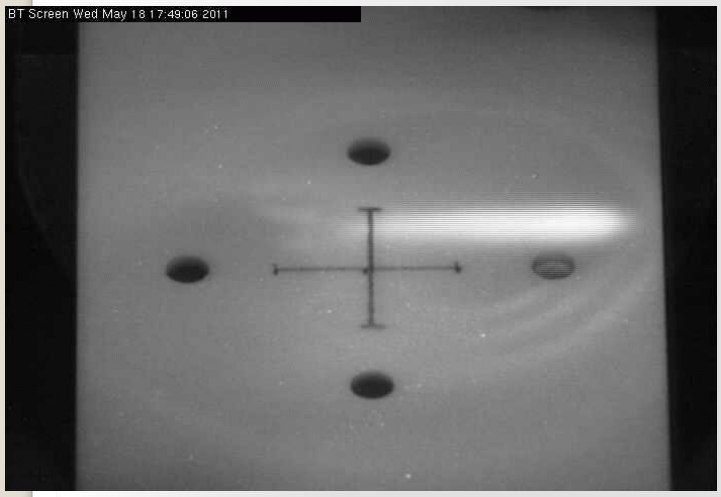
Requirements for Image Processing

Requirements from operation :
NOT so severe

- EPICS !
- Stable operation
- Update Rate
  - at least 1Hz; Faster is better
  - 10Hz is enough for most purpose (for Human-based feedback or tuning)
  - 30 Hz is not necessarily required
  - Needs external trigger
- Background subtraction for Screen Monitor
- Capability for complex calculation
  - Beam Size Monitor with the SR Interferometer
  - Nonlinear curve fitting is required

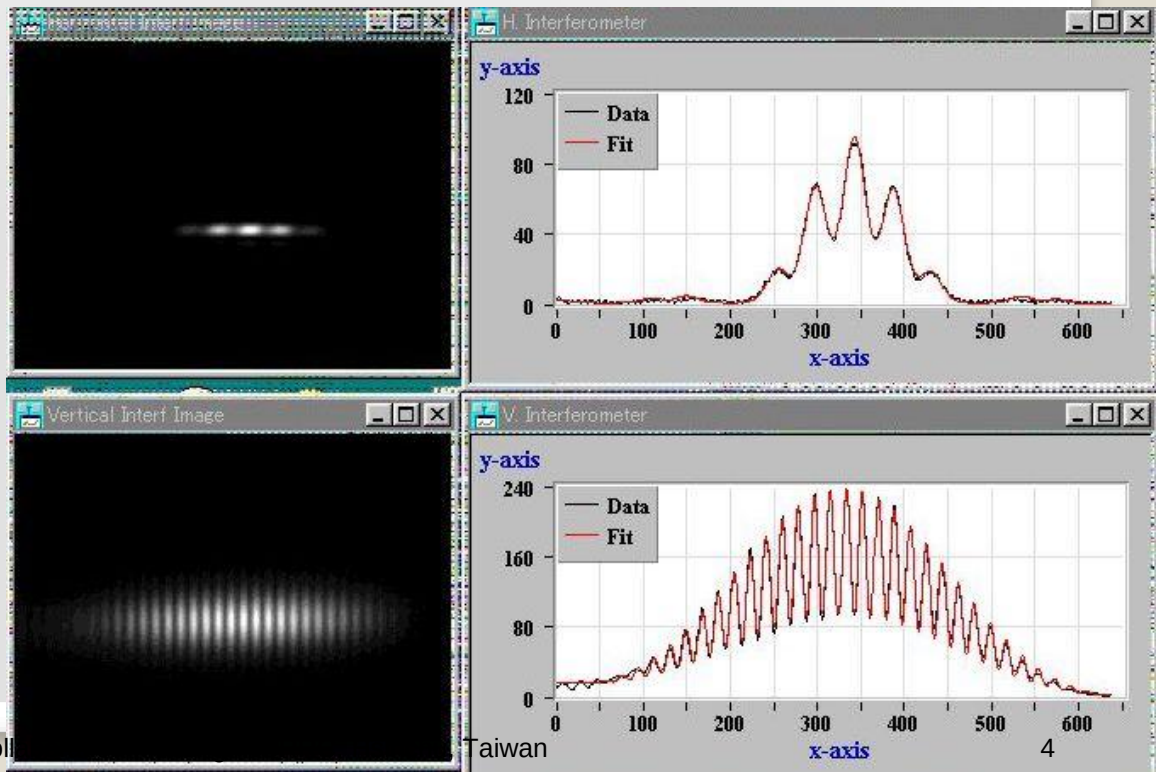
Example images : KEK-PF

ScreenMonitor (Beam Transport Line)



SR Monitor

SR Interferometer (Beam Size Monitor)



There are so many possibilities …

- Hardware
  - PC / Embedded platform
- Operating System
  - Linux / Windows
- Interface
  - Image Grabber Board (PCI, PCIe, etc)
  - USB / FireWire (IEEE1394)
  - GigE (Ethernet)
  - Camera Link
- Camera
  - Speed (Frame Rates [fps])
  - Resolution [pixels] / number of bits
  - CMOS or CCD

What was important for us?

- Stability
- Rapid application development
- Cost Effectiveness
- Support of colleagues and/or company
- Long-term operation
  - “Lifetime” of software and hardware

Our misery experience

- We have been used Windows PC and frame grabber for interferometer analysis
- The software is 'discontinued'
  - Hardware/Software Only supports Win98!
- Didn’t work after the “Windows Update”

MUST be avoided in the new system

Our Solution

- Yokogawa PLC-Based System
- CPU : F3RP61 with Linux OS
- Image Acquisition Module (UM02)
  - PCI Interface to CPU

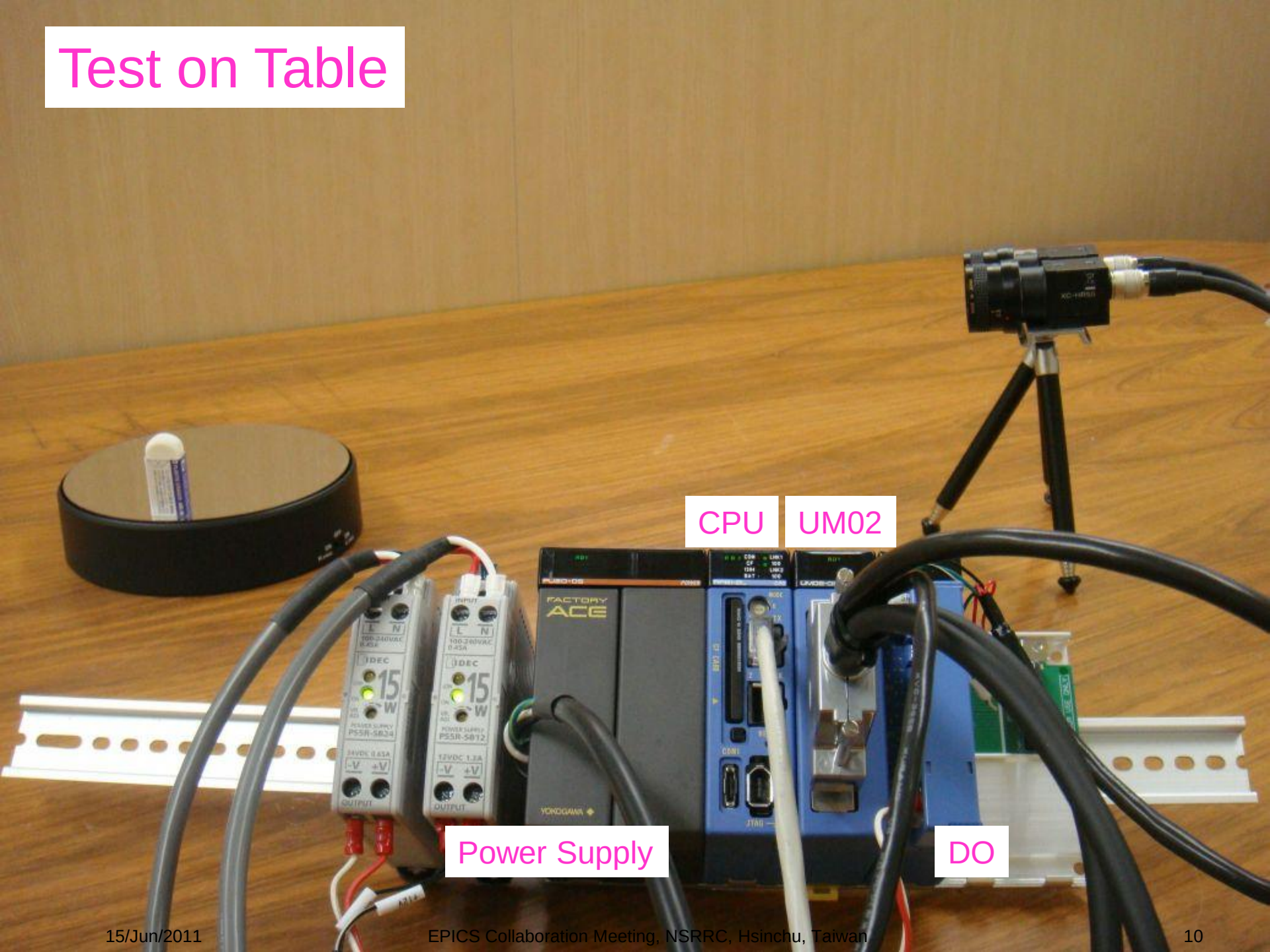
This system is supposed to fulfill the requirements. Especially for “Hardware Reliability” and “Product Lifetime”, and “support of colleagues”

Benefit of PLC-Based system

- Many Digital and Analog I/O module
  - cheaper than VME
- We can easily control other equipments such as screen monitor driver
- Reliable Hardware (No FAN)
- Compact

- Easy to handle by EPICS
- Linux have many software tools

Test on Table



Main Specifications of F3UM02

item	Specification
Number of Channels	2 ch
Compatible Camera	Single Tap (8bit/pixel) Dual Tap (16bit/pixel) RGB Color (24bit/pixel)
Max. Connections	2 Color RGB Cameras (6 Monochrome Cameras)
ADC	100MHz
Resolution of Digitizer	8 bits
Camera Resolution	16K x 16K
Trigger	External / Internal (software)

Record Support

- New record type "graphics record"
  - Originally developed for other project (2003)
  - Just remove several unused functionality
- Raw Image (waveform)
- Reduced-size Image
- H/V Size information
- Background subtraction
- Image analysis
  - projection to horizontal/vertical direction
  - peak position, peak value, FWHM, etc
  - Possible to enable/disable these calculations

Device Support

- PCI bus between CPU and UM02 module
- devGrF3UM02.c

  dbd : device(graphics, INST_IO, devGrF3UM02, "F3UM02")

- Interface graphicsRecord to hardware

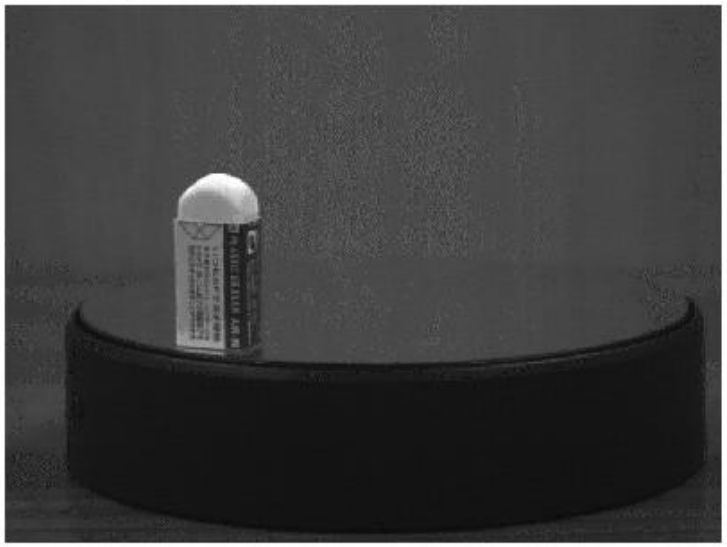
- issued by "I/O_interrupt" scan request
- just transfer raw image data to record

Display

- For test : python + PIL (Python Imaging Library)
- For Operation : EDM

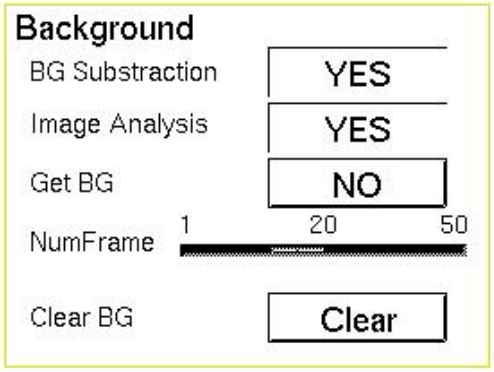
Example : BG Subtraction

Raw Image



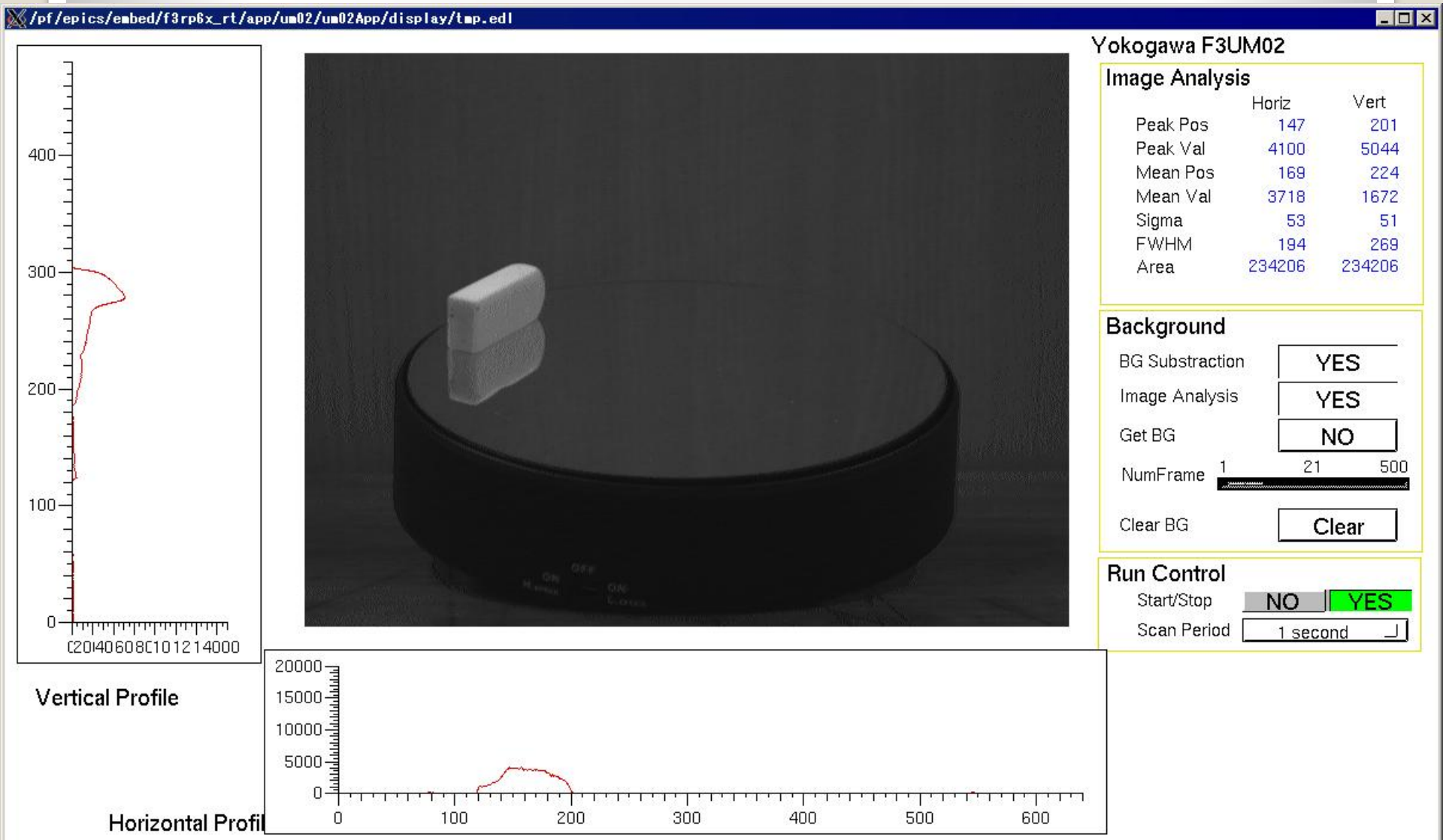
Background

Corrected Image (=Raw-BG)

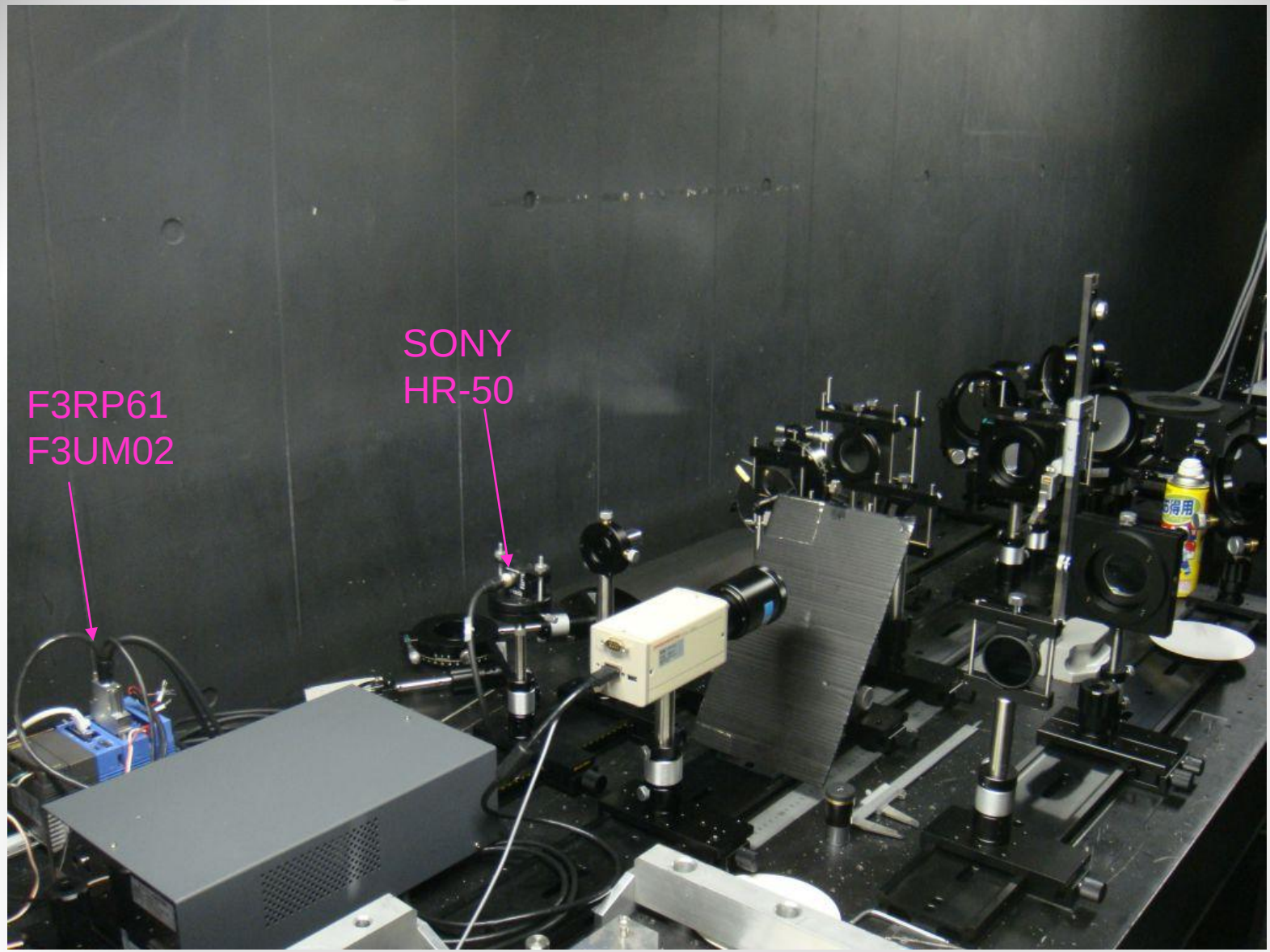


Example : Profile, peak, etc

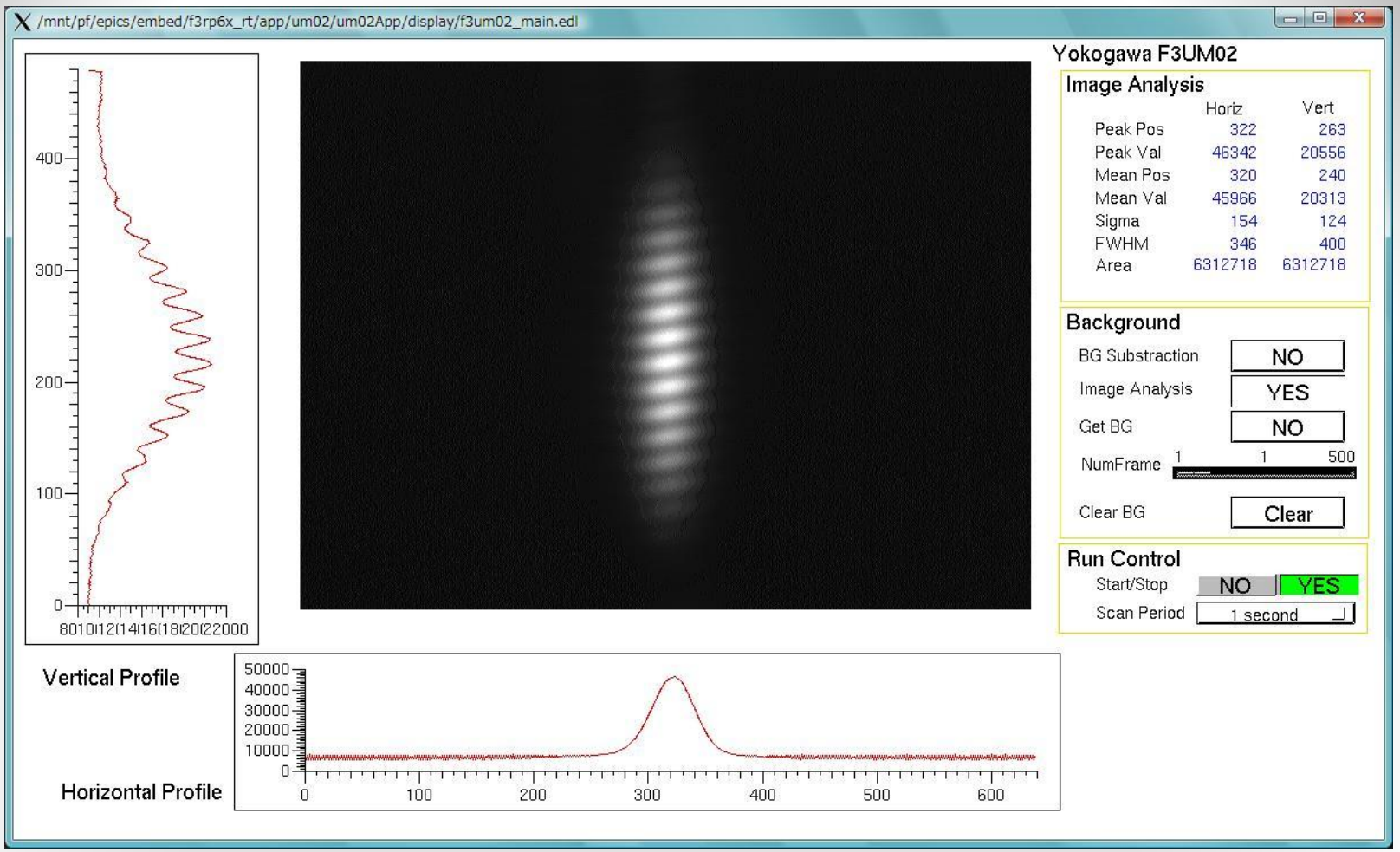
These parameters are calculated in the GraphicsRecord



Test on Optical Table



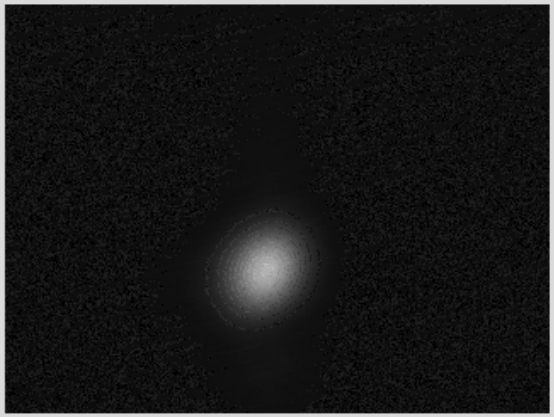
Example : Interferometer

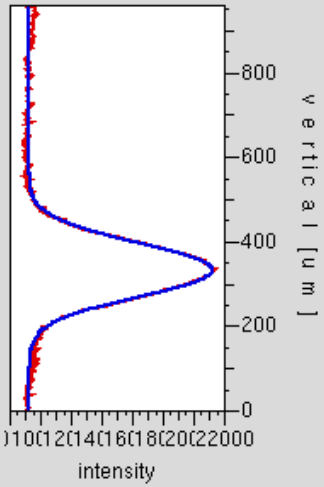


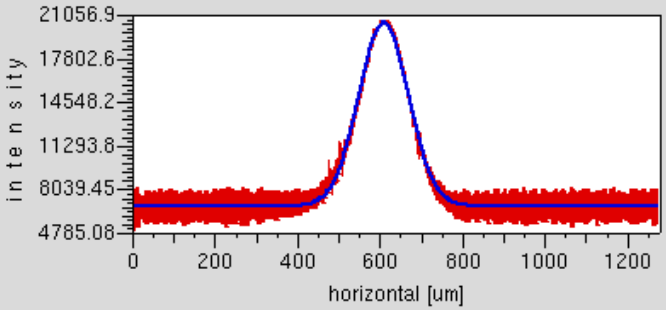
Due to deformation and dirty of the mirror and extraction window, interference fringe is not good as other beam line. We will replace the mirror and window in this summer.

/home/takai/profileApp/testApp/display/sample.edl

Beam Profile Monitor @ BL27







Fitting function:

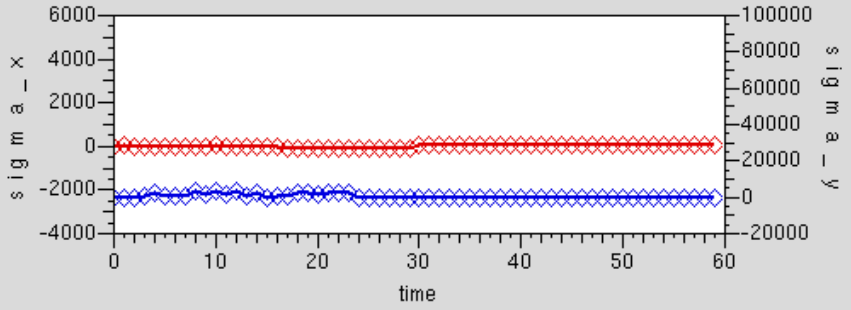
$$f(x) = a*exp(-(x-b)^2/(2*c^2))+d$$

Initial parameters:

a = 13000	b = 600	c = 60	d = 6000
a = 15000	b = 400	c = 40	d = 9000

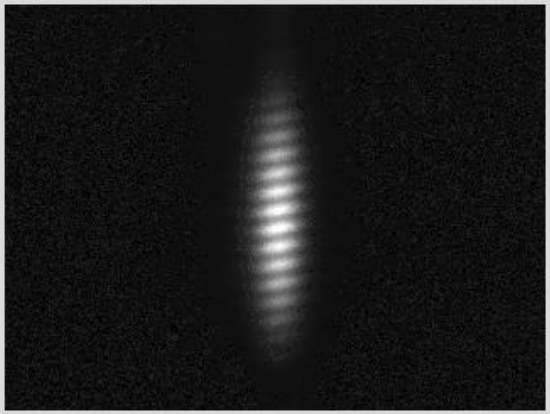
Fitting result [um]: sigma_x = 60.36

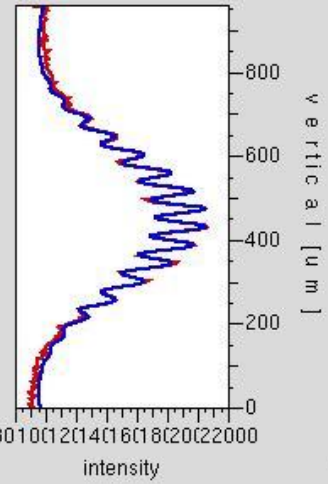
sigma_y = 62.39

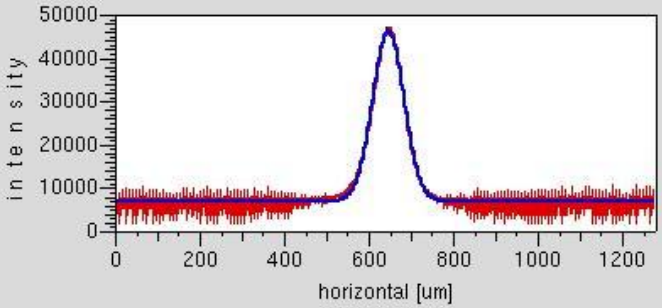


/home/takai/profileApp/testApp/display/sample_sinc.edl

Beam Size Monitor @ BL27







Fitting function:

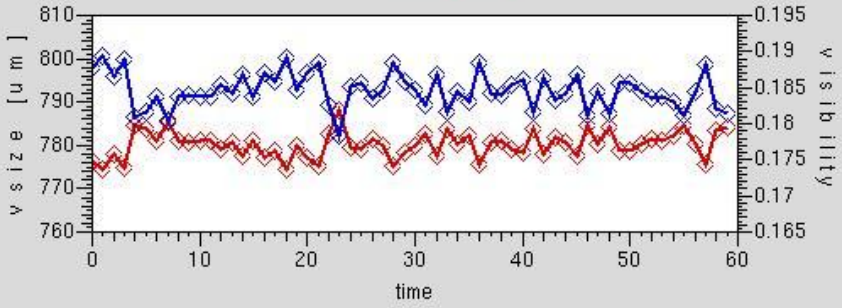
$f(x) = a*sinc(A(x-b))^2*(1+c*cos(B(x-b)))+d$

Initial parameters:

a = 13000	b = 600	c = 60	d = 6000
a = 9000	b = 453	c = 0.18	d = 9000
A = 0.008	B = 0.15		

Fitting result [um]: visibility = 0.18

vsize [um] = 784.3



As a first step, nonlinear curve fit is tested by gnuplot. If we use other (fast) PC or server machine, we can use matlab or other program.

Performance

- Free Run Mode : 15Hz update rate
- Trigger Mode
  - tested : 1Hz, 2Hz, 5Hz, 10Hz
- Linux RT version is not used yet

Performance : CPU Load (1)

1) NO Analysis, NO Channel Access

Repetition Period	CPU Load (Typ)	CPU Load (max)
1 sec	3.0 %	4.0 %
0.5 sec	3.7 %	7.3 %
0.2 sec	16.0 %	17.0 %
0.1 sec	31.0 %	32.6 %

2) Analysis Only

Repetition Period	CPU Load (Typ)	CPU Load (max)
1 sec	18.6 %	19.0 %
0.5 sec	37.0 %	37.3 %
0.2 sec	91.3 %	91.9 %
0.1 sec	N.A.	N.A.

CPU Load (2)

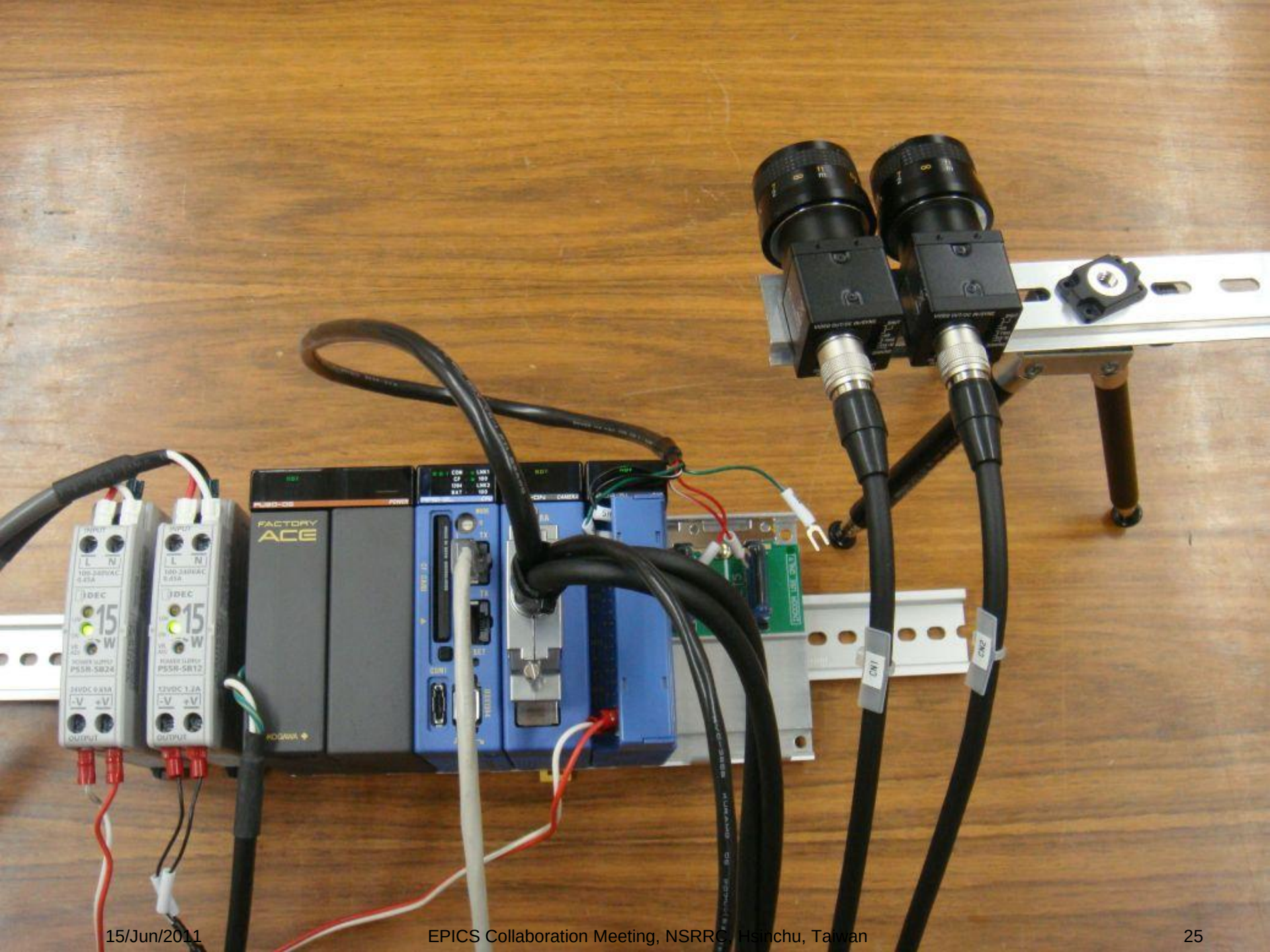
3) Channel Access Only. (NO Analysis)

Repetition Period	CPU Load (Typ)	CPU Load (max)
1 sec	6.7 %	7.3 %
0.5 sec	13.7 %	14.0 %
0.2 sec	34.0 %	35.0 %
0.1 sec	69.0 %	70.0 %

We plan to optimize ‘Analysis’ routine.

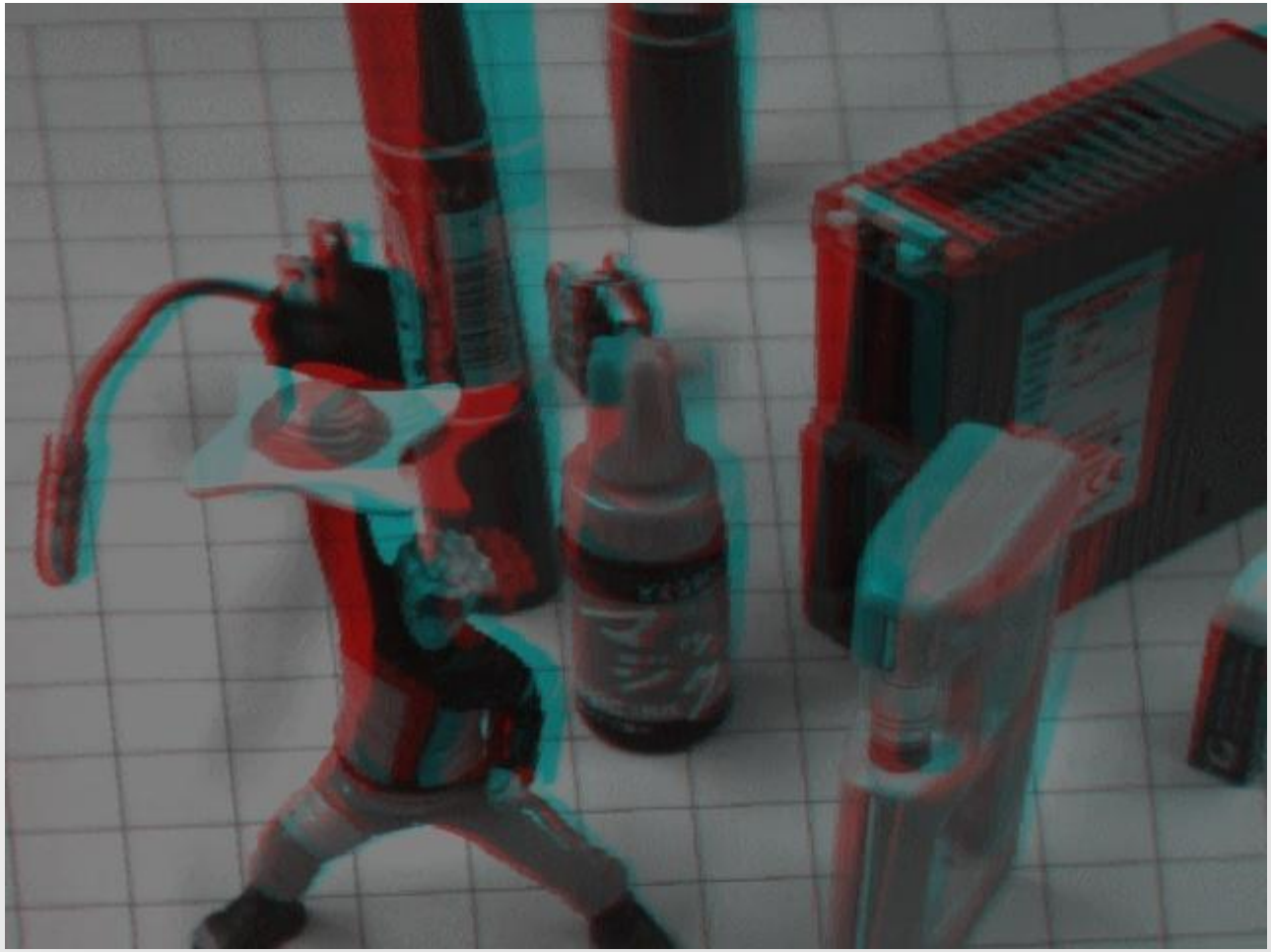
How to do with NTSC signal?

- In the beam transport line, we have been used about 10 CCD cameras with NTSC output, and commercial video switcher is used for many years.
- UM02 input signal : RGB color
- We must use NTSC – RGB converter
  - Confirmed to capture the image
  - Linearity is not evaluated yet


FACTORY
ACE
KOYANA
CN1
CN2

Just for fun

- 3D Movie with two camera; Phantogram



Conclusion

- We have developed image acquisition system on embedded platform
- New Record/Device support is developed
  - Basic analysis is performed inside the record
- Testing in PF-Ring, Linac

- What's Next?
  - Improvement of RT performance
  - Calibration of NTSC to RGB converter
  - Documentation
  - Asyn / AreaDetector / SynApps ?

References

- PCaPAC
  - http://accelconf.web.cern.ch/AccelConf/pcapac2010/papers/thpl018.pdf

"Channel Access Everywhere" Policy

Adopt: Keep It Simple and Stupid
Avoid : Reinventing the wheel