BL4S

Markus Joos, Christoph Rembser, Jorgen Petersen, Candan Dozen, Timothy Brooks, Cenk Yildiz, Lau Gatignon

Outline

1. Introduction/Goal
2. Preparation for T9 beamline
   a. Modify webcam
   b. Calibration
   c. Build a set-up
3. T9 BeamLine experimental Set-up
4. Data Collection
5. Data Analysis

Intoroduction/Goal

- The main idea is to see the track of the high energy particles with a webcam sensor in T9 beamline
  - Find out how exactly a webcam sensor will respond to high energy particles

Preparation for T9 beamline

- Tape the lens to keep the ambient light outside
- We have developed a program to save captured beams images from webcam
- Find a detector that we can use to compare images captured by webcam
- Build a set-up to hold webcam and professional detector.

T9 Beam Line

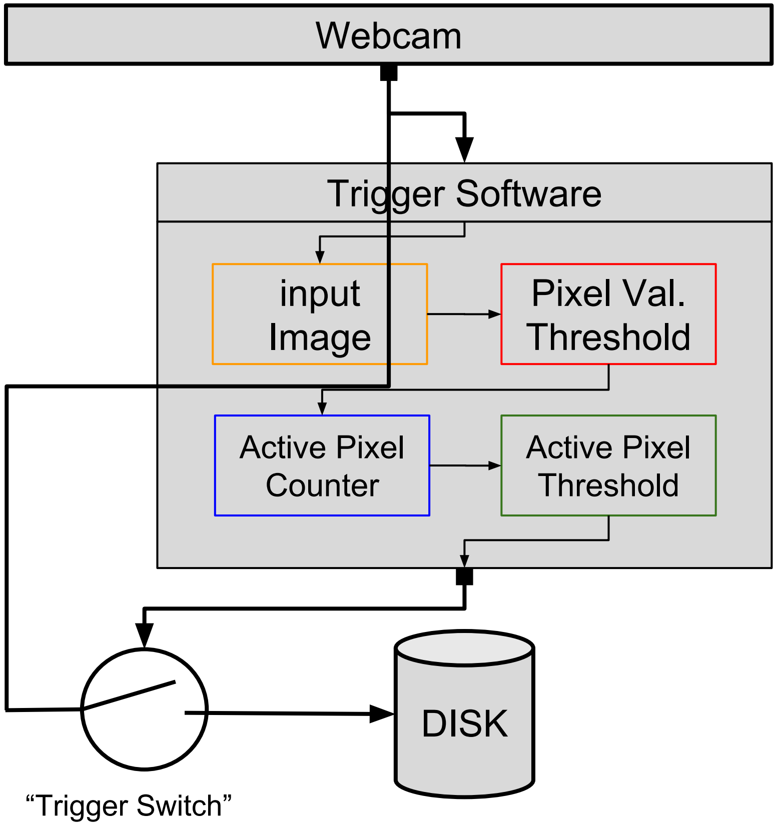
- T9 beam is one of the beam lines of the experimental zone “East Area”
- A 24 GeV proton beam from the PS accelerator (Proton Synchrotron) hits a target .
- Pions, protons, kaons electrons, positrons and their antiparticles are created with different intensities.

Preparation for T9 beamline

- Beam at T9 is not constant. We get spills of particle
    - A spill has a duration of 400 ms
    - We get two spills per minute
- That means we need to design a system which starts capturing the images when the beam is on
    - Starting the image recording takes time
        - By the time the camera records images the spill is over
    - We have to record images permanently and look for frames with beam effects offline
- It is also not good to capture and store all images; waste of resources (storage capacity) and it is hard to analyse unnecessary/unuseable (dark images)
    - We have developed sw that only saves an image when a minimum number of pixels of the webcams cmos is active
    - This is what physicists call “Trigger”
- Hey! You have just learned the basic idea behind **trigger system** which is used for the big detectors at CERN!

Controlling The Webcam: Data Selection

- Capturing the image and feeding the Trigger software which is written in C++ by using OpenCV (Open Computer Vision)
- Trigger consists of basic image processing modules: applying pixel value threshold (filter by pixel value), active pixel counter and threshold for the number of active pixels
- At the end, we get a logic value, “true” or “false”
    - true: image will be recorded
    - false: image will not be recorded
    - We have to make sure that we are not discarding interesting data (this is known as trigger efficiency)



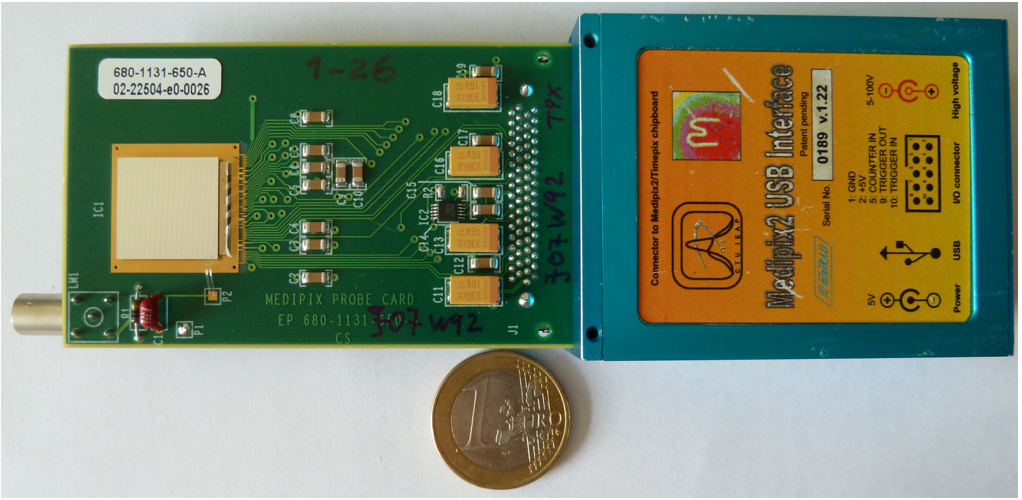
Finding a Professional detector

- Remember the first slide, we said we need to compare images captured by the webcam with a professional detector
- Umm, maybe CMS can help us!? --Not really... We cannot move the webcam into the CMS detector right?

http://home.web.cern.ch/sites/home.web.cern.ch/files/image/inline-images/old/cms_visit.jpg

Finding a Professional detector: TimePix

Timepix is a silicon based particle detector that was designed at CERN for various applications such as detection of cosmic particles(there is one on the ISS) or x-rays in medical imaging systems



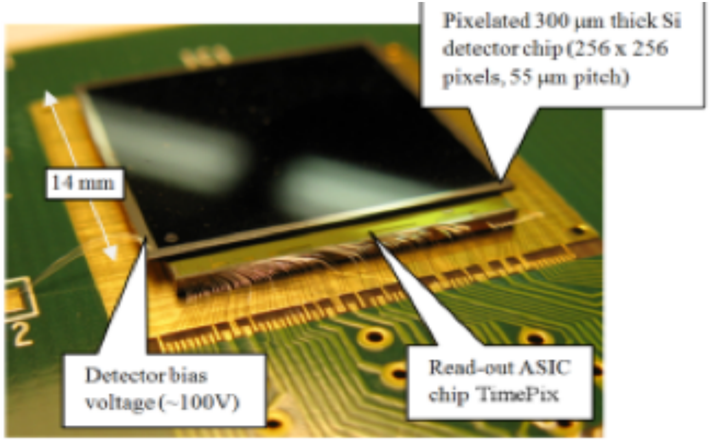
http://www.cobra-experiment.org/sites/site_cobra-experiment/content/e97949/e97964/e97967/e99482/P1000212_crop.JPG

Finding a Professional detector: TimePix

- It is really small/portable
  - We can move it easily, for example, we can put it front of a webcam (Eureka!)
- Easy to use
  - It has USB interface, user friendly software and well documentation
- All we need is just to connect it to a standard personal computer and it is ready for data taking!

Finding a Professional detector: Timepix

- Silicon Hybrid pixel Detector
- Designed using a commercial 0.25Mm technology with a cell of 55x55 Mm2
- It has matrix formed by **256x256** pixels
- Each pixels can be independently operated in one of 3 modes:
  - **Counting of the detected particles(Medipix)**
  - Measurement of the particle energy(TOT)
  - Measurement of the time of particle interaction (Timepix)



Pixel detector Timepix. Device consists of two chips connected by bomp-bonding technique. The upper chip is pixelated semiconductor detector (ussualy Silicon). The bottom chip is ASIC read-out containing matrix of 256 x 256 of preamplifiers comparators and counters.

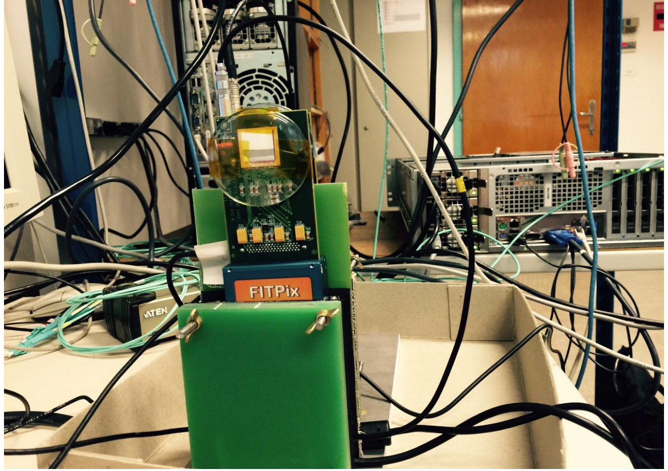
TimePix vs Webcam

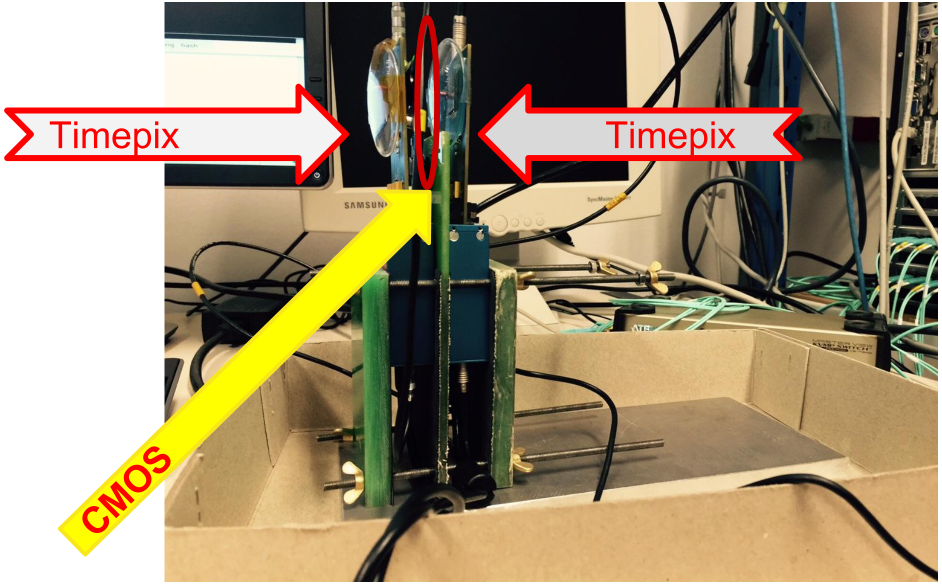
CMOS (Complementary Metal-oxide Semiconductor): one of the main technologies for the image sensor. CMOS sensors have a fast readout , lower power consumption, higher noise immunity and a small system size.

	Webcam	TimePix
Sensor	CMOS	CMOS
Resolution	1280 x 720	256 x 256
Number of Channels	3, RGB (Bayer filter)	1 Channel
Main Difference	Provides a value depending on the light energy	Counts detected particles/Measurement energy/time of particle interaction
Radiation Tolerance	unknown(May be the beam will destroy it)	Radiation hard

Build a Set-Up

- The webcam placed tightly in between two timepix
- Incident particle impinges the first timepix, leave hits on the pixel and going through from webcam and then reach the second timepix





T9 : Experimental Set-up

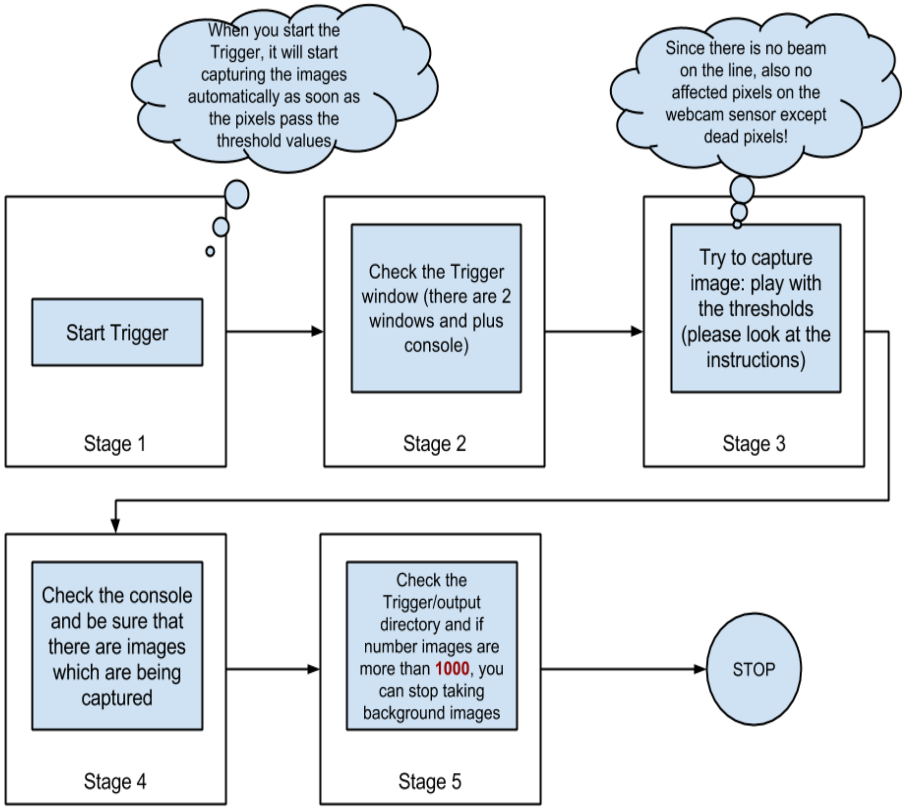
- The sandwich structure is located on a DESY table behind MNP17 magnet
- DESY Table is a kind of movable table which can be controlled remotely.
- This table allows us to move the sandwich in 2 dimensions in order to find the deflected beam and to move the webcam slowly into the beam spot
- <u>To do this</u>: *This will be done by the team of shift. One has to control the DESY table while other is checking the captured images in the control room.*

Background Measurement in T9

- Acquire a few images when the beam off

Stage 1:

- Trigger : Use the program that has been developed to control the webcam



Background Measurements in T9

- From one of the PCs in the control room login as “daquser” to pcbl4sleo4g : “ssh -Y daquser@pcbl4sleo4g.cern.ch”
    - *password* : BeamLine15
- Go to :

  cd /afs/cern.ch/user/d/daquser/public/webcam/Trigger

- Execute “Trigger.cpp” program

  ./run.sh

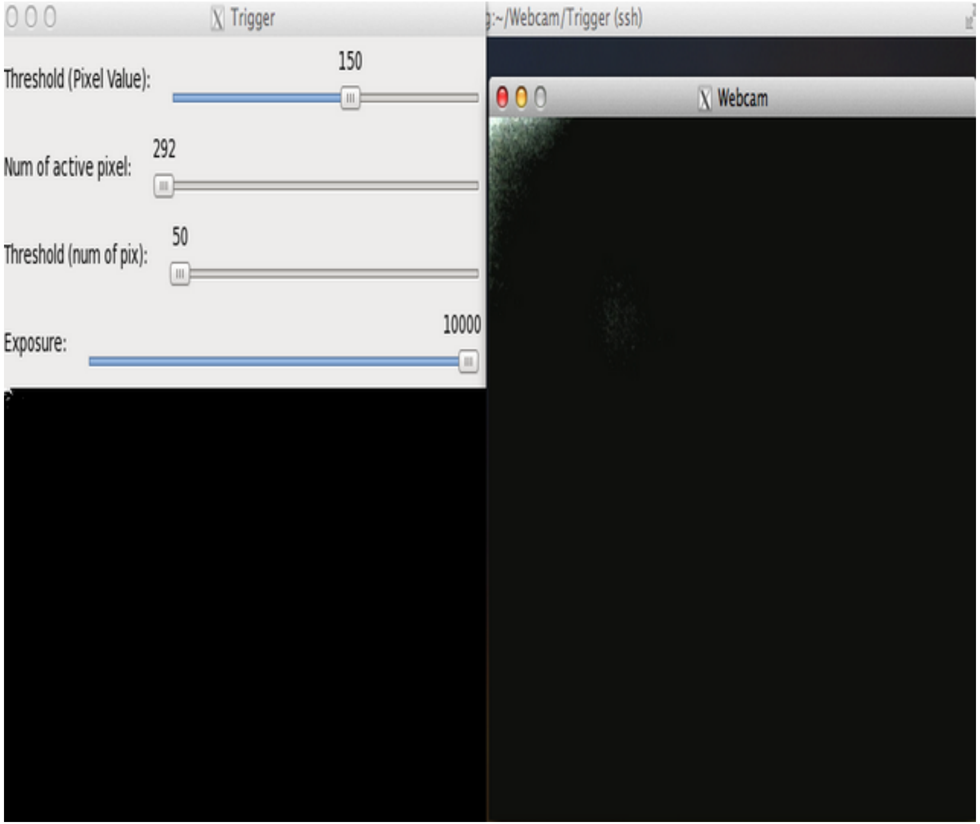
Background Measurement in T9

- **Stage 2:**

Window1: Trigger parameters

Window2: Webcam image

The propose of W1 is to control the settings of the image processing software



Background Measurement in T9

- **Stage 3:**
  - You can play with the Threshold settings by moving the sliders on Window1 or from Trigger.cpp program
- **Stage 4:**
  - After adjusting threshold value , you will see the message on the console “image saved”
- **Stage 5:**
  - All captured image will be saved in a “***output***” ***directory***
    - *You can change the directory name as you wish in Trigger.cpp program*

*sprintf(fileName, "**output**/frame_%d.tif", frameNumber);*

  - ****Take into account that*** *after saving frames use “stat ” command to have the captured time information from webcam. It will be very useful when we will compare the results*
    - **example : in the directory you saved the output files write* *stat * > output_stat.txt*

T9: Data Taking

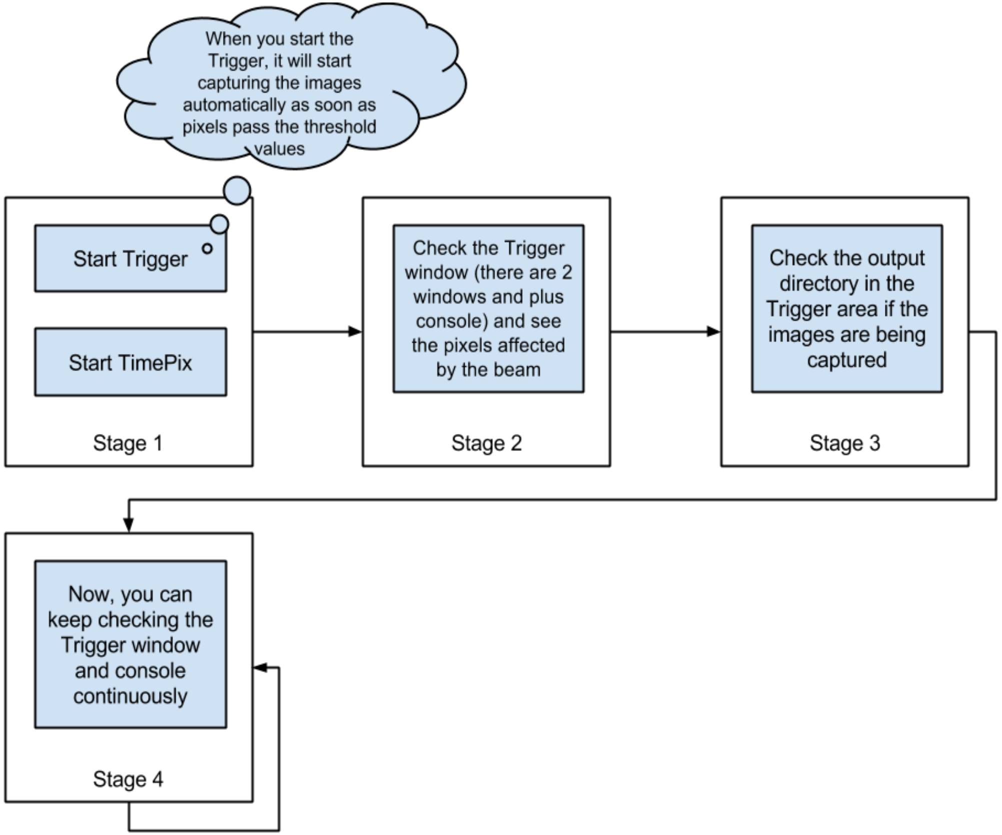
- Define a good position to find the deflected beam with using DESY table.
- We will use Timepix sensor in order to find the beam
- We don’t want to save anything just yet!

T9 : Data Taking

Once we have found the beam your detector will be ready for data taking!

Stage 1:

- First Start Trigger and follow the steps as you already did before
- Then Start Timepix



T9: Data Taking

- **<u>Stage 2:</u>**
  - The webcam sensor could be damaged by the beam
    - Therefore we will moved the webcam slowly into beam
    - As soon as we have seen the beam we will moved the webcam to a safe position and repeat the Background Measurement in order to check for any damage
  - It is normal that exposure to beam damages pixels in a CMOS. As long as the number of dead pixels low and their location known one can tolerate the effect. Our pixel detectors therefore use redundancy
  - If the CMOS is completely dead after the first exposure to beam we have to discuss plan B

T9 : Data Taking

- **Stage 3:**
  - Adjust the threshold value ~150 , check the console until you make sure the captured images are saved
- **Stage 4:**
- All captured image will be saved in a “***output***” *directory*
  - ****Take into account that*** *after saving frames use “stat ” command to have the captured time information from webcam. It will be very useful when we will compare the results from timepix*
  - ***example : in the directory you saved the output files write stat * > output_stat.txt*

T9: Timepix

- To operate the Timepix
  - Open a new terminal and go to “pixelman” directory

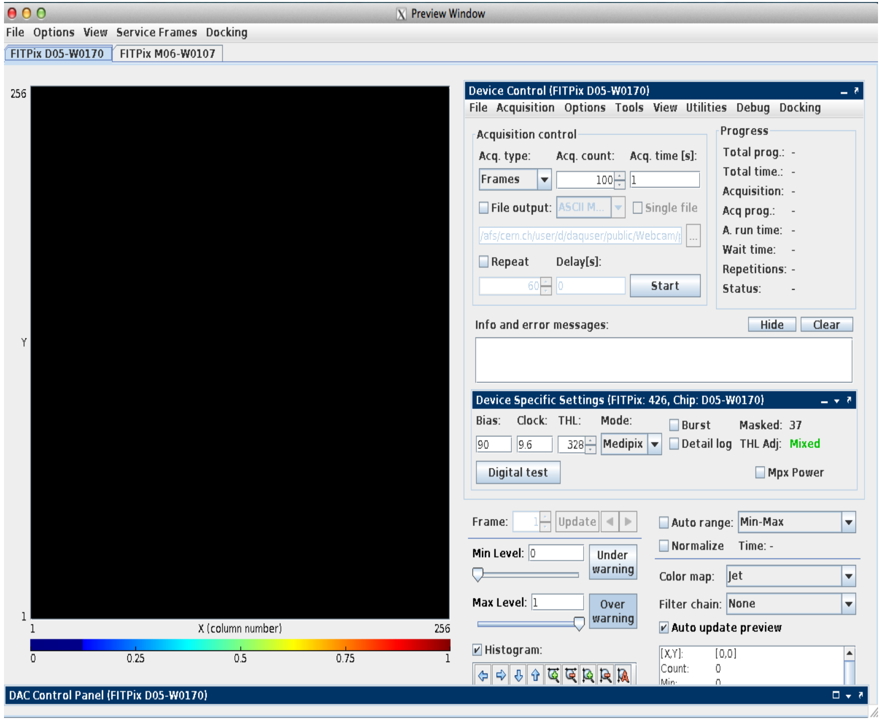
cd /afs/cern.ch/user/d/daquser/public/Webcam/pixelman/Pixelman_2013_09_25_x64

- Execute pixelman s/w

./pixelman.sh

T9: TimePix

- After execute the pixelman s/w The Timepix Control Panel will appear
    - All settings already saved !
    - Just press “Start” button!
    - It will start data taking and save all frames in a “output” directory
        - For detailed information to operate Timepix look into BACK-UP slides!



Ideas for other measurements

- **Change the direction of the structure**
  - Rotate the CMOS sensor by 90° along the vertical axis
  - Repeat all measurement again for the rotated CMOS sensor
  - **Can we see tracks of particles?**
- Put some metallic plates in front of the CCD
  - **Investigate the effect**
- Your turn!...

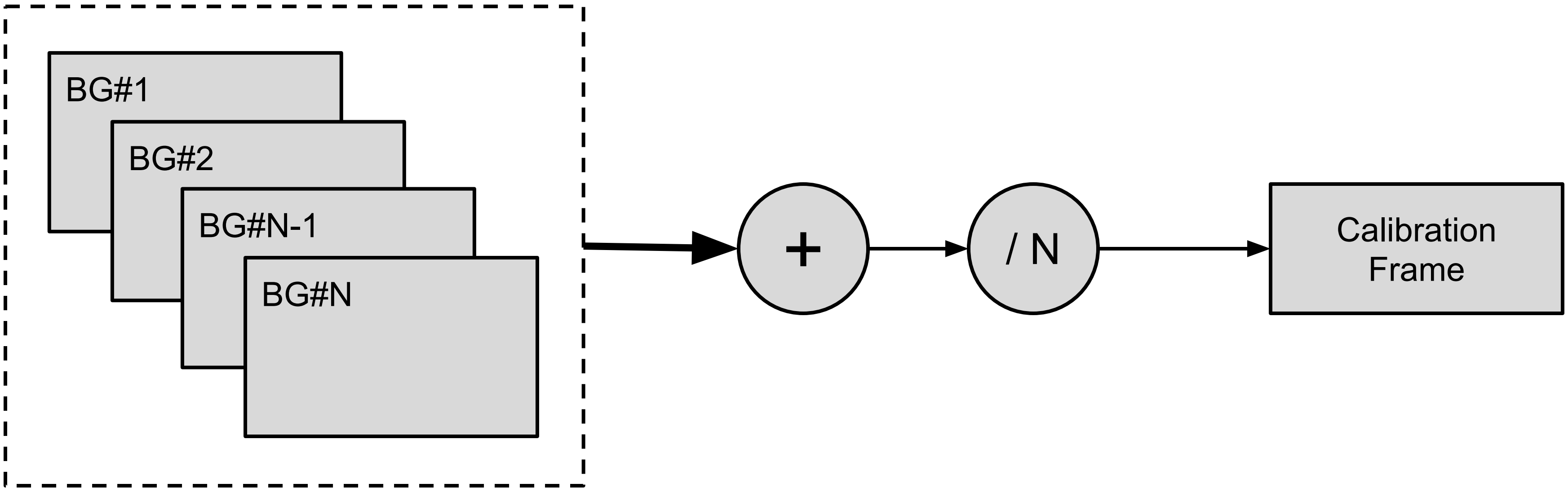
Image Analysis : Offline Analysis group's Task

- Now we have hundreds of Calibration and Beam frames stored in output directories and with names chosen by the shift team
- Time to Analyse the stored images!

Analysis: Background Measurement

- Identify:
  - Pixels that are never black
  - Pixels that are sometimes not black
  - Save the coordinates of these pixels in a calibration file.

Analysis: Background Measurement



Analysis: Background Measurement

- we have developed a program “Background_creator.cpp” to create a calibration frame:

  cd public/Webcam/Image_Analyis_withOpenCV/opencv/Analysis/BackgroundCreator

- To Execute “Background_createor.cpp” program

  ./run.sh

- it will produce a file (*.tiff extension) that you should save

Analysis: Background Subtraction

- We have developed a program “Subtract.cpp” to subtract the Calibration Frame from each Beam Frame (the output from data taking operations)
  cd /afs/cern.ch/user/d/daquser/public/Webcam/Image_Analyis_withOpenCV/opencv/Analysis
- To Execute “Subtract.cpp” program

  ./run.sh

- The program takes the calibration frame and subtracts it from each beam frame to produce new signal frames.

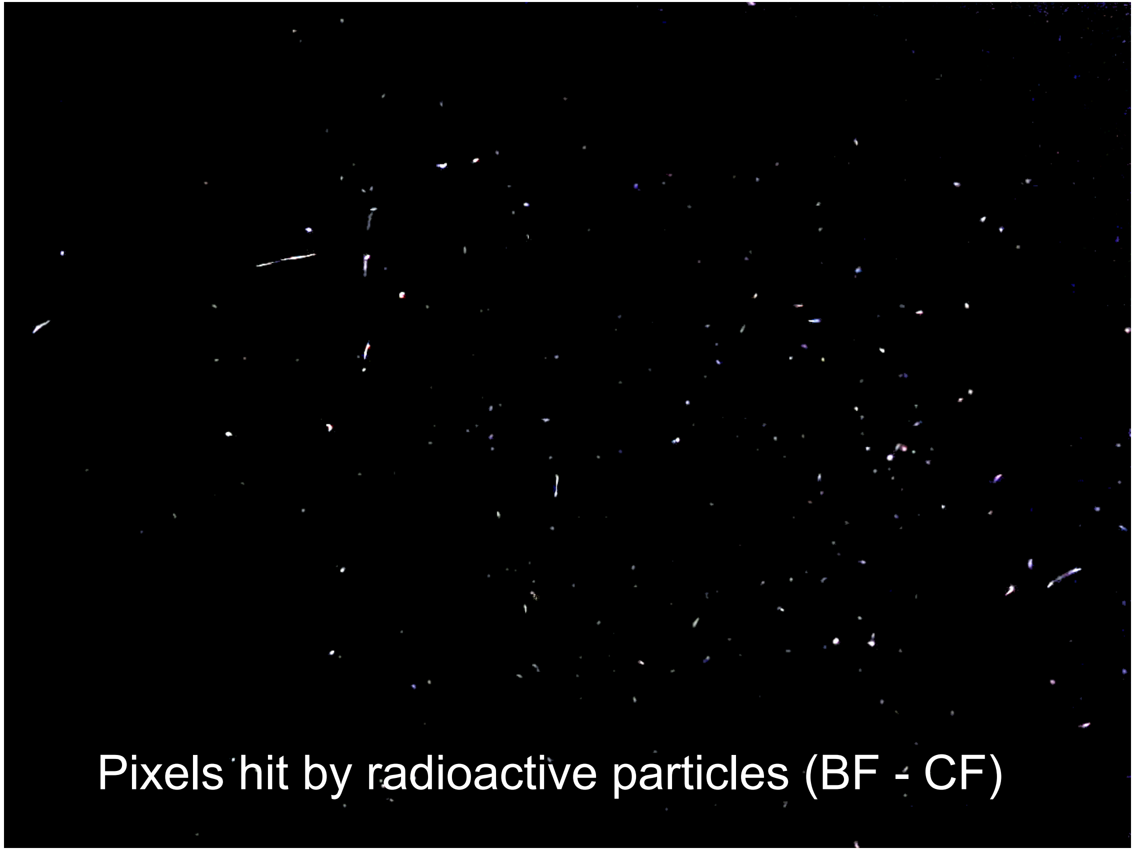
- Save the signal frames in a new directory

Demonstration

- *We have not yet exposed the CMOS to beam*
- *In order to test the background reduction we have used Strontium 90 Radioactive Source*

Note the blue pixel on the top right hand corner

The white lines and dots are most likely traces of beta particles(electrons)


Pixels hit by radioactive particles (BF - CF)

Analysis: Compare Results

- Compare the images which is taken from both webcam and timepix

Thank You!

BACK-UP

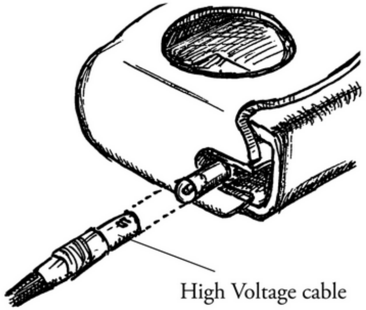
Preparing TimePix for operation

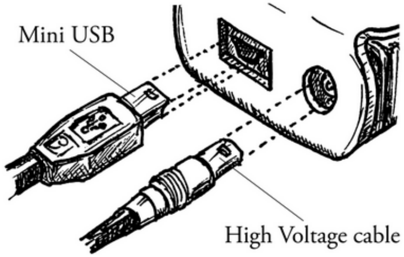
Timepix: Connecting the detector

- The Timepix detector needs to be connected a computer to operate
- Any computer with USB port and ability to run the Pixelman S/W
- Before making an attempt to connect the detector please ensure that
    - Installed Medipix device drivers
    - Installed Pixelman S/W suite
    - Closed any unnecessary programmes that you might have had running on your computer
- If you’ve done all that,you’re just about ready to go!

Timepix: Connect High Voltage Cable and Mini-USB

- High Voltage Cable provides the bias voltage to the silicon sensor.
- Connect the cable to each end of the detector
- Connect USB end of the cable to a convenient USB port on the computer and connect mini-USB end of the cable to the USB adapter on the detector



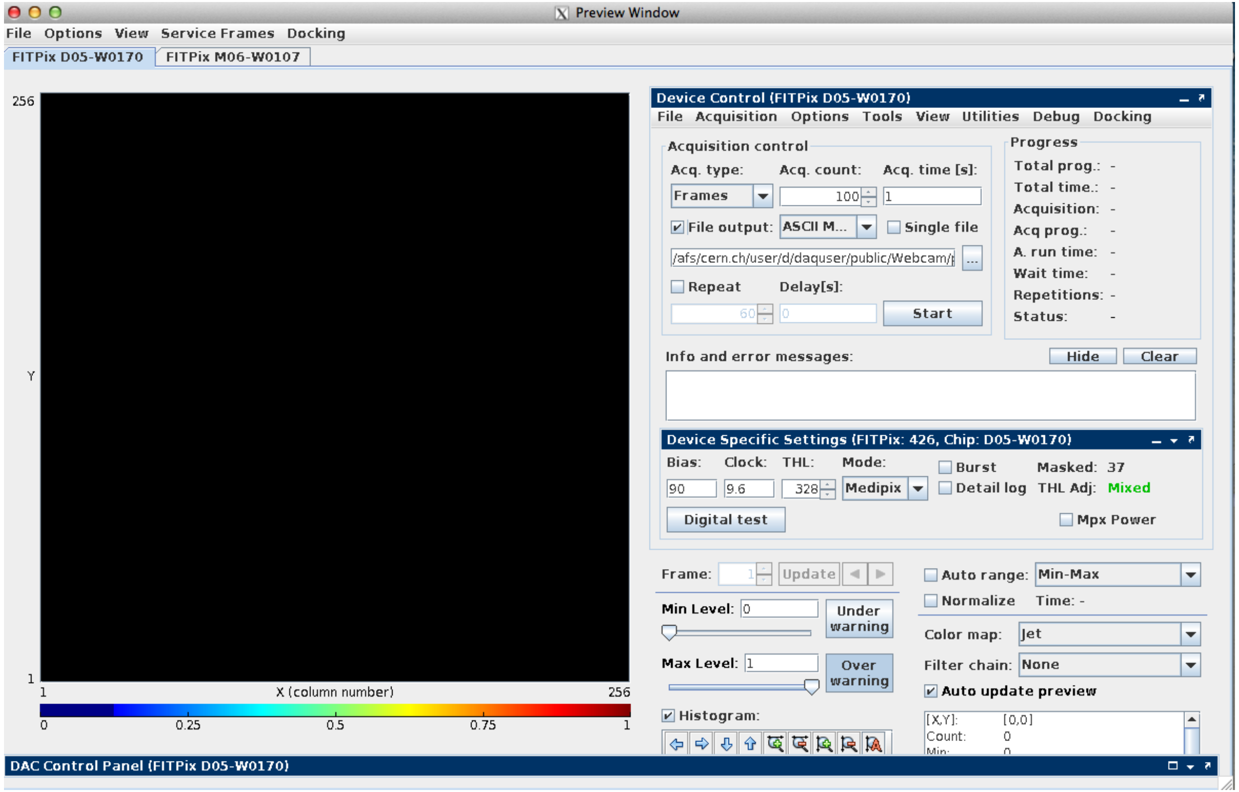


Timepix: Preparing the detector for operation

Prerequisites:

- Once you have connected the detector, it should be placed in its "default" position (for whatever experiment you're going to be doing)
- Remove any source of radiation to avoid affecting the configuration process
- Now, You can start the Pixelman software suite

Timepix: Preparing the detector for operation



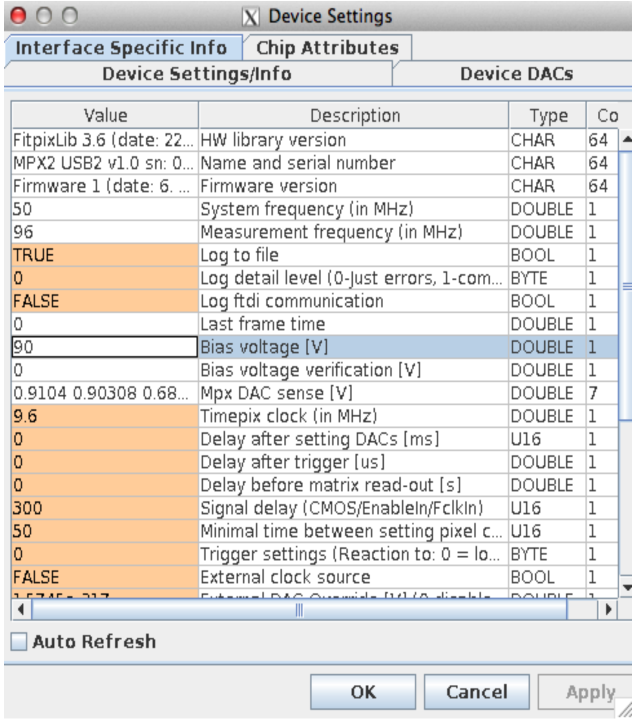
-
-

Timepix: Configuring the detector

- Reset the pixel Configuration by clicking on the “File”menu in the Device Control panel
- Select “Reset Pixels Cfg” ↠ “All Bits”
  - This will clear any previously stored settings for all of the pixels
- To check this by clicking on the “Service Frames” menu in the Preview Window and selecting any of the options except for “None”
  - All pixels should have the same value (example: the “Mask Bits” frame should only contain values of “1”, no pixels have been masked)

Timepix: Set the bias voltage to 90 or 94.5 V and Change the "Ikrum DAC" setting

- Click on the "Options" menu in the Device Control panel and click on "Device Settings..." This will open Device Settings dialog box
  - Select the "Interface Specific Info" panel .
  - Double click on the orange box to the left of the "Bias Voltage" and type the value
- Select the "Device DACs" tab , double clicking on the text box to the right of the "IKrum" and type "1"



Timepix: Performing the Threshold Equalisation

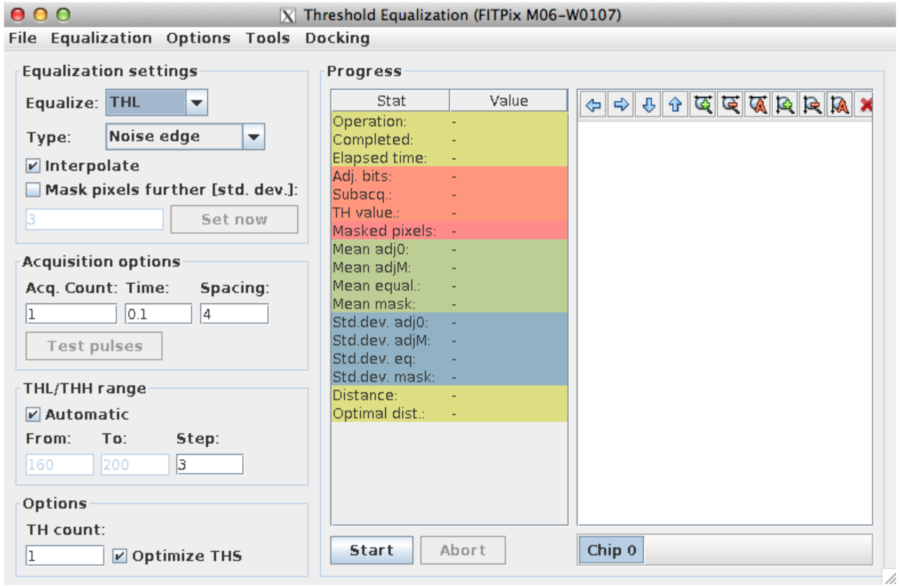
- Each pixel of the Timepix detector has a threshold associated with it.
- This is a number that corresponds to an amount of charge seen by the pixel that is "interesting"
    - "Interesting" charge is produced by ionising particles hitting the Timepix sensor
    - "Unintersting" charge comes from electronic noise
- By setting the threshold for each pixel, one can eliminate "unintersting" signals

Timepix: Performing the Threshold Equalisation

- The Threshold Equalisation procedure scans the pixels,
- Calculates a mean threshold value
- Stores a values for each pixel corresponding to the threshold adjustment needed for that pixel relative to the mean threshold value
- To operate it:
- Click "Tools" menu in the Device Control panel in the Preview Window
- Select "Threshold Equalisation" menu item

Timepix: Performing the Threshold Equalisation

- Check that all the values in the text boxes look the same as those in the Figure
- Press “Start”
- The procedure is running for a while
- Note that “THL” DAC setting should have updated to the nearest integer value of The threshold Mean Value



Timepix: The detector operation

- Just few thing left before you are completely ready to go
- Ensure That the “Tpx Clock” value in the Preview Window is set to “10” (Mhz)
- Set the “THL” value in the Preview window at a value roughly 40 below the mean value calculated by the Threshold Equalisation procedure
  - Example: if the THL value is “450” , set it at “410”
- The value is lowered, which actually raises the threshold for every pixel. That’s why the detector is run in “positive polarity”mode.(Device Settings/Info)

Timepix: The detector operation

- Put the Detector into “Medipix” mode
- Click “Mode” in the “Device Specific Settings” and choose “Medipix”
- Configure the frame display :
- Deselect the “Auto range”
- Set the “Min Level” to “0” , “Max Level” to about “1” and the colormap “Jet”
- Configure the Acquisition control in the “Device Control” panel
- “Acq.type” to “Frame” , “Acq. count” to a large number (~1000) , “Acq. time[s]” to “1”
- Select the “File output” check box and create a directory by clicking “...” box to save the output files

Timepix: The detector operation

- Now You Are Ready to operate the Timepix Detector
- Just press “Start”!

Timepix: Storing detector settings

- The Pixel Configuration:
  - open the “File“ in the Device Control panel
  - Select “Save Binary Pixels Cfg”
  - Save the .bpc file with a name of your choosing in a suitable location
- The whole detector configuration:
  - File» Save Config..» save the .mcf file with a name of your choosing in a suitable location

Saving detector configuration info along with your data allow you to know what state your detector was in when you took your data. Without knowing that the data is useless, SO follow these steps carefully!

Timepix: Detector settings

- Acq.type: “Frame”⤖ the data recorded in each time window is recorded separately.
- “Integral”⤖the data recorded in each time window is superimposed on all of the previous frames
- Acq. count: The number of frames to record
- Acq.time[s]: The length of time the detector records data per frame (the time window for data taking for each frame)
- File output: when ticked, your data will be saved to a file or set of files

Timepix: Detector settings

- Color map: Determines how the values are mapped to different colours.
  - Gray is the simplest- 0 is black , and the highest value is white, and everything in between is a shade of gray
  - Auto range (tick box): When ticked, The pixelman s/w automatically scales the brightest colour to the highest value recorded by a pixel, and the darkest colour to the lowest value (0)
  - If you are comparing different frames untick "Auto range" : Manually set the min and max value of the colour map using the "Min Level" and "Max Level" text boxes.

A Break: TimePix Test

- Sample images from Timepix detector with Strontium radioactive source
- Color map: Determines how the values are mapped to different colours.
  - brightest colour to the highest value recorded by a pixel, and the darkest colour to the lowest value

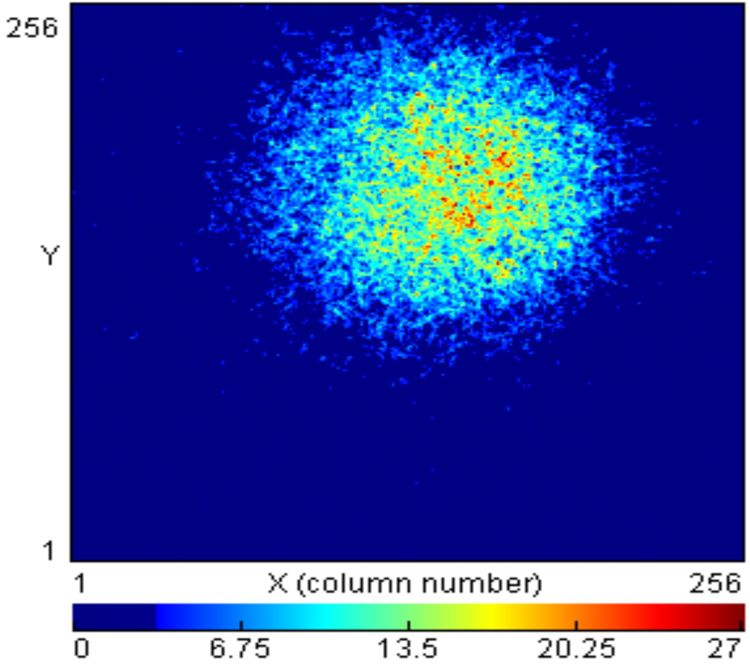


Image taken from Timepix with Beta Source

A Break: Webcam Test

- An image captured by the webcam
- A radioactive source was placed in front of the webcam

END…