Report from Zeuthen

Video

PITZ report and plans

Future plans

TINE Users Meeting
2020-10

Stefan Weisse

Video
Simplified SGP[].xml and Raw2Jpeg[].xml

```
<CIniExWrap>
 <!-- This is only a documentation excerpt / example! -->

 <ServerSettings>
  <Name>SRINT2.SGP</Name>
 </ServerSettings>

 <TineServerSettings>
  <Context>PETRA</Context>
  <ServerName>SRINT2.SGP</ServerName>
  <EquipmentModuleName>SGPJA1</EquipmentModuleName>
  <PortOffset>1</PortOffset>

  <FecName>ACC10PESRV27.10</FecName>
  <FecDescription>SGP Jai SDK v3 (GigE, ...) SSC</FecDescription>
  <FecLocation>n/a</FecLocation>
  <FecHardware>n/a</FecHardware>
  <FecResponsible>Stefan Weisse</FecResponsible>
  <Subsystem>Video</Subsystem>

 </TineServerSettings>

</CIniExWrap>
```

SGP[SRINT2.SGP].xml

- optional from 2020-09
- <TineServerSetting><ServerName> →

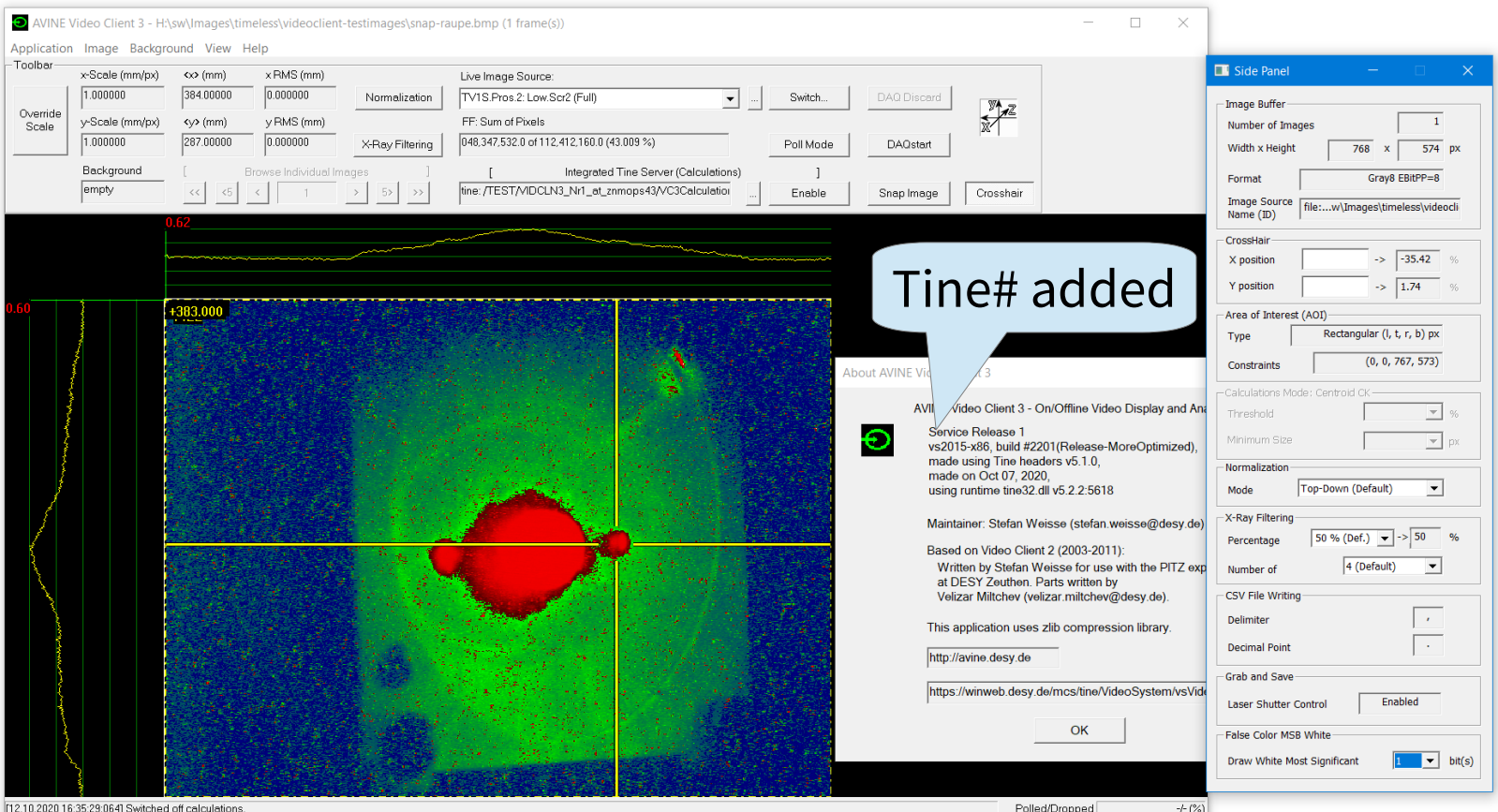
  <ServerSettings><Name>

- FecName: host name + ‘.’ + PortOffset
- FecDescription, FecLocation, FecHardware, FecResponsible, Subsystem: default values shown left, hardcoded in server executable

Video
SGP_Ebus

- SGP_Ebus: front-end video server to interface GigE Vision cameras via Pleora eBUS SDK v6 www.pleora.com
- for cameras that have been controlled via JAI SDK (discontinued)
  - PITZ, Petra (Diagnostic Beamline, E-Weg, SRINT2), Regae
  - Cameras by JAI, Basler, … (**NOT Allied Vision Tech. / Prosilica**)
- runtime paid license needed (MAC-address, USB-Dongle), except JAI CMOS cameras
  - license cost: currently one-time paid (per MAC-address 200 US $, per USB-Dongle 400 US $)
- development finished, to be tested in production at PITZ end of this year, foreseen to be introduced to Hamburg in 2021
- in general not happy with price vs. robustness and difficulty of use (no updates, login needed), but no better option known

Video
Video Client 3: Crosshair (user-defined)



- Crosshair (user-defined)
- to outline target position
- useful for align-ment
- requested by Mech-ZN and HH e-welding
- not released yet (testing…)

Zeuthen PITZ report and plans

- in general runs stable
- mostly Release 5 now (Release 4 still in operation on Java: Universal Slow Control (USC) server and some clients, Doocs servers / JDDD)
- TINE central servers (ENS, GENS, TIME server, Globals server) migrated to new host running Scientific Linux 7 (good for next couple of years)

Zeuthen PITZ report and plans

- Java
  - no Oracle Java any more
  - OpenJDK 11 (installed via DSM) is used
  - JDDD brings its own OpenJDK (v14.0.1 currently)
  - lack of developer resources

- Jaka Webstart (JAWS) phaseout planned
  - future: local installations of applications, synchronized from network repository (for Windows an intermediate solution by Stefan is provided)
  - DSM?

Zeuthen PITZ report and plans

- Windows: slow phaseout of Jaka Watchdog planned
  - to be replaced by Doocs Watchdog (Davit Kalantaryan)

- Archive System: March, PMarch, Archive Server
  - old installation on old host
  - no one has time to maintain it (…some volunteer?)
  - plan is to remove it

Future work
SGP_Vimba

- SGP_Vimba: front-end video server to interface GigE Vision cameras via Allied Vision Technologies (AVT) Vimba SDK www.alliedvision.com
- for cameras that have been controlled via Prosilica PvAPI (discontinued)
- PITZ, FS-BT (Petra User Beamlines), Salome, EMBL Hamburg
- cameras: AVT Prosilica
- free of charge
- development not started yet, foreseen to be ready in 2021

Potential Future work
Matlab support in Hamburg on Windows

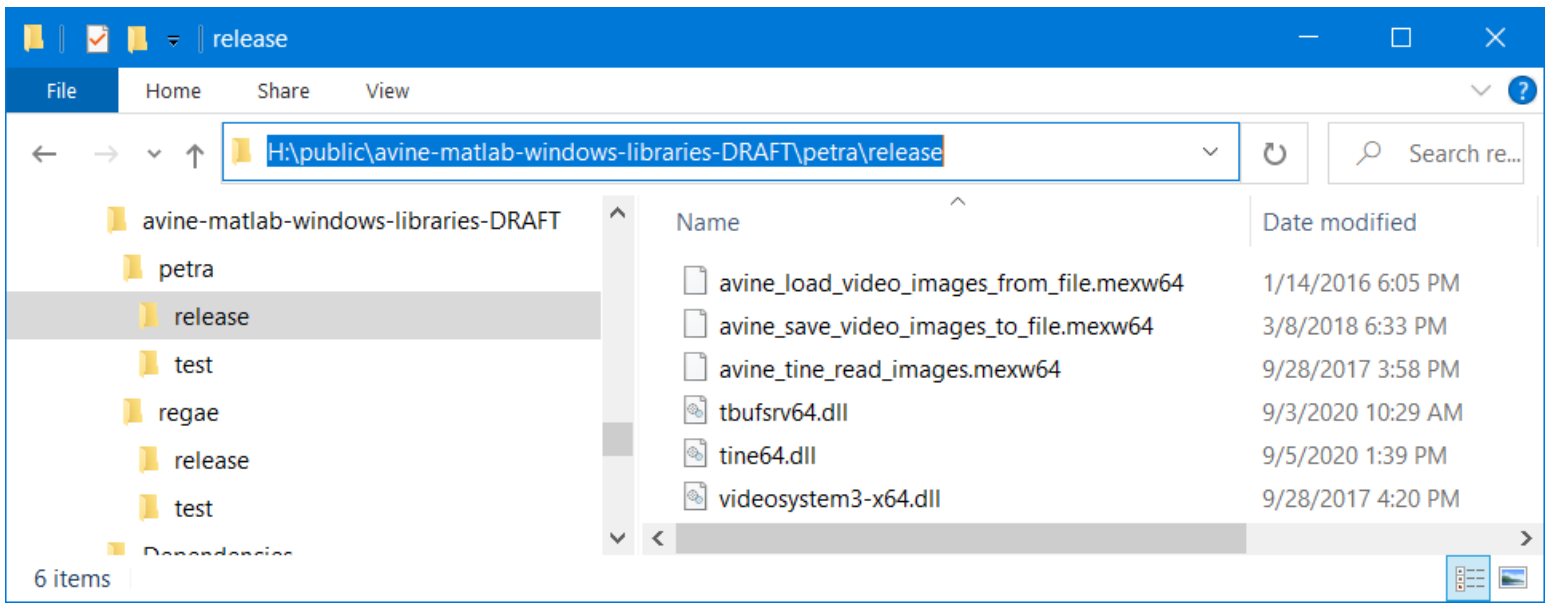
- mcamatlab server (Terminal server, users working in parallel)
  - L:\system64 (in Path environment variable)
    - contains e.g. tine64.dll, tbufsrv64.dll, videosystem3-x64.dll
    - tine64.dll v4.5.9 5244 (file date year 2016: isn't it a little bit old?)
    - users are able to change, e.g. files and folder names
  - L:\MATLAB\MiddleLayer\Release\links\tine (in Matlab search path)
    - contains mexw64 files (Matlab external function) and .m files (Matlab script)
    - contains outdated versions of avine_tine_read_images.mexw64 and avine_save_video_images_to_file.mexw64
  - L:\MATLAB\StefanWeisse\
    - for Avine Video, a good set of mex files (and libraries beneath) have been provided (but user needs to know this and adjust Matlab search path)

Potential Future work
Matlab support in Hamburg on Windows

- mcamatlab server
  - How to get to a consistent set of DLLs and MEXW64 files?
    - not changeable by users
    - updated regularly (e.g. on Windows Wartungstag)
  - Where can I put libraries / mexw64 files?
    - so that on subsequent updates of hosts, DSM packages etc. they are released?

Potential Future work
Matlab support in Hamburg on Windows

- Folder structure to put MEX files and DLLs for Windows
  - By Facility (to allow step by step rollout)
  - Two folders (named release and test): overdesigned?



Potential Future work
TINE kernel unload from memory

- Cleanup of Tine kernel maybe still needs improvement (e.g. when shared library is unloaded from memory)
  - a few years ago, full Matlab session was experiencing stochastic, delayed crashes, mainly on Linux
  - identified cause was cleanup (of Tine library?, of mex file's way to unload TINE library?), triggered by user calling in Matlab e.g. 'clear mex', 'clear all' to clean his workspace
  - introduced mexLock (Matlab API function call) to xcomm and avine_tine_read_images
    - Mex file and its libraries are not removed from memory when e.g. clear mex, clear all are called by Matlab user
  - amout of Matlab crashes greatly reduced

Potential Future work
TINE kernel unload from memory

- Cleanup of Tine kernel
  - first step, if Stefan has time: perform tests (with disabled mexLock) to see whether cleanup problem still exists

Potential Future work
Video transport via TINE: UDP Multicast → TCP

- Roadmap to support higher resolution (more than 1.5 Megapixel) and higher bit-depth (> 8 bit per pixel)
  - Basler acA2440-20gm CMOS 2464x2056 (5 Megapixel) 12 bit
  - at E-Weg, at SRINT2
    - operation with handbrake: limit resultion by 2x2 binning (1232x1028 pixels)

Jobs for Java Developers

- Problems
  - higher resolution than 2000x2000 pixels not supported in Java Video (Analysis Server)
  - higher bit depth than 8 bits per pixel partly not supported in Java Video (Analysis part in AcopVideo and/or Analysis server)
  - stable Multicast only possible for low bandwidth (few MB/s), requires time-consuming fine tuning

Potential Future work
Video transport via TINE: UDP Multicast → TCP

- Multicast is problematic, the higher the amount of packets
  - lots of packets per e.g. 10 MB video image (modern CMOS 5 Megapixel cameras already in use): 7124 packets (headers not included), at 10 images per second?
    - if one packet of a video frame is lost, full video frame is lost
  - Jumbo packets (i.e. 64.000 byte TINE PacketMTU) absolutely not recommended (can easily lead to IPv4 reassembly storm), packet size must be less than MTU (currently 1472 bytes)
  - only one multicast ‘stream’ per host IP, i.e. all servers on one hosts are combining their multicast-traffic, client needs to sort through

Potential Future work
Video transport via TINE: UDP Multicast → TCP

- migration to TCP for higher bandwidths (more than a few MB/s) is the only option
- but 1 stream per client (at higher bandwidths few clients can easily fill the servers outgoing network bandwidth)
- 30-50 MB/s per video stream easily imagineable on Gigabit network
- if four clients want to get a stream at 50 MB/s, this means 200 MB/s network bandwidth is needed at server (but more than Gigabit bandwidth on server output is neither tested nor imagineable, or?)

- How to keep things under control (stable transmission)?
  - hand managed list of client Ips which are allowed to access property Frame.Sched?
  - each IP can receive the video stream only once?
  - users can query the server to see a list of clients for this property (lists currently receiving clients as well as clients which have been receiving in the last n (1?, 5?) minutes)?
  - ?

Thank you for your attention!

Questions?

Comments?

Remarks?