

# TINE Video System v3 (VSv3)

## Talk Overview

1. Initial key-facts
2. Architecture
3. Early Steps
4. Surroundings
5. Development: Current status
6. Development: Next steps
7. Live Presentation



# TINE Video System v3 (VSv3)

## 1. Initial key-facts

- brainstorming started early 2007
- Vision of
  - component-based architecture
  - TINE as underlying data transport
  - high-level data transport layer
  - of simple, well-documented parts that can be connected
  - lots of interfaces to outside world
  - broad use of standardised interfaces



# TINE Video System v3 (VSv3)

## 2. Architecture

- Component-based small software parts
  - which contain at least one input or output transport layer
  - do intermediate processing, source grabbing, display, analysis, stream to DAQ, ...
  - inherent use of multi-core, multi-process and multi-threading without much of the hassle
- TINE R4 based data transport layer uses well-defined and agreed image format

# TINE Video System v3 (VSv3)

## 3. Early Steps

- sit together with collaborative people to tie a interchangeable data format for video (stream) data
- work towards TINE Release 4.0
- write any point that came to mind to paper in order to
  - minimize doubling of work
  - avoid redundancy
  - optimize re-use of components (flexible)
  - at least above average documentation (should be not too time consuming though)

# TINE Video System v3 (VSv3)

## 4. Surroundings

- documentation in-line Doxygen and directly in code (long descriptive variable names e.g.)
- platform-independent sourcecode where possible (easy migration)
- compile warning-free at highest warning level where possible
- use TINE Release 4 as minimum (basic clients should even work with 3)
- do not break downward compatibility by creating intermediate “chim” components

# TINE Video System v3 (VSv3)

## 5. Development: Current Status

### ● C: SGP (small grabber part)

- physical abstraction of image source, acquiring streams, images and do source control
- no additional functionality (no scale factor, no orientation change, ...)
- keep it simple
- TINE command-line based server (easy portable)
- well-defined set of properties
- avoid redundancy by basic code sharing between all forks
- platform-independent basic code (works on Linux, tested, minor TINE problems to be worked on)
- ready (beta versions available): Prosilica GigE (color and B/W), PCVision analogue camera support, Animation playback from disk, DLR radiation hardened design example
- under construction: Directshow, maybe Basler GigE API-based SGP

# TINE Video System v3 (VSv3)

## 5. Development: Current Status (cont'd)

- ready (early beta): Chim intermediate component
  - takes a new stream from (any) SGP
  - transforms it to VSV2 (old TINE Video System) based network transport
  - intermediate: does scale factor insertion, compression and orientation change if required

```

c:\WINDOWS\system32\cmd.exe - G52CompatLayer.exe
returned 36
[22.02.08 12:24:57 LTM] Registered all properties successful.
Exporting the following properties for SGP_USU2_2 (SGPUS2) via configuration file:
e:
> Type <READ ACCESS> -> type of server <-
> S_COMPRESSION <READ ACCESS> -> compression FOURCC <-
> C_CAMNAME <READ ACCESS> -> vsu2 cam name <-
> C_CAMID <READ ACCESS> -> vsu2 cam id <-
> AliveSince <READ ACCESS> -> online since <-
> NONSCHED.GET <READ ACCESS> -> get cur frame nonscheduled <-
> Name <READ ACCESS> -> name of image source <-
> FrameRate <READ ACCESS> -> current framerate (net avg) <-
> C_HEIGHT <READ ACCESS> -> height in pixels <-
> C_DYNAMICSCALE <READ ACCESS> -> scale factor (ro) <-
> C_ORIENTATION <READ ACCESS> -> orientation change <-
> C_BPP <READ ACCESS> -> bits per pixel <-
> C_SCALE <READ ACCESS> -> scale factor <-
> Status <READ ACCESS> -> status of image source <-
> Status <READ ACCESS> -> status code or N48 <-
> Version <READ ACCESS> -> version information <-
> ID <READ ACCESS> -> id code of server <-
> C_WIDTH <READ ACCESS> -> width in pixels <-
> DoRestart <READ!WRITE ACCESS> -> do restart/readback <-
> FRAME.GET <READ ACCESS> -> get current frame scheduled <-
  
```

# TINE Video System v3 (VSv3)

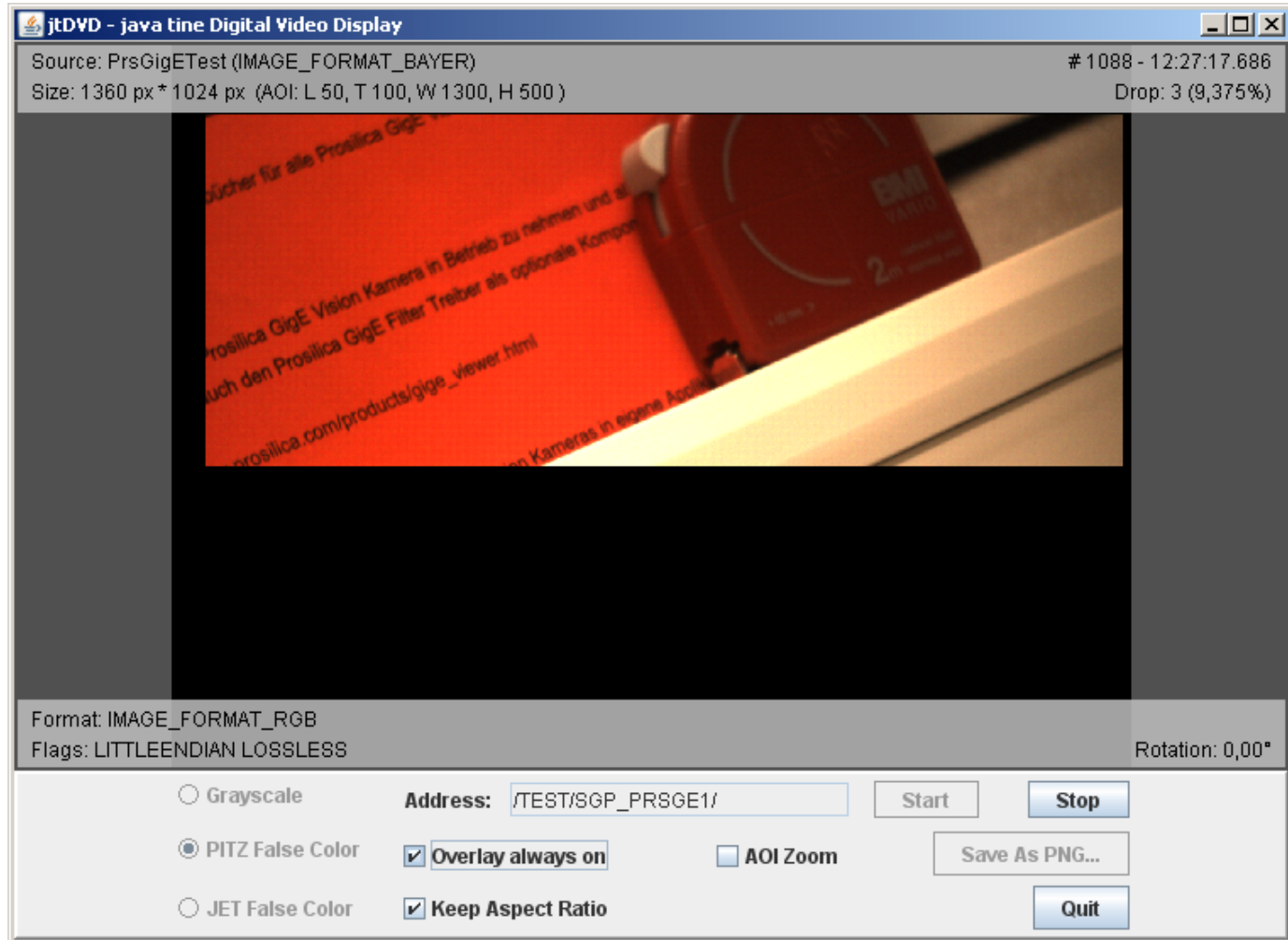
## 5. Development: Current Status (cont'd)

- jtDVD (java tine Digital Video Display)
  - or: Discrete Values Display
  - or: Dynamic Video Display
- **Ready**, Release Candidate Status, but still TINE R3 based
- Small, fast, 100% portable pure java application which takes frames from any new transport layer provider and displays live video with some options
  - false colour modes for grayscale video (properly scaled to bits per pixel like Video Client 2)
  - keep aspect ratio
  - proper area of interest handling (if only AOI from server is transferred)
  - raw value under cursor is provided
  - overlay status (display important properties)
    - Rotation: 0,00°
  - relatively low CPU load, Java 2 v1.5 as minimum required, tested up to now only java6
  - save cooked image data as PNG file for talks, papers, logbook, ...
    - Save As PNG...
    - Quit



# TINE Video System v3 (VSv3)

## 5. Development: Current Status (cont'd)



# TINE Video System v3 (VSv3)

## 6. Development: Next Steps

- Intermediate component which can do basic operation on source stream and delivers (changed) destination stream
  - scale factor insertion
  - orientation change of video data (rotation, flipping)
  - compression of data
  - out-of-AOI masking
- TINE Video Service v3
  - evolution out of TINE Video Service v2
  - improved property naming scheme
  - control and settings repository (also for components)
  - naming lookup
  - assignment between individual cameras and SGPs

# TINE Video System v3 (VSv3)

## 6. Development: Next Steps (cont'd)

- create easy to use interface-libraries for any demanded platform
  - VB, C/C++ (win32, lin32/64, solaris32/64), ROOT, MATLAB, Labview already set
- implement the transport layer as CF\_IMAGE into TINE
  - negotiate with Arthur proper compatibility with DOOCS
- implement in TINE R4.1 a buffer management in order not to copy a lot of video data too much around
  - If >1 component is running on one machine this will significantly improve speed
- do DAQ integration at PITZ
- (maybe) a Video Client 3 (Video Client 2 just with changed input transport design)



# TINE Video System v3 (VSv3)

## Live Presentation