LectureSight

Release 0.1.11

Aug 29, 2018

Table of Contents

LectureSight is an open source OSGI application that uses OpenCL to analyze a video stream in real-time and track the presenter by controlling a PTZ camera.

CHAPTER 1

Quick Start

1. Check the requirements for a Linux server with GPU, overview camera such as a webcam, and PTZ camera supporting VISCA or VAPIX.
2. Check that you have the software dependencies installed (Java, OpenCL and optionally GStreamer).
3. Install LectureSight from a release or build from source.
4. Update the default configuration in `conf/lecturesight.properties` for your overview camera (see some examples).
5. Start LectureSight.
6. Calibrate LectureSight so that the PTZ camera can successfully follow objects within the overview camera's field of view.
7. Set up a Scene Profile to ignore irrelevant regions of the overview image.
8. Watch LectureSight in action in the Object Tracker and PTZ Camera Control windows, and fine-tune the configuration to optimize tracking performance and camera movement.
9. Configure LectureSight to start and stop tracking automatically using the Scheduler.

CHAPTER 2

Community

Join the LectureSight community to ask for help, provide feedback or give suggestions.

Email lecturesight@googlegroups.com

Subscribe by sending a mail to: lecturesight+subscribe@googlegroups.com

CHAPTER 3

Issues

Report bugs or file feature requests on the LectureSight JIRA Issue Tracker

CHAPTER 4

Contents

4.1 Config

4.1.1 Calibration

After getting the overview camera and the PTZ camera to work, we have to calibrate the system so that the two can work together correctly.

Co-ordinates

LectureSight uses 3 co-ordinate systems:

- Video Analysis uses the dimensions of the overview camera image (for example 640x360). Pixel counts and movement thresholds defined for object tracking therefore refer to these dimensions.
- The Camera Operator uses a normalized co-ordinate system from -1 to 1 on both the horizontal and vertical axes, where the top-left is (-1, -1), centre is (0,0) and bottom-right is (1,1). Zoom is mapped from 0 (wide) to 1 (close).
- The Steering Worker uses the co-ordinate system of the PTZ camera.

  VAPIX cameras use degrees as co-ordinates. As LectureSight manages PTZ co-ordinates as integers, VAPIX co-ordinates are scaled up by 100 by the camera driver to preserve precision, so 37.65 degrees is represented as 3765.

Inverted cameras

If the Overview Camera is mounted inverted (up-side down), you can invert the overview image using:

```
cv.lecturesight.framesource.inverted=true
```

If the PTZ Camera is mounted inverted, you may need to add these properties for some VISCA cameras:

```
cv.lecturesight.ptz.steering.worker.relativemove.xflip=true
cv.lecturesight.ptz.steering.worker.relativemove.yflip=true
```

and this property for VAPIX cameras:

```
cv.lecturesight.vapix.camera.inverted=true
```

Scene Profile

Create a new Scene Profile to restrict the area in which the system tracks objects.

Scene Limits

These 4 configuration properties map the overview image to the camera PTZ co-ordinates.

```
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.left
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.right
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.top
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.bottom
```

Initially these values are not set, and thus the limits of the camera's co-ordinates are used as the scene limits.

Manual Calibration

To set values for the four scene limits, disable camera steering by using the following command on the console

```
cs:off
```

Move the production camera (using the PTZ Camera's remote control or web interface) first so that the PTZ Camera is centred on the top-left point of the overview image, and then on the bottom-right point.

Note the camera's co-ordinates at those points in the PTZ Camera Control window. Update the `lecturesight.properties` configuration with those values, and restart LectureSight.

Marker Calibration

!!! info "Automatic marker calibration is only supported for VAPIX cameras."

Marker calibration maps overview co-ordinates to camera pan and tilt values by matching positions on the overview image with camera preset positions.

> The calibration process creates horizontal and vertical conversion models using spline interpolation inside the range of the marker points, and linear extrapolation outside the range, to convert between overview image co-ordinates and camera positions. This will in general provide more accurate results than setting the scene limits manually.

To set up marker calibration:

- In the Scene Profile Editor, identify 3 or more fixed points on the image (for example the corner of a fixed blackboard, or a light switch). The set of points should cover most of the horizontal and vertical range of the presentation area.
- Create calibration markers at each point. Give each marker a name, for example `m1`, `m2`, `m3`, `m4`, `m5`, and save the profile.
- In the camera web interface, create a set of presets with the same names as the calibration markers. At each preset, the camera should be centred on the venue feature identified by the matching calibration marker.
- Restart LectureSight, or use the console command `cs:calibrate` to trigger marker calibration:

```
g! cs:calibrate
Automatic calibration completed
```

The calibration process discovers the pan and tilt co-ordinates of each preset by moving the camera in turn to each position, with a pause time of 2.5s between presets.

On subsequent startup, the Steering Worker will report whether automatic calibration has been used, and if successful, the values of the calculated pan and tilt thresholds:

```
Automatic calibration, camera pan/tilt limits: pan -3673 to 3649, tilt -2596 to 2348
```

Once automatic calibration has successfully run, changes to the configured scene limits will have no effect, as the calculated values will be used.

If no calibration markers have been created or there are too few matching markers and presets, the log will include:

```
Automatic calibration not possible
```

This can be caused by markers that are too close together in the x or y axes. Try to adjust the marker positions so they are distributed across the field of view, both horizontally and vertically.

Initial position

Set the initial position of the PTZ camera

```
cv.lecturesight.cameraoperator.ptz.pan=0.0
cv.lecturesight.cameraoperator.ptz.tilt=0.0
cv.lecturesight.cameraoperator.ptz.zoom=0.0
```

Frame width

Set the frame width of the PTZ camera at the configured zoom position, relative to the width of the overview image, which is 2 in normalized co-ordinates (-1 to 1).

For example a frame.width of 0.5 means that the PTZ Camera's image is 25% of the width of the overview image (0.5 / 2).

```
cv.lecturesight.cameraoperator.ptz.frame.width=0.5
```

You can verify visually that the frame width is correct by looking at the frame boundary guides on the PTZ Camera Control window.

4.1.2 Configuration Examples

Logitech C920 Overview Camera, Axis V5915 PTZ Camera

```
# --- Webcam V4L2 framesource
cv.lecturesight.framesource.input.mrl=v4l2:///dev/logitech[width=640;height=360;Power␣
↪Line Frequency=50 Hz;Focus, Auto=0;Focus (absolute)=0;Zoom, Absolute=200]
cv.lecturesight.framesource.v4l.format=MJPEG
cv.lecturesight.framesource.maxfps=25

# --- VAPIX Configuration (Axis cameras)
cv.lecturesight.vapix.camera.host=camera-hostname.some.domain
cv.lecturesight.vapix.camera.username=root
cv.lecturesight.vapix.camera.password=PASSWORD
cv.lecturesight.vapix.camera.pan.min=-17000
cv.lecturesight.vapix.camera.pan.max=17000
cv.lecturesight.vapix.camera.tilt.min=-2000
cv.lecturesight.vapix.camera.tilt.max=9000
cv.lecturesight.vapix.camera.zoom.min=1
cv.lecturesight.vapix.camera.zoom.max=9999
cv.lecturesight.vapix.camera.zoom.maxspeed=1
cv.lecturesight.vapix.updater.interval=100

# --- Video Analysis
cv.lecturesight.videoanalysis.change.threshold=30
cv.lecturesight.videoanalysis.cell.activation.threshold=2
cv.lecturesight.videoanalysis.object.cells.min=15
cv.lecturesight.videoanalysis.object.cells.max=128
cv.lecturesight.videoanalysis.object.dormant.max=1500
cv.lecturesight.videoanalysis.object.match.threshold=15

# --- Camera Steering Worker (relative movement)
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.pan=0.65
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.tilt=0.2
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.x=2000
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.y=2000
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.x=35
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.y=200
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.left=-2500
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.right=2500
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.top=-600
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.bottom=-5000

# --- Pan-only Camera Operator
cv.lecturesight.cameraoperator.ptz.pan=0.0
cv.lecturesight.cameraoperator.ptz.tilt=0.4
cv.lecturesight.cameraoperator.ptz.zoom=0.1
cv.lecturesight.cameraoperator.ptz.frame.width=0.8
cv.lecturesight.cameraoperator.ptz.timeout=500
cv.lecturesight.cameraoperator.ptz.idle.preset=Home
cv.lecturesight.cameraoperator.ptz.start.preset=Start
cv.lecturesight.cameraoperator.ptz.target.limit=2

# --- Scene Profile
cv.lecturesight.profile.manager.active.profile=overview

# --- Scheduler: watch the Galicaster schedule file
```

(continues on next page)

(continued from previous page)

```
cv.lecturesight.scheduler.enable=true
cv.lecturesight.scheduler.schedule.file=/usr/share/galicaster-repository/attach/
↪calendar.ical
cv.lecturesight.scheduler.timezone.offset=2
cv.lecturesight.scheduler.tracker.leadtime=10

# --- Enable DropWizard metrics collection
cv.lecturesight.util.metrics.enable=true

# --- Save overview snapshots
cv.lecturesight.framesource.snapshot.file=/opt/ls/record/overview.png

# --- Status Reporting
cv.lecturesight.status.enable=true
cv.lecturesight.status.name=hahn1
cv.lecturesight.status.url=http://lsdashboard.some.domain/lecturesight/
cv.lecturesight.status.interval=60
```

Raspberry Pi Overview Camera, VISCA Vaddio HD-USB PTZ Camera

```
# --- Overview camera
cv.lecturesight.framesource.input.mrl=rtph264://rpi-cam.some.domain:8554

# --- VISCA Camera Configuration
com.wulff.lecturesight.visca.port.device=/dev/ttyUSB0
com.wulff.lecturesight.visca.updater.interval=200

# --- Video Analysis
cv.lecturesight.videoanalysis.change.threshold=48
cv.lecturesight.videoanalysis.cell.activation.threshold=3
cv.lecturesight.videoanalysis.object.cells.min=20
cv.lecturesight.videoanalysis.object.cells.max=128
cv.lecturesight.videoanalysis.object.dormant.max=1500

# --- Camera Steering Worker
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.pan=0.5
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.tilt=0.1
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.x=3000
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.y=1000
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.x=300
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.y=200
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.left=-5500
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.right=3870
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.top=-600
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.bottom=-5000
cv.lecturesight.ptz.steering.worker.relativemove.focus.fixed=true
cv.lecturesight.ptz.steering.worker.relativemove.move.initial.delay=5000

# --- PTZ Camera Operator
cv.lecturesight.cameraoperator.ptz.pan=0.3
cv.lecturesight.cameraoperator.ptz.tilt=-0.40
cv.lecturesight.cameraoperator.ptz.zoom=0.55
cv.lecturesight.cameraoperator.ptz.timeout=500
cv.lecturesight.cameraoperator.ptz.idle.preset=0
cv.lecturesight.cameraoperator.ptz.target.limit=2
```

(continues on next page)

(continued from previous page)

```
# -- Scene Profile
cv.lecturesight.profile.manager.active.profile=overview
```

4.2 Core

4.2.1 Configuration

The *Configuration* service in the `lecturesight-utils` bundle provides configuration settings to LectureSight services.

At startup, the configuration is loaded from the files `conf/lecturesight.properties` and `conf/build.properties`

When running, many configuration settings can be updated live through the command-line Console or the System Configuration user interface.

Commands

Command	Description
config:defaults	Show default configuration values
config:load	Load configuration from filename
config:save	Save configuration to filename
config:set	Set configuration key to value
config:show [prefix]	Show all configuration values, optionally matching the prefix.
config:buildinfo	Shows the value of the immutable cv.lecturesight. buildinfo property, if set.
config:version	Show the value of the immutable cv.lecturesight. version property, if set.

4.2.2 Console

The console provides a command-line interface to LectureSight services and the OSGI container.

!!! warning The shell provides access to all local files. Do not enable console access from non-local IPs unless you have additional security precautions in place such as a firewall.

The console shell is provided by Apache Felix Gogo

Access

The console is launched on startup by the `start_lecturesight.sh` script. To disable launching the console, add the property:

```
-Dgosh.args=--noi
```

to the `CONFIG_OPTS` variable in the startup script.

Connect to the console from another terminal window using telnet:

```
telnet localhost 2501
```

or netcat:

```
nc localhost 2501
```

Configuration

Access to the console is configured in `conf/config.properties`:

```
osgi.shell.telnet.ip=127.0.0.1
osgi.shell.telnet.port=2501
osgi.shell.telnet.maxconn=2
```

Commands

The following commands may be helpful:

Command	Action
help	List available console commands
lb	List bundle status
scr:list	Show Service Component Runtime status

Services

The following services provide console commands:

Command Prefix	Service
config	Configuration
cs	Camera Steering Worker
display	Display Service
ls	Heartbeat
metrics	Metrics
scheduler	Scheduler
va	Video Analysis
felix, gogo, obr, scr	Commands provided by the OSGI container

4.2.3 Frame Sources

A frame source provides images to LectureSight for processing, typically from an overview camera.

The *FrameSourceProvider* service in the `lecturesight-framesource-impl` bundle provides the infrastructure responsible for managing FrameSource implementations. It discovers video input plugins and is responsible for setting up configured *FrameSources* with the proper input plugin.

Configuration

Key	Default	Description
cv.lecturesight.framesource.input.mrl	v4l:///dev/video0[width=320;height=240]	MRL of the video input from the overview camera
cv.lecturesight.framesource.inverted	false	If true, rotates the framesource 180 degrees. Used for cameras mounted inverted (upside-down)
cv.lecturesight.framesource.maxfps	60	Maximum fps rate at which frames should be read from the device.
cv.lecturesight.framesource.snapshot.file		Filename to which overview image snapshots should be saved periodically
cv.lecturesight.framesource.snapshot.interval	60	Interval in seconds to save overview image snapshots

The Media Resource Locator (MRL) has the following form:

```
type :// path [options]
```

Part	Meaning
type	the type of the input, determines which input plugin is used
path	path to the input, usually a Linux device or file
options	additional arguments for the input plugin

V4L Frame Source

The `lecturesight-framesource-v4l` bundle provides a FrameSource implementation for accquiring frames from Video4Linux and Video4Linux 2 devices such as webcams and frame grabbers.

Arguments for creation of a new FrameSource from this implementation can be provided in the FrameSource MRL. If an argument is not present in the MRL, the default value is taken from the configuration properties.

Usage

The *type* for this FrameSource implementation is *v4l* or *v4l2*. The *path* is the path to a Linux video device such as `/dev/video`

Available arguments are `width`, `height`, `standard`, `channel`, `quality`.

Examples

Video4Linux device `/dev/video0` as input with QVGA resolution.

```
cv.lecturesight.framesource.input.mrl=v4l:///dev/video0[width=320;height=240]
```

Video4Linux2 device `/dev/video0` as input with QVGA resolution.

```
cv.lecturesight.framesource.input.mrl=v4l2:///dev/video0[width=320;height=240]
```

Configuration

Key	Default	Description
cv.lectureSight.framesource.v4l.channel	0	Default video input. Usually not used with USB webcams but rather with capture cards. This can be useful with capture cards, since they are by default set to tuner input and need to be set to composite (usually 1).
cv.lectureSight.framesource.v4l.format	YUYV	YUYV or MJPEG for webcams
cv.lectureSight.framesource.v4l.quality	0	Default encoding quality. Only used for devices that provide encoded video streams (such as MPEG2 or MJPEG). Value range depends on device driver.
cv.lectureSight.framesource.v4l.resolution.width	320	Default width for input frames.
cv.lectureSight.framesource.v4l.resolution.height	240	Default height for input frames.
cv.lectureSight.framesource.v4l.standard	0	Default video standard. Usually not used with USB webcams but rather with capture cards. Which value indicates a certain standard (eg. PAL-X/NTSC) depends on the driver of the video device.

V4L2 Device Controls

It is also possible to set V4L2 device controls by including them in the option list. To find out which controls the v4l2 device supports, look at the device information logged by the *V4LFrameGrabberFactory.createFrameGrabber* method. For example this device has a control named `Brightness` which can be set in the range 0 to 255:

```
2017-12-12 06:31:43.587 INFO    CM Configuration Updater (Update: pid=org.apache.
↪felix.fileinstall.235d4e28-9777-462b-93dd-d7a91af9d18e) V4LFrameGrabberFactory.
↪createFrameGrabber() : Name: Brightness = 128 Type: CTRL_TYPE_SLIDER Values: [ 0 ..␣
↪255 ] increment: 1
```

To set this value in the MRL, add it to the option list:

```
cv.lectureSight.framesource.input.mrl=v4l2:///dev/logitech[width=640;height=360;
↪Brightness=100]
```

GStreamer Frame Source

The `lectureSight-framesource-gst` bundle provides a Frame Source implementation that uses a user-defined GStreamer pipeline to capture frames. The pipeline definition is given in the format that is used in the `gst-launch` command. The implementation adds a color space element, a capsfilter and an appsink to the user-defined pipeline that are responsible for converting the frames to RGB format and hand them over to the system.

Usage

MRLs for this Frame Source contain the GStreamer pipeline definition in the format used in the gst-launch command.

```
gst://(gst-launch definition)[(options)]
```

Note: You must define a pipeline with at least two elements, otherwise the creation of the Frame Source will fail (issue LS-71).

Options

drop – optional, default: *true*

Sets the *drop* property of the `appsink` that hands the frames over to the system.

For real-time frame sources such as cameras it is recommended to set this value to *true* so that the newest frames is uploaded to the GPU for video analysis.

When testing with a non-live frame source such as video files, *drop* may be set to *false* so that the system gets every frame for analysis.

More information on the `appsink` element and its *drop* property are available in the GStreamer documentation.

Examples

Consume an RTSP stream from an Axis IP camera:

```
cv.lecturesight.framesource.input.mrl=gst://rtspsrc location=rtsp://venue1-camera.
↪someplace.edu/axis-media/media.amp ! rtph264depay ! avdec_h264 ! videoconvert
```

Use a V4L2 source:

```
cv.lecturesight.framesource.input.mrl=gst://v4l2src device=/dev/video0 !␣
↪ffmpegcolorspace [drop=true]
```

Use a gstreamer test pattern:

```
cv.lecturesight.framesource.input.mrl=gst://videotestsrc ! identity
```

RTPH264 Frame Source

The `lecturesight-framesource-rtph264` bundle provides a FrameSource implementation that reads H264 video from an RTP stream. This is a special-purpose gstreamer pipeline designed to consume video from a RaspberryPi camera with minimal latency.

Example:

```
cv.lecturesight.framesource.input.mrl=rtph264://venue1-camera.someplace.edu:8554
```

Videofile Frame Source

The `lecturesight-framesource-videofile` bundle provides a FrameSource implementation that reads frames from a video file using gstreamer.

It depends on the set of codecs installed in the host operating systems what formats are supported. Standard MPEG file formats should always be supported since they are included in most standard installations.

The *type* for this FrameSource implementation is *file*. The *path* is the path to a video file. There are no arguments for this FrameSource implementation.

Example, using the file `/opt/ls/media/overview.mp4` as a frame source:

```
cv.lecturesight.framesource.input.mrl=file:///opt/ls/media/overview.mp4
```

4.2.4 Heartbeat

The `lecturesight-heartbeat` bundle provides the *Heartbeat* service. It is responsible for controlling the execution of the video analysis services. It listens for several OpenCL service signals that indicate that all services have finished their work for the current frame and kicks off the analysis of the next frame.

Configuration

Key	Default	Description
cv.lecturesight.heartbeat.autostart	2500	Delay time in ms after startup before enabling tracking and camera control. Set to -1 to make LS wait for 'ls:run' command in the console before tracking.
cv.lecturesight.heartbeat.listens.to	VALID_DONE	comma-separated list of signal name the Heartbeat service waits for before kicking off the processing of the next frame. Do not change this property.

Console Commands

Command	Description
ls:run	Activate the video analysis subsystem.
ls:step [frames]	Run the video analysis subsystem run for the given number of frames. If the argument is omitted, the video analysis is run for 1 frame. This command is especially useful for debugging when working with a video file frame source instead of a live video input.
ls:pause	Pause the video analysis service without de-initializing the service.
ls:restart	Re-initializes the heart beat service and start the video analysis subsystem. This command might be used when the listens.to property was changed since the internal signal barrier will be newly setup up.
ls:stop	Stops the video analysis subsystem and de-initializes the heartbeat service.

4.2.5 Logging

Logging is provided by TinyLog, and configured in `conf/log.properties`.

!!! tip LS-187: Exceptions during bundle startup are not logged by TinyLog, and will be visible on the console only

The default configuration logs to the console. This configuration will log to console and file:

```
tinylog.level = debug

# INFO/ERROR Logging to console
tinylog.writer1 = console
tinylog.writer1.format = {date:HH:mm:ss} {level}: {class_name}.{method}() -- {message}
tinylog.writer1.level = info

# DEBUG/INFO/ERROR logging to file
tinylog.writer2 = file
tinylog.writer2.filename = log/ls.log
tinylog.writer2.level = DEBUG
tinylog.writer2.format = {date:yyyy-MM-dd HH:mm:ss.SSS} {{level}|min-size=7} {thread}
↪{class_name}.{method}() : {message}
```

Available configuration options are described in TinyLog Configuration.

4.2.6 Metrics

The Metrics Service records LectureSight activity for later analysis and quality improvement.

When the metrics service is enabled, metrics are saved to the file `metrics/metrics.json` when LectureSight exits, or at the end of each scheduled event if the Scheduler is enabled.

Configuration

Key	Default	Description
cv.lecturesight.util.metrics. enable	false	Set to true to enable the metrics service
cv.lecturesight.util.metrics. csv.enable	true	Enable CSV reporting
cv.lecturesight.util.metrics. csv.interval	30	Interval to write out updates to CSV files
cv.lecturesight.util.metrics. jmx.enable	true	Enable JMX reporting
cv.lecturesight.util.metrics. log.enable	true	Enable log reporting
cv.lecturesight.util.metrics. log.interval	300	Interval to write out updates to log file

Console Commands

Command	Description
metrics:list	List the keys of all registered metrics.
metrics:pause	Suspend metric reporting.
metrics:reset	Reset all metrics.
metrics:resume	Resume metric reporting.
metrics:save	Save metrics summary to `metrics/metrics.json`
metrics:show	Show the metrics JSON summary

4.2.7 OpenCL

The *OpenCLService* in the `lecturesight-opencl-impl` bundle is responsible for initializing the GPU.

OpenCL is configured through Java system properties that are set in the `start_lecturesight.sh` script. To change these properties, edit the script before starting LectureSight.

Configuration

Property	Default	Description
ocl.device.type	GPU	Set the *CLDevice.Type*: CPU or GPU
ocl.use.gl	false	Use current OpenGL context if true
ocl.profiling	false	Enable profiling if true

Profiling

If OpenCL profiing is enabled, LectureSight will save profiling information to the files `frametimes-TIMESTAMP.csv` and `profiling-TIMESTAMP.csv`.

4.2.8 Scene Profile

The Scene Profile defines the regions of the overview image that are ignored for tracking purposes.

Profiles can be edited by the Scene Profile Editor and are stored in the `profiles` directory.

Configuration

Key	Default	Description
cv.lecturesight.profile.manag er.active.profile	default	Name of the active profile.

Note that the default profile cannot be saved, so to use a custom profile, create a new profile and update this setting with the new profile name.

4.2.9 Scheduler

The `lecturesight-scheduler` bundle provides a service that loads a schedule from an iCalendar (RFC-2445) file and starts and stops object tracking and camera control for each event.

The service is designed to allow LectureSight to follow the recording schedule of an Opencast capture agent such as Galicaster. Changes to the file are detected and the internal schedule is updated automatically. When the file is deleted, all events are removed.

The video analysis and tracking components may need a certain time to adapt to the scene before producing correct tracking results. To prevent unnecessary camera movement caused by false positives, the service can be configured to start camera control some time after the object tracking has been activated.

Configuration

Key	Default	Description
cv.lecturesight.scheduler.enable	false	Whether to enable the scheduler.
cv.lecturesight.scheduler.schedule.file	schedule.ics	The filename of the iCal file holding the schedule.
cv.lecturesight.scheduler.agent.name		A capture agent name the service will look for in case the iCal holds schedules for more than one capture agent. If not set, the service will take every event from the iCal into account.
cv.lecturesight.scheduler.timezone.offset	1	The time zone offset to add to the event times from the schedule.
cv.lecturesight.scheduler.tracking.leadtime	0	The time (in seconds) the service will wait after the object tracking has been activated before the camera control is activated.

Console Commands

Command	Description
scheduler:start	Activates tracking and camera steering.
scheduler:stop	Deactivates tracking and camera steering.
scheduler:status	Shows the scheduler status: active or idle

4.2.10 Status Service

The *StatusService* provided by the `lecturesight-status` bundle sends LectureSight configuration and status information to a remote service such as a dashboard.

Configuration

Key	Default	Description
cv.lecturesight.status.enable	false	Set to true to enable status updates
cv.lecturesight.status.url		The URL of a remote service endpoint for HTTP POST updates
cv.lecturesight.status.name	lectures ight	A descriptive name of the server or venue
cv.lecturesight.status.interv al	60	Interval in seconds between status updates

POST data

The status service sends an HTTP POST request with the following multipart/form-data fields:

Field name	Type	Description
name	text/plai n	Name configured in `cv.lecturesight.status.name`
status	text/plai n	Tracking status: active or idle
metrics	applicati on/json	Metrics summary from the Metrics Service
profile	text/plai n	The active scene profile definition
overview-image	applicati on/octet- stream	The overview image snapshot if configured: file contents of `cv.lecturesight.framesource.snapsh ot.file`

4.3 Development

4.3.1 Style Guide

Code

- LectureSight uses the maven checkstyle plugin to enforce java code style. The checkstyle rules are defined in `src/docs/checkstyle/lecturesight-checkstyle.xml`

- Indents are two spaces.

Documentation

The following conventions are used in documentation:

- LectureSight has L and S capitalized.
- GStreamer has G and S capitalized.
- Names that refer to classes or interfaces are italicized and follow the case rules used in the source, for example *HeartBeat* service.
- Module names (src/modules/) are quoted with backticks, for example `lecturesight-heartbeat` module.
- File names, configuration values or text entries (for example console commands) are quoted with backticks, for example `conf/lecturesight.properties`
- Compulstory command-line or console command arguments are shown in angle brackets, for example `config:set <key> <value>`
- Optional command-line or console command arguments are shown in square brackets, for example `ls:step [frames]`
- User Interface elements such as menu entries are referred to using **emphasis**

4.4 Hardware

4.4.1 Camera Layout

For best results, the overview camera should be co-located with the PTZ camera. For ease of calibration, position both cameras in the rear centre of the venue.

4.4.2 Overview Camera

LectureSight requires an overview camera which captures the entire presentation area.

Camera Types

USB webcams, other v4l devices such as analog SD video cameras connected to a fast frame grabber, or any image source which can be provided through a GStreamer pipeline can be used.

For real time operation, devices that provide raw video streams are recommended, as encoding and decoding of frames can lead to several hundred milliseconds of delay.

Though the resolution for the overview camera should not be too high in order not to jeopardize real-time performance (usually VGA), the image quality of the model chosen as overview camera directly impacts tracking accuracy and reliability. Cheap USB webcams, for example, sometimes show a habit of aggressively adjusting color channels. Such

Fig. 1: PTZ and Overview Camera mounting

behavior can compromise correct function of the tracker. With higher quality (720p) webcams, color- and contrast-stable images can be achieved which is optimal for the video analysis.

Also most stationary analog video cameras produce a stable image that meets the needs of the tracking algorithms. In order to use an analog video camera as the overview camera, a frame grabber has to convert the analog signal to digital frames. It is suggested to use internal PCI(e) frame grabber hardware with direct memory access (DMA). Such frame grabbers write raw frame directly into the host system's memory from where they can directly be copied to GPU memory, thus avoding unnecessary memory copy or encoding/decoding operations that can produce delays.

USB WebCams

In general any USB WebCam that is compatible with Video4Linux can serve as overview camera for LectureSight. Experiments with different models have shown that the more stable the image delivered by camera is the more stable is the tracking. Especially automatic color adjustment can cause the object tracker to reset frequently. The following USB cameras have been tested with the system:

Logitech

- HD Pro Webcam C910
- HD Pro Webcam C920
- HD Pro Webcam C930
- FaceVision (no longer in business)
- TouchCam N1

The following USB extension products have been tested with LectureSight, for cases where the capture agent is not located near to the overview camera:

- Unitek Y-262 20m USB 2.0 active extension cable
- ATEN USB 2.0 extender over CAT5/6 cabling (up to 60m)

IP Cameras

IP Cameras can be used through the GStreamer Framesource bundle, which supports any GStreamer pipeline, including RTSP sources. See LS-60 for details.

Overview cameras which have been tested include:

- Axis P1428-E network camera (with a stream configured for 1280x720 or 640x360).

4.4.3 PTZ Camera

LectureSight steers a pan-tilt-zoom (PTZ) camera in real-time to follow the presenter, based on video analysis of the overview image. Three camera control protocols are supported: VISCA, VAPIX and ONVIF.

Sony VISCA Protocol

The system supports Sony's VISCA protocol out-of-the-box. The driver detects the model and version of the camera and loads a fitting parameter set. If the camera model is unknown, the driver loads the profile for the Sony EVI-D30. Most Sony PTZ cameras can be switched into a D30 compatibility mode.

The following cameras have been tested with LectureSight and have a dedicated camera profile:

- Sony EVI-D30
- Sony EVI-D70
- Sony EVI-D100
- Vaddio ClearVIEW HD-USB

AXIS VAPIX Protocol

The system supports the AXIS VAPIX protocol for PTZ cameras. The following cameras have been tested with LectureSight:

- AXIS V5915 PTZ Network Camera

ONVIF Protocol

The system supports the ONVIF protocol for cameras. Any PTZ camera which supports ONVIF may work with LectureSight, although only the AXIS V5915 camera has been tested with ONVIF.

4.4.4 Server Requirements

Operating System

LectureSight is designed to run on a Linux system equipped with a GPU and running a GUI desktop. Ubuntu 16.04 or later is recommended.

GPU

The video processing portions of the system have been implemented for the GPU using the OpenCL standard for cross-platform parallel computing. Most modern graphics cards are compatible with OpenCL. Graphics cards with an NVIDIA GPU that are label as *CUDA compatible* are OpenCL compatible. ATI graphics chips that are labeled *Stream SDK compatible* are also compatible with OpenCL.

For use in a real-time scenario, it is suggested to use a GPU with a least six OpenCL compute units and at least 512 MB of graphics memory. Such a configuration can be found, for example, in graphics cards equipped with the NVIDIA GT 220 chip set.

For NVIDIA GPUs, the number of CUDA units divided by eight yields the number of OpenCL compute units. For example, the GT 220 has 48 CUDA units, thus the GPU provides 6 compute units in OpenCL.

The GPU must support the following image formats (see LS-165)

- BGRA / UnsignedInt8
- INTENSITY / UnsignedInt8

The following GPUs are known to work with LectureSight:

- NVIDIA NVS 310, NVS 315, GT 220, GeForce GTX 750 Ti

These GPUs do not support the image formats required by LectureSight and thus can not be used:

- AMD ATI Radeon HD 6770M (older iMac)
- NVIDIA GeForce GT 750M (newer iMac)

- Intel HD Graphics 5000 (Mac Air)

CPU

LectureSight was designed with the goal of being capable of running on a single 2 GHz core. The system should not take up all CPU resources in a modern system so that video recording software can run alongside on the same computer. Thus LectureSight should run on any modern system. The suggestion is to use a system with a CPU of at least the performance of an Intel Core 2 Duo with 2,2 Ghz.

RAM

Since video ananlysis is nearly entirely done on the graphics card, there are no special memory requirements for LectureSight when run stand-alone. The usual default memory configuration for the Java VM suffice.

Storage

LectureSight requires less than 50 MB of disk space when installed. Additional space may be used by log files or metrics if configured. No other data is incrementally saved.

4.5 Install

4.5.1 Software dependencies

OpenCL

LectureSight requires OpenCL 1.1 or later drivers for GPU hardware.

Java

LectureSight requires Java 8 or later.

GStreamer

These LectureSight modules make use of GStreamer:

- videofile framesource
- gst framesource
- rtph264 framesource

LectureSight requires GStreamer 1.8 or later.

4.5.2 Install a release

Download the latest LectureSight release from

https://bitbucket.org/bwulff/lecturesight/downloads/

Extract the software to `/opt/ls` (or another location of your choice)

```
tar -xfvz lecturesight-0.3-rc1.tar.gz
mv lecturesight-0.3-rc1/ /opt/ls
```

Select a camera bundle

You can now Start LectureSight

4.5.3 Build from source

You can build LectureSight from source if you want to run newer code than the last release, or need to make local changes.

Check that you have git and Apache Maven installed.

Create a destination folder for LectureSight:

```
mkdir -p /opt/ls/bundles/application
```

Clone the LectureSight repo:

```
git clone https://bitbucket.org/bwulff/lecturesight.git
```

Check out the branch to build:

```
cd lecturesight
git checkout 0.3-sprint
```

Copy the runtime OSGI bundles and configuration files:

```
cp -R runtime/* /opt/ls/
```

By default the `demonstration` profile will be built, this deploys a dummy PTZ camera so you run LectureSight without access to a real PTZ camera.

Build the LectureSight `demonstration` profile and install the resulting OSGI bundles:

```
cd src
mvn clean install -DdeployTo=/opt/ls/bundles/application
```

Alternatively, edit the `production-with-gui` maven profile in `src/pom.xml` to enable or disable the camera modules that you need for your installation, (see the `ptz-####-only` profiles for the required modules).

```
cd src
mvn clean install -DdeployTo=/opt/ls/bundles/application -Pproduction-with-gui
```

A successful build should end like this:

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 30.631 s
[INFO] Finished at: 2017-12-07T17:07:41+02:00
[INFO] Final Memory: 81M/429M
[INFO] ------------------------------------------------------------------------
```

You can now Start LectureSight

4.5.4 Start LectureSight

With the default configuration, LectureSight will use the video device `/dev/video0` as overview camera with a resolution of 320x240, and a simulated (dummy) PTZ camera.

Change to the LectureSight directory:

```
cd /opt/ls
```

Review the default configuration:

```
nano conf/lecturesight.properties
```

Start LectureSight:

```
./bin/start_lecturesight.sh
```

Console

The Console shell will appear, followed by start-up INFO log entries:

```
____________________________
Welcome to Apache Felix Gogo

g! 16:04:16 INFO: ScriptingStub.activate() -- Activated
16:04:16 INFO: ConfigurationServiceImpl.<init>() -- LectureSight version 0.3
...
```

Type `help` to see the available console commands.

GUI

You will see the main application window appear. The **Services** menu should be populated with a number of entries.

Shutdown

To shut down LectureSight, press `Ctrl-C` at the console, or use the console command `stop 0`, or select **Services | Quit LectureSight** in the GUI.

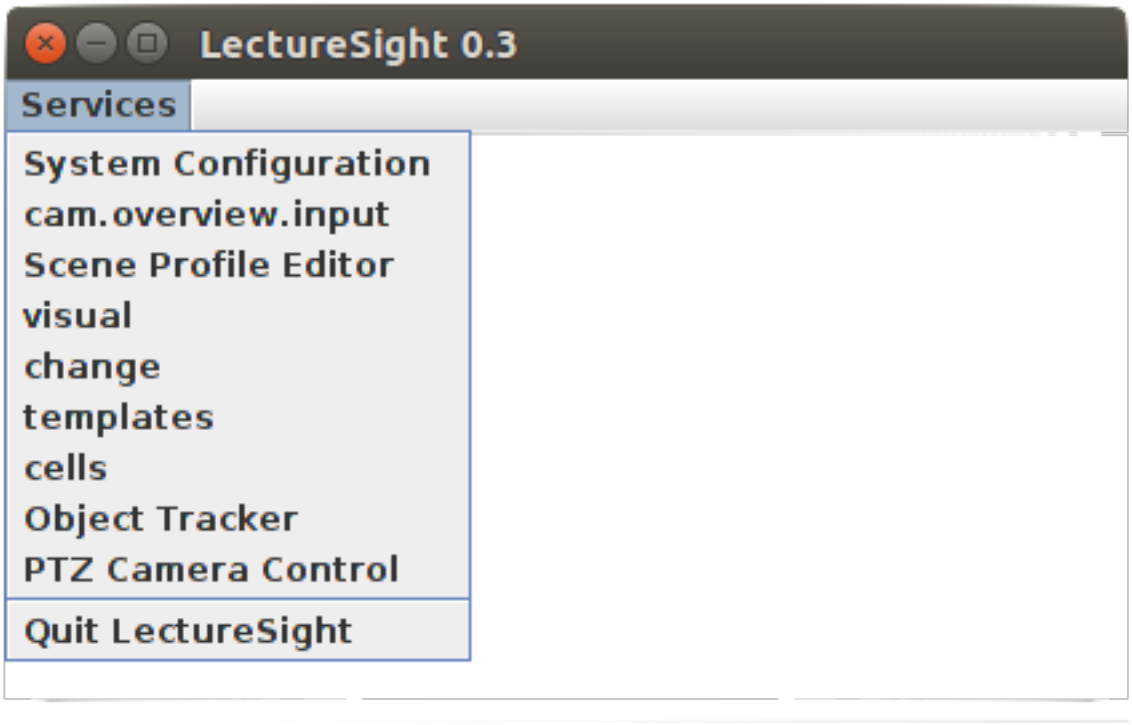

Fig. 2: LectureSight main window

Troubleshooting

OpenCL

If you only see the entry **System configuration**, it is possible that the system was not able to initialize the OpenCL platform and GPU successfully.

Look for the OpenCL device reports in the console, for example:

```
OpenCL device report:

NVIDIA Corporation NVS 315 (driver version: 375.66)

Compute units :  1 at 1046 MHz max

Memories :  global   : 964.4375 MB
constant : 64.0 KB
local    : 48.0 KB

Workgroups :  1024 threads max in 3 dimensions
2D Image size :  16384x16384 max
Work item sizes:  1024 1024 64
```

If you don't find an OpenCL device report in the console, this means that the OpenCL service was not able to find and initalize the OpenCL platform. In this case check the installation of the graphics card driver.

Bundle startup

At the console, type:

```
lb
```

to list installed bundles. The set of system bundles will be listed, followed by LectureSight bundles. A status of `Resolved` or `Installed` indicates that the bundle has not started correctly. For example:

```
g! lb
START LEVEL 2
ID|State      |Level|Name
 0|Active     |    0|System Bundle (4.2.0)
 1|Active     |    1|Apache Felix Configuration Admin Service (1.2.8)
...
38|Active     |    1|LectureSight Object Tracker API (0.3.0)
39|Active     |    1|LectureSight GUI API (0.3.0)
40|Installed  |    1|LectureSight Video4Linux FrameSource (0.3.0)
```

You can attempt to start the bundle to get more information about the cause of the failure. For example:

```
g! start 40
org.osgi.framework.BundleException: Unresolved constraint in bundle cv.lecturesight.
↪framesource.v4l [40]: No matching native libraries found.
```

4.6 Modules

4.6.1 Pan-Tilt-Zoom Camera Operator

The Pan-Tilt-Zoom Camera Operator follows target by moving the camera left, right,up and down as required. This camera operator currently only makes changes to the camera's pan and tilt. Zooming, for now, is locked at the initial value.

This camera operator can be restricted to "pan only" by setting the property *cv.lecturesight.cameraoperator.ptz.tilt.lock = true*

Configuration

Pan, tilt and zoom positions are specified in the normalized co-ordinate system, which represents the camera's pan and tilt positions as ranging from -1 to 1, and the zoom position as ranging from 0 to 1.

Key	Default	Description
cv.lecturesight.cameraoperator.ptz.pan	0.0	Sets the initial pan position (-1 to 1)
cv.lecturesight.cameraoperator.ptz.tilt	0.0	Sets the initial tilt position (-1 to 1)
cv.lecturesight.cameraoperator.ptz.tilt.lock	false	Sets whether the tilting is disabled or not
cv.lecturesight.cameraoperator.ptz.tilt.offset	0.0	Adjust the target's tilt value, for example if you want the camera to centre on the torso, not the head. Ignored if *tilt.lock=true* (-1 to 1)
cv.lecturesight.cameraoperator.ptz.zoom	0.0	Sets the initial zoom position (0 to 1)
cv.lecturesight.cameraoperator.ptz.frame.width	0.5	Sets the width of the PTZ camera's frame (0 to 2)
cv.lecturesight.cameraoperator.ptz.frame.height	0.5	Sets the height of the PTZ camera's frame (0 to 2)
cv.lecturesight.cameraoperator.ptz.frame.trigger.width	0.65	Sets the proportion of the frame width in which the target object can move without triggering the camera to move (0 to 1)
cv.lecturesight.cameraoperator.ptz.frame.trigger.height	0.8	Sets the proportion of the frame height in which the target object can move without triggering the camera to move (0 to 1)
cv.lecturesight.cameraoperator.ptz.target.timeout	2500	Sets the time in milliseconds after the last target movement after which a target will no longer be tracked.
cv.lecturesight.cameraoperator.ptz.tracking.timeout	60000	Sets the time in milliseconds after the last target movement to return to the initial tracking position (0 to disable)
cv.lecturesight.cameraoperator.ptz.idle.preset		The camera preset to move to when idle, if set
cv.lecturesight.cameraoperator.ptz.start.preset		The camera preset to move to at start of tracking (used instead of pan, tilt and zoom values above)

4.6.2 Dummy Camera

The `lecturesight-ptzcontrol-dummy` bundle provides a simulated camera for evaluating or testing LectureSight without a real PTZ camera.

Configuration

Key	Default	Description
cv.lecturesight.ptzcontrol.du mmy.delay	100	Interval between position updates.

4.6.3 VAPIX PTZ Controller

VAPIX is the Axis camera API:

http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php

The communication with the camera is based around HTTP response and requests. The returning value for success is

- HTTP_NO_CONTENT (204): Command has been sent

- HTTP_OK (200): Command sent

and response in text format. The returning text format is structured as [propertyName]=[propertyValue]

Configuration

Key	Default	Description
cv.lecturesight.vapix.camera.host	127.0.0.1	The host name / ip address for the camera.
cv.lecturesight.vapix.camera.username	admin	The username that will be used to authenticate on the camera
cv.lecturesight.vapix.camera.password	admin	The password to use for authentication.
cv.lecturesight.vapix.camera.pan.min	-17000	The minimum pan value to use in translating LectureSight values to camera values. VAPIX uses degrees with fractions for the range that the camera can pan.
cv.lecturesight.vapix.camera.pan.max	17000	The maximum pan value.
cv.lecturesight.vapix.camera.pan.maxspeed	100	The maximum speed for changing the pan value.
cv.lecturesight.vapix.camera.tilt.min	-2000	The minimum tilt value to use in translating LectureSight values to camera values. VAPIX uses degrees with fractions for the range that the camera can tilt.
cv.lecturesight.vapix.camera.tilt.max	9000	The maximum tilt value.
cv.lecturesight.vapix.camera.tilt.maxspeed	100	The maximum speed for changing the tilt value.
cv.lecturesight.vapix.camera.zoom.min	1	The minimum zoom level value.
cv.lecturesight.vapix.camera.zoom.max	9999	The maximum zoom level value.
cv.lecturesight.vapix.camera.zoom.maxspeed	100	The maximum speed for changing the zoom level.
cv.lecturesight.vapix.update.interval	200	The interval to send responses to all the registered camera listeners.

The username should have Administrator privileges to be able to manage camera presets.

Inverted camera

If the camera is mounted inverted, set these 3 properties:

```
cv.lecturesight.vapix.camera.inverted=true
cv.lecturesight.vapix.camera.tilt.min=-9000
cv.lecturesight.vapix.camera.tilt.max=2000
```

4.6.4 ONVIF PTZ Controller

Open Network Video Interface Forum (ONVIF) is a community to standardize communication between IP-based security products, in this case PTZ cameras.

The communication with the camera is defined as a web service. A wrapper handles most of the Simple Object Access Protocol (SOAP) messaging by using Java Architecture for XML Binding (JAXB) to map the objects to Extensible Markup Language (XML).

!!! warning "Beta code" The ONVIF PTZ Controller has not been extensively tested or used in production.

Configuration

Key	Default	Description
cv.lecturesight.onvif.camera.host	127.0.0.1	The host name / ip address for the camera.
cv.lecturesight.onvif.camera.username	admin	The username that will be used to authenticate on the camera, the user is a ONVIF / web service specific user that has access to the web service. The user should have Administrator privileges to be able to manage camera presets.
cv.lecturesight.onvif.camera.password	admin	The password to use for authentication.
cv.lecturesight.onvif.camera.pan.min	-1700	The minimum pan value to use in translating LectureSight values to camera values. ONVIF's internal values for pan range from -1 to 1. This minimum value is mapped to that range.
cv.lecturesight.onvif.camera.pan.max	1700	The maximum pan value.
cv.lecturesight.onvif.camera.pan.maxspeed	100	The maximum speed for changing the pan value, internally mapped to 1.
cv.lecturesight.onvif.camera.tilt.min	-200	The minimum tilt value. Tilt value range is -1 to 1.
cv.lecturesight.onvif.camera.tilt.max	900	The maximum tilt value.
cv.lecturesight.onvif.camera.tilt.maxspeed	100	The maximum speed for changing the tilt value, internally mapped to 1.
cv.lecturesight.onvif.camera.zoom.min	1	The minimum zoom level value. Zoom range is 0 to 1.
cv.lecturesight.onvif.camera.zoom.max	9999	The maximum zoom level value.
cv.lecturesight.onvif.camera.zoom.maxspeed	10	The maximum speed for changing the zoom level, internally mapped -1 to 1.
cv.lecturesight.onvif.updater.interval	80	The interval in ms to request position updates.

WSDL

The Web Service Definition Language (WSDL) for the different versions and devices:

API	Version	WSDL
Device Management	1.0	http://www.onvif.org/ver10/device/wsdl/devicemgmt.wsdl
	2.0	http://www.onvif.org/ver20/ptz/wsdl/ptz.wsdl
Media	1.0	http://www.onvif.org/ver10/media/wsdl/media.wsdl
	2.0	http://www.onvif.org/ver20/media/wsdl/media.wsdl
PTZ	1.0	http://www.onvif.org/onvif/ver10/ptz/wsdl/ptz.wsdl
	2.0	http://www.onvif.org/ver20/ptz/wsdl/ptz.wsdl
All	2.0	http://www.onvif.org/onvif/ver20/util/operationIndex.html

4.6.5 ONVIF Library

This camera implementation is based around the ONVIF wrapper classes written by Milgo and available on GitHub at: https://github.com/milg0/onvif-java-lib.

The onvif-java-lib is deployed under the Apache License, Version 2.0 of January 2004.

4.6.6 VISCA Camera

The `wulff-visca-service` bundle provides a driver for cameras speaking the serial VISCA protocol defined by Sony.

On activation the service tries to initialize all VISCA cameras on the configured serial device.

Upon discovery of a VISCA camera, the driver determines camera vendor and model and tries to load a matching device profile. If no matching profile exists, the driver loads a default profile that has the same configuration as the profile for the Sony EVI-D30. Most VISCA cameras can be configured to run in D30 compatibility mode.

Configuration

Key	Default	Description
com.wulff.lecturesight.visca. port.device	/dev/tty S0	Serial device for camera communication
com.wulff.lecturesight.visca. port.speed	9600	Serial device speed
com.wulff.lecturesight.visca. port.databits	8	Serial device data bits
com.wulff.lecturesight.visca. port.parity	none	Serial device parity
com.wulff.lecturesight.visca. port.stopbits	1	Serial device stop bits
com.wulff.lecturesight.visca. updater.interval	100	Interval in ms at which camera position is requested
com.wulff.lecturesight.visca. updater.poll.focus	false	If true, report camera focus setting periodically.

Camera Profiles

Camera profiles are defined in `wulff-visca-service/src/main/resources/profiles` for these cameras:

- Sony D70
- Sony H100S
- Sony H100V
- Sony SGR-300H
- Vaddio ClearVIEW HD-USB

The following is an example of a camera profile definition. It is the default camera profile, which is why the values for camera.vendor.id and camera.model.id are set to DEFAULT. In actual camera profiles, the values are numeric (byte) values.

```
camera.vendor.id=DEFAULT
camera.vendor.name=ACME Inc.
camera.model.id=DEFAULT
camera.model.name=Unknown Model
camera.pan.min=-32767
camera.pan.max=32767
camera.pan.maxspeed=18
camera.tilt.min=-32767
camera.tilt.max=32767
camera.tilt.maxspeed=17
camera.zoom.min=0
camera.zoom.max=65535
camera.zoom.maxspeed=7
camera.home.pan=0
camera.home.tilt=0
```

4.6.7 Absolute Move Steering Worker

4.6.8 Steering Worker

The `lecturesight-steeringworker-relativemove` bundle provides the Relative Move Camera Steering Worker, which is responsible for moving the camera.

The steering worker is given a target position, monitors the camera position and decides whether to move the camera and at what speed. The target position is updated by the Camera Operator, or can be set via console command or in the PTZ Camera Control window.

In order to produce smooth camera movements, the steering worker will gradually decrease the speed of the camera movement as it gets closer to the target. Also if the target position is already near the actual position of the camera the steering worker will produce slow correction moves.

Under a certain distance the steering worker will not produce any correction moves which compensates for slightly oscillating targets.

Configuration

Scene limits, alpha and stop values are specified in the PTZ camera co-ordinate system.

Key	Default	Description
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.left		Left-most limit of the scene from the overview image
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.right		Right-most limit of the scene from the overview image
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.top		Top-most limit of the scene from the overview image
cv.lecturesight.ptz.steering.worker.relativemove.scene.limit.bottom		Bottom limit of the scene from the overview image
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.pan	1.0	Maximum pan speed(0 to 1)
cv.lecturesight.ptz.steering.worker.relativemove.move.damp.tilt	1.0	Maximum tilt speed(0 to 1)
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.x	400	Horizontal region in which the pan speed should decrease
cv.lecturesight.ptz.steering.worker.relativemove.move.alpha.y	400	Vertical region in which the tilt speed should decrease
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.x	10	Stop pan movement when camera is within this horizontal distance of the target
cv.lecturesight.ptz.steering.worker.relativemove.move.stop.y	10	Stop tilt movement when camera is within this vertical distance of the target
cv.lecturesight.ptz.steering.worker.relativemove.xflip	false	Flip x co-ordinates if required when camera is mounted inverted.
cv.lecturesight.ptz.steering.worker.relativemove.yflip	false	Flip y co-ordinates if required when camera is mounted inverted.
cv.lecturesight.ptz.steering.worker.relativemove.move.initial.delay	2500	Initial delay in ms after setting initial position before starting to move.
cv.lecturesight.ptz.steering.worker.relativemove.focus.fixed	false	If true, disable auto-focus after setting initial position.
cv.lecturesight.ptz.steering.worker.relativemove.autostart	true	Controls if the camera steering is active when the module is started.

Console Commands

Command	Description
cs:on	Activates the camera steering.
cs:off	Deactivates the camera steering.
cs:move	Make the camera move the specified pan and tilt coordinates. The coordinates are normalized in a value range between -1.0 and 1.0, where -1 is left / bottom and 1.0 is right / top.
cs:zoom	Makes the camera set the specified zoom. The zoom value is normalized to a value range of 0.0 to 1.0. where 0 is neutral and 1 is maximum zoom.
cs:home	Makes the camera move to its home position. For most models this will be pan=0.0 and tilt=0.0.

4.6.9 Video Analysis

The `lecturesight-videoanalysis-templ` bundle performs template-based video analysis and provides the *ObjectTracker* service.

The tracker will follow a maximum of 6 targets, using template-matching between frames to maintain persistence. New targets are identified when cells in the overview image exceed defined change thresholds. Targets are dropped when they are inactive for a defined period.

The tracker provides the list of active targets to the Camera Operator which is responsible for selecting one or more targets to follow. The Object Tracker display shows the targets being tracked.

Configuration

Key	Default	Description
cv.lecturesight.videoanalysis.change.threshold	48	Pixel color change threshold. A pixel has changed if the summed difference in all color channels of the pixel in two consecutive frames exceeds this value.
cv.lecturesight.videoanalysis.cell.activation.threshold	3	Threshold for when a cell is activated. If there are more than this number of changed pixels in a cell, the cell is activated. A cell is 8x8 pixels.
cv.lecturesight.videoanalysis.object.cells.min	2	Minimum number of active cells in a cluster to be considered a tracking target.
cv.lecturesight.videoanalysis.object.cells.max	128	Maximum number of cells that a tracking target may consist of.
cv.lecturesight.videoanalysis.object.match.threshold	15	Template match score must exceed this value for the object at the template location to be considered the same object between successive frames.
cv.lecturesight.videoanalysis.object.move.threshold	3	Movement threshold: the target is considered to have moved between two successive frames if the distance between positions exceeds this value.
cv.lecturesight.videoanalysis.object.dormant.min	400	Minimum time in ms that a tracking target may be dormant before it is discarded from the list of targets.
cv.lecturesight.videoanalysis.object.dormant.max	8000	Maximum time in ms that a static tracking target may be dormant before it is discarded from the list of targets.
cv.lecturesight.videoanalysis.object.dormant.age.factor	0.5	Scaling factor that increases the dormant timeout value as the object ages. Older objects have a higher timeout value.
cv.lecturesight.videoanalysis.object.active.min	500	Minimum time in ms that a tracking target must be active before it is included in the target list provided to the camera operator.
cv.lecturesight.videoanalysis.object.timeout	60000	Maximum timeout in ms for target. This timeout is applied when the target has moved at least a certain distance from its original position.

Template-Matching Video Analysis uses the *ConnectedComponents* service, which has the following configuration options:

Key	Default	Description
cv.lecturesight.blobfinder.blobs.max	100	The maximum number of blobs that an instance of a BlobFinder can work with. This value affects the size fo several GPU buffers. Thus this may help to solve memory shortages on older GPUs.
cv.lecturesight.blobfinder.blobsize.min	20	Default minimum size (in pixels) of a valid blob. This value is usually set by the consumer when instantiating a BlobFinder.
cv.lecturesight.blobfinder.blobsize.max	10000	Default maximum size (in pixels) of a valid blob. This value is usually set by the consumer when instantiating a BlobFinder.

Commands

Command	Description
va:reset	Clears the target list and resets the tracker.

4.7 Modules

4.7.1 PTZ Camera Control

The **PTZ Camera Control** display is used for verifying camera calibration and movement, and optimising camera movement when tracking.

Verifying Calibration

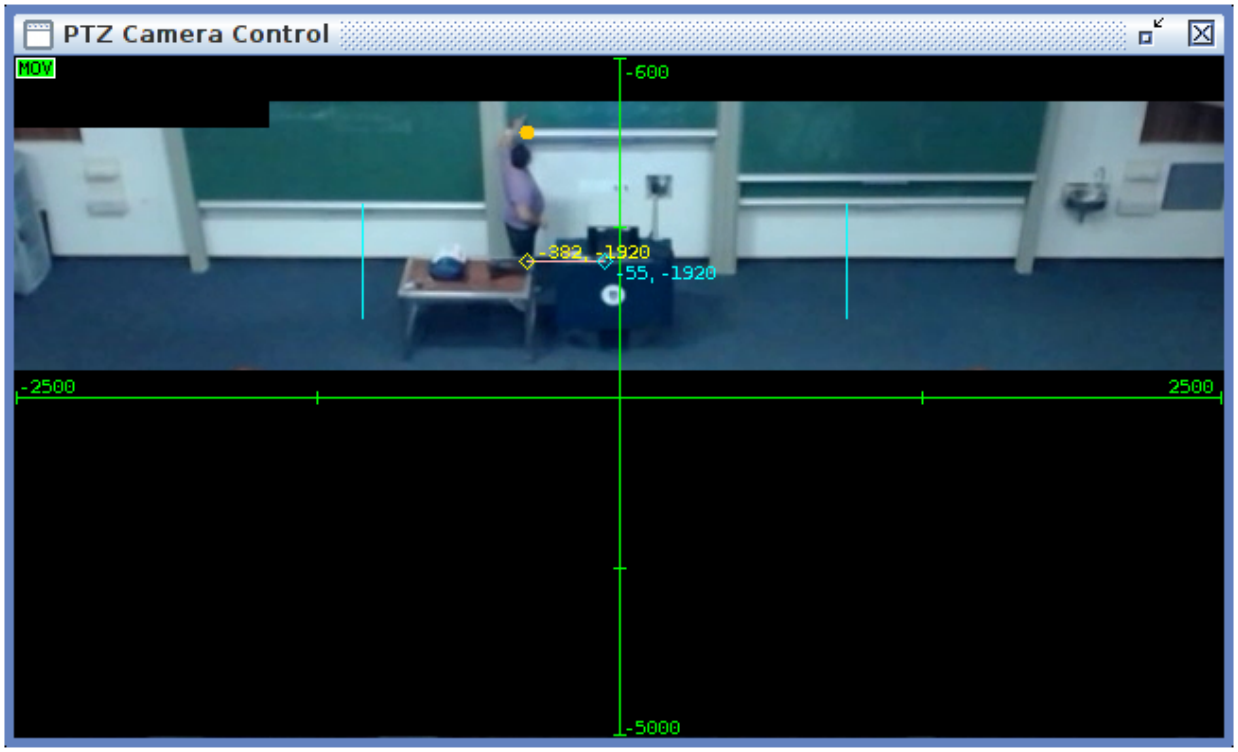

Fig. 3: PTZ Camera Control

The cyan lines show the frame width of the PTZ Camera's field of view at the configured zoom level.

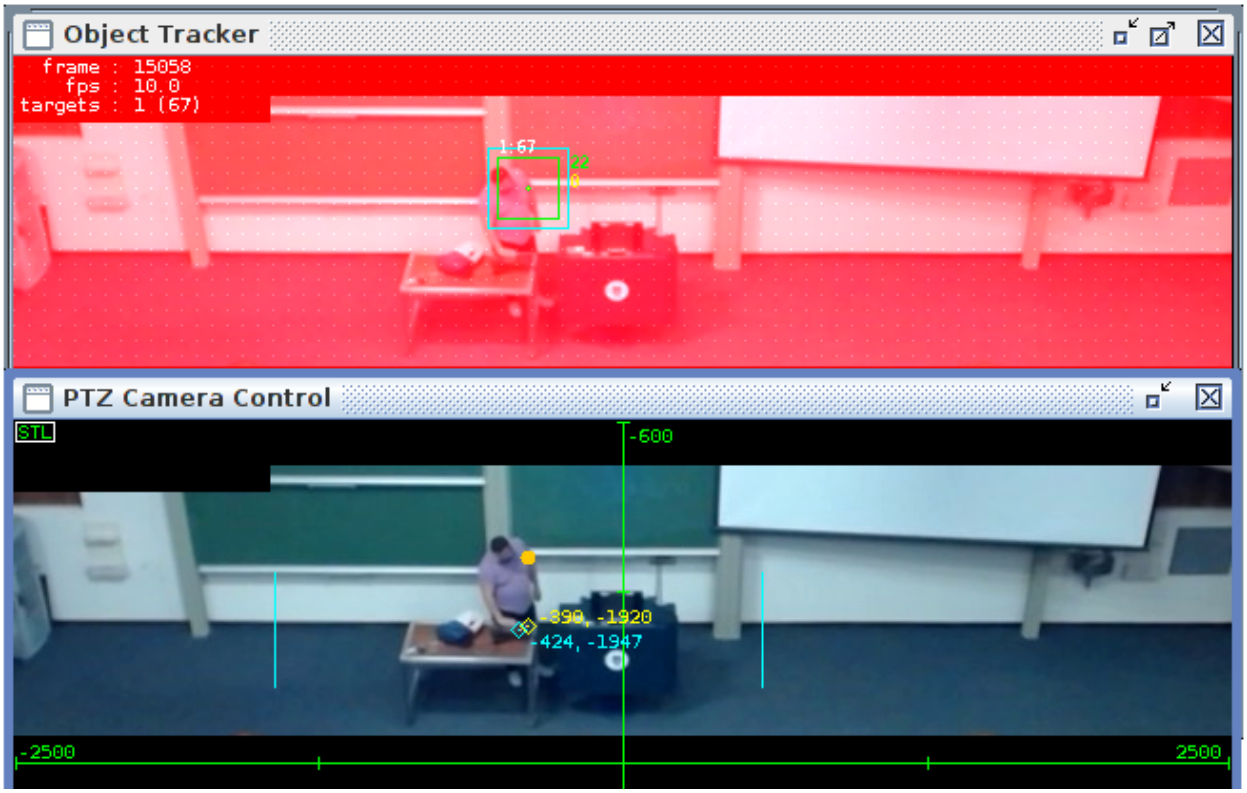

Fig. 4: PTZ Camera Control and Object Tracker

Optimising tracking performance

The orange dot shows the target position provided by the Steering Worker by the Camera Operator. The yellow diamond shows the target position for the PTZ Camera, and the cyan diamond shows the actual camera position.

4.7.2 System Configuration

The **System Configuration** table can be used to modify system configuration properties. Edit the values by clicking on them and then typing a new value.

Some configuration parameters, including those for video analysis, are interactive in that changes on them take effect as soon as you hit enter after editing a values. Other properties such as the path to the overview camera video device, are used only at system start.

4.7.3 Services Menu

The **Services** menu provides access to the LectureSight displays and configuration windows.

If not all menu items are shown, it may indicate that some bundles have not started correctly (for example as a result of configuration issues).

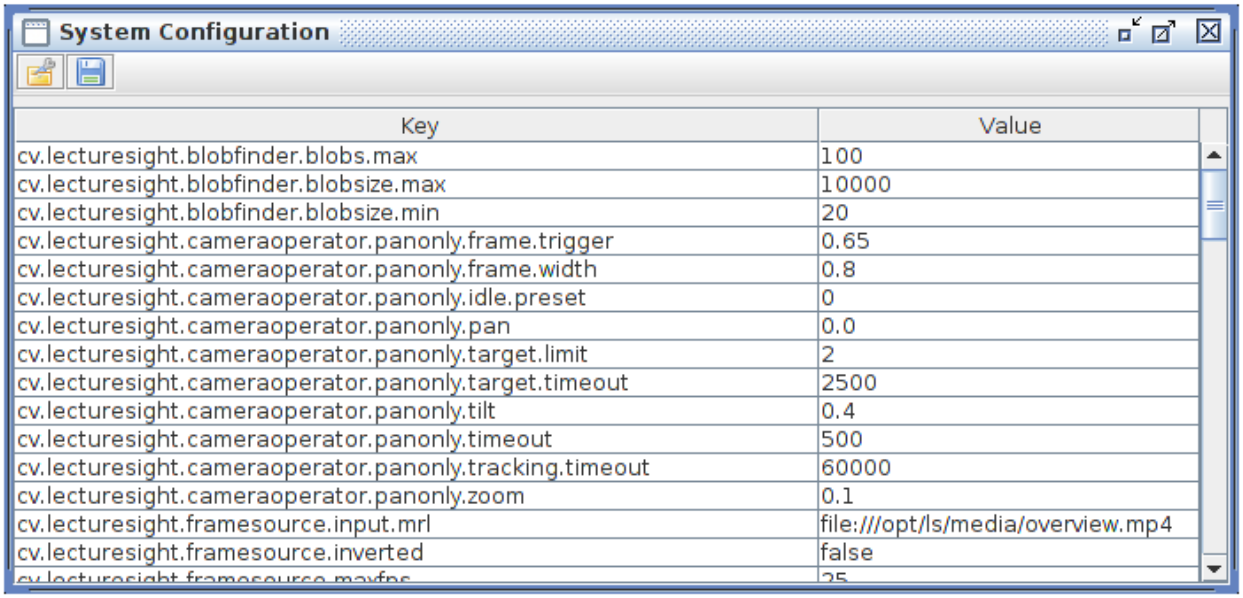

Fig. 5: System Configuration

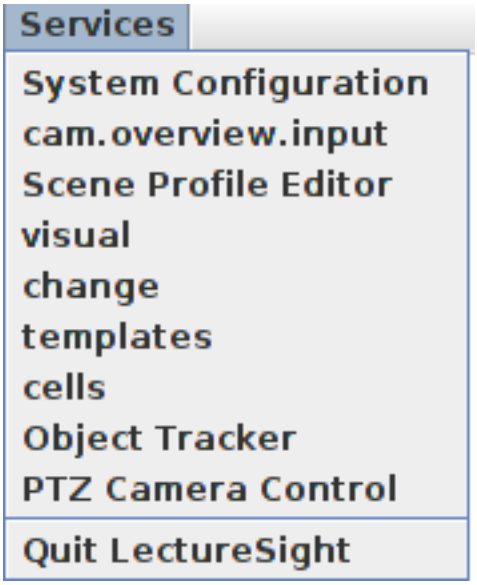

Fig. 6: Services Menu

4.7.4 Object Tracker

The **Object Tracker** display is produced by the Video Analysis module, and shows the targets currently identified for possible camera tracking.

Single Target

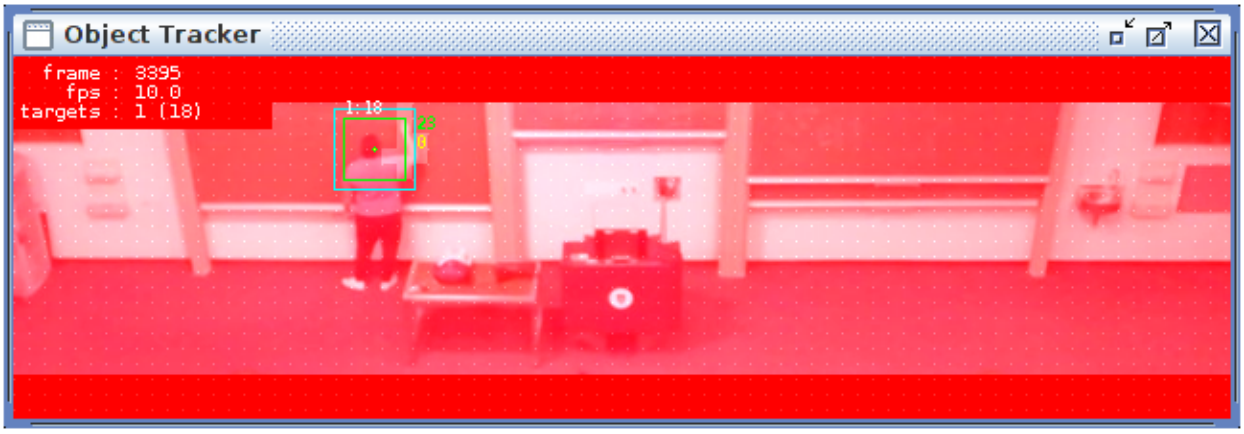

Fig. 7: Object Tracker showing single target

Multiple Targets

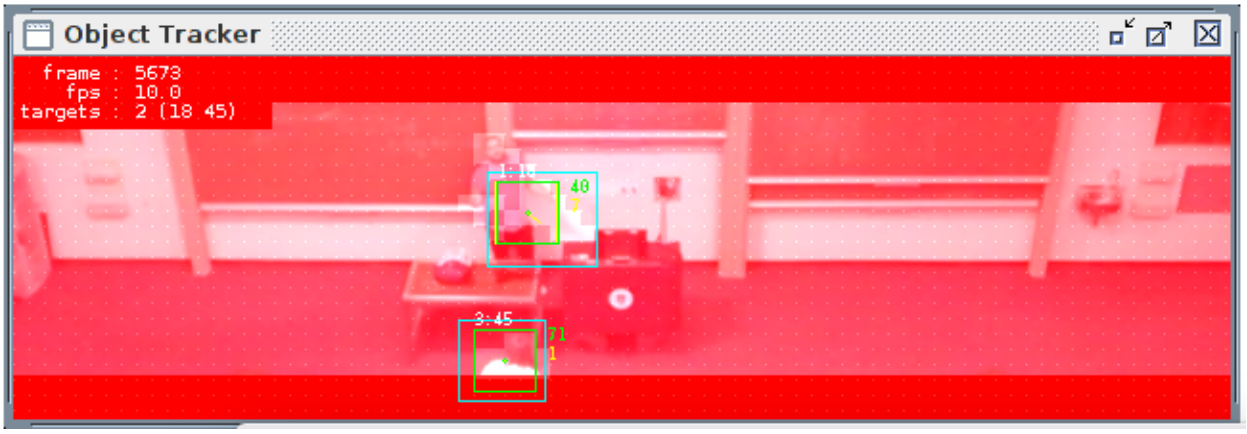

Fig. 8: Object Tracker showing two targets

Other displays

The Video Analysis module also produces four other displays:

- **visual**
- **change**
- **cells**

- **templates**

4.7.5 Overview Camera

The Overview Camera display is shown on the Services menu as **cam.overview.input**. It shows the input from the overview camera with the active scene profile mask applied (ignore masks are shown as black regions).

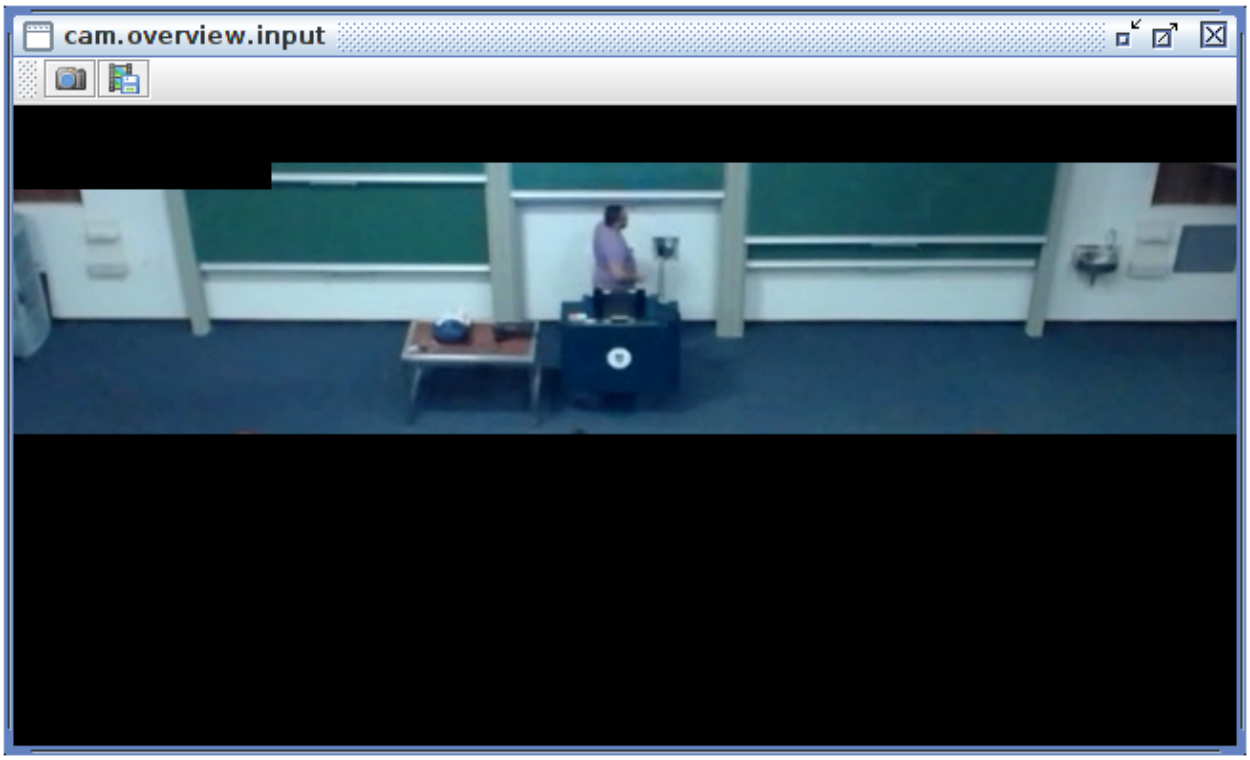

Fig. 9: Overview Camera display

4.7.6 Scene Profile Editor

The **Scene Profile Editor** allows zones and points to be defined which affect the video analysis and camera calibration.

Ignore Zones

Ignore Zones are masked out from the input frame before video analysis. Add ignore regions to mask areas of the overview input that are not of interest for tracking, for example audience seating, doorways, or projector screens.

Calibration Markers

Calibration Markers identify specific points on the overview image. These can be used for marker calibration, or for later reference (for example to verify that the overview camera has not moved since installation).

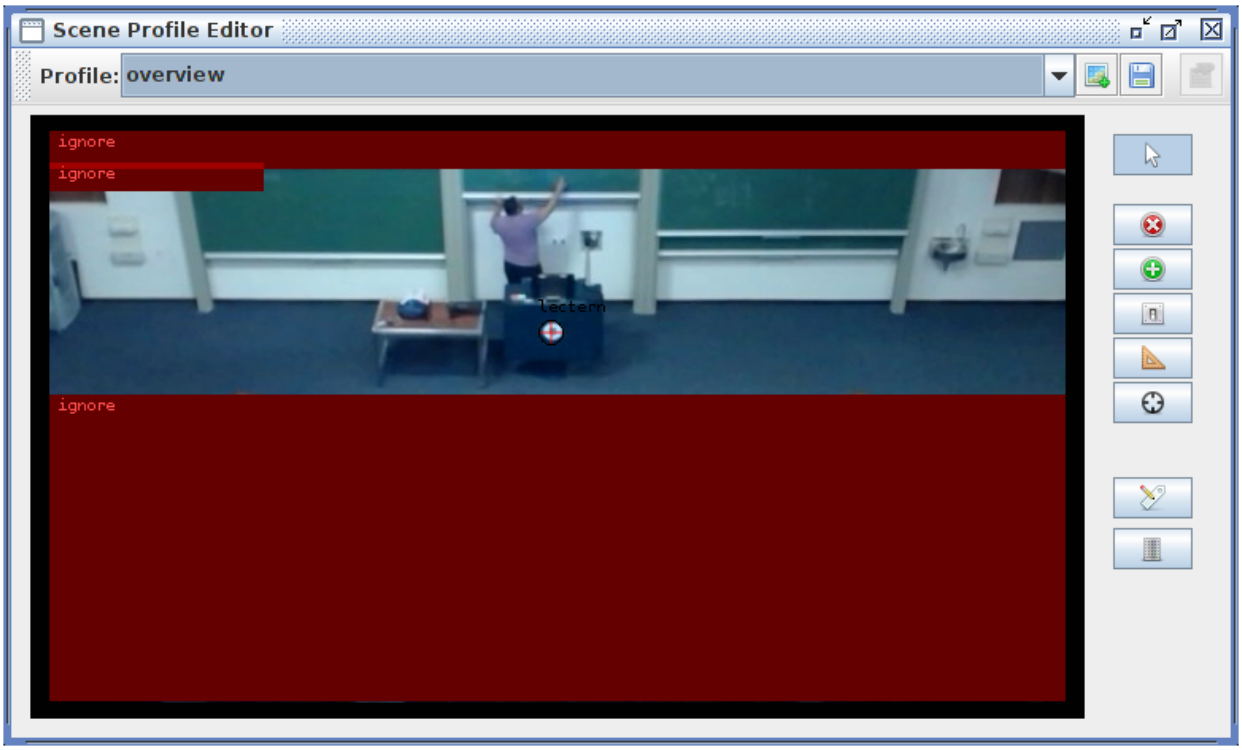

Fig. 10: Scene Profile Editor

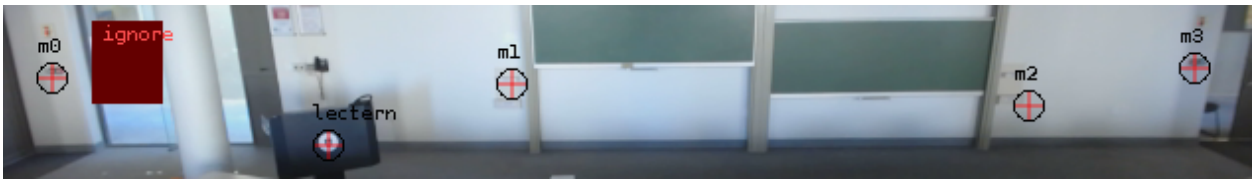

Fig. 11: Calibration markers

Other zone types

Tracking Zones, **Trigger Zones** and **Measure Zones** are not presently used by LectureSight.

Create profile

Note that the default profile cannot be saved, so to use a custom profile, create a new profile first.

When you are done creating the profile, click the **Save** button to save the profile to a file in the `profiles` directory. Every time you save a profile or choose one from the dropdown, the profile will be activated so you can see how the system performs when the profile is active.

To use your profile permanently, set the name of the active profile to use in `lecturesight.properties`

```
cv.lecturesight.profile.manager.active.profile=myprofile
```