1. LectureSight 0.3-rc2 . . . 2
2. LectureSight Project Home . . . 6
2.1 Development . . . 13
2.1.1 Camera Operator (bak) . . . 13
2.1.2 LectureSight UI - first mockup . . . 14
2.2 Operation . . . 14
2.2.1 Getting Started . . . 14
2.2.1.1 1. Prerequisites . . . 14
2.2.1.1.1 1.1. Hardware . . . 14
2.2.1.1.2 1.2. Software . . . 15
2.2.1.2 2. Installation . . . 16
2.2.1.3 3. Configuration . . . 16
2.2.1.3.1 3.1. Overview Camera . . . 17
2.2.1.3.2 3.2. PTZ-Camera . . . 18
2.2.1.3.3 3.3. Calibration . . . 18
2.2.2 Hardware Components . . . 21
2.2.2.1 Camera Setup . . . 21
2.2.2.2 Overview Cameras . . . 22
2.2.2.3 PTZ Cameras . . . 23
2.2.3 Service Bundle Manuals . . . 23
2.2.3.1 Camera Steering Worker . . . 23
2.2.3.2 Decorator Color Histrogram . . . 26
2.2.3.3 Decorator Head . . . 26
2.2.3.4 Foreground Region Tracker . . . 27
2.2.3.5 Frame Source Manager . . . 28
2.2.3.6 Heart Beat . . . 29
2.2.3.7 iCal Scheduler . . . 30
2.2.3.8 Object Tracker . . . 31
2.2.3.9 Simple Camera Operator . . . 32
2.2.3.10 Video4Linux Frame Source . . . 33
2.2.3.11 Video Analysis Services . . . 34
2.2.3.12 Video File Frame Source . . . 36
2.2.3.13 VISCA Camera Driver . . . 37
2.3 Quick start guide LectureSight 0.3 . . . 38
2.4 Research . . . 42
2.4.1 Bibliography . . . 42
2.4.2 Other technologies, solutions or projects . . . 42
2.4.3 Publications . . . 43
3. Final 0.3 Release Sprint . . . 43
3.1 LectureSight Server - Development Overview . . . 45

LectureSight 0.3-rc2

Download LectureSight Unknown macro: 'button'

Requirements

The following minimal requirements need to be fulfilled in order to properly run LectureSight:

- CPU: Core2 Duo 2.4 GHz
- GPU: OpenCL 1.1 compatible
- Disk free: 20 MB
- Overview Camera: see list of tested models
- Operating System: Linux (Mint/Ubuntu tested)
- Java: Java 7 (Oracle)
- GStreamer when using video files as input (see Video File Framesource Manual)

See section 1.1 Hardware for an in depth discussion on the hardware requirements.

Installation

Issue with OpenCL on Ubuntu 14.04 / Linux Mint 17
There has been an issue with the ICD Loader installed with the NVIDIA driver version 311.38 on Ubuntu 14.04 / Linux Mint 17, that prohibited LectureSight from finding the OpenCL platform. The issue can be fixed by installing the package "libopencl-1.2-1" or by using the driver update package "nvidia-331-upadtes".

1. Get the installation package by clicking the Download button above or from command line via

```
>  wget http://opencast.jira.com/wiki/download/attachments/62914604
/lecturesight-0.3-rc2-x86_64.tar.gz
```

2. Extract the archive e.g. to **/opt**

```
> tar -vzxf lecturesight-0.3-rc2-x86_64.tar.gz -C /opt
```

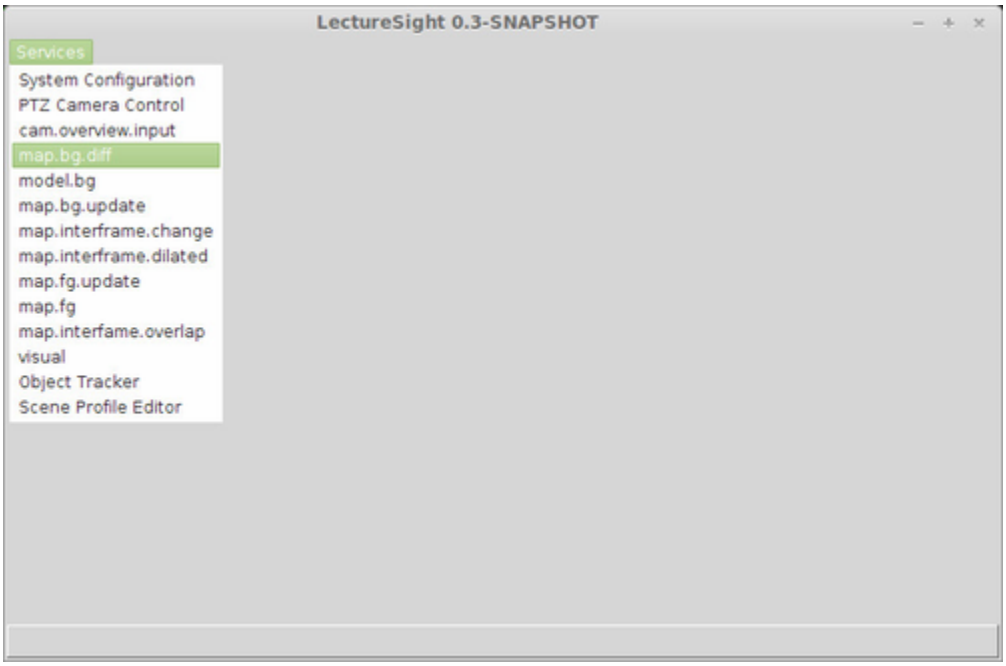
3. Edit the file **conf/lecturesight.properties** and set the correct paths to your overview camera device and the serial device to which your VISCA camera is connected, also adjust the resolution for the overview camera

```
#
# Initial configuration for LectureSight
#

cv.lecturesight.framesource.input.mrl=v4l:///dev/video0[width=640;
height=480]

cv.lecturesight.ptz.visca.ports=/dev/ttyUSB0

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

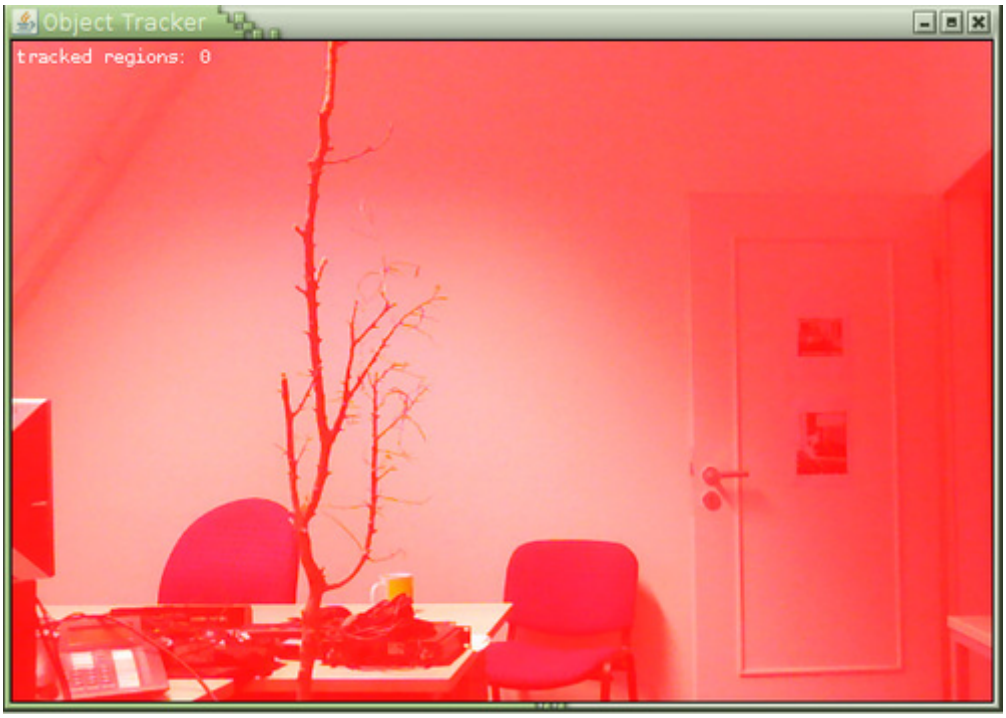
4. After the initial configuration, start LectureSight via the start script

```
> ./start_lecturesight
```

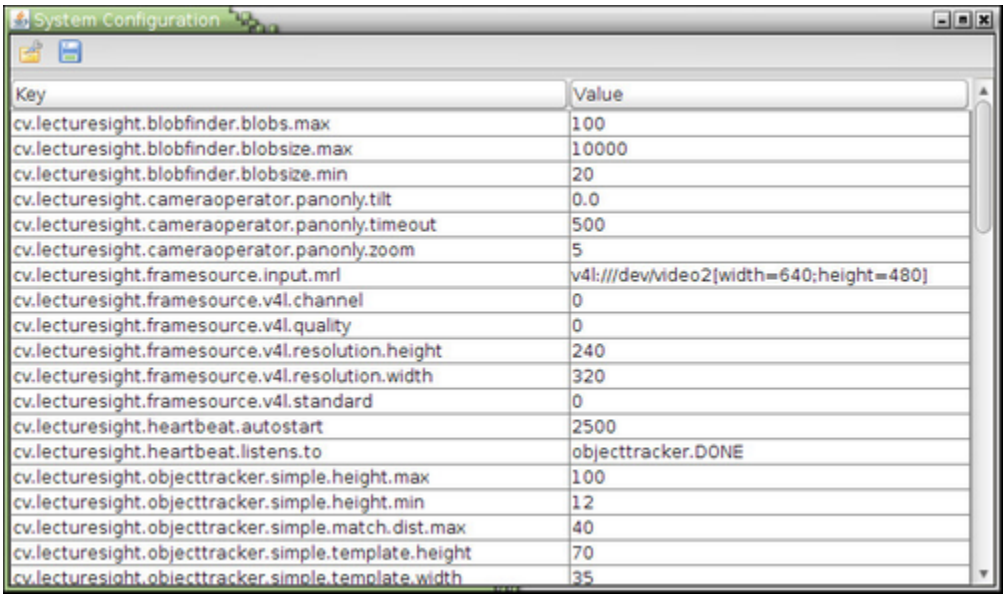
Configuration



After LectureSight has started the main window will come up (see picture above). In the upper left corner there is a menu called "Services", it holds all the available user interfaces. Choose the menu option "Object Tracker" to bring up the object tracker visualization and see if the system is already working. If you can not find the menu item this means that the system has not been initialized successfully. If the system is working you will the overview camera view with a red tint in the Object Tracker window.



You can access every system configuration property via the "System Configuration" window. The list provides the configuration keys and values. You can edit the values by clicking on them and then typing a new value. Some configuration parameters, like e.g. those for the video analysis, are interactive in that changes on them take effect as soon as you hit enter after editing a values. Other properties, like e.g. the path the to overview camera video device, are used only at system start.



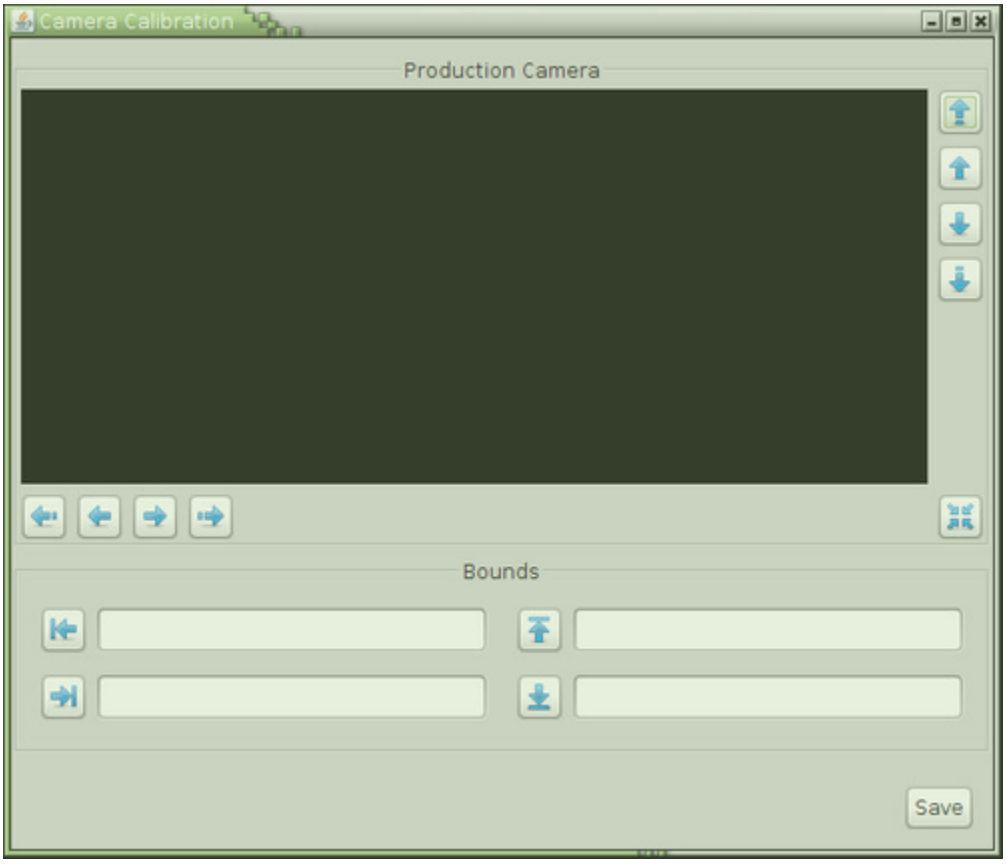
Key	Value
cv.lecturesight.blobfinder.blobs.max	100
cv.lecturesight.blobfinder.blobsize.max	10000
cv.lecturesight.blobfinder.blobsize.min	20
cv.lecturesight.cameraoperator.panonly.tilt	0.0
cv.lecturesight.cameraoperator.panonly.timeout	500
cv.lecturesight.cameraoperator.panonly.zoom	5
cv.lecturesight.framesource.input.mrl	v4l:///dev/video2[width=640;height=480]
cv.lecturesight.framesource.v4l.channel	0
cv.lecturesight.framesource.v4l.quality	0
cv.lecturesight.framesource.v4l.resolution.height	240
cv.lecturesight.framesource.v4l.resolution.width	320
cv.lecturesight.framesource.v4l.standard	0
cv.lecturesight.heartbeat.autostart	2500
cv.lecturesight.heartbeat.listens.to	objecttracker.DONE
cv.lecturesight.objecttracker.simple.height.max	100
cv.lecturesight.objecttracker.simple.height.min	12
cv.lecturesight.objecttracker.simple.match.dist.max	40
cv.lecturesight.objecttracker.simple.template.height	70
cv.lecturesight.objecttracker.simple.template.width	35

System configurations can be loaded and saved using the buttons in the toolbar.

⚠ Every bundle in LectureSight comes with it's default configuration. If you set a property value in the UI that differs from the default configuration that comes with the bundle, then this property will be written to the properties file when you choose to save the configuration. This way the properties you will find in the configuration file reduce to those that you have touched in your configuration.

Camera Calibration

The most important configuration step is the camera calibration. It is needed in order for the system to correctly map coordinates from the overview camera to movement angles of the production camera. From the "Services" menu choose "PTZ Camera Calibration". The automatic camera control is deactivated as long as the calibration window is up.



If you have your production camera also connected to the machine as a video device, you can use it directly in the calibration UI. Add the following line to **conf/lecturesight.properties**

```
cv.lecturesight.setup.video.input=v4l:///dev/video1[width=640;
height=480]
```

If you don't have the production camera connectedm, the calibration UI will still work but the production camera view will stay black.

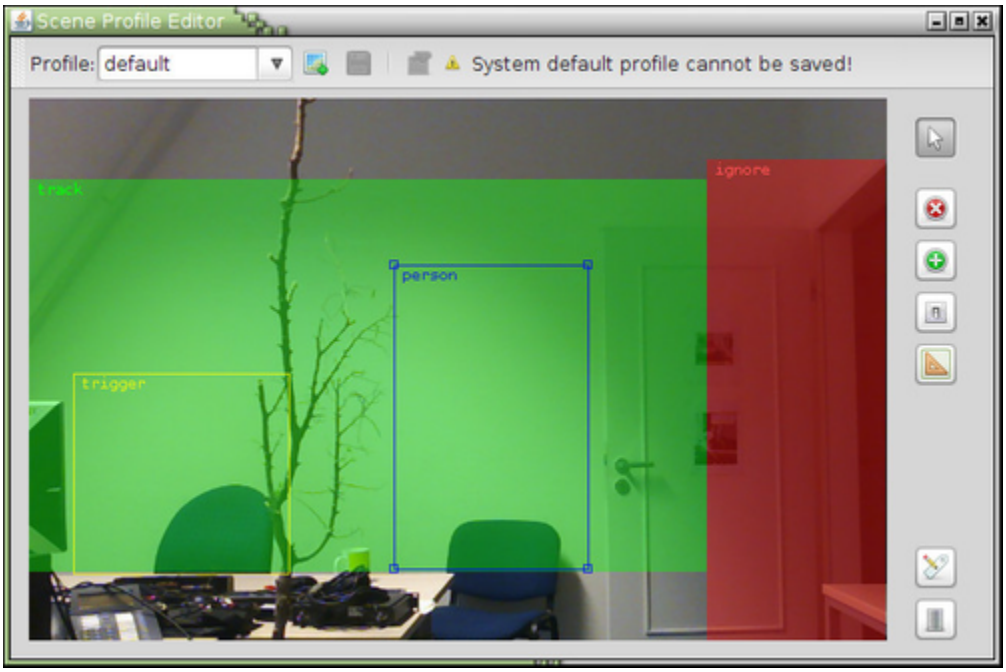
For the calibration we need to know, and tell the system, which coordinates from the PTZ camera correspond to the limits of the overview camera image. Look for significant points at the each side of the overview camera image and try to drive the production camera to those points. You can either use the remote control of the production camera or you can steer the camera from within the user interface by using the buttons at the right and below the production camera view. The button in the lower right corner will make the camera move to the center. When you have driven the camera to a limit click the corresponding button in the "Bound" frame. The current value from the production camera will shown up in the text field next to the button. Repeat the procedure for each limit. When you have the values for all four scene limits, click the "Save" button to apply the values.

⚠ By clicking "Save" in the calibration UI you apply the values to the system configuration. To save the system configuration permanently you have to use the "Save" button in the "System Configuration" window after you have applied the values.

Scene Configuration

The system can be customized to certain presentation site by setting areas in the overview image that are ignored. This is especially useful in rooms where the audience is visible in the overview image. Masking out parts of the scene that are not relevant is likely to increase performance

since it keeps the system from tracking persons that are not of interest for the recording. You can access an interactive editor the the scene profile by choosing "Scene Profile Editor".



Click on the icon next to the profile selection box to create a new profile. You will be asked for a screen name and the name of a file to save the profile to. With the buttons on the right you can choose which type of area to create. The following types of areas can be defined:

- **Ignore** - Those are areas that are completely neglected by the system. Define them especially in parts of the overview image in which there will be motion (e.g. people in the audience) that is not of interest for the camera control. This will optimize performance of the video analysis.
- **Tracking** - (not implemented yet)
- **Trigger** - Trigger zones emit events when tracked objects enter of leave them. The camera operator script can then act on those events.
- **Person** - (experimental feature) With a person zone, you can tell the system the average size of a person in the scene. When there is a person zone defined in the profile, the system will adjust certain parameters of the video analysis accordingly, when the profile is activated.

In LectureSight 0.3-rc2 you only need to define the ignore and the person areas. As a rule of thumb, mask out all parts of the scene in which tracking is not of interest, leave only the area in which you want the presenter to be tacked without ignore areas. Also, by defining the person area, you can help the system discriminate between persons and other moving object.

Finally, when you are done creating the profile, click the "save" button to safe 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. When you want the system to use your profile permanently, set the name of the profile to use for the following property

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

LectureSight Project Home

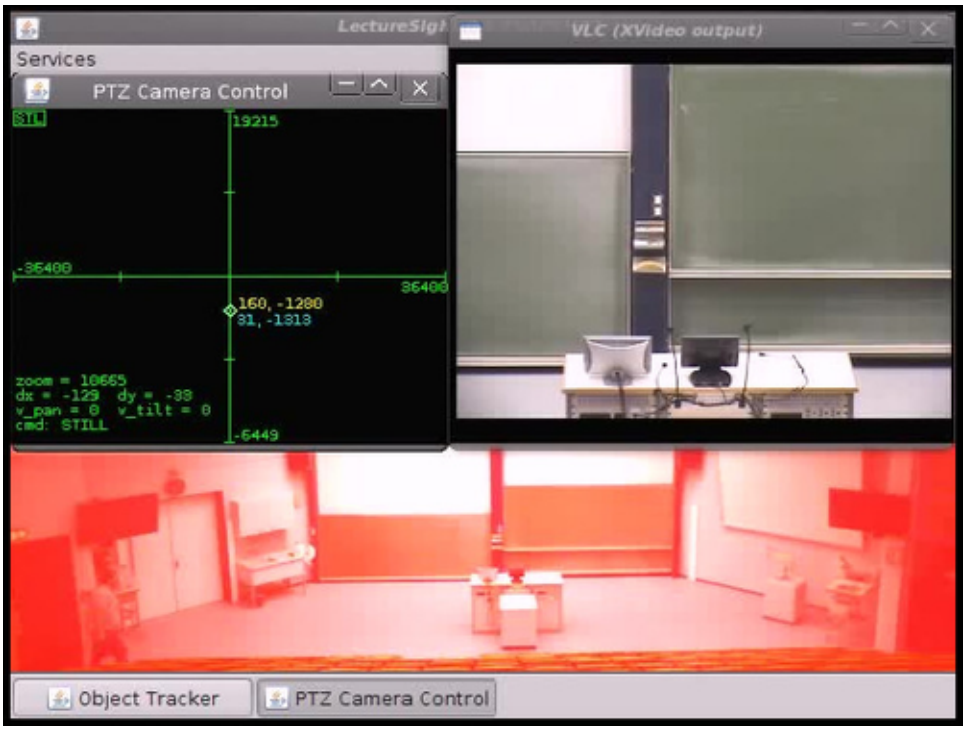
Welcome to the LectureSight project!

This site is dedicated to research and development around an open source camera tracking solution for use in the classroom (and beyond...). The software is being developed under the umbrella of the Opencast Community and in accordance with the Opencast code conventions.

Mailing List

lecturesight@googlegroups.com

Subscribe by sending a mail to:



Reference documentation is now available at docs.lecturesight.org.

This Wiki is organized into three main sections that mirror the main topics we take on in this project.

- Research - This section is devoted to the academic aspects of the project. Here you can find, publications, thoughts and links around the research on this project and computer vision in general.
- Development - This section holds all the development documentation of the project. It is the central place where developers can find information needed to develop modules for LectureSight.
- Operation - This section holds tutorials on how to operate LectureSight and detailed manuals for each module of the system. This is the place where users should go to get started.

Space Index

0-9 ... 8	A ... 0	B ... 1	C ... 3	D ... 3	E ... 0
F ... 3	G ... 1	H ... 2	I ... 1	J ... 0	K ... 0
L ... 4	M ... 0	N ... 0	O ... 4	P ... 2	Q ... 1
R ... 1	S ... 2	T ... 0	U ... 0	V ... 4	W ... 0
X ... 0	Y ... 0	Z ... 0	!@#$ ... 0		

0-9

1. Prerequisites

LectureSight is a software build up of modules for the OSGI software modularization framework [All07 http://opencast.jira.com/wiki/node30.html#alliance2007osgi] for the Java programming language. Java itself is platform-independent but LectureSight relies

1.1. Hardware

As mentioned, as the host system for LectureSight a fairly modern personal computer that is capable

A

lecturesight+subscribe@googlegroups.com

Directions

Project Homepage: www.lecturesight.org

Documentation: docs.lecturesight.org

Issue Tracker: issues.lecturesight.org

Code Repository: code.lecturesight.org

Loading

Twitter Widget

Widget

#46090407081

1615232

Currently on our plate

Key	Summary
LS-234	Unify Relative and Absolute Move Steering Workers
LS-91	Make OpenCL Executor flush task queue and clFinish() command queue
LS-83	Implement Scripting APIs
LS-16	Allow for the admin to adapt camera velocity in the UI

4 issues

of running Linux is sufficient. CPU LectureSight was designed with the goal of being capable of running on a single 2 GHz core. The system should not take

1.2. Software

Operating System In principal every modern Linux distribution can serve as host operating system. A kernel supporting Video4Linux or Video4Linux2 is the only requirement for the video analysis. LectureSight has already been tested with the following distr

2. Installation

The pre-compiled binary distribution for production systems can be downloaded here lecturesight-0.3-rc1.tar.gz Extract the software e.g. to /opt/ tar -xfvz lecturesight-0.3-rc1.tar.gz mv lecturesight-0.3-rc1 /opt Change into the lecturesight directory and

3. Configuration

The whole system configuration is loaded from a Java properties file that, by default, is conf/lecturesight.properties. You can edit this file with a text editor or directly in the LectureSight GUI (Menu: Services > System Configuration) where you can als

3.1. Overview Camera

By default LectureSight will try to use /dev/video0 as the overview camera with a resolution of 320x240 pixels. This is only the default resolution that most web-cams should be capable of. In operation, however, a resolution around VGA (640x480 or so) sho

3.2. PTZ-Camera

Out of the box there is no PTZ-camera configured in LectureSight, though the driver for cameras speaking Sony's VISCA protocol is shipped with the system. You can activate the driver by setting the configuration property cv.lecturesight.ptz.visca.ports an

3.3. 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. First we have to tell the PTZ camera where in it's field of operation the bound of the scene lie that the overview c

B

Bibliography

C

Camera Operator (bak)

Hoshen and R. Kopelman. Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm. Physical Review B, 14(8):3438, 1976 O. Kalentev, A. Rai, S. Kemnitz, and R. Schneider. Connected component labeling

This page describes how the camera operator works. The camera operator basically interprets data sent by a "tracker" and translates into commands for the PTZ-Camera through the "camera-control". The camera operator is just a part of the OpenTrack system.

Camera Setup

In the current version the overview camera and the PTZ camera must be mounted in the same location. camera_setup_scaled.png

Camera Steering Worker

Bundle File: lecturesight-steeringworker-impl.jar The bundle provides the so called "Camera Steering Worker". It is responsible for moving the camera. The steering worker is given a target position and is constantly monitoring the cameras position and d

D

Decorator Color Histrogram

Bundle File: lecturesight-decorator-color.jar The bundle provides a service that implements an ObjectDecorator. For object that fit configurable size criteria, the service creates and updates color histograms. The number of bins of the histograms can be

Decorator Head

Bundle File: lecturesight-decorator-head.jar The bundle provides a service that implements an ObjectDecorator. It purpose is to find the head of a tracked person. A K-means clustering is performed on the foreground pixels of a given object creating a Vo

Development

E

F

Final 0.3 Release Sprint

The University of Manchester supports the current sprint towards the final 3.0 release version of LectureSight. Development activity during the sprint will focus on the over-all stability of the software and the robustness of the camera tracking. The aime

Foreground Region Tracker

Bundle File: lecturesight-regiontracker-impl.jar The bundle provides the Foreground Region Tracker Service. It tracks regions in the foreground model by mapping

G

Getting Started

This section should serve as a step-by-step guide to setting up and running LectureSight in a productive environment First discussed are the prerequisites of hard- and software. The reader may skip that section completely if the host system has already b

labels produced by the connected component analysis in subsequent frames. The service is als

Frame Source Manager

Bundle File: lecturesight-framesource-impl.jar The FrameSource bundle provides the infrastructure responsible for managing FrameSource implementations. It discovers video input plugins and is responsible for setting up FrameSources with the propper in

H

Hardware Components

Current Test System The current test system that is part of the experimental setup in the lecture hall has following configuration CPU Intel Core 2 Duo RAM 2 GB GPU Nvidia GT 220, 512 MB CAM FaceVision TouchCam N1 OS Ubuntu 10.04

Heart Beat

Bundle File: lecturesight-heartbeat The bundle provides the so called "Heart Beat" 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 hav

I

iCal Scheduler

Bundle File: lecturesight-scheduler.jar The bundle provides a service that loads a schedule from an iCalendar (RFC-2445) file and starts /stopps object tracking and camera control accordingly. Changes to the file are detected and the internal schedule is

J

K

L

LectureSight 0.3-rc2

Download LectureSight Requirements The following minimal requirements need to be fulfilled in order to properly run LectureSight: CPU: Core2 Duo 2.4 GHz GPU: OpenCL 1.1 compatible Disk free: 20 MB Overview Camera: see list of tested models https://open

LectureSight Project Home

Welcome to the LectureSight project! This site is dedicated to research and development around an open source camera tracking solution for use in the classroom (and beyond...). The software is being developed under the umbrella of the Opencast Community h

LectureSight Server - Development Overview

LectureSight UI - first mockup

M

N

O

Object Tracker

Bundle File: lecturesight-objecttracker-impl.jar The bundle provides an Object Tracker Service for tracking and re-discovering of objects. It relies on the Foreground Region Tracker Service and considers moving regions of a reasonable size as objects of

Operation
This section is the source of information about setting up and running LectureSight in the lecture hall or seminar room. Here you can find tutorials for beginners as well as detailed manuals for every functional module in LectureSight. LectureSight runnin

Other technologies, solutions or projects
There are numerous solutions to provide automated tracking in a classroom setting, some of them use radio or IR technology, some ex post analysis of long shot recordings, some have mats on the floor. Here's the place to describe these. Current solution at

Overview Cameras
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. Es

P

PTZ Cameras
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. Mos

Publications
This is a collection of documents around the LectureSight project: Heger, Garry (2010): Open Track. Bachelor's Thesis at HES-SO http://opencast.jira.com/wiki/download/attachments/18383148/garry_heger_open_track_report.pdf?version=1&modificationDate=128203

Q

Quick start guide LectureSight 0.3
Download LectureSight Requirements The following minimal requirements need to be fulfilled in order to properly run LectureSight: CPU: Core2 Duo 2.4 GHz GPU: OpenCL 1.1 compatible Disk free: 20 MB Overview Camera: see list of tested models https://openc

R

Research
This section is dedicated to the academic aspects of the project. Here you can find a list of

S

Service Bundle Manuals
In this section detailed manuals for all bundles that hold implementations of services in

publications about the project as well as a list of scientific papers and books that have influenced the work on the project. Also there is an overview of the ar

LectureSight can be found. The subsections explain the functionalities of those modules and give insight into the inner workings of the services. Further more all co

Simple Camera Operator

Bundle File: lecturesight-cameraoperator-simple.jar The Simple Camera Operator bundle provides a most basic implementation of a camera operator service. When activated it drives the camera into home position and waits for targets. When the object tracke

T

U

V

Video Analysis Services

Bundle File: lecturesight-videoanalysis-impl.jar The bundle provides services for real time video analysis. The implementation is responsible for finding moving objects of a certain size in the scene. In detail the bundle holds three different services

Video File Frame Source

Bundle File: lecturesight-framesource-videofile.jar The bundle provides a FrameSource implementation that reads frames from a video file. It is based on GStreamer. It depends on the set of codecs installed in the host operating systems what formats are su

Video4Linux Frame Source

Bundle File: lecturesight-framesource-v4l.jar The bundle provides a FrameSource implementation for accquiring frames from Video4Linux and Video4Linux 2 devices. Arguments for creation of a new FrameSource from this implementation can be provided in the

VISCA Camera Driver

Bundle File: lecturesight-ptzcontrol-visca.jar The bundle provides a driver for cameras speaking the VISCA protocol defined by Sony. On activation the service tries to initialize all VISCA cameras on a configured serial devices. Upon discovery of a VISC

W

X

Y

Z

!@#$

Development

Camera Operator (bak)

This page describes how the camera operator works. The camera operator basically interprets data sent by a "tracker" and translates into commands for the PTZ-Camera through the "camera-control".

The camera operator is just a part of the OpenTrack system. This page aims to give some technical details about it. For running OpenTrack, refer to "Building and running OpenTrack" Wiki.

How it works

1. The camera operator get a set of TrackedObject from the tracker and must orient the camera to them. This must be done in serveral lecture room sizes and conditions. This is why the system requires a process of calibration (explained further).
2. The tracker delivers many motion information. Some of them could not be human and thus must not be tracked. The camera operator tries to differentiate the tracked objects (based on the size of them).

Calibration process

The calibration process consists of a succession of action the user has to do.The programm called "OpenTrackCalib" is a C++/OpenCV based application. When the calibration is done, the result is a XML-File acting as configuration file for the camera operator.

OpenTrackCalib shows two windows:

1. The main one is the view of the total view camera. A cross cursor (+) or sometimes a grey box is drawn on the windows. According to the step of calibration, the user is prompted to put the cursor to a significant location of to set a box size.
   a. The cursor can be moved with the directional arrows (**left/right/up/down**).
   b. A box can also be resized with the direction arrows: **left/right** for the left side, **CTRL+left/right** for the right size, **up/down** for the upper side, **CTRL+up/down** for the lower side.
2. The secondary windows shows advices about how to choose significant point.

OpenTrackCalib by itself is not enough. The program needs to do a correspondance between the total view camera and the PTZ-Camera. By now only the total view camera image is shown. This is why we now need to see the PTZ-Camera image.

In fact, you can use any program you like for showing the PTZ-Camera image. I recommend VLC media player because you can add a mark in the center of the image, which will allow you to perform a accurate calibration. Follow the steps:

- If VLC is not installed, type: sudo apt-get install vlc
- Run VLC with the PTZ-Camera as input: vlc v4l2:///dev/videoN where N is the number of the device
- In the VLC window, click: "Tools -> Effects and Filters -> Video Effects -> Vout/Overlay -> Add text -> Text: + -> Position: Center -> Close": a white cross should have appared in the center of the window. The cross will be use to make correspondance between points of the two cameras.

You are now ready to process the calibration. Just follow the instructions given through the console and process till the end. A XML-File will be saved in the current directory.

Get the calibration tool

The calibration tool can be checked out from this svn: svn checkout https://opencast.jira.com/svn/OPENTRACK/trunk/tools/OpenTrackCalib

A readme file explains how to run the tool. The source code is also available.

Important! The calibration is done only once. After the calibration, the set of cameras can not be moved! Set up the two camera in the final operating position and perfom the calibration with a person acting as "lecturer". A good calibration will then lead to a good tracking quality!

Camera Operator

The camera operator is loaded in Felix. The pattern is simply a loop reading tracking information every defined period of time (usually 2-3 seconds). The is done in a Thread.

Description of the camera operator structure:

- CameraOperatorService.java: is the service launched through Felix, at startup, the service reads the config files and create the CameraOperator (Thread).
- CameraOperator.java: is the Object where most of the tracking logic is performed. It's a running thread that periodically process data of the tracker and sends commands to the camera-control.

The calibration data are loaded once at the creation. Some other parameters are set throught the standard "properties" file.

The file "opentrack.camera-operator.properties" (located in $FELIX_PATH/conf/services) has the two following mandatory values:

- org.opencastproject.opentrack.cameraoperator.width=NUMBER

- org.opencastproject.opentrack.cameraoperator.height=NUMBER

These two values correspond to the resolution of the total view camera.

TO DO

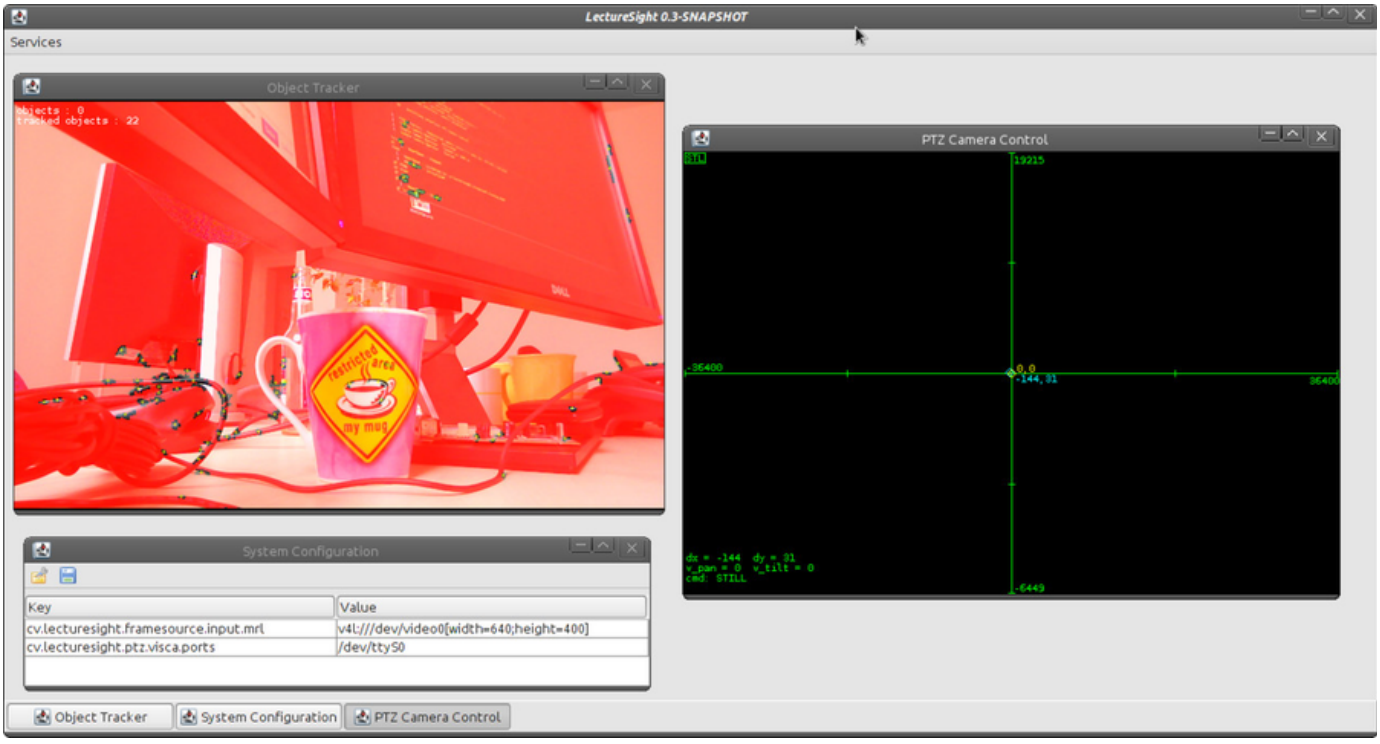
- No move zone remains to implement
- The system only prints the commands the camera should receive. As no real environement has been set (I mean in a lecture room), the dependency with the camera-control is quite inexistant.
- Correct pending bugs

LectureSight UI - first mockup

Unknown macro: 'mockup'

Operation

This section is the source of information about setting up and running LectureSight in the lecture hall or seminar room. Here you can find tutorials for beginners as well as detailed manuals for every functional module in LectureSight.



Getting Started

This section should serve as a step-by-step guide to setting up and running LectureSight in a productive environment First discussed are the prerequisites of hard- and software. The reader may skip that section completely if the host system has already been set up.

1. Prerequisites

LectureSight is a software build up of modules for the OSGI software modularization framework [All07] for the Java programming language. Java itself is platform-independent but LectureSight relies on certain features of the Linux operating systems, so the software must be run on Linux. Also LectureSight uses the GPU for the video ananlysis and thus a fairly modern graphics card will be needed to run the system. Also the choice of the overview camera can have an effect on the tracking performance.

The next section gives an in-depth discussion about hardware components that can be used with LectureSight. The following section describes the software requirements for running LectureSight.

1.1. Hardware

As mentioned, as the host system for LectureSight a fairly modern personal computer that is capable of running Linux is sufficient.

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 defualt memory configuration for the Java VM suffice.

Mass Storage

For the whole system roughly 50 MB of mass storage must be reserved. Despite of log files that may need to be rotated, there are no sites where data is incrementally saved.

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 C UDA 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.

GPUs 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)

Overview Camera

As overview camera, USB webcams as well as analog SD video cameras connected to a fast frame grabber device can be used, or any image source which can be provided through a GStreamer pipeline.

Though the resolution for the overview camera should not be too high in order not to jeopardize real-time performance (usually VGA), the image qualtiy 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 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.

See the Overview Cameras section for a collection of different models that have already been tested with LectureSight.

1.2. Software

Operating System

In principal every modern Linux distribution can serve as host operating system. A kernel supporting Video4Linux or Video4Linux2 is the only requirement for the video analysis. LectureSight has already been tested with the following distributions

- Ubuntu: 10.04, 11.10, 12.04
- Slackware: 13

OpenCL

The OSGI-bundles that make up LectureSight come with all their dependencies included. Thus there is no need for third-party software to be installed on the host system except the OpenCL implementation. The OpenCL implementation is usually part of the driver package that comes with the graphics card.

For NVIDIA products the Linux driver can be downloaded from the website

http://www.nvidia.com/object/unix.html

For ATI products the Linux driver can be downloaded from

http://support.amd.com/de/gpudownload/linux/Pages/radeon_linux.aspx

The driver installations include the OpenCL library implementation suitable for the particular graphics card and should install an .icd file under /etc/OpenCL/vendors that points to the library file.

2. Installation

The pre-compiled binary distribution for production systems can be downloaded here

lecturesight-0.3-rc1.tar.gz

Extract the software e.g. to /opt/

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

Change into the lecturesight directory and start the software

```
cd /opt/lecturesight-0.3-rc1
./bin/start_lecturesight.sh
```

By default LectureSight will try to use the video device /dev/video0 as overview camera with a resolution of 320x240 pixels (QVGA). It will not try to initialize any PTZ-camera.

You will see an empty window coming up. The ``Services'' menu should be populated with a number of entries. If you only see the entry ``System configuration'' it is most likely that the system was not able to initialize the OpenCL platform successfully. You can find out if the OpenCL platform was initialized correctly by looking for the OpenCL device report in the console

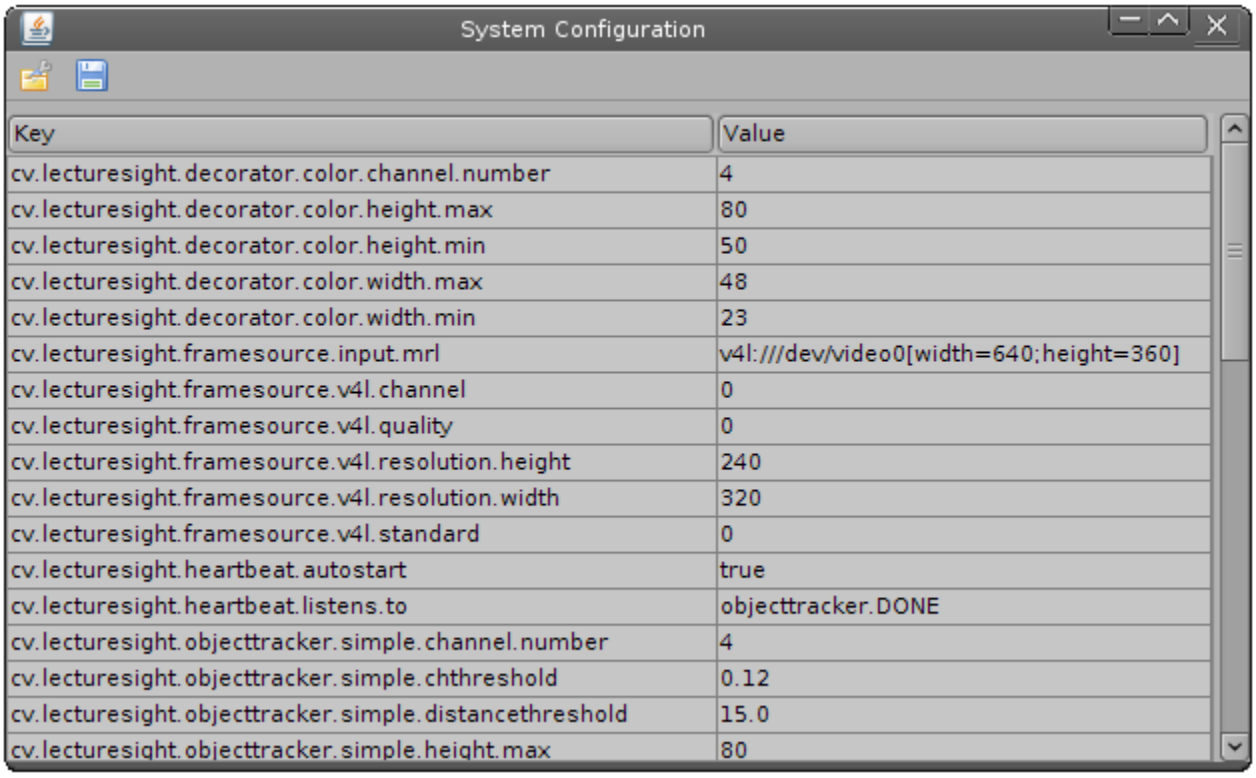
```
[ i ] OpenCL Service : OpenCL platform: NVIDIA CUDA OpenCL 1.0 CUDA
3.2.1 FULL_PROFILE
[ i ] OpenCL Service : Device report: NVIDIA Corporation GeForce GT 220
(driver version: 260.19.06)

Compute units :   6 at 1360 MHz max

     Memories :   global   : 1023.3125 MB
                             constant : 64.0 KB
                             local    : 16.0 KB

   Workgroups :   512 threads max in 3 dimensions
2D Image size :   4096x32768 max
```

If you don't find such an output 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.

3. Configuration

The whole system configuration is loaded from a Java properties file that, by default, is conf/lecturesight.properties. You can edit this file with a text editor or directly in the LectureSight GUI (Menu: Services > System Configuration) where you can also save the difference to the default configuration.

Every bundle in LectureSight comes with it's default configuration. If you set a property value in the UI that differs from the default configuration that comes with the bundle, then this property will be written to the properties file when you choose to save the configuration. This way the properties you will find in the configuration file reduce to those that you have touched in your configuration.



3.1. Overview Camera

By default LectureSight will try to use /dev/video0 as the overview camera with a resolution of 320x240 pixels. This is only the default resolution that most web-cams should be capable of. In operation, however, a resolution around VGA (640x480 or so) should be used. You can configure the input device and resolution via the input.mrl property of the FrameSource service. The following line shows the default configuration

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

See section 3.2.2 for details about the configuration of the Video4Linux frame source. In case you are using a capture card for the video input you might have to set additional parameters such as the number of the video input to use (as is the case with most IVTV cards).

⚠ After you have changed the input.mrl property, the system has the be restarted in order for the change to take effect. Either change the configuration in an external editor or save the configuration to conf/lecturesight.properties and restart the system.

You can stop the system either by pressing **Ctrl+C** or by stopping the system core bundle (that has always the ID 0) using the stop command in the console. Both ways shut down the system orderly.

```
stop 0
```

Then start the system again. When the video device is configured correctly you should notice something like the following in the console on startup

```
[ i ] Video4Linux FrameSource : Device name: FV TouchCam N1
[ i ] Video4Linux FrameSource : Device supports formats: YUYV - 1 -
640x480 (1/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5 - ) - 640x360 (1/30 - 1
/20 - 1/15 - 1/10 - 1/5 - 1/1 - ) - 640x400 (1/30 - 1/25 - 1/20 - 1/15
- 1/10 - 1/5 - ) - 384x216 (1/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5 - )
- 352x288 (1/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5 - ) - 320x240 (1/30 -
1/25 - 1/20 - 1/15 - 1/10 - 1/5 - ) - 320x200 (1/30 - 1/25 - 1/20 - 1
/15 - 1/10 - 1/5 - ) - 176x144 (1/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5
- ) - 160x120 (1/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5 - ) - 160x128 (1
/30 - 1/25 - 1/20 - 1/15 - 1/10 - 1/5 - ) - 768x480 (1/20 - 1/15 - 1/12
- 1/10 - 1/7 - 1/5 - ) - 1280x720 (1/11 - ) - 1280x800 (1/9 - ) -
```

After the restart you should see an entry ``Video Input'' in the ``Services'' menu. When selected a window should come up showing the live video from the overview camera.

3.2. PTZ-Camera

Out of the box there is no PTZ-camera configured in LectureSight, though the driver for cameras speaking Sony's VISCA protocol is shipped with the system. You can activate the driver by setting the configuration property **cv.lecturesight.ptz.visca.ports** and updating the driver bundle or restarting the system.

Start by setting the configuration of the serial device connected to the VISCA camera. Add the following line to the configuration file or set the property in the UI accordingly

```
cv.lecturesight.ptz.visca.ports=/dev/ttyS0
```

If you use a USB-to-serial converter to connect the camera, the device will probably be named something like /dev/ttyUSB0.

After setting the property in the UI you can activate the driver by updating the driver bundle. To do this you have to find out the ID of the bundle via the command

```
lb
```

You will see a list of all the bundles that are currently installed, their ID, state, runlevel and name. Search for the bundle called ``LectureSight PTZ-Camera Control VISCA'' and type

```
update <id>
```

This will tell the system re-initialize the bundle containing the VISCA driver. Now that there is a device configured for the driver, it will activate and try to initialize the VISCA camera connected to the serial port. If the camera could be initialized successfully you should see the following message in the console

```
[ i ] VISCA Factory : Model ID: 1038
[ i ] VISCA Factory : Initialized Sony Inc. EVI-D70 on /dev/ttyS0 (bus
id: 1)
```

3.3. 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. First we have to tell the PTZ camera where in it's field of operation the bound of the scene lie that the overview camera observes. Second we need to tell the system the approximate size of a person. Lastly, in most lecture or seminar rooms, it is a good idea to restrict the area in which the systems tracks objects.

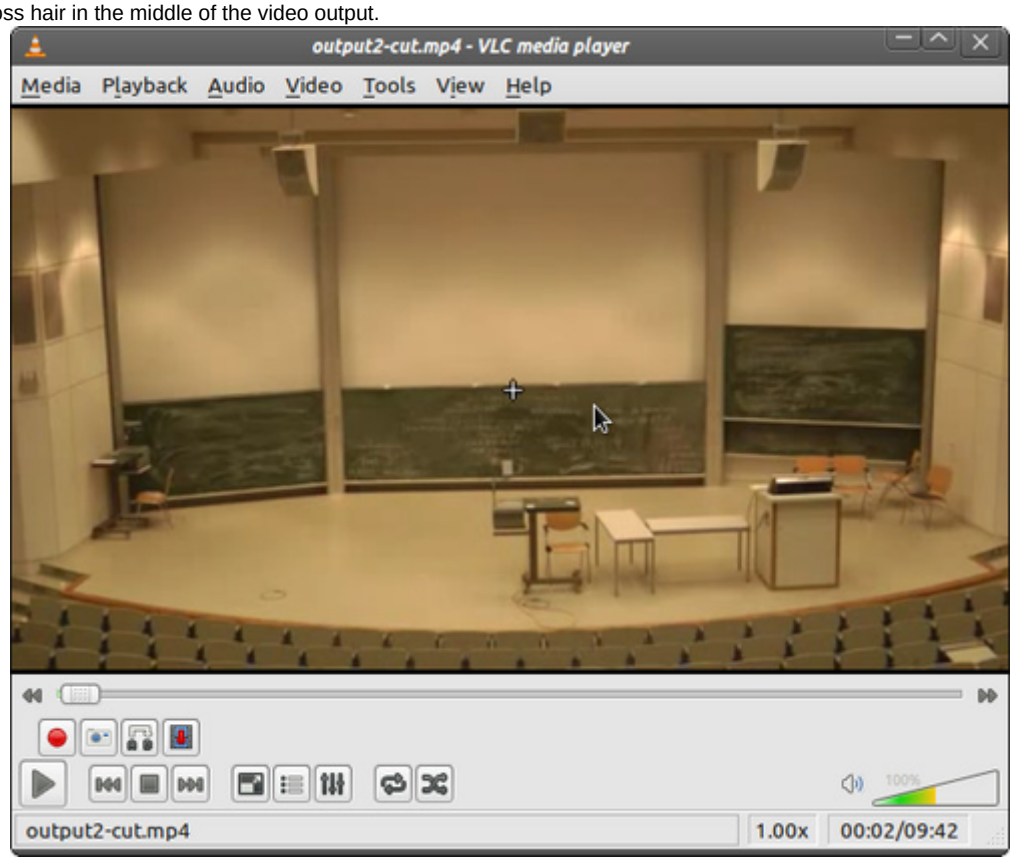
Scene bounds

Setting the scene bounds is the most crucial part of the calibration. Without the scene bounds set correctly the system will not be able to translate coordinates of tracked objects into angles for the PTZ camera.

For adjusting the scene bounds you need to view the image from the PTZ camera. Ideal for the calibration process is a video player with overlay functionality like VLC, which we will use in this tutorial. The following steps are needed to prepare for the calibration:

1. Open the VLC player
2. Open the capture device your PTZ camera is attached to (Menu: Media -> Open Capture Device).
3. Open the "Adjustments and Effects" window (Menu: Tools -> Effects and Filters) and navigate to the "Vout/Overlay" tab inside the "Video Effects" tab
4. Activate the option "Add Text" and type a "+" into the "Text" field, choose "Center" as position.

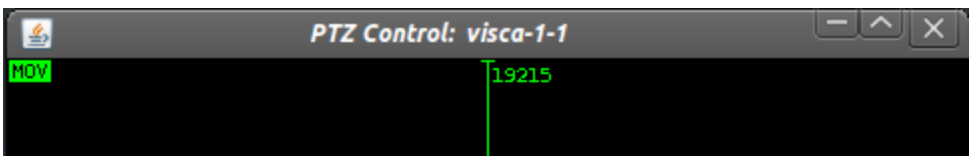
Now we have a cross hair in the middle of the video output.

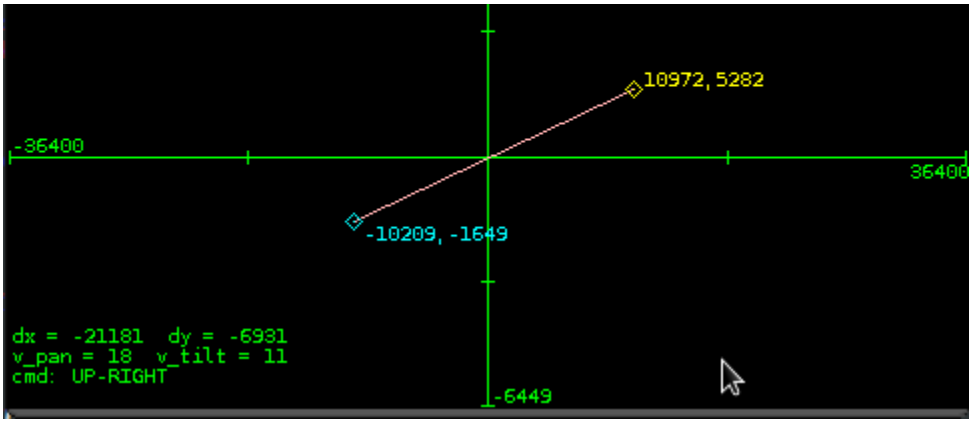


Before you can control the camera yourself you must deactivate the camera operator module that normally controls the PTZ camera. In the console type **lb**, then search the bundle list for the bundle "LectureSight (simple) Camera Operator" and find its ID. Then to stop it type **stop <ID>**, you should see the log message

```
[ i ] Simple Camera Operator : Deactivated
```

With the camera operator deactivated you can now use the "PTZ Camera Control" window to move the camera.





You can set a target position for the camera by simply clicking into the window. Next perform the calibration:

1. Bring up the "Video Input" window so you can see the image from the overview camera.
2. Find a significant spot at the left edge of the overview image.
3. Move the camera so that the spot you identified before is under the cross hair.
4. Set the property **cv.lecturesight.ptz.steering.worker.scene.limit.left** to the X value (first blue number) from the PTZ control window.
5. Repeat steps 2. to 4. for

- **cv.lecturesight.ptz.steering.worker.scene.limit.right**
- **cv.lecturesight.ptz.steering.worker.scene.limit.top**
- **cv.lecturesight.ptz.steering.worker.scene.limit.bottom**

⚠ Don't forget to save the configuration to a file (in the 'System Configuration' window).

Tracking area

The module for setting the tracking areas in the scene is still in development. But there is already a mechanism for restricting the tracking to certain areas in the scene: A simple monochrome mask image can used to mask out areas in the scene, where tracking is not wanted, such as the area of the auditorium or the doors of the room.

The first step to creating such a mask image is to take a snapshot of the room from the overview camera. You can do this by bringing up the 'Video Input' window from the 'Services' menu. When you have the overview camera in the right position, click the camera button in the toolbar. LectureSight will save a snapshot from this window as a PNG file in the directory you specify in the save dialog.

ℹ By the way: You can save snapshots from every display window that is available in LectureSight by clicking the camera button in the toolbar.

Best for creating a mask image is an image manipulation application that can handle image layers like PhotoShop. In this tutorial we will use the free Linux application GIMP that should come with most Linux distributions.

To create an scene mask perform the following steps:

1. Load the snapshot of the scene.
2. Create a new layer (called 'mask' or something) above the original scene.
3. On that layer paint all areas black in which no tracking should be conducted.
4. Create a new, all white layer between the original scene and the mask so that only the mask layer and a white background are visible.
5. Save the whole project in the applications native (layer supporting) format. This way you can change the scene mask if needed later on.
6. Save the image as a PNG file. The application will ask you if you want to export the image as it is visible at that time.

See the two images below for an example of a scene mask image.

When you are done creating the scene mask, you can configure the system to use it with the configuration property

```
cv.lecturesight.framesource.input.mask=mask.png
```

⚠ You need to restart the system if you configure the scene mask in runtime.

Hardware Components

Current Test System

The current test system that is part of the experimental setup in the lecture hall has following configuration

CPU	Intel Core 2 Duo
RAM	2 GB
GPU	Nvidia GT 220, 512 MB
CAM	FaceVision TouchCam N1
OS	Ubuntu 10.04

Camera Setup

In the current version the overview camera and the PTZ camera must be mounted in the same location.

Overview Cameras

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

USB extension products

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).

Microsoft Kinect

Integration for the Microsoft Kinect is currently being developed. In the first step the Kinect's depth sensor and optical camera will be used as FrameSources.

Since the Kinect provides a depth image of an area up to four meters from the camera with an horizontal angular aperture of 58.5 degree, it seems especially promising for use in smaller scenarios like deployment in a seminar room.

PTZ Cameras

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.

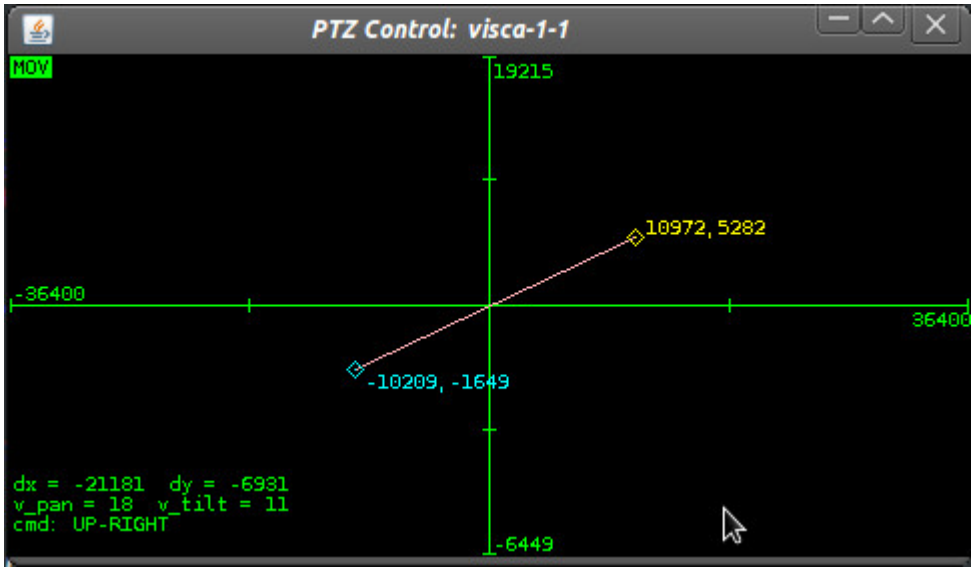
Service Bundle Manuals

In this section detailed manuals for all bundles that hold implementations of services in LectureSight can be found. The subsections explain the functionalities of those modules and give insight into the inner workings of the services. Further more all configuration parameters and console commands for the services are lsited and explained. Operators may use this section as a reference for deployment and configuration.

Camera Steering Worker

Bundle File: lecturesight-steeringworker-impl.jar

The bundle provides the so called "Camera Steering Worker". It is responsible for moving the camera. The steering worker is given a target position and is constantly monitoring the cameras position and decides whether to move the camera and at which speeds. The target position 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 is closing in on 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.



Manual Camera Control
In normal operation the Steering Worker will get the target position from the Camera Operator service. If you want to control the camera manually you have to deactivate the camera operator service first since otherwise target positions set manually will be overwritten by the camera operator.

Console Commands

`cs:start`

Initializes the worker thread and activates the camera control.

`cs:stop`

Deactivates and camera control and shuts down the worker thread.

`cs:on`

When the worker thread is running, this command activates the camera control.

`cs:off`

When the worker thread is running, this command deactivates the camera control. Afterwards the worker will still monitor camera position and display it in the UI, but it won't send movement commands to the camera.

`cs:restart`

Updates the configuration and restarts the service.

`cs:move <normalized position>`

Sets the target position for the steering worker. Coordinates must be in normalized form.

`cs:stopmove`

Tells the service to stop all current camera movements.

`cs:home`

Tells the service to move the camera into it's home position.

`cs:zoom`

(Currently disabled due to bug in steering worker) Tell the service to set the zoom factor for on the camera.

Configuration Properties

cv.lecturesight.ptz.steering.worker.autostart

Default: true

Tells the service if it should start operation right when it is activated. In production this property should be set to *false* since the iCal Scheduler service will take care for activating camera control when a recording is scheduled.

cv.lecturesight.ptz.steering.worker.interval

Default: 50

The interval rate in milliseconds the steering worker logic is called with. By default it is called every 50 milliseconds, thus 20 times a second.

cv.lecturesight.ptz.steering.worker.move.alpha.x

Defaults: 4000

The horizontal distance (in units from the camera's internal coordinate system) to the target position at which the steering worker begins to decrease the movement speed.

cv.lecturesight.ptz.steering.worker.move.alpha.y

Default: 2000

The vertical distance (in units from the camera's internal coordinate system) to the target position at which the steering worker begins to decrease the movement speed.

cv.lecturesight.ptz.steering.worker.move.stop.x

Default: 10

The horizontal distance (in units from the camera's internal coordinate system) to the target position at which the steering worker doesn't produce any correction moves.

cv.lecturesight.ptz.steering.worker.move.stop.y

Default: 10

The vertical distance (in units from the camera's internal coordinate system) to the target position at which the steering worker doesn't produce any correction moves.

cv.lecturesight.ptz.steering.worker.move.damp.pan

Default: 1.0

(0.0 ... 1.0) The dampening value the pan speed is multiplied with when moving the camera. By default the camera is operate with its maximum speed (1.0 = 100%). If the value is setto 0.5, the camera will be operated at half it's maximum speed (0.5 = 50%).

cv.lecturesight.ptz.steering.worker.move.damp.tilt

Default: 1.0

(0.0 ... 1.0) The dampening value the tilt speed is multiplied with when moving the camera. By default the camera is operate with its maximum speed (1.0 = 100%). If the value is setto 0.5, the camera will be operated at half it's maximum speed (0.5 = 50%).

cv.lecturesight.ptz.steering.worker.scene.limit.left

Default: none

The pan value in the coordinate system of the camera that corresponds to the left edge of the overview image. If this property is set to "none" or empty, the value will be taken from the camera's profile. Thus, when the scene limits are not configured the steering worker operates on the camera's whole range of movement.

cv.lecturesight.ptz.steering.worker.scene.limit.right

Default: none

The pan value in the coordinate system of the camera that corresponds to the right edge of the overview image. If this property is set to "none" or empty, the value will be taken from the camera's profile. Thus, when the scene limits are not configured the steering worker operates on the camera's whole range of movement.

cv.lecturesight.ptz.steering.worker.scene.limit.top

Default: none

The pan value in the coordinate system of the camera that corresponds to the upper edge of the overview image. If this property is set to "none" or empty, the value will be taken from the camera's profile. Thus, when the scene limits are not configured the steering worker operates on the camera's whole range of movement.

cv.lecturesight.ptz.steering.worker.scene.limit.bottom

Default: none

The pan value in the coordinate system of the camera that corresponds to the bottom edge of the overview image. If this property is set to "none" or empty, the value will be taken from the camera's profile. Thus, when the scene limits are not configured the steering worker operates on the camera's whole range of movement.

Decorator Color Histrogram

Bundle File: lecturesight-decorator-color.jar

The bundle provides a service that implements an ObjectDecorator. For object that fit configurable size criteria, the service creates and updates color histograms. The number of bins of the histograms can be configured.

Configuration Properties

cv.lecturesight.decorator.color.width.min

Default: 23

Minimum width for an object to be anaylsed by this decorator.

cv.lecturesight.decorator.color.width.max

Default: 48

Maximum width for an object to be anaylsed by this decorator.

cv.lecturesight.decorator.color.height.min

Default: 50

Minimum height for an object to be analysed by this decorator.

cv.lecturesight.decorator.color.height.max

Default: 80

Maximum height for an object to be anaylsed by this decorator.

cv.lecturesight.decorator.color.channel.number

Default: 4

Number of channels for the histogram.

Decorator Head

Bundle File: lecturesight-decorator-head.jar

The bundle provides a service that implements an ObjectDecorator. It purpose is to find the head of a tracked person. A K-means clustering is performed on the foreground pixels of a given object creating a Voronoi tessellation. The prototypes of the clustering are the centroids of the cells. The prototype with the smallest y value that is closest to the vertical centroid axis of the whole object is chosen as the prototype for the head. Thus the cell associated with the chosen prototype should hold the pixels that make up the image of the head of the analyzed person. A detailed description of the method is given in [WFec].

Configuration Properties

cv.lecturesight.decorator.head.k

Default: 7

Number of prototypes for the clustering (K).

cv.lecturesight.decorator.head.iterations.max

Default: 100

Maximum number of iterations for the clustering algorithm in one run.

Foreground Region Tracker

Bundle File: lecturesight-regiontracker-impl.jar

The bundle provides the Foreground Region Tracker Service. It tracks regions in the foreground model by mapping labels produced by the connected component analysis in subsequent frames. The service is also capable of filtering regions based on their width, height and total pixel count.

Configuration Properties

cv.lecturesight.regiontracker.maxnum

Default: 30

[1 . . . MAX INT] Maximum number of regions that should be tracked.

cv.lecturesight.regiontracker.size.min

Default: 50

[1 . . . MAX INT] Minimum number of total pixels in a tracked region.

cv.lecturesight.regiontracker.size.max

Default: 7000

[1 . . . MAX INT] Maximum number of total pixels in a tracked region.

cv.lecturesight.regiontracker.width.min

Default: 15

[1 . . . MAX INT] Minimum width of a tracked region.

cv.lecturesight.regiontracker.width.max

Default: 100

[1 . . . MAX INT] Maximum width of a tracked region.

cv.lecturesight.regiontracker.height.min

Default: 30

[1 . . . MAX INT] Minimum height of a tracked region.

cv.lecturesight.regiontracker.height.max

Default: 200

[1 . . . MAX INT] Maximum height of a tracked region.

cv.lecturesight.regiontracker.debug

Default: false

[true/false] Whether or not the debug mode for the Region Tracker Service should be active.

Frame Source Manager

Bundle File: lecturesight-framesource-impl.jar

The FrameSource bundle provides the infrastructure responsible for managing FrameSource implementations. It discovers video input plugins and is responsible for setting up FrameSources with the propper input plugin. It also registers the FrameSourceProvider service on startup, which is the point in the system where other services get the (masked) input image from the overview camera.

Configuration Properties

cv.lecturesight.framesource.input.mrl

Default: v4l:///dev/video0[width=320;height=240]

MRL of the video input from the overview camera that is served by the FrameSourceProvider. The MRL must have the following form:

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

With

- `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`

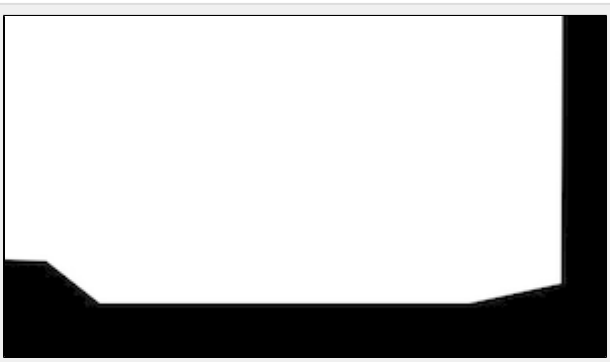
For example, the default value for this property tells the system to use the Video4Linux device /dev/video0 as the overview source with QVGA resolution.

cv.lecturesight.framesource.input.mask

Default: none

The path to the input mask image file for masking the input provided by the FrameSourceProvider. The masking is deactivated if value is none. Otherwise the provided mask image is laoded when the framesource is activated and the mask is applied to each input frame before it is made available to consuming services: Each pixel in the input frame is set to black if the first component of the color value of the corresponding pixel in the mask image equals 0.

The following is an example of a mask image and the resulting masked scene



The following constraints apply to mask image files:

- the file format must be either PNG or JPEG
- the dimensions of the mask image must be at least those of the video input
- for a masking pixel the first component of the color value must equal 0

Heart Beat

Bundle File: lecturesight-heartbeat

The bundle provides the so called "Heart Beat" 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.

Console commands

`ls:run`

Lets the service activate the video analysis subsystem.

`ls:step <number of frames>`

Lets the service run the video analysis subsystem run for the given number of frames. If the argument is omitted the video analysis is run only a single iteration. This command is especially useful for debugging when working with a video file frame source instead of a life video input.

`ls:pause`

Pauses the operation of the video analysis services without de-initializing the service.

`ls:restart`

Re-initializes the heart beat service and start the video analysis subsystem. This command mighty 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.

Configuration Properties

cv.lecturesight.heartbeat.listens.to

Default: cv.lecturesight.foreground.cleaning.DONE

A comma separated list of signal name the heart beat service waits for before kicking off the processing of the next frame. By default the service listens to the signal that indicates that the foreground service has finished its cleaning operations on the foreground map since this is the last computation that is done.

Changing this value can cause serious malfunction of the video analysis components. For normal operation this property should not be touched.

cv.lecturesight.heartbeat.autostart

Default: true

This property tells the heart beat service if it should activate the video analysis right when itself is activated (eg. when the system is started) or not. In production this value should be set to *false* since the iCal scheduler will take care for activating the video analysis when a recording is scheduled.

iCal Scheduler

Bundle File: lecturesight-scheduler.jar

The bundle provides a service that loads a schedule from an iCalendar (RFC-2445) file and starts/stopps object tracking and camera control accordingly. Changes to the file are detected and the internal schedule is updated automatically. When the file is deleted, all events are removed. The implementation works internally with a periodically called routine that ensures that the state of object tracking and camera control components are consistent with the schedule. This way the system will, for instance, start to control the camera even if started after a scheduled begin of a recording.

Depending on the video analysis implementation, the tracking components may need a certain time to adapt to the scene before producing correct tracking results. To prevent false camera movement caused by false positives the services can be configured to start camera control some time after the object tracking has been activated. Usually automatic lecture recordings are started with some lead time an then cut afterwards. It is supposed to set the lead time for the tracker scheduler to some value smaller then this lead time.

Configuration Properties

cv.lecturesight.scheduler.schedule.file

Default: schedule.ics

The name of the iCal file holding the schedule.

cv.lecturesight.scheduler.agent.name

Default: none

A capture agent name the service will look for in case the iCal holds schedules for more than one capture agent. If this property value is empty, the service will take every event from the iCal into account.

cv.lecturesight.scheduler.timezone.offset

Default: 1

The time zone offset to add to the event times from the schedule.

cv.lecturesight.scheduler.tracking.leadtime

Default: 10

The time (in milliseconds) the service will wait after the object tracking has been activated before the camera control is activated. Tracking algorithms may need a certain amount of time to adapt to the
scene. In this period false positives could lead to unwanted camera movements, so starting the camera control some time after the object tracking might be useful.

Object Tracker

Bundle File: lecturesight-objecttracker-impl.jar

The bundle provides an Object Tracker Service for tracking and re-discovering of objects. It relies on the Foreground Region Tracker Service and considers moving regions of a reasonable size as objects of interest. In each frame, all candidate regions are checked against all objects in scene (we use a combined distance measure from physical distance and color histogram distance). Candidate regions which cannot be matched to existing objects create new objects, objects which cannot be matched to existing candidate regions stay active for a short time, until they are onsidered to not be part of the scene any more. New objects are checked against older inactive objects, using the distance measure.

Configuration Properties

cv.lecturesight.regiontracker.maxnum

Default: 30

[1 . . . MAX INT] Maximum number of regions that should be tracked.

cv.lecturesight.objecttracker.simple.width.min

Default: 15

[1 . . . MAX INT] Minimum width of a region, that can be considered as being a human in the scene.

cv.lecturesight.objecttracker.simple.width.max

Default: 60

[1 . . . MAX INT] Maximum width of a region, that can be considered as being a human in the scene.

cv.lecturesight.objecttracker.simple.height.min

Default: 12

[1 . . . MAX INT] Minimum height of a region, that can be considered as being a human in the scene.

cv.lecturesight.objecttracker.simple.height.max

Default: 100

[1 . . . MAX INT] Maximum height of a region, that can be considered as being a human in the scene.

cv.lecturesight.objecttracker.simple.ttl

Default: 600000

[1 . . . MAX INT] Time after which a not moving object is assumed to have left the scene. (in milliseconds)

cv.lecturesight.objecttracker.simple.regionmatch.threshold

Default: 5.0

[0 . . . MAX DOUBLE] Threshold, which defines minimum similarity of two objects in terms of the described distance measure. If distance(obj1 , obj2) > regionmatch.threshold, two objects are assumed to be distinct.

cv.lecturesight.objecttracker.simple.chthreshold

Default: 0.12

unused

cv.lecturesight.objecttracker.simple.distancethreshold

Default: 15.0

[0 . . . MAX DOUBLE] Threshold, which defines minimum similarity of two objects in terms of the euclidean distance. If distance(obj1 , obj2) > distancethreshold, two objects are assumed to be distinct. It is used in cases, where color histogram information plays no role.

cv.lecturesight.objecttracker.simple.channel.number

Default: 4

[1 . . . 255] Color Histograms of objects are stored three-dimensional arrays, where dimensions represent R/G/B channels. The cannel number defines the bin number of the histogram. A histogram with bin number 255 therefore distinguishes 16 million different colors.

Simple Camera Operator

Bundle File: lecturesight-cameraoperator-simple.jar

The Simple Camera Operator bundle provides a most basic implementation of a camera operator service. When activated it drives the camera into home position and waits for targets. When the object tracker discovers targets the this camera operator selects the biggest target and constantly tries to pan to where the target is. The tilt position of the camera is set in the configuration properties. When the object tracker loses the target this service waits for a configurable amount of time before dropping the target and starting to look for another target.

Configuration properties

cv.lecturesight.cameraoperator.simple.tilt

Default: 0.0

The tilt value that will be used when moving the camera.

~~cv.lecturesight.cameraoperator.simple.zoom~~

Default: 5

(Currently disabled due to bug in steering worker) The zoom value that will be set when the camera operator is activated.

cv.lecturesight.cameraoperator.simple.timeout

Default: 500

Time time in milliseconds the camera operator waits when the tracker loses the current target before looking for a new target.

Video4Linux Frame Source

Bundle File: lecturesight-framesource-v4l.jar

The bundle provides a FrameSource implementation for accquiring frames from Video4Linux and Video4Linux 2 devices. 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 take from the configuration properties.

Usage

The type for this FrameSource implementation is v4l. The path is the path to a linux video device (/dev/videoX). Availabel arguments are width, height, standard, channel, quality, their meanings are the same as those of the properties in the next section 3.2.2.

Example usage: v4l:///dev/video0[width=320;height=240]

The example MRL tells the system to create a FrameSource using a Video4Linux device /dev/video0 as input with QVGA resolution.

Configuration Properties

cv.lecturesight.framesource.v4l.resolution.width

Default: 320

Default width for input frames.

cv.lecturesight.framesource.v4l.resolution.height

Default: 240

Default height for input frames.

cv.lecturesight.framesource.v4l.standard

Default: 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.

cv.lecturesight.framesource.v4l.channel

Default: 0

Default video input. Usually not used with USB webcams but rather with capture cards. This can be usefull 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.quality

Default: 0

Default encoding quality. Only used for devices that provide encoded video streams (such as MPEG2 or MJPEG). Value range depends on device driver.

> For real time operation only devices that provide raw video streams should be used since encoding and decoding of frames can lead to several hundred milliseconds of delay.

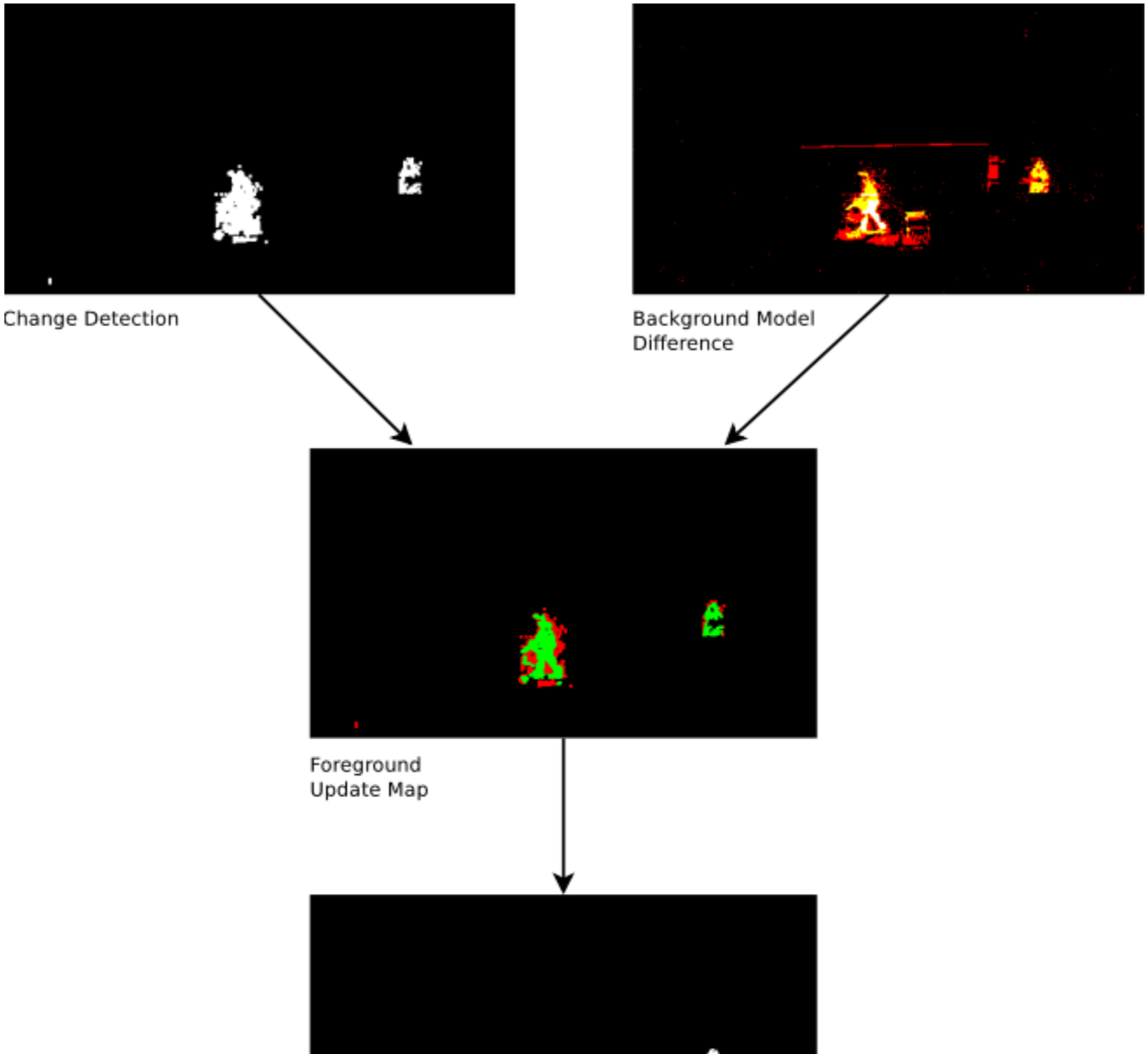
Video Analysis Services

Bundle File: lecturesight-videoanalysis-impl.jar

The bundle provides services for real time video analysis. The implementation is responsible for finding moving objects of a certain size in the scene. In detail the bundle holds three different services

- Change Detection Service
- Background Model and Difference Service
- Foreground Regions Service

The Foreground Regions Service uses the Background Model Service and the Change Detection Service to build the foreground model. The foreground model is build up over time. Regions that show minor activity are aged and removed after some time. Other services can order the Foreground Service to exclude regions from this mechanism. Also other services can order the Foreground
Service to immediately discard a foreground region.



Foreground Map

Console Commands

`fg:reset (no arguments)`

Re-initializes the Foreground Model.

`bg:reset (no arguments)`

Re-initializes the Background Model.

Configuration Properties

cv.lecturesight.videoanalysis.changedetect.threshold

Default: 18

[0 . . . 255] Threshold for the change detection. Lower values yield detection of slighter chnages in the scene.

cv.lecturesight.videoanalysis.background.threshold.low

Default: 30

[0 . . . 255] Low threshold for background differencing.

cv.lecturesight.videoanalysis.background.threshold.mid

Default: 120

[0 . . . 255] Mid threshold for background differencing.

cv.lecturesight.videoanalysis.background.threshold.high

Default: 220

[0 . . . 255] High threshold for background differencing.

cv.lecturesight.videoanalysis.background.update.alpha

Default: 0.5

[0.0 . . . 1.0] Factor for the background model update. The higher the value, the faster is the update of the background model. A value of 0.0 yields no update at all. A value 1.0 yields a background model that basically is always equal to the preceding frame.

cv.lecturesight.videoanalysis.foreground.ccl.maxblobs

Default: 1000

[1 . . . MAX INT] Maximum number of connected components that the Connected Component Analysis can discover. This value exists merely due to technical reasons.

cv.lecturesight.videoanalysis.foreground.ccl.blobsize.min

Default: 20

[1 . . . MAX INT] Minimum number of pixels a region in the foreground map must consist of to be taken up for further analysis in the Foreground Service. Regions smaller than this are discarded very early in the analysis process.

> ⚠ This value is useful for filtering out small artifcats that result from noise in the camera sensor. Also this value should be adjusted when working with video streams of high resolution because with the image size also the size of artifacts rises.

cv.lecturesight.videoanalysis.foreground.ccl.blobsize.max

Default: 10000

[1 . . . MAX INT] Maximum number of pixels a region in the foreground map can consist of to be taken up for further analysis in the Foreground Service. Changes in the scene that have an unplausible size are filtered out by this value.

cv.lecturesight.videoanalysis.foreground.decay.threshold

Default: 0.04

[0.0 . . . 1.0] Minimum pixel activity ration (Nchnaged /N) for a region to be protected from aging in the current iteration.

cv.lecturesight.videoanalysis.foreground.decay.alpha

Default: 5

[1 . . . MAX INT] The value by which the intensity of a region is decreased when aging. Note that intensity values in the foreground map are in [0.. 255].

cv.lecturesight.videoanalysis.foreground.decay.phantom.thresh

Default: 5

[0 . . . 255] Threshold for phantom removal. If a pixel at the edge of a region has a neighbour that is less similar than this threshold, the pixel is removed from the foreground map. This mechanism helps in situation where, for example, a static object like a chair, that was part of the background model is moved. The resulting false positive detections often show greater regions that are similar or equal to the sorrounding background (in contrast to correct detections).

Video File Frame Source

Bundle File: lecturesight-framesource-videofile.jar

The bundle provides a FrameSource implementation that reads frames from a video file. It is based on 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.

Usage

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 usage:` `file://testvideos/sample.mpg`

The example MRL tells the system to create a FrameSource from the file testvideos/sample.mpg.

Requirments

The implementation is based on GStreamer4Java. The bundle does not provide the native libraries, they have to be installed in the host operating system. To install the required libraries under Ubuntu Version 10.04+ the following apt command can be used:

```
sudo apt-get install libgstreamer0.10-0 gstreamer0.10-ffmpeg gstreamer0.
10-alsa /
gstreamer0.10-plugins-bad gstreamer0.10-plugins-bad-multiverse /
gstreamer0.10-plugins-base gstreamer0.10-plugins-base-apps /
gstreamer0.10-plugins-good gstreamer0.10-plugins-ugly /
libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
```

VISCA Camera Driver

Bundle File: lecturesight-ptzcontrol-visca.jar

The bundle provides a driver for cameras speaking the VISCA protocol defined by Sony. On activation the service tries to initialize all VISCA cameras on a configured serial devices. Upon discovery of a VISCA camera the driver determines camera vendor and model and tries to load a fitting device profile. If no fitting profile is existing 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 Properties

cv.lecturesight.ptz.visca.ports

Default: none

A comma-separated list of serial devices that are connected to one or more VISCA cameras (ie. /dev/ttyS0). If this property value is empty the service will not activate.

Camera Profiles

The following is an example of a camera profile definition. It is the default camera profile. Thats 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=RoboCam
camera.pan.min=-880
camera.pan.max=880
camera.pan.maxspeed=18
camera.tilt.min=-300
camera.tilt.max=300
camera.tilt.maxspeed=14
camera.zoom.min=0
camera.zoom.max=16
```

```
camera.zoom.maxspeed=7
camera.home.pan=0
camera.home.tilt=0
```

Quick start guide LectureSight 0.3

Download LectureSight

Requirements

The following minimal requirements need to be fulfilled in order to properly run LectureSight:

- CPU: Core2 Duo 2.4 GHz
- GPU: OpenCL 1.1 compatible
- Disk free: 20 MB
- Overview Camera: see list of tested models
- Operating System: Linux (Mint/Ubuntu tested)
- Java: Java 7 (Oracle)
- GStreamer when using video files as input (see Video File Framesource Manual)

See section 1.1 Hardware for an in depth discussion on the hardware requirements.

Installation

Issue with OpenCL on Ubuntu 14.04 / Linux Mint 17
TODO: Hint: Install following pkgs in order to avoid the NVIDIA ICD bug (from xmas'14):

sudo apt-get install nvidia-331 nvidia-331-uvm nvidia-opencl-dev nvidia-modprobe

1. Get the installation package by clicking the Download button above or from command line via

```
>  wget http://opencast.jira.com/wiki/download/attachments/62914604
/lecturesight-0.3-rc2-x86_64.tar.gz
```

2. Extract the archive e.g. to **/opt**

```
> tar -vzxf lecturesight-0.3-rc2-x86_64.tar.gz -C /opt
```

3. Edit the file **conf/lecturesight.properties** and set the correct paths to your overview camera device and the serial device to which your VISCA camera is connected, also adjust the resolution for the overview camera

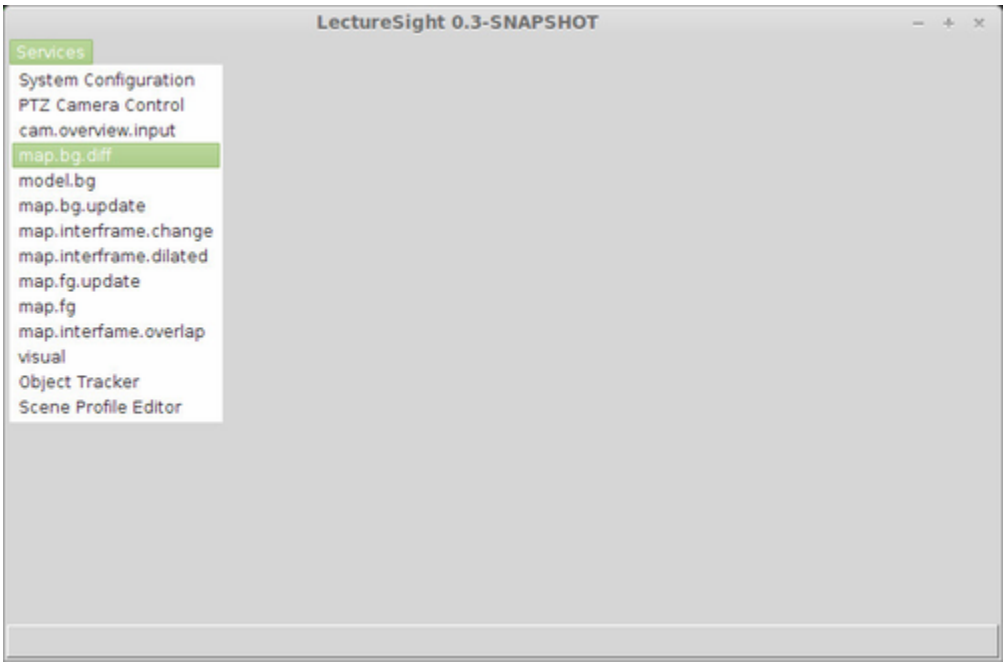
```
#
# Initial configuration for LectureSight
#
```

```
cv.lecturesight.framesource.input.mrl=v4l:///dev/video0[width=640;
height=480]

cv.lecturesight.ptz.visca.ports=/dev/ttyUSB0

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

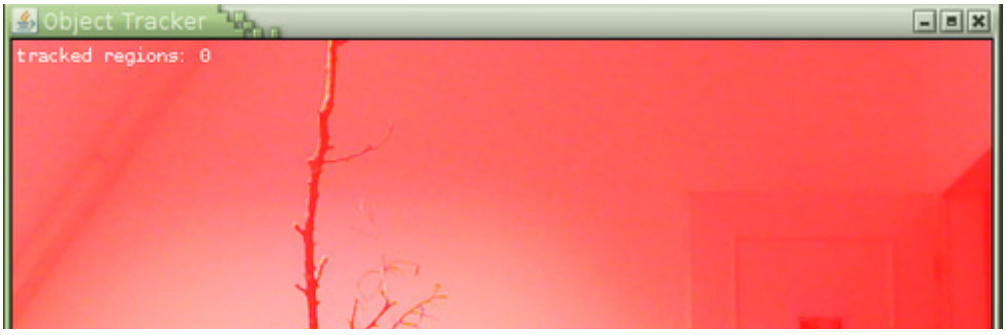
4. After the initial configuration, start LectureSight via the start script

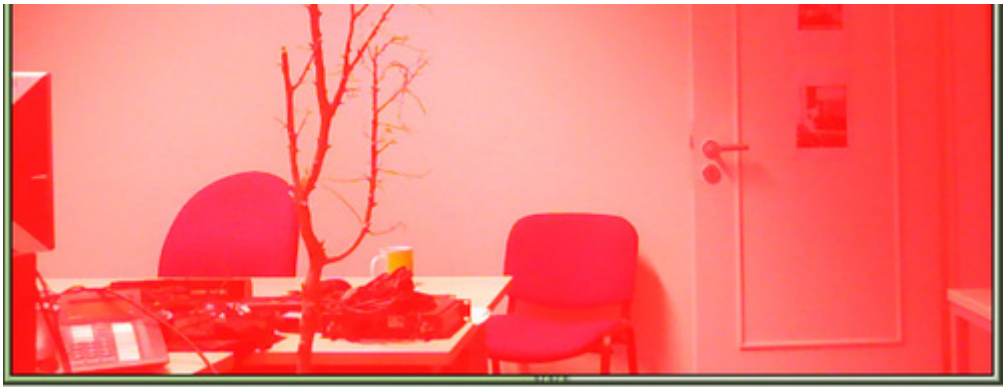
```
> ./start_lecturesight
```

Configuration

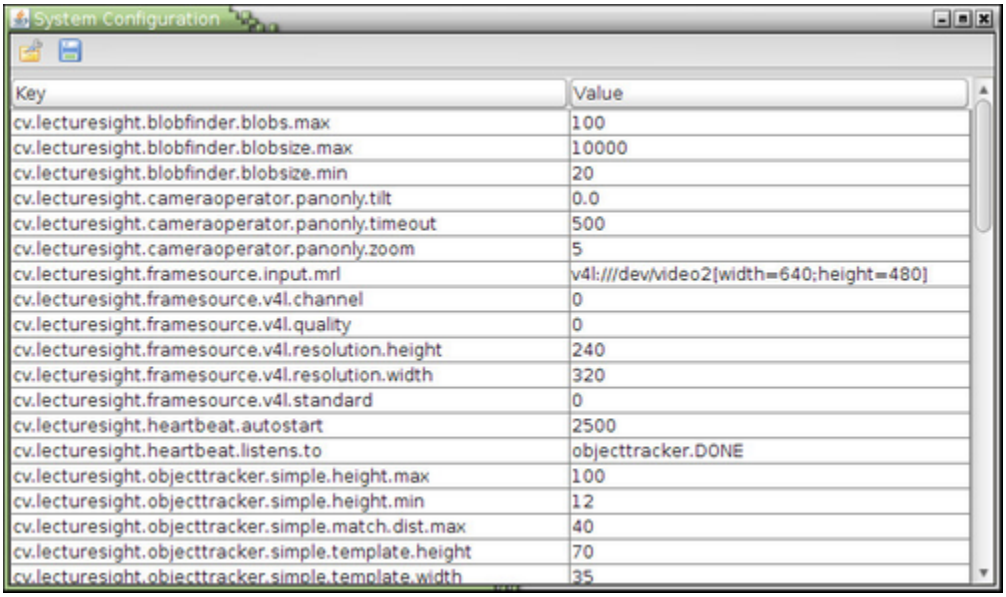


After LectureSight has started the main window will come up (see picture above). In the upper left corner there is a menu called "Services", it holds all the available user interfaces. Choose the menu option "Object Tracker" to bring up the object tracker visualization and see if the system is already working. If you can not find the menu item this means that the system has not been initialized successfully. If the system is working you will the overview camera view with a red tint in the Object Tracker window.





You can access every system configuration property via the "System Configuration" window. The list provides the configuration keys and values. You can edit the values by clicking on them and then typing a new value. Some configuration parameters, like e.g. those for the video analysis, are interactive in that changes on them take effect as soon as you hit enter after editing a values. Other properties, like e.g. the path the to overview camera video device, are used only at system start.

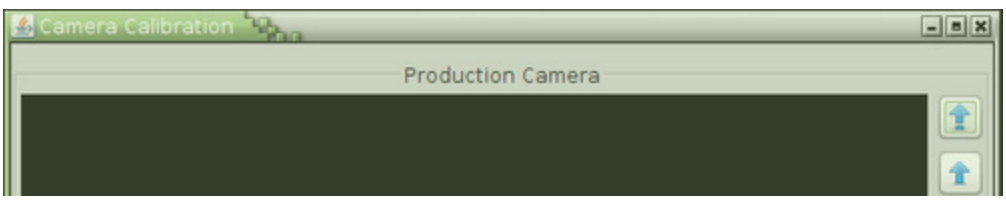


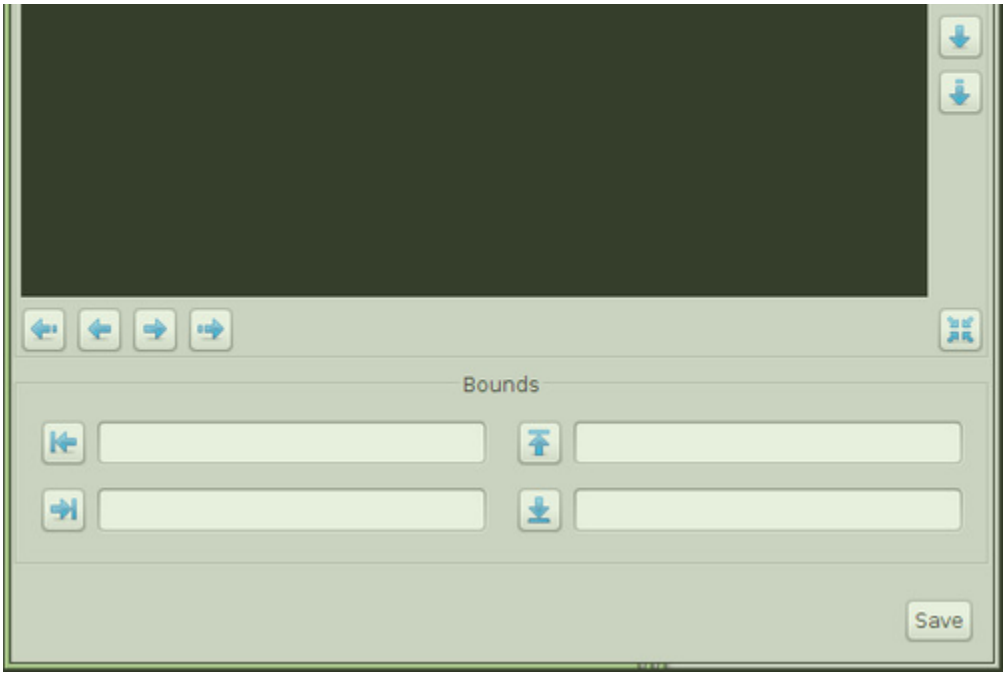
System configurations can be loaded and saved using the buttons in the toolbar.

> ⚠ Every bundle in LectureSight comes with it's default configuration. If you set a property value in the UI that differs from the default configuration that comes with the bundle, then this property will be written to the properties file when you choose to save the configuration. This way the properties you will find in the configuration file reduce to those that you have touched in your configuration.

Camera Calibration

The most important configuration step is the camera calibration. It is needed in order for the system to correctly map coordinates from the overview camera to movement angles of the production camera. From the "Services" menu choose "PTZ Camera Calibration". The automatic camera control is deactivated as long as the calibration window is up.





If you have your production camera also connected to the machine as a video device, you can use it directly in the calibration UI. Add the following line to **conf/lecturesight.properties**

```
cv.lecturesight.setup.video.input=v4l:///dev/video1[width=640;
height=480]
```

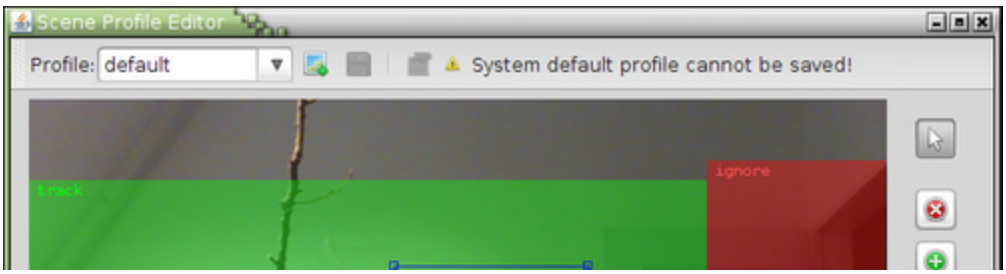
If you don't have the production camera connectedm, the calibration UI will still work but the production camera view will stay black.

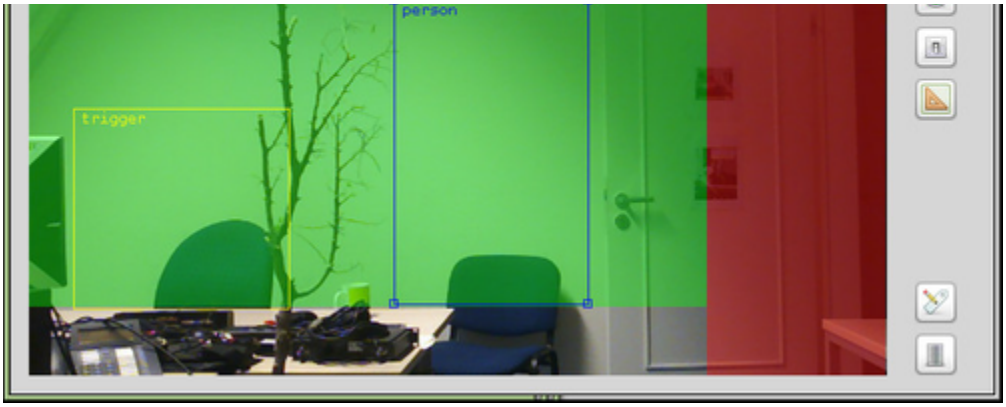
For the calibration we need to know, and tell the system, which coordinates from the PTZ camera correspond to the limits of the overview camera image. Look for significant points at the each side of the overview camera image and try to drive the production camera to those points. You can either use the remote control of the production camera or you can steer the camera from within the user interface by using the buttons at the right and below the production camera view. The button in the lower right corner will make the camera move to the center. When you have driven the camera to a limit click the corresponding button in the "Bound" frame. The current value from the production camera will shown up in the text field next to the button. Repeat the procedure for each limit. When you have the values for all four scene limits, click the "Save" button to apply the values.

By clicking "Save" in the calibration UI you apply the values to the system configuration. To save the system configuration permanently you have to use the "Save" button in the "System Configuration" window after you have applied the values.

Scene Configuration

The system can be customized to certain presentation site by setting areas in the overview image that are ignored. This is especially useful in rooms where the audience is visible in the overview image. Masking out parts of the scene that are not relevant is likely to increase performance since it keeps the system from tracking persons that are not of interest for the recording. You can access an interactive editor the the scene profile by choosing "Scene Profile Editor".





Click on the icon next to the profile selection box to create a new profile. You will be asked for a screen name and the name of a file to save the profile to. With the buttons on the right you can choose which type of area to create. The following types of areas can be defined:

- **Ignore** - Those are areas that are completely neglected by the system. Define them especially in parts of the overview image in which there will be motion (e.g. people in the audience) that is not of interest for the camera control. This will optimize performance of the video analysis.
- **Tracking** - (not implemented yet)
- **Trigger** - Trigger zones emit events when tracked objects enter of leave them. The camera operator script can then act on those events.
- **Person** - (experimental feature) With a person zone, you can tell the system the average size of a person in the scene. When there is a person zone defined in the profile, the system will adjust certain parameters of the video analysis accordingly, when the profile is activated.

In LectureSight 0.3-rc2 you only need to define the ignore and the person areas. As a rule of thumb, mask out all parts of the scene in which tracking is not of interest, leave only the area in which you want the presenter to be tacked without ignore areas. Also, by defining the person area, you can help the system discriminate between persons and other moving object.

Finally, when you are done creating the profile, click the "save" button to safe 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. When you want the system to use your profile permanently, set the name of the profile to use for the following property

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

Research

This section is dedicated to the academic aspects of the project. Here you can find a list of publications about the project as well as a list of scientific papers and books that have influenced the work on the project. Also there is an overview of the area of automated camera control for presentation recording.

Bibliography

- Hoshen and R. Kopelman. Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm. Physical Review B, 14(8):3438, 1976
- O. Kalentev, A. Rai, S. Kemnitz, and R. Schneider. Connected component labeling on a 2d grid using cuda. Journal of Parallel and Distributed Computing, 2010.
- D. Harwood I. Haritaoglu and L. Davis. W 4 : Real-Time Surveillance of People and Their Activities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):809 – 830, August 2000.
- R.E. Kalman et al. A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1):35--45, 1960.
- KA Hawick, A. Leist, and DP Playne. Parallel graph component labelling with gpus and cuda. Parallel Computing, 2010.

Other technologies, solutions or projects

There are numerous solutions to provide automated tracking in a classroom setting, some of them use radio or IR technology, some ex post analysis of long shot recordings, some have mats on the floor. Here's the place to describe these.

Current solution at ETH Zürich (phase-out)

At ETH, camera tracking is part of our PLAYMOBIL capture agent in some of the rooms we have. It works as a live tracking system in that it follows the lecturer during the lecture without any post-processing. The system uses technology from the surveillance business in that movements in a defined area are being detected by one camera (i.e. the software behind it) and reported to another camera tracking the lecturer. This solution calls for a customized setting for each classroom, but it does not ask for the lecturer to wear any trackable object; it certainly has limitations when more than one person is present in the defined area.

Cameras
Our solution is based on a system from NIO Security Munich and is using a Panasonic "WV-CP280/G4" and "WV-LZA61 Vario Objektiv" camera for overview and a Sony PTZ Cam (EVI D100) for tracking, using the VISCA protocol to communicate with the CloseView solution from NIO SEC.

Classroom
We have tried different position for the cameras in the lecture halls. It's important that the cameras are facing down to the lecturer so you can mask the students and the projection. Cameras have to be next to each other for the system to work properly. For large classrooms the Sony PTZ has problems because of the long distance to the lecturer and the light situation.

Demo
http://www.podcast.ethz.ch/channels/details?id=720

GaliTracker

Cf. http://wiki.teltek.es/display/GaliTracker/GaliTracker+Home

In-progress solution at University of Saskatchewan

At Usask we use automated post-production based on templates for all of our video retouching. This allows us to capture as high fidelity and wide angle as we can afford to (limited mostly by storage space), then customize the views for different output formats/styles/preferences/devices /etc. Right now we have simple bounding boxes that crop camera views to give us different levels of zoom or different aspect ratios depending on the room. We have various different setups:

1. Large lecture theater with high zoom presentation camera focused on the instructor station. The zoom camera actually captures the whole stage and is cropped, as one instructor just sits and writes on the sympodiums, while the others in this room walk back and forth across the stage.

2. Short but wide classroom that sits 80. The wide angle lens we have on the camera in this classroom only captures half the front of the room, and has quite a bit of barrel distortion at the edges of the frame (it's essentially a little security camera). We crop depending what the instructor tells us. Note though that the crop is a predetermined parameter on a per-class/per-room basis, we don't have an auto-cropping/tracking going on, but it would be nice.

Publications

This is a collection of documents around the LectureSight project:

- Heger, Garry (2010): Open Track. Bachelor's Thesis at HES-SO
- B. Wulff, R. Rolf (2011): OpenTrack - Automated Camera Control for Lecture Recordings
- B. Wulff (2012): GPU-based real-time Video Analysis for Lecture Recordings, Bachelor's Thesis at the University of Osnabrück
- B. Wulff, A. Fecke (2012): LectureSight - An Open Source System for Automatic Camera Control in Lecture Recordings
- B. Wulff, A. Fecke, L. Rupp, K.-C. Hamborg (2013): The LectureSight System in Production Scenarios and Its Impact on Learning from Video Recorded Lectures

Final 0.3 Release Sprint

The University of Manchester supports the current sprint towards the final 3.0 release version of LectureSight. Development activity during the sprint will focus on the over-all stability of the software and the robustness of the camera tracking. The aimed release 0.3 will be made ready for deployment in lecture theaters and autonomous working in production in January 2015.

About this page
A list of milestones has been set to frame the work on the system. Each milestone is reflected by a JIRA ticket. The actual work items for each milestone are reflected by sub-tasks belonging to the milestone tickets. This page gives an overview on the status of the sub-tasks for each milestone, the data is pulled from JIRA each time this page is opened or refreshed.

LS-72: Improve over-all stability of the system

Key	Summary	T	P	Status
LS-98	Profile Editor: error when trying to save to not existing file			CLOSED
LS-95	Make video file frame			

LS-73: Improve reliability of people tracking

Key	Summary	T	P	Status
LS-99	Introduce foreground cleaning mechanisms			CLOSED
LS-96	Wrap up new Video Analysis Module			CLOSED

	source loop			CLOSED
LS-93	Add support for setting values of V4L2 controls			CLOSED
LS-90	Add Listener interface to Configuration service			CLOSED
LS-85	OpenCL OutOfResourcesException on some systems			CLOSED
LS-63	In Scene Profile Editor initially loads default profile instead of configured one			CLOSED

6 issues

LS-94	Extend Display Service by ability to save frames to files			CLOSED
LS-92	Make Connected Component modules listen to OCLSignals			CLOSED
LS-88	Move temp saving of last frame into FrameSource			CLOSED
LS-87	Try detection approaches			CLOSED
LS-79	Add shadow removal			CLOSED
LS-78	Improve Background Model			CLOSED

8 issues

LS-74: Head Tracking

Key	Summary	T	P	Status
LS-81	Implement k-means clustering in OpenCL			CLOSED

1 issue

LS-75: Scriptable Camera Operator

Key	Summary	T	P	Status
LS-84	Create Scripting Infrastructure based on JSR-223 API			CLOSED
LS-83	Implement Scripting APIs			IN PROGRESS
LS-82	Define APIs			CLOSED

3 issues

LS-103: QA for 0.3 Release

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Resolution	Status
LS-108	Documentation		2015-02-27 04:02 PM	2017-12-05 09:12 AM		Benjamin Wulff	Benjamin Wulff		Duplicate	RESOLVED
LS-107	Configured Scene Profile is not loaded when LS starts		2015-02-03 03:00 PM	2016-05-11 11:22 AM		Benjamin Wulff	Benjamin Wulff		Fixed and reviewed	RESOLVED
LS-106	LS Acquires False Object and Maintains it		2015-01-28 01:31 PM	2016-04-20 12:08 PM		Stephen Marquard	Stuart Phillipson		Fixed and reviewed	RESOLVED
LS-105	When new VISCA service is active, every log message get printed twice		2015-01-21 03:14 PM	2015-12-01 12:05 PM		Benjamin Wulff	Benjamin Wulff		Fixed and reviewed	RESOLVED
LS-104	put module documentations as MD into bundle so Manual UI can use them		2015-01-21 02:58 PM	2017-12-05 09:12 AM		Benjamin Wulff	Benjamin Wulff		Won't Fix	RESOLVED
LS-89	ConcurrentModificationException on shut down		2014-10-29 04:06 PM	2016-03-09 07:57 AM		Benjamin Wulff	Benjamin Wulff		Fixed and reviewed	RESOLVED
LS-66	File chooser of Configuration Editor		2014-09-	2016-04-		Benjamin	Benjamin		Fixed	

	should have LS current working directory selected initially		16 03:24 PM	28 08:43 PM	Wulff	Wulff		and reviewed	RESOLVED
LS-54	Remove unnecessary native libs from Videofile FrameSource bundle		2014-04-13 09:44 PM	2015-11-04 11:12 PM	Waldemar Smirnow	Benjamin Wulff		Fixed and reviewed	RESOLVED

8 issues

Gadgets can only be viewed in browsers

LectureSight Server - Development Overview

Key	Summary	T	Created	Updated	Assignee	Reporter	P	Status	Resolution
LS-186	Explore Java+UDP: Example UPD-Client & Server		2016-07-18 05:04 PM	2016-07-18 05:04 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-184	Create documentation for LectureSight Server		2016-06-22 03:00 PM	2016-06-22 03:00 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-183	Implement PTZ Service that uses the SRG-300's REST interface		2016-06-22 02:55 PM	2016-06-22 02:55 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-182	Implement VISCAoverIP PTZ Service		2016-06-22 02:54 PM	2016-07-18 05:04 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-181	Implement Gogo-Shell commands for controlling multiple pipelines		2016-06-22 02:53 PM	2016-06-22 02:53 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-180	Implement control service for multiple pipleines		2016-06-22 02:53 PM	2016-06-22 02:53 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved
LS-179	Enable multi-instance ability in relevant services		2016-06-22 02:52 PM	2016-06-22 02:52 PM	Benjamin Wulff	Benjamin Wulff		OPEN	Unresolved

7 issues