LectureSight
Automatic Camera Control in Lecture Recordings

Benjamin Wulff    Alexander Fecke
bwulff@uos.de    afecke@uos.de

University of Osnabrück, Germany

03.04.2013
Table of Contents

1 Introduction
   - Project Overview
   - Motivation

2 Technology
   - System Architecture
   - Camera Setup
   - Video Analysis
   - Camera Control

3 Deployment
   - Scene Regions
   - Scripting
   - Advanced scenarios
Project Overview

Context of the Project

Opencast Community Project
−→ close integration with Opencast Matterhorn
  runs in same OSGI container (Apache Felix)
  camera control activated by recording schedule
  initiated by ETH Zürich ID Multimedia Services
  partner in development of Opencast Matterhorn
  replacing closed-source tracking solution
Project Overview

Context of the Project

- Opencast Community Project
Project Overview

Context of the Project

- Opencast Community Project
  → close integration with Opencast Matterhorn
Project Overview
Context of the Project

- Opencast Community Project
  → close integration with Opencast Matterhorn
  - runs in same OSGI container (Apache Felix)
  - camera control activated by recording schedule
Project Overview

Context of the Project

- **Opencast Community Project**
  - close integration with Opencast Matterhorn
    - runs in same OSGI container (Apache Felix)
    - camera control activated by recording schedule

- initiated by ETH Zürich ID Multimedia Services
Project Overview
Context of the Project

- Opencast Community Project
  - close integration with Opencast Matterhorn
    - runs in same OSGI container (Apache Felix)
    - camera control activated by recording schedule

- initiated by ETH Zürich ID Multimedia Services
  - partner in development of Opencast Matterhorn
  - replacing closed-source tracking solution
Motivation

to cut costs, most lectures are recorded automatically—no camera operator.

B. Wulff, A. Fecke

LectureSight
Motivation

- to cut costs, most lectures are recorded automatically
  → no camera operator
Motivation

- to cut costs, most lectures are recorded automatically → no camera operator
- problem especially in big lecture halls
Motivation

Static camera perspectives
Motivation

Static camera perspectives

1 (cleverly chosen) wide shot
Motivation

Static camera perspectives

1. (cleverly chosen) wide shot

2. close shot on the lectern
Motivation

In a perfect world...
Camera Setup

PTZ Camera

Overview Camera
Video Analysis

Continuously maintained foreground mask

Change Detection & Weighting

Background Model

Object Analysis & Tracking

Foreground Map
Video Analysis

Movie Time!
Video Analysis

Finding a person’s head

\[
\text{head: } \min(|x_{\text{centroid}} - x|) \wedge \max(y)
\]
Video Analysis
Finding a person’s head

\[ P_{\text{head}} : \min(|x_{\text{centroid}} - x|) \land \max(y) \]
Video Analysis
Finding a person’s head

\[ P_{\text{head}} : \min(|x_{\text{centroid}} - x|) \land \max(y) \]

- **K-means Clustering** on person silhouette
- **uneven** number of prototypes
- cluster positions are updated with every frame
Camera Control

PTZ camera is operated by a steering worker that decides about the camera movement: if correction moves are necessary which speeds for pan and tilt should be used.
Camera Control

PTZ camera is operated by a *steering worker* that decides about the camera movement:
Camera Control

PTZ camera is operated by a *steering worker* that decides about the camera movement:

- if correction moves are necessary
PTZ camera is operated by a *steering worker* that decides about the camera movement:

- if correction moves are necessary
- which speeds for pan and tilt should be used
Customizing LectureSight

Adjusting the system to the scene
Customizing LectureSight

Adjusting the system to the scene

Object Tracking Area

Trigger Area

Ignored by Video Analysis
Customizing LectureSight

A simple camera operator script

```javascript
var target = false;

function init() {
    home_pos = new Position(
        Configuration.get("director.home.x"),
        Configuration.get("director.home.y"));
}

function step() {
    if (!target) {
        Camera.setTarget(home_pos);
        target = findClosest(Tracker.trackedObjects, home_pos);
    } else {
        if (target.isTracked())
            Camera.setTarget(target.get("head.center"));
        else if (Tracker.currentTime - target.lastSeen > 2500)
            target = false;
    }
}

function findClosest(objects, pos) {
    ...
}
```
LectureSight in Action

Movie Time!
More advanced scenarios
More advanced scenarios
More advanced scenarios
More advanced scenarios
Thank you!

Benjamin Wulff
University of Osnabrück

bwulff@uos.de

www.lecturesight.org