project AVRiL

Design Specification
Version 1.0
19th Nov 2006

Advisors:
Dr. Sohaib Khan
Dr. Umar Saif

BSc Sproj '07 - Group 7:
Ahmad Humayun
Ozair Muazzam
Tayyab Javed
Yahya Cheema

avril.sproj.com

Note:
This is version 1.0 of the design specification.
Since we are following an iterative SE process this document is subject to revision as we progress through the project.

Section 1: Introduction

Purpose of this section: General background and reference information

1.1 Purpose Of This Document

This Design Specification is

1.2 Scope Of The Development Project

AVRiL will aim to automatically record, using PTZ cameras, university lectures in order to assist distance learning programs. We are assuming that the system would be used inside auditoriums only, in which only one instructor is teaching at a time. We also assume that the instructor will be standing and in motion most of the time, teaching a group of seated students who face away from the camera.

1.3 Definitions, Acronyms And Abbreviations

AVRiL:	Automated Video Recording of Lectures (project name)
TA:	Teaching assistant
PTZ:	Point-tilt-zoom
SDS:	Software Design Specification

1.4 References

SDS Outline (University of Texas at Austin)
[http://www.cs.utexas.edu/~almstrum/cs373/sp05/doc-stds/SDS-outline.html]

IEEE SDS Format
[http://suraj.lums.edu.pk/~cs405y07/IEEE%20SDD%20Format%20-%201998.pdf]

1.5 Overview Of Document

The rest of the document describes the architecture and design of our system.

Section 2 gives a birds-eye view of the system, its different components and how they interact with each other. Section 3 delves deeper and provides a detailed specification to be used when developing the system.

Section 2: System architecture description

Purpose of this section: a broad overview of the system design

2.1 Overview of modules / components

This section will introduce the various components and subsystems of AVRiL.

1. **Video Capturing**
   This component is responsible for capturing a suitable video stream for use in the final video output of the lecture. It includes system components such as a PTZ camera, a camera controller, a video receiver and physical design/direction rules such as camera placement.

   a) **PTZ camera**
      This is needed to record the lecture. The camera will be fixed on a particular location but on that axis, it will be able to carry out pan-tilt and zoom operations.
   b) **Camera controller**
      The controller class is responsible for interfacing with a PTZ camera and providing a suitable interface to direct the camera to a particular point, query its location, turn it off and on etc…
   c) **Video Receiver**
      This is needed to properly capture the streaming input from the camera. It will include functions such as display video, starting/stopping and pausing recording and splitting the video into frames for output to other modules.
   d) **Physical Design Rules**
      Although not a part of the automated system, rule definitions regarding the best places for mounting a camera, camera height based on the what the camera is capturing (e.g. Lecturer, audience, board) etc.. need to be mentioned in the implementation of the system

2. **Lecturer Tracking**
   The goal of lecturer tracking is to effectively capture the lecturer's movements without any obstructions to the lecture environment. We aim to do this by keeping a camera focused on the lecturer at all times and tracking lecturer movements using a combination of background subtraction and a motion tracking. The lecturer tracking component will interface with the video receiver module of the capture component for video input and will output to the camera controller.

a) Background Subtraction

We can take advantage of the fact that most lecture environments will have a constant background and dais and that that we will have access to capture this background when there are no external objects present.

i. Mosaics

The mosaic building module is responsible for building a collection of images of the background that we can use later for background subtraction. This module will include algorithms to capture pictures at a proper location and zoom level and to account for overlap in images.

ii. Subtraction

Given a particular frame with its coordinates, this module is responsible for aligning it with a suitable image from the mosaic using a proper transformation and returning a background-subtracted frame to the user.

b) Motion Tracking

Given a video stream, this module will detect objects that have moved between subsequent frames and will return a direction and an approximate measurement of the movement.

3. Audience Localization

This component is responsible for locating a talking person in the audience and interfacing with capture module to get footage of audience participation. This can be done through various means such as having the audience using a microphone equipped with a transmitting beacon or using sound-source localization techniques.

4. Slide Transition Detection

This component is responsible for detecting transition events on a slide-show presentation. It will output timestamps of all transitions that took place during a lecture.

a) PowerPoint Plug-in

This plug-in will integrate with Microsoft PowerPoint and time-stamp all slide transitions.

b) Slides Capture

This module will capture a screenshot of the appropriate slide at each time-stamp.

5. **Direction Mixing**
   The direction component is responsible for integrating direction rules for mixing the different video streams and PowerPoint Slides in order to produce a video output with high production quality.

   a) **Rule Manager**
      This component will allow users to define rules that decide on which stream should be added to the mixer, and turn these rules on and off.

   b) **Mixer**
      This component is responsible for directing the proper stream to the encoder for the final video output.

6. **Encoder**
   The encoding component will take an input raw video stream and encode it into a suitable output format.

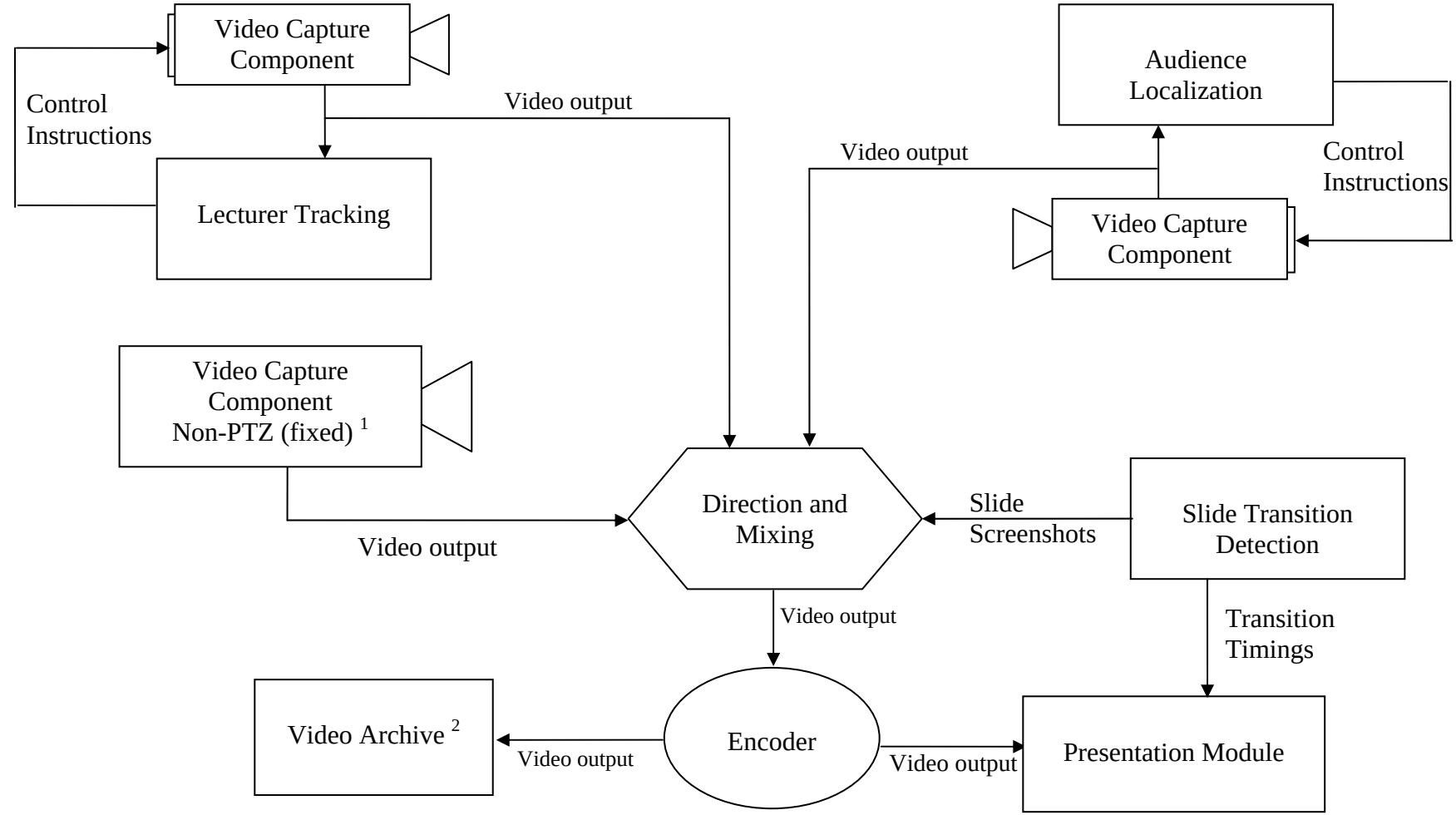
7. **Presentation Module**
   This component will take the encoded video and the slides and synchronize them together using the timestamp information for a rich user experience.

2.2 Structure and Relationships

The video capture component used for capturing the lecturer will interface with the lecturer tracking component and similarly the audience tracking capture component will interface with the audience localization component.

All video output and slide screenshots will go to the Direction and Mixing module which will decide between them to produce a final high-quality video output based on direction rules. A diagram is given on the next page that shows a high-level structure of the system and the relationships between the different components.

Structure and Relationship Diagram



[1] can be used for capturing board or for a wide-angle camera to capture dais

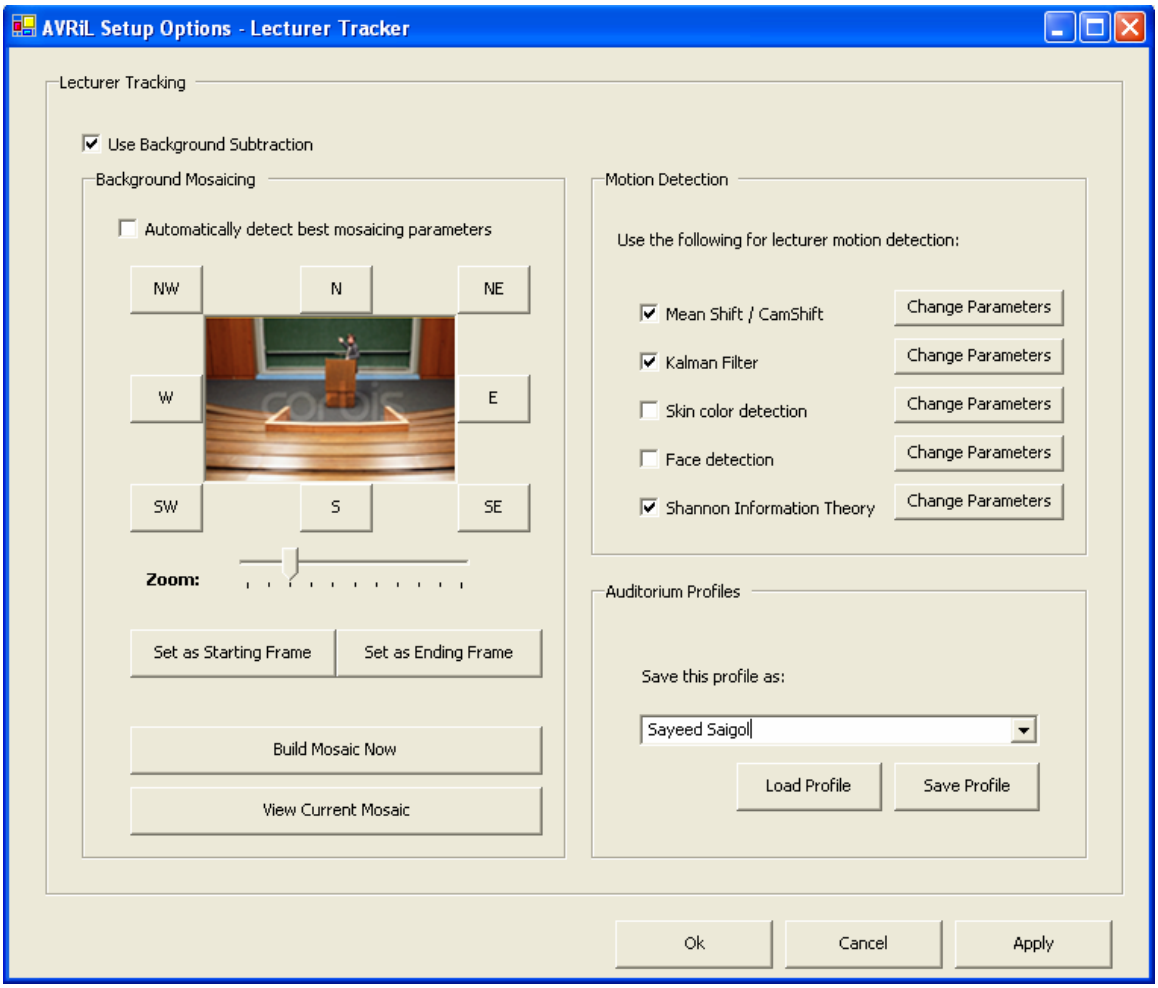
[2] e.g. use may be for storing in video library or for airing on TV

2.3 User interface issues

Main principles of the product's user interface

The system will not have any interface with the lecturer or with the audience. An interface will be provided for a *system handler* (as defined in the requirements specification) to setup the system and fine tune a few variables. Also, the viewers of the lecture will have an interface through which they can view the recorded output.

1. **System Handler Interface**
   This will be used by an administrator for initially setting up the system. A sample screen of how the setup of the Lecturer Tracker component might be is shown below:



2. **Interface for presenting the lecture to the viewer**
   This will be the interface presented to a remote attendee, a sample of what this may look like is given below:

* Courtesy Liu, Qiong et al., Microsoft Research "Automating Camera Management for Lecture Room Environments"

Section 3: Detailed description of components

Purpose of this section: This will serve as the basic reference when implementing the system. Since this is version 1.0 and we are working on a project that involves experimentation and research work, we are not currently in a position to give a very detailed design and hence this version only includes a very basic description of all features. Version 2.0, the next major revision after we've finalized the design will be the basis we will use as reference during the implementation phase.

3.1 Detailed Description of Functional Requirements

3.1.1 Component Design Template

This section gives the format that we will be using in the following sections for defining the functional requirements of the system.

Purpose: Description of the functional requirement and its reason(s)
Inputs: Which inputs; in what form/format will inputs arrive; what sources will be used and how can the inputs vary
Outputs: The form, shape, destination, and volume of the output; output timing; process by which the output is stored

3.1.2 Camera Controller

Purpose: To provide a basic set of methods and properties for interfacing with a camera. This will be an interface/abstract class which a specific implementation class will implement for each type of camera used.
Inputs: Inputs will be the direction controls for panning and tilting and zoom controls. Input may also be an instruction to query the status of the camera.
Outputs: Output will be results for query commands and success status of movement commands.

Resources: A PTZ camera which has a link to the component.

3.1.2 Lecturer Tracker

Purpose: To keep track of the lecturer so a camera can stay focused on him/her.
Inputs: Video Sequence from camera designated to keep track of lecturer.
Outputs: Pan-Tilt direction to which the camera should move based on lecturer movement direction and zoom level based on lecture movement frequency.

3.1.3 Audience Localization

Purpose: To locate attendees who take part in the lecture.
Inputs: Audio sequence of active attendee
Outputs: Directions to the camera controller for pointing to the active attendee

3.1.4 Slide Transition Detection

Purpose: To record slide transitions and capture screenshots of the slides when transitions happen.
Inputs: Lecture slides in PPT format and PowerPoint events triggering slide transition
Outputs: Timestamps of transitions and screenshots of slides at that time

3.1.5 Direction Mixing

Purpose: To select the most appropriate shot to put in the output video
Inputs: Direction rules, different video streams to choose from, slide screenshots and transition timestamps
Outputs: A high-quality video similar to one made by a professional director.

3.1.5 Encoder

Purpose: To encode raw video input in a suitable format for playback and storage.
Inputs: Raw video input
Outputs: Encoded video output

3.1.6 Presentation Module

Purpose: To provide a good interface to remote users for experiencing the lecture.
Inputs: Lecture video, PowerPoint Slides, slide transition timestamps
Outputs: Presents a screen with the video playing, and slides transition in sync on the side.

Section 4: Reuse and relationships to other products

We will be using the Intel® ***OPENCV*** library and its collection of computer vision algorithms wherever they can fit in our project rather than writing our algorithms from scratch.

The system is designed to be highly modular and implementation independent and hence changing components or re-writing them will not break code. For example getting a new camera and re-writing a camera controller class will not break code but will fit seamlessly with the other components. Similarly, you can plug-in different algorithms for lecturer tracking, use a combination of background subtraction and motion tracking or just use motion tracking without breaking code.

Section 5: Design decisions and tradeoffs

We are still experimenting with our options and have not made any significant design decisions yet. Expect to see more of this in version 2.0 of this document.

Section 6: Pseudo code for components

Not written yet :P