

Automatic Projector Tilt Compensation System

Ganesh Ajjanagadde James Thomas Shantanu Jain

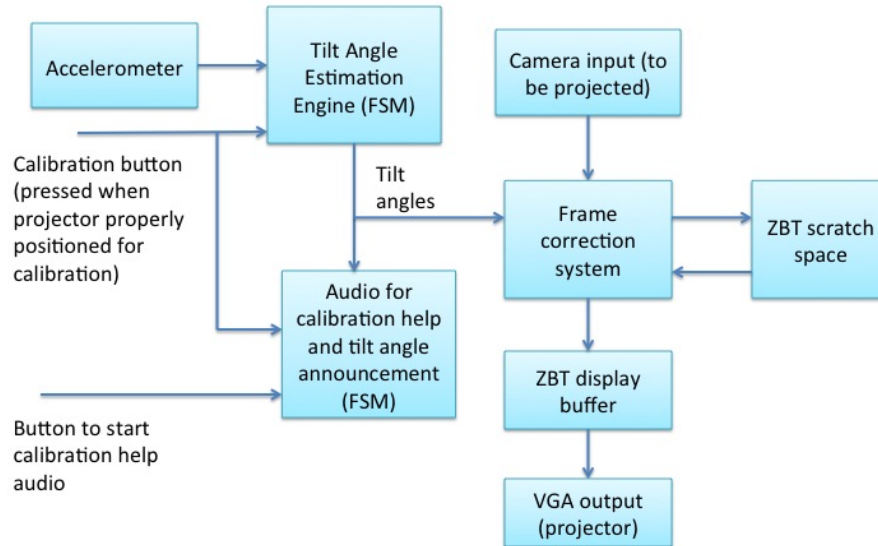
October 30, 2014

1 Introduction

Due to the advances in semiconductor technology, today's display projectors can incorporate fairly sophisticated digital processing algorithms for various enhancements to the visual appearance. Moreover, there is an increasing prevalence of portable projectors that benefit from fast, automated setup. One desired functionality is keystone/tilt correction. This means that even when the projector is tilted, the projector is able to "pre-warp" the image so that its output appears unskewed. In this project, we will be projecting the output of a camera, and will connect the camera's output to the FPGA board, and the FPGA board's VGA output to the projector. We will mount an accelerometer on the projector and measure its signals to determine the projector's tilt angles on two axes. We will run algorithms on the FPGA that correct the camera output based on the tilt angles and produce the results at the VGA output for the projector to display. We will calibrate the accelerometer through a special calibration mode. As a stretch goal, we will also attempt to use the camera combined with the accelerometer itself to calibrate by projecting a known image, such as a white rectangle. This should result in a gain to the quality of keystone/tilt correction, and one of our goals is to qualitatively observe and quantify such differences in quality. Finally, we will design digital logic that produces a voice with help commands, and also lets the user know of the new tilt angles whenever they are changed.

2 Implementation

2.1 Block Diagram



2.2 Modules

Accelerometer

- Includes logic to convert signals from the accelerometer (whether in analog or SPI digital format) to normalized digital values that can be fed to the tilt angle estimation engine

Tilt angle estimation engine

- When calibration button is pressed, updates state to set the current values from the accelerometer as the reference values for an untilted projector
- By comparing the values from the accelerometer to the stored reference values, produces estimates of the angles of tilt in the two axes of interest

Audio

- When the calibration help button is pressed, goes into state in which it periodically plays a recording instructing the user to adjust the projector so that its output looks unskewed and then hit the calibration button
- When the calibration button is pressed, exits the help state if it is in it
- When a change in the current angles of projector tilt is detected, plays recordings that specify the angles
- Audio samples are stored on compact flash

Frame correction system

- Based on the current angles of projector tilt, adjusts an incoming frame from the camera using a linear transformation (implemented with a matrix multiplication) so that when it is projected it appears unskewed
- Uses the ZBT scratch space to store each frame from the camera (reading pixels from the stored frame for the matrix multiplication) and possibly the results of intermediate computations
- Stores the output frame from the linear transformation in a display buffer (also ZBT)
- An auxiliary module will feed the pixels in the display buffer to the VGA output at the appropriate clock edges

3 Timeline

There is five weeks time in which to complete the project. In these five weeks, we envision four major stages to the project:

1. Initial attempt at implementing each block
Time: 2 weeks
In this stage, each team member will create an initial implementation of modules he is responsible for. He is expected to have his modules working as a standalone component. Each module will include simple test benches in order to facilitate this goal.
2. Module Integration; System Building
Time: 2 weeks
This stage will connect the individual modules from stage 1 together to form the initial system. This is expected to be the most hands-on and collaborative stage, as mistakes will often have a domino effect on the system. In the worst case, we may have to modify parts of our system design or change the behavior of some modules in order to compensate; in lesser situations, bugs may be uncovered and fixed. This is expected to be the lengthiest stage.

3. Intensive debugging

Time: 1 week

After the system has been put together and functions minimally, the team will focus on debugging. The goal of debugging is to make the system robust. Testing will include the common usage scenarios as well as edge cases, and a limited set of malicious inputs.

4. Stretch Goals

Time: As available

Once the system satisfies the test coverage plan, the team will focus on adding additional features to the system, as constrained by time. These possible additional features are described below.

4 Individual Responsibility

Ganesh

- Algorithm - Identifying the appropriate algorithm, considering factors of complexity, hardware constraints, and implementability.
- Transform Module - Implementing the algorithm, with input parameters

James

- Accelerometer - Identifying the appropriate type and model, interfacing with the labkit.
- Calibration module - Entering calibration mode, collecting data, making it available as necessary.

Shantanu

- Audio output - Identifying necessary audio outputs, triggers for audio. Recording audio, transforming it into the appropriate format.
- Audio module - loading audio onto the labkit, standardizing trigger parameters.
- Test Setup - attaching accelerometer to projector, wiring to labkit.

5 Stretch Goals

- Camera-based adjustment

The camera-based adjustment would augment the accelerometer by displaying a test pattern output to the projector and using an edge detection/corner detection algorithm to determine the skew caused by the projector's tilt. It would be able to correct the projector image in a wider variety of scenarios, including variable distance to the screen. This is difficult as it is complex and time consuming to interface the camera with the labkit and implement these additional algorithms.

- External display of transformed image
View the transformed image on a non-distorted screen, such as a PC LCD monitor.