NovaToast SmartVision Project Requirements

Jacob Anderson
William Chen
Christopher Kim
Jonathan Simozar
Brian Wan

Revision History
v1.0: Initial creation of the document and first draft.
Introduction
Smart Vision is about making a “smart home” using FLIR’s mobile thermal sensor, the Lepton. The Lepton will be used to monitor an area in a house, and send notifications based on certain events, such as a person falling, frost on a window, or if an object is blocking the camera. The interface will be controlled by an external web application.

With each feature there are edge cases we must consider. If we implement fall detection, we need to be able to differentiate between an actual fall or something like a pet sleeping on the floor. In the case of the temperature monitor we need to distinguish when an area will remain at a certain temperature, such as a fire starting in your house and ignore something like a lit candle.

Aside from these features, there are general challenges we must face. One in particular be detecting if the camera is blocked. If the camera is blocked, no events can be detected. Another issue is if the camera shuts off unexpectedly. A similar problem is if the network connection is broken. These two cases would ultimately result in Smart Vision being unusable. These are several of many possible issues that we may face and we must decide how to handle and account for each appropriately.

The problems that can occur in homes have solutions today in the form of many different products, but not one single system. Currently, there are no products that automate the home as Smart Vision does, but many are certain to follow. Smart Vision will be an all-in-one system that can serve multiple purposes in many applications.
Glossary of Terms

- **GPIO**: General purpose input-output; a bus system for a wide range of applications.
- **I2C**: A 2-way serial computer bus used for connecting low-speed devices to larger systems.
- **Lepton**: A mobile thermal imaging sensor developed by FLIR.
- **OpenCV**: An efficient image processing library available for many languages. Used for real-time applications.
- **Raspberry Pi**: A low cost, miniature computer system offering an ARM-based processor.
- **SPI**: Serial Peripheral Interface; allows synchronous serial communication between devices.
System Architecture Overview
The initial setup will be with a Raspberry Pi Model B connected by GPIO to the Lepton module. The Raspberry Pi is a small computer that contains a stripped-down Linux distribution. It has a Broadcom system-on-a-chip with a 700MHz ARM processor, 512 MB of shared RAM, and a 100 Mb ethernet port. There is a concern that the Pi will not be powerful enough to match the software specifications in which case, either using a different Raspberry Pi model or a more powerful Windows machine will be used as a backup.

The Lepton is a longwave infrared imaging sensor. The particular model used has no shutter, so it works from calibration, meaning, once calibrated, it will be able to detect temperatures colder or hotter than the object which it was calibrated to. With the potential addition of a shutter, it will be able to be accurately calibrated every time. This way, it will be able to detect absolute temperatures rather than relative ones. The Lepton sends the image as a stream of bits over a serial connection which is then pieced together in the Pi.

The interface between the Pi and the Lepton uses SPI and I2C. SPI feeds data into the Pi, while I2C is what the Pi uses to issue commands to the Lepton.

The Pi will be controlled by a web interface built with Javascript, Java, and the Spring MVC framework. This application will first scan the network for the Pi, then connect to it. Once the connection is made, there will be a web-based GUI that the user can open and receive input from the Pi and issue commands.

The Pi will be running OpenCV, an extensive image processing library. This will be written in C/C++ for maximum performance. OpenCV will allow the Pi to take the image provided by the Lepton and analyze it for regions of interest.
Requirements

*Use Cases:*
- Monitor specific area for temperature and raise a flag at certain temperatures
- Raise a flag when someone has fallen and hasn’t stood up after certain amount of time
- Send a notification if camera is off
- Send an alert if something is blocking the camera for extended period of time

*User Stories:*
- Be able to configure Smart Vision from anywhere with internet access.
- Add user authentication for configuration through web app
- Set phone number & email for alert.
- Select a certain spot to monitor
- Set certain temperature(s) to send alert
- Set how long a person is on the floor before sending alert
- Toggle receiving notifications
Appendices

- Lepton
- Raspberry Pi
- SPI, I2C (communication protocols)
- OpenCV
- CUTE, JUnit, QUnit (unit testing frameworks)
- Spring MVC
- Trello
- GitHub
- SourceTree
- Eclipse
- IntelliJ
- Sublime Text