Team11:

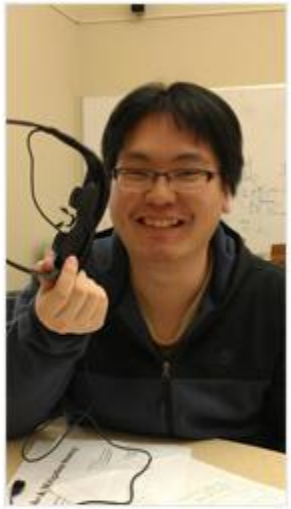
Reality Translator

Eugene Bin, Liang Yeet Ooi, Samantha Tan, Jungmin Yoon

Team Members

Eugene Bin

jebroni99@gmail.com



Liang Yeet Ooi

looi@andrew.cmu.edu

Samantha Tan

samanthatan@cmu.edu

Jungmin Yoon

cheeki101@gmail.com

Our Websiite:
http://www.ece.cmu.edu/~ece549/spring12/team11/

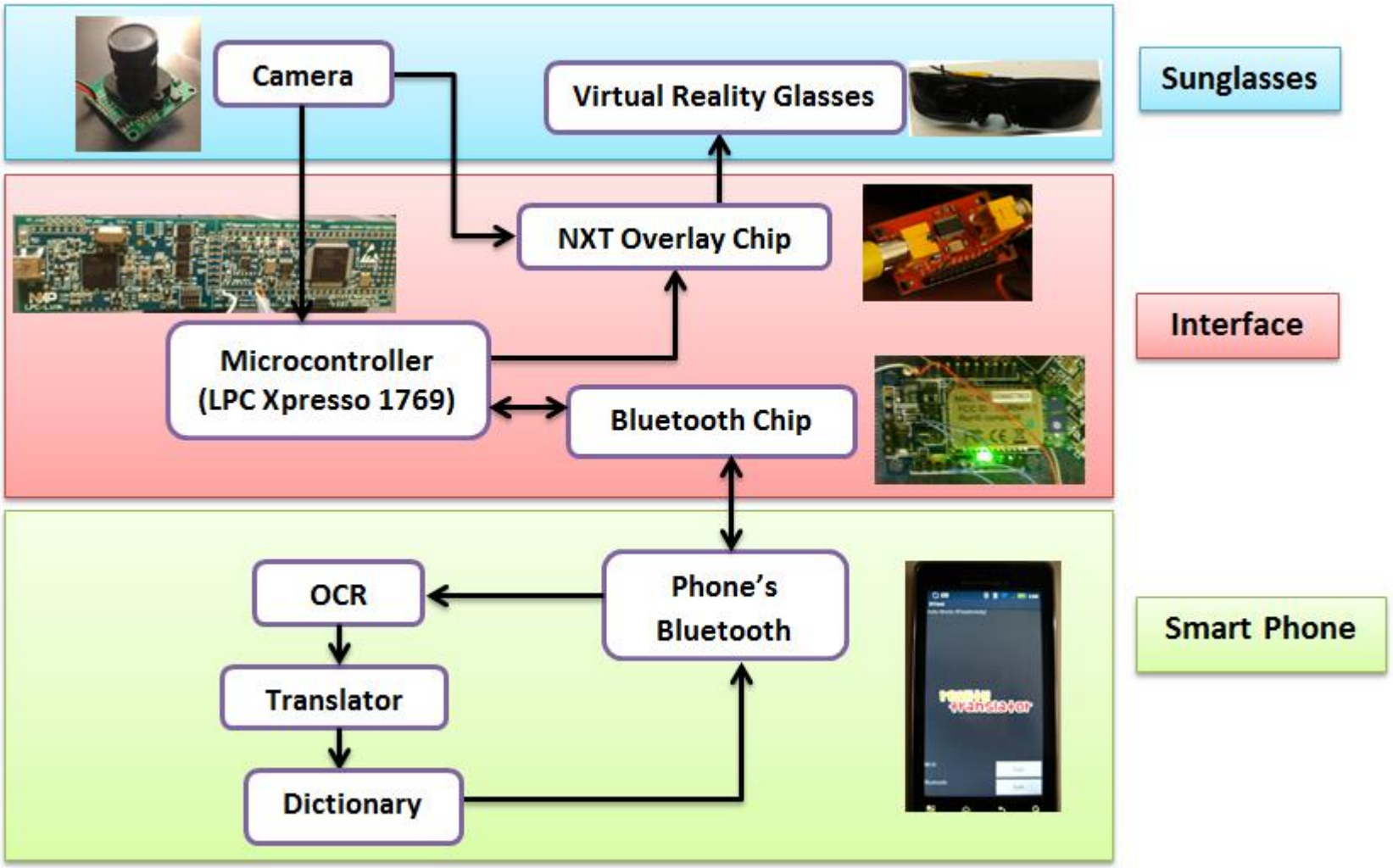
Concept

- Imagine putting on sunglasses and suddenly being able to understand any language you see, Reality Translator(RT) does exactly that! RT will integrate virtual reality sunglasses, a camera, and a WiFi/Bluetooth-enabled smart phone to translate languages and to emulate augmented reality.

Goals

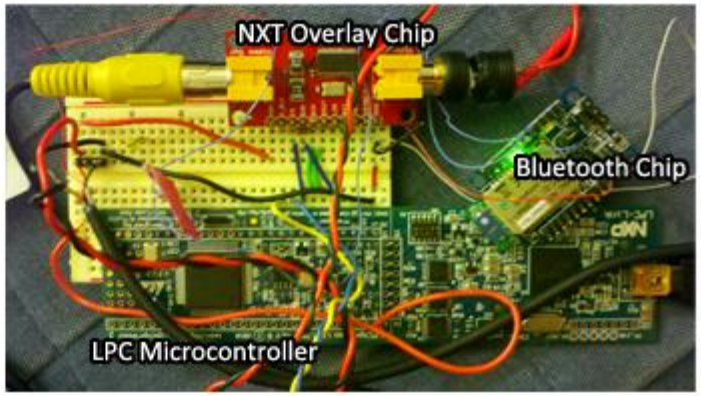
- Reality Translator translates and renders translations on the glasses in near instant real time.
- Reality Translator emphasizes on ease of use – allowing for portability and inconspicuous hands free use
- Timing Requirement: The delay between processing the image and rendering translated text is less than 2 seconds.

Architecture



Components

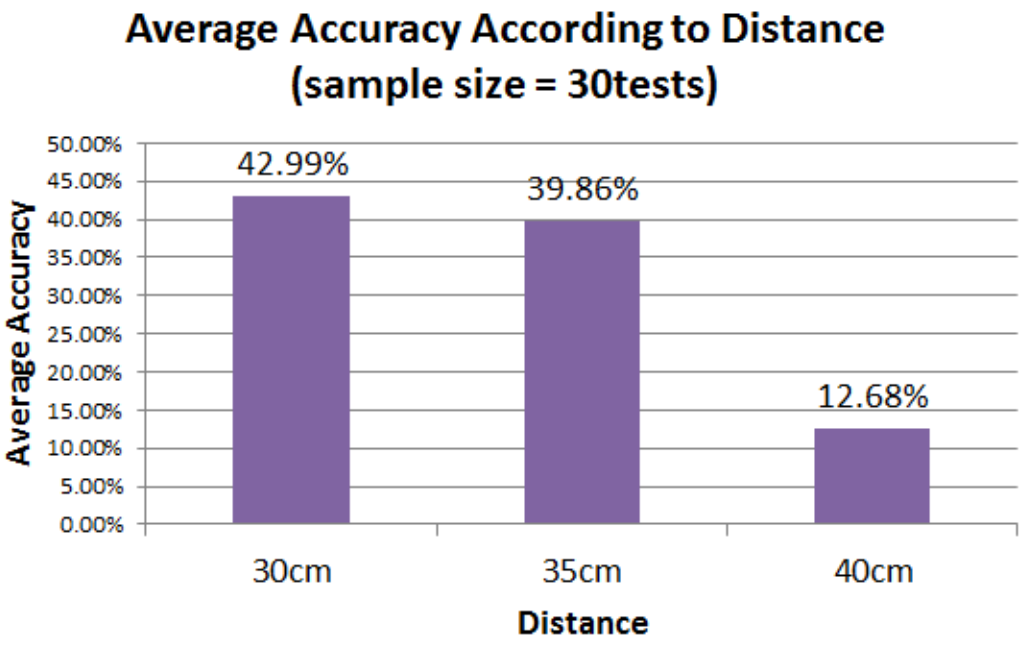
- NTSC to Raw converter
  - MAXIM MAX9526EVKIT+
- TTL NTSC JPEG camera
- LPC Microcontroller
- NTSC Overlay
  - Using MAX7456
- Vuzix Wrap 920 Virtual Reality Glasses
  - 640x480 pixels
  - Truecolor
- Bluetooth Dongle
  - 1.5 Mb/s
- OCR
  - Tesseract
- JAVA
  - Android app

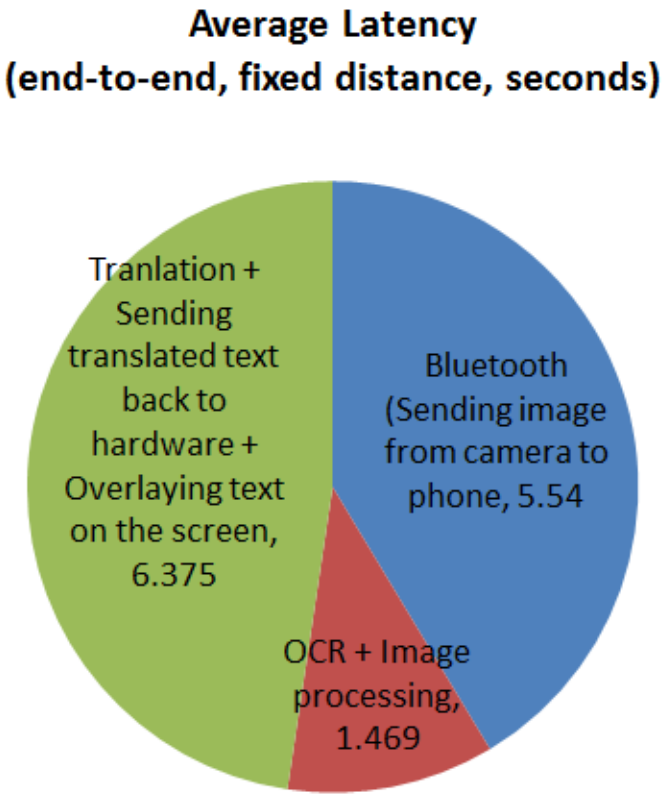


Experiment

- Tests
  - Power - Directly measured as product runs
    - The device must be usable for an acceptable length of time(2hrs)!
  - Accuracy - How many words is the software able to correctly recognize
    - The translations + character detection must be correct!
    - Tested on a small set of images to measure and improve accuracy
  - Latency - Time between camera beginning sending image and glasses receiving translated text.
    - Service must be as fast as possible for maximum enjoyment!

Experiment Results





Insight from Measurements

- Latency increased from idle to image processing case is noticeably small
  - Latency from image processing still ~1 second
- Latency from Bluetooth transfer is significantly big
  - Makes up for 40%+ of the total latency
- Accuracy varies with font, size, and color of text
  - Overall accuracy rises as image size grows, with an upper bound
  - Accuracy >90% for black san serif fonts
  - <20% for anything different
  - Edge detection worsens accuracy significantly
- Power consumption reasonable (approx. 2 hours)
  - No changes needed

Performance

- Minimizing Bluetooth latency
  - Increased baud rate for data transfer
    - Attempt to speed up transfer time
    - However no significant improvement
- Improving accuracy
  - Increase image by 100% + sharpen
    - Results indicate larger images increases OCR
    - Limited enlargement by android's stack space
  - Ignore all symbols prior to applying dictionary
    - Most of the text from OCR were random symbols that are rare in everyday text, so parse them out

Open Issues

- Current issues:
  - Camera freezes as it's transferring data
  - OCR... open ended problem
- Extra features wishlist:
  - Add Lucas Kanade tracker or gyro
  - More language support (ie. Japanese)
  - A good method of packaging

Conclusion

- Things we learned
  - Developing on LPC controller
  - Bluetooth protocol
  - Text overlay
  - Android development
  - Image processing
- Accomplishments
  - Successfully integrating software with hardware
- What would we have done differently
  - Use different hardware
    - Camera + overlay chip not optimal for our project
  - Invest more time on more efficient image processing algorithms