Our Team

Istvan Kreisz
EE

Keith Miller
CSE

Trevor Berry
EE

Zlatan Aksamija
Advisor
Imagine..
Pegasus-21

Webcam → Turret → Transducer array → User

language, volume control

iPhone app
System Block Diagram
What’s new?

- Data Server:
  - Research Rig

- iPhone App:
  - Redesigned, more user-friendly UI
  - Under-the-hood improvements: error handling, multithreading

- Power Supply:
  - Redesigned with a PCB
  - New components with higher current carrying capacity

- Turret:
  - 3D printed chassis and Soundlazer mount
Specifications

- Rotation Speed: 17°/sec
- Range: 10 ft
- Maximum Sound Intensity (Indoors):
  - On Axis: 65 dB
  - 5 ft Off-Axis: 52 dB
- Power Input: 2 x 120 V (Power Supply, Computer)
Demo

PEGASUS 21
## Demonstration Guide

<table>
<thead>
<tr>
<th>Server-Side</th>
<th>Client-Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Turn on server</td>
<td>1. N/A</td>
</tr>
<tr>
<td>2. Accept Wi-Fi connection</td>
<td>2. Connect to “pegasus-server” Wi-Fi</td>
</tr>
<tr>
<td>3. Wait</td>
<td>3. Open app</td>
</tr>
<tr>
<td>4. Wait</td>
<td>4. Create new account*</td>
</tr>
<tr>
<td>5. Accept TCP connection</td>
<td>5. Login</td>
</tr>
<tr>
<td>6. Wait</td>
<td>6. Take 15 sec video*</td>
</tr>
<tr>
<td>7. Train neural network*</td>
<td>7. Wait for training*</td>
</tr>
<tr>
<td>8. Target Acquisition</td>
<td>8.</td>
</tr>
</tbody>
</table>

---

N/A: Not applicable

* Indicates a step that may vary based on specific conditions or requirements.
## Demonstration Guide

### Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

### Client-Side

1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
Demonstration Guide

Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

Client-Side

1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
Demonstration Guide

Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

Client-Side

1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
# Demonstration Guide

<table>
<thead>
<tr>
<th>Server-Side</th>
<th>Client-Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Turn on server</td>
<td>1. N/A</td>
</tr>
<tr>
<td>2. Accept Wi-Fi connection</td>
<td>2. Connect to “pegasus-server” Wi-Fi</td>
</tr>
<tr>
<td>3. Wait</td>
<td>3. Open app</td>
</tr>
<tr>
<td>4. Wait</td>
<td>4. Create new account*</td>
</tr>
<tr>
<td>5. Accept TCP connection</td>
<td>5. Login</td>
</tr>
<tr>
<td>6. Wait</td>
<td>6. Take 15 sec video*</td>
</tr>
<tr>
<td>7. Train neural network*</td>
<td>7. Wait for training*</td>
</tr>
<tr>
<td>8. Target Acquisition</td>
<td>8.</td>
</tr>
</tbody>
</table>
# Demonstration Guide

## Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

## Client-Side

1. **N/A**
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
# Demonstration Guide

## Server-Side

1. Turn on server  
2. Accept Wi-Fi connection  
3. Wait  
4. Wait  
5. Accept TCP connection  
6. Wait  
7. Train neural network*  
8. Target Acquisition  
9. Target Tracking  
10. Change song and volume

## Client-Side

1. N/A  
2. Connect to “pegasus-server” Wi-Fi  
3. Open app  
4. Create new account*  
5. Login  
6. Take 15 sec video*  
7. Wait for training*  
8.  
9.  
10. Change song and volume
## Demonstration Guide

<table>
<thead>
<tr>
<th><strong>Server-Side</strong></th>
<th><strong>Client-Side</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Turn on server</td>
<td>1. N/A</td>
</tr>
<tr>
<td>2. Accept Wi-Fi connection</td>
<td>2. Connect to “pegasus-server” Wi-Fi</td>
</tr>
<tr>
<td>3. Wait</td>
<td>3. Open app</td>
</tr>
<tr>
<td>4. Wait</td>
<td>4. Create new account*</td>
</tr>
<tr>
<td>5. Accept TCP connection</td>
<td>5. Login</td>
</tr>
<tr>
<td>6. Wait</td>
<td>6. Take 15 sec video*</td>
</tr>
<tr>
<td>7. Train neural network*</td>
<td>7. Wait for training*</td>
</tr>
<tr>
<td>8. Target Acquisition</td>
<td>8.</td>
</tr>
</tbody>
</table>
Demonstration Guide

Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

Client-Side

1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
Demonstration Guide

**Server-Side**
1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

**Client-Side**
1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8.
9.
10. Change song and volume
# Demonstration Guide

## Server-Side

1. Turn on server
2. Accept Wi-Fi connection
3. Wait
4. Wait
5. Accept TCP connection
6. Wait
7. Train neural network*
8. Target Acquisition
9. Target Tracking
10. Change song and volume

## Client-Side

1. N/A
2. Connect to “pegasus-server” Wi-Fi
3. Open app
4. Create new account*
5. Login
6. Take 15 sec video*
7. Wait for training*
8. 
9. 
10. Change song and volume
## Development Cost

<table>
<thead>
<tr>
<th>Part</th>
<th># of pieces</th>
<th>Cost per piece</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servo</td>
<td>2</td>
<td>$9</td>
<td>$18</td>
</tr>
<tr>
<td>Power Supply Parts</td>
<td>1</td>
<td>$80</td>
<td>$80</td>
</tr>
<tr>
<td>3D Printing</td>
<td>4</td>
<td>$10</td>
<td>$10</td>
</tr>
<tr>
<td>Desktop Rig</td>
<td>1</td>
<td>$3000</td>
<td>$3000</td>
</tr>
<tr>
<td>Webcam</td>
<td>1</td>
<td>$40</td>
<td>$40</td>
</tr>
<tr>
<td>SoundLazer</td>
<td>1</td>
<td>$250</td>
<td>$250</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$3398</strong></td>
</tr>
</tbody>
</table>
# Production Cost

<table>
<thead>
<tr>
<th>Part</th>
<th># of pieces</th>
<th>Cost per piece</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servo</td>
<td>2</td>
<td>$6</td>
<td>$12</td>
</tr>
<tr>
<td>Power Supply Parts</td>
<td>1</td>
<td>$25</td>
<td>$25</td>
</tr>
<tr>
<td>3D Printing</td>
<td>4</td>
<td>$5</td>
<td>$5</td>
</tr>
<tr>
<td>TX2</td>
<td>1</td>
<td>$400</td>
<td>$400</td>
</tr>
<tr>
<td>Webcam</td>
<td>1</td>
<td>$9</td>
<td>$9</td>
</tr>
<tr>
<td>SoundLazer</td>
<td>1</td>
<td>$150</td>
<td>$150</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$601</strong></td>
</tr>
</tbody>
</table>
Questions...
Backup Slides
System Overview

- Users take a short video clip of their faces using our iPhone app
- This video is used to train our neural network to recognize a face
- Once trained, the turret mounted webcam searches for and tracks the user, focusing the audio beam in their direction
- Users use the iPhone app to select the language and control the volume of the audio
Some Background: Acoustic Directionality

-Directionality of sound correlates with the size of speaker compared to wavelength. The formula for far field directivity of a flat circular piston in an infinite baffle is

\[ p(\theta) = \frac{p_0 J_1(k_a \sin \theta)}{k_a \sin \theta} \]

where...

- Piezoelectric transducers approximate the ideal acoustic piston

- Directionality is achieved by maximizing the piston radius and minimizing the wavelength

(“Loudspeaker” n.d.)
Sound From Ultrasound

Technology to create directional, highly focused soundbeams:

-An array of transducers emits ultrasonic carrier waves in the direction of target

-Air, as a nonlinear medium, acts as a demodulator (no need for receiver)

-When the modulated wave reaches a person the modulated sound becomes audible

(Nema, “Ultrasonic Directive Speaker”)
Sound From Ultrasound

[Diagram showing the process of sound from ultrasound, including an audio input, modulation scheme, ultrasonic transducer array, and interaction with air.]

http://www.soundlazer.com/what-is-a-parametric-speaker/
Data Server - Preexisting Technologies

- OpenCV: computer vision and media manipulation library (Python)
- OpenFace: facial recognition project
- Torch: scientific computing Framework (Lua)
- SciKit-Learn: machine Learning library (Python)
- dlib: highly optimized machine learning
- FFMPEG: converts video formats
- Basic socket and serial libraries: communication to other subsystems
Universal Power Supply - Block Diagram
Universal Power Supply - Circuit Diagram
Voltage Ripple at 1 Ampere
Voltage Ripple at 2 Amperes (Max Current)

V_{ripple} = 1.2 \text{ Vpk-pk} @ 120Hz
Universal Power Supply - Circuit Redesign
Data Server - Overview

- Receives user information from the iPhone application
- Communicates over serial with the turret to run the facial recognition and determine the appropriate servo positions
- Multiprocessing enables us to run the iPhone server process separate from the turret process
- Utilizes processor affinity
- Trains/stores neural network
iPhone App - Overview

• Provides a user interface to interact with the system
• Has two main functionalities:
  • Allows the user to train the neural network by sending a video to the server
  • Allows the user to control the volume and language of the audio
iPhone App - Specs

▪ Development: Xcode 8.2, Swift 3

▪ TCP connection with data server

▪ Username, language and volume settings are sent as simple strings

▪ Video file is sent to the server in .mov format

▪ Multithreaded: communicating with the server runs off the main thread

▪ Used frameworks: SwiftSocket, CoreData, MobileCoreServices,
  SwiftSpinner
Universal Power Supply - Overview

- Linear Regulating Power Supply
- Capable of supplying 30 Watts
- Two voltage levels
- Low noise and low ripple
Pegasus Turret - Overview

- Controls direction of the camera & audio beam
- Locks and tracks a specific user
Pegasus Turret - Functionality

- Webcam sends live video to server
- Server sends back updated X & Y position to Arduino
- Arduino turns servos to the Updated position
- Arduino and Soundlazer are powered from Power Supply
Pegasus Turret - Specs

- Serial connection between Arduino and TX1
- Serial connection between Webcam and TX1
- Rotation angles:
  - In X direction: 0 - 180
  - In Y direction: 0 - 130
- PWM signal from arduino controls the servos
- Used library: “Servo.h”: Allows easy communication between the Arduino and the servos
Pegasus Turret - What’s new?

- 3D printed chassis and Soundlazer mount
### Soundlazzer - Measured Performance

#### SPL at 7 Feet

<table>
<thead>
<tr>
<th></th>
<th>Indoors</th>
<th>Outdoors</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Axis</td>
<td>65dB</td>
<td>63dB</td>
</tr>
<tr>
<td>5ft</td>
<td>52dB</td>
<td>48dB</td>
</tr>
</tbody>
</table>

#### SPL at 30 Feet

<table>
<thead>
<tr>
<th></th>
<th>Indoors</th>
<th>Outdoors</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Axis</td>
<td></td>
<td>55dB</td>
</tr>
<tr>
<td>5 ft.</td>
<td></td>
<td>47dB</td>
</tr>
</tbody>
</table>