IPRO 305

Descriptive Virtualized Reality Environment for Visual Guidance

Team

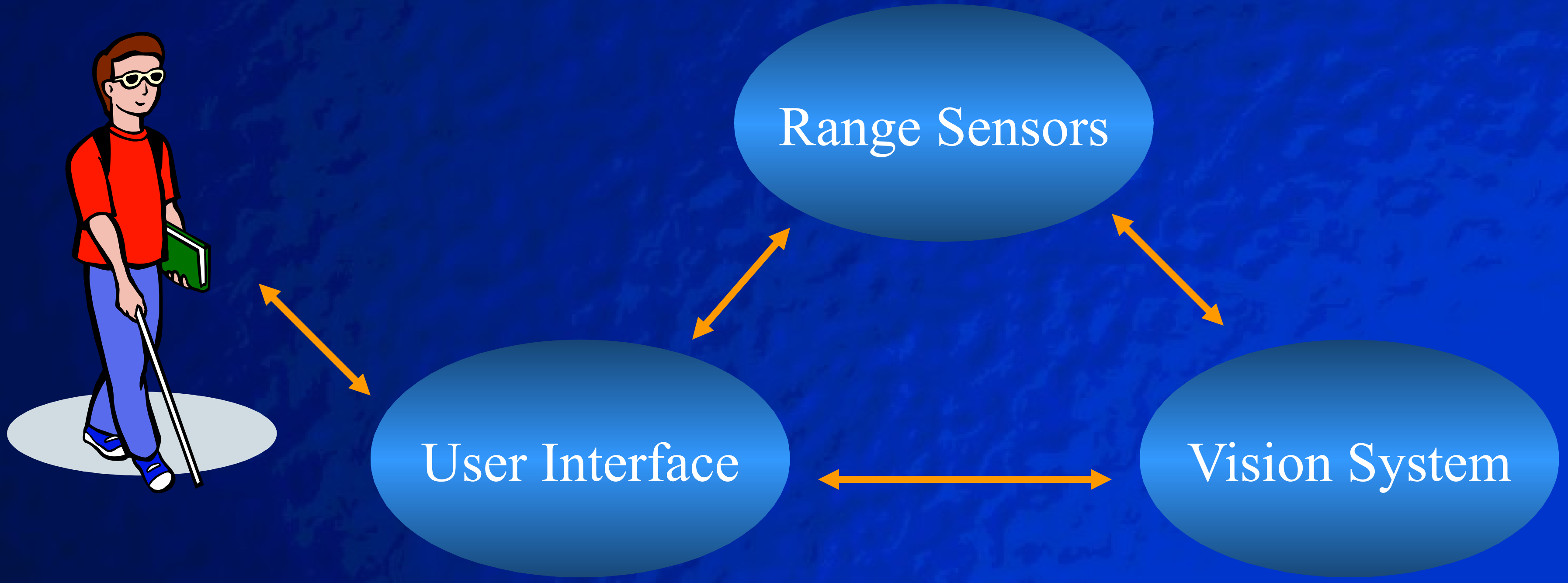
- Team Members
  - Akta Agheda
  - Chirag Bhatt
  - Moummar Bhatty
  - Safiya Bhojawala
  - Aziz Bodal
  - Taewhan Kim
  - Arun Mathew
  - Luqman Soorma
  - Prasad Ullal
- Advisor
  - Prof. Gady Agam

Introduction

- Need for this system
  - Provide a degree of independence to visually impaired
  - Safety
- Strategy
  - Conducted research
  - Divided into sub groups

System Overview

- Three modules communicate with each other through a network socket interface



IPRO 305

Obstacle Avoidance System

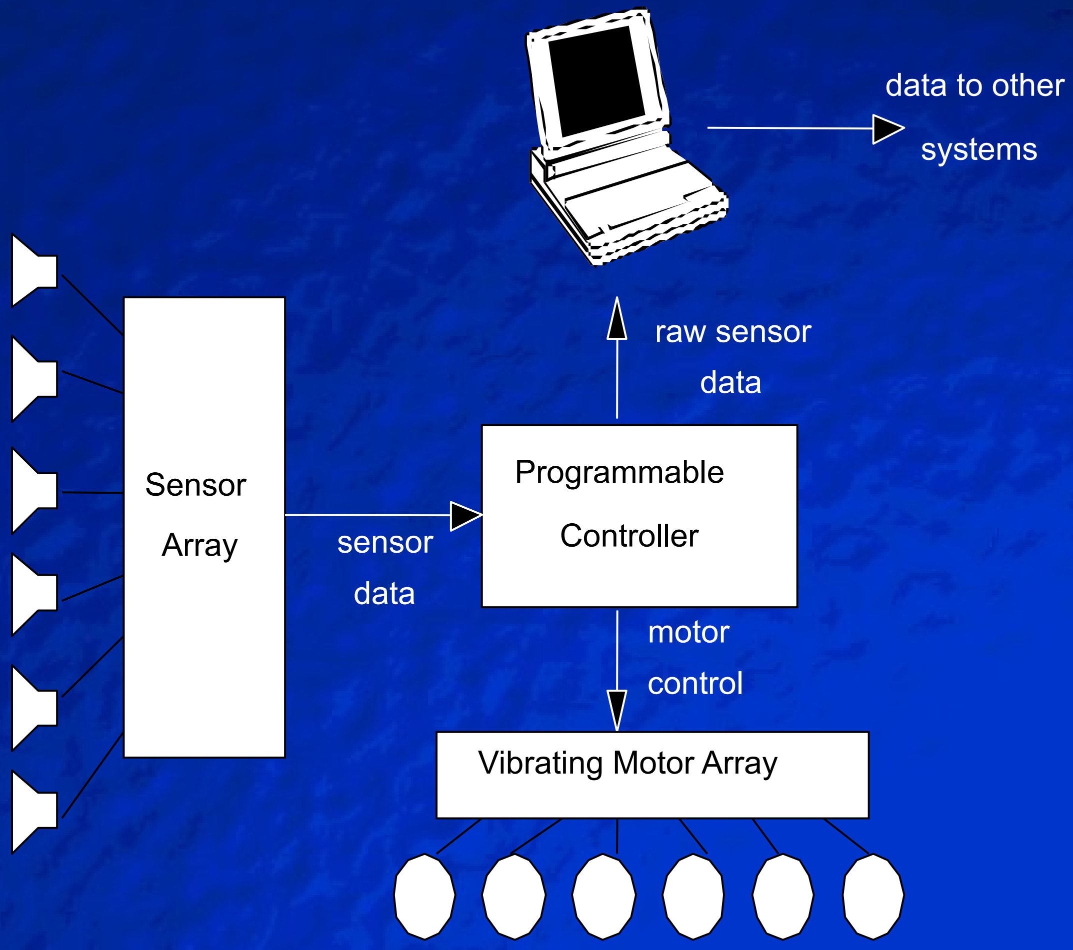
Object Detection and Ranging using Sonar Sensors

Introduction

- Need for range sensing
  - Object avoidance
  - Hazard recognition (speeding vehicles)
  - Distance determination
- Real-time feedback system
  - Haptic interface (vibrating motors)
  - Auditory feedback through voice synthesis

System Overview

- Sonar Range Sensors
- Programmable Controller
- Laptop Processing System
- Vibrating Motor Feedback



System Description - I

Sonar Range Sensors

- Polaroid 7000 Range Sensors
  - Max range of 35 feet
  - Accuracy of +/- 0.4 inch
- Arrangement
  - 6 sensors, 60º apart
  - Placed on belt worn by user
- Interface
  - Connected to programmable controller via data cable

System Description - II

Programmable Controller

- OOPic
- Functions
  - Read sensor data
  - Estimate object distance, speed
  - Determine collision probability
  - Activate appropriate motor
  - Communicate with computing unit
- Programming

System Description - III

Vibrating Motors

- DCM 154 – Used in pager type applications
- Activation depending on signal sent from programmable controller

System Description - IV

Laptop Processing System

- Software System Features
  - Linux based
  - Read sensor data via RS 232
  - Process speed and distance data
  - Determine emergency situations
  - Respond to information requests from other sub-systems
  - Send emergency signals to other sub-systems

Conclusion

- Accomplishments
  - Designed structural system
  - Partially implemented our design
  - Laid foundation for future development
- Difficulties
  - Limited time for system construction
  - Limited technical know how of embedded systems
- Future Plans
  - Completing and improving a working system
  - Testing system in real world situations

IPRO 305

Stereo Vision using Cameras

Goal:

To build an artificial vision system using high – level sensor: Camera

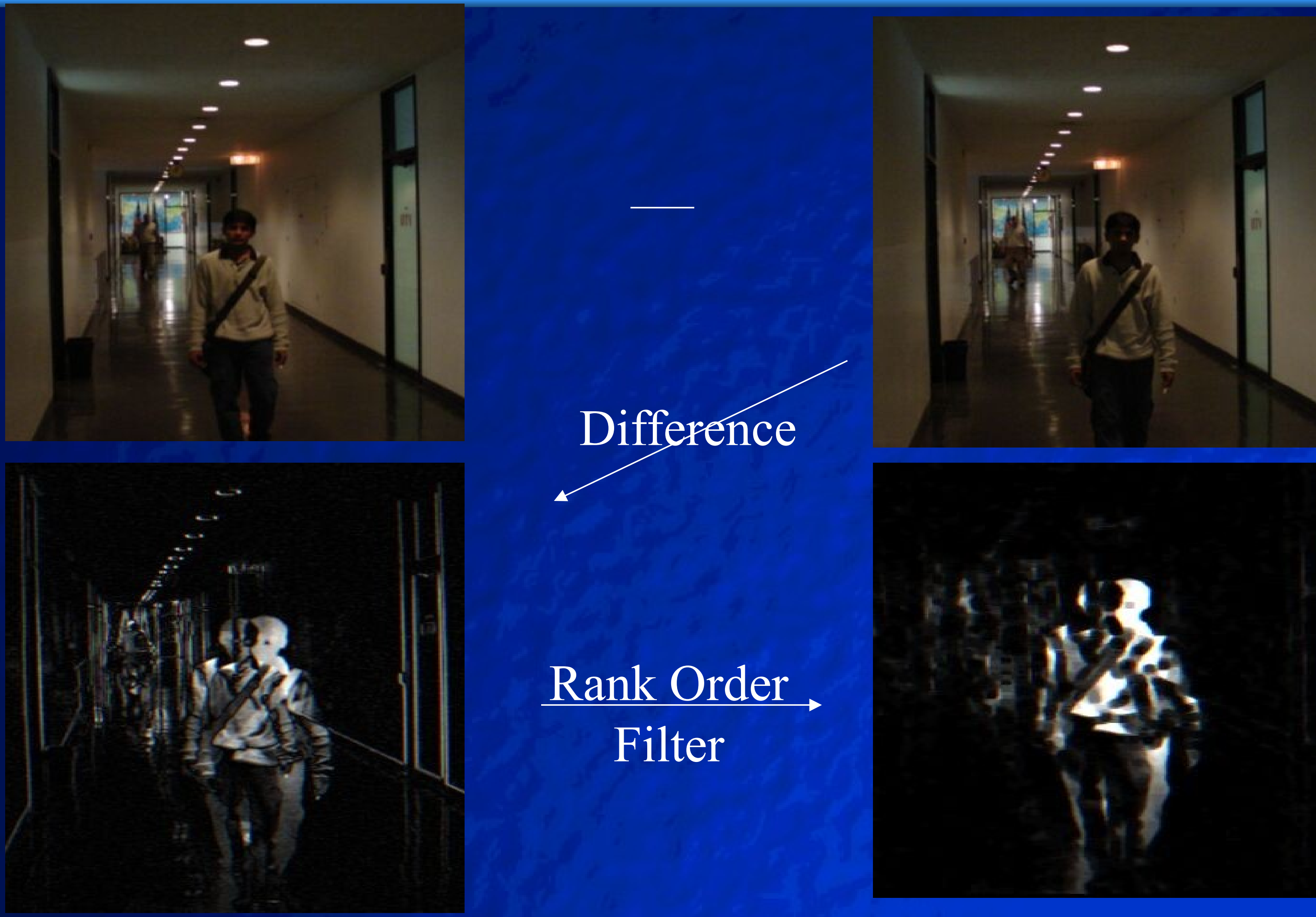
Introduction

- Two cameras placed on either shoulder of the user.
- One directed horizontally and the other placed at an angle.
- Cameras are connected to a laptop.

Objectives

- Detection of moving objects.
- Determine the size and position of the object in the image.
- Locate rectangular shapes in an image.
- Locate obstacle/step in the path of the user.
- Detect & recognize street signs.
- Report information to other sub-groups

Image Filter



Accomplishments

- Image enhancement
- Filtered out noise
- Detect region of movement
- Analysis
- Edge detection using enhanced images.
- Research
  - Laser – line method.
  - Hough Transform

Future Steps

- Determine relative size and shape of object.
- Implement Hough Transform.
- Detect man-made obstacles.

IPRO 305

User Interface

Creating a system tailored to the needs of the visually impaired

User Research

- Research Method

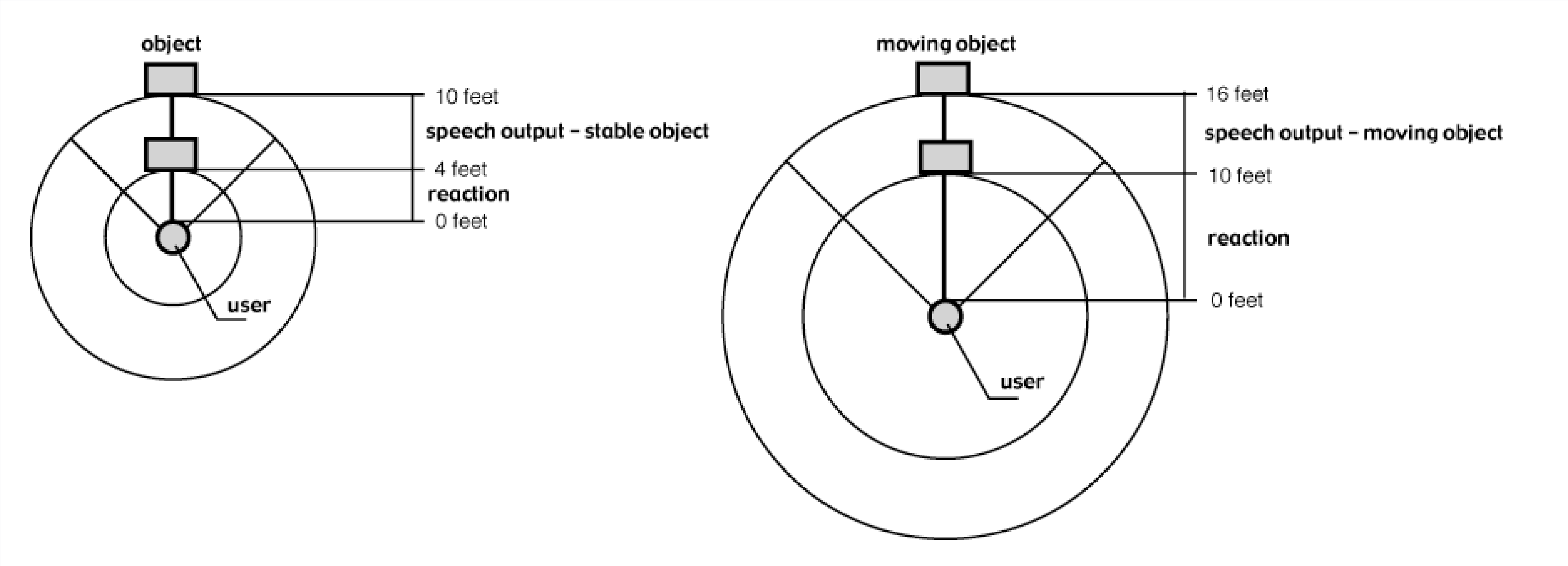
- User interview

- Observation

- Key Insights
  - Object information user needs
  - Priority of object properties to be detected
  - Existing way of object detection and suggestion for voice interface
  - Benefit of using non-speech output as well as voice output
  - Problems and issues for object detection

Analysis - 1

- Distance for Voice Feedback



Feedback for Stable Object

Feedback for Moving Object
- normal walking speed

Accomplishments

- Voice Interface Contents

- Voice output contents

OBJECT		ENVIRONMENT
MOVING OBJ	STABLE OBJ	GROUND
Object identity Direction Distance Size Color Texture	Object identity Location Distance Size Color Texture	Ground division Ground section Ground level

- Voice output : speech + non-speech

		Speech	Non - speech
Purpose	Warning		O
	Description	O	
	Guidance	O	O
	Detection	O	O
Information	Location	O	
	Object identity	O	
	Direction	O	
	Distance	O	O
	Size	O	
	Speed	O	O
	Color	O	
	Texture	O	

- Voice input : requesting information

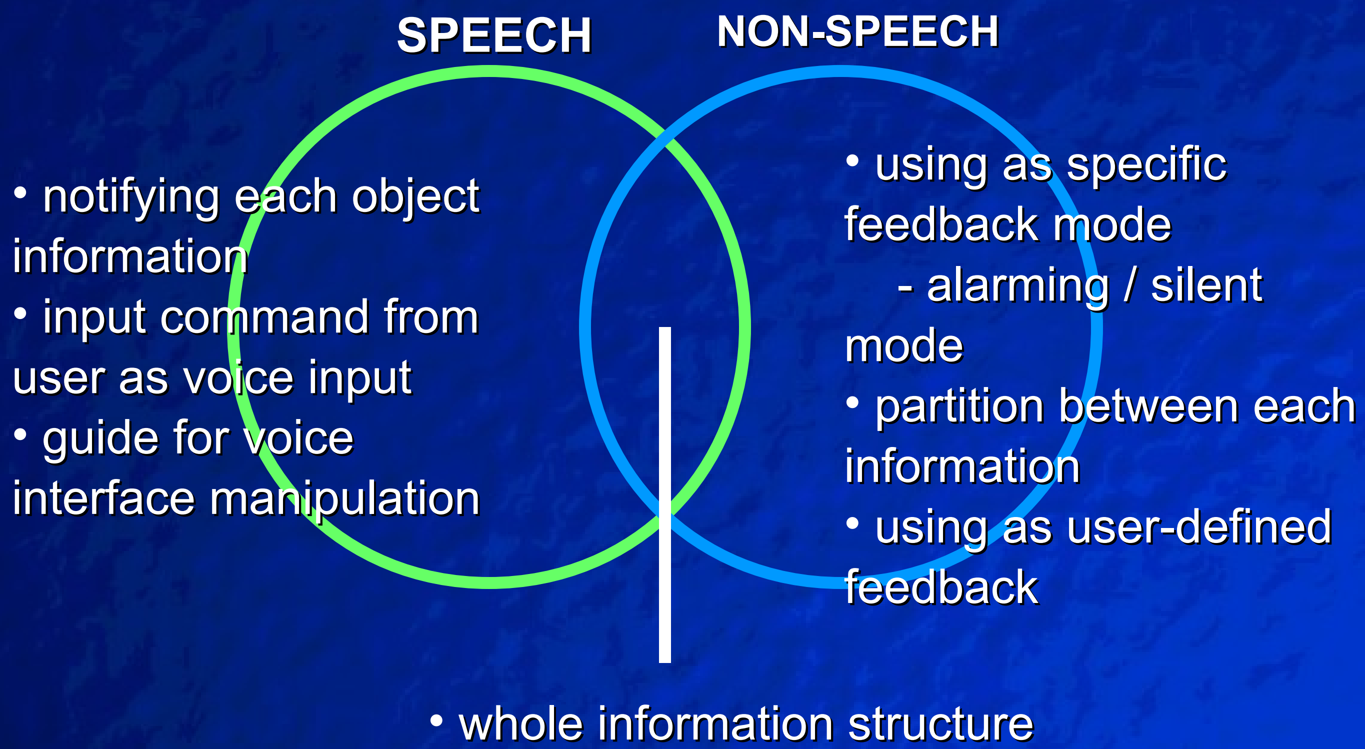
Input Command	Function
object	Repeat previously detected object identity
distance	Repeat previously detected object's distance
direction	Repeat previously detected object's direction
speed	Repeat previously detected object's speed
repeat	Repeat previously detected object's information (object, distance, direction, and speed)

- Voice input : manipulating system

Input Command	Function
detect on	Turning on voice interface
detect off	Turning off voice interface
stop	Stop / pause notifying information
resume	Resume notifying information
silent on	Turning off voice feedback, start silent mode using quiet sound for detecting object in specific situation such as conversation (information in silent mode causes limited object information – it'll just notify existence of object as well as distance from differentiating pitch between beeps)
silent off	Turning off silent mode
alarm on	Turning on warning mode
alarm off	Turning off warning mode
volume up	Turning feedback volume up
volume down	Turning feedback volume down

Analysis - 2

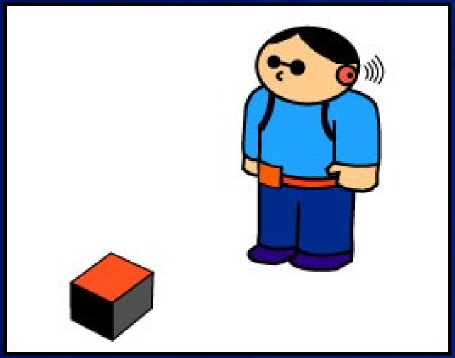
Speech & Non-speech Contents



Scenario

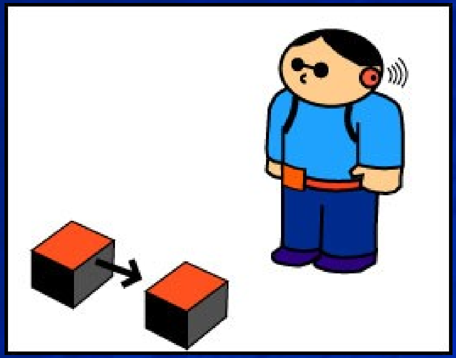
- Scenario of Use

- Stable object



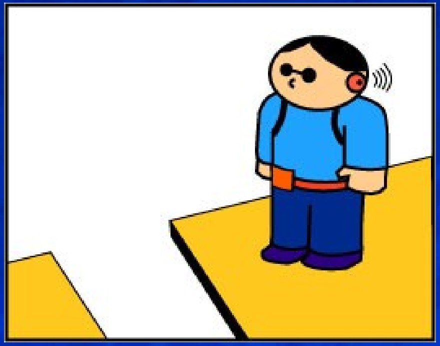
Voice output structure:
start of information / object identity / location / distance / end of information
Voice output content:
" beep / box / 11o'clock / 5feet / beep "

- Moving object



Voice output structure:
start of information / object identity / direction / distance / end of information
Voice output content:
" beep / box / 11 to left / 7feet / beep "

- Ground



Voice output structure:
start of information / 'ground' / ground status / distance / end of information
Voice output content:
" beep / ground / pavement / 5feet / beep "

Voice-user interface

Main()
VUinterface.cpp

VoiceSyn class
VoiceSyn()
void readInput(G1 cmdPort)
void readInput(G3 cmdPort) void
readInput(G2 cmdPort)
void say(apstring message)
float distance, time, speed,
directionH, directionV,size;
char text[20];
apstring newMessage
const char beep

VoiceRec class
VoiceRec()
void convertVoice(file)
void readCmd()
void sendCmd(G1 cmdPort)
void sendCmd(G2 cmdPort)
void sendCmd(G3 cmdPort)

Voice-user Interface

The following speech synthesis software were identified, and reviewed:

- Festival
- Screader 1.8
- Via-voice TTS SDK for Linux (by IBM)
- EmacSpeak
- Say

Interface Design

User → Voice Input → Text → **INTERFACE** → Text → Voice Output → User

↓ ↑

Group 1 / Group 2

In addition, the sensors (Group 1) and the camera/OCR (Group 2) will be able to communicate with each other.

Voice Recognition

- **IBM ViaVoice Speech Recognition (ASR) SDK for Linux V3**
  - The SDK provides the necessary tools to develop applications that incorporate speech recognition, both dictation and command and control, using the Linux for x86s operating system. It includes a robust set of application programming interfaces (APIs) that allow Linux applications to access speech recognition resources. This new version of the SDK includes a User Guru GUI sample along with source data and open source for the top level GUI from the **IBM ViaVoice Dictation for Linux Product.**

Conclusion

- Project accomplishments
  - Each sub group
- Team experience
  - Real world problems
  - Communication and co-ordination
  - Cross-disciplinary experience
- Acknowledgements