GMU MiniCAVE

Edward J. Wegman

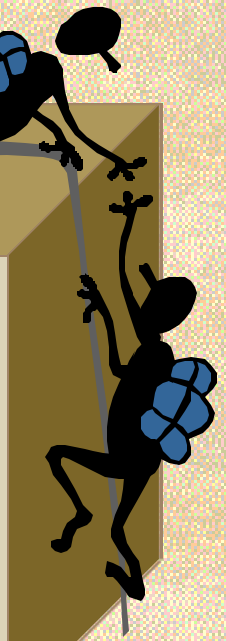
Juergen Symanzik **Patrick Vandersluis**

Xiaodong Fu **Ying Zhu**

Rida Moustafa **Qiang Luo**

Nkem-Amin Khumbah **Fernando Camelli**

Antoinette Dzubay **Robert Wall**



Workshop on Data Visualization in Statistics

July 8, 1998

Outline of Talk

- ✓ The CAVE™ VR Environment
- ✓ Motivation for the MiniCAVE™
- ✓ The MiniCAVE™ Environment
  - Windows NT/Pentium II Implementation
  - Voice Interface
- ✓ Technical Challenges

Trademark Notes

- CAVE is a trademark of the Electronic Visualization Laboratory at the University of Illinois, Chicago
- MiniCAVE is a trademark of the Center for Computational Statistics at George Mason University

CAVE Concept

- ✓ A Projection-based Immersive VR System
  - Silicon Graphics-based with 8 to 12 processors
  - RE^2 or RE Infinity graphics engines
  - CRT-based projection system
  - Stereographics Crystal Eyes shutter glasses
  - Head tracking
  - Usually 3 to 5 wall cube
- ✓ Developed originally at U. Illinois, Chicago
  - Carolina Cruz-Neira

CAVE Strengths

- ✓ Effective immersive environment
  - Lightweight non-intrusive glasses
  - Can see own hands and other participants
- ✓ Effective for group VR
  - Good tool for group collaboration

CAVE Weaknesses

- ✓ CRT Projectors
  - Projectors not very bright
  - Shock, vibration & heat, hard to keep focus
  - Geometric distortion at wall interfaces
- ✓ Tracking
  - One user tracked, badly distorted stereo for users not at viewpoint
- ✓ User Interface
  - Usually 3-D extension of desktop metaphor

CAVE Weaknesses

- ✓ Expensive
  - $1,000,000 fully outfitted
  - $600,000+ SGI computers
  - $30,000 per projector

Motivation

- ✓ Installed MATLAB 5 on SGI Onyx and Pentium
  - Benchmarks on 200 megahertz Pentium Pro ($3000) and 200 megahertz SGI Onyx ($120,000) similar
- ✓ Liquid Crystal Projectors sharp, bright, and stable under shock, vibration and temperature variation
- ✓ Stereographics Crystal Eyes technology available for Windows NT

MiniCAVE Concept

- ✓ Windows NT/Intel Pentium II 400 mhz
- ✓ LCD/DLV-based projection systems
- ✓ 12 ft cubes scaled to 6 ft cubes
- ✓ Tracking optional, reduced latency
- ✓ Voice command metaphor
- ✓ $100,000 entry level

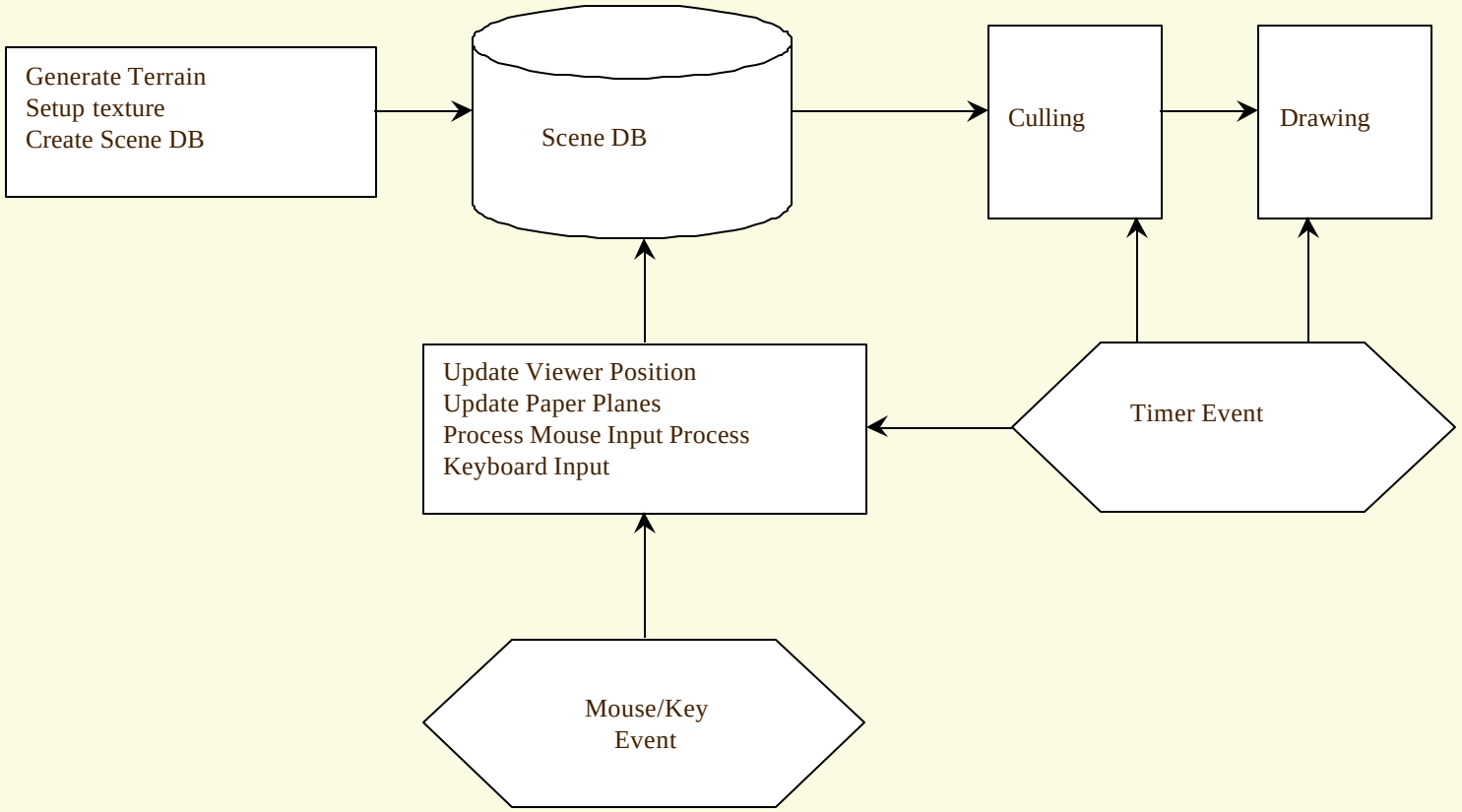
VR from Workstation to PC

- ✓ Project Purpose
  - Can the NT workstation really match the SGI workstation in 3D graphics area?
  - Possibility of VR implementation in PC environment, especially MiniCAVE.
  - Explore the hardware and software capacities of PC for VR application.

✓ Initial Approach

1. Porting one SGI GL application, SkyFly, to OpenGL, which is a platform-independent 3D API.
2. Porting this application to NT environment.
3. Performance comparison between SGI and NT.
4. Stereo display on PC using CrystalEyes.
5. Controlled by voice command.

Structure of Skyfly Flight Simulation Program

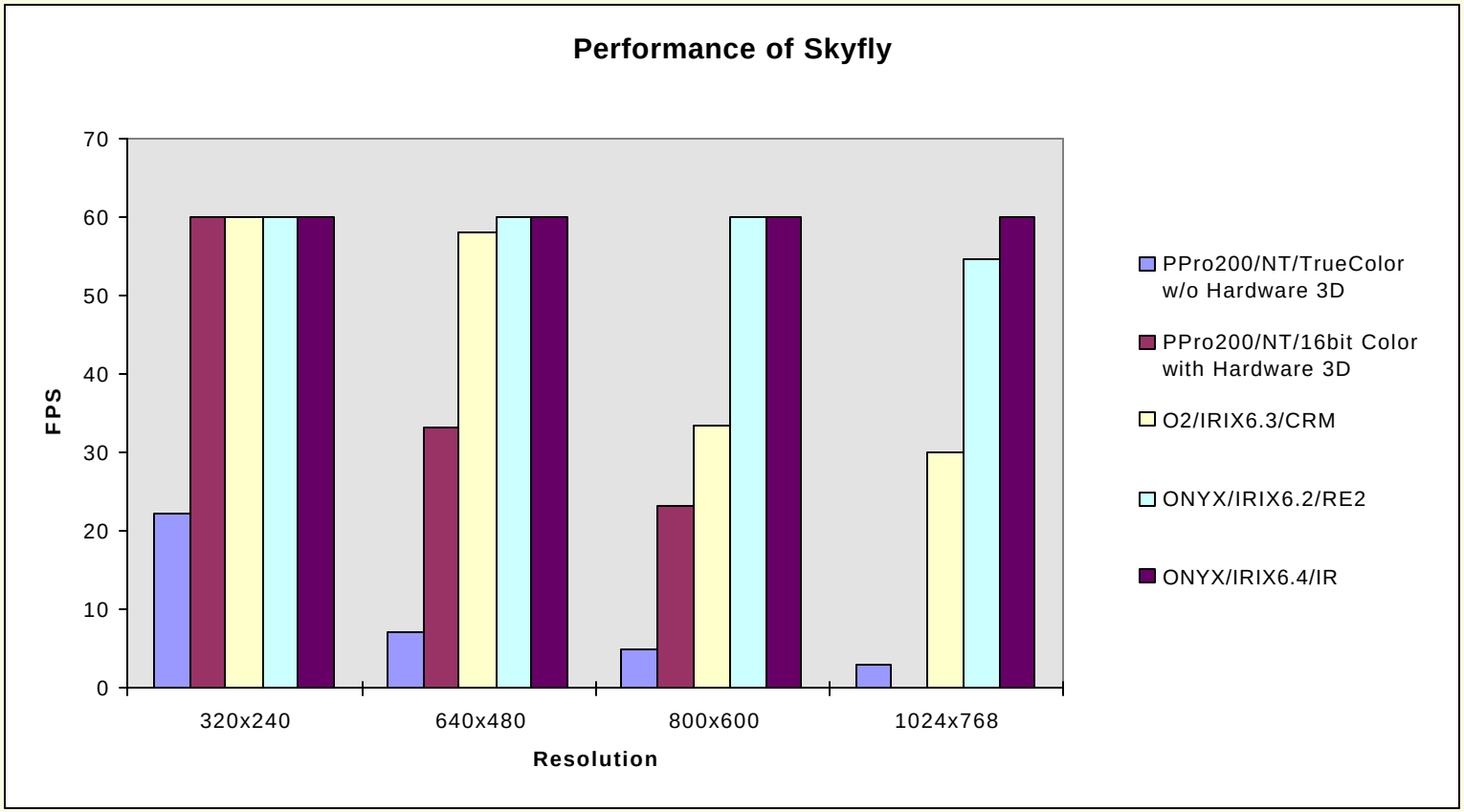


✓Porting Skyfly to NT

- Using OpenGL interface (GLUT) instead of SGI's proprietary mouse/key GL interface.
- OpenGL source code compiled on NT and SGI without modification.

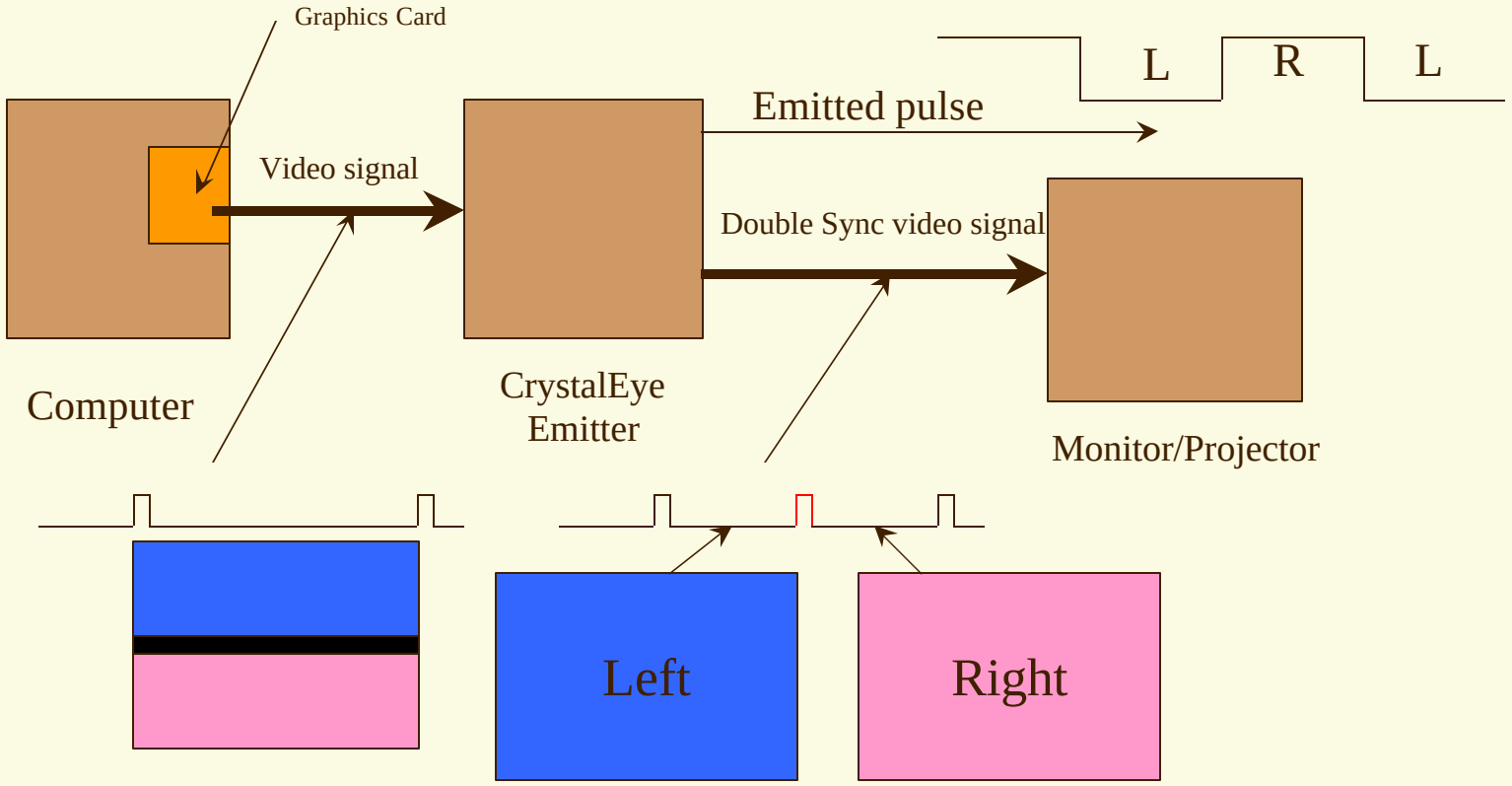
GL/OpenGL features	GL Version	OpenGL Version	Comments
Triangle Strips	✓	✓	Terrain
Texture Mapping	✓	✓	Clouds & terrain
Lighting	✓	✗*	Terrain and paper planes

✓Performance Comparison



- ✓ Stereo using CrystalEyes
  - Above-below stereo.
  - Image resolution 1024x384 each eye.
  - Vertical refresh rate 120-150 (60-75 each eye).
    - SGI monitor can handle both 120 and 150.
    - CRT projector can only handle 120 refresh rate.
  - PII 333 + Permedia 2 can achieve 30 fps

✓ Principles of above-below CrystalEyes stereo



Speech Motivation

- ✓ User Interfaces (Van Dam)
- ✓ Shortcuts in Xgobi
- ✓ User Controls in the C2 Stats Application

Speech Recognition Products

Dragon Dictate

IBM ViaVoice

Speech Recognition Technology

Evaluation

1. Can you train the software to understand additional words?
2. Is the software speaker dependent or speaker independent?
3. Can you store different pronunciations of one word into the same database so a spoken word is compared with different pronunciations?
4. How good is performance (i.e., percentage of correctly identified words before and after training)?
5. Does accuracy depend on speaker / accent / training?
6. What improves accuracy relative to the various sources of imprecision?
7. How long does it take to reach various levels of performance?
8. What if we aren't able to train? What should we expect?

Measures of Performance

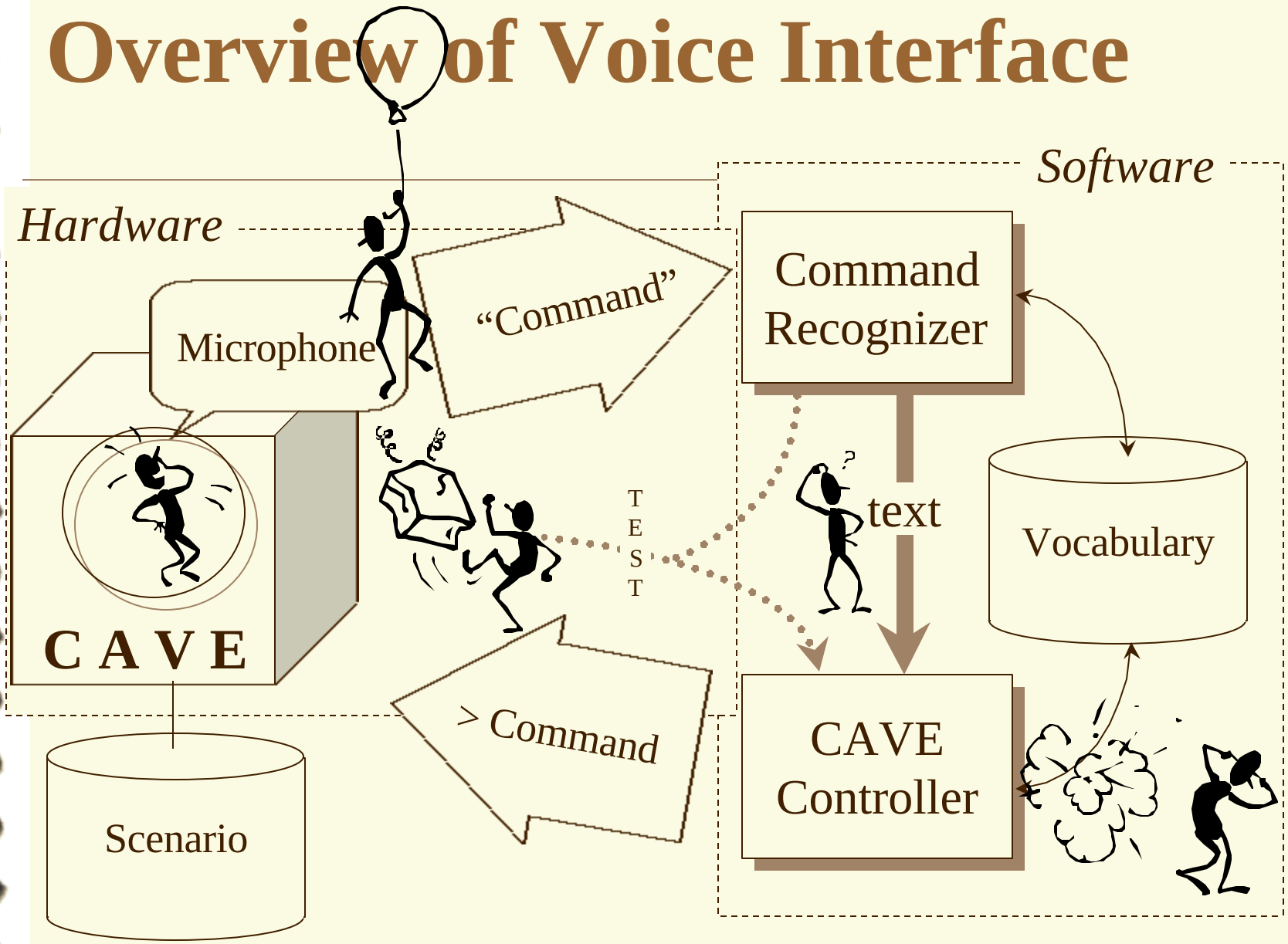
Summary

Attribute	Metric
✓ **Setup Time** – Installation Time – Basic Training Time	✓ Duration
✓ **Accuracy** (WYSIWYG) – Untrained and Trained – Under Various Speakers	✓ Error rate (%)
✓ **Speaker Independence** – out-of-box performance	✓ No-Training Accuracy
✓ **Robustness** – Scope of Vocabulary	✓ Number of *New* Words Needed
✓ **Response Latency** – Command & Control	✓ Command-to-Text Cycle Time

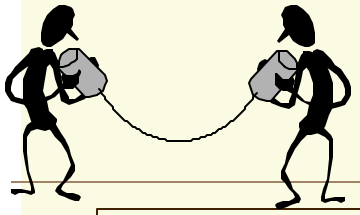
- What's the good format for voice-controlled navigation in this application? What are the recommendations on the maximum number of words for the application? What do they suggest as maximum number of words in one command?
  We currently only have a simple command format: single word, with a around 20 words vocabulary. The generalization appears achievable.

- ✓ Voice Control
  - command set
    left, *right*, *up*, *down*, *fast*, *slow*, *forward*, *reverse*, *stop*, *start*.
  - Link directly to Dragon Dictate (locally), or link to custom-DLL(network possible).

Overview of Voice Interface



Command Recognizer



Requirement:Capture Voice Command & Output Text

Command Recognizer → text

Speech Recognition Software

Visual Basic Command Front End

Custom Dynamic Linkable Library (.dll)

Command Vocabulary

API

Speech Recognition Details

- ✓ **First Cut Assumption:** Speech recognition and Skyfly software reside on same machine
- ✓ Visual Basic (VB) Command Front End (FE)
  - Trap & redirect spoken word using
    - DragonDictate DDSpeech Control
    - VB Controls (e.g., text boxes)
    - C-Interface defined in a custom DLL
  - Flow
    - MiniCAVE-immersed user issues a command
    - DDSpeech control returns word and a recognition indicator to FE
      - If (Word Found) Then Call interface to pass word
      - Else VB displays message

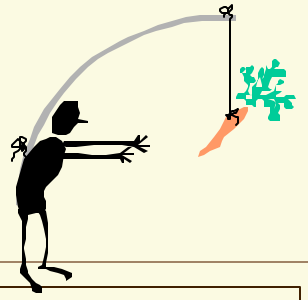
Speech Recognition Details (continued)

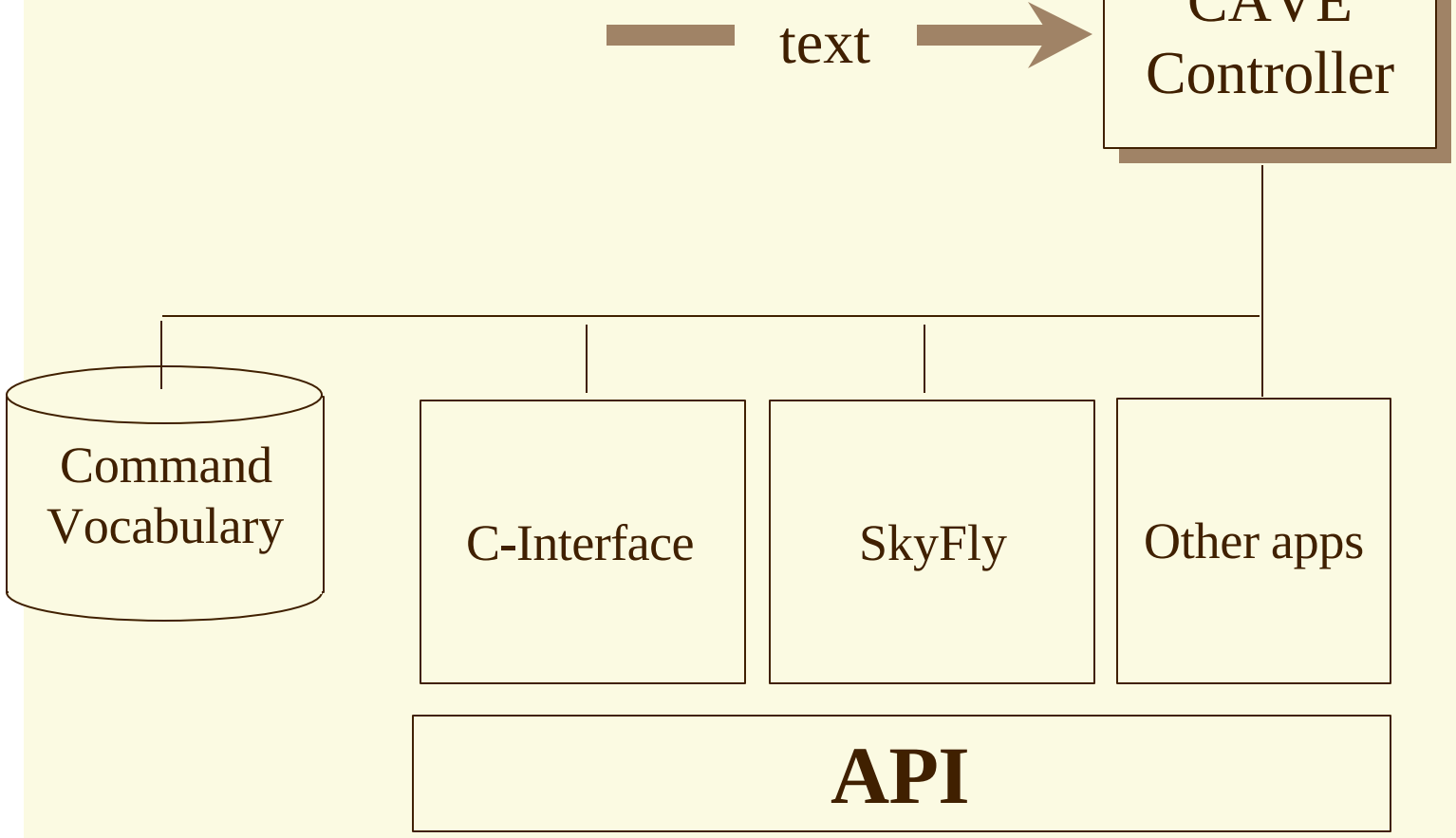
- ✓ Connecting VB to C Interface
  - Create DLL containing C function
  - Link C function into VB FE using “Declare” statement
- ✓ Custom Dynamic Link Library Components
  - Librarian
  - Windows Exit Procedure
  - .def File
  - Make File
  - C Interface (variable length string, integer length, integer return code)
    - return code: 0 = No Error; 1 = Error (“command not recognized”)

. . . Use Visual C++ to create the DLL

CAVE Controller

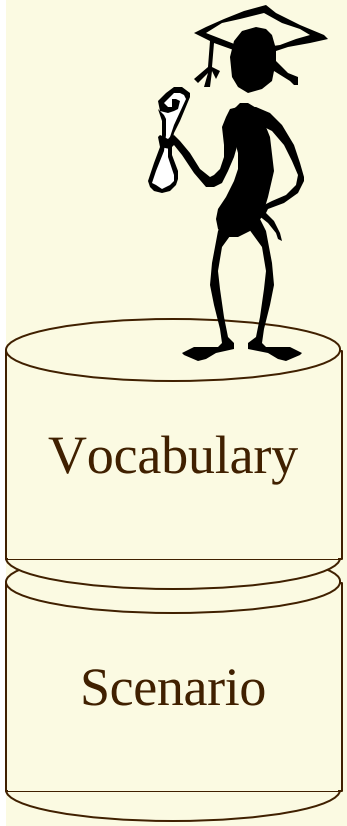
Requirement:Capture & Execute Text Command





Domain Components

Requirement:"Specialize" the Application



- ✓ **Command & Control Vocabulary**
- ✓ Dictation Vocabulary

- ✓ **3-dimensional Graphical Renderings**
- ✓ Visual Images, Maps, etc.
- ✓ Textual Data
- ✓ Application Functions, Formula, etc.
- ✓ Numerical Data

Approach

- ✓ Integrate successfully on one machine then attempt a multi-machine solution
- ✓ Recognition of spoken word causes delay in SkyFly program

Speech Recognition Project

N-node Implementation Alternatives: **Option 1**: Single Machine

- ✓ SkyFly program:
  - Receives the speech recognition event
  - Interprets the word and produces the motion
- ✓ No communication method such as RPC is required
- ✓ Easy to implement, but...
  - Recognition of the spoken word causes a delay in the SkyFly program
  - Motion is “jerky” as you move from left-to-right or vice versa
  - Running both applications on the same NT machine may not be a viable alternative

Speech Recognition Project

N-node Implementation Alternatives: **Option 2**: Single- or Multi-Machine

- ✓ Uses the message passing interface PVM to communicate between the speech recognition application and the SkyFly program
  - Non-synchronized sending and receiving or messages
  - MASTER-SLAVE architecture
- ✓ Visual C++ or C++ program:
  - Receives the speech recognition event & bundles information
  - Acts as the MASTER process -- initializes PVM and spawns a SLAVE process
- ✓ Fly-over program:
  - Acts as the SLAVE process
  - Receives and unbundles the information

Speech Recognition Project
N-node Implementation Alternatives: **Option 3**: Single- or Multi-Machine

- ✓ Visual Basic (VB) Application
  - Receives the speech recognition event
  - Passes the recognized word via a custom built DLL
- ✓ Fly-over program uses the same DLL to receive the word
- ✓ Can be extended to a multi-machine solution using RPC or PVM
- ✓ Easy to integrate speech controls into the VB application
- ✓ Didn't implement the back-end call to the DLL
- ✓ We're not sure it works; back-end call to DLL not implemented

Technical Challenges - Successes

- ✓ Port of SkyFly Stereoscopic Demo to NT successful with adequate frame rates on 333 megahertz machine.
- ✓ CrystalEyes interface on NT successful
- ✓ Voice recognition using Dragon Dictate successful
  - – but requires training of speech recognizer.

Future Directions

- ✓ Speech Enhanced ExplorN
- ✓ MRI Controlled through Speech
- ✓ Stat/GIS application such as VirGIS
- ✓ Military Applications to Command and Control

Remaining Challenges

- ✓ Easy Ones
  - "Stereo Lock" Synchronization
  - "Image Lock" Synchronization
- ✓ Hard Ones
  - Edge Blending with Digital Projectors
  - Digital Projectors Themselves
    - Frame Rates
    - Decay

Current Status

- ✓ Patent Disclosure Filed
- ✓ Demonstrated at 1998 World Congress on Information Technology
- ✓ CRADA signed with U.S. Army (White Sands Missile Range) - awaiting funding
- ✓ Planned EDA/Data Mining Application with Voice Interface.
- ✓ Machine Stolen…Gads!

Contact

- Edward Wegman
  - ewegman@gmu.edu
- Juergen Symanzik
  - symanzik@galaxy.gmu.edu
- Website
  - www.galaxy.gmu.edu/papers/MiniCAVE.ppt