
Visualization tools & demos and the ICSI Realization group

Dan Ellis

International Computer Science Institute, Berkeley CA
<dpwe@icsi.berkeley.edu>

Outline

- 1 Visualization: Goals and philosophy
- 2 Tour of existing tools
- 3 Using `recogviz`
- 4 Future developments & demos



1 Visualization: Goals and philosophy

Why work on visualization tools?

- **Visual representations give insights**
 - able to scan large amounts of data
 - a perspective different from other analyses
- **Interactive exploration gives insights**
 - 'chasing down' anomalous phenomena
- **Graphical, interactive demonstrations are accessible & appealing**
 - for giving demos to visitors
 - for communicating to colleagues

BUT

- **Visualization is often non-critical**
 - tools must be prefabricated, easy to use



Goals of visualization tool development

- **“A demo on every desk”**
 - facilitate understanding/insight & diagnose problems/anomalies
 - support presentations & provide figures
 - promote ‘demo culture’
 - minimize barrier?
- **Tomorrows demos...**
 - ...can’t be predicted, so
 - establish framework
 - provide modules
- **UI coding is tedious & short-lived**
 - prefer high-level, portable solution
- **Flexibility to twist to new ends?**



Choosing an infrastructure

xwaves, lyre

- **Special purpose vs. scripting language**
 - need expandability
 - want to be able to modify
- **Choice of scripting language:**

	sh	perl	Tcl/Tk	java	matlab
graphics		?	√	√	√
command shell	√		√		√
run programs	√	√	√		
sig.proc. support					√
user extensions		√	√	√	√
'nice' syntax		√	√?	√?	
local expertise	√	√	?	?	√

→ **Tcl/Tk + extensions framework**



2

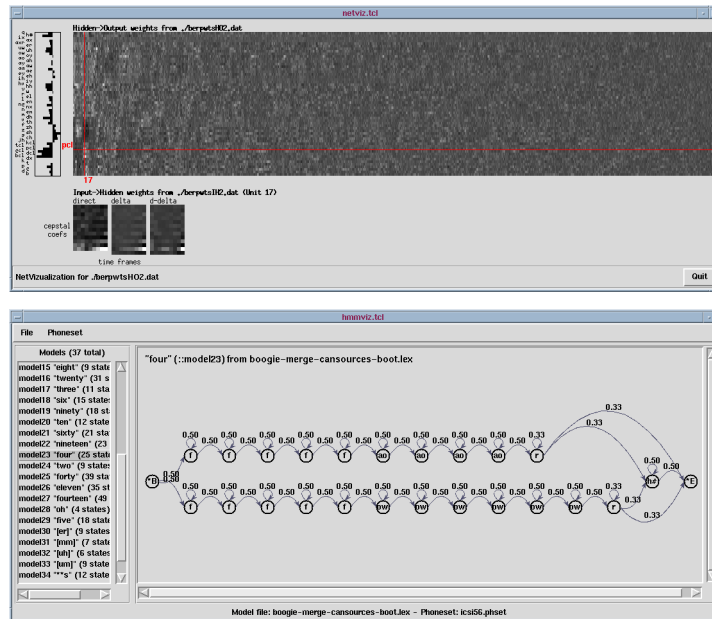
Tour of existing tools

- **Early experiments**
 - **netviz** - neural-net weights display
 - **hmmviz** - HMM model display
- **Old but useful**
 - **sgramImg** - on-demand web spectrograms
 - **pfview** - pfile inspection
 - **simpleui** - basic interactive recognition
- **The current generation**
 - **berpdemo98** - interactive speech application
 - **recogviz** - recognizer display & comparison



netviz & hmmviz

- Early experiments

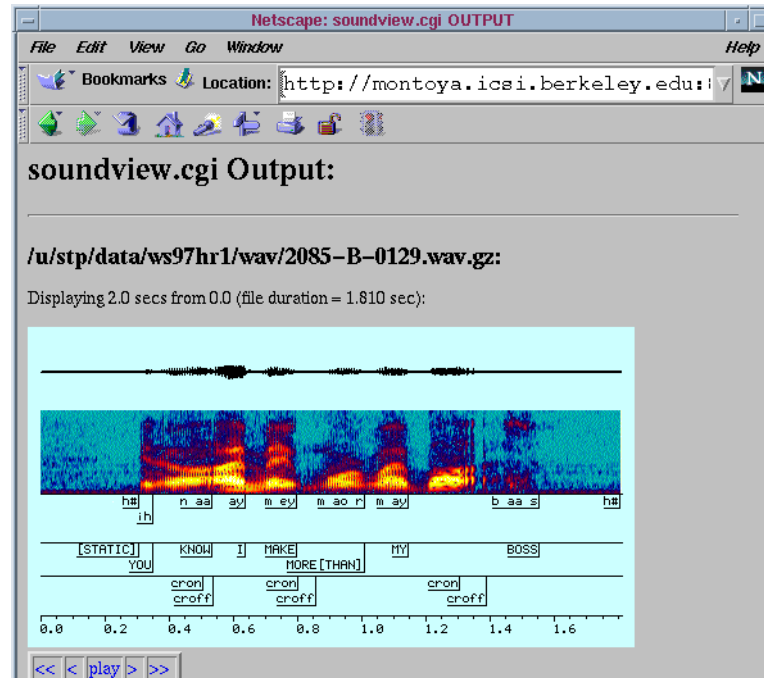


- Advantages of Tcl
 - netviz is 300 lines (+ floatArray extension)
 - clean postscript generated by hmmviz



sgramImg.cgi

- **CGI script: spectrogram GIF on-demand**



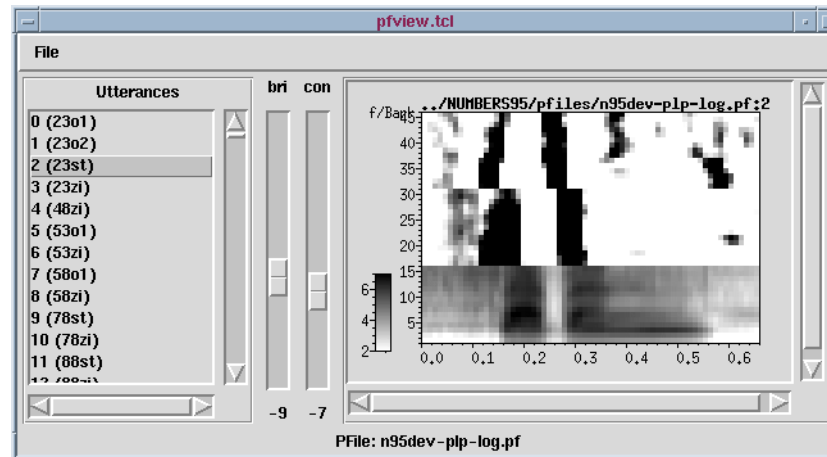
- **Automatically includes xlabel annotation**
- **Try it:**

<http://montoya.icsi.berkeley.edu:8080/~dpwe/cgi-bin/soundview.cgi>



pfview

- Display contents of pfiles as images

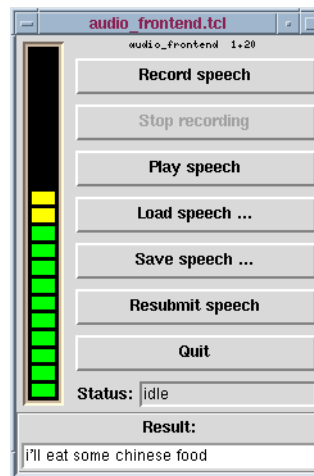


- Provision for multiple panels, labels, PS output
- Try it:
> ~dpwe/projects/pfview/pfview.tcl pfile=<pfile>
[idlist=<utid file>]

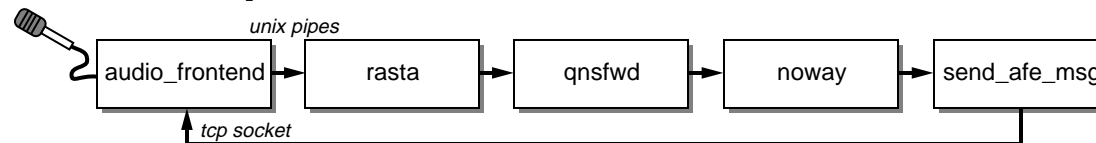


simpleui

- Minimal interactive demo of recognizer
- Just connects `audio_frontend` to a recognizer
 - easy to modify for other grammars, features



- Simple structure:



- Try it:

> /u/drspeech/share/bin/simpleui



berpdemo98

- Our 'old' application plus recogviz modules

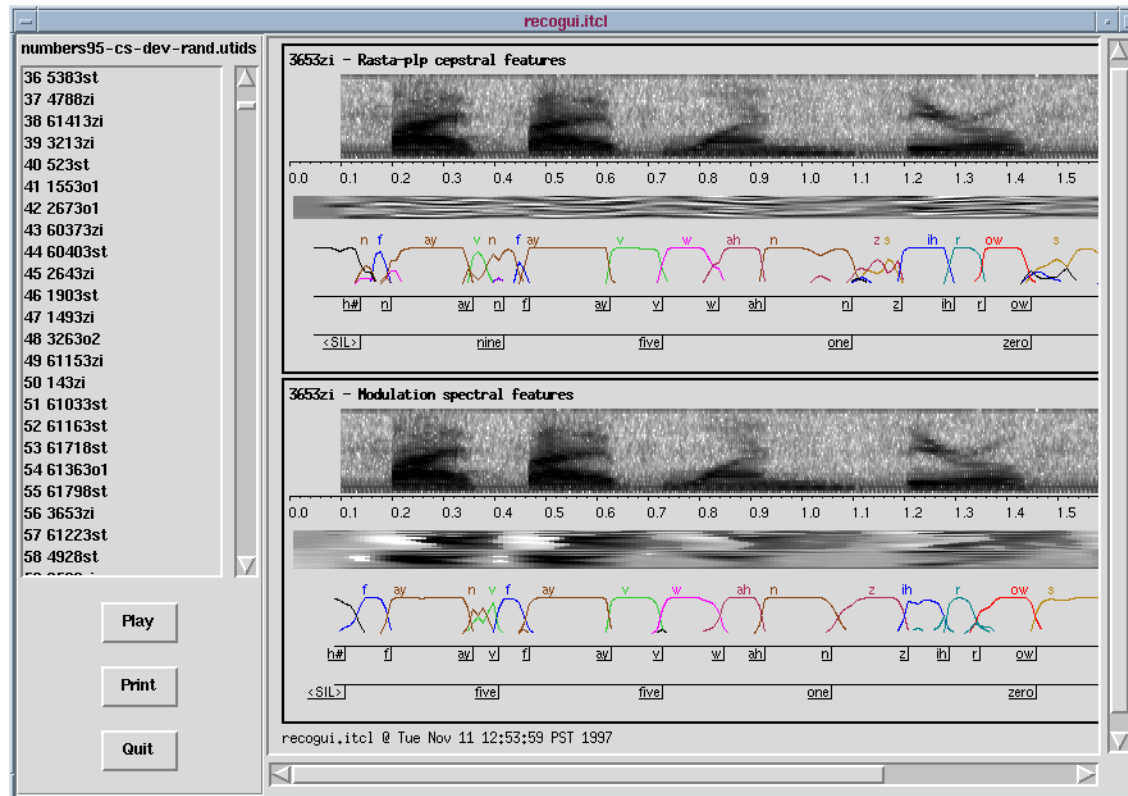
The screenshot displays the 'berpdemo98' application window. On the left is a control panel with buttons for 'Record speech', 'Stop recording', 'Play speech', 'Load speech ...', 'Save speech ...', 'Resubmit speech', and 'Quit'. A status indicator shows 'Status: idle'. The main area contains a spectrogram at the top, a waveform below it, and a phonetic analysis plot. The plot shows various phonetic segments labeled with phonemes like 'h#', 'pɔl', 'aoo', 'w l', 'liy', 'ey', 'tcl', 's', 'f', 'aun', 'chnt', 'age', 'n', 'iy', 'z', 'f', 'p', 'uw', 'd'. Below the plot, a list of words is shown: '<SIL>', 'i', 'wanna', 'eat', 'some', 'chinese'. At the bottom, the 'Recognized Words' section displays 'i wanna eat some chinese food'. The 'BeRP Response' section contains the text: 'Please provide more information in order to narrow your choices. For example: WHAT DAY WOULD YOU LIKE TO EAT?'

- **Try it:**
> /u/drspeech/share/bin/berpdemo98



recogviz

- **Motivation: compare recognition techniques**
 - at each stage in process (signal, features, probs)



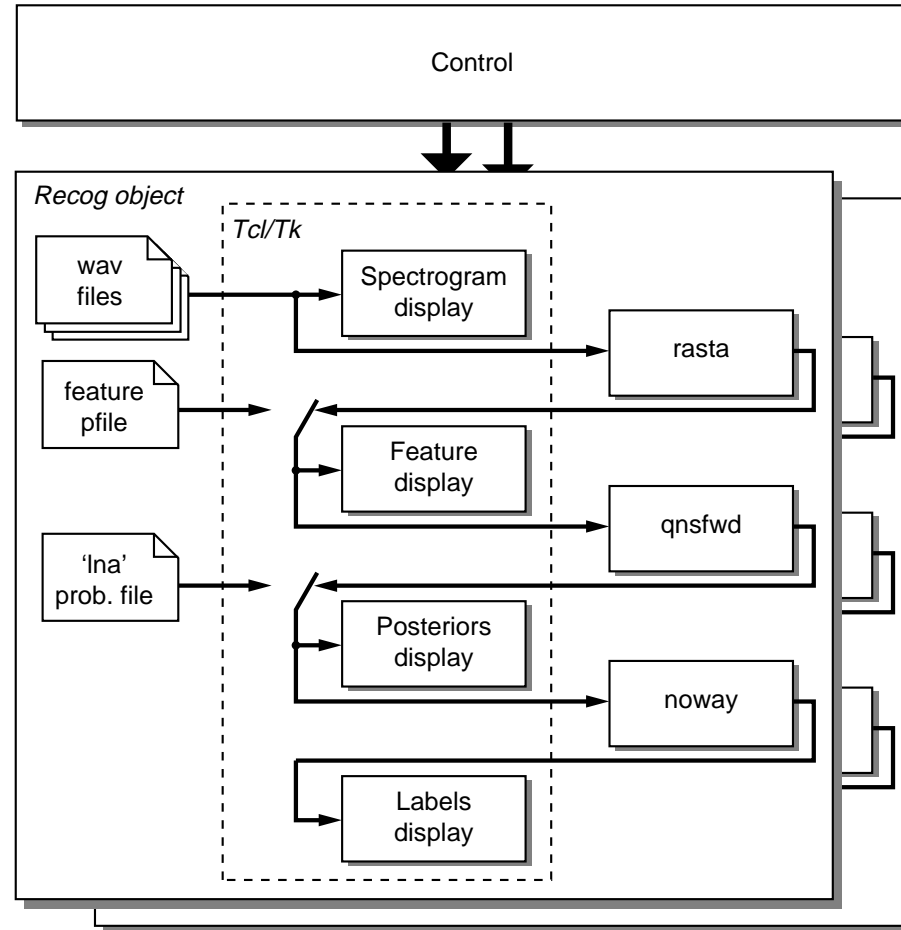
- **Try it:**
 - > /u/dpwe/projects/recogviz/recogviz



3

Using recogviz

- System structure



Using recogviz (cont.)

- **Configuration files: (Tcl syntax)**
 - defaults=./defaults.def - common definitions

```
set NUMBERSDIR    "/u/drspeech/data/NUMBERS95"  
set listfile      "$NUMBERSDIR/list/numbers95-cs-dev.utids"  
set wavfilecmd    "numbers95_findfile prefix=$NUMBERSDIR/ utid=%u"  
set samplerate    8000  
set frametime     0.010  
# Standard network geometry/type  
set mlp3_hidden_size 500  
set mlp3_output_size 56  
# Other common values  
set ftr_start     0  
...
```

- recog1=./recog1.def

```
set title         "Rasta-plp cepstra"  
set DPWE          "/u/dpwe/projects/NUMBERS95"  
set ftr_file      "$DPWE/pfiles/n95dev-lras-plp-cep.pf"  
set ftr_width     27  
set norm_file     "$DPWE/pfiles/n95tr-lras-plp-cep.norm"  
set weights       "$DPWE/results/1997sep17/n95tr-embed+iter2.weights"  
...
```

+ recog2=./recog2.def

- **Use dr_recog params files? (y0/noway)**



4

Future developments & demos

- **“A demo on every desk”**
 - but demos are more than graphics: message
- **Illustrations for research advances**
 - feature domains
 - recognizer combinations
- **Application-scale demos**
 - grandson of BeRP
 - ?



Creating new demos

- **Available pieces:**

hackable top-level scripts

simpleui recogviz berpdemo

embeddable Tcl programs

**audio_frontend.tcl
probs.tcl**

[incr Tcl] classes

**CanvFA.itcl Recog.itcl
... ...**

Tcl source files

**CanvLabels.tcl
cgiutils.tcl
...**

Tcl/Tk loadable extensions

**libaprl libfarray_otcl
 libsound_otcl
... libpfif_otcl**

Binary programs

**itkwish rasta
 qnsfwd
 noway**

- **Get bits via Tcl package system**

see also: <http://www.tcltk.com>, ...



Future work

- **Improving the existing bits**
 - cleaning up modules
 - better example shells
 - documentation
- **Generating new demos**
 - .. within existing tools (recogviz)
 - .. employing existing pieces
- **Collaborative applications**
 - ThisL
 - Daedelus
- **Wider questions:**
 - distributable demos?
 - cross-platform demos?
 - applications vs. basic research?

