EECE593 Winter 2002 Project Manual
Vocal Tract Visualisation

David Pritchard

April 21, 2002

1 Overview

For my project, I have created a 3D reconstruction of the human vocal tract. The intention of this project was to practice the volume image processing techniques learned in class, while working on a real problem.

Data for the project was obtained from a set of Magnetic Resonance Imaging scans published by Olov Engwall [EB99]. Like many linguistics researchers, he used this data to calculate the area of the vocal tract along its length. This area data can then be used to understand how the vocal tract forms different sounds. Acquiring these MRI scans is a slow process, typically requiring 40 seconds or more. Consequently, vowels are typically imaged instead of consonants. Engwall does describe the capture of consonants and fricatives, but I am uncertain of how this was achieved.

Engwall's data was separated into three sets: 18 slices in the transverse plane (through the pharynx), 18 tilted slices (through the velum), and 14 slices in the coronal plane (through the mouth and face). This data was captured in a single scan.

For my reconstruction, I registered these data sets in order to determine their relative positions. I then resampled the data and merged it into a single 3D volume. Finally, I had to segment the data in order to separate the vocal tract from the rest of the head.

2 Requirements

To view my results, you will need:

- Matlab, release 12.1 (version 6.1.0.450).

- The Matlab Image Processing Toolbox, version 3.1.

- The Matlab Optimization Toolbox, version 2.1.1.

- The SDC Morphology Toolbox for Matlab, version 1.1.

The version numbers listed above are those I used; later versions will likely also work, provided that drastic changes don't take place. The two required Matlab toolboxes are shipped with the complete release of Matlab; you can see if you have them installed by typing `help images` and `help optim` in Matlab.

3 Setup

The only setup required to begin using the project relates to installation of the SDC Morphology Toolbox. I used a 30-day demonstration licence of this toolbox, which can be downloaded for free from `http://www.mmorph.com`. Please follow their directions for setup.

After installation of the SDC Morphology Toolbox, there are still a few steps required to get everything to run. extra steps to get everything working under Linux:

1. Add the `mmorph1.1/lib` directory to `LD_LIBRARY_PATH` before starting Matlab.

2. Run the `mmorph1.1/lib/mmstartup.m` script after starting Matlab.

3. Run the setup script, `main.m`

A transcript of my own startup sequence is shown below.

```
[8] seinfeld ~/course/eece593/project/src % setenv LD_LIBRARY_PATH
${LD_LIBRARY_PATH}:`pwd`/morph1.1/lib
[9] seinfeld ~/course/eece593/project/src % matlab


                        < M A T L A B >
                Copyright 1984-2001 The MathWorks, Inc.
                    Version 6.1.0.450 Release 12.1
                            May 18 2001



  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, visit www.mathworks.com.


>> run('morph1.1/lib/mmstartup.m');
Expiring in 9 days.
SDC Morphology Toolbox V1.1 15Jan02 successfully installed
Type one of these commands:
help mminfo : to see the list of basic concepts
help morph  : to see the list of functions
help mmdemos: to see the list of demonstrations
web(which('index.html')): to access the HTML documentation
>> main
Loading data
Resampling in z
Transforming data into registered positions
Segmenting data
Transforming segmented data into registered positions
>>
```

Be aware that the final stage is quite slow, and requires a lot of memory. On a Pentium IV 1.8 GHz with 500 MB of memory, calculation required about ten minutes in total. After completion of calculation, I usually save the workspace to disk, to avoid any need of recomputation. An abbreviated version of this output is included in the examples directory.

4   Results

After setup, the Matlab workspace will contain a number of variables. The most important ones are the following:

- `ph`: the pharynx dataset, after histogram equalisation and interpolation in z.

- `ti`: the tilted dataset, after interpolation in z.

- `co`: the coronal dataset, after interpolation in z.

- `phtico`: the three datasets merged after registration. The merge process is simplistic, using just a maximum operator in regions where two datasets overlap. Useful for verifying registration process.

- `ph_seg`: segmented version of `ph`. A binary volume indicating the vocal tract regions.

- `ti_seg`: segmented version of `ti`. A binary volume indicating the vocal tract regions.

- `co_seg`: segmented version of `co`. A binary volume indicating the vocal tract regions.

- `phtico_seg`: the merged segmented datasets, using the same registration transformation as `phtico`. This is the final output.

There are also a few more minor variables:

- `ti_seg_reg`: registered version of `ti_seg`. Uses the same rotation/translation as was used in `phtico`

- `co_seg_reg`: registered version of `co_seg`. Uses the same rotation/translation as was used in `phtico`

- `phtico_fg_mask`: a rough segmentation of phtico dataset, separating the head from the background. Useful for visualisation of the head's shape.

- `ptE`: 4×4 transformation matrix between the tilted co-ordinate frame and the pharynx co-ordinate frame. If `ptE` is multiplied by a point in the tilted space, the result is the same point's position in the pharynx space.

- `ptI`: brightness adjustment co-efficient for registration. Given a voxel in the tilted dataset, the intensity must be multiplied by `ptI` to obtain the correct intensity in the pharynx dataset.

- `tcE`: 4×4 transformation matrix between the coronal co-ordinate frame and the tilted co-ordinate frame. If `tcE` is multiplied by a point in the coronal space, the result is the same point's position in the tilted space.

- `tcI`: brightness adjustment co-efficient for registration. Given a voxel in the coronal dataset, the intensity must be multiplied by `tcI` to obtain the correct intensity in the tilted dataset.

Only `phtico_seg` and `phtico_fg_mask` are included in the file in the examples subdirectory.

5 Viewing results

To view the volume data, there are a number of useful commands.

5.1 volshow

The `volshow` command is used to show a single slice from a volume.
```
volshow( volume, sliceNum, slicePlane )
```

Parameters

- `volume`: the volume to display.

- `sliceNum`: the slice number within the volume to display. Default is 1.

- `slicePlane`: the plane to take slices in. Default is `'z'`.

Example

To display the midsaggital plane of the registered dataset, type:
```
volshow( phtico, 73, 'x' )
```

5.2 volanim

The `volanim` command is used to show multiple slices from a volume using an animation.
```
volanim( volume, slicePlane, delay )
```

Parameters

- `volume`: the volume to display.

- `slicePlane`: the plane to take slices in. Default is `'z'`.

- `delay`: the time between slices, in seconds. Default is 0.01.

Example

To show the coronal dataset from a front view, starting with the lips and moving back to a slice through the eyes, type:
```
volanim( co, 'z' )
```

5.3 vtiso

The `vtiso` command is used to show a 3-D isosurface rendering of a volume.
```
vtiso( volume, isovalue, colour, doCLF, doSmooth )
```

Parameters

- `volume`: the volume to display.

- `isovalue`: the isovalue to use when displaying.

- `colour`: the colour to use for the isosurface. Default is fleshlike $\begin{bmatrix} 1 & .75 & .65 \end{bmatrix}$.

- `doCLF`: clear the figure window before rendering isosurface. Turn this off if you want to render multiple isosurfaces at once. Default is on (1).

- `doSmooth`: smooth the volume data before rendering isosurface. Speeds up rendering, but may change surface unacceptably. Default is on (1).

Example

To display the vocal tract without smoothing, type:
```
vtiso( phtico_seg, 15000, [.8 .5 .45], 1, 0 )
```

5.4 vtisocutaway

The `vtisocutaway` command is used to show 3-D isosurface renderings of both the vocal tract and a cutaway version of the head simultaneously.

Parameters

- `phtico_seg`: the segmented vocal tract dataset.

- `phtico_fg_mask`: the head dataset.

- `pad`: how much of the head to cutaway. A value of 0 corresponds to half, and positive values cut away more of the head. Default is 0.

Example

To display the vocal tract and head, type:
```
vtisocutaway( phtico_seg, phtico_fg_mask )
```

References

[EB99] O. Engwall and P. Badin. Collecting and analysing two- and three-dimensional MRI data for Swedish. *Kungl Tekniska Hogskalen Institutionen for Tal, Musick och Horsel Quarterly Progress and Status Report*, 1999.