LANDMARKER: A VTK-BASED TOOL FOR LANDMARKING OF POLYGONAL SURFACES

Tron A. DARVANN

3D Craniofacial Image Research Laboratory
(School of Dentistry, University of Copenhagen;
Copenhagen University Hospital;
Informatics and Mathematical Modelling, Technical University of Denmark)
Denmark
trd@odont.ku.dk

1. ABSTRACT

Aim: To present "landmarker" and its functionality for landmarking and visualizing surfaces. Furthermore, to share the experience of developing and using landmarker for cephalometric purposes. Finally, to make the application and code available to the community.

Methods: landmarker consists of a collection of Tcl/Tk procedures that has been written and extended since 1999. The procedures use the Visualization ToolKit (VTK) library which is an open source, freely available software system (www.kitware.com). The setting has been a multi-disciplinary research laboratory where engineers, dentists and medical doctors work together on projects involving analysis and visualization of 3-dimensional medical, mainly craniofacial, data. Often, there has been a lack of a tool for landmarking, inspection or manipulation of surfaces, and using VTK with Tcl as a script language has in many cases provided a reasonable customized solution with a reasonable amount of effort.

Results: The main functionality of landmarker is a graphical user interface for manual point landmarking of surfaces. It supports e.g. numbering/ordering/naming of landmarks; loading of predefined landmark sequence and names; saving/loading/editing landmark files. Other functionality: registration/transformation/smoothing/decimation/animation of surfaces, coloring based on input from external file, visualization of lines and curves, and more.

Conclusions: landmarker is free for non-commercial use as long as a reference is given in any publication to which it has contributed. The first public version will be available for download "as is" (without user support) by mid-January 2008, for Windows and Linux.

2. INTRODUCTION

landmarker consists of a collection of Tcl/Tk [1] procedures and C++ programs that use the Visualization ToolKit (VTK) [2] library which is an open source, freely available software system (www.kitware.com).

3. VERSIONS

landmarker exists in three different flavours: 1) An executable version (.exe) for Microsoft Windows (2000, XP and Vista) created using VTKit from www.bioengineering-research.com/vtk/vtkit.htm. This version is easy to install, but contains no source code and does not allow Tcl scripting. 2) A Tcl/Tk open source version for Microsoft Windows. This version is relatively easy to install, but requires installation of a binary executable version of VTK (downloadable from www.kitware.com). It includes all source code, and allows scripting (the Tcl procedures of landmarker may be called from your own Tcl applications). 3) A Tcl/Tk open source version for Linux. This version is somewhat less easy to install as it requires compilation of VTK, as well as compilation of the parts of landmarker that have been written in C++. The main functionality of landmarker (the parts written in Tcl/Tk) will run under linux without compilation, e.g. together with the binary distribution of VTK now included with Fedora.

4. FUNCTIONALITY

An overview of the functionality of landmarker is given in the following.

Tools for point landmarking of polygonal surfaces:

- Simple GUI for manual point landmarking of polygonal surfaces
- Landmark numbering/ordering/naming/
- Loading of predefined landmark sequence and names
- Coloring of landmarks
- Saving landmark files in simple text format
- Loading and editing of previously saved landmark files
- Merging of landmark files
- Merging of landmarks based on distance
- Creation of spline curve from point landmarks, save as simple text file (.sln)
- Loading and visualization of multiple landmark files

- Some support for landmarking in orthogonal and oblique image slices

Visualization of polygonal surfaces, landmarks, lines and curves:

- Surface color, transparency
- Save snapshot of scene as .jpg, .bmp, .tif
- Multiple surfaces
- Animation of series of polygonal surfaces (e.g. time evolution)
- Load animation file containing list of series of polygonal surface files to be animated
- Color surface according to scalar (color per triangle) by reading native color (.clr) file
- Display and color lines and curves by reading native text files (.crv, .sln)

Tools for manipulation of polygonal surfaces:

- Read surfaces in .stl, .vtk, .g, .flex, .obj, .cyb, .tri formats.
- Write surfaces in .stl, .vtk formats.
- Cutting (by plane, by sphere, by pattern of landmarks)
- Merging
- Flipping (mirroring) around specified axis
- Arbitrary transformation by editing of transformation matrix
- ICP registration of surfaces
- TPS transformation of surfaces
- Landmark based registration: scaling, similarity, rigid body, affine transformations
- Connected region extraction
- Smoothing, Decimation, Subdivision, Cleaning
- Triangulation of surface delineated by curve
- Retrieve information on number of polygons/points
- Deformation of surface to match other surface (closest point deformation)
- Deformation of curve to match surface (closest point deformation)
- Interpolation between surfaces

Measurement:

- Distance between landmarks
- Calculation of surface area and volume (closed surfaces only)
- Closest point distance from all triangle centers in one surface to another surface
- Closest point distance from point to surface
- Intersection point between surface and line

Creation of polygonal surfaces:

- Surface from interactive image thresholding and use of Marching Cubes algorithm
- Triangulation of surface delineated by landmarks
- Delaunay triangulation of surface from points

Some support for volumetric 3D images:

- Read Analyze™ 7.5 image files (.hdr/.img) (only if written with same byte ordering as on the machine where it is read)
- Read landmarker native image format (.hea/.img)
- Write landmarker native image format (.hea/.img) with header in simple ASCII text format
- Interactive window/level
- Image invert
- TPS transform
- Orthogonal and oblique slices
- Interactive thresholding by single threshold

Scripting:

- Scripts may be written in Tcl calling landmarker's procedures, e.g. for repeated manipulation/visualization of many surfaces (only possible using one of the open source versions of landmarker)

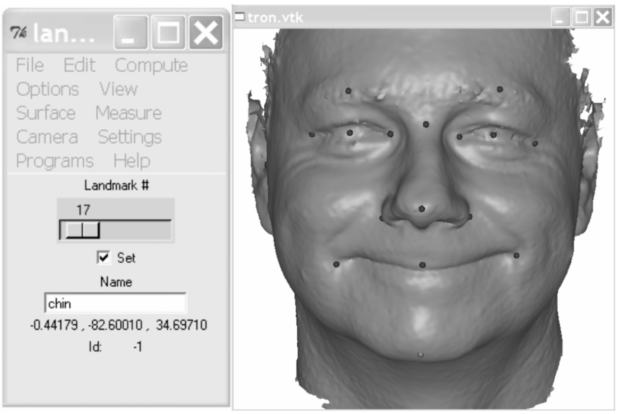
5. DOWNLOAD

landmarker may be downloaded from the following website: http://www.lab3d.odont.ku.dk/landmarker/. It is free for non-commercial use as long as a reference is given in any publication to which it has contributed. Before downloading, please read and accept the Disclaimer and Copyright notice provided on the homepage. Also, some example data files may be downloaded from the website.

6. EXAMPLES

6.1 Landmarking a Surface

landmarker may be used in order to point-landmark a polygonal surface. Figure 1 shows the Main Menu and the Surface Window after a face surface has been loaded and landmarked.



Landmark #
17
☑ Set
Name
chin
-0.44179 , -82.60010 , 34.69710
Id: -1

Figure 1. The landmarker application after a face surface has been loaded. Left: Main Menu; Right: Surface Window.

After starting landmarker, a surface may be loaded via the Main Menu by “File->Load(New)->Surface File”. The Main Menu contains a landmark number slider (indicating the number of the Current Landmark), a “Set” button (indicating whether the Current Landmark has been placed or not), a landmark Name text entry field and the 3D coordinate of the Current Landmark. Before any landmark has been placed, the landmark number slider indicates landmark 0. A landmark may be placed by moving the cursor by use of the mouse to the location of interest (e.g. after rotating and zooming the surface for a good view) and pressing keyboard character “p” (for “pick”), resulting in a sphere being displayed at the surface under the cursor. Fine adjustments of the landmark position may be carried out by moving the cursor and pressing “p” repeatedly until the desired landmark location has been achieved. Another landmark may be placed by advancing the landmark number slider by one, then pressing “p” again etc. In the example in Figure 1, 18 landmarks (counting from 0) have been placed, showing the landmark 17 as the Current Landmark named “chin”, and indicated by a landmark with brighter color than the other landmarks. An efficient way to place many landmarks on a surface is to use the left hand to press “p” while moving the cursor and advancing the landmark number slider with the mouse using the right hand. Often a set of similar surfaces are to be landmarked (e.g. for population studies) and in order to facilitate keeping track of landmark locations and their order, a predefined sequence of landmark names may be loaded from file. Sometimes it is impossible to place one or more particular landmarks on a surface (e.g. due to imperfections of the surface), and these landmarks may be left out and “flagged” as such by unsetting the “Set” button. After all landmarks have been placed, their coordinates, names and “Set” status (on or off) may be saved to a file in simple text format.

6.2 Visualizing a Surface with Landmarks via Menu

As an example, here is how to visualize a surface with landmarks already saved to a file, then saving a snapshot of the resulting scene in the Surface Window (e.g. as the scene shown in Figure 1):
1) Load surface file: File->Load(New)->SurfaceFile
2) Load landmark file: File->Load(New)->LandmarkFile
3) Save a snapshot: Save->Snapshot

6.3 Visualizing a Surface with Landmarks via Script

The same task as described in Section 6.2 may be achieved by executing a Tcl script calling landmarker’s procedures. The following Tcl code demonstrates this:

```
source C:/programs/landmarker/landmarker.tcl
ReadSurfaceFile tron.vtk
ReadLandmarkFile tron.log
SaveSnap snap.jpg
```

Here, the first line starts landmarker and at the same time makes its procedures (e.g. ReadSurfaceFile, ReadLandmarkFile and SaveSnap) available to the rest of the script.
Figure 2 presents example visualizations created by use of landmarker.

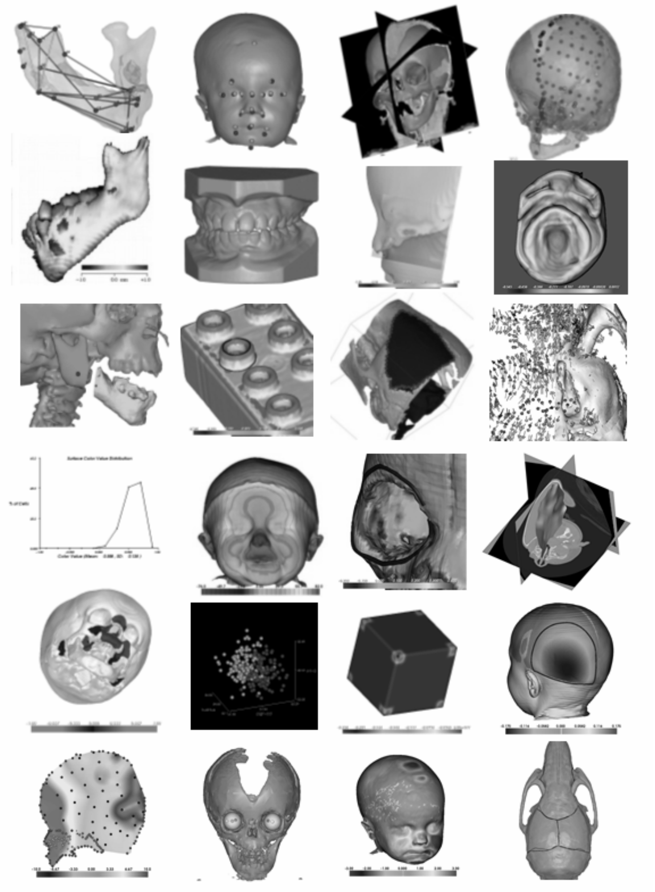


Figure 2. Example visualizations created by landmarker.

7. ACKNOWLEDGEMENTS

Contributions to the development of landmarker have been made by Per Larsen, Nuno Hermann, Stéphanie Lanche, Dan Govier, Nao Kakimoto, Hildur Ólafsdóttir, Aliyah Shivdij, Andrea Van Pelt, Bjarne K. Ersbøll and Sven Kreiborg. The user interface for landmarking was inspired by the Mvox-software [3] by Morten Bro Nielsen. Thanks to the VTK developers for providing VTK, and to David Hume for the Tcl La linear algebra package (http://www.hume.com). The initial creation of landmarker was facilitated by a grant from The Smile Train Foundation to Nuno Hermann.

References

[1] Schroeder W, Martin K, Lorensen B. The Visualization Toolkit. An Object-Oriented Approach To 3D Graphics. 3rd Edition. Kitware, Inc. publishers 2004
[2] Welch, BB. Practical Programming in Tcl and Tk. 2nd Edition. Prentice Hall 1997
[3] Bro-Nielsen M. Mvox: Interactive 2-4D medical image and graphics visualization software. In: H.U. Lemke, M.W. Vannier, K. Inamura, A.G. Farman (eds): Proc. Computer Assisted Radiology (CAR'96), Paris, France, June 26-29, pp. 335-338, Elsevier, Amsterdam, 1996