Usage notes for the DLTdv7 MATLAB based digitizing program

last updated: Ty Hedrick, August 28th, 2018

DLTdv7 is a digitizing (i.e. video annotation) environment written in MATLAB designed to acquire 3D coordinates from 2-9 video sources calibrated via a set of Direct Linear Transformation (DLT) coefficients and distortion-corrected using an arbitrary transform that can accommodate pinhole or fisheye lens models. It can also digitize uncalibrated videos, recording only the X,Y locations of the markers in one or more videos.

Table of Contents

System Requirements:

DLTdv7 requires the base MATLAB environment and works in Windows, Linux and Mac OS. It was developed and tested with MATLAB r2017a, r2017b and r2018a but will likely work in any newer MATLAB version.

Changes in version 7:

- Zoom in or out of images using the mouse wheel
- Work with videos that have different frame rates
- Multi-track is back after going missing in DLTdv6
- Save and load project files that bundle all the video locations, calibrations, data and interface state together
- Export data as *.csv (flat) or *.tsv (sparse) for use elsewhere
- Timeseries window tracks are clickable

Quickstart instructions for DLTdv7

See https://biomech.web.unc.edu/dltdv/ for additional information and tutorial data

1) Place the DLTdv7.m file in your /home/user/Documents/MATLAB directory or otherwise add it to the MATLAB path

2) Run DLTdv7 by entering the command name into the MATLAB console and pressing enter; the Controls widow should appear

3) Click on the "New Project" button in the Controls window

4) Select the number of movie files you intend to simultaneously digitize

5) Browse to the first video file and open it; repeat the process for additional video files

6) If you have calibrated cameras and more than 2 videos you will be asked whether or not you wish to load a DLT calibration coefficients file - load it if available & desired, see below for more information

7) If you have non-linear distortion transform files you can add them by clicking the “Set Undistortion profile” button on each camera window

8) Begin digitizing with the following commands and keystrokes:

left click (or Mac mouse click) - digitize a point

right click (or Mac control-click) - remove a digitized point

f key - forward one frame

b key - back one frame

= key - zoom in around the mouse pointer

- key - zoom out around the mouse pointer

r key - restore the original zoom level

mouse wheel down: zoom in around the mouse pointer

mouse wheel up: zoom out around the mouse pointer

9) Explore the auto-track options if desired, see details below for more information

10) Click the "Add a Point" button to digitize more than one point through the video sequence

11) Click the "Save project" button to save the project, use Export points to export data in text format for other applications

12) Click "Quit" or close one of the DLTdv7 windows

Detailed Usage Instructions

Initializing

When you first start the DLTdv7 program, most of the interface is blank or disabled, only the “New Project”, “Load Project” and “Quit” buttons are active. “Quit” exits the program immediately, “Load Project” starts analysis by browsing for a previously saved (*dvproject.mat) project file, “New Project” starts the analysis process by first bringing up a dialog box asking how many videos you intend to digitize (1 – 9). After noting the number of videos you intend to digitize, you'll need to select the video files, one after the other. Finally, you'll have the option to load DLT coefficients. The DLT coefficients file should be a comma delimited matrix with one column for each of the cameras you're digitizing from and no header, see further details below in the DLT Coefficients sections. After picking the video file(s) (and coefficients file if desired) the initial video frames are displayed and the rest of the interface is activated. At this point the program is initialized and the "New project" button is replaced with a “Recompute 3D points button” and the “Load project” button with a “Save project” button. If you've made an error in creating a new project simply Quit and restart.

Recompute 3D points

After your project is initialized, the "New Project" button is replaced with "Recompute 3D points". This will let you pick and apply a different set of DLT coefficients without having to re-click your file. Note that if you create a new project and then load points in from a text file, if you don't pick the same DLT coefficients you used before your 2D points, 3D points and DLT coefficients will not be in agreement and you should immediately use the Recompute 3D points button to re-select your intended DLT coefficients; this will re-synchronize everything.

Acquiring data - mouse clicks and keystrokes

Points are acquired by clicking on the appropriate location in one of the video frames

left click

digitizes a point in the frame you clicked in (or prints a warning if it is unable to do so). The digitized point is shown by an empty red circle. If the program is in DLT mode it then either draws a blue line of zero DLT reconstruction error on the other video frames or draws a green diamond where the reconstructed 3D point falls given the existing digitized locations and DLT coefficients.

right click

removes the digitized point in the frame where the click occurred. If possible, DLT information is updated accordingly. On a Macintosh with a one button mouse you may enter "right" clicks by holding down the Control (Ctrl) key and clicking the mouse.

f key

moves **forward** one frame in all video streams

b key

moves **back** one frame in all video streams

= key

zooms the current video frame **in** around the mouse pointer

- key

zooms the current video frame **out** around the mouse pointer

r key

restores the original zoom; the mouse scroll wheel will also zoom in and out around the pointer location.

Please see the KeyboardMap PDF document for an exhaustive list of all keyboard shortcuts.

Video controls [main window]

The video display is controlled by the elements in the blue section of the controls window.

video gamma: a slider control that changes the video image intensity map to make the images lighter or darker

frame number: a slider that sets the position within the video streams

display in color: turns color display on and off for color videos

Subtract background: performs simple background subtraction to highlight moving objects. For performance reasons, background subtraction operates out of the image cache and will not work if you skip to a frame with the slider or text box, but will if you advance through a sequence of frames.

video offsets: each video window includes an offset box that let you adjust the relative position of the video streams. For example, if the offset entries are 0, -1, 1 then video #2 frame n-1 and video #3 frame n+1 are shown with video #1 frame n in the display. Offsets are always relative to the first video.

Video controls [movie windows]

Frame rate (Hz): DLTdv7 will auto-detect the frame rate of the video or you can type the known frame rate in. The relative frame rates of the different videos matter, but the actual frame rates do not at this time.

Frame offset: Adds an offset to the timebase of the video displayed; the first video always has an offset of zero; subsequent videos can have a positive or negative offset. Units are frames of the video in question.

H. mirror: Performs a horizontal mirror transform on the image

Set Undistortion profile: Allows the user to set a nonlinear undistortion operation specified as a MATLAB distortion/undistortion transform pair saved in a separate MATLAB data file. The easyWand5 wand-calibration tool produces output compatible with DLTdv7.

Point and auto-tracking controls

Auto-tracking and multiple point functions are controlled by the elements in the green section of the controls window. Autotrack mode *off* employs no autotracking, *semi* advances one frame and uses the autotracker to guess the point location in the new image but then waits for user input, *auto* advances one frame, guesses a point location in the new frame, and if the fit is good enough advances again without user input. Autotrack mode can be changed while the program is running in *auto* mode. This may be necessary if the autotracker locks on to a static portion of the image. Autotrack *multi* mode is only

available by use of the menu rather than the menu keyboard shortcut (the *x* key); *multi* mode acts like *semi* mode but operates over all points not just the active point, either operating over all videos or only the current video if the "Update all videos" checkbox is unchecked. A mouse-wheel click in *multi* mode will begin automatic multi mode, where all current points are tracked with auto-advance.

The autotracker functions by trying to find a match between a small group of pixels around the known point in the current video frame with an equivalently sized group of pixels in the next frame. The group of pixels in question is displayed in the **Autotrack search image** section of the controls window. The size of this small group of pixels is controlled by the "**Autotrack search area size**" field in the controls window. If the match between the pixels in the current and next frames (the **Autotrack fit**) is greater than the "**Autotrack threshold**" then the autotracker proceeds to examine the next frame (auto mode) or draws the next frame and new point on the screen but waits for additional user input (semi mode). The most appropriate search area size and threshold values depend on the quality of the video recording, size and contrast of the markers and so on. Some experimentation will be needed to determine the best values, reasonable starting points are provided as defaults. The autotracker returns integer X and Y coordinates. If a 3D point can be calculated during autotracking operations, autotrack success also depends on the DLT residual remaining below the value in the "**Threshold:**" text box.

The "**Autotrack predictor**" menu changes the routine used to predict the next location of the point during autotrack operations. *extended Kalman* is best for most moving points and *static point* is best for points that move little or not at all.

The **Add a Point** button creates the data structures for a new point and switches the interface to place new inputs into that point. Values for other points are displayed as light blue circles and diamonds. The **Current Point** switches the interface between different points created via the Add a Point button.

Update all videos controls whether point tracking and frame advance operations apply to all videos or only the current video window. Updating a single video window is faster, so you may wish to uncheck this option for speed.

DLT visual feedback controls whether feedback related to the 3D calibration, i.e. the blue epipolar line and green ideal point location diamond are shown. If your calibration is high quality you may wish to use this feedback to help identify less-visible points.

Show 2D tracks controls whether or not DLTdv should display the time-series of each point on the movie images. The length of the sequence displayed is the same as the window width in the DLTdv timeseries window.

Find marker centroid uses MATLAB image analysis toolbox functions to try and find a black or white marker (set by the Color menu) in the autotrack region. It complements the existing predictive and pattern based autotracker.

Timeseries window

In addition to the controls window and the movie windows, DLTdv includes a timeseries window that lets you display the time series of either the [x,y] (i.e. [u,v]) points clicked in one of the cameras or one of the [X, Y, Z] 3D coordinates. The active point is in red, other points are in blue but can be made active by clicking in the timeseries window. The "Window width (frames):" textbox controls how many frames are displayed in the timeseries window; it defaults to the whole file.

Saving Data

DLTdv7 can save its output as a project file or as a set of text files. The project file is a MATLAB data file that includes the paths to the videos, all interface settings, all clicked data and the calibration information; it can be used to rapidly reload a project after working, saving and stopping.

Like previous DLTdv versions, DLTdv7 can also save the accumulated data as a set of text files in "flat" or "sparse" format.

The flat format is a comma delimited text files with one row for each frame in the video file.

[prefix]xypts.csv – a comma separated data file with X1, Y1, X2, Y2, etc. for each frame

[prefix]xyzpts.csv – a data file with X,Y,Z DLT output for each frame

[prefix]xyzres.csv – a data file with the DLT residual for each frame

[prefix]offsets.csv – a data file with video1 offset, video2 offset, etc. for each frame

Each of these files has a minimal, auto-generated header line

If the Spline toolbox is installed, DLTdv7 can generate spline-smoothed points to within the 95% confidence interval of the measurement. This is presented as an optional task during while saving as it relies on a Monte Carlo approach and can take a while for large data sets. If generated, this will result in two additional files:

[prefix]xyzCI.csv - a data file with the +-95% confidence intervals for each value in [prefix]xyzpts.csv

[prefix]xyzFilt.csv - a data file with the 3D coordinates smoothed to fall within the 95% confidence intervals 95% of the time

The "sparse" format creates *.tsv (tab-separated) output that describes the data arrays as a list of [row,column,value] entries, but only for non-null data. This dramatically reduces the file size for long videos where only a few frames where clicked, but is more difficult to examine or modify in other programs.

Loading Data

DLTdv7 can load previously saved projects or text file data. Projects can only be loaded immediately after starting DLTdv7. In principle, text data could be loaded at any time following initialization. In practice, I suggest that data be loaded immediately following initialization. Although some efforts are made to ensure that the loaded data structures match the videos, in practice it is not possible to be certain that the videos and digitized point data match - caution on the part of the user is recommended! Finally, the text file load function attempts to determine the proper video offsets, but may fail to do so if the offsets changed midway through the video sequence.

Direct Linear Transformation (DLT)

DLT is a reasonably straightforward method for calibrating cameras such that images from two or more cameras can be used to reconstruct point locations in three dimensions. A complete overview of this method and some of the alternatives is well beyond the scope of a simple help file, I recommend that the interested reader visit http://www.kwon3d.com for an excellent introduction. DLT residuals result when the [X,Y] pairs from the cameras do not result in a perfect solution and are the mean square error in pixels about the [X,Y,Z] location returned by the DLT operation. Note that a pixel may not represent an equivalent distance in real units along each of the separate axes. The DLT residual for the current point is displayed in the Controls window and the collection of residuals is saved in the [prefix]xyzres.csv file. Note that placing the second [X,Y] pair on the blue "line of zero DLT residual" or epipolar line will result in a residual near zero; examination of the scope of the line should convince the reader that the DLT residual is an imperfect and incomplete measurement of the reconstruction error. Furthermore, reliance on the blue line during the digitizing process may bias the user toward smaller residuals that might be obtained by digitizing each point separately, without using information obtained from other points.

DLTdv7 does not generate DLT parameters for the different video sources, instead it loads a pregenerated set of DLT coefficients stored in a comma separated text file. These DLT coefficients can be generated in many ways, one of which is use of the companion program DLTcal5.m for static calibration points or easyWand5.m for wand-calibration; please see the appropriate help file for additional information.

Movie file formats and manipulation

DLTdv7 relies on MATLAB's built in media reading abilities as provided by the mmreader function or VideoReader class. In principle, this should allow DLTdv7 to read any AVI, mp4, or MOV files that can be decoded by your operating system. In practice, your ability to read files depends on whether or not the appropriate media codec is installed on your computer. Furthermore, performance in reading heavily compressed video files can be poor, consider testing different file save formats from your camera hardware and trying DLTdv7 on different computers to determine which format works best for you.

DLTdv7 ignores color information by default but can display it via a user-selectable option in the interface. Color information is not used in the automatic point tracking routines, the color image is converted internally to grayscale.

DLTdv7 can also read uncompressed versions of the Vision Research (Phantom Camera) *.cin or *.cine movie format and the IDT/Redlake *.mrf uncompressed raw format.

Performance and tuning

MATLAB has some intrinsic limitations that make DLTdv7 slower than equivalent software implemented in other programming environments or languages. The slowest operation in the program is reading the next video frame(s) from disk and then copying them to the screen. This means that higher resolution videos will reduce performance, and digitizing two videos simultaneously is approximately twice as slow as digitizing one. To achieve the best possible performance the movie files should be copied to a local hard disk attached either internally or via USB 3.0, Firewire or lightning drive enclosure. Graphics card performance can also be a problem on some systems, especially laptops and Linux systems, reducing the size of the digitizing window and zooming in rather than expanding the window to full screen may speed up the graphics performance. Finally, unchecking the "Update all videos" check box has DLTdv7 only load the next frame of the active video, greatly speeding operations. Other videos get their frames loaded as soon as they're made the active window.

Exposing subfunctions

DLTdv7 includes many useful functions embedded as subfunctions in the main program. If you're interested in using any of these simply copy them from the DLTdv7 file and make sure that they're in the MATLAB path.

Bugs

If you encounter any bugs, feel free to fix them and send me the fix for incorporation into the copy on the bitbucket repository! If you're not able to track down the bug, please send in a bug report; describing the bug and the situation that triggers it as completely as possible. Bug reports should be sent to Tyson Hedrick, <thedrick@bio.unc.edu>.

Modifying the auto-tracking predictor algorithm

In addition to the image matching routine described earlier, the auto-tracker also attempts to predict the location of the point in the next frame by fitting an equation to previously digitized points and extrapolating the position of the point in the next frame. This location is then used as the center point for the image mapping search. The current algorithm for predicting the point location is a linear Kalman filter, with linear fit and static point predictions as a backup. The Kalman filter performs well in a wide variety of

circumstances, but may not be the best choice for all conditions. If you have special requirements or a better formal description of the underlying system generating the point sequence you can write your own subfunction and add it to the set of options. See the "AutoTrackPredictor" subfunction in DLTdv7.m for the framework for adding your predictor.

License, usage agreement and citations

This program is provided as-is, no warranty is provided. I encourage users to make improvements and fixes to the software as the mood strikes them; substantial or useful additions should be returned to the community by emailing the improvements to me (Tyson Hedrick, thedrick@bio.unc.edu) or asking for access to the DLTdv bitbucket repository at https://bitbucket.org/thedrick/dltdv.

The program should be referred to in the text of a scientific publication as custom digitizing software with a citation to:

Hedrick, T. L. (2008), Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems, Bioinspiration & Biomimetics