3d models from archival film/video footage

Shawn Graham, Carleton University

January 19 2018

It is possible to make 3d models from archival film/video footage, although the *quality* of the resulting model may require a significant amount of sculpting work afterwards to achieve a desireable effect. It depends, really, on why one wants to build a 3d model in the first place. Archaeologists for instance might want to work with a 3d rendering of a building or site now lost.

The workflow

The workflow has a number of steps:

1. obtaining the video (if it is on eg. youtube)
2. slicing the video into still images
3. adding camera metadata to the images
4. computing matched points across the images
5. triangulation from the matched points
6. surface reconstruction

Necessary software

nb these are all open-source or free-to-use programs

1. Youtube-dl https://rg3.github.io/youtube-dl/
2. ffmepg https://www.ffmpeg.org/
3. exiftool https://www.sno.phy.queensu.ca/~phil/exiftool/
4. regard3d http://www.regard3d.org/
5. meshlab (for post-processing) http://www.meshlab.net/

Step One Downloading from Youtube

Archival or interesting footage of all kinds may be found on youtube and other video streaming services. Youtube-dl is a sophisticated program for downloading this footage (and other associated metadata) from youtube and some other sites. Find a video of interest. Note the url. Then:

```
youtube-dl https://www.youtube.com/watch?v=nSB2VeTeXXg
```

Try to find video that does not have watermarks (the example above has a watermark and probably is not the best source video one could use). Look for videos that are composed of long cuts, that sweep smoothly around the site/object/target of interest. You may wish to note the timing of interesting shots, as you can download or clip the video to those passages (see the youtube-dl documentation)

Step Two Slicing the Video into Stills

ffmepg is a powerful package for manipulating video and audio. We use it to cut the video into slices. Consult the full documentation to work out how to slice at say every 5 seconds or 10 seconds (whatever is appropriate to your video). Make a new directory in the folder where you've downloaded the video with `mkdir images`. Then the command below slices at every second, numbers the slices and puts them into the `frames` subdirectory:

```
ffmpeg -i "downloaded-film.mp4" -r 1 frames\images-%04d.jpeg
```

Windows users would call ffmpeg with ffmepg.exe (if they haven't put it into their system's path variable).

Step Three Adding Camera Metadata

We will be using Regard3d to stitch the images together. Regard3d needs to know the camera make, model, focal length (mm), and sensor width (mm). We are going to fudge this information with our best approximation. 'Sensor width' is the width of the actual piece of hardware in a digital camera upon which light falls. You'll have to do some searching to work out the best approximation for this measurement for the likely camera used to make the video you're interested in.

Find the camera database that Regard3d uses (see the documentation for Regard3d for the location on your system). It is a csv file. Open it with a text editor (eg Sublime Text or Atom. **not** Excel, because Excel will introduce errors). Add the make, model, and sensor width information following this pattern:

```
make;model;width-in-mm
```

Regard3d reads the exif image metadata to work out which camera settings to use. Focal length is read from the exif metadata as well. We assign these like so, from the command line in your `frames` folder:

```
exiftool -FocalLength="3.97" *.jpeg

exiftool -Make="CameraMake" *.jpeg

exiftool -Model="CameraModel" *.jpeg
```

Note that the make and model must absolutely match what you put into the camera database csv file - uppercase, lowercase, etc matters. Also, Windows users might have to rename downloaded exiftool file to `exiftool.exe` and put it into their path variable (alternatively, rename it and then put it in the `frames` folder so that when you type the command, your system can find it easily).

Step Four Computing Matches

Open Regard3d and start a new project. Add a photoset by selecting your `frames` directory. Note that when you used the exiftool, the original images were copied within the folder with a new name. Don't select those original images. As the images load up, you will see whether or not your metadata is being correctly read. If you get NaN under make, model, focal length, or sensor width, revisit step three again carefully. Click ok to use the images.

Click on compute matches. Slide the keypoint density sliders (two sliders) all the way to 'ultra'. You can try with just the default values at first, which is faster, but using 'ultra' means we get as many data points as possible, which can be necessary given our source images.

This might take some time. When it is finished, proceed through the next steps as Regard3d presents them to you (the options in the bottom left panel of the program are context-specific. If you want to revisit a previous step and try different settings, select the results from that step in the inspector panel top left to redo).

The final procedure in model generation is to compute the surfaces. When you click on the 'surface' button (having just completed the 'densification' step), make sure to tick off the 'texture' radio button. When this step is complete, you can hit the 'export' button. The model will be in your project folder - .obj, .stl., and .png. To share the model on something like Sketchfab.com zip these three files into a single zip folder. On sketchfab, you upload the zip folder.

Step Five Clean Up

Double click on the .obj file in your project folder. Meshlab will open and display your model. The exact tools you might wish to use to enhance or clean up your model depends very much on how your model turned out. At the very least, you'll use the 'vertice select' tool (which allows you to draw a box over the offending part) and the 'vertice delete' tool. Search the web for help and examples for the effective use of Meshlab.