Developing and customizing ultrasound image processing tools with MATLAB

Catherine Laporte

École de technologie supérieure, Montreal, Canada

catherine.laporte@etsmtl.ca

Background: Over recent years, thanks in great part to researchers from the Ultrafest community, several user-friendly software packages have been developed to facilitate and/or automate the analysis of vocal tract ultrasound images (e.g. Articulate Instruments, 2010; Li et al., 2005; Laporte, 2018; Tiede, 2020). While these tools contribute to improving the quality and reproducibility of research in articulatory phonetics, they do not necessarily meet the demands of every conceivable study. Even if many of these tools are open-source, their customization can remain a significant challenge. Furthermore, ultrasound images pose unique challenges that are not typically addressed by basic image processing tutorials or textbooks. MATLAB offers a suitably programmable platform, along with a mature Image Processing toolbox, that can be built upon to develop customized ultrasound image analysis applications. In addition, several third-party MATLAB packages exist that can be used to this end as well. A hands-on tutorial will be presented covering basic image processing operations, as well as more advanced, ultrasound appropriate methods.

General purpose image processing tools: MATLAB's Image Processing Toolbox offers basic gray level image processing functionality, including various noise reduction, contrast enhancement and edge detection filters based on local pixel neighborhood analysis, as well as automated gray level threshold selection based on global pixel intensity statistics. A good understanding of the inner workings of these basic methods and their shortcomings is useful when handling ultrasound images, as these often fail to satisfy the underlying assumptions.

Complementary tools for ultrasound image processing: Many more advanced image processing tools have been developed by the computer vision, pattern recognition and medical image analysis research communities over the past decades that are well-suited to ultrasound image processing in general, and vocal tract ultrasound image processing in particular. Many of these tools have mature, open-source MATLAB implementations. The phase symmetry (Kovesi, 1997) and Frangi filters (Frangi et al., 1998) are of particular interest to detecting tongue surface features from ultrasound images (Karimi et al., 2019; Naga Karthik et al., 2020). Both these filters are designed to robustly detect ridge-like structures from image measurements performed at multiple scales. Phase symmetry is a measurement of simultaneous even symmetry and absence of odd symmetry across multiple scales and feature orientation in images. The Frangi filter, on the other hand, measures the directionality of the local image Hessian (essentially, the curvature of image pixel intensity), also evaluated at multiple scales. When combined, these tools offer a powerful means of isolating the bright ridge caused by the reflection of ultrasound off the tongue surface.

Binary image processing: The MATLAB Image Processing Toolbox also provides a set of morphological operators that can be used to process blobs (also known as *connected components*) in black and white images, previously obtained by suitably thresholding a gray scale image. These operations include dilation, erosion and combinations thereof which can be used to fill in holes or remove noise in binary images, as well as various useful measures of blob geometry. A task that arises rather commonly in image analysis amounts to deciding which blobs in a binary image (obtained previously using feature enhancement and thresholding methods on a gray scale image) are part of a meaningful structure of interest, and which should be discarded. Automatic clustering of lit pixels based on their proximity is often useful in such cases. The MATLAB Statistics and Machine Learning Toolbox offers tools to this end, including k-means clustering, which will form a user-defined number of clusters k, whose center locations and membership are iteratively optimized. In many practical cases, the required number of clusters can vary from one data set to the next or be otherwise difficult to estimate *a priori*. DBSCAN (Ester et al., 1996) is an alternative clustering algorithm for which there exist a few different MATLAB implementations, and which optimizes cluster composition based on a minimal cluster size and a maximum distance from a point to its nearest neighbor within its cluster.

Video processing: Video sequences can, and are often processed as a series of independent 2D images. However, in many instances, it can be useful to process information from multiple frames at once. This can range from carrying information from one frame to provide *a priori* information to process the next frame (as is often done in tracking), to processing the entire video sequence as if it were a 3D image (the third dimension, being time). In ultrasound video recordings, one simple but useful application of the latter approach is the computation of a binary 2D image mask that can later be used to discard the non-image content that is typically displayed on the screen of the ultrasound machine (e.g. black background, participant name or pseudonym, scanner settings, etc.) and is often recorded along with the actual ultrasound image data. While these non-image data mostly remain static during a video recording, the ultrasound image data varies over time (e.g. with tongue or probe motion). Thus, the variance of pixel intensity over the time dimension can be used towards computing the required mask.

Conclusion: While automated ultrasound image analysis remains challenging, there now exists a wealth of documented algorithms along with exploitable software implementations thereof. Improved functionality beyond the type of methods discussed here can be achieved by leveraging application-specific constraints, e.g. through direct modeling or machine learning.

References

Articulate Instruments. (2010). Articulate Assistant Advanced; v 218.04 [Computer program] http://www.articulateinstruments.com/downloads/

Ester, M. *et al.* (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pp. 226-231.

Frangi, A.F., *et al.* (1998). Multiscale vessel enhancement filtering. In *MICCAI*, pp. 130-137.

Karimi, E. *et al.* (2019) Fully-automated tongue detection in ultrasound images. *Computers in Biology and Medicine* 111:103335.

Kovesi, P. (1997). Symmetry and asymmetry from local phase. In *10th Australian Joint Conf. on Artificial Intelligence*, Vol 190, pp.2-4.

Laporte, C. (2018). SLURP: Speech and Language Ultrasound Research Package. [Computer progam] https://github.com/cathylaporte/SLURP.

Li *et al.* (2005). Automatic contour tracking in ultrasound images. *Clinical Linguistics and Phonetics* 19(6-7), 545-554.

Naga Karthik, E.M.V. *et al.* (2020). Automatic tongue surface extraction from 3D ultrasound vocal tract images. *Journal of the Acoustical Society of Amercia*, in press.

Tiede, M. K. (2020). GetContours: tongue contour fitting software; v 2.4. [Computer program]. https://github.com/mktiede/GetContours.

Keywords: image processing, ultrasound, MATLAB