Supplemental Text 1

Development of the oesophageal 3-D Reconstruction

In the initial phase of this study, a combination of inputs, including data from HRM, TBE, and endoscopic images of the oesophagus, from 80 treatment-naive achalasia patients, were retrospectively collected to create a software that automatically reconstructs the oesophageal lumen and the LES 3-dimensionally. The extracted dataset of TBE and OGD images was stored in pseudonymized JPEG format. Pseudonymous HRM data were extracted from the Medical Measurements Systems (MMS) Investigation & Diagnostic Software in fcu format. Subsequently, the pseudonymous fcu files of each patient were imported into the MMS Investigation & Diagnostic Software, wherein one representative HRM swallow sequence was selected and converted into comma-separated values (csv) format. A preliminary examination using distinct selections of HRM swallowing sequences revealed no statistically significant differences. The precise placement of sensors at the midpoint of the LES and directly beneath the upper oesophageal sphincter (UOS) was documented.

The software was developed in Python 3 as a stand-alone web application that can be installed on the Windows operating system by conversion to an executable file (.exe) through PyInstaller(14). The user interface was created with PyQT5(15) where the Plotly(16) generated 3-D reconstruction is embedded as a locally executed web application using Dash(17). This setup aims to enable a broad and easy applicability.

The 3-D reconstruction algorithm expects up to n (restricted by computational performance and time) TBE and optional endoscopic images (.jpeg format). At the same time, the latter include their measuring position in centimeters (cm) in their file name, manometry values exported in .csv format, ordered from lowest to highest sensor on the catheter, and medical user input.

First, the user imports the manometry file and, in the next step, marks measuring points in the TBE image for mapping the data. To map the pressure values of the manometry data and extract the length of the oesophagus in cm later, the user must place two sensor positions of the manometry catheter in the image (bottom to top and preferably far apart). The location of the points can be extracted from the "Laborie stationary measurement & analysis software." The positions should represent the position of the sensor at the middle of the LES and directly below the UOS at the level of the clavicle in the TBS image.

To achieve a more realistic three-dimensionality, endoscopic images can be integrated every 1 cm; otherwise, the 2-dimensional radius in the TBE image is used for approximation. For correct mapping of the endoscopic images, the user must set a zero-reference point along the vertical axis. Their filename includes their height from bottom to top. Additionally, the upper boundary of the LES and the oesophageal outlet are marked in the TBE image to indicate the length of the LES.

Next, the user inserts the circular form of the oesophagus in the endoscopic and vertical shapes in the TBE images for clinical correctness. Since this is a time-consuming step, a first attempt to automate the extraction using the Framework OpenCV-Python(18) was implemented. The underlying logic implies first converting the image to grayscale after removing all redundant black pixels, e.g., OGD image borders. Next, a Gaussian Blur is applied to smooth out noise, and finally, contours surrounding the relevant dark area (i.e., form and diameter) are marked.

This method causes some deviation since the distance of the inserted diameter from the camera is not standardized but is based on the medical user's estimation. The medical user is included in the workflow to double-check the algorithm's performance and enhance trust in the reconstruction. After the user input is complete, our use case-adapted mapping algorithm generates the multi-modal 3-D reconstruction of the oesophagus in a stepwise manner along its vertical shape, since for various forms of achalasia, the oesophagus is not positioned vertically in the body. Therefore, the algorithm iterates through sections of the oesophagus for a piecewise reconstruction that is based on the approximation of a central line to map diameter information correctly, as explained in the following and illustrated in Figure 1a-f in more detail:

- First, the middle point of the upper edge of the oesophagus is determined to calculate the shortest path towards the user-given exit point of the esophagus in the TBE, i.e., the *sensor path* that mirrors the manometry catheter for later mapping of the pressure values. This decision is based on the assumption that gravity leads to the sensor following the shortest path.
- The slope is calculated for each point on the sensor path to construct its perpendicular. The calculation is based on placing a regression line through two points shortly in front of and behind the respective point.
- Next, the resulting perpendicular's slope is stored in the form of a *rotation matrix* for each point along the regression line. This information indicates how much the oesophagus is tilted at this point, which is necessary later to position the oesophagus' cross-sections along its vertical form.
- Then, the intersection points of the vertical line with the shape of the oesophagus from the TBE are determined, and the length of this intersection line is calculated. This length estimates the oesophagus' diameter at this point - if no endoscopic images are available.

- The intersection line's mid-point is calculated for each point along the sensor path, and together, they form the *center path*, which is used for the spatial arrangement of the reconstruction.
- If no endoscopy is present, circles with the extracted 2-dimensional diameter are constructed for all points on the center path. They, in turn, consist of a cumulation of points located along the circle's edge.
- If endoscopy is present, these points do not form a circle but a polygon. For sections of the oesophagus where no endoscopic images are available, the shape of the polygon is interpolated.
- The circles or polygons are then tilted with the rotation matrix according to the calculated angle and spatially shifted to the correct position (their center = center point on the center path).
- Finally, the surface color is mapped with the manometry data, following the Laborie Software color scale. For each frame of the manometry data, the algorithm iterates over the values of the approximated *sensor path* and computes the respective pressure values through distance calculation of the two closest pressure sensor values.

The resulting approximation of the patient's oesophagus displays color-scaled pressure values whose movement can be tracked throughout the entire duration of the HRM swallow sequence (Figure 1a-f).

A possible drawback to the chosen approach is its inaccuracy regarding the esophagus's highly curved/bent shapes. The perpendicular on which the generated cross-section is based is difficult to calculate for extreme bends. Possibly, further optimizing the chosen section sizes to handle curves during center path calculation could generate a more reliable reconstruction for such cases. Further, the calculated pixel-to-cm (px-cm) ratio that is applied for TBE and endoscopic images alike is approximated as follows. The Euclidean distance (in pixels) is calculated between the first and second sensor points along the approximated center path. With the given length between the two sensor points on the catheter in cm and that distance approximated in pixels, the px-cm is equal to the division of the latter with the first.

The final software allows the interactive and dynamic 3-D reconstruction of the TE and the LES and displays the lengths of the TE and the LES. Finally, the calculation of numerical value indices has been integrated. These indices are calculated independently for the TE and the LES. They are derived from volumetric data and high-resolution manometry pressure values extracted from the reconstructed 3-D images along the HRM sequence.

References:

14. Cortesi D. PyInstaller [Software]. 2022.
15. PyQt5 [Software]. Riverbank Computing Limited, 2022. 2022.
16. Plotly [Software]. Plotly Inc. . 2022.
17. Dash [Software]. Plotly Inc. 2022.
18. OpenCV-Python [Software]. OpenCV team. 2022.