EG UK Computer Graphics & Visual Computing (2022)
M. Turner and P. Vangorp (Editors)

Interactive Visualisation of the Food Content of a Human Stomach in MRI

C. Spann[1], S. Al-Maliki[1,2], F. Boué[3], É. Lutton[3] and F. P. Vidal[†1]

[1] Bangor University, UK
[2] Basrh University, Iraq
[3] INRAE, UMR MIA Paris 518 and LLB-CEA, UMR 12, Saclay University, France

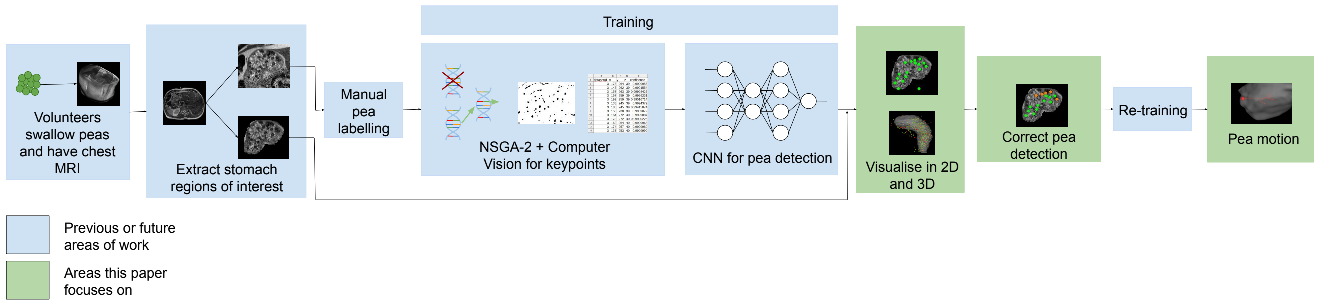

Volunteers swallow peas and have chest MRI
Extract stomach regions of interest
Manual pea labelling
Training
NSGA-2 + Computer Vision for keypoints
CNN for pea detection
Visualise in 2D and 3D
Correct pea detection
Re-training
Pea motion
Previous or future areas of work
Areas this paper focuses on

Figure 1: *Whole project pipeline.*

Abstract

Most medical imaging studies into human digestion focus on the organs themselves and neglect the content under digestion. Instead, analysing food inside digestive organs and any subsequent motion can provide valuable information about the digestive tract. This study is part of a larger project, with previous work done to automatically detect peas in a human stomach from MRI scans but it produced too many false positives. Our study therefore aims to accurately visualise peas in a human stomach whilst also providing facilities to correct the mistakes made by the previous pea detection. Our solution is a visualisation and correction tool split into 2D and 3D visualisation areas. The 2D areas show three sequential stomach slices with detected peas as green circles and allows the user to correct the pea detection. Peas can be added, removed or marked as unsure. The 3D area shows a Marching Cubes rendering of the stomach with spherical glyphs as the peas. Due to the way the data was acquired, some pea motion was also visualised. Aside from difficulties interpreting the data due to acquisition artefacts, our tool was found to be very easy to use, with some minor improvement suggestions for interacting with the images. Overall, the software achieved its aims of visualising the peas and stomach whilst also providing methods to correct the pea data. Future work will look into improving the pea detection and more work into following the pea motion.

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Machine learning;*** • ***Human-centered computing*** $\rightarrow$ ***Empirical studies in visualization;*** • ***Applied computing*** $\rightarrow$ ***Systems biology;***

1. Introduction

This work consists of visualising, analysing and correcting the results of automatically detected stomach contents in magnetic resonance imaging (MRI) (see Figure 1). In the use case we consider in this paper, the aim is to follow some specific food. This is a multi-disciplinary collaboration between Bangor University and the Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAE). It involves physicists, agro-food specialists and computer scientists and is a part of long

[†] Secretary UK Chapter of the Eurographics Association

DOI: 10.2312/cgvc.20221171

term research focused on the understanding of the influence of food structure on digestion.

The usual focus of gastrointestinal tract (GIT) imaging techniques is on the organs [MG13], while in this project we aim at following food digestion. Gastric emptying was observed for ingested frozen garden peas which were chosen because they keep their shape in early gastric stages. Therefore, peas are good markers for tracing internal movements in the stomach. New acquisition protocols were thus developed, however adapted image processing are also needed. This is because in this acquisition protocol, the spatial exploration is performed in an order that reduces measurement interference between slices. As a consequence, spatial proximity does not correspond to temporal proximity. This means that the slices are not scanned in sequential order. Furthermore, a high frequency of image artefacts (for example, motion artefacts) makes automated detection and visualisation difficult.

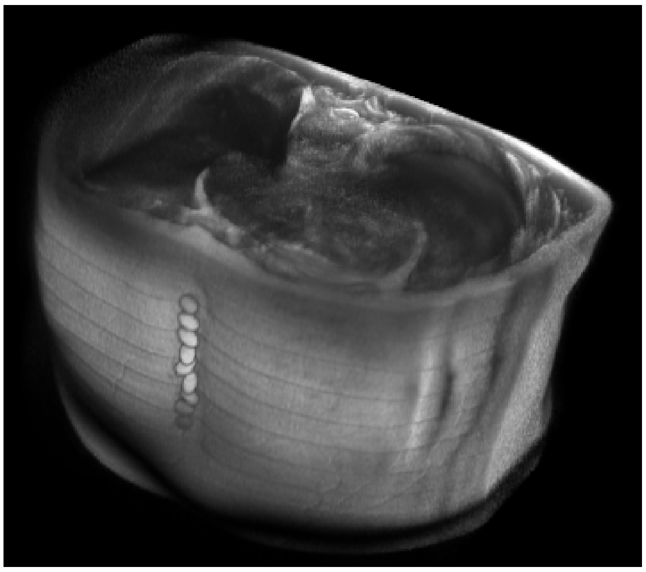
The aims of this paper therefore are: i) to visualise the stomach, ii) to visualise the detected peas, and iii) to provide functionality to correct any erroneously detected peas.

The rest of this paper is structured as follows. Section 2 describes common computer graphics processing methods as well as challenges with 2D and 3D visualisations. Section 3 explains how the MRI data was acquired, the pipeline leading from data to visualisation, as well as visualisation design. Section 4 shows the feedback received from our collaborators. Finally, Section 5 reviews if the visualisation achieved its aims and makes suggestions for future work.

2. Related Work

Common 3D visualisation methods for voxel data include Volume Ray Casting and Marching Cubes. Volume Ray Casting involves firing "rays" through the data and compositing values found along the ray . It often produces very high quality results (see Figure 2a) but incurs significant computational costs. Whilst Volume Ray Casting is a volume rendering algorithm, Marching Cubes on the other hand can be used to generate isosurfaces. The general premise is to divide the voxel space into cubes, each cube's vertices are made up of 8 voxels. A surface triangle is then generated from a set of 15 predefined configurations by determining which of the voxels intersect the cube. It then uses normalised gradients to perform shading. Marching Cubes can generate very high quality surfaces suitable for medical imaging tasks [LC87].

Segmentation is often applied to medical images to highlight useful or interesting structures within the image. One common method is using a convolutional neural network (CNN) for deep learning. CNNs often require large amounts of training and testing data to function effectively. Furthermore, there is inherent "uncertainty" in the resulting predictions made by the CNN as each class is assigned a "confidence" that it will be the correct label. Visualising uncertainty can help improve the decision making process done by clinicians. Visualisations that do not convey uncertainty are often not trusted by clinicians, or they may lead to poorer judgement calls [GSW*20]. One way that uncertain


(a) *Volume Ray Casting*

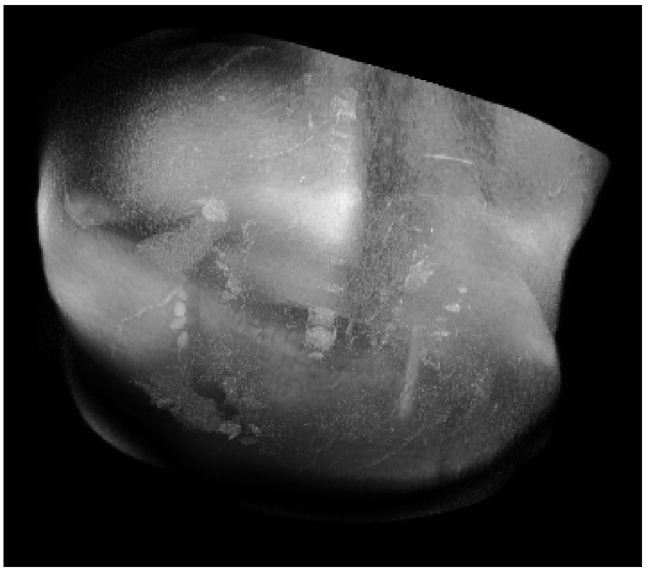

(b) *MIP*

Figure 2: *Maximum Intensity Projection (MIP) and Volume Ray Casting on Dataset 3. Note poorer depth perception with MIP compared to Volume Ray Casting*

be visualised is through different colours representing the uncertain regions.

Images may suffer from noise produced during acquisition which can lead to poorer quality visualisations. Furthermore, noisy images can also be more difficult to produce accurate segmentations. Therefore it is important to utilise suitable noise reducing methods, ideally without removing useful information from the image. One such method is anisotropic diffusion, which preserves edge information [CTCL10].

It can be difficult to decide between visualising data in 2D or 3D [DRST15]. Clinicians commonly utilise a 2D slice view when analysing image data, and cycle through each individual slice to find the interesting structures [LSBP18]. 2D methods can be very precise [DRST15], however this can be time-consuming and hinder full analysis of the data [LSBP18]. 3D visualisations can provide a wider overview and greater context for the data [LSBP18, DRST15], however they may suffer from occlusion, where interesting objects may be hidden behind other structures. Furthermore, some visualisations may suffer from depth perception and context preservation issues. One example is MIP which suffers from both poor context preservation and occlusion problems [YLRW10] (see Figure 2b). One solution is to apply rotation or allow user interaction to improve depth perception. Combining both 2D and 3D visualisations may perform better and provide the user with a better quality view than either alone [DRST15].

Finally, some medical images capture the time dimension. Such images may be called 2D + time images, 3D + time images or 4D images. Capturing the time dimension is necessary for tracking the motion of objects, and is especially important in medical imaging since, for example, the food content of the stomach is not stationary. One common approach for visualising the motion of an object is to track its trajectory or flow [MLLW15]. For example, keep track of previous positions to build a path to show locations that the object has previously visited.

3. Methods

3.1. Acquisition

The data used in this study is part of a multi-scale measurement dataset about human digestion. It includes small scale *in vitro* measurements using large facilities (small-angle neutron scattering (SANS), small-angle X-ray scattering (SAXS), deep UV fluorescence imaging) [LTLF*17, FBT*18] and *in vivo* medical images. *In vivo* information corresponds to an MRI of the stomach and duodenum of healthy human volunteers. The result is a set of Digital Imaging and Communications in Medicine (DICOM) files, each file made up of a 2D image called a "slice" (see Figure 3).

It is important to note however that there is a limitation of MRI scanning. The MRI machine is essentially a very large magnet, which when activated aligns all the water molecules in the subject's body. When the magnetic field is switched off, the water molecules oscillate, producing radio waves that are picked up by the scanner. However, if all the water molecules are excited at once, the resulting image will have too much noise. Therefore, the slices are scanned in groups with large gaps in-between, giving the water molecules time to essentially "relax" before the next slices are scanned. For example, in Dataset 3 there are 90 slices, and the first group of slices were taken in the following order: 1, 10, 19, 28, 37, 46, 55, 64, 73, and 82. The second group of slices has this order: 2, 11, 20, 29, 38, 47, 56, 65, 74, 83 and so on. The slices in a particular group are all taken at the same time, but there is a time delay between each group. This means that, even though slices 1 and 2 are next to each other in the Cartesian space, there is a time delay between their acquisitions (see Figure 4). We therefore

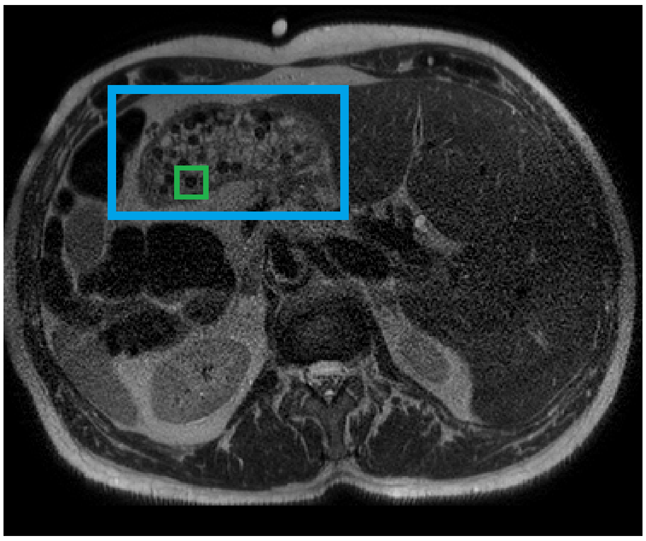


Figure 3: *Slice 53 from Dataset 3. A 2D cross-section of the stomach is visible here in the upper left highlighted by the blue rectangle. The swallowed peas are the dark circular regions inside. One example pea is highlighted with a green square*

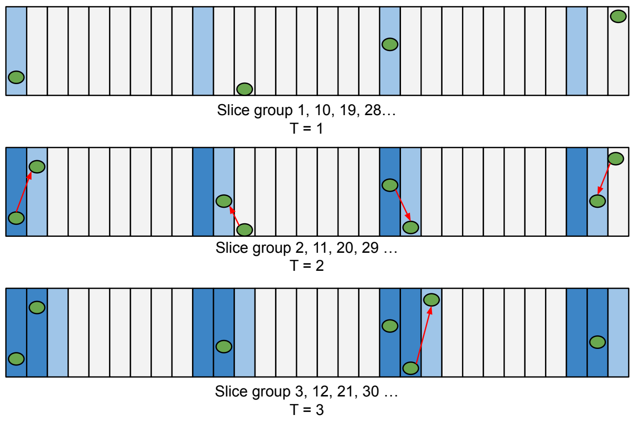

Slice group 1, 10, 19, 28…
T = 1
Slice group 2, 11, 20, 29 …
T = 2
Slice group 3, 12, 21, 30 …
T = 3

Figure 4: *Example showing the MRI acquisition order for three time steps. Each horizontal bar is a section of the MRI data at different timesteps. Light blue bars show current slices, and dark blue bars show previously scanned slices. Each red arrow shows the pea movement in-between slice scans. This example starts off with four real peas, which after the third time step appear as seven peas.*

consider these datasets to be 4D (i.e 3D + time) due to the time delay between slices. There are a total of 27 datasets that have been produced this way.

3.2. Pea Detection

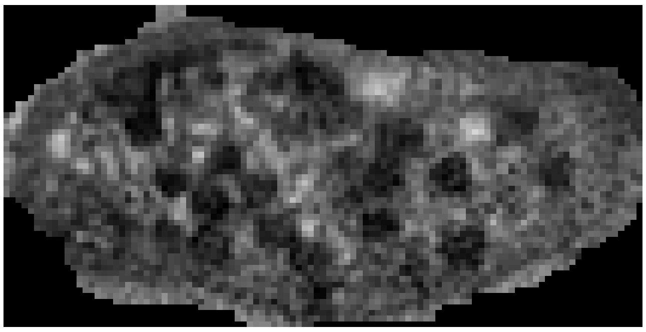
In each 2D slice, the stomach cross-section is manually segmented with the help of a region of interest (ROI) creation tool we developed. The user marks out the stomach area for which the tool creates two ROIs. The first only includes the relevant stomach pixels with any surrounding pixels set to zero. The second includes an automatically calculated 20 pixel border that includes the surrounding tissue pixels (see Figure 5). The first ROI is used primarily for 2D and 3D visualisation of the stomach, whilst the bordered ROI is used for the pea detection since some computer vision techniques require a non-zero border.

Each bordered stomach ROI is analysed. Object detection is performed as a multi-objective optimisation task, solved thanks to the Non-dominated Sorting Genetic Algorithm II (NSGA-2) algorithm and computer vision techniques are then applied to produce keypoints. NSGA-2 is a multi-objective evolutionary algorithm designed to have a lower computational complexity than previous multi-objective algorithms [DPAM02]. Keypoints are often identified in object detection tasks because they have local invariant features that are resistant to variations in the main object [RLJ09]. To produce these keypoints, NSGA-2 creates optimum (minimum and maximum) threshold values based on a set of features. For example, the skewness for each voxel for each image is one such feature. This feature will be given a min and max value. These min and max thresholds are then applied to each stomach image to create binary masks, i.e. regions in-between the min and max will be set to 1, otherwise 0. Areas of interest from the binary masks are then extracted. This is done by finding contours in each image using the OpenCV library. The center points of these contours will become the keypoints to be classified.

For classification, a CNN was chosen. CNNs are ideal for image recognition tasks and involve sliding a convolutional kernel across each input image to learn image features [XZS20]. Our CNN was built with the Google Tensorflow framework [ABC*16]. It was trained and tested on Datasets 1 to 10 using leave-one-out cross-validation using manually highlighted peas i.e. when testing Dataset 1, Datasets 2 to 10 are used for training.

Keypoints were created for Dataset 3 and Dataset 5 since they were deemed to have fewer image artefacts compared to the other datasets. When passed the keypoints, the CNN returns a 'confidence' between 0-1 for each keypoint. A confidence rating closer to 1 means that the neural network model believes the point to be a pea. It is important to note however that this does not necessarily mean the network is indeed correct about this prediction. We decided to label a keypoint as a pea if the confidence is greater or equal to 0.98 (i.e >= 98%). This automatic computation produced too many peas. For example, in Dataset 3, 504 peas were detected when there are only supposed to be 20-40 according to the experimental protocol.

There are several explanations for this. The first is straightforward: neural network errors. In some cases, the neural network has identified regions as peas which cannot be, for example, peas outside the boundary of the stomach. Supervised learning algorithms, in this case CNNs, are known to work best with a huge database of training data. As usual for such *in vitro* studies, the size of the cohort is small (here 10 volunteers), the development of adapted image recognition algorithms is thus difficult due to the availability of only small learning sets. Also, the execution of CNNs are controlled by hyperparameters. For example, the number of epochs i.e. the number of times the neural network will make a complete pass through the training data. Poor choice of hyperparameters can inhibit the potential of the model and lead to underfitting or overfitting. Furthermore, the CNN used for these experiments assigns a confidence rating to each pea. The current threshold is at 0.98 confidence. This parameter is very sensitive: increasing or decreasing this will change the number of peas marked as a true pea.


(a) *Rounded ROI*

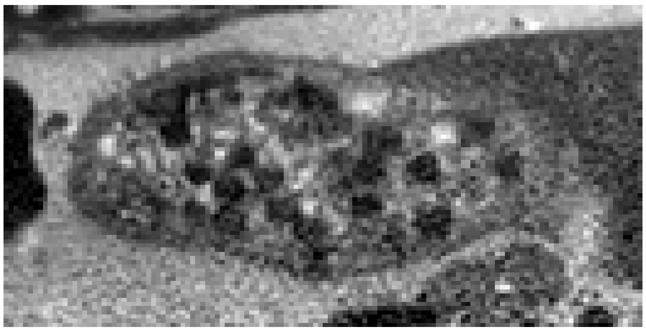

(b) *Bordered square ROI*

Figure 5: *Two ROIs from Dataset 3, Slice 51. Note Figure 5a has rough edges since it was manually drawn.*

There are also explanations relating to the nature of the MRI acquisition. The first is because of the time delay between slice acquisitions as mentioned previously (see Figure 4). This means that the contents of the stomach (i.e. the peas) can move in-between acquisitions and may be scanned multiple times. For Dataset 3, there are 9 time steps, which means that there are 9 opportunities for a single pea to be scanned. This means that a single pea could be duplicated up to 9 times in Dataset 3. Since a quantity of 20 to 40 peas were swallowed per volunteer, there could be between 180 to 360 peas in the MRI data. Secondly, depending on the slice thickness (i.e. z-spacing), one pea could straddle multiple slices at the same time. For example, in Dataset 3 the peas are approximately 5mm in diameter, and the slice thickness is 2mm. This means it is possible for one pea to appear in 3 slices simultaneously.

To compensate for these inevitable defects, additional information provided by experts can be used. This can be done through interactive visualisations. We thus propose a visualisation tool that allows an expert user to manually correct the neural network errors. The corrections can then be used to retrain the CNN.

3.3. Implementation Detail

Both the cropping tool and the visualisation tool were written in Python, using the libraries VTK and PyQt5. Python was chosen because our tool must be fully portable across different platforms, since some users have Apple Macs, some have Windows and some are using Linux. VTK is useful for aiding in the visualisation, for instance it provides an implementation of algorithms such as Marching Cubes and Anisotropic Diffusion [SML18]. PyQt5 is the python binding of Qt, which is useful as a windowing and widget toolkit.

3.4. Visualisation Design

The data has multiple dimensions which could be viewed simultaneously. It could be viewed in 2D for each slice but it could also be viewed as a 3D scene. Also, because of the acquisition method, there is a time dimension to the data too. Therefore, the tool is split into two main areas: a 2D + time area and a 3D + time area to provide as much visual information and context to the user as possible.

The 2D + time area consists of the manually segmented stomach slices, each slice having a semi-transparent 2D mask overlayed on top. Circular areas of each 2D mask are highlighted to show the position of the detected peas. The peas can have one of three labels: good pea, bad pea or unsure. Good peas are coloured green, unsure peas are orange, whilst bad peas are marked as transparent, i.e. they are removed from the visualisation. See Figure 6 for an example of a single slice at different stages of pea marking.

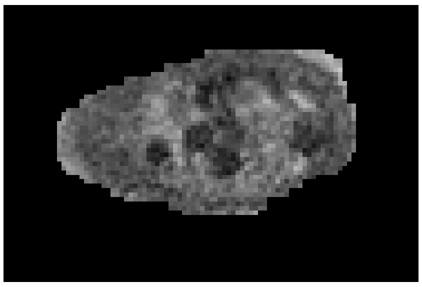
Three sequential slices are shown to see if there could be any peas straddling multiple slices. There are two rows of sequential slices. The top row shows the detected peas, and the bottom row shows the original slices so that they can be compared when correcting. Crosshairs are displayed that show the position of the cursor in the other slices, as well as the current position in 3D. These crosshairs help to provide context to the user for the 2D views, but also how this maps to the 3D model.

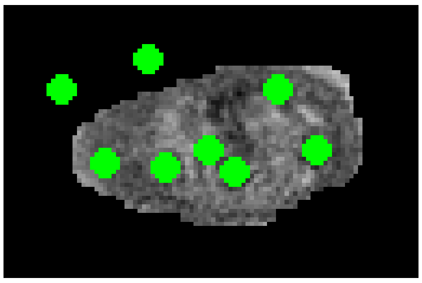
For the 3D view, the skin and stomach boundaries are rendered using Marching Cubes. This is because for these objects, we are only interested in the boundaries that they provide. The skin can be hidden or shown by the user. The stomach has some transparency applied to it so the peas can be visible, and is smoothed using anisotropic diffusion to reduce noise while preserving its outline. The 3D peas are rendered using VTK's glyph filter, which places a shape for each point in the dataset. In this case, the shape is a sphere and the points represent the pea locations. The peas are coloured based on the confidence rating. Peas with a higher confidence are green, peas with a lower confidence are red (see Figure 7). The scene also includes an axis helper in the form of a model of a person so the user knows the current orientation in relation to the human body if they interact with the model.

Since there may be duplicate peas that have been scanned multiple times, a form of clustering is employed. Since we know the order of scanning and the format of the time steps, we can cluster peas that are close to each other in terms of Euclidean distance. For example, for Dataset 3 we know there are 9 opportunities for a single pea to be scanned. So, for each pea, we need to pick 8 other peas, one from each of the next eight timesteps. Each pea is picked by calculating the euclidean distances from each pea in the next timestep to the previous pea. The pea with the shortest distance is chosen and assigned to the cluster. Each cluster group has a unique id, along with a cluster colour and time index. Each cluster of peas likely represents a single real pea. Each pea inside the cluster group represents a location of that real pea at a particular timestep (i.e. x, y, z, t). This means that an approximate motion of the pea in its cluster can be observed (see Figure 8).

3.5. User Interaction

As well as visualising the data, the tool must be able to correct the pea detection. Therefore, the tool allows the user to add, remove or mark peas as unsure in the 2D view. To do this, there are three "pea modes": removal mode, unsure mode and add mode. These modes


(a) *Rounded ROI*


(b) *Detected Peas*

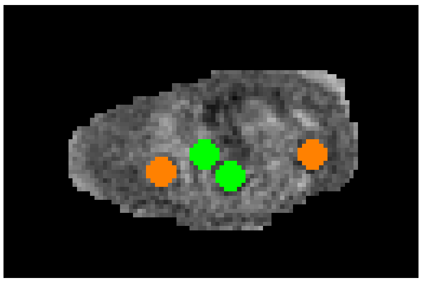

(c) *Corrected*

Figure 6: *Slice 39 from Dataset 3 at varying stages of our pipeline. Figure 6a is a plain rounded ROI. Figure 6b shows peas detected by the CNN. Figure 6c shows the following corrections: four peas removed and two are marked as unsure (orange). The remaining two green peas will be treated as "good" peas.*

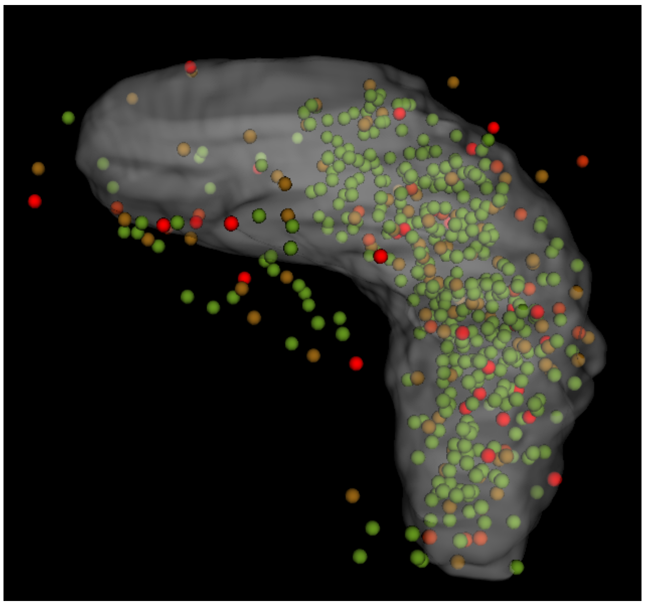
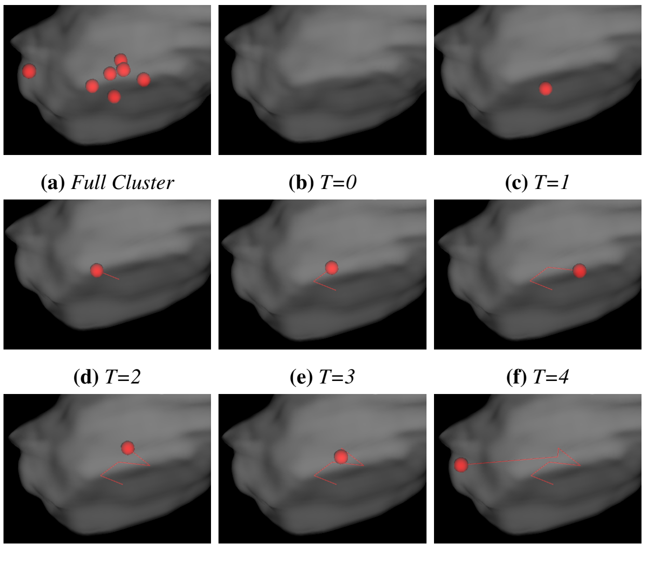


Figure 7: *3D stomach and peas. The peas are coloured in 3D depending on the confidence rating given by the CNN. Peas coloured green have a high confidence, whereas peas coloured red have a low confidence. The tan coloured peas will have a middle level of confidence.*



(a) *Full Cluster* **(b)** *T=0* **(c)** *T=1* **(d)** *T=2* **(e)** *T=3* **(f)** *T=4* **(g)** *T=5* **(h)** *T=6* **(i)** *T=7*

Figure 8: *A pea cluster and each timestep for that cluster. Figure 8a shows the full pea cluster, showing seven detected peas that could belong to the same true pea at different timesteps. Figures 8b to 8i show approximated motion of the pea through each timestep.*

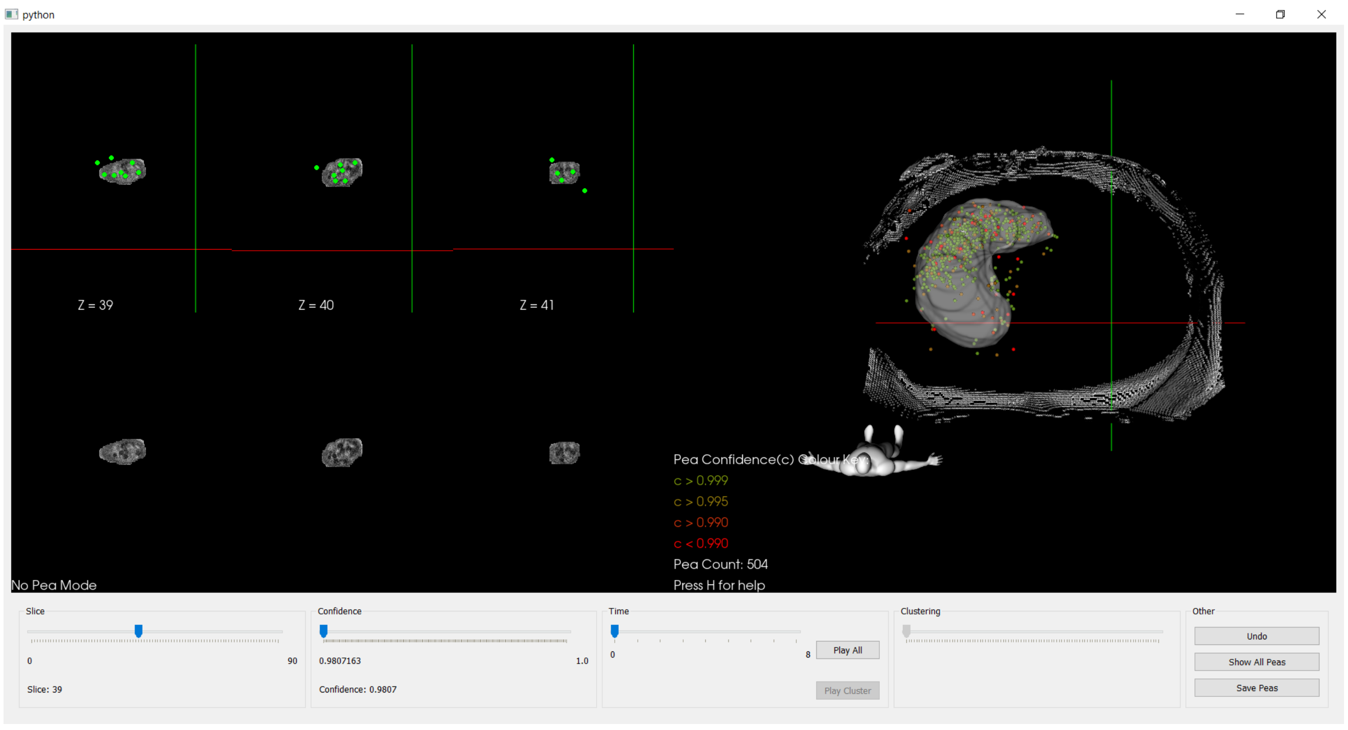
are turned on by pressing one of the number keys. The action is then performed by left mouse clicking. Any corrections can be saved to a comma-seperated values (CSV) file to be used in re-training the CNN or for future reference.

The visualisation tool has methods of aiding correction or analysis tasks with the peas. When the mouse hovers over a pea in the 2D view, some first order statistics are shown. The 2D slices can be navigated using a scroll bar and can be zoomed in and out using the mouse wheel to help analyse smaller regions. The 3D models can be rotated using the left mouse button and zoomed in using the mouse wheel. Since the data has a time dimension, both the 2D and 3D views can cycle through each time step. This allows the user to see motion in the 3D view, and how that relates to the 2D views. There is also a confidence slider that allows the user to set a confidence threshold. Any peas that have a confidence lower than the threshold are hidden (see Figure 9). The user can also undo, which is implemented by providing an abstract action object for each correction (i.e. one for adding a pea, one for removing a pea etc.) that provides a do, and undo method. All performed actions are held in a stack, and to undo the latest action calls its undo method. Undoing the corrections can be very useful for mistaken clicks or if the user changes their mind about a particular correction. The pea cluster groups can be toggled to be visualised which will show the cluster colours instead of the confidence colours. Furthermore, while playing back the motion of the peas, the path that each clustered pea travels on is also visualised in the form of connecting 3D lines. The vertices between each connecting line represents a pea position at a previous timestep (see Figure 8). Lastly, the user can press the 'h' key to display help for the controls of the program.

4. Results

After development, the visualisation and edit tool was shown to our research collaborators. They were given the task of correcting any erroneous peas using the tool and to provide feedback. They performed corrections on Dataset 3, which had 504 peas detected, and then Dataset 5 which had 363 peas detected. Both Dataset 3 and Dataset 5 have 46 stomach slices to analyse. After the collaborators corrections, for Dataset 3 there were 408 as good peas, 113 unsure, and 100 bad for a total of 621 peas. For Dataset 5 there were 234 good peas, 80 unsure and 120 bad for a total of 434 peas. Note that the overall totals for the corrected peas are greater than the original totals since some new peas were added as good or unsure peas.

The research collaborators found that each dataset was very difficult to interpret, due in particular to acquisition artefacts such as motion artefacts (see Figure 10). The full process initially took a few hours for a user to scan all slices of Dataset 3 and filter the pea positions proposed by the CNN but by the time they had finished Dataset 5 this was reduced to around half an hour. The controls for correcting the peas have been described as quite easy and fluid since one hand can be on the number keys for changing the pea mode and the other hand on the mouse or track pad for correcting. The simultaneous visualisation of three slices has been considered as very useful for evaluating confidence in pea classification, as well as the linked 2D and 3D views for locating peas positions. It was reported however that the zoom command was more difficult



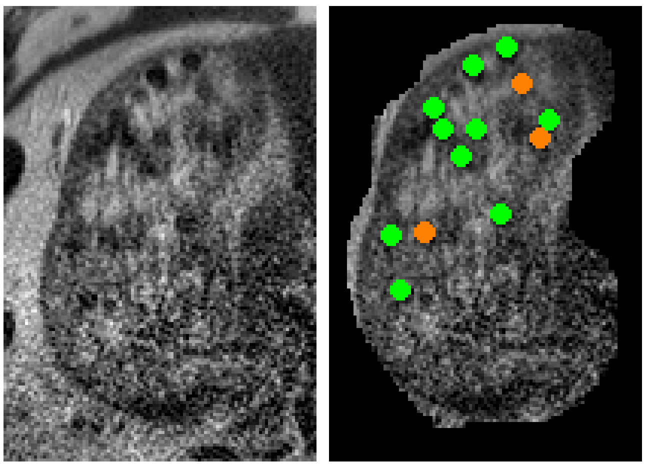
Pea Count: 504
Press H for help
Slice: 39
Confidence: 0.9807

Figure 9: *The visualisation and corrections tool. The left half is the 2D editing area. The right half shows the scene in 3D. There are tools at the bottom of the window for interacting with the visualisation.*

to manage and the greatest zoom size for each stomach slice is not always the same. This was due to the differently sized ROIs and the size of the window, as the constraint is to keep the entire slice in the view field. This meant it is more difficult to relate the sizes of the peas from one slice to the peas of another slice. If this constraint is relaxed then partial views become an option which may be more useful in some occasions. The pea clustering and their interpretation in terms of local possible trajectories has also been considered as another interesting tool (Figure 8), not yet widely exploited however in the current experiments. The aim would be to follow the movements of peas inside the stomach and their progressive ejection through the duodenum with time.

5. Conclusions and Future Work

The software achieved its aims of accurately visualising the results of the pea detection to a satisfactory standard according to our collaborators. Furthermore, the software has enabled correcting pea mistakes made by the original pea detection. These corrections will provide valuable insight into how to create a much more accurate pea detection. For instance, the new unsure category could be explored, as well as combining methods such as radiomics with deep learning. Radiomics involves analysing medical images by extracting a large amount of features, for example first order statistics such as min, max and mean values. Currently, the pea clustering only takes into account the time-step and euclidean distance. However, there is a range of other possible ways to cluster, for example we could cluster based on similar pixel intensities,



(a) *Raw ROI* **(b)** *Corrected*

Figure 10: *Slice 67 from Dataset 3 that demonstrates the difficulty in manual classification. Many of the potential pea points in Figure 10a appear noisy and could be classified as other stomach content. Furthermore, the stomach boundary itself is unclear towards the bottom right. Figure 10b shows the same slice but with some corrected peas. Note there are three peas marked as unsure (orange).*

or by using some radiomic features. An alternative to visualising movement in each pea cluster is to instead interpolate a single pea point for each cluster. Another possible future project would be to create a reverse algorithm. In other words, we model a stomach with 20-40 peas in it and model the motion as we simulate slice scanning. However, this would require more prior information to accurately model pea movement. Another idea would be to find an automatic way of tackling the problem of peas straddling multiple 2D slices. This could take into account the pea size and pixel spacing to automatically detect pea "fragments" that could all belong to a single pea. This will be difficult however since we would need to find a way of taking into account multiple images for each individual classification, made more difficult by the acquisition protocol. Finally, since the tool performs lots of intensive image operations and is written in Python, the program can be fairly slow especially for rendering time sequences. Therefore, the program could be sped up by analysing the code for bottlenecks and re-writing parts in C++.

Ethical approval

The study protocol has been approved by the Ethics Committee Lyon Sud-Est IV, and it has been registered in the Clinical Trial Registry (clinicaltrials.gov; NCT03265392) (see https://clinicaltrials.gov/ct2/show/NCT03265392). All volunteers gave their written informed consent after being provided with oral and written information about the aims and protocol of the study.

Acknowledgements

This work was partially supported by a STSM Grant from COST Action CA15118 (FoodMC) and the MRI data used in this study were collected at CEA-SHFJ with the support of IR4M CNRS/Orsay University (Xavier Maître and Luc Darrasse) in the framework of the IDI/Paris Saclay PhD of Daniela Freitas.

Acronyms

CNN convolutional neural network.
CSV comma-seperated values.
DICOM Digital Imaging and Communications in Medicine.
GIT gastrointestinal tract.
INRAE Institut national de recherche pour l'agriculture, l'alimentation et l'environnement.
MIP Maximum Intensity Projection.
MRI magnetic resonance imaging.
NSGA-2 Non-dominated Sorting Genetic Algorithm II.
ROI region of interest.
SANS small-angle neutron scattering.
SAXS small-angle X-ray scattering.

References

[ABC*16] Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M., Kudlur M., Levenberg J., Monga R., Moore S., Murray D. G., Steiner B., Tucker P., Vasudevan V., Warden P., Wicke M., Yu Y., Zheng X.: TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)* (2016), pp. 265–283. 4

[CTCL10] Chao S. M., Tsai D. M., Chiu W. Y., Li W. C.: Anisotropic diffusion-based detail-preserving smoothing for image restoration. In *Proceedings - International Conference on Image Processing, ICIP* (2010), pp. 4145–4148. doi:10.1109/ICIP.2010.5653571. 2

[DPAM02] Deb K., Pratap A., Agarwal S., Meyarivan T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6* (4 2002), 182–197. doi:10.1109/4235.996017. 4

[DRST15] Dubel S., Rohlig M., Schumann H., Trapp M.: 2D and 3D presentation of spatial data: A systematic review. In *2014 IEEE VIS International Workshop on 3DVis, 3DVis 2014* (7 2015), Institute of Electrical and Electronics Engineers Inc., pp. 11–18. doi:10.1109/3DVis.2014.7160094. 3

[FBT*18] Floury J., Bianchi T., Thévenot J., Dupont D., Jamme F., Lutton E., Panouillé M., Boué F., Feunteun S. L.: Exploring the breakdown of dairy protein gels during in vitro gastric digestion using time-lapse synchrotron deep-UV fluorescence microscopy. *Food Chemistry 239* (2018), 898 – 910. doi:10.1016/j.foodchem.2017.07.023. 3

[GSW*20] Gillmann C., Saur D., Wischgoll T., Hoffmann T., Hagen H., Maciejewski R., Scheuermann G.: Uncertainty-aware brain lesion visualization. *Eurographics Workshop on Visual Computing for Biology and Medicine* (2020), 97–101. URL: https://diglib.eg.org, doi:10.2312/vcbm.20201176. 2

[LC87] Lorensen W. E., Cline H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics 21* (1987), 163–169. doi:10.1145/37401.37422. 2

[LSBP18] Lawonn K., Smit N. N., Bühler K., Preim B.: A survey on multimodal medical data visualization. *Computer Graphics Forum 37* (2018), 413–438. doi:10.1111/cgf.13306. 3

[LTLF*17] Lutton E., Thevenot J., Le Feunteun S., Floury J., Panouille M., Dupont D., Roblin P., Boue F.: Evolution of food gel structures during simulated gastro-intestinal digestion using small angle scattering at SOLEIL synchrotron. In *5th International Conference on Food Digestion* (Apr. 2017). Poster. URL: https://hal.archives-ouvertes.fr/hal-01581553. 3

[MG13] Masselli G., Gualdi G.: CT and MR enterography in evaluating small bowel diseases: when to use which modality? *Abdominal Imaging 38*, 2 (Apr 2013), 249–259. doi:10.1007/s00261-012-9961-8. 2

[MLLW15] Mi Y., Liu L., Lin W., Wang W.: Extracting recurrent motion flows from crowded scene videos: A coherent motion-based approach. In *Proceedings - 2015 IEEE International Conference on Multimedia Big Data, BigMM 2015* (7 2015), pp. 371–376. doi:10.1109/BigMM.2015.20. 3

[RLJ09] Rudinac M., Lenseigne B., Jonker P.: Keypoint extraction and selection for object recognition. In *IAPR Conference on Machine Vision Applications* (2009), pp. 191–194. URL: https://www.researchgate.net/publication/228458561. 4

[SML18] Schroeder W., Martin K., Lorensen B.: *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*, 4.1 ed. Kitware, 2018. Available from https://gitlab.kitware.com/vtk/textbook. 5

[XZS20] Xin R., Zhang J., Shao Y.: Complex network classification with convolutional neural network. *TSINGHUA Science and Technology 25* (2020), 447–457. doi:10.26599/TST.2019.9010055. 4

[YLRW10] Yang L., Ling F., Rao N. N., Wang Z. K.: A new algorithm of maximum intensity projection based on context-preserving illustrative volume rendering model. In *Proceedings - 2010 3rd International Congress on Image and Signal Processing, CISP 2010* (2010), vol. 5, pp. 2381–2386. doi:10.1109/CISP.2010.5648245. 3