C-0153 HoloLens for medical imaging using post-mortem fetal micro-CT data

ControlNr.: #0959

Keywords: Anatomy, eHealth, Computer applications, CT, Experimental, Computer Applications-3D, Computer Applications-Virtual imaging, Foetus

Status: SUBMITTED

Type: Scientific Exhibit

Authors: C. M. Smith[1], S. C. Shelmerdine[2], O. J. Arthurs[2], I. Baskaran[2], N. J. Sebire[2], D. Mohamedally[2]; [1]London, -/UK, [2]London/UK

Previous Next

Section: Aims and objectives

1. Aims & Objectives

- Demonstrate applicability of HoloLens technology for viewing post-mortem fetal micro-CT imaging data.
- Develop a pipeline focusing on the required editing of 3D segmentations for rendering in virtual reality (VR), file format and storage needs for medical holographic applications and the necessary functionality of a holographic application interface.

2 Background

2.1 Why medical imaging in virtual reality?

The field of medical imaging stands to benefit significantly from advances in virtual reality [1]. Currently, most medical scans are viewed in two dimensions (2D): a series of image slices that a trained radiologist scrolls through to mentally stitch together a picture inside the three-dimensional (3D) body. It precludes direct interaction with the 3D image and hampers the perception of depth and spatial relationships. Being able to transform 2D image slices into patient-specific 3D holographic models which can be interacted with in their anatomical context and viewed from different angles can significantly aid pre-surgical planning, professional education and diagnostic explanation to patients [2].

The focus is not on diagnostic applications, but rather on the added contextual information that can be gained from rendering of organ surfaces as they appear in the body, and the powerful visual representations that can be achieved through VR. With the ongoing advances in VR technology, it is also likely that unidentified uses will be elucidated for healthcare over the coming years. For example, Microsoft's recent integration of their machine learning portal with the HoloLens technology [3] could provide a powerful platform for data analysis and machine learning from 3D health data.

2.2 Barriers to overcome

However, there are several barriers to the widespread adoption of VR for medical imaging which stem partly from the lack of competence in deployment of VR solutions among healthcare professionals and the relative infancy of the technology. There are also barriers that need to be overcome in terms of adaptation of the technology from its gaming use. Most VR games are pre-built solutions that have all their 3D assets statically written into the game. In contrast, a VR medical imaging application would ideally be able to pull assets (3D models) from a patient repository dynamically, as re-building the application to view each patient case is inhibitive. In addition, the 3D models produced from medical scans generally tend to be too large and complex for the current computational power of VR headsets. And, without the addition of materials to the models, they would not appear as realistic organs.

Other important considerations are file format and storage. 3D models can be stored in a range of file formats and it is necessary to convert the scan segmentations into the correct format for VR headsets. The models also need to be stored in way that can be connected to the larger medical data infrastructure – so as to make this data (securely) available for wider data analysis and machine learning goals. For a holographic imaging application, data will not just consist of 3D meshes, but also additional annotations about the case (e.g. images, text, audio). Therefore, research is needed into new file formats that can incorporate extensible holographic data in a temporal manner.

These barriers call for the development of a pipeline that will streamline and abstract the current complexities of a medical holographic application and to present solutions in an intuitive way for use by healthcare professionals.

Section: Methods and materials

1. Fetal micro-CT

For this project, the imaging data was provided by Great Ormond Street Hospital (GOSH) in the form of a microfocus computed tomography (micro-CT) scan of a 13 week fetal cadaver.

Micro-CT an imaging technique that allows 3D image reconstruction of tiny objects at micron resolution. At GOSH, micro-CT is being developed as a novel technique for fetal post mortem in early stage miscarriages.

This scan provides an interesting angle for demonstrating a holographic application – it provides challenges in terms of segmentation and shows the visual power of VR.

2. Segmenting organs from fetal micro-CT

Literature reviews were carried out to compare different segmentation software. Slicer-3D was chosen due to being free, open-source and cross platform. Slicer-3D offers a wider range of segmentation tools and scripting capabilities through a python console and segmentation module [4].

After trials with Slicer-3D's 'grow from seeds' and 'fast –grow cut' techniques, the final internal organ models were produced using Slicer's 'fill between slices' tool [5], which extrapolates between user markings. This produced the finest control with minimal spread outside the boundaries (Fig. 1). The skin model was produced using Slicer's thresholding tool [6]. This is an example of automated segmentation (possible due to the high contrast between skin and outside region). This can also be automated using a python script in Slicer's console [7]. All models were exported in .stl format.

3. Mesh processing application

An application to process the meshes ready for VR was developed as a multi-threaded JavaFX application that controls Blender 3D software in the background. JavaFX is a Java based graphical-user-interface software for delivering desktop applications and was chosen to avoid processing times being affection by internet connection speeds and increased security of data. Multi-threading in Java prevents the application from 'blocking' whilst

Sidebar

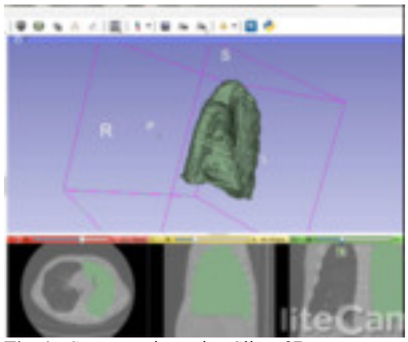

Fig. 1 : Segmentation using Slicer 3D

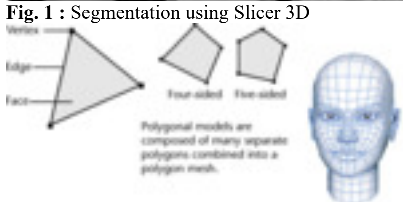

Fig. 2 : Representation of a 3D polygon surface mesh.

Software	Use	Reason
Slicer-3D	Segment CT scans	Python scripting, many segmentation tools available
JavaFX in eclipse	Mesh Processing App	High quality GUI tools, ability to execute shell commands and multi threading
Blender, Python	Mesh Processing script	High quality mesh decimation algorithm, many 3D modelling tools available, python scripting to automate in processing app.
Unity, C#	HoloLens app development	HoloLens development required to use Unity. Good debugging using game mode and console, intuitive interface
Visual studio	Building Unity game for Universal Windows platform	Required for UWP development – good linting for C sharp.
Azure Blob storage	Store patient case files to load at runtime	Most appropriate for large size single entity database management – Azure account provided
HoloLens Emulator	Testing for HoloLens OS in virtual machine	Useful to test app during development when headset not available

Fig. 3 : Summary of software used to develop the pipeline.

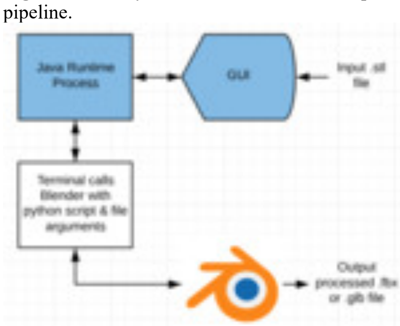

Fig. 4 : Architecture of the mesh processing app.

the mesh is being processed. Blender is a 3D modelling software chosen for the ability to run Python scripts 'headless' (i.e. in the background of the application), simplifying the process for users.

The main actions of the application are: import stl file, apply decimate modifier to simplify surface, smooth surface, add material based on organ type and export as glTF2 format – a lightweight 3D format for rendering in VR. Finally, the mesh processing application incorporated the ability to upload processed meshes into a patient case and store the case in a repository (Azure Blob storage).

Blender's inbuilt 'decimate' function was used to simplify the mesh surface. This algorithm is based on iterative vertex decimation: iteratively selecting a vertex for removal, removing all adjacent faces, and re-triangulates the resulting hole.

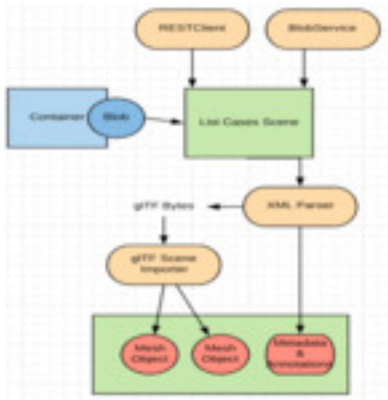

Fig. 5 : Architecture of the HoloLens app.

4. A new file type - Medical Holographic Interchange Format (MHIF)

This was developed as a blueprint for how to store medical imaging data as holographic cases. Based on XML (exensible markup language) schema it is extensible, temporal and stores vector (positional) data to link annotations to meshes.

The data is categorised into 3 main sections within the XML: metatdata, mesh data and annotation data. The mesh and annotation sections contain their own metadata elements and attributes that documents the temporal details such as time and author of edit. This also assists locating data when parsing the file. Annotations, for now, are of type image, audio or text but this can be extended in future iterations. The mesh, image and audio data are stored as base 64 string to reduce the file size.

5. HoloLens application

5.1 Application overview

The final part of the pipeline is concerned with displaying the processed 3D segmentations in virutal reality as a holographic patient case using Microsoft's HoloLens device. The aims were as follows:

1. To produce an interactive hologrpahic scene using organ segmentations from fetal micro-CT to demonstrate how 3D holographic representations can be displayed as a patient case in VR.

2. To research and implement a solution for dynamically loading patient case files including mesh objects at runtime from a holographic 'repository' – to simplify the process for use by healthcare professionals and overcome the dependency on 'building' 3D assets each time a user wants to view their meshes.

The application for HoloLens was written in as a Unity game, using C sharp and built into a Universal Windows Platform (UWP) app using Visual Studio.

5.2 Dynamic loading of 3D meshes

The solution researched and implemented here to achieve dynamic loading of the mehses was possible through a recent advancement in 3D file format: glTF2 is a new 3D format devised by the Khronos group in 2016 [8]. It is an application neutral runtime asset delivery format based on the JSON standard that compresses the size of 3D scenes and minimizes runtime processing and has been described as the 'JPEG of 3D file formats'.

Since its inception the group have provided several libraries for import/export of the file type. For this project, the Khronos group 'glTFlib' for Unity was used and adapted to achieve runtime loading of a case file (in xml format) from blob storage which is then parsed for mesh data (and metadata) and is displayed in the scene. In this way, the user can be inside the running HoloLens app, upload a new case file to storage and refresh to see their new

case without needing to exit the app or re-build. Not only does this circumvent the need for knowledge of Unity or UWP apps but significantly speeds up the pipeline from CT to HoloLens for 3D surface meshes.

Section: Results

1. Pipeline overview

An overview of the pipeline implemented can be seen in figure 6. It starts from the DICOM (CT) data and progresses through segmentation, processing, holographic case creation and uploading to HoloLens application dynamically from storage.

Whilst the solutions behind the mesh processing and HoloLens application provide automated and streamlined blueprints for dynamically loading holographic patient cases, it is clear that the main limitation overall is producing the initial segmentations from CT scans. The success and accuracy of the segmentation is key producing interesting and relevant holograms. There are several research teams investigating machine learning techniques for segmentation [9] and their work should be considered alongside any further development of holographic applications from segmented CT scans.

2. Mesh processing app; results & discussion

Overall, the mesh processing application implemented meets all the requirements laid out at the start of the project - it takes an unprocessed 3D mesh segmentation in .stl format and outputs a smaller, processed mesh in glTF format without the user requiring pre-exsisting knowledge of 3D modelling. The user interface for the application is shown in figure 7.

The mesh decimation algorithm produced very good results, strictly preserving the topology of the meshes even at high levels of decimation. The fetal skin surface was simplified at a ratio of 0.04 and maintained its appearance well.

Sidebar

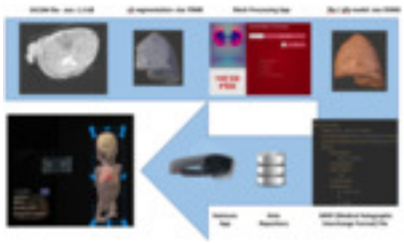

Fig. 6 : Medical holographic pi

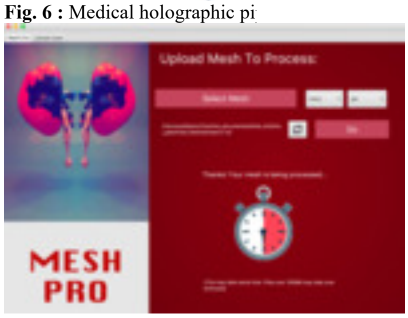

Fig. 7 : Mesh processing app us

Organ	Initial Face Count	Initial File Size	Final File Size	Processing Time (ms)
Skin	> 2,000,000	68 MB	4.4 MB	82662
Brain	700,000	32.2 MB	4.4 MB	21041
Lung	500,000	26.6 MB	4.3 MB	20241
Heart	200,000	9.9 MB	4.1 MB	8133

Fig. 8 : Comparison of mesh si: times.

However, it should be mentioned that the application is a balance between simplifying the process for users without knowledge of 3D modelling and providing the user more control over the processing of their models. If users wanted more control over the appearance of the final model (e.g. a more complex, specialised material) they can, however, use the Blender script and make modifications directly.

Fig. 9 : Medical Holographic Interchange Format (.MHIF) schema.

Future work:

Further iterations could look at improving the graphical user interface and providing more options for mesh processing outcomes based on user preference.

3. Medical Holographic File Format; results & discussion

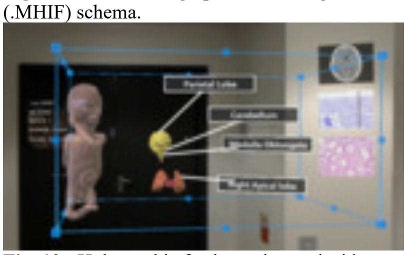

Fig. 10 : Holographic fetal case layout inside HoloLens.

The file format designed stores extensible, temporal data in a format that is easy to parse. It lays out a blueprint for what information may be useful when thinking about how to store holographic medical data in future and how to integrate this into the wider patient case. The mesh data, being compressed in base 64 assists in compression of the file size.

Future Work:

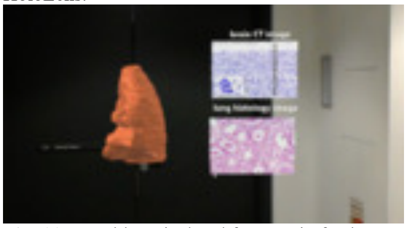

Fig. 11 : Fetal lung isolated from main fetal hologram.

Further iterations of the file type could involve looking into a binary format using the initial structure laid out here, for increased speed and efficiency.

4. HoloLens application; results & discussion

Fig. 12 : Fetal hologram with organs visible through skin layer and 'billboard' menu.

The HoloLens application was implemented successfully and achieved both of the 2 main goals: fetal holograms and dynamic loading. The layout of the fetal scene is shown in screenshots in figures 10-12. The metadata is displayed in a menu which has 'billboard' and 'tagalong' components so that it's always facing the user and slightly within the field of view. The meshes are central and can also be seen in an expanded state which is controlled through voice commands.

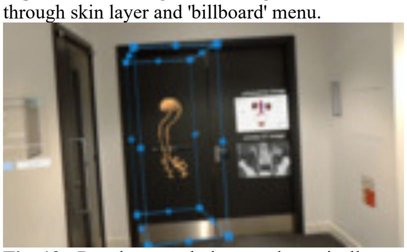

Fig. 13 : Renal system hologram dynamically loaded into running HoloLens application.

The organ meshes have several intuituve gesture and speech components that control them. A bounding box component helps to alert the user to when they are manipulating the meshes – they can be scaled and rotated. The most natural action for a user when looking at the objects is to reach out and try to select them. For this reason, all meshes are made 'hand draggable'.

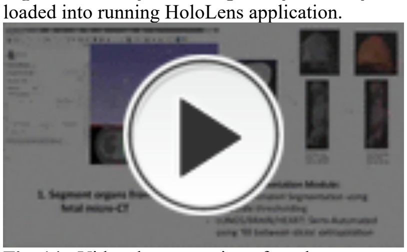

Fig. 14 : Video demonstration of mesh processing and HoloLens application.

The 'annotations' from the patient case (images and audio) and displayed in canvases just off to the right, and can be moved to inspect closer. Text annotations can be displayed as 'tool tips' – in future iterations these can be dynamically added at exact points on the meshes by providing vector coordinates as an attribute in the case file.

Future Work:

Further work is being conducted at the UCL computer science department focusing on recording annotations from inside the HoloLens app and saving these changes back into the .mhif files, creating a 'full cycle' dynamic, editable holographic system.

Section: Conclusion

This project has achieved several aspects of a functioning medical holographics pipeline; notably including a method to dynamically load 3D models at runtime from a specially designed medical holographic file type residing in a holographic repository. This has been demonstrated using data from fetal micro computed tomography (CT) scans and Microsoft's mixed reality HoloLens device, and can act as a blueprint for further development to ensure the readiness of hospitals to benefit from the rapid advancement of holographic technology.

Section: Personal information

C. M. Smith is a doctor with background in digital health. Research was completed as part of MSc computer science at University College London.

I. Baskaran is a 2nd year computer science student, University College London.

S. C. Shelmerdine is a Consultant in Paediatric Radiology and Paediatric Radiology Research Fellow at Great Ormond Street Hospital.

O. J. Arthurs is an Academic Consultant in Paediatric Radiology at Great Ormond Street Hospital and National Institure for Health Research (NIHR) Clinician Scientist.

N. J. Sebire is Professor of Paediatric and Developmental Pathology at ICH.UCL and Consultant Paediatric Pathologist at GOSH. He is the GOSH Chief Research Information Officer (CRIO).

D. Mohamedally is Principal Teaching Fellow for Applied Software Engineering and Industry Projects Director for the UCL Industry Exhange Network.

Section: References

[1] R. Oakes, "How Virtual Reality and 3D are Revolutionizing Medical Imaging", The Doctor Weighs In, May 4, 2017. [Online] Available: https://thedoctorweighsin.com/how-virtualreality- and-3d-are-revolutionizing-medical-imaging/ [Accessed Jun 2018]

[2] A.Chebanova, "Making Use of Virtual Reality in Healthcare", Steel Kiwi. [Online], Available: https://steelkiwi.com/blog/making-use-of-virtual-reality-in-healthcare/ [Accessed Jun 2018]

[3] Microsoft, "MR and Azure 307: Machine learning", Microsoft Windows Dev Centre, Apr 7th, 2018. [Online], Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mr-azure-307. [Accessed June 2018].

[4] 3D Slicer, “Docmentation/Nightly/Modules/Segmentation”, 3D Slicer. [Online], Available:https://www.slicer.org/wiki/Documentation/Nightly/Modules/Segmentations. [Accessed July 2018].

[5] PerkLab Research, “Quick Manual Segmentation with Contour Interpolation”, YouTube. [Online], Available: https://www.youtube.com/watch?v=u93kI1MG6Ic. [Accessed June 2018].

[6] G. Sampogna, “3D Slicer: Threshold Effect and Robust Statistics Segmenter”, YouTube. [Online], Available: https://www.youtube.com/watch?v=-IZV-rG9C7Q. [Accessed June 2018].

[7] A. Lasso, “ExtractSkin.py”, GitHub. [Online], Available: https://gist.github.com/lassoan/1673b25d8e7913cbc245b4f09ed853f9. [Accessed July 2018].

[8] Khronos Group, “Gltf Overview, Runtime 3D Asset Delivery”, Khronosgroup.org. [Online], Available: https://www.khronos.org/gltf/. [Accessed June 2018].

[9] G.Wang, M.Zuluaga, W.Li, R.Pratt, P. Patel, M.Aertsen, T.Doel, A.David, J.Deprest, S.Ourselin, T.Vercauteren, “DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation”, IEEE Transactions on Pattern Analysis.