Fully Automated 3D Colon Segmentation and Volume Rendering in Virtual Reality

Wanze Xie\textsuperscript{1} *, Xiaofan Lin\textsuperscript{2} *, Trevor Hedstrom\textsuperscript{2} *, Zifeng Li\textsuperscript{2} *, Larry Smarr\textsuperscript{1,2}, and Jurgen P. Schulze\textsuperscript{1,2}

\textsuperscript{1} California Institute for Telecommunications and Information Technology, UC San Diego, California, USA
\textsuperscript{2} Department of Computer Science and Engineering, UC San Diego, California, USA

Abstract. Deep learning algorithms have provided efficient and effective approaches for biomedical image segmentation. However, it is difficult for an untrained user to mentally or visually restore the 3D geometry of the segmentation result from MRI image stacks. Existing MRI image reconstruction approaches require human intervention, and the learning curve makes clinical adoption even harder.

By incorporating U-Net and volumetric rendering techniques, we present an automatic and intuitive application to assist surgeons to visualize and identify particular areas in MRI scan. Our system can accurately segment the colon section and create a manipulable 3D volume rendered in virtual reality environment with mask alpha blended. We hope that our system can support doctors to conduct biomedical imaging analysis and surgical planning in real clinical scenarios.

1 Introduction

Recent research \cite{1, 2} has demonstrated that deep learning approaches, notably Convolutional Neural Networks (CNNs), have become increasingly effective and popular for assisting medical imaging processing tasks, such as detection, segmentation, and classification. Research in the medical field has shown evolving interest in using such technology to conduct disease analysis \cite{3}, surgical planning \cite{4} and doctor-patient communication \cite{5}. However, even though deep learning models can be applied to segment various medical images such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scan images \cite{1, 2}, it is still difficult to visualize the results. Notably, the segmentation of complex 3D structures is still typically presented and evaluated with 2D images.

Popular 3D applications such as 3D Slicer and OsiriX provide effective ways to display medical data as 3D structures. However, the rendering results are usually limited to 2D screens, and the complicated software often has a steep learning curve. Therefore, it is necessary to have an intuitive and user-friendly tool for doctors to quickly identify an organ as a 3D structure with clear boundaries. Recent advancement in Virtual Reality and Augmented Reality technology

* These four authors contributed equally
have enabled various software applications [6], but few actual products have been created to augment the doctors’ capability to more effectively inspect medical image stacks.

In this paper, we introduce an integrated automatic colon segmentation and visualization system for MRI image stacks, based on the U-Net Convolutional Network architecture, with volumetric rendering using DirectX12 and OpenVR. It can load a series of DICOM image files as input, perform fully automated segmentation and analysis, and highlight the predicted segmentation of the colon in the volume rendered visualization of the MRI images inside virtual reality using the HTC Vive head mounted display (HMD). Unlike traditional desktop screens, virtual reality technology has provided a more intuitive approach for doctors to locate the colon in the abdominal area in MR Enterography. We further evaluate the advantage of our system in multiple use cases for medical imaging analysis:

1. Help doctors more easily find large bowels and a related area inside the MRI image of patients identified with Crohn’s Disease.
2. Provide manipulable 3D visualization in a collaborative environment to help doctors conduct surgical planning.
3. Export an automatically generated 3D mesh for 3D printing and help doctors to use the printed model to explain their Crohn’s Disease to the patient, which facilitates doctor-patient communication. The source code can be found at (omitted for double-blind review)

2 Discussion of Relevant Work

2.1 Auto-Segmentation using Machine Learning

Auto-segmentation using machine learning requires per-pixel classification with high accuracy. The large variety of body structure across different patients and various types of medical imaging methods with different setting parameters, such as contrast and resolution, also make this task even more complicated.

However, the recent success of research for deep learning makes sophisticated models available. CNNs is one of the most researched deep learning algorithms in medical image analysis [8], on which the U-net model (a fully convoluted neural network) is based. This is because it can preserve spatial relationships when filtering input images. In radiology, spatial relationship is one of the most critical information images must provide.[7]

2.2 Crohns Disease and MRI Volumetric Visualization

Gcer et al.[11] establishes a qualitative index from magnetic resonance imaging (MRI) activity, which can directly link to Crohn’s disease activity. In their experiments, correlation was found between MRI activity index and endoscopic acute inflammation scores (eAIS), which could be used as a featuring score for Crohn

Volumetric rendering has been studied to a degree in medical contexts. It is highly practical to use this technique to visualize medical images in Virtual Reality [15], as it allows for full 3D rendering and depth perception. This gives a more intuitive understanding of the orientation and structures within medical scans.

3 Methodology

3.1 Data Collecting and Processing

Raw Data Collection We collected and compiled dedicated MRI data set from the medical center. A male patient (63yo) with Crohn’s Disease was included to perform MR Enterography as a routine clinical evaluation. Body MR studies were acquired at the medical center on different scanners with contrast-enhanced Spoiled 3D Gradient Echo sequences on a Siemens 1.5T MR scanner for the first exam on GE 1.5T MR scanners for multiple follow-up exams. The acquisition uses a coronal slice with a thickness of 1.5mm, the number of slices ranging from 48 to 144, and in-plane resolution around 0.78mm × 0.78mm. The patient’s size dictates the field of view with the craniocaudal coverage requested to include the rectum up to the colon, so the in-plane resolution varies. This dataset, with three different protocols and exams on three different scanners at different times (2012, 2016 and 2017, details in supplement), also represents the variability of real clinical applications and reflects if the proposed method is generalizable.

Creating the Training Dataset To establish supervised training dataset, manual segmentation masks were included together with the raw DICOM files. Colon areas were segmented manually using the free software application 3DSlicer which were converted into 2D array before feeding them into the neural network model, as shown in Fig. 2.

![Fig. 2: Sample Segmentation of the MRI Images](image-url)
3.2 Colon Segmentation

Model Training and Mask Prediction We implemented the standard U-Net model[1] and adapted the input and output resolution to fit our training data set. Due to the limited data sets available, we performed data augmentation on our training data sets by generating a large number of new training images based on our existing images. After trained the model with desired testing accuracy, we integrated it into our system and use it for mask prediction, which will be subsequently used for the volume rendering in virtual reality with alpha blending. A simple overview of the pipeline of our segmentation system is demonstrated below in Fig. 4.

![Training and Prediction Procedure](image)

Fig. 4: Overview of our colon segmentation system’s data pipeline.

Model Evaluation Since there is no existing baseline dataset for us to evaluate the result, we obtained two different set of manually segmented images from the same dataset to create an objective comparison between our AI-predicted segmentation ($P_1$) and manual segmentation ($G_1$ and $G_2$). In the following sections, we will be using $G_1$ for our model’s ground truth, and the evaluation on $G_2$ as our baseline performance with representative inter-reader variability.

Chosen Metrics: For this study, we chose Dice Coefficient, Global Consistency Error, Volumetric Similarity, Rand Index and Hausdorff Distance, from three different statistics groups organized by Tasha et al. to minimize the correlations between different metrics, as well as to give us a complete idea to understand our prediction’s performance from different points of view[9]. Additionally, we include Mean Surface Distance from Dice’s paper as an improved version of Hausdorff Distance to mitigate the influence of noise[10].

Results: As table 1 shows, the proposed method achieves similar segmentation accuracy well within the range of inter-reader variability. The unprocessed
prediction result from the model has already out-performed humans as it is quantitatively better than our second manual segmentation.

**Post-Processing (Volume Denoising)** In addition, we can further improve the performance and get better evaluation results by incorporating a de-noising and post-processing step. As Figure 6 shows, the mask predicted directly from the model contains noisy texels in the 3D model. To mitigate the disconnected noise artifacts, we converted our 3D refinement problem to a simple graph connectivity problem and implemented a graph algorithm to locate the most massive tree from in a forest such that we can preserve texels that are mapped to the nodes within the main tree and only the largest tree is preserved. The algorithm is split into several parts: Histogram Voting, Graph Construction, Tree Selection, Back Conversion and Boolean And Filtering.

Let’s define our raw prediction as $P \in \{0, 1\}^{256 \times 256 \times S}$ where $S$ indicates the stack size and our histogram matrix will be $H \in \mathbb{Z}^{32 \times 32 \times S}$

1. **Histogram Voting**: We will be counting the number of texels within a certain range of the original predicted volume:

   $I_p(x, y, z) = \begin{cases} 
   1, & \text{if } P(x, y, z) \geq 0.5 \\
   0, & \text{otherwise}
   \end{cases}$

   $H(x, y, z) = \sum_{(i,j,k) = (8x, 8y, z)}^{(8(x+1)-1, 8(y+1)-1, z)} I_p(i, j, k)$

2. **Graph Construction**: We construct a graph, $G = (V, E)$ on those coordinates as vertices whose histogram values are larger than $t$ and connect nearby coordinates within range $r$:

   $V = \{(i, j, k) : H(i, j, k) \geq t\}$, $E = \{(v_1, v_2) : |v_1 - v_2| \leq r, \forall v_1, v_2 \in V\}$

3. **Tree Selection**: We run a Depth First search to determine the largest connected component $G_l = (V_l, E_l)$

4. **Back Conversion**: We construct a filter $F_{256} \in \{0, 1\}^{256 \times 256 \times S}$ based on $G_l$. But first, we convert $G_l$ back to matrix form $F_{32} \in \{0, 1\}^{32 \times 32 \times S}$ And
upsample $F_{32}$ to $F_{256}$.

$$F_{32}(x, y, z) = \begin{cases} 1, & \text{if} (x, y, z) \in V_1 \\ 0, & \text{otherwise} \end{cases}$$

5. Boolean-And Operation: After performing this step, we will have our refined prediction result $P_2 = P \odot F_{256}$

The post-processing algorithm has shown promising results from the result of $P_2$ in Table 1 and Figure 6.

### 3.3 Volume Rendering in Virtual Reality

**Techniques** Upon loading, the 16-bit DICOM data is copied into the red channel of a 3D `R16G16_UNORM` texture. The volume shader is rendered into a cube with a radius of 0.5. The start and end points of integration on the cube are found analytically using a simple ray-cube intersection algorithm, done in object-space in the pixel shader. This gives two points of intersection, a start and an end, to be sampled between along the pixel ray. The end point is moved closer based on the current value in the depth buffer so that opaque objects can correctly show depth. Optionally, a ray-plane intersection can be done to push the starting point back according to an arbitrary slicing plane.

Before the main integration loop starts, the ray origin is transformed from object space to texture space, which is as simple as offsetting it by 0.5 since the object is a cube with a radius of 0.5. Integration along the pixel-ray is then done, summing the color along each point using front-to-back alpha blending. If more than 128 non-empty samples are taken, or if the sample accumulates an overall density above 0.98, the loop exits early.

**Color and Masking** To sample the masked density at a point in texture-space, the texture is sampled and the red and green channels are multiplied together. We can optionally perform thresholding at this point, to better see the structures in the volume. The final sample color is simply the density value. Optionally, the mask value can be used to show the mask in a different color and a higher density, to show where the mask sits inside the volume, shown in Figure 8.

**Lighting** To approximate the optical thickness of the volume between the sampling point and the light source using, Beer’s Law is adopted as described in
Just two small steps are summed towards the light source, using a simplified version of the sampling function that skips color computation for speed. This provides a rough, local approximation of lighting, enough to provide an extra sense of depth and orientation of surfaces. Figure 8 shows the lighting effect.

**Interaction** We offer many utilities to interact with the data. The volume can be grabbed and moved using the virtual reality system’s controllers. Other utilities include lighting (shown in Figure 8), thresholding, and manual editing of the mask with vibrational feedback proportional to the tissue density (Figure 10). These features help the user to intuitively view and manipulate the data as a physical object easily.

### 4 Discussion and Conclusion

In this paper, we introduce a fully automated system that supports automatic colon segmentation with manipulable volume rendering in virtual reality. Our evaluation metrics showed that the deep learning model we trained using U-Net has the potential to understand complex abdominal structures and to segment out colon area with moderately high accuracy. We also showed that this model also do not need a significant amount of training data. We also explored how histogram voting and connected components preserving algorithms as noise reduction methods can significantly improved the prediction results.
This paper presents the advantages of combining volumetric rendering and virtual reality to visualize segmentation results. Using this technique, we could load a series of DICOM image files as input, construct a 3D volume of the patient’s abdominal area, and automatically highlight the colon region in the context. We also discussed how various utilities could be added to the VR application to make it more powerful and efficient. We showed how combining deep learning and virtual reality technology can assist doctors to identify and examine the colon with Crohn’s disease.

Future work for our system has three different aspects. First, a more complex and larger set of MR Enterography data will be organized for training to make the model more robust. Secondly, we will speed-up the deep learning inference and rendering speed, which currently could take one minute on a computer with a single GPU. A further optimization will make the system potentially deployable to augmented reality devices. The last one is to implement a multi-user interface inside the virtual environment so that different doctors can remotely join a same session and conduct surgical planning together. We believe that our system has great potential to be a powerful tool for doctors and researchers to examine biomedical imaging data.

References


