O DESENVOLVIMENTO DE UM SIMULADOR DE COLONOSCOPIA

VICTOR KUNDE BERGMANN[1]; RAFAEL PICCIN TORCHELSEN[2]; ANDERSON MACIEL[3];JOÃO CARLOS BECKER[4]; LUCAS MORAIS[5];LEOMAR ROSA JR[6]

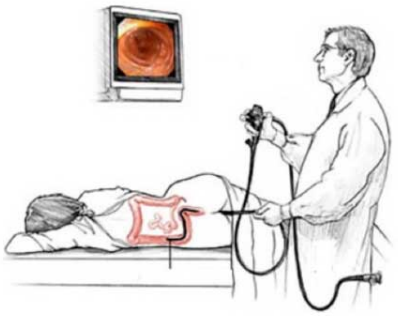
[1]Universidade Federal de Pelotas – vic.kunde@gmail.com
[2]Universidade Federal de Pelotas – rafael.Torchelsen@inf.ufpel.edu.br
[3]Universidade Federal do Rio Grande do Sul - andi.maciel@gmail.com
[4]Universidade Federal do Rio Grande do Sul -beckerjc95@gmail.com
[5]Universidade Federal de Pelotas – lzmorais@inf.ufpel.edu.br
[6]Universidade Federal de Pelotas – leomarjr@inf.ufpel.edu.br

1. INTRODUÇÃO

Devido a dificuldade para o aprendizado em cirurgias, visto que são procedimentos que podem trazer medo e desconforto aos pacientes, criou-se a necessidade do desenvolvimento de equipamentos capazes de simular tal processo, facilitando não apenas a visualização como também a prática, porém são poucas as cirurgias que podem ser simuladas, devido às limitações tecnológicas. A colonoscopia, representada na figura 1, se encaixa como uma das possíveis, visto que durante a sua execução é necessária uma tela para visualizar a ferramenta dentro do paciente.

O projeto tem como objetivo o desenvolvimento de um programa capaz de simular uma colonoscopia com confiabilidade e precisão com o uso dos softwares Unity (motor gráfico que será nosso ambiente trabalho) e PBD (framework que pode simular partículas, corpos rígidos, molas, entre outros, foi descrito por Matthias Müller em 2007), com o objetivo de auxiliar no aprendizado de novos médicos, facilitando a sua capacitação em tal procedimento.

Figura 1: imagem representativa de uma colonoscopia sendo realizada



2. METODOLOGIA

Para o desenvolvimento, decidimos usar o motor gráfico Unity, será o ambiente onde posicionamos nossos objetos e descreveremos seus comportamentos.

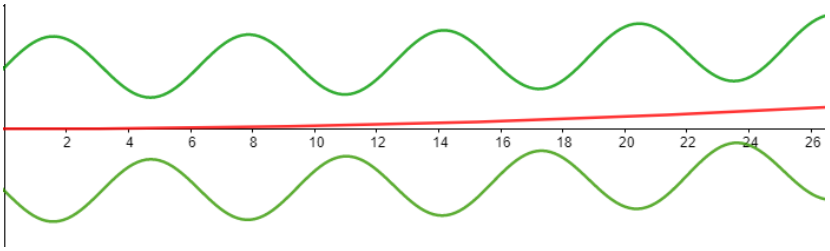
Infelizmente a Unity não apresenta um bom sistema de partículas, que é necessário para a implementação de objetos não rígidos. Então usaremos a biblioteca “PositionBasedDynamics” (PBD) feito por JAN BENDER(2017), esse código nos auxiliará a controlarmos o comportamento do intestino e do endoscópio com maior facilidade.

2.1 Intestino

Figura 2: imagem do modelo do intestino que será usado no simulador

Após termos o modelo inicial do intestino grosso (malha composta por vértices e triângulos que pode ser visto na figura 2), precisaremos dar movimento a ele (o nome do movimento característico do intestino grosso é “peristaltico”). Para isso, os vértices farão um movimento de senóide ao redor de uma linha auxiliar que indicará o centro do intestino em toda sua extensão como pode ser visto na figura 3.

Figura 3: imagem representativa da formação do movimento peristáltico



Porém se todos os vértices da malha estiverem se movimentando ao mesmo tempo, teremos um grande custo computacional, para evitarmos isso diminuiremos o nível de detalhe dos vértices distantes da ponta do endoscópio.

Figura 4: demonstração das partículas que se movimentaram

Na imagem vemos em azul onde estaria a ponta do endoscópio, e em amarelo as partículas que estão sendo simuladas no momento. O resto do intestino se encontra rígido.

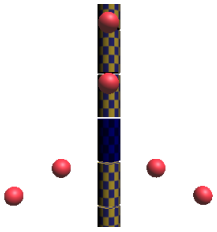
2.3 Endoscópio

Para simularmos o endoscópio usaremos um recurso da biblioteca PBD, que é descrito no artigo “Direct Position-Based Solver for Stiff Rods”(2018), inicialmente feito para simular cordas e molas, porém se encaixa perfeitamente neste uso. Com a câmera na ponta, o endoscópio é um conjunto de corpos rígidos, conectados por “juntas”, onde as curvas podem ser feitas.

Figura 5: imagem de parte do endoscópio fazendo curvatura

Chegamos na parte onde o usuário terá o controle, na operação de colonoscopia, o cirurgião tem controle sobre a curvatura da ponta do endoscópio (algo que a biblioteca PBD nos dá controle, e pode ser visto em uso na figura 5), a rotação no eixo do endoscópio e a entrada e saída do objeto. Para os dois últimos movimentos tivemos que usar molas entre um dos segmentos do endoscópio e partículas da qual o usuário terá o controle da posição. Fizemos dessa forma para que possamos calcular o retorno háptico e também para não causar um movimento irrealista na ferramenta.

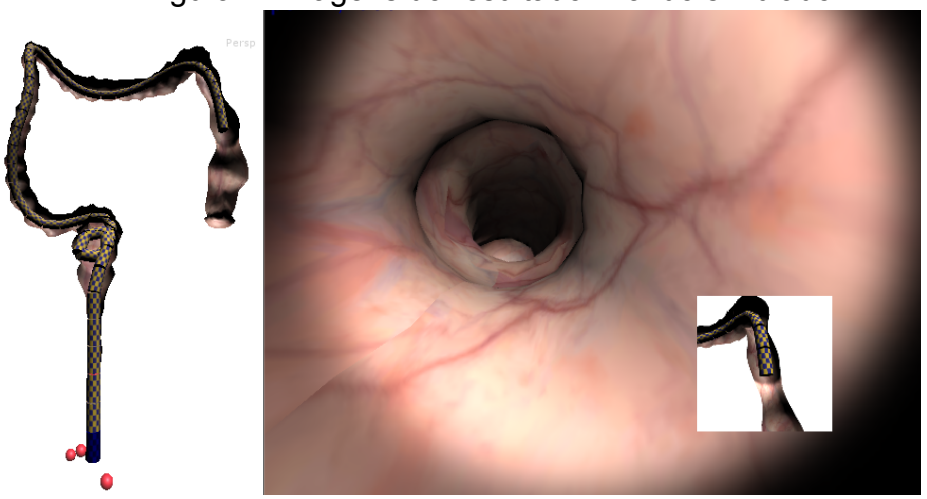
Figura 6: imagem do endoscópio e das partículas que puxaram o segmento em azul



3. RESULTADOS E DISCUSSÃO

Como podemos ver na figura 7, o endoscópio consegue chegar até o final do intestino, conseguimos usar todos os movimentos da ferramenta, além de calcular a força que é aplicada nas molas que movimentam o endoscópio, e esses valores podem ser usados para dar retorno háptico ao usuário.

Figura 7: imagens do resultado final do simulador



Além disso, formou-se um “nó” no início do intestino, algo que realmente acontece durante a operação, onde a simulação se mostra similar ao procedimento.

A malha executa o movimento peristáltico da maneira esperada, porém não temos o retorno da colisão, ou seja, quando o endoscópio colide com a malha do intestino, apenas o endoscópio reage à colisão. A implementação da solução desse problema será um dos próximos passos do nosso projeto.

4. CONCLUSÕES

O programa segue em constante desenvolvimento, com o objetivo de ser o primeiro simulador de colonoscopia com retorno háptico. Ainda temos muito trabalho pela frente, visto que além de estarmos desenvolvendo uma ferramenta que poderá diminuir a dificuldade de aprendizado de novos cirurgiões, é um ótimo material de estudo e pesquisa para nossa equipe.

A escolha de usar um motor gráfico como a Unity nos ajudou muito no desenvolvimento, pois podemos usar diversas funções já implementadas. Além disso, a biblioteca de simulação física também foi essencial, e continuará sendo de extrema importância no desenvolvimento da nova malha do intestino.

Buscamos além de mudar a forma que a malha é gerada, também desenvolver um sistema de notas para a execução da cirurgia.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Unity engine. Versão: 2019.2.3f; Unity Technologies. Disponivel em: https://unity.com

PositionBasedDynamics. feito por JAN BENDER. Disponivel em: https://animation.rwth-aachen.de/software/position-based-dynamics/

Bender J.; Müller M.; Macklin M. A survey on position based dynamics. **EUROGRAPHICS** 2017 Tutorials. 2017.

Deul C.; Kugelstadt T.; Weiler M. e Bender J. Direct Position-Based Solver for Stiff Rods, **COMPUTER GRAPHICS forum** vol 37, p 313--324, 2018.

Müller, M.; Heidelberger, B.; Hennix, M.; e Ratcliff, J. Position based dynamics. **J. Vis. Comun. Image Represent**. vol 18, 2 (Apr.), p 109--118, 2007.

Berndt I, Torchelsen R, Maciel A. Efficient surgical cutting with position-based dynamics. **IEEE Comput Graph Appl** vol 38(3):24--31, 2017

Annaliese M. Plooy, Andrew Hill, Mark S. Horswill, Alanna St. G. Cresp, Marcus O. Watson, Soong-Yuan Ooi, Stephan Riek, Guy M. Wallis, Robin Burgess-Limerick, David G. Hewett, Construct validation of a physical model colonoscopy simulator, **Gastrointestinal Endoscopy**,vol 76, 144-150, 2012.