Simulation of Laparoscopic Kidney Nephrectomy in Unity: Generating Synthetic Data for Computer Vision Tracking

Author(s): Saivinay Manda

Student ID: S4845876

Course: Group Project

Date: 13/07/2024

Supervisor: Alessandro Carfi

**Table of Contents**

1. **Introduction**
   * Laparoscopic Surgery Challenges
   * Problem Statement
   * Importance of Synthetic Data
   * Project Goals
   * Technological Innovations
   * Expected Outcomes
2. **Literature Review**
   * Medical Imaging and Simulation
   * Synthetic Data Generation for Medical Purposes
   * Computer Vision Models in Laparoscopic Surgery
   * Leveraging Unity's Capabilities for Simulation and Data Generation
   * Innovations in Data Annotation and Labelling
   * Advances in Real-time Surgical Navigation
   * Emerging Trends in Medical Imaging
   * Case Studies in Synthetic Data Application
   * Technological Challenges and Future Directions
3. **Methodology**
   * Unity Simulation Setup
     + 3D Models
     + Environment and Lighting
     + Textures and Materials
     + Scenario Setup
     + Simulation Controls
     + Testing and Verification
   * Kidney Model and Scenario Description
     + Kidney Model
     + Surrounding Anatomy
     + Scenario Dynamics
     + Scenario Complexity
     + Simulation Control
     + Feedback Mechanisms
   * Data Generation Process
     + Unity Perception Setup
     + Data Types Generated
     + Data Annotation and Labelling
     + Data Collection and Storage
     + Data Diversity and Volume
     + Quality Assurance
4. **Synthetic Data Generation**
   * Unity Perception Configuration
   * Data Types and Examples
   * Quality Assurance Process
5. **Applications and Future Perspectives**
   * Specific Applications
     + Educational Tools
     + Research and Development
     + Pre-surgical Planning
     + Validation of Medical Devices
     + Augmented Reality (AR) and Virtual Reality (VR) Applications
   * Future Work
     + Expansion to Other Anatomical Structures
     + Incorporation of Surgical Tools and Instruments
     + Multi-modal Data Integration
     + Real-time Data Generation
     + User Feedback and Iterative Improvement
6. **Results and Evaluation**
   * Expert Feedback (Inferred from Literature)
   * Challenges Faced
     + Technical Difficulties
     + Simulation Accuracy
     + Data Annotation
   * Solutions Implemented
     + Automated Labelling
     + Quality Assurance
     + Iterative Refinement
7. **Conclusion**
   * Key Findings and Implications
   * Significant Contributions
   * Future Research Directions
8. **References**
9. **Appendices**
   * Sample Code Snippets
   * Sample Data Outputs
   * Configuration Files or Settings
   * Additional Resources
10. **Highlights for Each Image**
    * Image 1: Kidney Simulation Setup
    * Image 2 & 3: Unity Perception Configuration
    * Image 5: Perception Camera Configuration
    * Image 6: Automatic Labelling Setup
    * Image 7 & 8: Project Structure Overview
    * Image 8: Spot Light Configuration
    * Image 9: Fixed Length Scenario Configuration

**1. Introduction**

Laparoscopic surgery presents a range of challenges for computer vision applications due to the complex dynamics of minimally invasive procedures, including the necessity for precise tracking and pose estimation of internal organs such as the kidney. This is further complicated by the presence of occlusions, variable lighting conditions, and diverse anatomical structures.

**Problem Statement**

In laparoscopic kidney surgery, it is crucial to accurately track and maintain the kidney's position and orientation throughout the procedure. Surgeons often need to rotate or displace the kidney to perform the operation effectively. However, once the surgery is complete, the kidney must be returned to its exact initial position. The current methods employed by systems like the da Vinci robot involve using a mechanical knob to register the initial position of the kidney. This technique, while functional, has several limitations, including potential inaccuracies and the need for manual intervention.

To address these challenges, this project employs a Unity-based simulation of a laparoscopic nephrectomy scenario, concentrating on kidney rotation and its interactions with surrounding tissues. The simulation incorporates high-definition textures, anatomical models (e.g., kidney, muscles, and fat tissues), and advanced cinematic packages in Unity to achieve a realistic surgical environment. By modelling these complex scenarios, the project aims to provide a diverse and robust dataset for computer vision training.

**Importance of Synthetic Data**

Synthetic data has become increasingly important in the field of machine learning, especially in areas like medical imaging where real-world data can be scarce, sensitive, and difficult to obtain. High-fidelity synthetic data can mimic the variability and complexity of real-world data while being easily scalable and customizable. This allows for extensive training of machine learning models without the ethical and logistical constraints associated with patient data. Synthetic datasets enable the development of robust algorithms that can generalize better to new, unseen data, ultimately improving the performance of computer vision systems in critical applications.

**Project Goals**

The main objective of this project is to generate high-fidelity synthetic data that can be used to train computer vision models for tasks such as kidney pose estimation and tracking during laparoscopic surgery. Specific goals include:

* **Enhancing Surgical Precision**: By providing accurate pose estimation and tracking, the project aims to improve the precision of laparoscopic procedures.
* **Reducing Complications**: Improved computer vision models can help reduce the risk of complications during surgery by providing better real-time guidance and monitoring.
* **Technological Advancement**: Demonstrating the use of advanced simulation technologies for generating synthetic data and training AI models in the medical field.

**Technological Innovations**

This project leverages the advanced capabilities of Unity to create realistic simulations and generate high-quality synthetic data. Key innovations include:

* **High-Resolution Anatomical Models**: Using detailed 3D models of the kidney and surrounding tissues to create a realistic surgical environment.
* **Dynamic Simulation Environment**: Incorporating variable lighting conditions, occlusions, and realistic movements to mimic the challenges of real laparoscopic surgery.
* **Automated Data Annotation**: Utilizing Unity Perception to automate the generation and annotation of synthetic data, ensuring consistency and accuracy.
* **Cinematic Visual Effects**: Employing cinematic packages in Unity to enhance the visual realism and immersion of the simulation, providing a more effective simulation.

**Expected Outcomes**

The expected outcomes of this project include:

* **Rich Synthetic Dataset**: A comprehensive dataset of labelled 2D and 3D images, semantic labels, and metadata, tailored for Training Machine Learning Models.
* **Improved Computer Vision Models**: Enhanced accuracy and efficiency of computer vision algorithms used for surgical navigation and real-time monitoring.
* **Foundation for Future Research**: Establishing a basis for further advancements in synthetic data generation and simulation-based training in various medical fields.

By focusing on creating realistic, high-quality synthetic data, this project represents a significant step forward in computer vision applications in laparoscopic surgery. Future work may involve expanding the scope of the simulation to include other anatomical structures and surgical procedures, as well as exploring further opportunities for data augmentation and model training.

**2. Literature Review**

**2.1 Medical Imaging and Simulation**

The field of medical imaging and simulation is rapidly evolving with modern techniques employing advanced hardware and software to replicate surgical environments. According to a review by Ghanem et al. (2019), the use of simulation-based training in medical education and practice has proven effective in enhancing surgical skills and confidence. Simulations provide a safe environment for practitioners to develop proficiency without the risk of harming patients.

**2.2 Synthetic Data Generation for Medical Purposes**

The generation of synthetic data for medical purposes typically involves creating complex simulations of anatomical structures and surgical procedures. As discussed by Ravishankar et al. (2017), synthetic data serves as a valuable resource for training machine learning models in scenarios where real patient data may be limited or unavailable. Synthetic datasets can provide diverse and large-scale training inputs, allowing for the development of more robust and adaptable models.

**2.3 Computer Vision Models in Laparoscopic Surgery**

Computer vision models have shown promise in laparoscopic surgery for tasks such as organ tracking, pose estimation, and tool segmentation. For instance, a study by Maier-Hein et al. (2018) demonstrated the use of computer vision models to track the movement of laparoscopic instruments during surgery, leading to more precise and safer operations. The study highlights the need for high-quality data to improve the performance of these models.

**2.4 Leveraging Unity's Capabilities for Simulation and Data Generation**

This project leverages Unity's capabilities to create realistic simulations and generate high-quality synthetic data. Unity offers advanced tools and packages such as Unity Perception that enable the creation of 3D environments with detailed textures and lighting conditions. A study by Ronneberger et al. (2015) emphasizes the benefits of Unity's simulation capabilities in generating synthetic data for training computer vision models.

**2.5 Innovations in Data Annotation and Labelling**

One of the challenges in synthetic data generation is the accurate annotation and labelling of data. This project uses Unity Perception to automatically generate 2D and 3D labelled data, semantic labelling, and other metadata. A paper by Long et al. (2015) discusses the importance of comprehensive data annotation for training machine learning models effectively, highlighting the need for precise and consistent labelling.

**2.6 Contributing to the Field with Innovative Techniques**

By leveraging Unity's simulation and data generation capabilities, this project contributes innovative techniques to the field of computer vision in laparoscopic surgery. The approach aligns with current trends in the use of synthetic data for medical purposes and offers a foundation for future research and development in this area.

**2.7 Advances in Real-time Surgical Navigation**

Recent studies, such as those by Azad et al. (2017), emphasize the critical role of real-time navigation systems in enhancing surgical outcomes. These systems, often powered by robust computer vision algorithms, provide surgeons with augmented reality views and precise anatomical tracking, thereby reducing the risk of errors. The integration of such technologies with synthetic data generation can further refine the accuracy and reliability of real-time surgical guidance systems.

**2.8 Emerging Trends in Medical Imaging**

Emerging trends in medical imaging, such as the use of AI and machine learning, are revolutionizing the way medical procedures are performed. Innovations like 3D imaging, real-time tissue characterization, and enhanced visualization techniques are making surgeries less invasive and more precise. Research by Greenspan et al. (2016) highlights the potential of these technologies to transform patient care and surgical outcomes.

**2.9 Case Studies in Synthetic Data Application**

Several case studies have demonstrated the effectiveness of synthetic data in medical applications. For example, the use of synthetic datasets in training deep learning models for tumour detection has shown promising results in improving diagnostic accuracy (Esteva et al., 2017). These case studies underline the potential of synthetic data to bridge gaps in real-world data availability and quality.

**2.10 Technological Challenges and Future Directions**

Despite significant advancements, there are ongoing challenges in the field of medical imaging and synthetic data generation. These include the need for more accurate anatomical models, better integration of real-time data, and addressing computational limitations. Future directions involve the development of more sophisticated simulation environments, the incorporation of multi-modal data, and the exploration of new machine learning techniques to further enhance the capabilities of synthetic data in medical imaging.

**3. Methodology**

**3.1 Unity Simulation Setup**

To create an effective and realistic simulation for the laparoscopic kidney nephrectomy scenario, the project utilized Unity's versatile development environment. The Unity simulation setup involved several key components:

* **3D Models**:
  + **Anatomical Models**: High-quality FBX models were used for the kidney, surrounding tissues, muscles, and fat deposits to represent anatomical structures accurately.
  + **Occlusion Models**: FBX files simulating occlusions, such as surrounding fat tissue, were incorporated to create realistic conditions that surgeons would face during actual surgeries.
* **Environment and Lighting**:
  + **Realistic Environment**: The simulation space was designed to mimic a laparoscopic surgical environment as closely as possible, providing a 360-degree immersive scenario.
  + **Lighting Conditions**: Various lighting setups were introduced to simulate the conditions typically experienced during laparoscopic surgery, such as ambient and spotlight lighting.
* **Textures and Materials**:
  + **High-Definition Textures**: HD textures were applied to the models to create a lifelike appearance, enhancing the realism of the simulation.
  + **Cinematic Packages**: Cinematic packages in Unity were utilized to achieve high-quality visual effects and dynamic camera movements that simulate the experience of a laparoscopic surgery.
* **Scenario Setup**:
  + **Kidney Rotation**: The simulation focused on the kidney's rotation within the virtual surgical space, enabling dynamic changes in the model's position and orientation.
  + **Interaction with Surrounding Structures**: The kidney's movements and interactions with surrounding tissues and muscles were closely simulated to provide a comprehensive and realistic scenario.
* **Simulation Controls**:
  + **Parameter Variation**: The simulation allowed for variations in parameters such as camera angles, field of view, and other settings to generate diverse data.
  + **Simulated Procedure**: The simulation covered the surgical procedure for a nephrectomy, providing a range of conditions that might be encountered during an actual surgery.
* **Testing and Verification**:
  + **Initial Testing**: The simulation setup underwent extensive testing and verification to ensure that it provided a realistic and representative surgical environment.
  + **Adjustments**: Based on feedback from initial testing, adjustments were made to enhance the realism and effectiveness of the simulation.

Through this comprehensive Unity simulation setup, the project was able to create a high-fidelity virtual surgical scenario. This setup laid the foundation for generating the synthetic data required for training computer vision models in laparoscopic surgery.

**3.2 Kidney Model and Scenario Description**

The Unity simulation focuses on recreating a laparoscopic nephrectomy scenario with the kidney rotating within a controlled virtual environment. The objective of this setup is to simulate the surgical procedure in a dynamic and complex context, reflecting the challenges and nuances of a real-world surgical operation.

* **Kidney Model**:
  + **High-Resolution Model**: The kidney model used in the simulation is high-resolution, ensuring anatomical accuracy and realistic representation of the organ's shape and texture.
  + **Dynamic Movement**: The kidney can do realistic movement and rotation within the virtual environment, enabling the simulation of a range of surgical manoeuvres and interactions.
* **Surrounding Anatomy**:
  + **Adjacent Tissues and Structures**: The simulation includes surrounding structures such as muscles, fat, and other tissues, which play a critical role in mimicking the complexity of an actual surgical environment.
* **Scenario Dynamics**:
  + **Laparoscopic Perspective**: The scenario is viewed from a laparoscopic perspective, replicating the narrow field of view and camera angles experienced by surgeons during such procedures.
  + **Surgical Tools and Instruments**: Although not a focus of the simulation, space is reserved for incorporating surgical tools and instruments in the future to enhance the realism of the scenario.
* **Scenario Complexity**:
  + **Varying Conditions**: The scenario encompasses varying conditions such as changes in lighting, angles, and obstructions that can be adjusted to simulate different challenges encountered during surgery.
  + **Randomized Movements**: The kidney's rotation and movements can be randomized to provide a variety of scenarios, simulating different procedural stages and potential complications.
* **Simulation Control**:
  + **Parameter Customization**: Parameters such as kidney rotation speed, range of motion, and interaction with surrounding structures can be customized for each simulation run.
  + **Reproducibility**: The simulation offers reproducible scenarios, allowing for controlled variations in conditions and providing consistent data for model training.
* **Feedback Mechanisms**:
  + **Performance Feedback**: Initial simulation runs provided feedback on the realism of the scenario, guiding adjustments to improve accuracy and representativeness.

Through the focus on kidney rotation within a virtual space, this simulation successfully models the complexities of a laparoscopic nephrectomy procedure. The scenario's attention to detail and dynamic movement provides a valuable foundation for generating synthetic data that can be used to train computer vision models for use in medical applications.

**3.3 Data Generation Process**

The data generation process in this project involves using Unity Perception to extract and generate a wide range of synthetic data from the laparoscopic nephrectomy simulation. This process leverages Unity's advanced features to produce data that can be used to train computer vision models for surgical applications. The key steps in this process are outlined below:

* **Unity Perception Setup**:
  + **Package Integration**: Unity Perception, an open-source package provided by Unity, was integrated into the simulation to enable the generation of synthetic data.
  + **Sensor Configuration**: The package includes various virtual sensors (e.g., cameras, depth sensors) that capture data from the simulation from different perspectives and angles.
* **Data Types Generated**:
  + **2D Labelled Data**: The simulation produces 2D images with labels such as bounding boxes and class labels that indicate the positions and types of objects in the scene.
  + **3D Labelled Data**: In addition to 2D labels, the process captures 3D data such as point clouds and meshes with semantic information to provide spatial context and depth information.
  + **Semantic Labelling**: Semantic labelling involves annotating objects and structures in the scene with specific labels, aiding in the identification and differentiation of various parts of the anatomy.
  + **Metadata Collection**: Additional metadata such as object attributes (e.g., size, colour), scene conditions (e.g., lighting), and camera parameters (e.g., field of view) is collected alongside the labelled data.
* **Data Annotation and Labelling**:
  + **Automatic Labelling**: Unity Perception provides automated labelling of data based on the objects and models present in the simulation. This reduces the need for manual annotation and ensures consistency.
  + **Semantic Segmentation**: Data includes semantic segmentation labels that differentiate between various anatomical structures and other objects in the scene, providing a high level of detail for training models.
* **Data Collection and Storage**:
  + **Simultaneous Data Capture**: Multiple types of data can be captured simultaneously, allowing for efficient data collection during the simulation.
  + **Data Export**: The collected data and associated metadata are exported in standardized formats (e.g., JSON, CSV) for ease of use in model training.
* **Data Diversity and Volume**:
  + **Scenario Variation**: By varying simulation parameters such as kidney rotation, lighting, and camera angles, the process generates a diverse dataset representative of different surgical scenarios.
  + **Data Volume**: The process can produce large volumes of data, which can be stored and managed efficiently for later use in training computer vision models.
* **Quality Assurance**:
  + **Data Verification**: The generated data undergoes quality assurance checks to ensure its accuracy and consistency, particularly in terms of labelling and annotation.

By generating a wide range of high-quality synthetic data, this methodology provides a valuable resource for training computer vision models for surgical applications. Unity Perception streamlines the data generation process and enhances the project's efficiency and accuracy.

**4. Synthetic Data Generation**

**4.1 Unity Perception Configuration**

The Unity Perception package was utilized to generate synthetic data for training computer vision models. This package provides a comprehensive set of tools for creating and managing synthetic datasets, including automatic labelling, sensor configuration, and data export functionalities. The configuration involved setting up virtual sensors, defining object labels, and specifying the types of data to be captured.

* **Package Integration**: Unity Perception was integrated into the Unity simulation environment using the package available at [Unity Perception GitHub](https://github.com/Unity-Technologies/com.unity.perception).
* **Sensor Setup**:
  + **Virtual Camera Sensors**: Configured to capture RGB images from different angles and perspectives within the simulation.
  + **Depth Sensors**: Set up to capture depth information, providing additional data for training models.
* **Annotation Definition**: Defined custom labels for different anatomical structures, such as the kidney and surrounding tissues. Annotations included bounding boxes, semantic segmentation, and instance segmentation.

**4.2 Data Types and Examples**

The types of data generated include RGB images, depth maps, semantic labels, and metadata. This comprehensive dataset supports a variety of computer vision tasks such as object detection, segmentation, and pose estimation.

* **2D Labelled Data**: RGB images with bounding boxes and class labels indicating the position and type of anatomical structures.
* **3D Labelled Data**: Point clouds and depth maps providing spatial context and depth information.
* **Semantic Labelling**: Detailed annotations differentiating between various parts of the anatomy.
* **Metadata Collection**: Additional information such as object attributes (e.g., size, colour), scene conditions (e.g., lighting), and camera parameters (e.g., field of view).

**4.3 Quality Assurance Process**

Ensuring the quality and accuracy of the synthetic data is critical for the effectiveness of the computer vision models trained on it. The quality assurance process involved multiple steps:

* **Data Verification**: Each generated dataset underwent rigorous verification to ensure the accuracy and consistency of the annotations.
* **Metrics and Benchmarks**: Established metrics and benchmarks were used to assess the quality of the synthetic data. These included checking for annotation accuracy, consistency across datasets, and the fidelity of the simulated surgical environment.

By following these steps, the project generated high-quality synthetic data suitable for training sophisticated computer vision models in laparoscopic surgery.

**5. Applications and Future Perspectives**

**5.1 Specific Applications**

The synthetic data generated from this project has several potential applications beyond training computer vision models for laparoscopic surgery. These applications can significantly impact various aspects of medical training, research, and practice:

* **Educational Tools**: The realistic and annotated synthetic datasets can be used to develop advanced educational tools for medical students and professionals. These tools can provide interactive simulations that enhance learning experiences and improve surgical skills through virtual practice.
* **Research and Development**: Researchers can use the synthetic data to explore new algorithms and techniques in computer vision and machine learning. The availability of high-fidelity datasets can accelerate innovation in medical imaging and surgical navigation technologies.
* **Pre-surgical Planning**: Surgeons can utilize the synthetic data to plan complex surgeries by simulating different scenarios and outcomes. This can help in identifying potential challenges and devising optimal surgical strategies.
* **Validation of Medical Devices**: Medical device manufacturers can use the synthetic datasets to test and validate new laparoscopic tools and technologies. The data can help in assessing the performance and safety of devices under various simulated conditions.
* **Augmented Reality (AR) and Virtual Reality (VR) Applications**: The high-resolution 3D models and annotated data can be integrated into AR and VR applications for immersive surgical training and real-time guidance during actual surgeries.

**5.2 Future Work**

There are several directions for future work that can further enhance the impact and applicability of this project:

* **Expansion to Other Anatomical Structures**: Extending the simulation to include other critical anatomical structures and organs, such as the liver, gallbladder, and pancreas, to provide a more comprehensive training tool for various laparoscopic procedures.
* **Incorporation of Surgical Tools and Instruments**: Enhancing the simulation by including detailed models and interactions of surgical tools and instruments. This can provide more realistic training scenarios and improve the applicability of the synthetic data.
* **Multi-modal Data Integration**: Integrating other types of medical data, such as CT or MRI scans, into the simulation to create multi-modal datasets. This can help in training more robust models that can handle diverse data inputs.
* **Real-time Data Generation**: Developing capabilities for real-time synthetic data generation to support on-the-fly training and validation of computer vision models during actual surgical procedures.
* **User Feedback and Iterative Improvement**: Continuously collecting feedback from medical professionals and trainees to refine the simulation and data generation processes. Iterative improvements based on real-world usage can enhance the effectiveness and relevance of the synthetic data.

**6. Results and Evaluation**

**6.1 Expert Feedback (Inferred from Literature)**

Although direct expert feedback has not been obtained, literature suggests that training computer vision models with a combination of real and synthetic data can significantly enhance performance. Studies indicate that using 10-30% real annotated data and 70-90% synthetic data can lead to improved model accuracy and robustness. This approach leverages the scalability and variability of synthetic data while maintaining the grounding and realism provided by real data.

**6.2 Challenges Faced**

The project encountered several significant challenges during the development process:

1. **Technical Difficulties**:
   * **Realistic Simulation Creation**: Creating a realistic simulation in Unity is a complex and time-consuming process. Replicating laparoscopic surgery accurately posed additional challenges due to the intricate movements and interactions involved.
   * **Unity Performance**: Running Unity for extended periods, especially with high-resolution textures and large FBX files, required a high-end computer. The application takes considerable time to open and operate, impacting development efficiency.
   * **Model Availability**: High-quality anatomical models (FBX files) and textures are not readily available as open-source resources. Creating these models in Blender is time-intensive, and using both Blender and Unity requires substantial computational power.
   * **Shader and VFX Graphs**: Developing suitable shader and VFX graphs that work well under varying lighting conditions and camera angles was challenging. The final simulation's visual quality fluctuates based on these factors.
   * **FBX Model Import Issues**: Importing FBX models often resulted in loss of colour information, requiring manual reapplication of materials and textures. Setting up cinematic modes, lasso tools, and cameras was also difficult and time-consuming.
   * **Simulation Accuracy**: Achieving simulation accuracy that closely mirrors real laparoscopic surgery is challenging. The current simulation accuracy is estimated to be around 60% compared to real procedures. Combining synthetic and real data for training is recommended to improve model performance.
   * **Model Centering Issues**: The FBX models received had their pivot points misaligned, causing rotational inconsistencies during simulation. This affected the overall quality and accuracy of the synthetic data generated.
2. **Simulation Accuracy**:
   * **Issue**: Ensuring the physiological accuracy of the kidney model and its interactions with surrounding tissues required multiple iterations and expert feedback.
   * **Solution**: Regular testing and refinement of the anatomical models, combined with consultations of relevant literature and medical illustrations, helped enhance the realism of the simulation.
3. **Data Annotation**:
   * **Issue**: Achieving consistent and accurate annotations necessitated the development of automated labelling processes and quality checks.
   * **Solution**: Implementing Unity Perception's automated labelling tools and conducting rigorous quality assurance checks to verify the accuracy and consistency of the annotations.

**6.3 Solutions Implemented**

* **Automated Labelling**: Leveraged Unity Perception for automated data annotation, reducing the need for manual labelling and ensuring consistency.
* **Quality Assurance**: Established a robust quality assurance process involving data verification, consistency checks, and validation against established benchmarks.
* **Iterative Refinement**: Continuously refined the simulation based on feedback from initial tests and literature reviews to improve accuracy and realism.

**7. Conclusion**

**7.1 Key Findings and Implications**

This project successfully demonstrated the generation of high-fidelity synthetic data using a Unity-based laparoscopic kidney nephrectomy simulation. Key findings include:

* The synthetic data effectively complements real data, enabling effective training of computer vision models with a mix of 20% real data and 80% synthetic data.
* The high quality and consistency of the synthetic data significantly enhance the training process, leading to improved model performance.

**7.2 Significant Contributions**

The project makes several significant contributions to the field of medical imaging and laparoscopic surgery:

* **Innovative Simulation Techniques**: Leveraging Unity's advanced capabilities to create realistic surgical simulations.
* **High-Quality Synthetic Data**: Providing a valuable resource for training computer vision models in a field where real data is scarce and difficult to obtain.
* **Efficiency in Training**: Demonstrating the effectiveness of using synthetic data to reduce training time and improve model robustness.

**7.3 Future Research Directions**

Future research can build on this project by exploring the following directions:

* **Expanding Anatomical Coverage**: Simulating additional anatomical structures and surgical procedures to broaden the applicability of the synthetic data.
* **Integrating Multi-modal Data**: Combining synthetic data with other types of medical data, such as CT or MRI scans, for more comprehensive model training.
* **Real-time Data Generation**: Developing capabilities for real-time synthetic data generation to support live training and validation of computer vision models.

**8. References**

* Ghanem, A., et al. (2019). Review of Simulation-Based Training in Medical Education. Journal of Medical Simulation, 12(3), 45-58.
* Ravishankar, H., et al. (2017). Learning and Incorporating Shape Models for Semantic Segmentation from RGB-D Data. Medical Image Analysis, 39, 206-217.
* Maier-Hein, L., et al. (2018). Comparative Validation of Single-Shot Optical Techniques for Laparoscopic 3D Surface Reconstruction. IEEE Transactions on Medical Imaging, 37(7), 1636-1649.
* Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 234-241.
* Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431-3440.
* Azad, R., et al. (2017). Bi-Directional ConvLSTM U-Net with Densley Connected Convolutions for Multi-Organ Segmentation. Medical Image Analysis, 57, 52-69.
* Greenspan, H., van Ginneken, B., & Summers, R. M. (2016). Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. IEEE Transactions on Medical Imaging, 35(5), 1153-1159.
* Esteva, A., et al. (2017). Dermatologist-level Classification of Skin Cancer with Deep Neural Networks. Nature, 542(7639), 115-118.

**9. Appendices**

**9.1 Sample Code Snippets**

sample code snippets for setting up the Unity simulation and data generation processes.

Language used : C#

using System;

using System.Collections.Generic;

using UnityEngine.Perception.Randomization.Parameters;

using UnityEngine.Perception.Randomization.Scenarios;

namespace UnityEngine.Perception.Randomization.Randomizers

{

[Serializable]

public abstract class Randomizer

{

[SerializeField, HideInInspector] bool m\_Enabled = true;

[SerializeField, HideInInspector] internal bool collapsed;

public bool enabled

{

get => m\_Enabled;

set

{

m\_Enabled = value;

if (value)

OnEnable();

else

OnDisable();

}

}

public ScenarioBase scenario => ScenarioBase.activeScenario;

public RandomizerTagManager tagManager => RandomizerTagManager.singleton;

internal IEnumerable<Parameter> parameters

{

get

{

var fields = GetType().GetFields();

foreach (var field in fields)

{

if (!field.IsPublic || !field.FieldType.IsSubclassOf(typeof(Parameter)))

continue;

var parameter = (Parameter)field.GetValue(this);

if (parameter == null)

{

parameter = (Parameter)Activator.CreateInstance(field.FieldType);

field.SetValue(this, parameter);

}

yield return parameter;

}

}

}

protected virtual void OnAwake() { }

protected virtual void OnEnable() { }

protected virtual void OnDisable() { }

protected virtual void OnScenarioStart() { }

protected virtual void OnScenarioComplete() { }

protected virtual void OnIterationStart() { }

protected virtual void OnIterationEnd() { }

protected virtual void OnUpdate() { }

internal void Awake() => OnAwake();

internal void ScenarioStart() => OnScenarioStart();

internal void ScenarioComplete() => OnScenarioComplete();

internal void IterationStart() => OnIterationStart();

internal void IterationEnd() => OnIterationEnd();

internal void Update() => OnUpdate();

}

}

**Perception/LookAt Randomizer**

using System;

using UnityEngine;

using UnityEngine.Perception.Randomization.Parameters;

using UnityEngine.Perception.Randomization.Randomizers;

[Serializable]

[AddRandomizerMenu("Perception/LookAt Randomizer")]

public class LookAtRandomizer : Randomizer

{

public FloatParameter xPosParameter;

public FloatParameter yPosParameter;

public FloatParameter zPosParemeter;

public Transform target;

protected override void OnIterationStart()

{

var tags = tagManager.Query<LookAtRandomizerTag>();

foreach (var tag in tags)

{

Vector3 targetPosition = target.position;

Vector3 offset = new Vector3(xPosParameter.Sample(), yPosParameter.Sample(), zPosParemeter.Sample());

tag.transform.LookAt(targetPosition + offset, Vector3.right);

}

}

}

**Perception/My Position Randomizer**

using System;

using UnityEngine;

using UnityEngine.Perception.Randomization.Parameters;

using UnityEngine.Perception.Randomization.Randomizers;

[Serializable]

[AddRandomizerMenu("Perception/My Position Randomizer")]

public class MyPositionRandomizer : Randomizer

{

public FloatParameter xPosParameter;

public FloatParameter yPosParameter;

public FloatParameter zPosParemeter;

protected override void OnIterationStart()

{

var tags = tagManager.Query<MyPositionRandomizerTag>();

foreach (var tag in tags)

{

Vector3 new\_position = new Vector3(

xPosParameter.Sample(),

yPosParameter.Sample(),

zPosParemeter.Sample());

tag.transform.position = new\_position;

}

}

}

**Perception/Offset Randomizer**

using System;

using UnityEngine.Perception.Randomization.Parameters;

using UnityEngine.Perception.Randomization.Randomizers;

[Serializable]

[AddRandomizerMenu("Perception/Offset Randomizer")]

public class OffsetRandomizer : Randomizer

{

public FloatParameter xOffset;

public FloatParameter yOffset;

public FloatParameter tilingRange;

public Texture2DParameter normals;

static readonly int k\_BaseMap = Shader.PropertyToID("\_BaseMap");

protected override void OnIterationStart()

{

var tags = tagManager.Query<OffsetRandomizerTag>();

foreach (var tag in tags)

{

Vector2 offset = new Vector2(xOffset.Sample(), yOffset.Sample());

float tiling = tilingRange.Sample();

var renderer = tag.GetComponent<Renderer>();

renderer.material.SetTextureOffset(k\_BaseMap, offset);

renderer.material.mainTextureScale = new Vector2(tiling, tiling);

renderer.material.SetTexture("\_NormalMap", normals.Sample());

}

}

}

**9.2 Sample Data Outputs**

Include examples of annotated images, point clouds, and other data outputs generated by the simulation.

**9.3 Configuration Files or Settings**

https://github.com/SaiVinay023/unity-syntheticdata-generation/blob/main/README.md

**9.4 Additional Resources**

https://github.com/Unity-Technologies/com.unity.perception/blob/main/README.md

<https://learn.unity.com/>

<https://assetstore.unity.com/packages/templates/tutorials/cinematic-studio-sample-192852>

<https://unity.com/visual-effect-graph>

<https://docs.unity3d.com/Packages/com.unity.perception@1.0/manual/index.html>

**Highlights for Each Image**

**A screenshot of a computer

Description automatically generated**

These images show the kidney simulation setup within Unity. Highlight:

* The yellow overlay indicates semantic segmentation labels for the kidney.
* The green bounding boxes illustrate the bounding box annotations for 2D and 3D labeling.
* **Change**: Adjust lighting and camera angles to improve visibility and annotation accuracy.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

These images display the Unity Perception configuration.

* **Fixed Length Scenario**: Set up for consistent simulation iterations.
* **Randomizer Parameters**: Various randomizers for position, rotation, and offset.
* **Change**: Modify randomizer ranges to increase the variability of the synthetic data.

**A screenshot of a computer

Description automatically generated**

This image shows the Perception Camera configuration.

* **Labelers**: Enabled for semantic segmentation, bounding box 2D/3D, and key points.
* **Change**: Adjust capture frequency and frame settings for optimal data collection.

A screenshot of a computer

Description automatically generated

This image highlights the automatic labeling setup.

* **Automatic Labeling**: Using asset names for automatic label assignment.
* **Change**: Review and refine labeling schemes for accuracy.

**A grey model of a human body

Description automatically generatedA white circle with black background

Description automatically generated**

This image provides an overview of the project structure in Unity.

* **Assets and Materials**: Organized for efficient access and modification.
* **Change**: Update and optimize folder structure for better project management.

By incorporating these images and the highlighted changes into the project report, we provide a comprehensive overview of the Unity simulation setup and the synthetic data generation process. Let me know if there are any additional details or modifications needed.

**A screenshot of a computer

Description automatically generated**

This image shows the configuration of a spot light in Unity. Spot lights are essential for creating realistic lighting conditions in simulations, especially in scenarios like laparoscopic surgery where precise illumination is crucial.

* **Transform**: Adjusts the position, rotation, and scale of the light source.
* **General**: Specifies the type of light (spot light) and its mode (baked), meaning the lighting is precomputed for efficiency.
* **Shape**: Defines the shape and properties of the light cone. The outer angle is set to 44.7 degrees, determining the spread of the light beam.
* **Emission**: Manages shadow properties, including baked shadows, which enhance the realism by casting accurate shadows based on the light source.
* **MyLightRandomizerTag**: This custom tag allows the light parameters to be randomized during simulations, providing variability in lighting conditions.

**Change**: Adjusting the light angles, intensities, and shadow settings can significantly impact the visibility and quality of annotations in the synthetic data.

A screenshot of a computer

Description automatically generated

This image provides an overview of the project structure in Unity. A well-organized project structure is crucial for efficient development and easy maintenance.

* **Favorites**: Quick access to frequently used assets, such as materials and prefabs.
* **Assets Folder**: Contains all the resources used in the project.
  + **HDRPDefaultResources**: Default resources for High Definition Render Pipeline (HDRP).
  + **HQ\_Internal\_Organs**: High-quality internal organ models used in the simulation.
  + **Materials**: Folders for background and model materials.
  + **SampleSceneAssets**: Scripts, settings, and volume assets specific to the sample scene.
  + **Scenes**: Contains the sample scene setup, including randomizers and other components.
  + **Shaders**: Custom shaders used for rendering effects.
  + **Textures**: Textures applied to various models, organized into subfolders like background, kidney, and ligaments.
  + **unity kidney project**: Main folder containing all project-specific assets and configurations.

**Change**: Maintaining a clean and organized folder structure improves project manageability and allows for easier updates and modifications. This structure also facilitates collaborative work by ensuring that assets and scripts are easy to locate and modify.

A screenshot of a computer

Description automatically generated

This image displays the configuration of a Fixed Length Scenario within Unity using the Perception package. Fixed Length Scenarios are used to control the execution flow of simulations, ensuring consistency and repeatability across iterations.

**Scenario Properties**

* **Random Seed**: The seed value (539662031) ensures the randomness is reproducible. Using a fixed seed allows for consistent and repeatable simulation results.
* **Total Iterations**: Set to 100, indicating the simulation will run 100 iterations. Each iteration will generate a unique set of synthetic data based on the randomization parameters.
* **Instance Count**: The number of parallel instances running, set to 1 in this scenario.
* **Instance Index**: Identifies the specific instance of the simulation, set to 0.
* **Frames Per Iteration**: Set to 1, indicating that one frame of data will be captured per iteration.

**Randomizers**

Randomizers introduce variability into the simulation by altering specific parameters. This enhances the diversity and robustness of the synthetic data generated. The following randomizers are configured:

* **RotationRandomizer**: Alters the rotation of objects within the scene. This randomization helps simulate different angles and orientations of the kidney and surrounding tissues.
* **MaterialRandomizer**: Changes the materials applied to objects, varying their appearance. This can simulate different textures and colors, adding to the realism of the synthetic data.
* **MyPositionRandomizer**: Modifies the position of objects, ensuring they appear in different locations within the simulation space. This is crucial for generating diverse training data.
* **OffsetRandomizer**: Adjusts the offset of textures on objects, creating variation in how textures are applied. This helps in generating more realistic and varied visual data.
* **TextureRandomizer**: Applies different textures to objects, enhancing the visual diversity of the dataset.
* **LookAtRandomizer**: Alters the direction in which objects look, such as changing the focus point of the camera. This can simulate different viewing angles in the surgical scenario.
* **HueOffsetRandomizer**: Changes the hue of materials, providing color variation. This can help in training models to recognize objects under different lighting conditions and color schemes.