NeeCo: Novel Instrument Deformation Image Synthesis Based on Dynamic 3D Gaussian Reconstruction

Tianle Zeng, Junlei Hu, Gerardo Loza Galindo, Pietro Valdastri *Fellow, IEEE*, and Dominic Jones*, *Member, IEEE*

***Abstract*— Computer vision-based technologies significantly enhance the automation capabilities of robotic-assisted minimally invasive surgery by advancing tool tracking, detection, and localization. However, the lack of high-quality labeled image datasets constrains these techniques, which require large amounts of data for training. The high-dynamic surgical scenario poses a considerable challenge to the image synthesis methods. This research introduces a novel method using 3D Gaussian Splatting to overcome the scarcity of surgical image datasets. We propose a dynamic 3D Gaussian model to represent dynamic surgical scenes, enabling the rendering of surgical instruments from unseen viewpoints and deformations with real tissue backgrounds. Utilizing a dynamic training adjustment strategy, we address challenges posed by poorly calibrated camera poses from real-world dynamic scenes. Additionally, we propose a method based on dynamic Gaussians for automating the generation of annotations for our synthetic data. For evaluation of the method, we construct a new dataset with 7 scenes 14,000 frames recording tool and camera motion, as well as an articulation of the tool jaw, with a background of an ex-vivo porcine model. Using this dataset, we synthetically replicate the deformed instrument of ground truth data, allowing direct comparisons of synthetic image quality. Experimental results illustrate that our method generates photo-realistic labeled image datasets (29.87 PSNR). We further compare the performance of three U-Net and YOLO models trained on real, synthetic, and mixed synthetic images, respectively, by assessing their performance on an unseen real-world image dataset. Our results show that the performance of models trained on synthetic images and real images differs by less than 1.5% across various metrics, while the model trained on the mixed synthetic dataset shows an improvement in model performance by nearly 10%.**

***Index Terms*— Surgical Data Science, Surgical AI, Data generation, 3D Gaussian splatting, Laparoscopy.**

I. Introduction

SURGICAL robotics can significantly enhance automation and intelligence in minimally invasive procedures like laparoscopic surgery. By integrating Robotic-Assisted Minimally Invasive Surgery (RAMIS) with Computer Assisted Interventions (CAI), this technology not only improves the precision and flexibility of surgical operations but also reduces patient recovery times and complication rates. Accurate real-time tracking, segmentation, and classification of surgical instruments are essential for both RAMIS and CAI. These capabilities facilitate intelligent surgical navigation, optimize surgical planning, and enhance overall surgical efficiency and safety, which are crucial steps towards automating laparoscopic surgery. Achieving these goals relies heavily on deep learning-based image understanding methods, such as semantic segmentation and object detection.

Deep learning-based methods require a substantial amount of annotated image data for training to ensure robustness in complex and highly dynamic surgical scenarios. However, the lack of high-quality labeled surgical image datasets has constrained the development of these methods [1]. This scarcity can be attributed to several factors. Ethical considerations in recording surgical videos make it complex to manage and share medical data [2]. In the laparoscope, a limited field of view variable lens distortion under different focuses often results in poor image quality which is impossible to calibrate. During the procedures, surgical tools, blood, and diathermy smoke regularly occlude the camera's view, leading to incomplete visual information. These factors make the generation of a high-quality image dataset difficult, and generating detailed annotations requires significant time and labor costs.

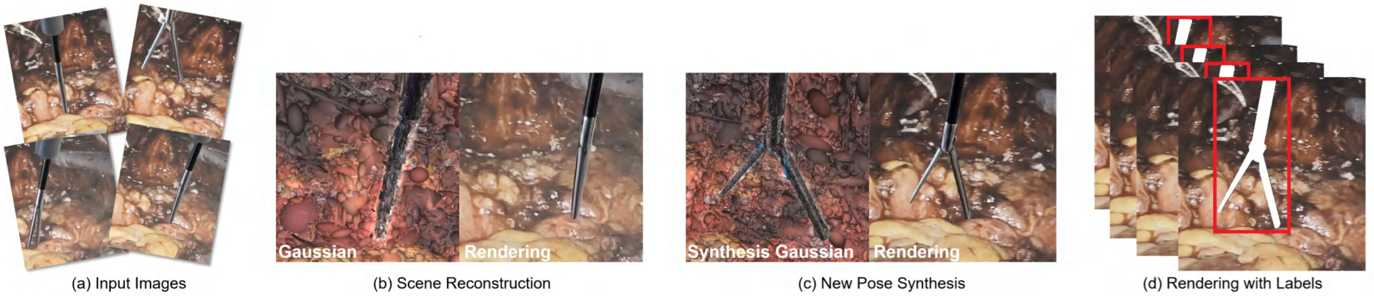
As a result, developing methodologies for addressing data scarcity in laparoscopy has become a key research focus. The use of 3D virtual simulators to generate synthetic images [3] and render textures using photo rendering software [4] gives one option. However, these images lack realism, failing to replicate the lighting and textures of real scenes, making them unsuitable for direct use in training neural networks. Alternatively, methods using Simultaneous Localization and Mapping (SLAM) techniques [5], [6], [7] reconstruct static surgical scenes by generating individual meshes for each frame and rendering images. However, these methods cannot generate annotated data, necessitating manual annotation, which limits their application in supervised learning tasks. Additionally, the quality of the images generated by these methods is often subpar.

To overcome the challenges of manual annotation and improve image quality, some studies have explored the use

THis paper was submitted for review on XXX

T. Zeng, G. L.Galindo, J. Hu, P. Valdastri, and D. Jones are with the STORM Lab UK, School of Electronic and Electrical Engineering, University of Leeds, Leeds, UK. (e-mail:D.P.Jones@leeds.ac.uk).

Pietro funding link?



(a) Input Images (b) Scene Reconstruction (c) New Pose Synthesis (d) Rendering with Labels

Fig. 1. **Method Overview.** (a) Our method takes images of surgical instruments in various poses as input. (b) Then we reconstruct the 3D Gaussian representation of the surgical scene. (c) After the reconstruction, our method can predict the 3D Gaussian representation of unseen deformations of the surgical instruments. (d) We render them as realistic images, and additionally can render the tool annotations automatically. In (d) the red boxes indicate the bounding boxes, and the white areas represent the instrument masks.

of image synthesis [8] and weak annotations [9]. Other works have proposed the use of generative models and Image-to-Image Translation (I2I) techniques [10] to generate simulated data for model training. These methods aim to produce realistic surgical images through fitting and adjustment, effectively addressing the annotation problem. However, images generated by these approaches are predominantly suited for static scenes, whereas laparoscopic surgery are often highly dynamic, involving instrument motion, changes in instrument pose, and soft tissue deformation.

Neural Radiance Fields (NeRFs) [11] and 3D Gaussian Splatting (3DGS) [12] have shown significant potential for image dataset generation. These technologies can reconstruct 3D scenes and achieve high-quality image rendering, as well as render images from novel viewpoints. This capability allows these methods to generate images from previously unseen camera views, facilitating data augmentation and significantly increasing the diversity of generated images. Moreover, both methods have the potential to handle dynamic scenes, recent methods that combine NeRFs or 3DGS with time-dependent neural displacement fields [13], [14] have become representative works in dynamic surgical scene reconstruction. Compared to NeRF, 3DGS has an advantage in reconstruction, enabling explicit scene representation and offering scene-editing capabilities that NeRF lacks. This makes 3DGS more suitable for reconstructing dynamic surgical scenes and generating corresponding image data.

Although 3DGS-based methods [15], [16] have achieved dynamic representation of surgical scenes, they still face several challenges. Firstly, these methods perform well when using simulated datasets because they have access to ground truth camera poses and scene point cloud motion. However, their performance degrades significantly when processing real-world datasets [16]. This degradation occurs because these methods rely on structure-from-motion (SfM) techniques to calibrate camera poses from the input scene for 3DGS initialization. In real-world dynamic scenes, due to significant changes between frames, SfM techniques struggle to match frames and register corresponding points accurately, resulting in substantial initialization errors. These errors severely affect 3DGS training. Current dynamic 3DGS methods are, therefore, challenging to apply to real-world scenarios. Method [13], [17] utilizes additional sensors to record precise camera pose data, enabling it to be applied to real surgical scenes. However, this approach significantly limits the method's generalizability due to the requirement for additional sensors, which is a major constraint. Most laparoscopic surgery environments cannot support the addition of precise pose sensors to the camera, thus this approach does not fundamentally solve the problem. Second, these approaches cannot handle dynamic scenes involving surgical instruments, as they focus on removing instruments to visualize the dynamic background. This limitation makes the generated images unsuitable for training neural networks for instrument-related tasks. However, instrument recognition is essential for advancing laparoscopic surgical robotics. Furthermore, these methods can only learn and temporally encode the observed deformations, and can thus not generate novel deformations outside of the training set. Additionally, while these methods can render images, they cannot generate corresponding annotated data, necessitating manual annotation before use in neural network training.

To address the aforementioned challenges, we propose a novel method for generating surgical image datasets that can synthesize images including instrument in novel camera view and tool deformation using dynamic 3D Gaussian reconstruction, as illustrated in Fig. 1. In our approach, we train a canonical Gaussian model to accommodate scene deformations in the canonical space and use a New Pose Synthesis (NPS) weights that deform the canonical instrument Gaussians into various poses. This technique not only tackles the challenge of reconstructing dynamic surgical scenes with instruments but also offers the ability to render Gaussians for unseen tool poses and jaw angles. We introduce a novel dynamic optimization strategy to mitigate the issues of inaccurate initial camera poses in real dynamic scenes , enabling our method to operate directly on any real dynamic scene dataset without the need for additional sensors to provide precise camera poses. Ultimately, through rendering, we can obtain realistic images and generate corresponding accurate annotations, thereby supplying reliable training data for downstream tasks. In our analysis, we compare the quality of our generated images with state-of-the-art image generation methods and use our synthetic frames to train surgical instrument detection and segmentation models. The results demonstrate that our method can produce high-quality images, and models trained on our synthetic data achieve performance comparable to those trained on real image data. Our contributions can be summarized as follows:

1.**Novel Dynamic Surgical Instrument Reconstruction Framework:** We propose an innovative framework for dynamic surgical instrument reconstruction that learns from previously observed instrument deformations. This framework can reconstruct instruments in dynamic surgical scenes and predict the view under tool movement, including unseen pose and position changes for instruments.

2.**Dynamic Adjustment Method for 3DGS Training:** We introduce a method for dynamically adjusting the 3DGS training process. By adopting different training strategies at various stages, our approach addresses the challenges posed by poor initialization due to inaccurate camera poses when using real-world scenes as input.

3.**Automatic Generation of Accurate Annotations:** The proposed method can automatically generate annotation information alongside the rendered images.

4.**Evaluation with Ground Truth (GT) Images:** We conducted additional experimental setups to obtain GT images that can be directly compared with the generated images. Through comparative experiments on image quality and neural network training using GT images as the benchmark, we ensure the reliability of our conclusions.

II. Relative Works

A. Surgical Scene Reconstruction

Numerous efforts have been made to reconstruct dynamic surgical scenes. These approaches can be categorized into two main types: implicit and explicit representation reconstruction.

1) Implicit Representation: Implicit representations, such as NeRF [11], have significantly advanced medical imaging. Unlike traditional methods that rely on spatial geometric information and the tracking of key deformation points for reconstruction, implicit representations use neural radiance fields and deformation fields to capture and represent scene deformation. This combination facilitates the effective reconstruction of dynamic scenes. Recently, EndoNeRF [14], inspired by dynamic NeRF [18], has emerged as a promising solution for dynamic surgical scene reconstruction. It uses tool-guided ray casting, stereo depth-cueing ray marching, and stereo depth-supervised optimization to achieve high-quality results, but suffers from lengthy training times. To address this, Forplane [19] optimizes training by conceptualizing surgical procedures as 4D volumes, decomposed into static and dynamic fields with orthogonal neural planes, reducing memory usage and accelerating optimization. However, these methods neglect the modeling of surgical instruments and produce non-editable, limited generalization scenes.

2) Explicit Representation: Explicit scene reconstruction methods, such as 3D Gaussian Splatting (3DGS) [12], overcome the limitations of implicit representation methods, which are difficult to edit. By manipulating obtained scene 3D Gaussian, it is possible to rotate and translate objects within the scene without sacrificing reconstruction quality. Additionally, these methods enable rapid training and real-time rendering of the reconstructed scenes.Similar to EndoNeRF [14], EndoGaussian [13] and EndoGS [20] use 3D Gaussians to represent surgical scenes. These methods process ordered images with continuous deformations over time, segmenting them into static scenes and introducing time-based deformation fields to stitch them together, reconstructing tissue deformations in dynamic surgical scenario. However, they face significant challenges: they struggle to accurately capture surgical instrument deformations, can only reconstruct past scenes without predicting future changes, and demand high-quality input data, including continuous temporal changes and precise camera poses.

B. Instrument Synthesis in Medical Imaging

Various methods generate surgical instrument images. Game engines and surgery simulators [21], [22] offer scalable, noise-free solutions but struggle to mimic real surface properties and textures accurately. Generative neural networks like GANs and Cycle-GANs [23], [24], [25], [26] create synthetic datasets resembling real image distributions. However, they cannot automatically generate corresponding annotations, requiring additional manual effort. Recent methods [27], [28], [29], [30], [31] integrate simulation environments with generative networks. Simulated medical images are enhanced using real image characteristics, producing high-quality, annotated synthetic datasets. However, these methods are designed for static images and struggle to alter the pose, orientation, and deformation of instrument end-effectors, limiting their applicability in dynamic surgical environments.

III. Methodology

A. Deformable Gaussian

Current dynamic 3DGS-based reconstruction methods typically decompose dynamic scenes into a series of consecutive static scenes based on temporal relationships [15]. These methods model the Gaussian representation of each static scene and then merge these representations to depict the dynamic scene. While this approach can represent dynamic scenes, it fails to accurately capture the deformation of Gaussians. Consequently, these methods are limited to representing only previously known scene deformations and cannot generate Gaussian representations for unseen scenarios. To truly understand and learn the inherent changes in scene deformation, it is crucial to grasp the dynamic alterations within the Gaussian representation of the scene.

Therefore, we introduce a deformable Gaussian model to more accurately describe scene deformation. The 3D Gaussians are initialized from point clouds generated by COLMAP [32], following the specified mathematical expression:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{1}$$

where μ denotes the mean value of the point cloud $P(x, y, z)$ and Σ is a 3D covariance matrix, expressed as $\Sigma = RSS^T R^T$. Here, R denotes a 3×3 rotation matrix, and S is a 3×3 diagonal matrix representing the scale. To simplify its representation, the rotation matrix R is converted into a vector r.

During initialization, the Gaussian is also assigned an opacity attribute σ, thus the 3D Gaussian is defined as:

$$G(\mu, r, s, \sigma) \tag{2}$$

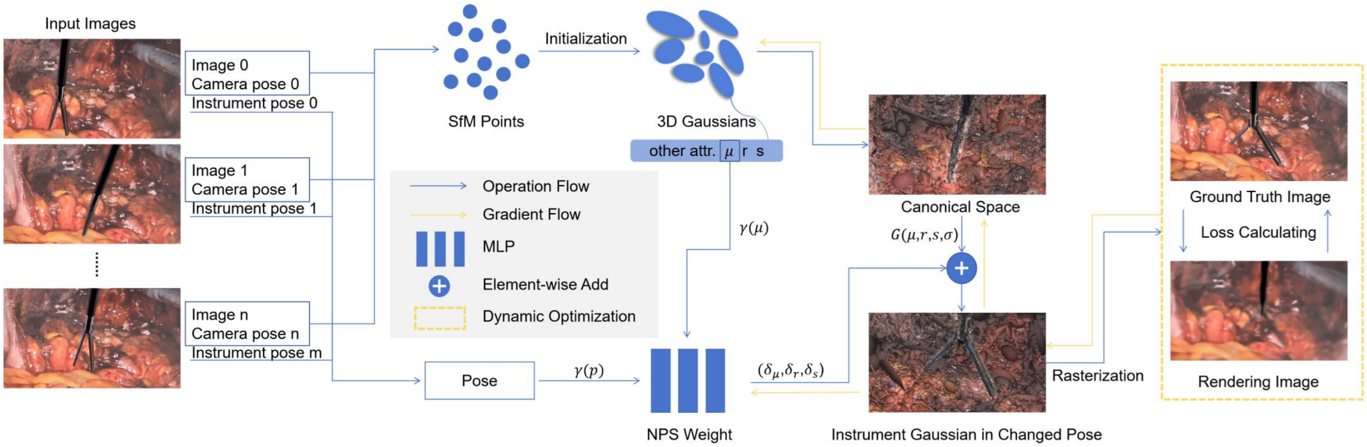


Fig. 2. **Training Process Overview.** Given a set of unordered images of a laparoscopic surgery procedure, our method represents the dynamic scene using 3D Gaussians and synthesizes the Gaussian representation of the instrument in a new pose from novel viewpoints. The standard 3D Gaussian representation of the scene is trained in a canonical space. An MLP is used to estimate the attribute changes during scene deformation, transforming the canonical Gaussians to the new deformation. The transformed Gaussians are then rendered using rasterization. The rendered images are compared with ground truth images to evaluate the training. The entire training process is optimized using a dynamic optimization strategy.

When the scene undergoes deformations, the modification of a Gaussian can be represented as $G'(\mu', r', s', \sigma')$. This reveals that updating only the attributes of the Gaussians in the deformation regions suffices to depict the scene's changes. Hence, we can express a scene's transformation using an incremental formulation:

$$G'(\mu', r', s', \sigma') = G(\mu + \delta\mu, r + \delta r, s + \delta s, \sigma + \delta\sigma) \quad (3)$$

To mitigate the impact of dynamic changes in the scene, we train this standard Gaussian model in the canonical space. To model these transformations effectively, if we can establish a mapping function $\mathcal{F}$ that inputs the scene deformation and current standard Gaussian, and outputs the Gaussian increments associated with these deformations, we can effectively represent the evolving scene. In this work, we introduce a learned New Pose Synthesis (NPS) weight as this mapping function. The NPS weight learns how the Gaussian representation of an instrument changes as it transitions from one pose to another, thereby enabling the prediction of the Gaussian representation in new poses. The core of these NPS weights is a Multi-Layer Perceptron (MLP) capable of predicting changes in the Gaussian properties. We set the depth of the MLP $D = 12$ and the dimension of the hidden layer $W = 256$.

The NPS weight takes two inputs: the current standard Gaussian attributes, μ, and the change parameters, p, to predict the incremental changes in the attributes u, r, and s. The μ parameter is derived from the XYZ coordinates of the point cloud. Motion in the instruments may be represented as a translation of these points, from which we derive μ. The change parameter p is constructed from recorded instrument poses, determined by quaternions that represent instrument rotation, and the opening angle of the tool jaw. In this way, the inputs μ and p encompass all seven degrees of freedom of possible movement in the scene (three each for rotation and translation, and one for the end-effector's operation). These processes can be summarized as:

$$(\delta\mu, \delta r, \delta s) = \mathcal{F}(\gamma(\mu), \gamma(p)) \quad (4)$$

where γ denotes the positional encoding, adapted from [11], which improves the training quality.

$$\gamma(\mu) = (\sin(2^k \pi\mu), \cos(2^k \pi\mu))_{k=0}^{L-1} \quad (5)$$

Where we set $L = 10$ for both μ and p. Note that during this process, we do not update the opacity σ. Opacity σ primarily affects the rendering process by determining the final rendered color. Since the color of the instrument typically does not change when its pose changes, we do not estimate the opacity. Nevertheless, σ of the Gaussian in the canonical space is still optimized during training.

B. Deformable Gaussian Training

The input is unordered images that capture a dynamic surgical scene with instrument deformation. Initially, we use structure from motion technique to calibrate the camera pose and generate a sparse point cloud that represents the scene, along with the change parameter p for each frame. The sparse point cloud is then initialized into Gaussians and transferred into the canonical space for training. As the scene transitions from between frames, the Gaussian attributes μ and p of the current frame are encoded and fed into the NPS weight, which attributes necessary to transform the Gaussians from the current frame to the next.

As training progresses, the NPS weight gradually learns to induce changes in the scene's Gaussian representation. As this is built from changes in μ and p between two frames, we can generate new frames by inputting arbitrary μ and p values. We further validate this ability to generate high-quality Gaussians representing unseen scene transformations in the experimental section IV. Following the 3DGS [12], we render the transformed scene's Gaussian $G(\mu + \delta\mu, r + \delta r, s + \delta s, \sigma)$ into an image. This rendered image is then compared with the

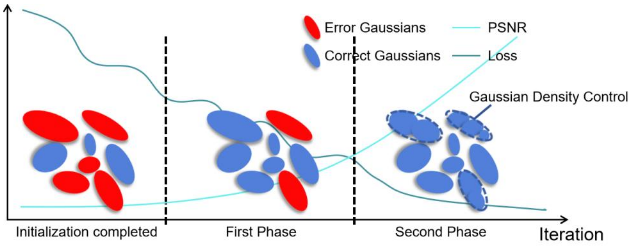


Fig. 3. **Dynamic Density Control.** In the first phase of training, the Gaussian distributions are newly initialized and contain numerous error points. During this phase, the loss function shows fluctuations, and the PSNR values are low but steadily increasing. In the second phase, the PSNR values generally exceed 20, and the loss function shows a consistent decline. The density control is gradually relaxed throughout the iterations, allowing for the splitting and cloning of Gaussian points to further enhance the training quality.

ground truth image of the transformed scene to calculate the loss function $\mathcal{L}$, a combination of $\mathcal{L}_1$ loss and a D-SSIM term:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}} \tag{6}$$

It is important to note that the rendering process depends on camera poses. As mentioned earlier, using real dynamic scenes often results in camera pose inaccuracies. Therefore, we assign a higher weight ($\lambda = 0.1$) to the $\mathcal{L}_1$ term. This is because SSIM (Structural Similarity Index Measure) evaluates structural information, contrast, and brightness, making it very sensitive to spatial relationships within the image. Any viewpoint discrepancies can lead to significant differences in corresponding structures, causing SSIM to drop sharply. By reducing the weight of the $\mathcal{L}_{\text{D-SSIM}}$ term, we minimize the impact of camera pose errors on the training outcome. After calculating the loss, we update the attributes of the Gaussians in the canonical space and also update all parameters of the NPS weight. Fig. 2 summarizes our training process.

C. Dynamic Training Adjustment

1) Dynamic Density Control Adjustment: Initialization errors from incorrect camera poses pose significant challenges to 3DGS training with dynamic real-world datasets. These errors introduce numerous erroneous points into the initialized Gaussian models, severely impacting 3DGS's density control mechanisms. Density control aims to enhance scene detail by cloning small Gaussians in sparse areas and splitting large ones in dense areas. However, when applied to erroneous Gaussians, it misplaces or redundantly multiplies error-prone Gaussians, exacerbating minor errors and causing training failure. While 3DGS can correct errors in static scenes, its robustness is compromised with dynamic datasets, as premature density control amplifies errors beyond its corrective capabilities.

Density control is predefined before training by three parameters: the densification interval P_{di}, the opacity reset interval P_{oi}, and a positional gradient threshold τ_{pos}. These fixed parameters do not accommodate the rapid dynamics of real-world scenes. Removing or delaying density control can mitigate these issues but may compromise the overall quality of the Gaussian representation, as density control is crucial for enhancing fidelity and detail.

To address this, we propose an adaptive density control strategy, illustrated in Fig. 3. We partition the training into two phases, guided by reductions in the loss function and improvements in the Peak Signal-to-Noise Ratio (PSNR) of rendered images, and dynamically adjust the density control during the training process.

In the first phase, Gaussians are newly initialized and often contain numerous errors. During this initial phase, we restrict density control to prioritize correcting these erroneous points to their accurate positions. We extend densification and opacity reset intervals and increase the gradient threshold, setting $P_{di} = 500$, $P_{oi} = 10,000$, and $\tau_{pos} = 0.0004$. Once the erroneous points are largely corrected in the first phase, the second phase commences as PSNR values exceed 20 and the loss function consistently declines. Here, we gradually reintroduce density control, allowing for strategic splitting and cloning of Gaussian points to enhance training quality. This phase effectively enhances geometric detail and refines overlapping areas, significantly improving model accuracy and robustness. Parameters for this phase are set to $P_{di} = 200$, $P_{oi} = 3000$, and $\tau_{pos} = 0.0002$.

2) Dynamic Spherical Harmonics Function Update: Similar to dynamic density control, we adopt an adaptive strategy for updating Spherical Harmonics (SH) coefficients. SH is effective for representing functions in 3D Gaussian, capturing illumination and details in complex scenes. Higher-order SH coefficients represent finer details but increase training complexity. In early training stages, we restrict SH updates to lower orders, focusing on correcting erroneous Gaussian initialization rather than complex details. This reduces computational overhead, allowing quicker convergence to a satisfactory state, primarily learning basic lighting and geometric structure. As training progresses, we gradually increase the order of SH updates to capture more complex details. In later stages, frequent and complex SH updates enable learning finer variations in lighting and scene details, improving overall performance. This strategy ensures initial stability and fully leverages higher-order SH's expressive power for high-precision scene reconstruction.

3) Uniform Motion Rendering: In the original 3DGS training process, input images are randomly selected, and the current Gaussian distribution is rendered based on the camera pose of the selected image, followed by a comparison with the image to compute the loss. However, since our input images include different poses of the instrument, there are significant variations in the instrument's appearance between two consecutive images. An instrument part visible in one frame might become the background in the next. Such substantial variations pose challenges for the NPS Weight to accurately predict Gaussian changes in the early stages of training. This method leads to a larger computed loss between the generated Gaussian distribution and the real image, thereby slowing down the convergence of the training process.

To address this issue, we introduce uniform motion simulation as shown in Fig. 4. In the first phase defined in 3, we sort the images according to the pose positions of the instrument, simulating a uniform motion with slow changes. This orderly variation is easier to learn, enabling the training process to converge more quickly. Once the training progresses

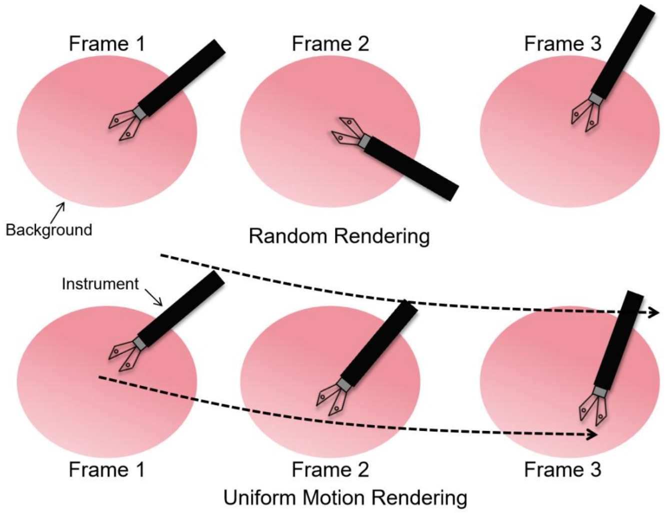


Fig. 4. **Uniform Motion Rendering:** The first row shows random rendering, where the instrument undergoes significant changes between frames. The second row shows uniform motion rendering, where the instrument's movement between frames is more consistent.

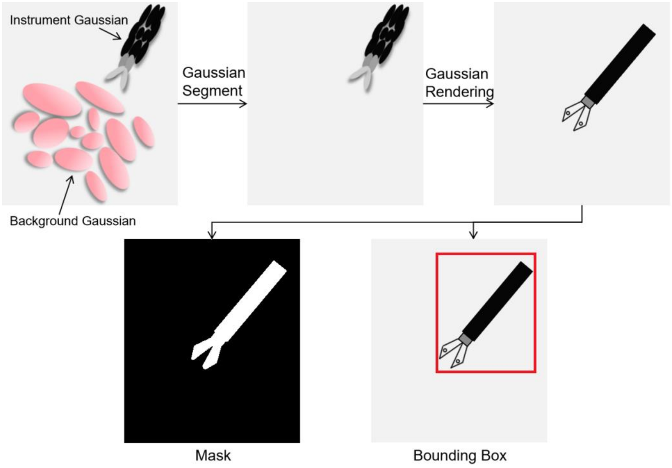


Fig. 5. **Annotation auto-generation:** First, the instrument Gaussians are extracted and rendered into 2D images. Based on the color differences between the foreground (instrument) and the background, the positions of the instrument pixels in the 2D image are identified, resulting in a mask and simultaneously generating a bounding box.

to the second phase, we revert to the random rendering strategy, allowing the model to adapt to larger variations in the instrument's poses. By this stage, the model has acquired a certain capability to handle smaller changes. Gradually introducing larger variations helps the NPS learn and adapt to complex changes of the instrument, thereby enhancing the overall performance and robustness of the model.

4) Dynamic Camera Pose Compensation: Due to inaccuracies in the input camera poses, training with imprecise poses may lead to overfitting on the training data. As mentioned in HyperNeRF [33], inaccurate camera poses in real-world datasets can cause spatial jitter between frames in the test or training set. This jitter can affect the rendering process, resulting in significant deviations between the rendered test images and the ground truth. To compensate for the rendering jitter caused by inaccurate camera poses, we designed a camera pose compensation mechanism. This compensation primarily occurs in the first phase of training:

$$(\delta\mu, \delta r, \delta s) = \mathcal{F}(\gamma(\mu), \gamma(p) + \Delta) \tag{7}$$

$$\Delta = (\mathcal{N}(0,1) - 0.5) \cdot \beta \cdot t_{phase} \tag{8}$$

where Δ represents the compensation, $\mathcal{N}(0,1)$ denotes the standard Gaussian distribution, and subtracting 0.5 adjusts the value range from $[0,1]$ to $[-0.5, 0.5]$. This adjustment allows for the simulation of camera pose errors in both positive and negative directions. The scaling factor β is empirically determined and has a value of 0.3. t_{phase} is a boolean value used to determine the current training phase; it is set to 1 during the first phase and 0 during the second phase.

D. Automatic Annotation Generation

To generate annotation files corresponding to surgical instruments, the first step is to segment these instruments from the background. For surgical tool Gaussians $G_I(\mu_I, r_I, s_I, \sigma_I)$, when deformation occurs, the new Gaussian representation is denoted as $G'_I(\mu_I + \delta\mu_I, r_I + \delta r_I, s_I + \delta s_I, \sigma_I))$. Conversely, for scene Gaussians $G_S(\mu_S, r_S, s_S, \sigma_S)$ that either remain unchanged or undergo minimal deformation, the NPS weight will generate little or no Gaussian increments, resulting in a new representation $G'_S(\mu_S + \delta\mu_S, r_S + \delta r_S, s_S + \delta s_S, \sigma_S))$, The increments $\delta\mu_I, \delta r_I, \delta s_I$ are significantly greater than $\delta\mu_S, \delta r_S, \delta s_S$. To identify significant changes, we experimentally establish a variation threshold $\mathcal{H}_{\delta\mu}, \mathcal{H}_{\delta r}, \mathcal{H}_{\delta s}$. If the increment in Gaussian attributes exceeds this threshold, we consider it indicative of substantial deformation, typically associated with the surgical tools in the scene. This method allows us to effectively segment the deformed surgical tool Gaussians from the rest of the scene.

We employ the differential Gaussian rasterization pipeline proposed by [12] to render the segmented instrument Gaussians. These 3D Gaussians are projected into 2D and rendered for each pixel using the following 2D covariance matrix Σ':

$$\Sigma' = JW\Sigma W^T J^T, \tag{9}$$

where J is the Jacobian of the affine approximation of the projective transformation, W is the view matrix transitioning from world to camera coordinates, and Σ denotes the 3D covariance matrix.

The color of the pixel on the image plane, denoted by $\mathbf{C}$, is calculated by α-blending the contributions of the N Gaussians, which are sorted from closest to farthest:

$$\mathbf{C} = \sum_{i \in N} \alpha_i c_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{10}$$

$$\alpha_i = \sigma_i e^{-\frac{1}{2}(\mu - u_i)^T \Sigma' (\mu - u_i)} \tag{11}$$

where c_i represents the color of each Gaussian along the ray, and u_i denotes the uv coordinates of the 3D Gaussians projected onto the 2D image plane.

Since we have segmented the Gaussians corresponding to the surgical instrument, the background is devoid of Gaussians and will consequently be rendered as black, with $\mathbf{C} = 0$.

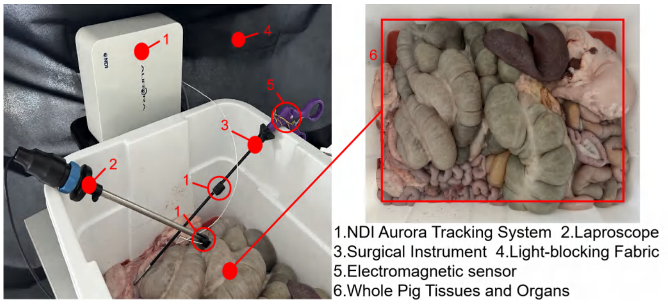

1
2
3
4
5
6
1.NDI Aurora Tracking System 2.Laproscope
3.Surgical Instrument 4.Light-blocking Fabric
5.Electromagnetic sensor
6.Whole Pig Tissues and Organs

Fig. 6. Data recording platform.

In the final 2D image, only the regions corresponding to the surgical instrument will have color, while the rest of the image will be close to black. We then set a threshold to differentiate between the black background (marked as background) and the colored instrument (marked as foreground), thus generating the corresponding 2D mask.

With this mask, we can automatically generate minimum bounding boxes, using the mask's outer contour, and bounding boxes, using the extremity pixel coordinates. This process is summarized in Fig. 5.

IV. Experiment

A. Data Recording

Due to the lack of publicly available datasets suitable for our method, which requires simultaneous camera movement and instrument deformation, we have constructed our experimental platform for data collection. Our experimental setup consists of: a laparoscope (Endoskill, MedEasy, China), laparoscopic instruments (Brand), an electromagnetic (EM) tracking system (Aurora, NDI, Canada), allowing 6-DoF motion tracking of the instrument and laparoscope. Additionally, we modified the instrument handle with a Hall-effect sensor to measure the jaw opening angle. The data recording platform is shown in Fig. 6.

During data collection, we used ex-vivo pig tissue and organs to create a more realistic surgical environment. Organs were harvested from pigs reared and slaughtered for the food chain. We collected data from various tissues and organs, including the liver, stomach, colon, and jejunum. These tissues and organs exhibit different visual characteristics such as color, texture, and shape, effectively simulating different surgical environments within the human body. During recording, we used the EM tracker and jaw angle sensor to capture the 7-DoF data of the surgical tools and the camera's 6-DoF pose. To achieve a more realistic simulation of real-world surgical scenarios, we exclusively utilized lighting sources from laparoscope.

B. Image Quality Experiment

In our image quality experiments, we focused on evaluating the images rendered by the proposed method (the last three images in Fig. 7).Initially, we rendered images with various backgrounds and identified the corresponding ground truth (GT) images for comparison. We selected three test datasets: Liver (Fig.7, first row) with diverse organs, fat, and tissue backgrounds; Bowel (second row) with uniform color but rich texture; and Stomach (third row) with uniform color and texture. Our method demonstrates the ability to predict the deformation of previously unseen surgical instruments. During training, we reserved 10% of the images as a test set, which did not participate in model training. During the rendering process, we input the 7-DoF data from the test set, allowing the model to predict and render these unseen deformations, and then compared them with the corresponding GT images (the first three images in each row of Fig.7).

We overlaid the rendered images with the GT images to generate difference maps, visualizing the discrepancies between them. As shown in Fig.7, the highlighted areas in the difference map indicate discrepancies, primarily located in the detailed regions of the surgical tool jaw and the image edges.

We selected several state-of-the-art (SOTA) methods for comparison, including 3DGS [12], NeRFies [34], 4DGS [15], and D-3DGS [16]. Due to the inability of 4DGS [15] and D-3DGS [16] to function properly on our collected real-world dataset (due to inaccurate camera poses), we incorporated the dynamic density control III-C.1 from our proposed method into these two methods to ensure they could operate on our dataset. The rendering results of all the comparison methods are visualized in Fig. 8. Following the evaluation methods utilized in [12], [34], [15], [16], we used photometric errors, including PSNR, SSIM, and LPIPS, as evaluation metrics for quantitative comparisons. The quantitative results on different datasets are summarized in Table I.

As shown in Table I, the rendered images from our proposed method outperform the SOTA methods across various evaluation metrics on different background datasets. Moreover, our method achieves satisfactory results even for unseen deformations. From Fig. 8, it can be observed that, although the proposed method exhibits some blurring in certain background regions, it excels in rendering surgical instruments, with both the main body and jaw rendered clearly and closely resembling the GT images.

3DGS [12] struggles to render surgical instruments as it finds it difficult to accurately represent static surgical tools within dynamic scenes. NeRFies [34] can render surgical instruments to some extent, but they are very blurry and hard to recognize. This is because NeRFies require high precision in the camera poses of input images, which is extremely challenging to calibrate accurately from real-world scenes, thus the inaccurate poses in the real dataset severely impact NeRFies' performance.

After incorporating dynamic density control, 4DGS [15] and D-3DGS [16] were able to operate on the real-world dataset, but their rendering quality was still inferior to our proposed method. The essence of 4DGS [15] and D-3DGS [16] is to divide a dynamic scene into numerous consecutive static scenes based on time t, represent each static scene, and then render these static scenes sequentially according to time t to synthesize the dynamic scene. This approach can only represent significant and slow dynamic changes in the scene, making it difficult to accurately render the rapid and subtle

TABLE I
QUANTITATIVE RESULT OF THE COMPARISON EXPERIMENT

	Liver			Stomach			Bowel		
Methods	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	18.01	0.708	0.643	17.21	0.701	0.553	18.33	0.722	0.631
NeRFies	23.12	0.772	0.493	22.17	0.763	0.441	21.71	0.714	0.512
4DGS	25.23	0.847	0.411	25.41	0.837	0.391	23.41	0.786	0.428
D-3DGS	24.01	0.811	0.462	23.32	0.803	0.422	24.68	0.835	0.431
Unseen Deformation	27.52	0.881	0.353	27.01	0.855	0.301	27.81	0.868	0.337
NeeCo	**28.88**	**0.902**	**0.273**	**29.81**	**0.893**	**0.274**	**29.87**	**0.913**	**0.281**

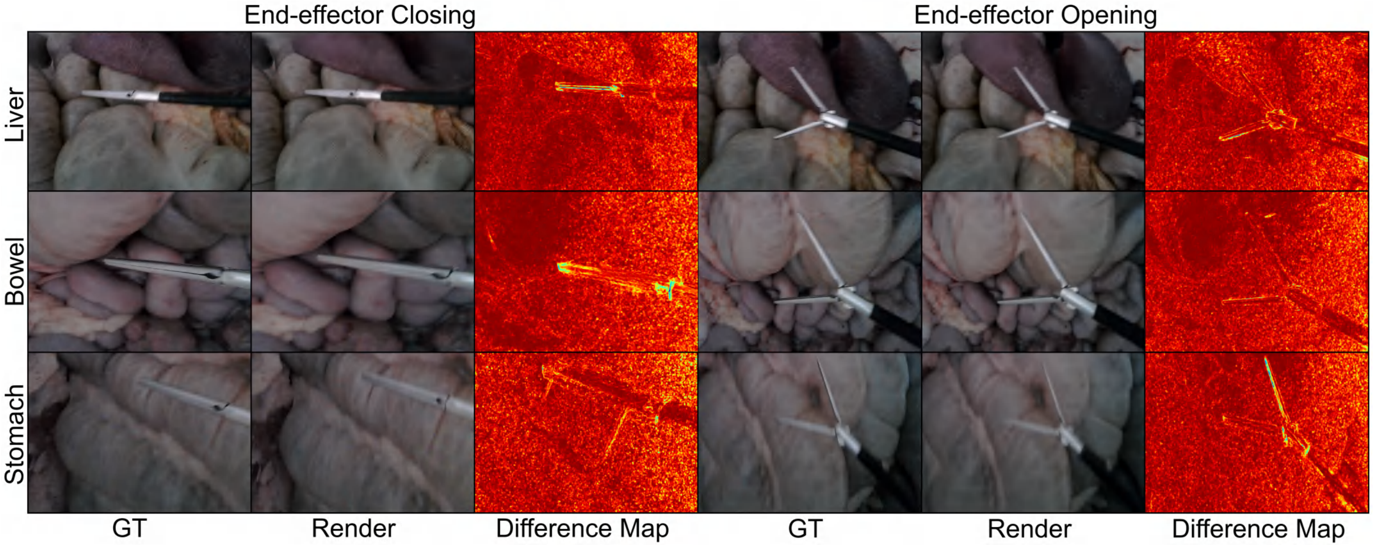


Fig. 7. Reconstruction result with difference maps. From left to right are the GT image, the rendered image, and the difference map. The difference map is created by overlaying the two images, with highlighted colors indicating areas of difference.

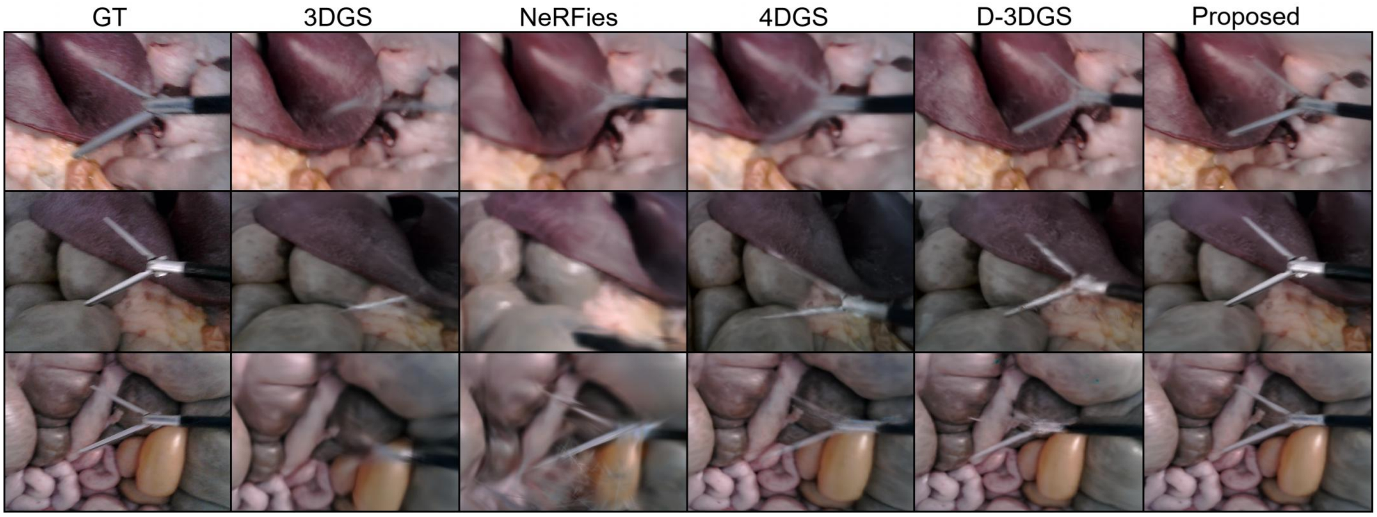


Fig. 8. Comparison of dynamic scene reconstruction using various methods. Each row represents the same training dataset, with the proposed method consistently showing more detailed reconstruction results, especially in the main body and jaw part of the surgical instruments.

deformations of surgical instruments commonly encountered in surgical scenes. Consequently, they struggle to clearly and accurately render parts of the surgical instruments that undergo rapid changes, such as the jaw.

In conclusion, our proposed method demonstrates superior performance in rendering surgical instruments with high clarity and detail, even in the presence of unseen deformations, compared to the existing SOTA methods. This advantage is particularly evident in dynamic surgical scenes, highlighting the robustness and effectiveness of our approach.

C. Neural Network Training Experiment

In our neural network training experiments, we validate the proposed method's effectiveness in using rendered images

TABLE II
DATA SOURCE COMPARISON ACROSS DIFFERENT MODEL TYPES (MEAN AND STANDARD DEVIATION)

Model	Metrics	REAL	SYNTHETIC	MIXED SYNTHETIC
YOLO	Precision ↑	0.703 ±0.007	0.694 ±0.013	0.776 ±0.011
	Recall ↑	0.831 ±0.003	0.826 ±0.011	0.901 ±0.015
U-Net	IoU ↑	0.617 ±0.008	0.601 ±0.011	0.683 ±0.012
	Dice ↑	0.763 ±0.004	0.751 ±0.009	0.812 ±0.013

for training neural networks. We focus on two common downstream tasks in RAMIS: object detection and semantic segmentation. Specifically, we use YOLOv5 [35] for object detection and U-Net [36] for semantic segmentation, both using our automatically generated labels.

During data acquisition, we collected an additional dataset to serve as a test set. This dataset contains 300 images featuring various backgrounds, instrument poses, and deformations. For YOLO and U-Net, we train three models from different source datasets: 1. real images with GT pose, 2. Synthetic rendering of the GT pose, and 3. a mix of rendered GT pose and new unseen poses. Each dataset contains 1780 images. All training sessions were executed on an NVIDIA GeForce RTX 4050(6G).

Both YOLO and U-Net are trained with their default parameters for 300 epochs. For YOLOv5, we use precision and recall as performance metrics, while for U-Net, we use IoU and Dice coefficients. To mitigate the effect of randomness on model performance evaluation, we conduct multi-folder experiments for each model.

Table II summarizes the performance of models trained with different image sets on the test dataset. For YOLO, we observe that the performance of the GT model and the Render model is very similar, with differences in Precision and Recall not exceeding 0.01. This indicates that neural networks trained with Render images can achieve performance comparable to those trained with GT images. The Augment model outperforms both the GT and Render models because it is trained with rendered images that include unseen deformations and different camera viewpoints, enhancing the diversity of the training dataset. This diversity allows the model to cover a wider range of deformations and scenarios during training, resulting in better performance.

For U-Net, we observe similar conclusions. The GT and Render models perform closely, while the Augment model demonstrates superior performance due to data augmentation. Notably, the Render and Augment models exhibit larger standard deviations compared to the GT model across all metrics. This is because rendered images lack the fine texture details of GT images, and background boundary rendering may introduce blurring, affecting overall performance and increasing standard deviations.

Overall, whether using YOLO or U-Net, the differences in performance metrics between the GT and Render models are less than 1.5%, while the Augment model shows nearly a 10% performance improvement compared to the GT model. This demonstrates that our synthetic images can not only train neural networks effectively, but alsoenhances the dataset by generating novel viewpoints and tool locations, improving training performance compared to GT images alone.

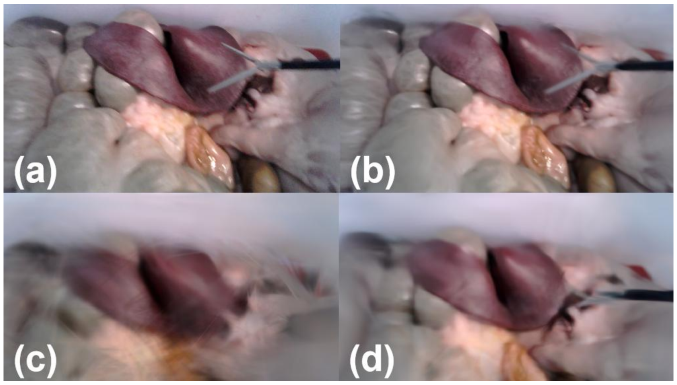


Fig. 9. Ablation study results: (a) GT image, (b) results with both Dynamic Density Control and Uniform Motion Rendering applied, (c) without Dynamic Density Control, (d) without Uniform Motion Rendering.

V. Ablation Study

Dynamic Training Adjustment is crucial in our method to address the challenges posed by poor camera pose estimates. We conducted ablation studies to validate the contributions of various training strategies within the Dynamic Training Adjustment framework. Specifically, Dynamic Density Control and Uniform Motion Rendering directly influence the rendering quality, so we visualized the impact of these modules on our proposed method. As shown in Fig. 9, the training without Dynamic Density Control (c) results in extremely poor rendering quality, while the absence of Uniform Motion Rendering (d) fails to accurately render the surgical instruments. In fact, without Dynamic Density Control, the model is likely to encounter training failures. We conducted multiple experiments on the Liver, Bowel, and Stomach datasets, recalibrating the camera poses for each input image using COLMAP with different parameter settings to obtain more accurate poses. However, we found that without Dynamic Density Control, the model had nearly an 80% chance of failing to complete the training, regardless of the COLMAP settings. This underscores the importance of this method in our study.

VI. Conclusion

This paper presents a novel pipeline for generating surgical instrument deformation images, contributing to the creation of realistic and diverse surgical image datasets compared to existing methods. Our approach introduces dynamic 3D Gaussian models to represent the deformation of instruments in dynamic surgical scenes and employs a dynamic density control strategy to address the challenges posed by poor camera poses in real-world datasets, which often hinder training. Our contributions include a novel surgical instrument image generation framework capable of rendering images from new viewpoints and unseen deformations, as well as a dynamic training adjustment strategy that enhances the applicability of Gaussian splatting-based methods on real-world datasets. Additionally, our method can generate accurate annotation files, addressing the significant challenge of the lack of annotated data in medical imaging datasets. Our experiments demonstrate promising results, outperforming recent work and achieving object detection and segmentation performances that closely resemble those of models trained on GT imaging. Moreover, the datasets generated using our method's capability to render new deformations and viewpoints further surpass the performance of models trained solely on GT imaging. By leveraging the dynamic nature of our approach, we address significant limitations in current methodologies, paving the way for more effective training and application of neural networks in automation surgery.

References

[1] L. El Jiani, S. El Filali, and E. H. Benlahmer, "Overcome medical image data scarcity by data augmentation techniques: A review," in 2022 International Conference on Microelectronics (ICM), 2022, pp. 21–24.

[2] A. K. Upadhyay and A. K. Bhandari, "Advances in deep learning models for resolving medical image segmentation data scarcity problem: A topical review," Archives of Computational Methods in Engineering, vol. 31, no. 3, pp. 1701–1719, 2024.

[3] A. J. Lungu, W. Swinkels, L. Claesen, P. Tu, J. Egger, and X. Chen, "A review on the applications of virtual reality, augmented reality and mixed reality in surgical simulation: an extension to different kinds of surgery," Expert review of medical devices, vol. 18, no. 1, pp. 47–62, 2021.

[4] Y. Liu, Y. Tian, G. Maicas, L. Z. C. T. Pu, R. Singh, J. W. Verjans, and G. Carneiro, "Photoshopping colonoscopy video frames," in 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, 2020, pp. 1–5.

[5] G. Luegmair, D. D. Mehta, J. B. Kobler, and M. Döllinger, "Three-dimensional optical reconstruction of vocal fold kinematics using high-speed video with a laser projection system," IEEE transactions on medical imaging, vol. 34, no. 12, pp. 2572–2582, 2015.

[6] N. Mahmoud, T. Collins, A. Hostettler, L. Soler, C. Doignon, and J. M. M. Montiel, "Live tracking and dense reconstruction for handheld monocular endoscopy," IEEE transactions on medical imaging, vol. 38, no. 1, pp. 79–89, 2018.

[7] L. Maier-Hein, A. Groch, A. Bartoli, S. Bodenstedt, G. Boissonnat, P.-L. Chang, N. T. Clancy, D. S. Elson, S. Haase, E. Heim et al., "Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction," IEEE transactions on medical imaging, vol. 33, no. 10, pp. 1913–1930, 2014.

[8] L. C. Garcia-Peraza-Herrera, L. Fidon, C. D'Ettorre, D. Stoyanov, T. Vercauteren, and S. Ourselin, "Image compositing for segmentation of surgical tools without manual annotations," IEEE transactions on medical imaging, vol. 40, no. 5, pp. 1450–1460, 2021.

[9] F. Fuentes-Hurtado, A. Kadkhodamohammadi, E. Flouty, S. Barbarisi, I. Luengo, and D. Stoyanov, "Easylabels: weak labels for scene segmentation in laparoscopic videos," International journal of computer assisted radiology and surgery, vol. 14, pp. 1247–1257, 2019.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," Communications of the ACM, vol. 65, no. 1, pp. 99–106, 2021.

[12] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," ACM Transactions on Graphics, vol. 42, no. 4, pp. 1–14, 2023.

[13] Y. Liu, C. Li, C. Yang, and Y. Yuan, "Endogaussian: Gaussian splatting for deformable surgical scene reconstruction," arXiv preprint arXiv:2401.12561, 2024.

[14] Y. Wang, Y. Long, S. H. Fan, and Q. Dou, "Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery," in International conference on medical image computing and computer-assisted intervention. Springer, 2022, pp. 431–441.

[15] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 20 310–20 320.

[16] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 20 331–20 341.

[17] Y. Huang, B. Cui, L. Bai, Z. Guo, M. Xu, and H. Ren, "Endo-4dgs: Distilling depth ranking for endoscopic monocular scene reconstruction with 4d gaussian splatting," arXiv preprint arXiv:2401.16416, 2024.

[18] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10 318–10 327.

[19] C. Yang, K. Wang, Y. Wang, X. Yang, and W. Shen, "Neural lerplane representations for fast 4d reconstruction of deformable tissues," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2023, pp. 46–56.

[20] L. Zhu, Z. Wang, Z. Jin, G. Lin, and L. Yu, "Deformable endoscopic tissues reconstruction with gaussian splatting," arXiv preprint arXiv:2401.11535, 2024.

[21] J. Moore, H. Scheirich, S. Jadhav, A. Enquobahrie, B. Paniagua, A. Wilson, A. Bray, G. Sankaranarayanan, and R. B. Clipp, "The interactive medical simulation toolkit (imstk): an open source platform for surgical simulation," Frontiers in Virtual Reality, vol. 4, p. 1130156, 2023.

[22] S. Grossi, M. Cattoni, N. Rotolo, and A. Imperatori, "Video-assisted thoracoscopic surgery simulation and training: a comprehensive literature review," BMC medical education, vol. 23, no. 1, p. 535, 2023.

[23] K. Lee, M.-K. Choi, and H. Jung, "Davincigan: Unpaired surgical instrument translation for data augmentation," in International Conference on Medical Imaging with Deep Learning. PMLR, 2019, pp. 326–336.

[24] M. Sahu, R. Strömsdörfer, A. Mukhopadhyay, and S. Zachow, "Endo-sim2real: Consistency learning-based domain adaptation for instrument segmentation," in International conference on medical image computing and computer-assisted intervention. Springer, 2020, pp. 784–794.

[25] M. Sahu, A. Mukhopadhyay, and S. Zachow, "Simulation-to-real domain adaptation with teacher–student learning for endoscopic instrument segmentation," International journal of computer assisted radiology and surgery, vol. 16, no. 5, pp. 849–859, 2021.

[26] Z. Zhang, B. Rosa, and F. Nageotte, "Surgical tool segmentation using generative adversarial networks with unpaired training data," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6266–6273, 2021.

[27] F. Mahmood, R. Chen, and N. J. Durr, "Unsupervised reverse domain adaptation for synthetic medical images via adversarial training," IEEE transactions on medical imaging, vol. 37, no. 12, pp. 2572–2581, 2018.

[28] E. Colleoni, D. Psychogyios, B. Van Amsterdam, F. Vasconcelos, and D. Stoyanov, "Ssis-seg: Simulation-supervised image synthesis for surgical instrument segmentation," IEEE Transactions on Medical Imaging, vol. 41, no. 11, pp. 3074–3086, 2022.

[29] D. Rivoir, M. Pfeiffer, R. Docea, F. Kolbinger, C. Riediger, J. Weitz, and S. Speidel, "Long-term temporally consistent unpaired video translation from simulated surgical 3d data," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 3343–3353.

[30] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 172–189.

[31] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 560–567, 2023.

[32] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113.

[33] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," arXiv preprint arXiv:2106.13228, 2021.

[34] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5865–5874.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer, 2015, pp. 234–241.