AN AUTONOMOUS AREA COVERAGE METHOD FOR ENDOSCOPIC CAPSULE ROBOTS

by

İbrahim Ömer Çelik

B.Sc. in Computer Engineering, Bilkent University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2021

ABSTRACT

AN AUTONOMOUS AREA COVERAGE METHOD FOR ENDOSCOPIC CAPSULE ROBOTS

Endoscopic examination with actively steerable capsule robots is an emerging technology for the diagnosis of various cancer types and many other gastrointestinal diseases such as Crohn's disease, ulcerative colitis, and hemorrhage. Ensuring a comprehensive screening with these robotic capsules is of paramount importance in the detection of such critical diseases. In this work, we propose a novel autonomous area coverage method for active capsule endoscopes, which aims to maximize the amount of monitored area in the human gastrointestinal tract in a time-efficient manner. We introduce a simulation environment and train a deep reinforcement learning model that learns to autonomously navigate magnetically-driven capsule robots for the area coverage task based on visual feedback provided by an on-board monocular camera. Under both physically and visually realistic capsule endoscopy circumstances our method performs successful reasoning and outperforms baseline coverage path planning approaches on human stomach organ models. Besides, we develop both localization and depth estimation methods for capsule robots by jointly training multiple convolutional neural networks in a self-supervised fashion through utilizing large-scale synthetic endoscopy data recorded in the simulation environment. Numerical assessments of the pose estimation network are presented in comparison with similar studies and the depth estimation method is qualitatively assessed on real endoscopy images and dense surface reconstruction task. Results demonstrate that the proposed learning-based coverage path planning approach in company with the monocular localization and surface reconstruction methods have the potential to become key software elements of the current and next-generation capsule robots and to pave the way for the development of future endoscopic systems.

ÖZET

ENDOSKOPİK KAPSÜL ROBOTLARDA OTONOM ALAN TARAMA

Aktif olarak kontrol edilebilen kapsül robotlar, endoskopik incelemelerde kullanılmak üzere geliştirilen yeni bir teknolojidir. Bu teknoloji sayesinde sindirim sisteminde meydana gelen çeşitli kanser türleri, Crohn hastalığı, ülseratif kolit ve kanama gibi birçok mide ve bağırsak hastalıkları teşhis edilmektedir. Bu robotik kapsüller ile kapsamlı bir taramanın sağlanması, bu tür kritik hastalıkların tespiti acisindan büyük önem arz etmektedir. Bu çalışmada, aktif kapsül robotlar için insan gastrointestinal yolunu en yüksek seviyede goruntulemeyi amaclayan bir otonom alan tarama yöntemi önerilmektedir. Çalışma kapsamında geliştirilen simülasyon ortamı sayesinde, kapsül robotları manyetik alan ile otonom olarak yönlendirebilmeyi öğrenen bir derin takviyeli öğrenme modeli eğitilmiştir. Metodumuzun hem fiziksel hem de görsel olarak gerçekçiliği sağlanan kapsül endoskopi koşullarında başarılı bir şekilde hareket edebildiği ve temel alan tarama yöntemlerine kıyasla daha iyi sonuçlar verdiği gösterilmiştir. Ayrıca, kapsül robotlara yer tayini ve üç boyutlu haritalama yeteneklerini kazandırmak için bir görsel odometri ve derinlik öğrenme yaklaşımı sunulmaktadır. Bu yaklaşımda, simülasyon ortamında elde edilen sentetik endoskopi verileri kullanılarak birden fazla evrişimli sinir ağı herhangi bir referans verisine ihtiyaç duymaksızın eğitilmiştir. Eğitilen bu modellerin hem sayısal hem de niteliksel değerlendirmeleri poz ve derinlik kestirimi ile beraber üç boyutlu yüzey oluşturma işlemi üzerinde sunulmuştur. Sonuçlar, önerdigimiz yer tayini ve haritalama yöntemlerinin eşlik ettiği öğrenmeye dayalı otonom alan tarama konseptinin, mevcut ve gelecek nesil kapsül robotların önemli yazılım bileşenleri olması ve ileride endoskopik sistemlerin geliştirilmesinin önünü açması bakımından yüksek bir potansiyele sahip olduğunu göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF SYMBOLS . . . xiii
LIST OF ABBREVIATIONS . . . xv
1. INTRODUCTION . . . 1
1.1. Contributions . . . 5
1.2. Organization of the Thesis . . . 6
2. RELATED WORK . . . 7
2.1. Endoscopic Localization Mapping and Path Planning . . . 7
2.2. Robotic Area Coverage . . . 8
3. SIMULATION . . . 10
3.1. Simulation Overview . . . 10
3.2. Generating 3D Organ Models . . . 11
3.3. Endoscopic Camera View . . . 12
3.4. Modeling Magnetic Interaction . . . 14
3.5. Modeling Intestinal Friction . . . 14
4. PROPOSED METHOD . . . 16
4.1. Area Coverage Path Planning . . . 16
4.1.1. Policy Learning . . . 17
4.1.2. Proposed Policy . . . 18
4.1.3. Reward Function . . . 21
4.1.4. Policy Optimization . . . 22
4.1.5. Policy Network Architecture . . . 23
4.2. Visual Odometry and Depth Learning . . . 25
4.2.1. Training Paradigm . . . 26
4.2.2. Network Architecture . . . 29

4.3. Dense Surface Reconstruction 31
5. EVALUATION 33
5.1. Area Coverage 33
5.2. Dataset 38
5.3. Pose Estimation Results 39
5.4. Depth Estimation Results 41
5.5. Surface Reconstruction Results 42
6. CONCLUSION 44
REFERENCES 46

LIST OF FIGURES

Figure 1.1. The high-level concept of autonomous navigation with a cognition based active path planning strategy for magnetically actuated capsule robots. The illustration is inspired from the see-think-act concept proposed by Siegwart *et. al.* for autonomous mobile robots [10]. 2

Figure 3.1. The illustration of synthetic images recorded in the simulation environment compared to the real endoscopic scenes for each gastrointestinal organ. 11

Figure 3.2. The process of generating realistic 3D gastrointestinal organ models used in the simulation. Computed Tomography (CT) images are first reconstructed, then the organs are extracted from the whole body via post-processing. Finally, generated flat textures are wrapped into each organ with the UV mapping technique. . . 12

Figure 3.3. Real-time post-processing effects applied on a sample image frame recorded inside a 3D colon model. These effects are used to mimic the important characteristics of real capsule endoscopy scenes. . . 13

Figure 4.1. Autonomous area coverage policy learning pipeline for endoscopic capsule robots. The policy π_c receives camera observations, the position and the velocity of the capsule, as well as the position of the magnetic actuator and outputs an action vector for planning the motion of the actuator. A built-in mapper functionality generates a real-time 3D sparse map in the simulation which provides a reward for the produced action based on the amount of covered areas. . . 19

Figure 4.2. Policy gradient optimization algorithm. 23

Figure 4.3. Deep neural network architecture for the proposed area coverage policy π_c. Visual and vector observations are encoded in parallel by a convolutional and a fully-connected networks respectively. The outputs of these networks are concatenated and then mapped into a three-dimensional action vector. 24

Figure 4.4. Representation of convolution and residual blocks used in the ResNet encoder architecture. 25

Figure 4.5. The demonstration of self-supervised training paradigm to estimate camera pose and depth information from monocular capsule endoscopy video sequences. Consecutive image frames I_t, I_{t-1} are fed into the two identical encoder-decoder CNN DispNet [59] to obtain the corresponding depth frames, and an encoder network PoseNet [22] takes the concatenation of the frames and predicts relative camera pose $P_{t\to t-1}$. a) Geometric consistency loss L_{GC} are computed by the geometric difference between depth frames that are represented on the same projection. b) Photometric loss L_p and the point-wise difference between I_t and its projected coordinates into I_{t-1}. 27

Figure 4.6. Algorithm for computing the geometric consistency loss. 28

Figure 4.7. Algorithm for computing the photometric loss. 29

Figure 4.8. Architectural details of the proposed networks a) the DispNet, b) PoseNet. The rectangular blocks represent the input dimensions after each convolution (blue) or deconvolution (red) operation is applied. The DispNet produces a same dimensional matrix with the input frame, which provides pixel-wise correspondence, and the PoseNet produces a 1x6 vector that represents 6 dof camera pose. 30

Figure 4.9. The flow diagram of the proposed dense surface reconstruction method. Using the pre-trained DispNet and the PoseNet, corresponding depth maps and pose values are estimated for each image in a monocular endoscopic video sequence. Then these images are fused with their depths by a surfel-based reconstruction method Elasticfusion. This flow provides a real-time 3D dense surface reconstruction experience using a single GPU during the capsule endoscopy procedure. 31

Figure 4.10. Dense surface reconstruction algorithm. 32

Figure 5.1. The overview of the experimental setup for training the proposed area coverage policy. Top-left: experimental setup in the simulation environment. Top-right: capsule camera view. Bottom: mapper view showing covered (green) and uncovered areas (red) of the organ on a 3D sparse map shown from different perspectives. The arrow and the black line represent the current position and the trajectory of the capsule respectively. 34

Figure 5.2. Learning curve for the proposed area coverage policy trained on a stomach model. Bold line and shaded areas depict the mean and standard deviation of the reward. 35

Figure 5.3. The area coverage status shown at 10 time steps during the policy inference process. Each time step t is 15 seconds. The inference results for a) the training stomach model and b) the two different test stomach models with different size and geometry are displayed. 36

Figure 5.4. Plots showing the comparison of coverage results in percentage for stomach organ models as the time progresses. 38

Figure 5.5. Pose estimation results for test stomach, small intestine and colon datasets. True and predicted trajectory plots are comparatively presented for each method. Trajectory lengths are also provided. . 40

Figure 5.6. Predicted depth images on synthetic test datasets for each gastrointestinal organ compared to their ground-truth counterparts. Depth images are represented in jet color map ranging from 0 to 1 (nearest to farthest) for better visualization. 42

Figure 5.7. Depth estimation results on real endoscopic images. The depth estimation network trained only on the synthetic data is tested on real endoscopy datasets for the stomach small intestine and colon. 43

Figure 5.8. Qualitative evaluation of the surface reconstruction method on a sample set of stomach and colon test datasets. The reconstruction results using the estimated depth and the pose values (top) are visually compared to the results when ground-truth depth values are used (bottom). 43

LIST OF TABLES

Table 5.1. The impact of reward weight parameters in model performance. . . 35

Table 5.2. Comparison of coverage path planning results. 38

Table 5.3. Comparison of pose estimation results. 41

LIST OF SYMBOLS

$\vec{a}_t$	Action vector at time t
$\vec{c}_t$	Capsule state vector at time t
c_x, c_y	Optical center
C_t	Cumulative area coverage until time t
C_{thresh}	Threshold value for cumulative area coverage
D_t	Depth map at time t
D_{t-1}^t	Depth map at time t projected on an image plane at $t-1$
$\vec{F}$	Electromagnetic force
$f(\vec{v})$	Velocity dependant frictional resistance function
$\vec{f}_{coulomb}$	Coulomb frictional resistance
$\vec{f}_{environment}$	Environmental frictional resistance
$\vec{f}_{visco}$	Visco-adhesive frictional resistance
f_x, f_y	Focal length
I_t	RGB input frame at time t
$J(\pi_\theta)$	Policy objective function
L	Total loss
L_{GC}	Geometric consistency loss
L_p	Photometric loss
$\vec{m}$	Magnetic dipole vector
$\vec{m}_t$	Magnet state vector at time t
o_t	Visual observations at time t
p_t	Homogeneous coordinates of an input frame at time t
$P_{t\to t-1}$	Relative pose from an input frame at time t to $t-1$
r_{ct}	Area coverage based reward at time t
$\vec{r}$	Distance between two magnetic poles
r_{vt}	Velocity based reward at time t
s	Lens skew coefficient
S_t	Environment state at time t
$\vec{v}_{max}$	Capsule maximum velocity

$\vec{v}_t$	Capsule velocity
w_1, w_2	Reward weight parameters
α	Constant penalty value in reward function
β	Scale parameter in reward function
Δ	Change in relative pose
γ	Scale parameter for velocity penalty
γ_{visco}	Viscosity coefficient
μ	Mean
μ_0	Free space permeability
∇	Gradient
∇_θ	Gradient of the parameter θ
π	Pi
π_c	Area coverage policy
$\pi(a_t\|S_t)$	Probability that policy takes action a_t given S_t at time t
q_m	Magnetic charge of a dipole m
σ	Standard deviation
θ	Policy parameters

LIST OF ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
ATE	Absolute Trajectory Error
CNN	Convolutional Neural Networks
CT	Computed Tomography
DispNet	Depth Estimation Network
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
EKF	Extended Kalman Filter
GAN	Generative Adversarial Networks
GI	Gastrointestinal
PoseNet	Pose Estimation Network
PPO	Proximal Policy Optimization
R-CNN	Recurrent Convolutional Neural Network
ResNet	Residual Networks
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RPE	Relative Pose Error
RPE_{rot}	Relative Pose Error in rotation
RPE_{trans}	Relative Pose Error in translation
SLAM	Simultaneous Localization and Mapping
TCIA	The Cancer Imaging Archive
UGI	Upper gastrointestinal
VO	Visual Odometry

1. INTRODUCTION

The number of estimated deaths caused by digestive system cancers is the highest among all the other cancer types according to a study conducted in 2019 [1]. Routine endoscopic screenings play a significant role in the early diagnosis and treatment of such cancer types as well as other critical gastrointestinal (GI) diseases such as Crohn's disease, Celiac disease, ulcerative colitis, and lesions. However, medical studies demonstrate that 11.3% of the upper gastrointestinal (UGI) cancer and 8% of the colorectal cancer patients had false-negative results in such routine screenings [2, 3]. Another study reports that 38% of the patients in the United States had at least one colorectal lesion missed in colonoscopic assessments [4]. These statistics indicate that current endoscopic and colonoscopic monitoring practices have a poor performance in detecting serious gastrointestinal disorders. Therefore, the objective of an automated scanning that maximizes the amount of recorded area in the human digestive tract is crucial in terms of ensuring an accurate and sensitive diagnostic protocol with endoscopic systems.

Such automated scanning is possible with actively and remotely controllable Wireless Capsule Endoscopes (WCEs) that are pill-sized, swallowable robotic devices used for imaging the entire gastrointestinal tract of patients. In recent years, researchers have made significant progress in developing active locomotion systems for these endoscopes (i.e. internal motors or electromagnetic steering) and proposed advanced localization, mapping, and path realization techniques to have guided navigation through the digestive tract and also to enable some diagnostic and therapeutic functions such as biopsy and targeted drug delivery [5, 6, 7, 8]. However, the poor disease detection rate problem with current endoscopic practices has not been considered in previous works. In this study, we address this problem for the first time in the literature by developing a fully automated area coverage path planning methodology for capsule endoscopes. In particular, we introduce a highly realistic simulation tool to realize the magnetically-driven capsule endoscopy procedure inside real human organ models and propose a cognition based path planner for an external magnetic actuator that aims

to maximize the monitored areas in a minimum possible operation time. The details of the simulation environment including a brief section for the proposed area coverage path planning method has also been introduced in our joint work VR-Caps [9].

Figure 1.1, inspired by the see-think-act concept [10], illustrates the proposed autonomous control and active navigation strategy for magnetically-driven capsule endoscopes at a high-level. In this strategy, a steady magnet, conceptually carried by a

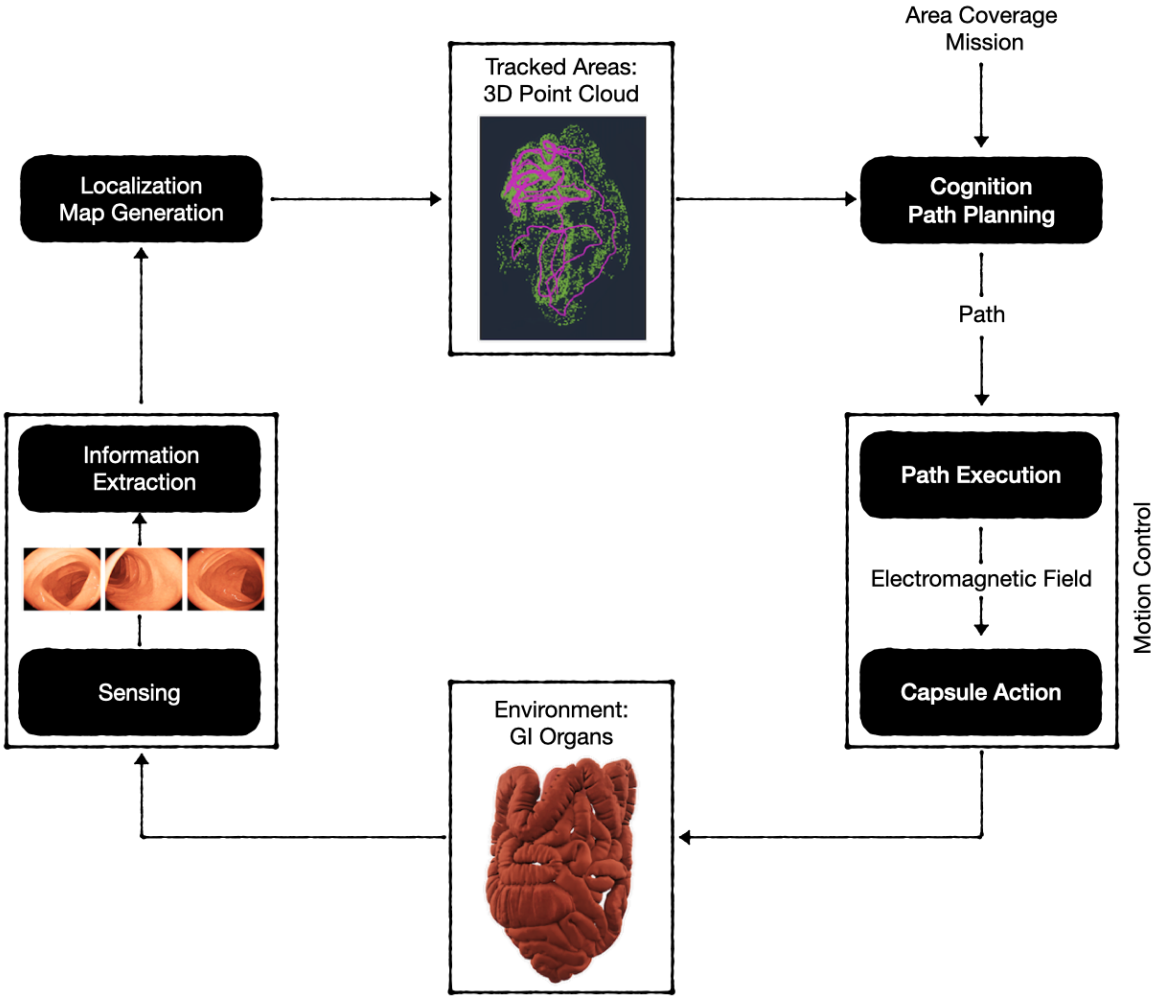


Figure 1.1: The high-level concept of autonomous navigation with a cognition based active path planning strategy for magnetically actuated capsule robots. The illustration is inspired from the see-think-act concept proposed by Siegwart *et. al.* for autonomous mobile robots [10].

robotic arm, can navigate a capsule robot inside the gastrointestinal organs through the electromagnetic field by moving the magnet over a patient body. The motion of

the magnet is based on the action inputs provided by a path planner that is trained for the area coverage objective. Using the camera feedback, a sparse 3D mapping functionality is implemented to keep track of the capsule, and monitored areas in the simulation. The generated map is also used for supervising the path planning model during the training. Then, all the sensory information is fed into the path planner, where it performs a cognition based decision making from any state upon the completion of the area coverage mission. This way, we achieve autonomy for endoscopic capsule robots and aim for increasing the accuracy and the sensitivity of diagnosing critical gastrointestinal disorders with the proposed coverage path planning method. In this work, we focus on realizing this closed-loop area coverage targeted autonomous navigation concept under a simulated circumstances. In particular, we address the problems of cognition path planning, path execution, and capsule navigation through electromagnetic steering. We also study the sensor-based localization problem in the scope of this work since we need to accurately predict the capsule location in order to apply the proposed concept to real-world scenarios.

In the context of active capsule endoscopes, developing robotic control systems accompanied with perception-related localization and mapping techniques often requires burdensome environment setups such as using ex-vivo animal organs or high-cost physical endoscopy simulators as test-beds [11, 12]. Testing and optimization of such algorithms is also a time consuming process in such physical setups as it requires an extensive experimentation phase under several different configurations. To address this challenge, prior works have developed simulation tools for active capsule endoscopes. Mura *et. al.* have proposed a vision-based haptic feedback system to control and assist the navigation of capsules [13]. Abu-kheil *et. al.* have introduced a simulation system focusing on the locomotion mechanism and path tracking functions for capsule robots [14]. Although there have been simulation systems proposed for WCEs, they have been limited to one aspect of the overall capsule endoscopy procedure. In this work, we address the lack of a comprehensive and realistic test-bed by proposing the virtual environment that simulates a wide variety of aspects under both physically and visually realistic circumstances such as varied organ types, several capsule endoscope designs (i.e. types and placement of sensors), and the strength, and position of internal

and external magnetic actuators as the source of active control and locomotion.

Moreover, a robust localization and mapping approach for handling real-world tasks is crucial since a robot needs to continuously localize itself based on sensory feedback and sends its state to the path planner while navigating in an unknown environment. Recently, deep learning techniques have achieved a great success in perception-related robotic tasks with the availability of large, and well-represented datasets and these techniques have also been widely adopted in the endoscopy field. However, acquiring such datasets particularly in biomedical domain is challenging due to privacy and ethical concerns. To overcome this challenge, some works have recorded data inside an ex-vivo porcine stomach and train deep neural networks for predicting the pose of the capsule robot [12]. Another work proposed an adversarial training approach that produces synthetic data from the distribution of real endoscopy data which yields an increase in the performance of a deep neural network for the depth estimation task [15]. In this work, we introduce a synthetically generated photo-realistic endoscopy dataset and enable the possibility of generating many more datasets under various configurations (i.e. lens and organ types, cross-patient data, tissue variations) using our simulation environment. By using this large sequential endoscopy dataset we jointly train multiple convolutional neural networks (CNN) to enable a domain specific localization and depth estimation capability for capsule robots. In fact, the models predict the position and the orientation of a capsule robot and the depth of the scene based on the visual feedback acquired from an associated monocular camera.

Another important task for endoscopic systems is to densely reconstruct the surface from monitored areas. The 3D surface of a well-scanned organ could help gastroenterologists to easily spot and to localize diseases rather than examining long (approximately 7 to 8 hours) video records [16]. Similarly, a reliable 3D map of the inspected organ could also assist surgeons to plan the required treatment procedures on a holistic view. Therefore, several studies have developed dense surface reconstruction methods in the endoscopy field with either the availability of ground-truth depth data [15] or using rule-based estimation techniques (i.e. shape from shading) having no domain adaptation capability on endoscopic data. [17, 18]. In this study, we present a 3D re-

construction pipeline from monocular endoscopy video sequences without requiring any labeled depth information. We perform RGB-Depth fusion using the depth and camera pose values estimated by the pre-trained pose and depth networks. Therefore, the pipeline enables capsule robots to have a domain-specific dense surface reconstruction capability without requiring additional sensors to obtain depth and pose data.

1.1. Contributions

In this work, we introduce one of the most comprehensive and open-source simulation platforms [1] for active capsule robots, which can lead to significant improvements in the endoscopy field:

- It offers both physically and visually realistic virtual test-bed, which helps to facilitate and accelerate the design, testing, and optimization process of endoscopic systems and associated algorithms under controlled and reproducible setups.
- Various physical aspects of magnetically-driven capsule endoscopy procedure (i.e. camera design, illumination, varied organ types, strength and placement of magnetic sources) can be easily modified and evaluated in the simulation.
- It enables generating cross-patient, large-scale, and photo-realistic endoscopy datasets that contributes to the performance of data-driven algorithms in various real-world medical image analysis tasks (i.e. disease classification, trajectory estimation) which has been demonstrated in our published paper [9].

To the best of our knowledge, this work extends the area coverage path planning problem for the first time in the endoscopy field to help reducing the miss rate of gastrointestinal diseases by introducing:

- A fully automated active navigation and area-coverage targeted path planning approach for magnetically actuated capsule endoscopes.
- A reinforcement learning setup for training a domain-specific and cognition based coverage policy.

[1] https://github.com/CapsuleEndoscope/VirtualCapsuleEndoscopy

- A novel reward function for the area coverage task considering the capsule endoscopy circumstances.

The proposed approach demonstrates a promising coverage performance on test stomach organs compared to the baseline approaches, also indicating a successful reasoning and adaptability to unseen environments.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents a brief summary of the related work in the endoscopy domain and robotic area coverage methods. Chapter 3 introduces the proposed virtual environment and reveals the details of its critical components including human organ models, magnetic interaction, gastrointestinal friction, and endoscopic camera view. Chapter 4 presents the proposed methodology for the area coverage, pose and depth prediction, and 3D reconstruction tasks. The experiments and results for these methods are presented in Chapter 5. Finally, chapter 6 states the concluding remarks, discussions, and future directions.

2. RELATED WORK

This chapter first presents related work on endoscopic localization, mapping, and path planning methods. Then, it reviews both traditional and learning-based coverage path planning techniques in the robotics field.

2.1. Endoscopic Localization Mapping and Path Planning

The recent progress in the development of robotic systems in the endoscopy field has given rise to the development of Simultaneous Localization and Mapping (SLAM) approaches. Grasa *et al.* [19] introduced a first visual SLAM approach based on Extended Kalman Filter (EKF) for localization and soft tissue mapping on synthetic monocular endoscopy sequences. In a follow-up study, they validated the feasibility of their approach also on human in-vivo image sequences [20]. Later, a fully dense RGB-Depth SLAM method was proposed particularly for endoscopic capsule robots [11]. This method uses frame-to-model camera tracking and surfel-based fusion method [21] and tackles some visual challenges typically occurring in endoscopic screenings including removal of specular reflections and correction of vignetting distortions.

Lately, deep learning techniques have been widely adopted in endoscopic visual localization and mapping strategies due to their tremendous success in various computer vision tasks. Turan *et al.* addressed the supervised visual odometry problem on capsule endoscopy data with pose and depth labels recorded inside a real pig stomach [8]. They use Recurrent Convolutional Neural Networks (R-CNN) to exploit both spatial and temporal dynamics in an end to end manner. Another study performed depth prediction using Generative Adversarial Networks (GAN) on a labeled human endoscopy data and demonstrated its success in an RGB-D SLAM method [15].

Subsequently, unsupervised VO approaches have received considerable attention to cope with the challenge of obtaining such labeled endoscopic datasets. A recent study developed a fully unsupervised VO method for endoscopic capsule robots [12] through

simultaneously training single-view depth and multi-view pose encoder-decoder CNN proposed in [22]. The outputs of the depth network are used with camera poses of nearby views generated from the pose network to reconstruct the target view. Then the loss function (photometric loss) is calculated by pixel-wise comparing the reconstructed and actual views, which provides a self-supervision during the training. Another recent work [23] also tackles the self-supervised visual odometry learning problem on public endoscopy video dataset Hamlyn [2] by training a U-Net [24] encoder-decoder architecture with the ensemble of loss functions presented in [25, 26]. In this work, we follow the same self-supervised learning paradigm with an additional scale consistency loss function proposed in [27], which yields comparable results with the existing approaches in the endoscopic pose estimation task.

In parallel to localization and mapping approaches, several works have attempted to develop path planning algorithms for active capsule robots. A camera-based haptic feedback system was introduced as a proof of concept work to guide the operator during the navigation [13]. Similarly, a recent work proposed an image based stomach fold detection algorithm for planning the desired path and presented various maneuver techniques to navigate the capsule robot on the generated path [28]. Another recent work proposed a learning based path planning approach to provide a continuous control for soft capsule endoscopes and tested its performance on the task of realizing predefined trajectories inside an ex-vivo porcine stomach [29]. This work also focuses on the learning-based adaptive path planning approach and stands out from the existing works by tackling the area coverage problem.

2.2. Robotic Area Coverage

Coverage path planning (CPP) was initially addressed by Hert *et al.* [30] and Choset *et al.* [31] and it has become a prominent problem in several robotic applications including terrain coverage [31], robotic demining [32], robot vacuum cleaner [33], and drone surveillance [34]. Early studies have focused on developing path planning methods via simple heuristics without considering the uncertainties stemming from the

[2] http://hamlyn.doc.ic.ac.uk/vision/

robot's actions and observations [35, 36]. To address this uncertainty issue, belief space path planning has emerged assuming the state of the robot cannot be known all the time [37, 38]. This method maintains the probability distribution over the possible states, which constitutes the belief space of the robot, and performs the best action given the current belief and the information about the environment.

More recently, learning-based techniques have become popular in autonomous robotic systems. In particular, deep reinforcement learning approaches have achieved great success in several reasoning tasks through a powerful representation of multi-dimensional and large scale input data with the action-reward based supervision mechanism [39, 40]. Existing works have studied various training approaches for developing the area coverage focused path planning methods. Zhu *et. al.* [41] used extrinsic reward-based reinforcement learning and Gupta *et al.* [42] considered imitation learning approach which is learning by imitating the behavior of an expert. Besides training approaches, several deep learning architectures for area coverage tasks are proposed to obtain an expressive feature representation including the combinations of vanilla neural networks [43], CNN [39], and Recurrent Neural Networks (RNN) [44]. In this work, we propose a DRL architecture that uses both feed-forward and residual networks for spatial representation of the sensory inputs and learns to autonomously cover the areas in a task-agnostic manner implying that the only supervisory signal is provided by an extrinsic reward.

3. SIMULATION

In recent years, the problem of transferring learned skills in simulated environments to real-world tasks has become a prominent topic in the robotics field. Numerous past works have tackled the problem of reducing the gap between simulation and reality. Recent studies demonstrated that deep learning models trained on highly realistic synthetic data, generated using the modern computer graphics technology, exhibit similar behavior when trained on real data for visual recognition tasks [45, 46]. As well as, learning control policies via reinforcement learning would be very tedious and time-consuming in real-world setups since it requires a huge number of trials until an agent successfully learns to solve a given task in complex settings. This makes simulation systems become an invaluable tool in terms of training and testing control policies under several experimental configurations. In this regard, creating both physically and visually realistic virtual environments play a significant role in having a smooth transition when applying learned skills to real-world scenarios. Therefore, we introduce a highly realistic virtual environment for simulating the active capsule endoscopy procedure and training the proposed autonomous area coverage path planning, as well as, pose and depth estimation methods by utilizing the data recorded via simulation environment. In this chapter, we present the details of the simulation environment and discuss several aspects of achieving realistic circumstances for realizing active capsule endoscopy operations.

3.1. Simulation Overview

Recent advances in computer graphics technology have enabled the creation of photo-realistic virtual environments. These advances have increased the popularity of dedicated 3D graphics development platforms, which allows for simulating the actions with accurate physics and generating high-quality rendering opportunities in real-time. In this work, we build the simulation environment using the popular graphics engine Unity [3] . This engine has a powerful built-in tool for simulating fundamental physics

[3]https://unity3d.com/

and other features such as physical camera configurations, real-time post-processing effects, etc.

The simulation setup consists of three major elements including a capsule robot, a 3D model of the human GI organs, and a big permanent magnet located over the organs. The capsule robot contains an endoscopic camera, a light source, and a tiny magnet. By moving the magnetic actuator the capsule robot is driven inside the organs through electromagnetic attraction force. In this way, we can navigate the capsule inside the 3D model and perform endoscopic video recording. Figure 3.1 illustrates the synthetic images for each gastrointestinal organ recorded by using our virtual environment compared to images taken from real endoscopy datasets [47, 48, 49]. As seen in the figure, we achieve a very close resemblance to the real endoscopic records. In the following sections, we present the details of creating such a resemblance in the simulation.

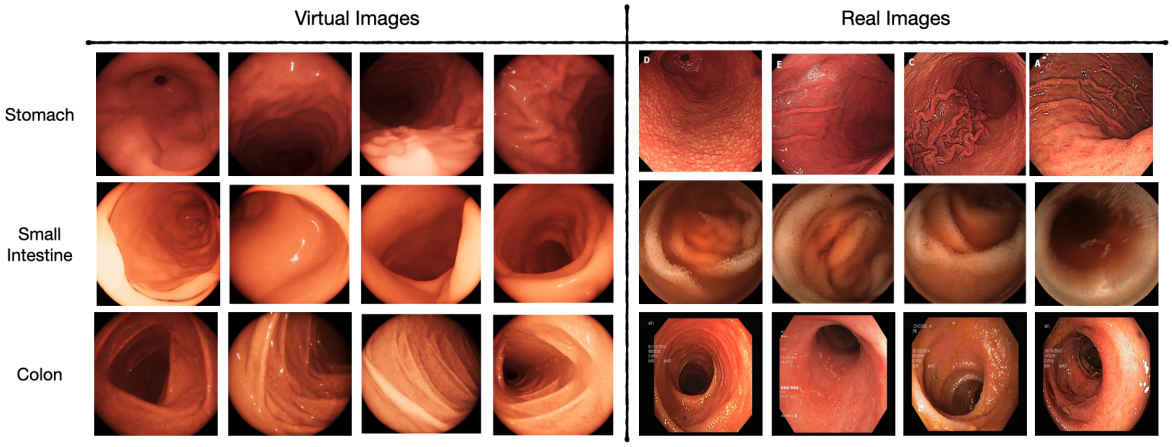


Figure 3.1: The illustration of synthetic images recorded in the simulation environment compared to the real endoscopic scenes for each gastrointestinal organ.

3.2. Generating 3D Organ Models

One of the major problems in achieving a realistic simulation for endoscopic procedures is to create human GI organ models with accurate representations of their 3D geometry and topology. To achieve that, we generate organ models using the Computed Tomography (CT) scans of real patients, which can be found in The Cancer

Imaging Archive (TCIA) [50]. Since these CT scans are recorded as a sequence of volumetric images, we perform 3D surface reconstruction via a free and open-sourced medical imaging tool Horos [4] . As seen in Figure 3.2, the reconstructed 3D model contains a whole body including skin, bones, organs, and some artifacts that occur depending on the imaging quality. Thus, we apply post-processing operations to extract GI organs from the body via an open-source 3D graphics modeling engine Blender [5] . As a last step, we generate a final texture for each organ by stitching various real endoscopic images acquired from the Kvasir dataset [49] and project them onto the surface of the extracted 3D models.

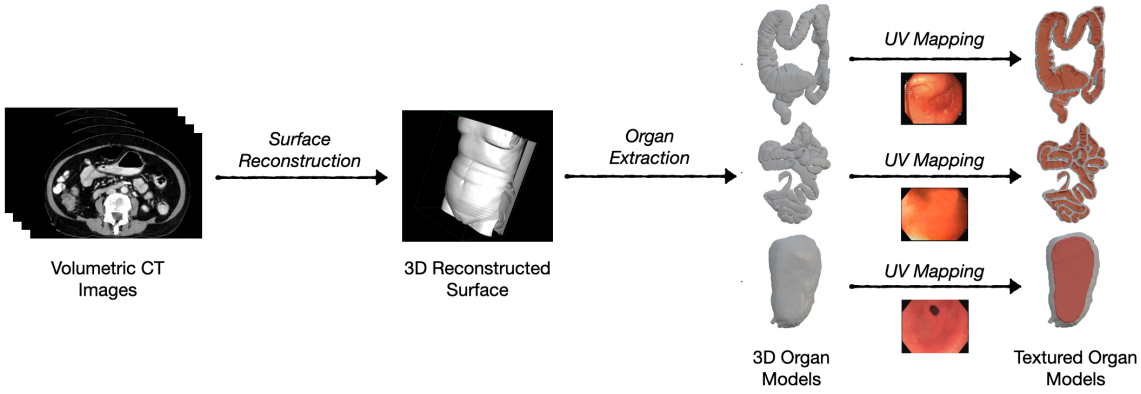


Figure 3.2: The process of generating realistic 3D gastrointestinal organ models used in the simulation. Computed Tomography (CT) images are first reconstructed, then the organs are extracted from the whole body via post-processing. Finally, generated flat textures are wrapped into each organ with the UV mapping technique.

3.3. Endoscopic Camera View

Simulating the characteristics of a capsule robot camera is crucial to achieve a close resemblance with real endoscopic views. The standard affine camera model is often used in capsule robots. In this model, the mapping between camera coordinates and its 2D projection on the image plane is determined by the camera intrinsic parameters including the focal length (f_x, f_y), the optical center (c_x, c_y), and the lens skew

[4] https://horosproject.org/
[5] https://www.blender.org/

coefficient s. With the configuration of these parameters, any type of affine camera can be modeled. In the simulation, we use one of the most popular capsule endoscopes Mirocam®[6] where its intrinsic parameters can be found in [7] .

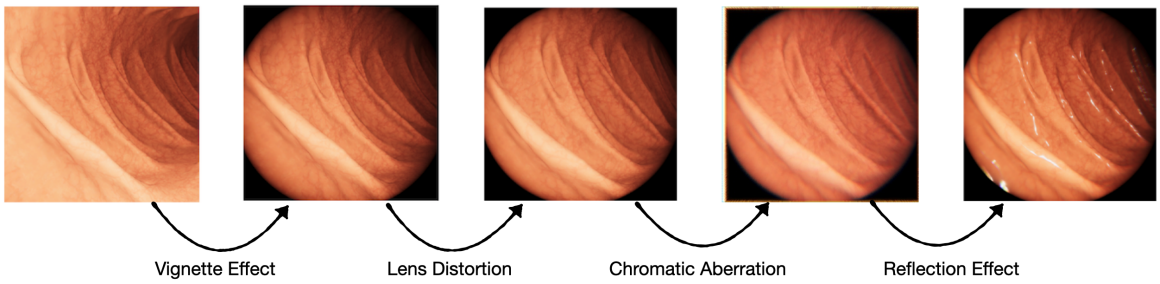


Figure 3.3: Real-time post-processing effects applied on a sample image frame recorded inside a 3D colon model. These effects are used to mimic the important characteristics of real capsule endoscopy scenes.

Besides simulating a physical camera, we provide several post-processing effects that are needed to mimic endoscopic images. The camera lenses in capsule endoscopes produce various undesired effects due to size constraints. First, they suffer from vignetting distortion that is peripherally darkening of an image plane. This occurs when a light source hits the lens aperture at a strong angle. Second, fish-eye lenses are commonly used in capsule endoscopes to maximize the view angle but they often introduce hemispherical lens distortions. Another problem is chromatic aberration causing distorted color combinations and blurry edges in high contrast areas. This happens when all the color wavelengths are not converged into the same focal point. Lastly, specular reflections often take place in endoscopic images when a light source hitting the inner organ mucosa. All these problems occurring during the endoscopic screenings can significantly affect capturing photo-realistic endoscopic images. Therefore, we apply these effects directly to the virtual camera buffer which allows for real-time rendering opportunities. Figure 3.3 demonstrates the changes after each effect is applied to an image frame recorded inside a colon model.

[6]https://www.intromedic.com/
[7]https://github.com/CapsuleEndoscope/VirtualCapsuleEndoscopy

3.4. Modeling Magnetic Interaction

Magnets are typically used as external locomotive instruments for actively navigated capsule endoscopes. Although the Unity platform provides a built-in engine for simulating the fundamental physics such as collision, gravity, etc, the simulation of electromagnetism is not provided. Hence, we simulate the Gilbert's magnetic dipole-dipole interaction model to calculate the force between permanent magnet to the magnet inside the capsule robot.

In this model, the force between to magnetic dipoles $\vec{m_1}$ and $\vec{m_2}$ are computed as follows

$$|\vec{F}| = \frac{\mu_0 q_{m_1} q_{m_2}}{4\pi r^2} \tag{3.1}$$

where μ_0 is the free space permeability ($4\pi \times 10^7 \frac{N}{A^2}$), q_{m_1} and q_{m_2} are the magnetic charges of the two corresponding poles, and r is the distance between these poles.

3.5. Modeling Intestinal Friction

Due to the viscoelastic characteristics of gastrointestinal organs, it is not possible to model the intestinal friction only by the Coulomb's law. Multiple frictional resistances occur during the navigation of capsule robots [51, 52]. These resistances act towards the opposite direction of the capsule and can be represented as a function of velocity $\vec{u}_f = -\vec{v}/\|\vec{v}\|$. According to a comprehensive work on the friction analysis of endoscopic capsule robots [53] total frictional resistance $\vec{f}(\vec{v})$ can be defined as

$$\vec{f}(\vec{v}) = \vec{f}_{coulomb} + \vec{f}_{environment} + \vec{f}_{visco} \tag{3.2}$$

where $\vec{f}_{coulomb} = -\mu\|N\|\vec{u}_f$ is the Coulomb friction, where N is normal contact force, $f_{environment}$ is environmental resistance which occurs due to pressure and elastic deformation of organ wall, and f_{visco} is the viscosity related resistance caused by intestinal

mucus.

The environmental resistance can be defined as

$$\vec{f}_{environment} = P(x)\vec{S}\sin(\theta) \tag{3.3}$$

where $P(\vec{x})$ is the pressure on the contact surface area $\vec{S}$ and *theta* is the skew angle of this pressure. This type of resistance only occurs in the lower gastrointestinal tract, thus, we discard $\vec{f}_{environment}$ when the capsule is navigated inside a stomach. The viscosity frictional resistance is defined as

$$\vec{f}_{visco} = -\gamma_{visco}\vec{v} \tag{3.4}$$

where γ_{visco} is the coefficient of intestinal mucus viscosity.

In the simulation, we model all these three forms of frictional forces based on the measurements given in [53]. This work demonstrated a logarithmic correlation between the capsule velocity and total frictional force applied to the capsule. To simulate the total frictional force, we define the following velocity dependant function

$$f(|\vec{v}|) = \begin{cases} C & |\vec{v}| \leq 1 \\ x\log(y|\vec{v}| + z) + C & |\vec{v}| > 1 \end{cases} \tag{3.5}$$

where C represents the constant frictional resistance. To approximate parameters, we perform curve fitting on the reported data points via a least-squares optimization technique [54]. The fitted values for the variables x, y, z, and C are 55.04, 0.23, 1.04, 100 respectively.

4. PROPOSED METHOD

This chapter presents proposed methodologies for reinforcement learning-based coverage path planning, pose and depth estimation as well as surface reconstruction tasks.

4.1. Area Coverage Path Planning

Area coverage path planning is the problem of determining the path of a mobile robot that seeks to efficiently visit or observe all the data points in a given environment. The efficiency in this problem is defined as the total amount of observed areas over the total time taken. To achieve an efficient are coverage strategy, the three fundamental questions needs to be addressed:

1. What is the current state of the robot?
2. Which areas have it covered so far?
3. Where will it go next?

In order to address the first and second questions, the robot needs to be equipped with a robust localization and mapping method to continuously determine its states during the navigation in an unknown environment. The answers for these questions are used to address the third question which can be tackled by a successful reasoning mechanism that continuously and actively determines the path of the robot based on environmental circumstances. The reasoning mechanism in area coverage task should always seek to explore new areas and cover as much area as possible in a given time without taking redundant actions.

To address the first question, we propose an unsupervised learning mechanism using deep neural networks to predict pose and depth information from the monocular camera feedback. For the second question, we develop a mapping functionality in our simulation environment to keep track of the visited places and visually perceived

locations of the organs. Regarding the last question, we train a reinforcement learning model that learns a policy to navigate endoscopic capsule robots for the area coverage task. In this chapter, we present the details of the proposed methods for area coverage policy learning, pose and depth estimation, and dense surface reconstruction problems.

4.1.1. Policy Learning

Reinforcement learning (RL) tackles the problem of training an agent that completes a given task in an unknown environment through an action-reward mechanism. The goal of the agent is to maximize the expected cumulative reward which results in making an optimal decision at each time step. For an arbitrary time step t, the agent finds itself in a state S_t that summarizes the information it receives from the environment. The fundamental assumption of RL is that the current state contains all the relevant information of previous states, which can be expressed as a Markov property

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2 \ldots, S_t] \tag{4.1}$$

where the selection of the next state depends only on the current state which makes all the previous states obsolete. Thus, the goal of the agent is to choose the optimal action from that state onwards to maximize the cumulative expected return. By applying this principle of optimality in each state, the agent learns how to behave optimally across all the states.

The main objective of the area coverage task is to maximize the information gain from any situation during the navigation of a robot. However, the robot cannot have a static and pre-determined path while exploring new areas due to uncertain outcomes of actions. For instance, the position of a robot can suddenly change due to exogenous events occurring during the navigation. Particularly, in the capsule endoscopy context, a capsule robot can likely be affected by random contraction and relaxation forces of organs which can take the robot into unexpected states. In this case, a coverage based path planner needs to account for these uncertainties and adapt itself accordingly in

order to cover as much area as possible from any state onwards. Therefore, Markov property plays a crucial role in allowing us to learn a stochastic policy that produces the right actions to take for the area coverage objective while considering the uncertain outcomes.

The stochastic policy can be defined as

$$\pi(a|s) = P[A_t = a|S_t = s] \tag{4.2}$$

describing the probability of taking an action a from the action space A_t in a state s from S_t. Based on the action provided by the policy, the agent receives a reward r_t which can be expressed as a function of the current state, the action itself, and the next state of the agent after taking the action

$$r_t = R(s_t, a_t, s_{t+1}). \tag{4.3}$$

4.1.2. Proposed Policy

Let us consider a robotic capsule equipped with a monocular RGB camera that is controlled inside the GI organs by an external electromagnetic actuator. We propose to learn a stochastic area coverage policy π_c that produces action vectors at each time step t for the actuator based on the sensory inputs provided by the capsule. As a result of this set of actions provided by the policy, we want to achieve maximum area coverage for the digestive tract of any patient in a minimum possible operation time.

To achieve that we develop a policy learning pipeline that is illustrated in Figure 4.1. The environment state S_t consists of the visual camera observation o_t, the state of the capsule robot (position, orientation, velocity) c_t, and the state of the permanent magnet m_t (position, orientation) and a deterministic 3D point cloud map M_t at time instance t.

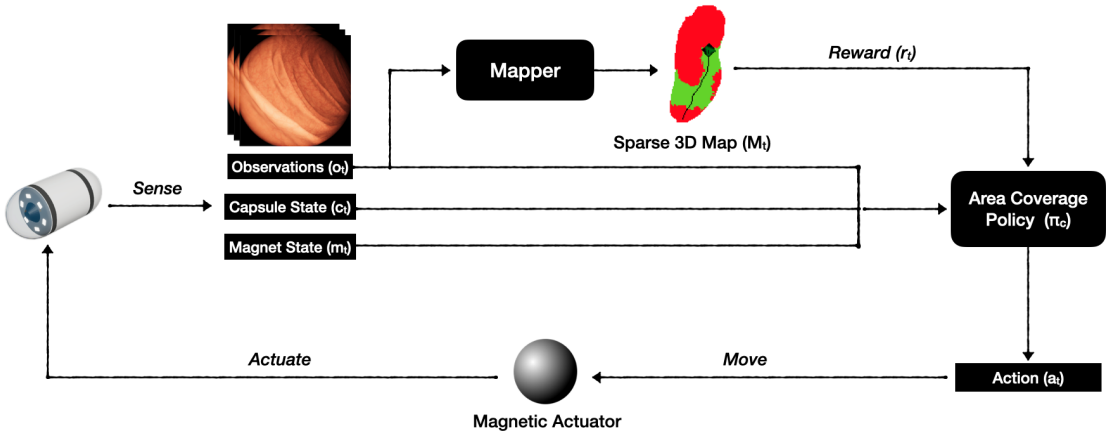


Figure 4.1: Autonomous area coverage policy learning pipeline for endoscopic capsule robots. The policy π_c receives camera observations, the position and the velocity of the capsule, as well as the position of the magnetic actuator and outputs an action vector for planning the motion of the actuator. A built-in mapper functionality generates a real-time 3D sparse map in the simulation which provides a reward for the produced action based on the amount of covered areas.

From the visual observations, we generate this map in the simulation to keep track of the monitored parts of an organ, and the trajectory of the capsule inside this map. The organ model is represented by a polygon mesh in 3D space. It consists of a set of vertices, edges, and faces that defines the shape of the model. In our model, the 3D map is created by real-time tracking the vertices of the related organ mesh. If a vertex has ever been visible inside the camera's field of view, that vertex is considered as observed and the observed set of points that represent the generated map. This allows us to calculate the exact ratio between covered and uncovered parts of the organ, as well as, the increase in the total coverage at each time step which provides a reward signal r_t on how good the performed action is in monitoring new areas.

The policy function takes the state $S_t - M_t = \{o_t, c_t, m_t\}$ as input and produces an action vector a_t that determines the motion of the magnetic actuator along the x, y, z axes. As the magnet moves based on the produced action vector, the capsule robot is manipulated inside the organ, hence, S_t gets updated. In this case, we are

learning an indirect relationship between the action taken by the magnet and the state of the capsule robot. In other words, although we consider the state of the magnet as an input, the policy is optimized based on the impact of the action on the capsule robot.

As we defined earlier, the reward function r_t does not only depend on the action but also the state description. In the proposed schema, although the generated map M_t is decided to be not used as an input to the policy due to the dimensionality problem, we have defined it as part of our state $S_t = \{o_t, c_t, m_t, M_t\}$. Since our reward r_t is determined based on the status of the M_t for the area coverage task, the map is implicitly involved in the policy learning. This also suggest the r_t being stationary where the same state, action, and next state pairs produce the same reward at any times ($R(s_t = s, a_t = a, s_{t+1} = s') = R(s_{t+k} = s, a_{t+k} = a, s_{t+k+1} = s')$) during the learning process. This stationarity allows us to converge optimal policies under Markov assumption which implies that the state definition contains all the relevant information required for decision-making. On the other hand, how the agent perceives the environment state is not fixed. Since the parameters of the policy function gets updated as the agent interacts with the environment, the encoding of the environment state can change over time. In this regard, it might be difficult to argue whether the Markov assumption entirely holds for the problem. However, we know that the change in the state representation is expected to become stationary after sufficient number of trials. Therefore, the convergence to the optimal policy can still be satisfied under the Markov property.

We define the area coverage policy as a time-based episodic task which means that an agent needs to complete a given task in a fixed-length time period. Therefore, we formulate the total return (R) as the sum of rewards obtained in a fixed time window T which is denoted as

$$R = \sum_{t=1}^{T} r_t. \tag{4.4}$$

The length of the episode is determined based on the complexity of the environment

which must be long enough to fully cover the environment without any redundancy. The training consist of a series of episodes wherein each episode the agent performs a set of actions and collects the reward accordingly until the expected cumulative reward converges at a maximum point. Therefore, the terminal criteria of the training is based on the convergence point where the agent completes the desired task successfully but it no longer collects higher reward in future episodes.

4.1.3. Reward Function

A successful area coverage policy for capsule endoscopes depends on satisfying the following objectives. First, the capsule robot should maximize the amount of covered area in a minimum possible time frame (i.e. repetitive actions should be avoided). Second, the capsule robot must have strict speed constraints in order to not damage the organ tissue during the navigation [55]. Based on these two objectives, we design a reward function for the endoscopic area coverage policy that depends on the linear difference in coverage amount between two consecutive time steps, and the speed of the capsule robot.

Total reward at time t is calculated as $r_t = w_1 r_{ct} + w_2 r_{vt}$ where r_{ct} is defined as

$$r_{ct} = \begin{cases} -\alpha & C_t - C_{t-1} < C_{thresh} \\ \beta(C_t - C_{t-1}) & C_t - C_{t-1} \geq C_{thresh} \end{cases} \tag{4.5}$$

where C_t denotes cumulative area coverage in percentage until t and α is the constant penalty given when the agent action does not result in significant increase in the coverage amount which is specified by C_{thresh}, and β is the scaling factor used for faster convergence. The parameters α, β, and C_{thresh} are empirically determined as 0.1, 0.1, 0.003 respectively.

The other reward component r_{vt} is a function of the capsule speed and defined

as

$$r_{vt} = \begin{cases} 0 & \|\vec{v_t}\| \leq v_{max} \\ -\gamma(\|\vec{v_t}\| - v_{max}) & \|\vec{v_t}\| > v_{max} \end{cases} \tag{4.6}$$

where the agent receives a penalty (negative reward), which is the difference between the capsule's current speed $\|\vec{v_t}\|$ and the maximum speed limit v_{max} scaled by a constant scalar parameter γ, when $\|\vec{v_t}\| > v_{max}$. This reward function enables the magnetic actuator to learn adjusting the strength of the electromagnetic force based on its distance from the capsule, thus, it keeps the speed of the capsule robot in safety limits during the endoscopic screening. Also, the parameters w_1, w_2 in the r_t denotes the trade-off between these two objectives which are empirically determined as 0.9 and 0.1 respectively. The empirical results for these weight parameters are presented in the section 5.1.

4.1.4. Policy Optimization

In the section 4.1.1, we have simply defined the policy as the probability distribution of actions over states $\pi(a_t|s_t)$. We extend this definition by parametrizing the policy as follows

$$\pi_\theta(a|s) = \pi(a|s,\theta) = P[A_t = a|S_t = s, \theta_t = \theta]. \tag{4.7}$$

This makes the probability distribution directly controllable by the parameters θ. In other words, the maximization of total reward is based on learning best set of parameters for each action state pair. We optimize the θ by the gradient ascent algorithm based on the objective function $J(\pi_\theta)$, also known as policy gradient optimization. The algorithm seeks for a local maximum in $J(\pi_\theta)$ by ascending the gradient of the policy with respect to θ which can be formulated as

$$\theta^{'} = \theta + \alpha \nabla_\theta J(\pi_\theta) \tag{4.8}$$

where $\nabla_\theta J(\pi_\theta)$ is the policy gradient denoted as

$$\nabla_\theta J(\pi_\theta) = \begin{pmatrix} \frac{\partial J(\pi_\theta)}{\partial \theta_1} \\ \frac{\partial J(\pi_\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial J(\pi_\theta)}{\partial \theta_n} \end{pmatrix} \tag{4.9}$$

and α is the learning rate that determines the step size of the parameter update. The gradient ascent optimization is performed on-policy, where each update only takes the data collected while the agent acts based on the latest version of the policy. The algorithm describing the overall policy gradient optimization process is shown in the Figure 4.2.

Algorithm 1 Policy Gradient Optimization

```
Randomly initialize the model parameters θ
for each episode {s_1, a_1, r_1, ..., s_T, a_T, r_T} under policy π_θ do
    for t = 1, 2, ..., T do
        ∇_θ J(π_θ) ← ∂J(π_θ)/∂θ
        θ ← θ + α∇_θ J(π_θ)
    end for
end for
```

Figure 4.2: Policy gradient optimization algorithm.

4.1.5. Policy Network Architecture

As mentioned in the previous section, a policy can be defined as a high-dimensional parametric function. In the context of DRL, such a function is represented by a deep neural network architecture. Figure 4.3 demonstrates the proposed network architecture for the are coverage policy π_c. It consists of two main encoder blocks, and one of them is the IMPALA ResNet [56] CNN architecture, which is used for encoding the visual observations. This encoder contains one convolutional block followed by two residual blocks, and it takes a batch of 64x64 RGB images as inputs and represents them in the latent vector space. As described in Figure 4.4, the convolutional block

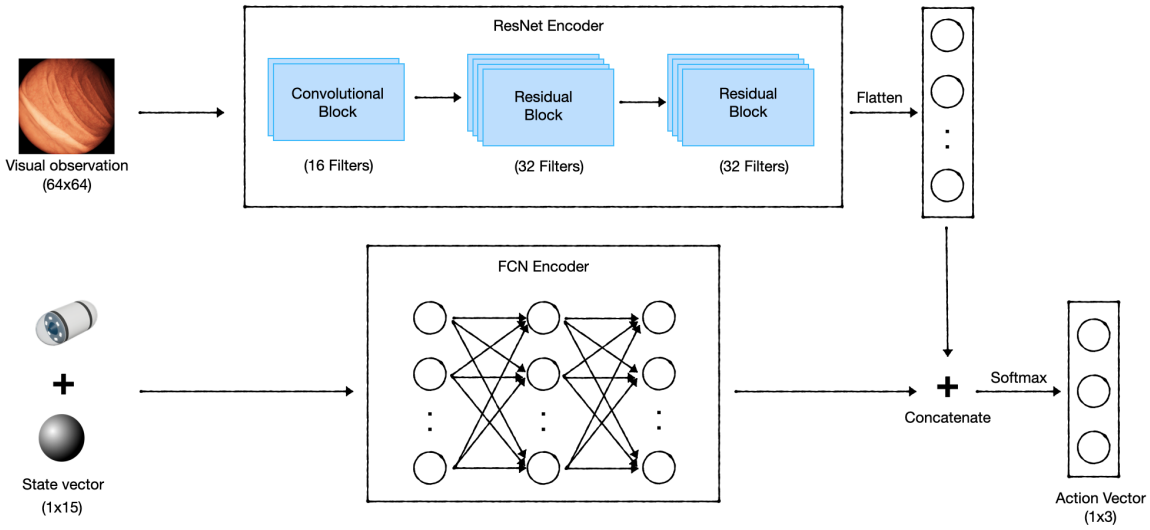


Figure 4.3: Deep neural network architecture for the proposed area coverage policy π_c. Visual and vector observations are encoded in parallel by a convolutional and a fully-connected networks respectively. The outputs of these networks are concatenated and then mapped into a three-dimensional action vector.

executes 3x3 convolution operation followed by a $3x3$ max pooling. The residual block performs two 3x3 convolution operation followed by a RELU activation function. It also includes a skip connection which provides an alternative path for the gradient calculations during the back-propagation process. The vector observations including the states of the capsule and the magnet are encoded by a fully-connected network that contains three layers and 256 hidden units for each layer. The output of this network is concatenated with the output of the ResNet encoder. Then the softmax function $softmax(X) = e^X / \sum e^X$ is used to map the concatenated vector to corresponding action probabilities from the action space and the one with the highest likelihood determines the final action vector.

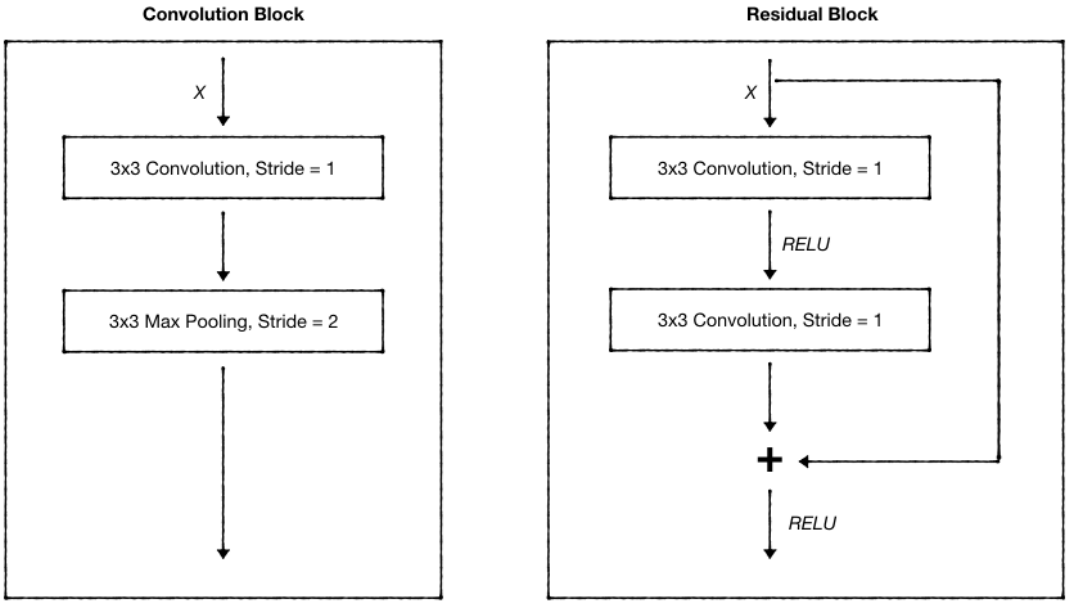


Figure 4.4: Representation of convolution and residual blocks used in the ResNet encoder architecture.

4.2. Visual Odometry and Depth Learning

Estimating the pose of a robotic capsule from the monocular visual feedback is crucial in terms of successful navigation, and determining the location of abnormalities inside the patient body. Such estimation can be obtained using visual odometry techniques. VO is the task of determining the pose (translation and orientation) of a mobile robot relative to its initial location by analyzing the stream of images captured by an associated camera.

In addition, a proper 3D reconstruction of scanned areas by capsule endoscopes depends on the accurate pose and depth information, however, these endoscopes often suffer from the size and battery limitations which prevent the use of additional hardware devices including depth sensors. Therefore, providing monocular pose and depth estimation capabilities carries a great importance for endoscopic robots.

Recent studies have proposed self-supervised deep learning frameworks for single-

image visual odometry and depth estimation problems [57, 58]. These self-supervised approaches have shown comparable performance over supervised approaches with the availability of a massive amount of image sequences as training data. In this work, we take the advantage of generating large-scale photo-realistic endoscopy data using our simulation and use them to train an end-to-end visual odometry and depth learning method in a self-supervised manner.

4.2.1. Training Paradigm

The training paradigm for the self-supervised visual odometry and depth estimation method is based on jointly training multiple neural networks proposed by [22, 27]. Training examples consist only of unlabeled endoscopic image sequences captured by a moving capsule camera inside GI tract organs. The overall training framework is illustrated in Figure 4.5.

As seen in part *a* of the figure, two consecutive RGB input frames (I_t, I_{t-1}) are fed into two identical encoder-decoder CNN DispNet used in [22] and originally adapted from the FlowNet architecture [59], and these networks output the corresponding pixel-wise depth map predictions (D_t, D_{t-1}). Another CNN PoseNet introduced in [22] takes the concatenation of I_t and I_{t-1} as input and produces a 6 degree of freedom (DoF) camera pose relative to the I_t which is represented as $P_{t\to t-1}$. Using the predicted relative pose $P_{t\to t-1}$, we obtain the warped depth map D^t_{t-1} by transforming D_t to 3D space and projecting back onto the image plane of I_{t-1}. With this way, we are able to represent both D_t and D^t_{t-1} in the same image plane, which allows us to get a supervision by computing the geometric consistency loss L_{GC} between D_t and D^t_{t-1}.

Let us define a set of valid points $\{p_1, p_2, \ldots, p_n\}$ which are successfully projected from D_t to the image plane of I_{t-1}. Then, we can define L_{GC} as

$$L_{GC} = \frac{1}{n}\sum_{i=1}^{n}\frac{|D^t_{t-1}(p_i) - D_{t-1}(p_i)|}{D^t_{t-1}(p_i) + D_{t-1}(p_i)} \tag{4.10}$$

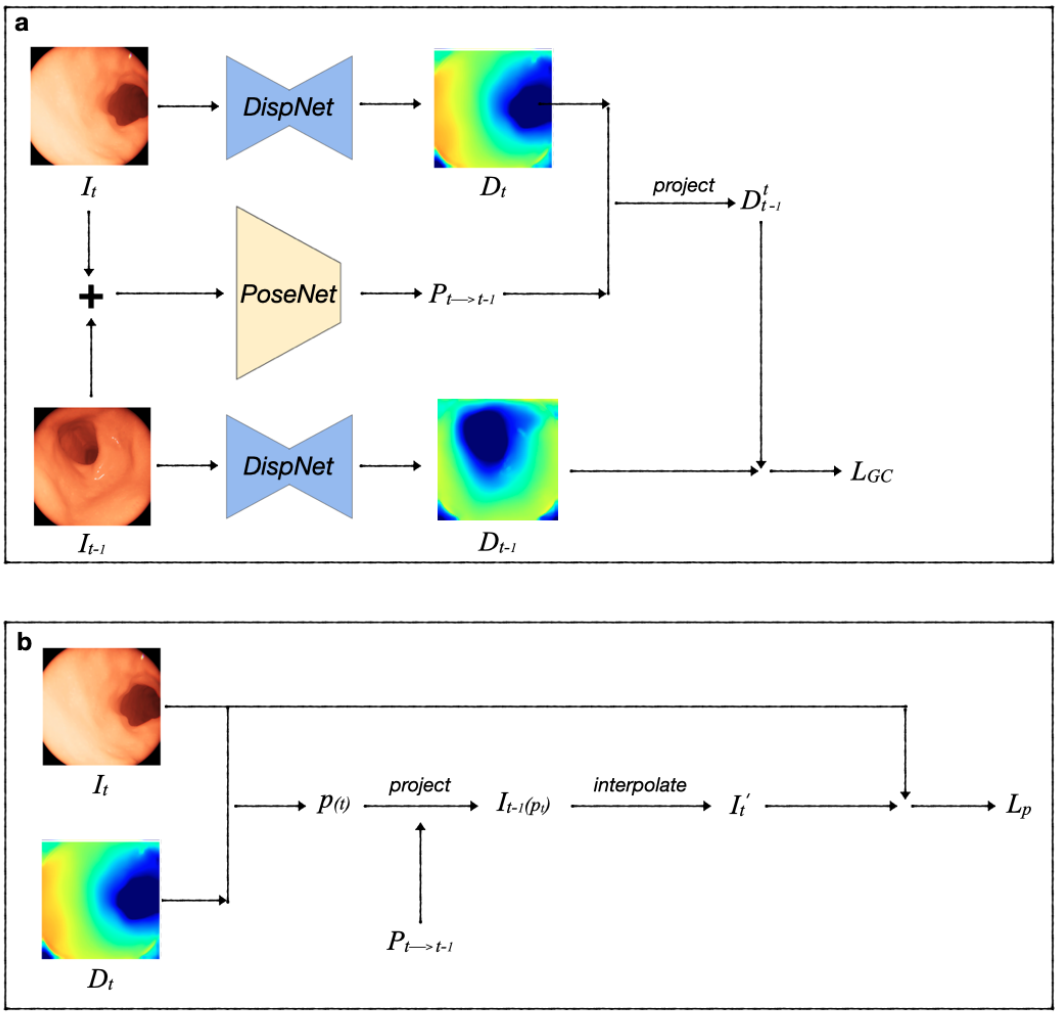


Figure 4.5: The demonstration of self-supervised training paradigm to estimate camera pose and depth information from monocular capsule endoscopy video sequences. Consecutive image frames I_t, I_{t-1} are fed into the two identical encoder-decoder CNN DispNet [59] to obtain the corresponding depth frames, and an encoder network PoseNet [22] takes the concatenation of the frames and predicts relative camera pose $P_{t\to t-1}$. a) Geometric consistency loss L_{GC} are computed by the geometric difference between depth frames that are represented on the same projection. b) Photometric loss L_p and the point-wise difference between I_t and its projected coordinates into I_{t-1}.

where it corresponds to the geometric difference between predicted depth maps for each consecutive image pairs normalized by their sum. In the training, the consistency loss is propagated to the overall video sequence and the normalization contributes to the numerical stability. The process for the calculation of the geometric consistency loss is

described in the Figure 4.6.

Algorithm 2 Geometric Consistency Loss

1: **for** $t = 1, 2, \ldots, N$ **do**

2: Estimate depth maps D_t, D_{t-1} from input frames I_t, I_{t-1} using DepthNet

3: Estimate relative camera pose $P_{t \to t-1}$ from $concat(I_t, I_{t-1})$ using PoseNet

4: Inverse project D_t into a 3D space and obtain the points in a projective plane

5: Perform linear transformation on these points using $P_{t \to t-1}$

6: Project the points onto the image plane of I_{t-1} and get the warped image D^t_{t-1}

7: Calculate the geometric consistency loss L_{GC} by taking the geometric difference between the pixels of the depth images D^t_{t-1} and D_{t-1}

8: **end for**

Figure 4.6: Algorithm for computing the geometric consistency loss.

Another supervisory signal is achieved by a depth-based differentiable image reconstruction procedure [60]. Similarly, given two input frames I_t and I_{t-1}, we synthesize a new image of the I_t by inversely warping I_{t-1}. The process is illustrated in section b of the Figure 4.5.

We first back-project the I_t into 3D space using the corresponding predicted depth image D_t in order to capture homogeneous coordinates of all pixels in the I_t which is denoted as p_t. Next, we perform a linear transformation on p_t using the predicted relative pose $P_{t \to t-1}$, and project them onto the previous input frame I_{t-1} which gives $I_{t-1}(p_t)$. As a final step, we acquire the warped image $I'_t(p_t)$ by bi-linearly sampling the neighbouring pixel values.

Then, we compute the photometric loss function L_p defined as

$$L_p = \frac{1}{n} \sum_{i=1}^{n} |I_t(p_i) - I'_t(p_i)| \tag{4.11}$$

where p_i stands for the points that are projected from I_t to I'_t. Figure shows 4.7 the algorithm for computing the photometric loss during the training.

Algorithm 3 Photometric Loss

1: **for** $t = 1, 2, \ldots, N$ **do**
2: Inverse project the input frame I_t into 3D space using its predicted depth D_t
3: Perform linear transformation on the inversely-projected points p_t using $P_{t \to t-1}$
4: Project p_t on the image plane of I_{t-1} through projective transformation.
5: Sample the pixels from I_{t-1} via the bilinear sampling method to reconstruct I_t
6: Calculate the photometric loss L_p by taking the pixel-wise intensity difference between I_t and its reconstructed version I'_t.
7: **end for**

Figure 4.7: Algorithm for computing the photometric loss.

Overall, the total loss is calculated as the sum of the two losses in the training which is stated as

$$L = \alpha L_{GC} + \beta L_p \tag{4.12}$$

where α and β are the hyper-parameters that represent the trade-off between these losses.

4.2.2. Network Architecture

As mentioned in previous section, we introduce two different CNN architectures used in this work for predicting depth maps and relative poses between two consecutive frames. The architecture details for these networks demonstrated in Figure 4.8.

DispNet is an encoder-decoder network that performs a pixel-to-pixel mapping between an input image and the corresponding depth map based on the self-supervised learning framework defined in the section 4.2.1. The encoder block consists of 14 convolutional layers where each pair of the layer have the same number of filters ranging from 32 to 512. The kernel sizes for the first and the next two convolution layers are

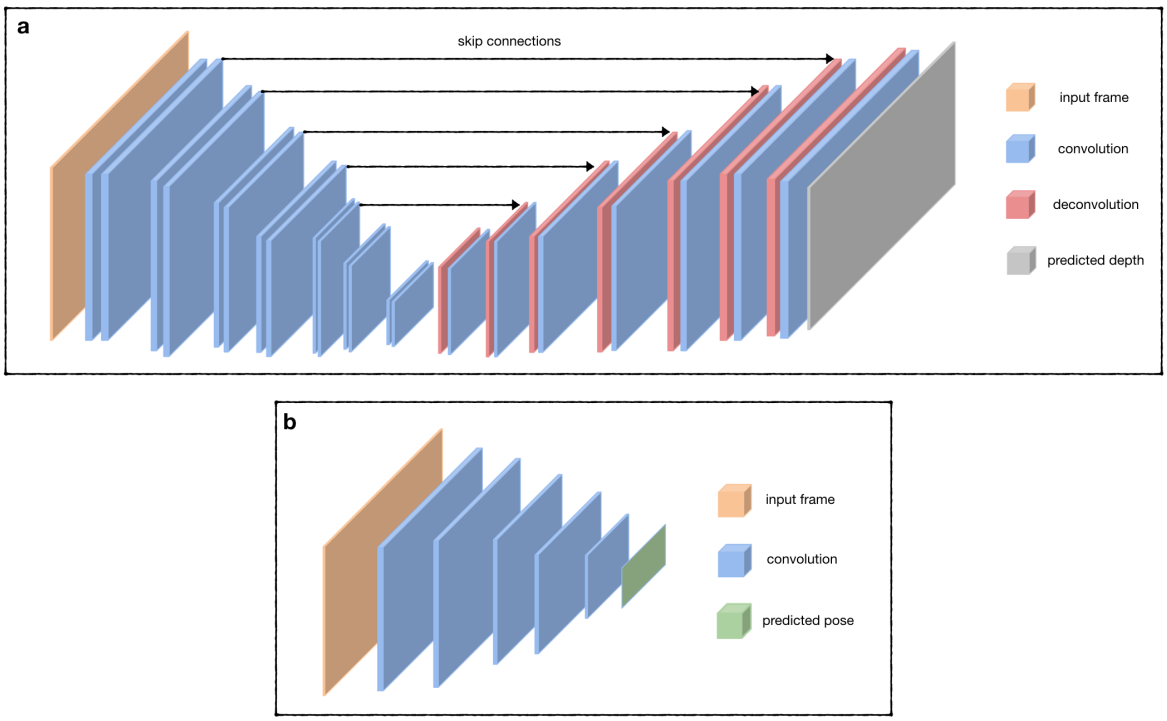


Figure 4.8: Architectural details of the proposed networks a) the DispNet, b) PoseNet. The rectangular blocks represent the input dimensions after each convolution (blue) or deconvolution (red) operation is applied. The DispNet produces a same dimensional matrix with the input frame, which provides pixel-wise correspondence, and the PoseNet produces a 1x6 vector that represents 6 dof camera pose.

7x7 and 5x5 respectively, and for the remaining layers, it is 3x3. The decoder block consists of the 7 pairs of convolution and deconvolution layers. Deconvolution is the transpose of the convolution operation and performs upsampling from the input image. The RELU activation function is used after all the convolution layers. Similar to the ResNet architecture, DispNet also includes from the output of encoder convolutions to the input of decoder deconvolution layers as seen in Figure 4.8.a. For the prediction layer, we use the sigmoid function stated as

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{4.13}$$

where it allows us to constrain the estimated depth values in a positive range.

The pose estimation network PoseNet proposed in [22] is shown in part b of Figure 4.8. The input for the PoseNet is the concatenation of two consecutive frames along all color channels (MxNx6). The PoseNet architecture contains 7 convolution layers. The kernel size is 3x3 for all three layers except for the first two layers with 7x7 and 5x5 sizes respectively. As a final step, these layers are followed by a 1D convolution with 1x6 output channels in order to project the input to a 1x6 vector which corresponds to a 6 DoF camera pose.

4.3. Dense Surface Reconstruction

The objective of a dense surface reconstruction method that provides a complete picture of the overall endoscopic screening process is to provide a convenient diagnosis and treatment opportunity for medical experts. Therefore, we propose to densely reconstruct the surface organ tissues from monocular endoscopic video sequences. Figure 4.9 shows the overall surface reconstruction pipeline. In the proposed schema, we use

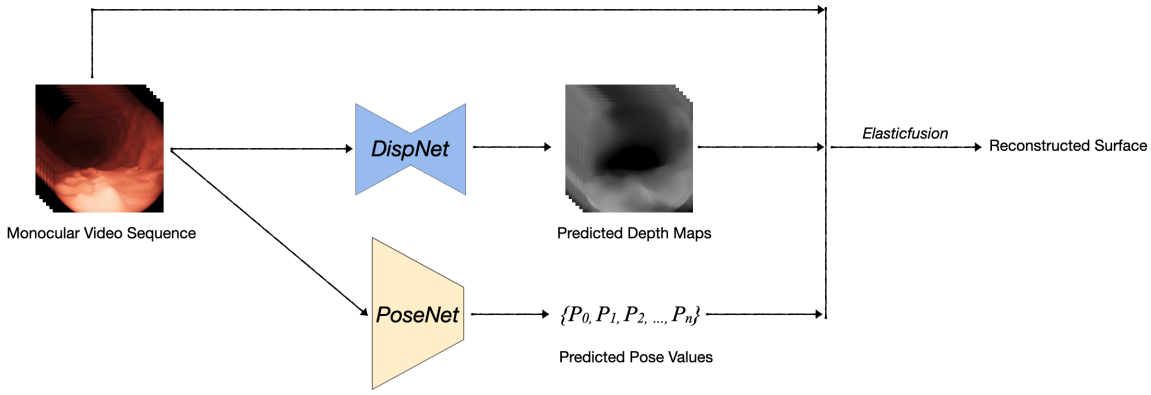


Figure 4.9: The flow diagram of the proposed dense surface reconstruction method. Using the pre-trained DispNet and the PoseNet, corresponding depth maps and pose values are estimated for each image in a monocular endoscopic video sequence. Then these images are fused with their depths by a surfel-based reconstruction method Elasticfusion. This flow provides a real-time 3D dense surface reconstruction experience using a single GPU during the capsule endoscopy procedure.

the estimated pose and depth results obtained from the corresponding networks after they are trained on large-scale endoscopy data with the described training paradigm.

In particular, a monocular endoscopic RGB video sequence is first used to predict associated depth map, and global camera poses. Then, this RGB sequence is fused by surface element model (surfel) based reconstruction method known as Elasticfusion [21].

The underlying idea behind the Elasticfusion can be described in three steps. First, it computes a surfel-based model of the scene by fusing RGB-D frames. Second, it segments the inactive area of the model which remains unobserved for a while, and it attempts to locally register the active parts of the model with every new frame. If registration is successful, the intersecting areas are fused with the active model. Fi-

Algorithm 4 Dense Surface Reconstruction Pipeline

```
Let M be the surface-element based global model of the environment
for t = 0, 1, ..., N do
    Estimate the depth D_t of the input frame I_t with DepthNet
    Estimate the global camera pose P_t for the I_t with PoseNet
    Segment active parts of the model M
    Perform depth map fusion into active parts of the M with I_t and D_t given P_t
    if camera revisits the inactive parts then
        Close a loop and register the active and inactive parts of the M together.
        Reactive the inactive parts
    end if
end for
```

Figure 4.10: Dense surface reconstruction algorithm.

nally, if an intersection between the active model and the inactive model found, the loop is closed, the models are merged and the inactive area is reactivated again. The biggest advantage of using Elasticfusion is its capability of generating dense surface reconstruction in real-time on a single GPU machine which makes our pipeline also works in real-time. Therefore, with the proposed schema, medical experts can simultaneously benefit from the reconstructed surface during the endoscopy procedure. The details of the whole reconstruction process can also be seen in Figure 4.10.

5. EVALUATION

5.1. Area Coverage

The experimental setup for training our area coverage policy is shown in Figure 5.1. The outer view of the setup including, a capsule robot inside a 3D organ model (stomach in this case) and the spherical permanent magnet located on top of the model, is displayed in the top-left. On the top-right, we demonstrate the camera view of the capsule, and based on this view we generate a sparse 3D map that is projected on a 2D plane in real-time as seen in the bottom parts of the figure. In the generated map, covered and uncovered regions of the organ instance is depicted by green and red points respectively. Arrow symbol represents the position and orientation, and the line shows the trajectory of the capsule robot based on the actions taken by the policy.

As stated in section 4.1.1, the terminal criteria for the policy are determined based on the length of the episode. We determined the length of each episode as 7500 iteration steps which corresponds to 150 seconds in real-time. After the completion of each episode the environment variables are reset to its initial state. The other hyperparameters including batch size and learning rate are empirically determined as 64 and $3e^{-4}$ respectively.

The choices of the weight parameters in the reward definition $r_t = w_1 r_{ct} + w_2 r_{vt}$ play a significant role in the convergence of the reinforcement learning model during the training. We experimented the model performance with different parameters to find a sweet spot between the coverage and speed safety objectives. The experiment results are given in the table 5.1 demonstrating the coverage performance as well as convergence and speed violation statuses on the training environment. The models are trained an equal number of iterations and the coverage results are recorded at the end of the episode. Based on this empirical evaluation, the model does not converge when coverage weight $w_1 < 0.5$ causing the capsule robot stuck in the initial states. The cases when $w_2 \geq 0.1$ has turned out to be sufficient for keeping the capsule speed

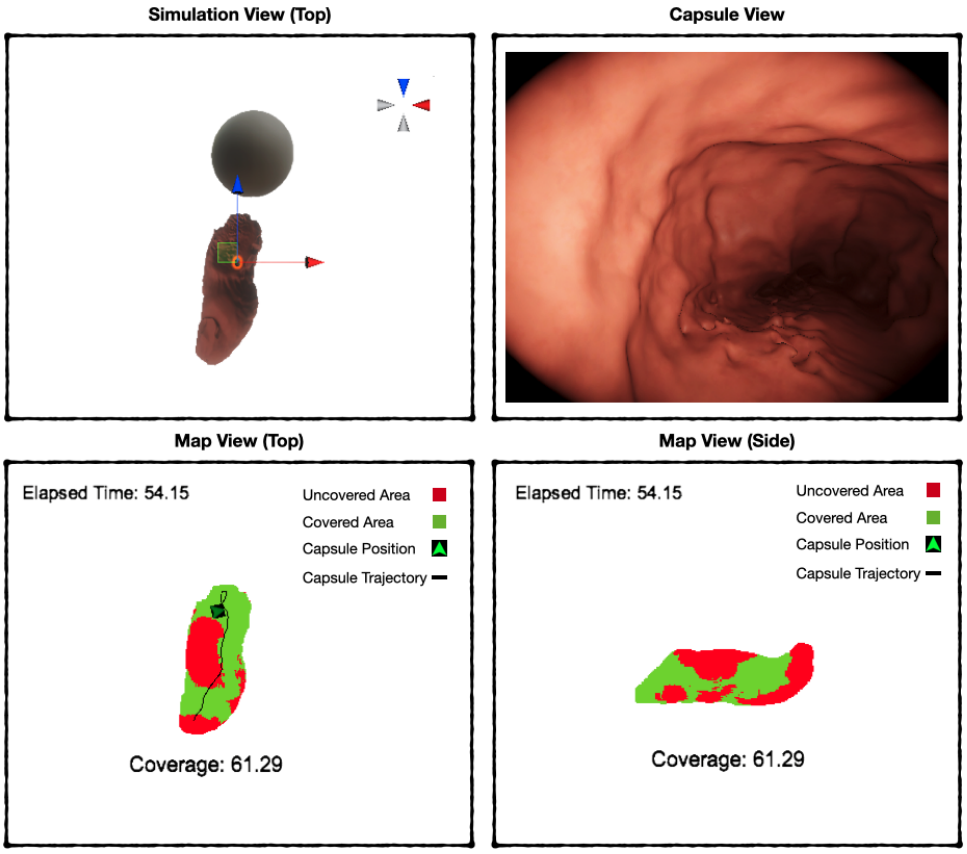


Figure 5.1: The overview of the experimental setup for training the proposed area coverage policy. Top-left: experimental setup in the simulation environment. Top-right: capsule camera view. Bottom: mapper view showing covered (green) and uncovered areas (red) of the organ on a 3D sparse map shown from different perspectives. The arrow and the black line represent the current position and the trajectory of the capsule respectively.

within safety limits. Therefore, the weight parameters w_1, w_2 are chosen as $0.9, 0.1$ respectively which allows the model to cover almost the whole area (97.16%) while still holding the speed requirement.

The one specific optimization method we used for learning the area coverage policy among the policy gradient methods π_c is the Proximal Policy Optimization (PPO) [61], which has achieved the state-of-the-art in a wide variety of robotic reasoning tasks.

Table 5.1: The impact of reward weight parameters in model performance.

Weights (w_1, w_2)	Training Coverage	Model Convergence	Speed Violation
$(0.9, 0.1)$	97.16%	Converged	Not Violated
$(0.7, 0.3)$	82.02%	Converged	Not Violated
$(0.5, 0.5)$	23.37%	Not Converged	Not Violated

Instead of using gradient ascent to directly maximize the objective, this method also provides a surrogate objective function that allows for a conservative approximation of how much $J(\pi_\theta)$ will change from the current policy as a result of the parameter update.

The performance of policy learning is assessed by tracking the progress of the obtained reward during the training. Figure 5.2 depicts the learning curve for our area coverage policy. We compute the mean and standard deviation of normalized reward scores for each $15K$ iteration step in the total of $1.5M$ iterations of training that takes approximately 22 hours on a single GPU. As seen in the figure, the learning curve converges after $1M$ iterations indicating that the policy learns to achieve the maximum cumulative reward after this point.

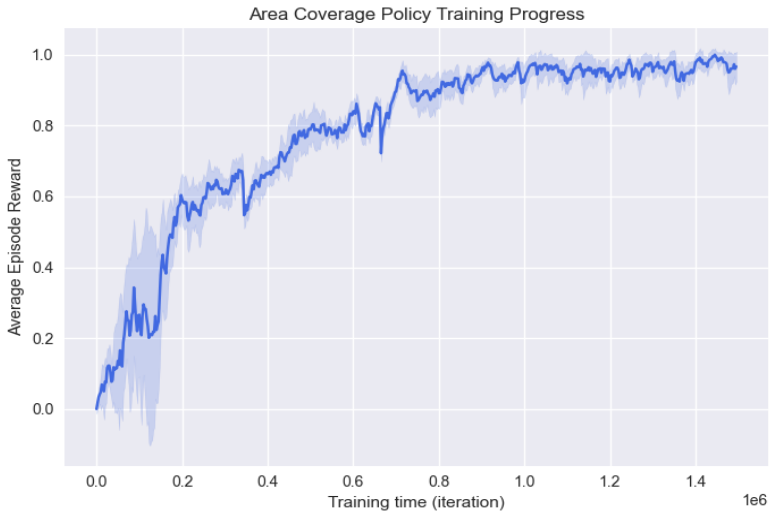


Figure 5.2: Learning curve for the proposed area coverage policy trained on a stomach model. Bold line and shaded areas depict the mean and standard deviation of the reward.

We trained the area coverage policy on a single stomach instance and evaluated it on two different stomach models. Figure 5.3 presents the performance of the policy for each organ during the inference session. It shows the coverage percentage and covered surfaces of the organs at 10 time instances, where each time interval is 15 seconds. Part a corresponds to the inference results on the training stomach model. In this

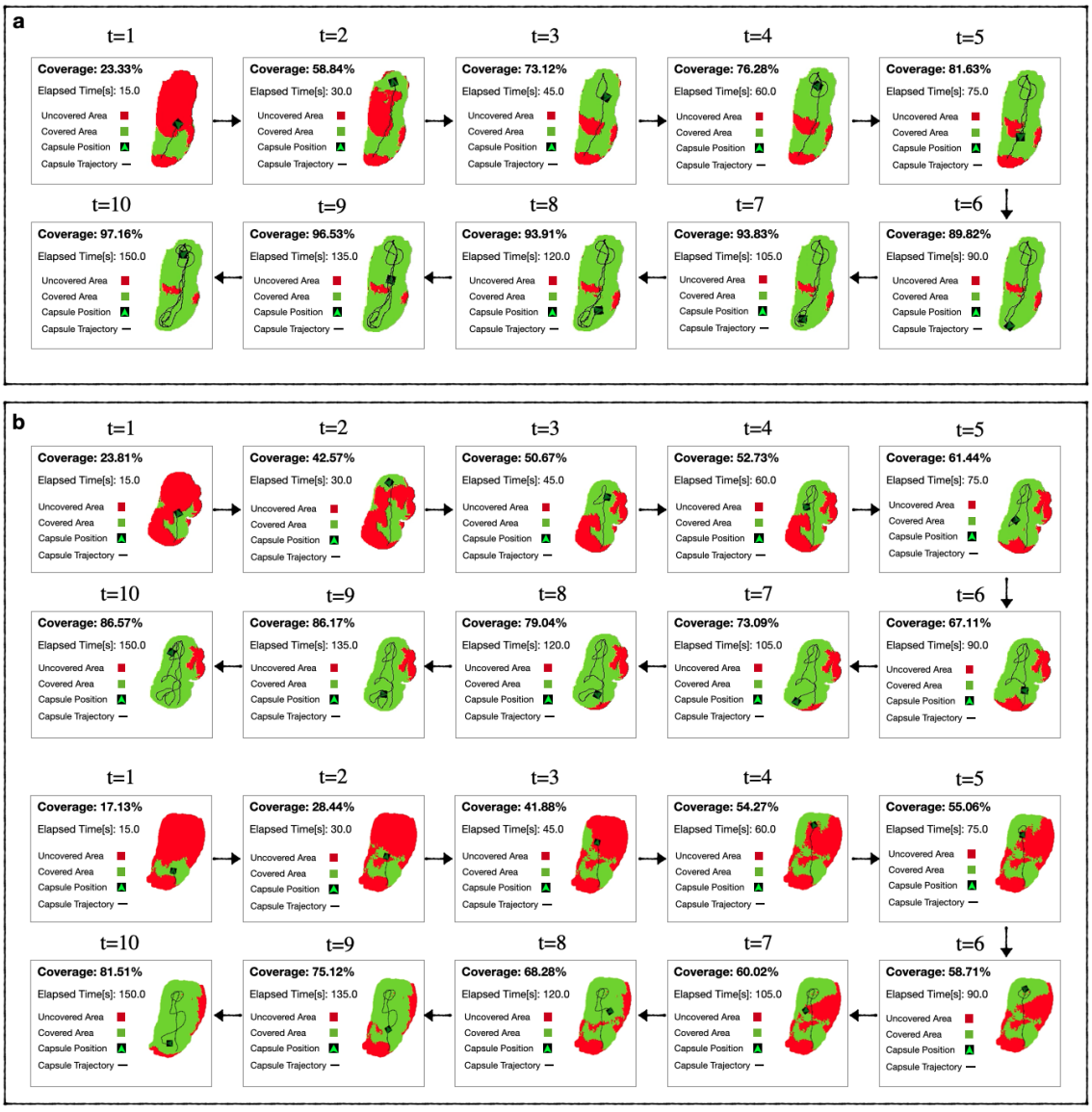


Figure 5.3: The area coverage status shown at 10 time steps during the policy inference process. Each time step t is 15 seconds. The inference results for a) the training stomach model and b) the two different test stomach models with different size and geometry are displayed.

case, the policy learned to cover 81.63% of the model in 75 seconds and complete the

task with the coverage of 97.16% in 150 seconds. In part b, the evaluation of the policy on two test stomach models with different sizes and geometry is shown, and the coverage status is recorded at the same time instances with the training. For the first test stomach, the policy was able to cover 61.44%, 86.17% of the organ in 75 and 150 seconds respectively. In the second test case, the policy achieved the coverage rates of 55.06%, 81.51% for the same time period. Despite a slight performance gap between training and test cases, our method still indicates a promising coverage performance in unseen stomach models. In all cases, the capsule robot followed different trajectories depending on the size and the shape of the organs indicating the adaptation of the policy to novel environments.

We also compare our method with three baseline path planning approaches. The first one is random planning where the magnetic actuator takes a random action at each time step. The second approach is grid-based coverage path planning where the path planner passes through every grid in a zigzag fashion on the rectangular search space. The third method also has a similar coverage strategy with the grid-based method but follows a more sparse path by simply having margins $(w/8, h/8)$ from the width and height of the search area respectively. The final coverage approach is called the Spiral Algorithm presented in [62]. These approaches have been evaluated in the same 150 seconds time interval used in our approach and the coverage results of each method at the end of the time are given in the table 5.2. In this case, the coverage performances of grid and region-based methods have the range of 46.8% and 57.5% for all the three stomach instances and random path planning has a very poor coverage performance (11.3% to 34.6%) as expected. The spiral algorithm also shows a similar performance with grid and region based methods except for one of the test organs, there it has 73.97% coverage which is close to the coverage of our method 81.81%. In general, our method outperforms the conventional area coverage path planning approaches. Besides, Figure 5.4 shows three plots representing the increase in the coverage for each method and organ over time. The plots are generated by recording the coverage statuses at every 15 seconds. At earlier stages, the region-based method shows a better coverage performance than the grid-based method, yet it has been surpassed in the later time steps. Although the region-based coverage and spiral methods demonstrate comparable

results in some time intervals, in general, it can be concluded that our method performs considerably better than the baseline approaches at all time steps.

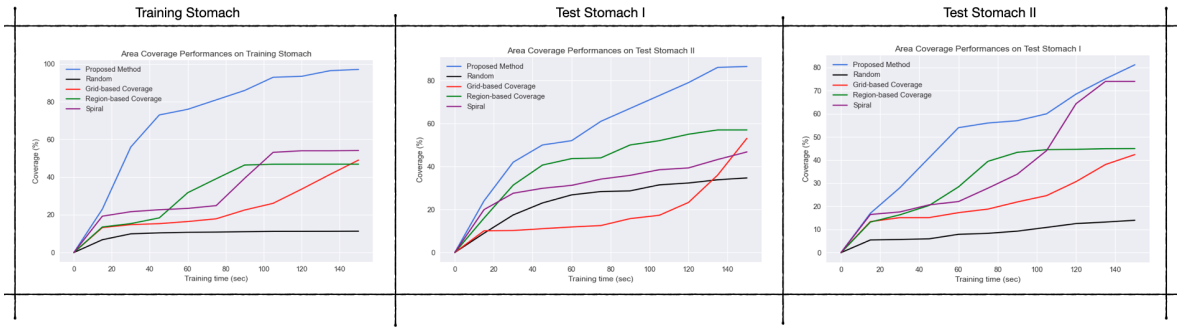


Figure 5.4: Plots showing the comparison of coverage results in percentage for stomach organ models as the time progresses.

Table 5.2: Comparison of coverage path planning results.

Environment	Random	Grid-based	Region-based	Spiral	Ours
Training Stomach	11.32%	49.05%	46.87%	54.12%	97.18%
Test Stomach I	14.02%	49.52%	44.97%	46.77%	86.57%
Test Stomach II	34.66%	53.08%	57.53%	73.97%	81.81%

5.2. Dataset

We have trained the proposed visual odometry and depth learning network on the synthetic endoscopic dataset that we have created using our simulation environment. The training dataset contains 6117 unlabeled image sequences that are recorded inside the digestive tract under realistic conditions. The learning of the network heavily depends on the disparity between two consecutive frames due to the loss functions L_p and L_{GC} defined in Section 4.2.1. In this regard, to provide enough disparities between consecutive frames, we sampled each of the third frames from the dataset which results in a total of 2039 images used for the training. Besides training data we have created three test datasets where each contains 200 images recorded inside the stomach, small intestine, and colon organs. After jointly training the pose and the depth networks, they can separately be used for inference. In the following sections, we present pose and depth estimation results of these networks on the test datasets.

5.3. Pose Estimation Results

The pose estimation performance is numerically evaluated based on the following metrics: Absolute Trajectory Error (ATE), Translational Relative Pose Error in translation (RPE_{trans}), and rotation (RPE_{rot}). The ATE measures the root mean square error (RMSE) between points of the pre-aligned trajectories averaged over all time steps. It can be denoted as

$$ATE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d(p_i, \hat{p}_i)^2} \tag{5.1}$$

where $p, \hat{p} \in R^3$ and $d(p, \hat{p}) = \sqrt{(p_1 - \hat{p}_1)^2 + (p_2 - \hat{p}_2)^2 + (p_3 - \hat{p}_3)^2}$ is the euclidean distance between true and predicted points.

The second metric RPE evaluates the relative drift between the points of two aligned trajectories over a fixed time step Δ. We can define the RPE_i between the points $(p_i, \hat{p}_i)$ over Δ as

$$RPE_i = (p_i^{-1} p_{i+\Delta})^{-1} (\hat{p}_i^{-1} \hat{p}_{i+\Delta}), \tag{5.2}$$

then the RPE_{trans} and the RPE_{rot} can be calculated as

$$RPE_{trans} = \frac{1}{n}\sum_{i=1}^{n} ||RPE_i||, \tag{5.3}$$

$$RPE_{rot} = \frac{1}{n}\sum_{i=1}^{n} \angle rot(RPE_i) \tag{5.4}$$

where $rot(RPE_i)$ denotes the rotation matrix of RPE_i and $\angle$ is the angle of the matrix. The results are reported in the case of $\Delta = 1$, which corresponds to calculating a relative drift per frame.

Figure 5.5 visually presents the ground-truth and predicted trajectories of the developed method Scale Consistent in comparison with the similar self-supervised pose estimation approaches used in Endo Odometry [12] and Endo Depth [23] papers. The numerical results showing the performance of all the methods for each test dataset are given in Table 5.3. In the colon and small intestine test cases, although the Endo Depth method is slightly better in RPE_{rot} metric our method performs the best in the mean ATE and RPE_{trans} metrics. The Endo Odometry method also shows comparable results yet it has the least performance in all metrics compared to other methods. For the stomach data, although our Scale Consistent method shows reasonable results, the Endo Depth method performs the best in all metrics and the Endo Odometry approach has a higher error and fails to predict a relatively accurate trajectory. In general, our Scale Consistent method shows the best results in translational error metrics for two of the test cases, yet the Endo Depth approach performs better in the rotational error. It can be concluded that both the Scale Consistent and Endo Depth approaches have comparable performances in the endoscopic pose estimation task and perform better than the Endo Odometry method. Also, we partly observe some deviations along the predicted trajectories compared to their ground-truth counterparts. It can be seen that all the methods suffer from such deviations that are usually caused by the strong illumination cases happening when the capsule robot becomes very close to the wall of the organs. Since feature points disappear in such strong illumination cases the model fails to track the camera pose, thus produces erroneous results.

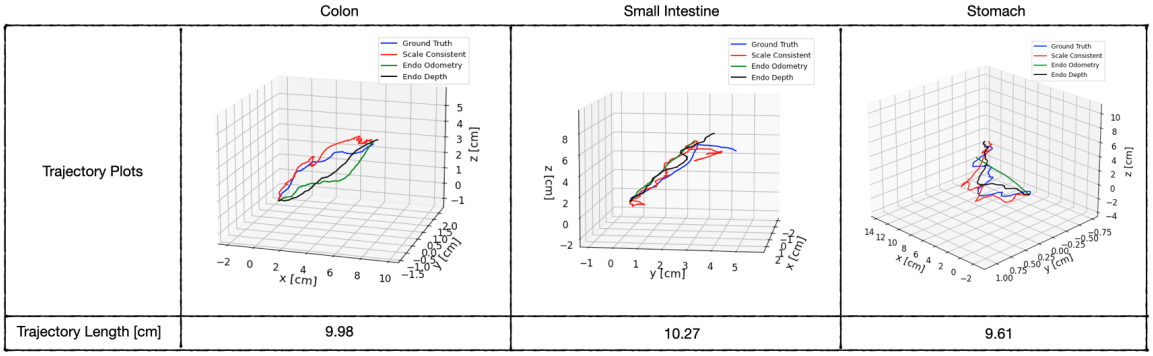


Figure 5.5: Pose estimation results for test stomach, small intestine and colon datasets. True and predicted trajectory plots are comparatively presented for each method. Trajectory lengths are also provided.

Table 5.3: Comparison of pose estimation results.

Colon			
Metrics	Endo Odom. [12]	Endo Depth [23]	Scale Consistent
ATE ($\mu \pm \sigma$) [cm]	0.94 ± 0.70	0.83 ± 0.62	0.78 ± 0.67
Trans. RPE ($\mu \pm \sigma$) [cm]	0.41 ± 0.18	0.21 ± 0.10	0.19 ± 0.07
Rot. RPE ($\mu \pm \sigma$) [deg]	2.26 ± 1.62	1.69 ± 0.81	1.86 ± 0.88
Small Intestine			
Metrics	Endo Odom. [12]	Endo Depth [23]	Scale Consistent
ATE ($\mu \pm \sigma$) [cm]	0.75 ± 0.72	0.69 ± 0.62	0.62 ± 0.50
Trans. RPE ($\mu \pm \sigma$) [cm]	0.41 ± 0.25	0.30 ± 0.12	0.23 ± 0.09
Rot. RPE ($\mu \pm \sigma$) [deg]	2.31 ± 1.18	1.82 ± 0.81	2.05 ± 0.86
Stomach			
Metrics	Endo Odom. [12]	Endo Depth [23]	Scale Consistent
ATE ($\mu \pm \sigma$) [cm]	1.52 ± 0.63	0.57 ± 0.45	0.65 ± 0.51
Trans. RPE ($\mu \pm \sigma$) [cm]	0.89 ± 0.53	0.07 ± 0.01	0.13 ± 0.07
Rot. RPE ($\mu \pm \sigma$) [deg]	4.71 ± 3.38	2.22 ± 1.59	2.51 ± 1.60

5.4. Depth Estimation Results

We qualitatively assess the depth estimation performance of the trained model for each organ. Figure 5.6 comparably illustrates predicted and ground-truth depth of sample images taken from the synthetic test datasets. The depth images are normalized into $[0, 1]$ scale and converted to the jet colormap for better visualization. Since the depth information of each pixel recorded in the simulation environment is known, we are able to generate very precisely and fine-grained ground truth depth maps. As seen in the figure, unsupervised training of the proposed depth network produces effective results compared with the true depth images.

We have further validated the depth estimation method on real endoscopic images. We considered three datasets Kvasir Stomach, Kvasir Ulcerative Colitis [49], and Red Lesion [48] to acquire sample images for the stomach, small intestine, and colon respectively. Since ground-truth depth images are not available for these real en-

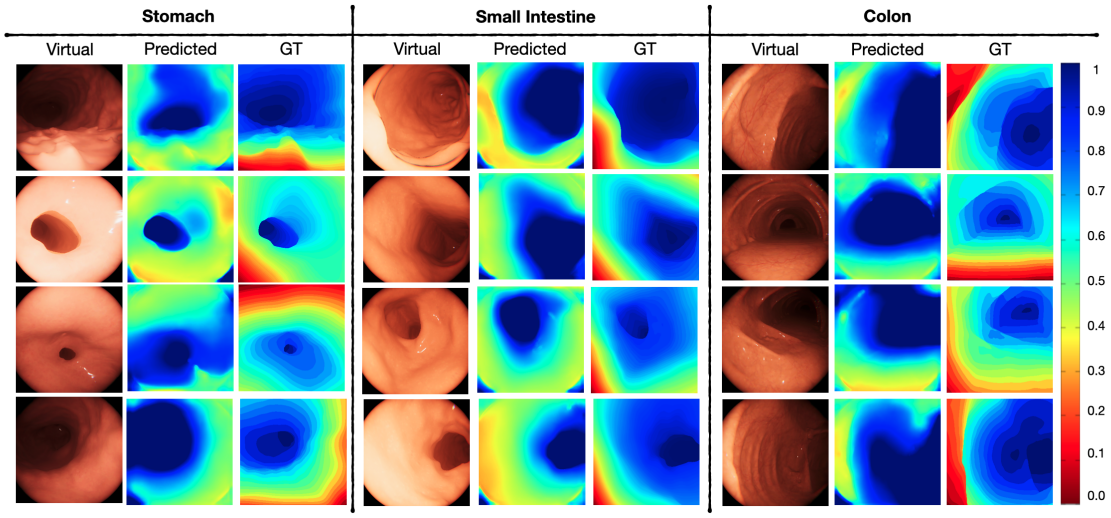


Figure 5.6: Predicted depth images on synthetic test datasets for each gastrointestinal organ compared to their ground-truth counterparts. Depth images are represented in jet color map ranging from 0 to 1 (nearest to farthest) for better visualization.

doscopy datasets we present raw images along with their predicted depth maps which can be seen in Figure 5.7. Qualitative results indicate that our depth network trained with synthetic data performs comparably well in real endoscopic images for depth estimation tasks. Hence, it implies that the synthetic data generated via our simulation environment is also effective in real-world data for the depth estimation task.

5.5. Surface Reconstruction Results

The performance of the proposed 3D dense surface reconstruction method described in the Section 4.3 is qualitatively evaluated on monocular capsule endoscopy video sequences recorded in the simulation. Figure 5.8 illustrates the reconstructed surfaces for stomach and colon organs in two scenarios. For the first scenario, we reconstructed the surface from a sample video sequence using predicted pose and depth images. For the second scenario, we reconstructed the same surface with the ground-truth pose and depth values to make a comparison. Although partial distortions occur during the reconstruction process due to the noise in the predicted depth maps, the results indicate an effective reconstruction performance compared to the ground-truth scenarios.

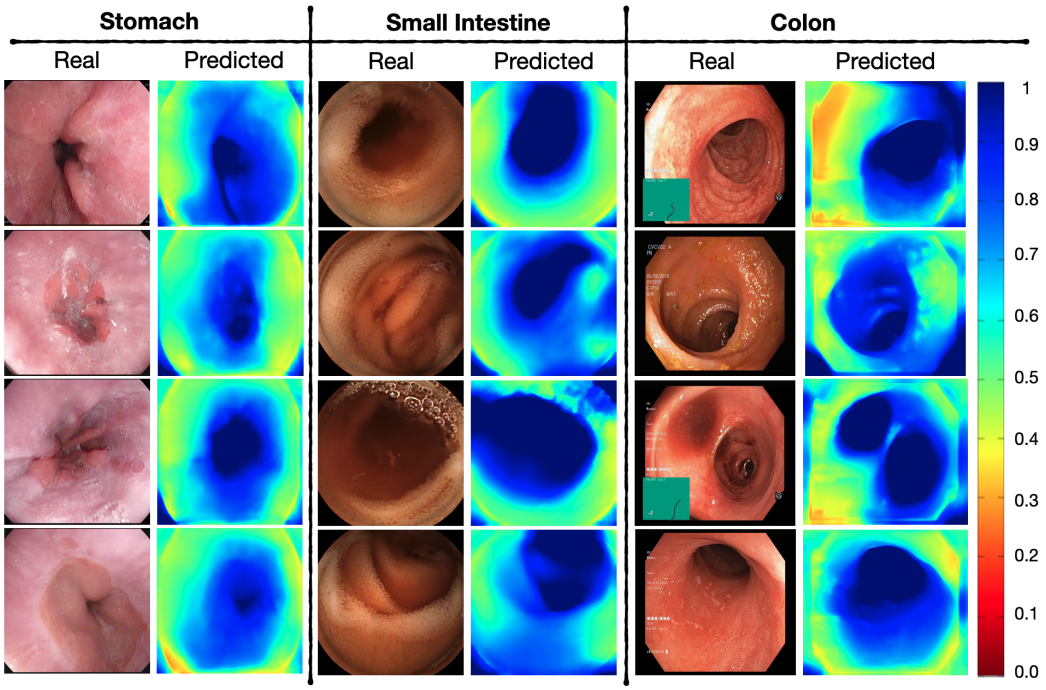


Figure 5.7: Depth estimation results on real endoscopic images. The depth estimation network trained only on the synthetic data is tested on real endoscopy datasets for the stomach small intestine and colon.

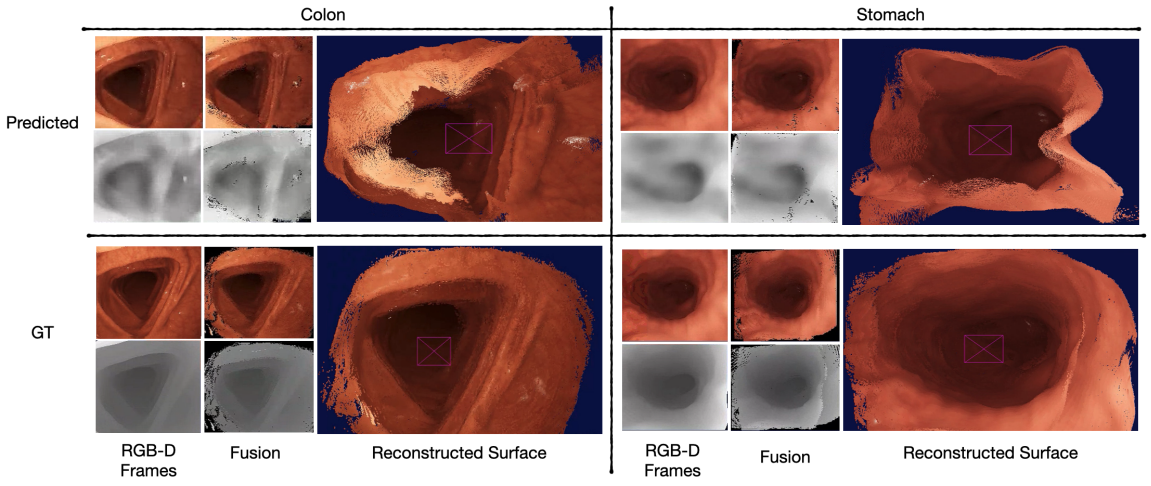


Figure 5.8: Qualitative evaluation of the surface reconstruction method on a sample set of stomach and colon test datasets. The reconstruction results using the estimated depth and the pose values (top) are visually compared to the results when ground-truth depth values are used (bottom).

6. CONCLUSION

In this study, we introduced a novel methodology for autonomous navigation of actively controlled capsule robots to maximize the scanned area in the gastrointestinal tract. We trained a deep reinforcement model to learn a stochastic area coverage policy that determines the action of the magnetic actuator based on an environment state. We proposed a hybrid neural network architecture that provides an expressive feature representation of both visual camera observations and other sensory inputs to define the mapping between the features and the corresponding action vector. We also designed a novel reward function that is based on the amount of newly monitored areas and the velocity of the capsule robots to keep it within the safety limits.

Results show that the proposed policy achieved 97% of coverage rate in the training stomach model and more than 80% in the other two test stomach models within 150 seconds at inference sessions. The policy produced reasonable trajectories for all cases and without much redundancies. Also, our method significantly outperformed the baseline approaches including grid-based, region-based and spiral coverage. On the other hand, the performance gap between training and test cases indicates a slight over-fitting on the trained model. In future work, we want to tackle this problem by applying the multi-agent learning paradigm, in which multiple robotic capsules on different organ instances are simultaneously trained. Another future work for the area coverage policy would be to use imitation learning concept. The model can learn the navigation of a capsule robot by mimicking the behaviors of gastroenterologists from several real active capsule endoscopy procedures.

Besides, we demonstrated a self-supervised learning framework for the localization of capsule robots from a monocular RGB camera. We trained two encoder-decoder CNN for depth estimation and a single CNN encoder for pose estimation tasks. The pose network showed less than 1 cm mean ATE in all the test trajectories created for each GI organ. In comparison with the similar self-supervised visual odometry approaches, our method performs slightly better in translational ATE and RPE metrics

and showed comparable performance on these metrics in rotation. Besides, we found out that the model learns better when there is an apparent difference between consecutive image frames. For that reason, we sampled each third frame from the monocular endoscopic video sequence dataset used for training. We also noted that in strong illumination or reflection cases, the model fails to accurately track the camera pose which ends up deviations in the estimated pose track of the capsule. To handle such cases, optical flow features that represent the movements of brightness patterns of an image can be utilized in potential future work.

In the depth estimation task, the qualitative results demonstrate that predicted depth maps are comparable with their ground-truth correspondences and they perform reasonably well in the dense surface reconstruction task. The depth estimation model also shows promising performance on real endoscopic datasets indicating the effectiveness of the model in real-world tasks. In the future, we want to extensively validate the other approaches by conducting real-world experiments i.e. autonomously navigating a capsule robot inside an ex-vivo pig stomach, and address the problem of transferring learned patterns in the area coverage and pose estimation tasks to real scenarios.

Finally, we conclude the thesis with the future improvements for the proposed simulation environment. From the physical aspects, we plan to implement the deformation of the soft organ tissue, and contraction and relaxation motion of the intestinal organs occasionally happening during the endoscopy procedure. We also plan to make several visual improvements such as simulating liquid reflections, yellow and white stains, and floating particles partly appearing in GI organs for the aim of minimizing the gap between the virtual and real world.

REFERENCES

1. Siegel, R. L., K. D. Miller and A. Jemal, "Cancer Statistics, 2019", *CA: A Cancer Journal for Clinicians*, Vol. 69, No. 1, pp. 7–34, 2019.

2. Menon, S. and N. Trudgill, "How Commonly Is Upper Gastrointestinal Cancer Missed at Endoscopy? A Meta-analysis", *Endoscopy International Open*, Vol. 2, No. 2, p. E46, 2014.

3. Than, M., J. Witherspoon, J. Shami, P. Patil and A. Saklani, "Diagnostic Miss Rate for Colorectal Cancer: An Audit", *Annals of Gastroenterology: Quarterly Publication of the Hellenic Society of Gastroenterology*, Vol. 28, No. 1, p. 94, 2015.

4. Lee, J., S. W. Park, Y. S. Kim, K. J. Lee, H. Sung, P. H. Song, W. J. Yoon and J. S. Moon, "Risk Factors of Missed Colorectal Lesions After Colonoscopy", *Medicine*, Vol. 96, No. 27, 2017.

5. Carta, R., M. Sfakiotakis, N. Pateromichelakis, J. Thoné, D. Tsakiris and R. Puers, "A Multi-coil Inductive Powering System for an Endoscopic Capsule with Vibratory Actuation", *Sensors and Actuators A: Physical*, Vol. 172, No. 1, pp. 253–258, 2011.

6. Yim, S. and M. Sitti, "Design and Rolling Locomotion of a Magnetically Actuated Soft Capsule Endoscope", *IEEE Transactions on Robotics*, Vol. 28, No. 1, pp. 183–194, 2011.

7. Kim, L., S. C. Tang and S.-S. Yoo, "Prototype Modular Capsule Robots for Capsule Endoscopies", *13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 350–354, 2013.

8. Turan, M., Y. Almalioglu, H. Araujo, E. Konukoglu and M. Sitti, "Deep Endovo: A Recurrent Convolutional Neural Network (RCNN) Based Visual Odometry Ap-

proach for Endoscopic Capsule Robots", *Neurocomputing*, Vol. 275, pp. 1861–1870, 2018.

9. İncetan, K., I. O. Celik, A. Obeid, G. I. Gokceler, K. B. Ozyoruk, Y. Almalioglu, R. J. Chen, F. Mahmood, H. Gilbert, N. J. Durr *et al.*, "VR-Caps: A Virtual Environment for Capsule Endoscopy", *Medical Image Analysis*, Vol. 70, p. 101990, 2021.

10. Siegwart, R., I. R. Nourbakhsh and D. Scaramuzza, "Introduction to Autonomous Mobile Robots", pp. 10–11, MIT Press, 2004.

11. Turan, M., Y. Almalioglu, H. Araujo, E. Konukoglu and M. Sitti, "A Non-rigid Map Fusion-based Direct SLAM Method for Endoscopic Capsule Robots", *International Journal of Intelligent Robotics and Applications*, Vol. 1, No. 4, pp. 399–409, 2017.

12. Turan, M., E. P. Ornek, N. Ibrahimli, C. Giracoglu, Y. Almalioglu, M. F. Yanik and M. Sitti, "Unsupervised Odometry and Depth Learning for Endoscopic Capsule Robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1801–1807, 2018.

13. Mura, M., Y. Abu-Kheil, G. Ciuti, M. Visentini-Scarzanella, A. Menciassi, P. Dario, J. Dias and L. Seneviratne, "Vision-based Haptic Feedback for Capsule Endoscopy Navigation: A Proof of Concept", *Journal of Micro-Bio Robotics*, Vol. 11, No. 1-4, pp. 35–45, 2016.

14. Abu-Kheil, Y., O. Al Trad, L. Seneviratne and J. Dias, "A Proposed Clinical Evaluation of a Simulation Environment for Magnetically-Driven Active Endoscopic Capsules", *Robotics in Healthcare*, Vol. 1170, pp. 87–94, Springer, 2019.

15. Chen, R. J., T. L. Bobrow, T. Athey, F. Mahmood and N. J. Durr, "Slam Endoscopy Enhanced by Adversarial Depth Prediction", *arXiv preprint arXiv:1907.00283*, 2019.

16. Fireman, Z., D. Paz and Y. Kopelman, "Capsule Endoscopy: Improving Transit Time and Image View", *World Journal of Gastroenterology: WJG*, Vol. 11, No. 37, p. 5863, 2005.

17. Ren, Z., T. He, L. Peng, S. Liu, S. Zhu and B. Zeng, "Shape Recovery of Endoscopic Videos by Shape from Shading Using Mesh Regularization", *International Conference on Image and Graphics*, pp. 204–213, Springer, 2017.

18. Koulaouzidis, A., D. K. Iakovidis, D. E. Yung, E. Mazomenos, F. Bianchi, A. Karagyris, G. Dimas, D. Stoyanov, H. Thorlacius, E. Toth *et al.*, "Novel Experimental and Software Methods for Image Reconstruction and Localization in Capsule Endoscopy", *Endoscopy International Open*, Vol. 6, No. 2, p. E205, 2018.

19. Grasa, O. G., J. Civera, A. Guemes, V. Munoz and J. Montiel, "EKF Monocular SLAM 3D modeling, Measuring and Augmented Reality from Endoscope Image Sequences", *Medical Image Computing and Computer-assisted Intervention (MICCAI)*, Vol. 2, 2009.

20. Grasa, O. G., E. Bernal, S. Casado, I. Gil and J. Montiel, "Visual SLAM for Handheld Monocular Endoscope", *IEEE Transactions on Medical Imaging*, Vol. 33, No. 1, pp. 135–146, 2013.

21. Whelan, T., R. F. Salas-Moreno, B. Glocker, A. J. Davison and S. Leutenegger, "ElasticFusion: Real-time Dense SLAM and Light Source Estimation", *The International Journal of Robotics Research*, Vol. 35, No. 14, pp. 1697–1716, 2016.

22. Zhou, T., M. Brown, N. Snavely and D. G. Lowe, "Unsupervised Learning of Depth and Ego-motion from Video", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

23. Recasens, D., J. Lamarca, J. M. Fácil, J. Montiel and J. Civera, "Endo-Depth-and-Motion: Localization and Reconstruction in Endoscopic Videos Using Depth Networks and Photometric Constraints", *arXiv preprint arXiv:2103.16525*, 2021.

24. Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation", *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241, Springer, 2015.

25. Godard, C., O. Mac Aodha and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-right Consistency", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

26. Godard, C., O. Mac Aodha, M. Firman and G. J. Brostow, "Digging into Self-supervised Monocular Depth Estimation", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019.

27. Bian, J., Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng and I. Reid, "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video", *Advances in Neural Information Processing Systems*, pp. 35–45, 2019.

28. Aghanouri, M., A. Ghaffari and N. D. Serej, "Image Based High-level Control System Design for Steering and Controlling of an Active Capsule Endoscope", *Journal of Intelligent & Robotic Systems*, Vol. 94, No. 1, pp. 115–134, 2019.

29. Turan, M., Y. Almalioglu, H. B. Gilbert, F. Mahmood, N. J. Durr, H. Araujo, A. E. Sarı, A. Ajay and M. Sitti, "Learning to Navigate Endoscopic Capsule Robots", *IEEE Robotics and Automation Letters*, Vol. 4, No. 3, pp. 3075–3082, 2019.

30. Hert, S., S. Tiwari and V. Lumelsky, "A Terrain-covering Algorithm for an AUV", *Underwater Robots*, pp. 17–45, Springer, 1996.

31. Choset, H., "Coverage for Robotics: A Survey of Recent Results", *Annals of Mathematics and Artificial Intelligence*, Vol. 31, No. 1-4, pp. 113–126, 2001.

32. Acar, E. U., H. Choset, Y. Zhang and M. Schervish, "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods", *The International Journal of Robotics Research*, Vol. 22,

No. 7-8, pp. 441–466, 2003.

33. Baek, S., T.-K. Lee, O. Se-Young and K. Ju, "Integrated Online Localization, Mapping and Coverage Algorithm of Unknown Environments for Robotic Vacuum Cleaners Based on Minimal Sensing", *Advanced Robotics*, Vol. 25, No. 13-14, pp. 1651–1673, 2011.

34. Yang, T., P. Li, H. Zhang, J. Li and Z. Li, "Monocular Vision SLAM-based UAV Autonomous Landing in Emergencies and Unknown Environments", *Electronics*, Vol. 7, No. 5, p. 73, 2018.

35. Ferguson, D., M. Likhachev and A. Stentz, "A Guide to Heuristic-based Path Planning", *Proceedings of the International Workshop on Planning Under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 9–18, 2005.

36. Garrett, C. R., T. Lozano-Pérez and L. P. Kaelbling, "FFRob: An Efficient Heuristic for Task and Motion Planning", *Algorithmic Foundations of Robotics XI*, pp. 179–195, Springer, 2015.

37. Valencia, R. and J. Andrade-Cetto, "Path Planning in Belief Space with Pose SLAM", *Mapping, Planning and Exploration with Pose SLAM*, pp. 53–87, Springer, 2018.

38. Schirmer, R., P. Biber and C. Stachniss, "Coverage Path Planning in Belief Space", *International Conference on Robotics and Automation (ICRA)*, pp. 7604–7610, 2019.

39. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level Control through Deep Reinforcement Learning", *Nature*, Vol. 518, No. 7540, pp. 529–533, 2015.

40. Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous Control with Deep Reinforcement Learning", *arXiv preprint arXiv:1509.02971*, 2015.

41. Zhu, Y., R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei and A. Farhadi, "Target-driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning", *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3357–3364, 2017.

42. Gupta, S., J. Davidson, S. Levine, R. Sukthankar and J. Malik, "Cognitive Mapping and Planning for Visual Navigation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2017.

43. Mirowski, P., R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, "Learning to Navigate in Complex Environments", *arXiv preprint arXiv:1611.03673*, 2016.

44. Chen, T., S. Gupta and A. Gupta, "Learning Exploration Policies for Navigation", *arXiv preprint arXiv:1903.01959*, 2019.

45. Gaidon, A., Q. Wang, Y. Cabon and E. Vig, "Virtual Worlds as Proxy for Multi-object Tracking Analysis", *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2016.

46. Tremblay, J., A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon and S. Birchfield, "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 969–977, 2018.

47. Toyoshima, O., T. Nishizawa, M. Arita, Y. Kataoka, K. Sakitani, S. Yoshida, H. Yamashita, K. Hata, H. Watanabe and H. Suzuki, "Helicobacter Pylori Infection in Subjects Negative for High Titer Serum Antibody", *World Journal of*

Gastroenterology, Vol. 24, No. 13, p. 1419, 2018.

48. [dataset] Coelho, P., A. Pereira, M. Salgado and A. Cunha, "A Deep Learning Approach for Red Lesions Detection in Video Capsule Endoscopies", *International Conference Image Analysis and Recognition*, pp. 553–561, Springer, 2018.

49. Pogorelov, K., K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, M. Riegler and P. Halvorsen, "KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection", *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pp. 164–169, ACM, New York, NY, USA, 2017, http://doi.acm.org/10.1145/3083187.3083212.

50. Clark, K., B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository", *Journal of Digital Imaging*, Vol. 26, No. 6, pp. 1045–1057, 2013.

51. Accoto, D., C. Stefanini, L. Phee, A. Arena, G. Pernorio, A. Menciassi, M. Carrozza and P. Dario, "Measurements of the Frictional Properties of the Gastrointestinal Tract", *World Tribology Congress*, Vol. 3, 2001.

52. Wang, K. and G. Yan, "Research on Measurement and Modeling of the Gastro Intestine's Frictional Characteristics", *Measurement Science and Technology*, Vol. 20, p. 015803, 2008.

53. Zhang, C., H. Liu, R. Tan and H. Li, "Modeling of Velocity-dependent Frictional Resistance of a Capsule Robot Inside an Intestine", *Tribology Letters*, Vol. 47, pp. 295–301, 2012.

54. Newville, M., T. Stensitzki, D. B. Allen, M. Rawlik, A. Ingargiola and A. Nelson, "LMFIT: Non-linear Least-square Minimization and Curve-fitting for Python", *Astrophysics Source Code Library*, pp. ascl–1606, 2016.

55. Shamsudhin, N., V. I. Zverev, H. Keller, S. Pane, P. W. Egolf, B. J. Nelson and A. M. Tishin, "Magnetically Guided Capsule Endoscopy", *Medical Physics*, Vol. 44, No. 8, pp. e91–e111, 2017.

56. Espeholt, L., H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning *et al.*, "Impala: Scalable Distributed Deep-rl with Importance Weighted Actor-learner Architectures", *arXiv preprint arXiv:1802.01561*, 2018.

57. Zhan, H., R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal and I. Reid, "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018.

58. Li, R., S. Wang, Z. Long and D. Gu, "Undeepvo: Monocular Visual Odometry through Unsupervised Deep Learning", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7286–7291, 2018.

59. Fischer, P., A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers and T. Brox, "Flownet: Learning Optical Flow with Convolutional Networks", *arXiv preprint arXiv:1504.06852*, 2015.

60. Fehn, C., "Depth-image-based Rendering (DIBR), Compression, and Transmission for a New Approach on 3D-TV", *Stereoscopic Displays and Virtual Reality Systems XI*, Vol. 5291, pp. 93–104, International Society for Optics and Photonics, 2004.

61. Schulman, J., F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal Policy Optimization Algorithms", *arXiv preprint arXiv:1707.06347*, 2017.

62. Gonzalez, E., O. Alvarez, Y. Diaz, C. Parra and C. Bustacara, "BSA: A Complete Coverage Algorithm", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2040–2044, 2005.