1 2 90

UNIVERSIDADE Đ
COIMBRA

Inês Dinis Félix

DEEP LEARNING FOR MARKERLESS SURGICAL NAVIGATION IN ORTHOPEDICS

Dissertação no âmbito do Mestrado Integrado em Engenharia Biomédica orientada pelo Professor Doutor João Pedro de Almeida Barreto e apresentada ao Departamento de Física da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2020

1 2 9 0
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Inês Dinis Félix

Deep Learning for Markerless Surgical Navigation in Orthopedics

Thesis submitted to the
University of Coimbra for the degree of
Master in Biomedical Engineering

Supervisors:
Michel Antunes
Carolina Raposo
J. P. Barreto

Coimbra, 2020

This work was developed in collaboration with:

P3d

PERCEIVE3D

Acknowledgments

Gostaria de expressar o meu agradecimento às seguintes pessoas, que de forma direta ou indireta, foram essenciais no meu percurso e na execução deste projeto.

Ao Michel e à Carolina, por toda a paciência, todo o apoio e pela partilha constante de conhecimento ao longo do último ano. Sem vocês, o resultado final não seria este. Ao Pedro Rodrigues, pela introdução ao tema. Ao professor João Barreto, pela oportunidade, orientação e tempo disponibilizado. E também à restante equipa da Perceive3D por me ter recebido tão bem, desde o primeiro momento. Vai ser um prazer para mim integrá-la no futuro.

A um nível mais pessoal, a todos os meus amigos, os mais antigos e os que tive oportunidade de conhecer durante os últimos 5 anos. Ao Marco, por estar sempre ao meu lado. E finalmente à minha família, e em especial ao meu pai, que tornou tudo isto possível.

Resumo

A Artroplastia Total do Joelho (ATJ) é um procedimento cirúrgico realizado em pacientes que sofrem de artrite do joelho. O posicionamento correcto dos implantes está fortemente relacionado com múltiplas variáveis cirúrgicas que têm um impacto tremendo no sucesso da cirurgia. Foram investigados e desenvolvidos sistemas de navegação baseados em computador, com o objetivo de auxiliar o cirurgião a controlar, com precisão, essas variáveis cirúrgicas. Esta tese centra-se na navegação em ATJ e aborda dois problemas que são apontados por muitos como fundamentais para a sua adoção consensual.

O primeiro problema é que as tecnologias existentes são muito dispendiosas e requerem incisões ósseas adicionais para a fixação de marcadores, geralmente muito volumosos, interferindo com o típico fluxo cirúrgico. Este trabalho apresenta um sistema de navegação sem marcadores que apoia o cirurgião na execução precisa do procedimento de ATJ. O sistema proposto utiliza uma câmara RGB-D móvel para substituir os sistemas de navegação ópticos existentes, eliminando a necessidade de marcadores. A metodologia apresentada combina uma abordagem eficaz baseada em *Deep Learning* para segmentar com precisão a superfície óssea com um algoritmo robusto baseado na geometria para registar os ossos com modelos pré-operatórios. O desempenho favorável da nossa metodologia é alcançado através (1) do uso de uma estratégia semi-supervisionada para gerar dados de treino a partir de dados reais de cirurgia ATJ, (2) utilizando técnicas eficazes de aumento de dados para melhorar a capacidade de generalização, e (3) utilizando dados de profundidade adequados. A utilidade deste método completo de registo sem marcadores, que generaliza para diferentes dados intra-operatórios, é evidente e os resultados experimentais mostram um desempenho promissor para ATJ baseada em vídeo.

O segundo problema está relacionado com a falta de precisão na localização de pontos de referência no joelho durante a navegação, o que pode levar a erros significativos no posicionamento dos implantes. Esta tese apresenta um método de prova de conceito que utiliza *Deep Learning* para a detecção automática dos pontos de referência apenas a partir de imagens. O objetivo é fornecer sugestões em tempo real para auxiliar o cirurgião nesta tarefa, o que pode ser útil na tomada de decisões e na redução da variabilidade. A validação experimental num ponto de referência mostra que o método atinge resultados fiáveis, podendo ser feita a sua aplicação aos restantes pontos de referência.

Palavras-chave: Navegação cirúrgica; *Deep Learning*; Segmentação de imagem; Estimação de pose; Cirurgia no joelho.

Abstract

Total Knee Arthroplasty (TKA) is a surgical procedure performed in patients suffering from knee arthritis. The correct positioning of the implants is strongly related to multiple surgical variables that have a tremendous impact on the success of the surgery. Computer-based navigation systems have been investigated and developed in order to assist the surgeon in accurately controlling those surgical variables. This thesis focuses in navigation for TKA and addresses two problems that are pointed by many as fundamental for its broader acceptance.

The first problem is that existing technologies are very costly, require additional bone incisions for fixing markers to be tracked, and these markers are usually bulky, interfering with the standard surgical flow. This work presents a markerless navigation system that supports the surgeon in accurately performing the TKA procedure. The proposed system uses a mobile RGB-D camera for replacing the existing optical tracking systems and does not require markers to be tracked. We combine an effective deep learning-based approach for accurately segmenting the bone surface with a robust geometry-based algorithm for registering the bones with pre-operative models. The favorable performance of our pipeline is achieved by (1) employing a semi-supervised labeling approach for generating training data from real TKA surgery data, (2) using effective data augmentation techniques for improving the generalization capability, and (3) using appropriate depth data. The construction of this complete markerless registration prototype that generalizes for unseen intra-operative data is non-obvious, and relevant insights and future research directions can be derived. The experimental results show encouraging performance for video-based TKA.

The second problem is related to the lack of accuracy in localizing landmarks during image-free navigation, that can lead to significant errors in implant positioning. This thesis presents a proof-of-concept method that uses deep learning for automatic detection of landmarks from only visual input. The aim is to provide real time suggestions to assist the surgeon in this task, which can be useful in decision making and to reduce variability. Experimental validation with one landmark shows that the method achieves reliable results, and extension to the remaining landmarks can be extrapolated.

Keywords: Surgical Navigation; Deep Learning; Image Segmentation; Pose Estimation; Knee Surgery.

Acronyms

Adam Adaptive momentum estimation.

AI Artificial Intelligence.

AR Average Recall.

ATJ Artroplastia Total do Joelho.

BOP Benchmark for Object Pose Estimation.

CAS Computer-Aided Surgery.

CNN Convolutional Neural Networks.

CT Computed Tomography.

CV Computer Vision.

DL Deep Learning.

EMT Electromagnetic Tracking.

FCN Fully Convolutional Network.

FN False Negatives.

FP False Positives.

FPN Feature Pyramid Network.

fps Frames Per Second.

GPU Graphics Processing Unit.

ICP Iterative Closest Point.

IoU Intersection over Union.

mAP Mean Average Precision.

ML Machine Learning.

MRI Magnetic Resonance Imaging.

MS COCO Microsoft Common Objects in Context.

OT Optical Tracking.

P3D Perceive3D S.A..

PnP Perspective-n-Point.

px pixels.

RANSAC RANdom SAmple Consensus.

ReLU Rectified Linear Unit.

RGB Red-Blue-Green.

RGB-D Red-Blue-Green and Depth.

RMSprop Root mean square prop.

ROI Regions of Interest.

RPN Region Proposal Network.

TKA Total Knee Arthroplasty.

TP True Positives.

VT Video Tracking.

List of Figures

List of Tables

Contents

1

Introduction

Computer-Aided Surgery (CAS) has improved the effectiveness of a large variety of orthopaedic procedures, such as total knee and hip arthoplasties [4, 5], spinal surgery [6] and arthroscopic interventions [7]. CAS allows the surgeon to get real-time feedback about the relative positioning of surgical instruments with respect to the anatomies of the patient, allowing the practitioner to more easily inspect and visualize the patient's anatomies, as well as get support in following a particular surgical plan [8, 9].

Total Knee Arthroplasty (TKA) is the main choice for improving the quality of life of patients suffering from knee arthritis [10]. There is a particular set of surgical variables (e.g. implant component alignment, soft-tissue balancing, lower leg alignment) that is important for the success of a TKA intervention, and not controlling these variables accurately can lead to pain, knee instability and even periprosthetic fractures [11]. Traditionally, these variables are controlled manually with mechanical instruments, requiring many years of training and experience for accurately combining all these surgical variables into an appropriate implant plan [2]. From the literature, satisfactory component and lower limb alignment is achieved only within 3 degrees of varus/valgus relative to mechanical axis [12, 13, 14], while the translation "safe zone" for implant sizing and joint line variation is +/- 3-4 mm [15, 16]. In order to assist the surgeon in appropriately determining these variables, several computer navigation systems have been developed [11, 17].

The navigation systems can be usually divided into two main groups: image-based navigation, whose main objective is to align a pre-operative model with the intra-operative anatomical data such that a pre-specified surgical plan is followed; and image-free navigation, which requires the acquisition of particular bone landmarks which are used for determining the surgical plan [18].

Existing navigation solutions for TKA establish the relation between anatomy and tools through a tracking technology that enables to determine relative 3D poses in real-time. There are three different tracking technologies: (1) Electromagnetic Tracking (EMT): utilizes magnetic fields to determine the pose of sensors for measuring magnetic flux [19]; (2) Optical Tracking (OT): comprised by a workstation that controls the system with infrared cameras for tracking the position and orientation of markers attached to the femur, tibia and surgical instruments [20]; (3) Video Tracking (VT): the most recent technology, originally developed for navigation in arthroscopy [7], uses a monocular RGB camera to detect markers attached to the bones and instruments and estimate their relative pose in 3D. In terms of accuracy, it has been shown in [21] that optical tracking is more accurate than EMT, and [22] states that video-based tracking is at least as accurate as optical tracking, the most widely used technology of all three solutions, when performing open surgery.

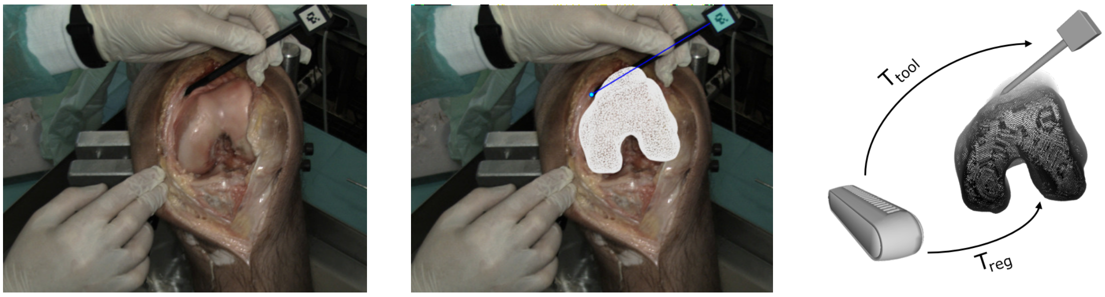
This thesis focuses in video-based navigation for TKA and addresses two problems that are pointed by many as fundamental for its broader acceptance:

1. The first problem concerns the attachment of markers to the anatomies, which is required by all existing tracking technologies used in CAS. The markers are bulky, can interfere with the surgical flow and their attachment consumes surgery time. This work presents a markerless image-based navigation system (Section 1.1) that supports the surgeon in accurately performing the TKA procedure.

2. It is well known in image-free navigation that the manual digitalization of landmarks is a time-consuming task and lacks accuracy [23, 24, 25]. This thesis presents a proof-of-concept method for automatic detection of landmarks from only visual input (Section 1.2). This could be relevant for guiding the surgeon in this task by providing real time suggestions.

1.1 Markerless video-based surgical navigation system

All tracking technologies require attaching markers to anatomy which consumes surgery time, are bulky and can interfere with the surgical flow and, more importantly, it endangers the patient by increasing risk of fracture in osteoporotic bone [26]. For these reasons, markerless navigation is highly desirable and can be the step to make navigation more widely used. Video-based tracking can be the step forward to accomplish markerless navigation if combined with a suitable method for visual 6D pose estimation able to determine the rotation and translation of targeted anatomies without the help of artificial fiducial markers (Figure 1.1).



(a) **(b)** **(c)**

Figure 1.1: Proposed video-based surgical navigation: **(a)** Frame with knee and tool; **(b)** Frame with overlaid 3D femur model and detected tool; **(c)** Video tracking using RGB-D camera allows to track bones and the tools, without requiring to fix markers to the bones.

The estimation of the 6D pose of a known object from an image is a long studied problem with the original P3P algorithm coming from the photogrammetry studies of the XIX century, and the topic having had substantial evolution in the last few years [27]. With the introduction of depth cameras, different methods have been presented for estimating the objects' pose from RGB-D images [28]. More recently, the dissemination of deep learning avoided the use of ad-hoc methods to establish explicit correspondences to be used as input in the geometric algorithms. Nevertheless, the pose estimation problem to be solved in this work is specially challenging because of (i) high medical accuracy requirements, (ii) robustness and resilience to large occlusions that often occur in the context of a TKA procedure, not only due to the tissues surrounding the targeted anatomy (bone) but also because of the tools and instruments used during the surgery, and (iii) real-time performance to know the location of the anatomy at every frame time instant.

The work of [2] is the first one trying to tackle these issues in the context of accomplishing markerless navigation for TKA. The work uses a commercial, off-the-shelf RGB-D camera and combines deep learning-based segmentation in RGB to locate the bone region and segment the point cloud, with a state-of-the-art method for 3D registration to align a pre-operative anatomical model of the patient such that the bone resections for the implant positioning can be guided according to a pre-operative plan. Despite showing promising results, the proposed system has some limitations: (1) bone surface segmentation from RGB images was trained on a limited set of femurs and no generalization analysis to bones not contained in the training set was performed, (2) it requires very specific training data that is scarce (intra-operative data with registered pre-operative 3D model), (3) the navigation of the proximal cut, which involves the segmentation and registration of the tibia, was not considered and is required for a successful TKA procedure, and (4) the registration errors (rotation error of 3.17 degrees and translation error of 6.18 mm) do not fulfil the medical accuracy requirements described previously (3 degrees/3-4 mm), and were obtained on a limited test set with data from knee joints also contained in the training set.

The work performed in this thesis tries to overcome the limitations of the prototype described in [2] and it is a clear step forward towards markerless navigation, demonstrating its feasibility in the near future. In particular, the contributions are:

1. A new approach for fast labeling which dramatically facilitates the generation of training data, without the need of tracking instrumentation and constant supervision;

2. Accurate femur and tibia segmentation from RGB images, showing high generalization capabilities to unseen bones from different TKA experiments;

3. The combination of robust and accurate bone segmentation with the use of a recent consumer RGB-D camera and a properly tuned registration algorithm, which leads to a system that is close to fulfilling the medical accuracy requirements. In this regard, we show that the most significant part of the error is due to the limited depth resolution arising from off-the-shelf sensors. Nevertheless, with the constant evolution of depth sensors performance [29], it is reasonable to anticipate that markerless navigation will be feasible with comercial, off-the-shelf sensors in the near future;

4. Extension to the segmentation and registration of the tibia, which presents additional difficulties when compared to the femur (e.g. more and larger occlusions).

1.2 Automatic detection of anatomical landmarks

Image-free systems require a data acquisition step, in which anatomical landmarks are gathered for navigation [30].

The acquisition of most anatomical landmarks is performed through manual digitalization using a marker tool (refer to Figure 1.2). This task is very challenging, time-consuming, demands high level of expertise and may lack accuracy [23, 24, 25]. Small errors in performing this step can lead to significant errors in the implant positioning [4]. Localization of landmarks also suffers from inter- and intra-observer

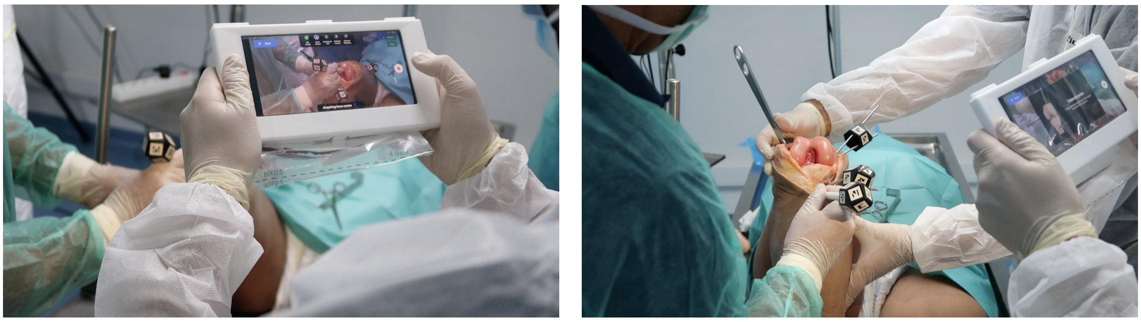


Figure 1.2: Acquisition of anatomical landmarks during a navigated TKA. Copyright by P3D.

variability [31]. For instance, Figure 1.3 shows localization of some landmarks acquired by an orthopaedic surgeon in different trials in the same knee.

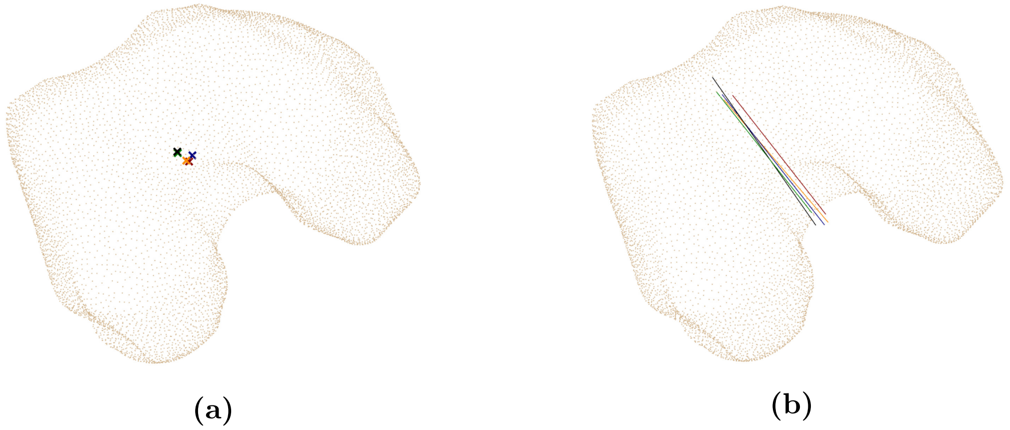


Figure 1.3: Femur landmarks **(a)** knee center and **(b)** whiteside's line acquired by an orthopaedic surgeon in 5 different trials (represented with different colors) in the same knee.

There are some works that aid the identification and detection of anatomical landmarks on 3D models [23, 24, 25], however, and to the best of our knowledge, none tackle yet the task of providing real time localization of the landmarks from input visual only in image-free navigation.

For this part of the thesis, it was developed a method for automatic detection and localization of bone landmarks in RGB images captured during the surgery, through using a Deep Learning network. The proposed method was conceived for image-based navigation, and aids in the landmark acquisition step, providing real time suggestions to the surgeon with predicted landmark locations. This can be useful in decision making and hopefully help to minimise the errors usually introduced at this stage.

1.3 Research Contributions

The work in this project has resulted in the submission of a method to the BOP Challenge 2019[1]: *Félix&Neves-ICRA2017-IET2019*, that ranked 6th place.

Additionally, a paper was submitted to the AE-CAI | CARE | OR 2.0 joint workshop [2]:

Inês Félix, Carolina Raposo, Michel Antunes, Pedro Rodrigues, João P. Barreto. Towards markerless computer-aided surgery combining deep segmentation and geometric pose estimation: Application in Total Knee Arthroplasty.

1.4 Document Overview

The remainder of this thesis is structured as follows:

Chapter 2 overviews some Deep Learning concepts that are useful for better understanding the proposed algorithms. In addition, it reviews the literature on the topics related to the subject of this thesis;

Chapter 3 provides a detailed description of the proposed markerless video-based navigation system and the experimental results;

Chapter 4 documents the submitted method to the BOP Challenge 2019;

Chapter 5 presents the proof-of-concept method for automatic detection of anatomical landmarks and its experimental validation;

Chapter 6 discusses the results and suggests future work.

[1] `https://bop.felk.cvut.cz/`

[2] `https://workshops.ap-lab.ca/aecai2020/`

2

Background and Related Work

This chapter provides a general introduction to Deep Learning (DL), with particular focus on DL techniques employed in medical applications. It also gives a compact review of previous work related to surgical navigation.

2.1 Deep learning

In recent years, several methods based on DL have emerged in Machine Learning and Artificial Intelligence research. Among the existing DL algorithms, the most popular in the Computer Vision field is Convolutional Neural Networks (CNN), with great breakthroughs in processing images and video [32].

CNN architectures are usually comprised of convolutional, pooling, and fully-connected layers [33]. Convolutional layers allow feature extraction and are composed of a set of filters or kernels, which outputs different feature maps. Each feature map is generated by convolving the input with a filter and then applying a nonlinear activation function, such as a ReLU. Filters used at early stages in the network capture low-level features from the input image, such as color, edge and texture information. As the network deepens, it gradually encodes more abstract features. Pooling layers are often placed in between convolutional layers to reduce the dimensionality of feature maps with the purpose of providing shift-invariance to the output. Nearing the end, one or more fully connected layers convert the feature maps into a feature vector. The last layer is an output layer in which the sigmoid operator is commonly used for binary classification tasks, while the softmax operator is chosen for multiclass classifications. An example of a CNN architecture is represented in Figure 2.1.

A major problem in DL is creating a model that performs well on training data, without overfitting, that is able to generalize to unseen data [34]. To overcome this issue, many strategies were designed for regularization.

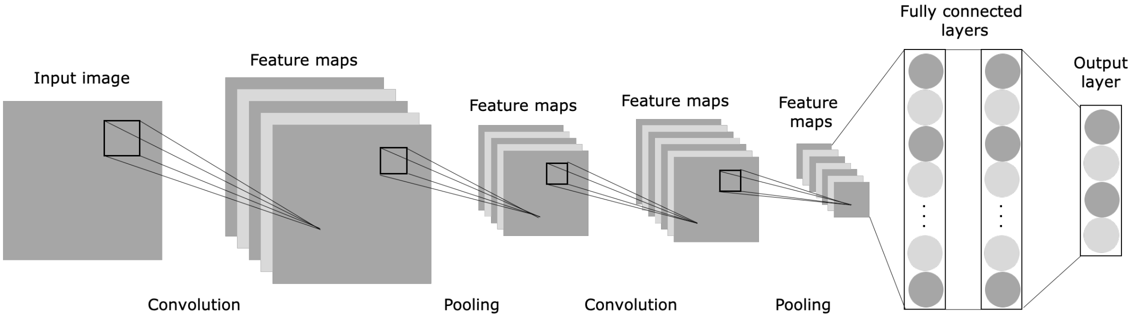


Figure 2.1: Example of a CNN architecture, comprised of convolutional, pooling and fully connected layers.

Dropout This method allows to combine different neural network architectures efficiently, and is implemented by probabilistically dropping out units in the network, along with its connections [35].

Data augmentation A leading cause for overfitting is training DL models with reduced datasets [36]. However, in most cases, the amount of training data is limited. Data augmentation techniques consist on performing transformations to the available data, in order to increase the size of the training set. In image related tasks, recurring methods are geometric transformations (e.g. rotations, shits, scaling) and photometric transformations.

Transfer learning This strategy helps to improve learning of a new task by transferring information from a previous learned task [37]. A common approach is initializing the neural network with weights pre-trained on similar data.

During the training of CNN, the model predicts the output, computes the model error using the selected loss function and then back-propagates to update the parameters (weights and bias) using the gradient descent method. As an optimization process, the goal is to find the parameters that minimizes the loss function. The choice of the loss function, depending on the specific task, impacts the performance of the network [38]. There are several optimization techniques for improving accuracy and training speed on neural network models.

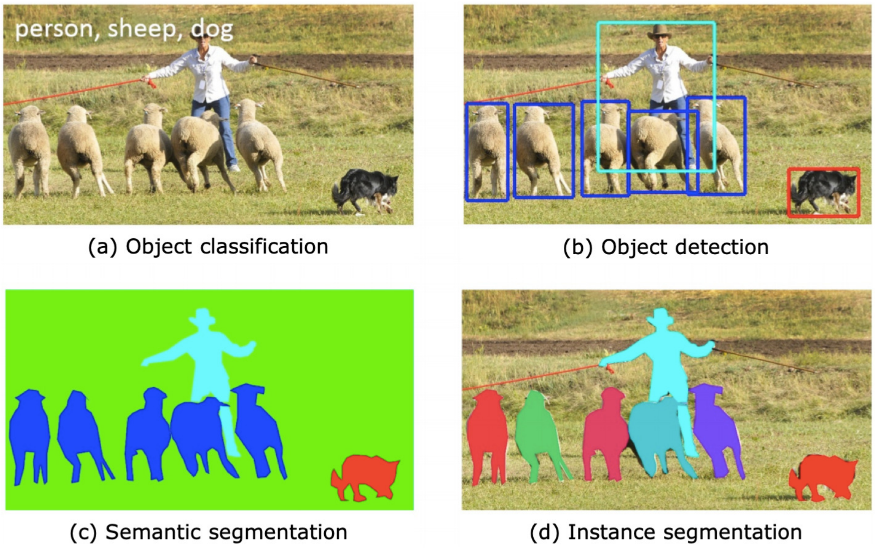
Hyperparameter tuning Hyperparameters can relate to the structure of the model (e.g. number of hidden layers, activation function) or the efficiency and accuracy of the model (e.g. learning rate, batch size, dropout) [39]. Hyperparameter tuning is the process of finding the combination of hyperparameters that allows the network to achieve the best performance.

Mini-batch algorithm Performing the gradient descent method over the entire training set to update only a single parameter is computationally expensive. The mini-batch algorithm accelerates the training process, by computing the gradient over batches of the training data instead [34]. Mini-batch size is taken as an hyperparameter of the model.

Momentum update Momentum helps the gradient descent to faster converge and reduces the oscillations [40]. This is performed by adding a fraction (a hyperparameter) of the update at the past step to the current update.

Adaptive learning rate methods The learning rate has a significant impact on the learning process of the model: setting it too high does not allow the loss function to converge; otherwise, setting it too low, results in slow learning. Hence, optimization methods, such as Adagrad [41], Adadelta [42], Root mean square prop (RMSprop) [43] and Adaptive momentum estimation (Adam) [44], adapt this hyperparameter during training.

Convolutional Neural Networks have been applied to the classification, detection and segmentation of objects and regions in images [32]. The classification task retrieves the objects categories presented in the image (Fig. 2.2a). The object detection task besides recognizing the object in the image, also locates it with bounding boxes (Fig. 2.2b). Segmentation tasks consist in pixel-wise location of each object. While semantic segmentation classifies each pixel into a class in a single mask (Fig. 2.2c), instance segmentation creates a different mask for each instance of an object (Fig. 2.2d).



(a) Object classification

(b) Object detection

(c) Semantic segmentation

(d) Instance segmentation

Figure 2.2: Representation of classification, detection and segmentation tasks. Figure adapted from [45].

2.1.1 DL in medical applications

Recently, the medical field has also adopted DL techniques, in particular Convolutional Neural Networks, for a wide range of applications in image analysis. These techniques are deemed helpful for intra-operative guidance [46], diagnosis and treatment [47].

Image segmentation plays a crucial role in many medical imaging applications, by providing the location of anatomical structures and other areas of interest [48]. For this purpose, the most commonly used architecture is U-Net [1], that comprises a contracting path (encoder) and an expansive path (decoder). The encoder portion repeats the sequence of 2 convolutions followed by a ReLU and a pooling operation for downsampling, while increasing the number of feature maps. The bottleneck portion is the middle part of the network, where usually dropout is added for regularization. Then, the decoder portion repeats the new sequence of a convolution operation to upsample the feature maps, followed by the typical convolution and ReLU combination. The feature maps from the encoder are copied to the decoder through skip connections, to enable precise pixel-wise localization. The final step in the network is a 1x1 convolutional layer to output a mask with the predicted class in each pixel. The U-net network is illustrated in Figure 2.3.

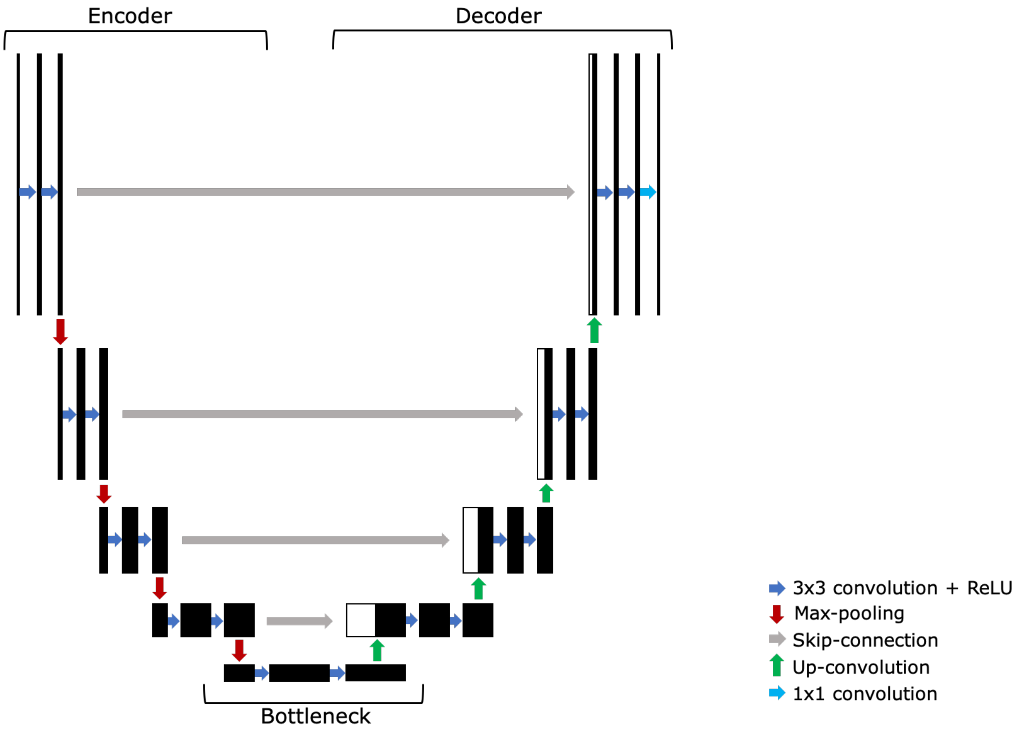


Figure 2.3: Diagram of the U-net network architecture [1]. Each box represents a multi-channel feature map and the arrows correspond to the different operations.

2.2 Image-based surgical navigation

As previously mentioned, image-based navigation require a 3D model of the anatomy, which is acquired pre-operatively through medical imaging (such as CT or MRI). The 3D model relates with tools and instruments in the operating room, using one of the tracking technologies, to support the surgeon in following a pre-specified surgical plan. The typical navigation workflow starts by rigidly attach a reference marker to the patient's anatomy, followed by touching the bone surface with a calibrated tool marker, allowing the 3D model to be registered with the anatomy [22, 49]. From here, the bone can be tracked considering the relative pose of the reference marker to the tracking system at each instance.

The MAKO Stryker Robotic Arm System [50] is an example of an image-based surgical navigation that uses optical tracking. It also qualifies as a semi-active robotic system since it has a robotic arm that assists the surgeon during the surgery.

2.2.1 Markerless video-based surgical navigation system

Markerless navigation from video-based tracking inevitably passes by being able to estimate the 6D pose of femur and/or tibia given an accurate 3D model of the bone obtained from a pre-operative image of the patient. Due to medical requirements, the method to accomplish 6D pose estimation must be accurate, presenting errors below 3 deg and 3-4mm, robust, being able to work in challenging situations where there are significant levels of occlusion, light changes and variability across subjects, and fast. State of the art methods for 6D pose estimation are reviewed next, divided into categories based on the input type.

Depth The common solution is 3D registration that consists in finding the rigid transformation that best aligns the input point cloud with the 3D model [51]. The 3D registration methods are usually comprised of global alignment, followed by local refinement performed with the Iterative Closest Point (ICP) [52] algorithm. The global alignment can be achieved by matching features extracted from the model and estimating the pose using a RANSAC-like framework [53]. However, these approaches fail when the point clouds are too smooth and/or noisy because of the difficulty in finding repeatable saliences that can be matched [51]. Global alignment can also be achieved by defining correspondences on points, such as the family of algorithms 4PCS [54, 55, 56] that uses hypothesize-and-test schemes, by finding sets of 4 points in one point cloud that are congruent to 4 points selected in the other, or the point pair feature approach, first introduced by Drost *et al.*

[57] and later improved by Vidal *et al.* [58], that creates a global model descriptor based on oriented point pair features and matches that model locally using a voting scheme. Registration methods are not suitable for the recognition of objects in complex and cluttered scenes, usually requiring a previous step for object detection/ segmentation. More recently, methods based on DL, such as VoxelNet [59], present deep networks for object pose estimation from 3D point cloud inputs.

RGB Classical methods determine the correspondences between the 3D model and the input image, and estimate the pose using the Perspective-n-Point (PnP) algorithm [60, 61]. Modern approaches combine CNN architectures to predict 2D keypoints with the PnP algorithm to retrieve the object pose [62, 63, 64]. Other methods propose end-to-end deep networks: SSD-6D [65] builds 6D pose hypotheses from viewpoints and in-plane rotations predictions; PoseCNN [66] trains the network to perform semantic labeling, 3D translation estimation and 3D rotation regression; Sundermeyer *et al.* [67] learns implicit orientation with Augmented Autoencoders. However, [65, 66, 67] report improved performance when using additional depth data to refine the poses with the ICP algorithm, moving them into the latter category.

RGB-D Methods can be divided into template-based and learning-based. Template-based approaches estimate the object pose by matching a defined template to the input image. Hinterstoisser *et al.* [68] extracts color gradients from the color images and computes 3D surface normals from the depth map. Learning-based approaches extract discriminative features from the data and use classification algorithms to predict pose hypotheses. Brachmann *et al.* [69] employs a random forest to obtain pixelwise classification and optimizes the output with a RANSAC-based scheme. Tejani *et al.* proposed Latent-Class Hough Forest, learning only from positive samples at the training stage. Recent methods are based on CNN architectures, such as PointFusion [70] and DenseFusion [71] that combine data coming from RGB and depth channels in end-to-end deep networks.

Conclusions Classical PnP approaches for 6D pose estimation from RGB cannot handle the problem because of the difficulty in locating keypoints in medical images, where bone surface is poorly textured, in an accurate and robust manner. Fortunately, the introduction of deep learning-based methods and generic 6D pose estimation from RGB images opened new possibilities by using deep networks to replace naive, ad-hoc schemes that establish explicit image-model correspondences. Nevertheless, and despite the many progresses in 6D pose from RGB, the outcome of the BOP challenge [3, 27] clearly shows that results are substantially inferior to the ones that can be accomplished with a depth camera. In addition, [71] shows that combining the color and depth information from RGB-D inputs boosts the performance of end-to-end deep learning methods. For this reason, and given the accuracy and robustness requirements in one hand, and the availability of commercial, off-the-shelf RGB-D sensors on the other, it is wise to include depth cues in the task of 6D pose estimation, instead of relying exclusively in RGB.

The aforementioned deep learning RGB/RGB-D methods for pose estimation require training the 3D model of an object instance. However, for video-based markerless navigation, the algorithm needs to handle high structure variation of bone surfaces, given that in each surgery a new 3D pre-operative model of the bones is considered. There are very recent research efforts in category level tracking [72], but the experimental results do not exhibit robust performance yet.

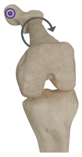
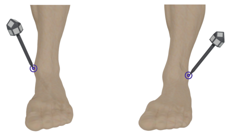
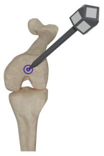
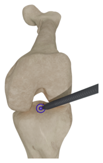
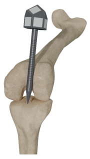
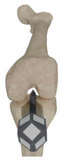
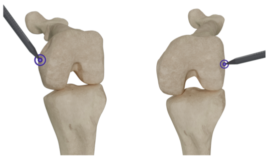
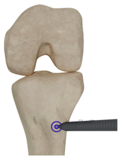
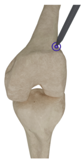
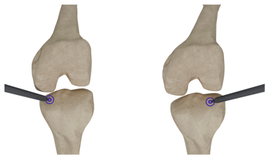
6D pose estimation is already being used in TKA navigation systems [49, 2]. Previous work [49] has employed 3D registration in the context of CAS for aligning a pre-operative model with the patient's anatomy. However, in [49] the anatomy is reconstructed by touching the bone with an instrumented tool and the registration process aligns 3D curves with a dense surface, still relying on the use of fiducial markers.

The first attempt to accomplish markerless surgical navigation in TKA that we are aware of is [2]. The presented prototype is an RGB-D based system that uses deep segmentation to leverage geometric pose estimation: first, it uses a deep learning technique to perform bone segmentation from RGB images; then, the depth information corresponding to the targeted anatomy is extracted from the depth map considering the segmented portion of the image; finally, the reconstructed point cloud is used to register the 3D pre-operative model. As stated before, this prototype has some limitations, thus in Chapter 3 we provide further improvements for this solution.

2.3 Image-free surgical navigation

Image-free navigation systems using optical tracking are the most widely used computer assisted solutions in TKA [4]. In this type of navigation, the acquisition of landmarks is a required step to create reference frames that relate the base markers attached to the bones to the patient's anatomy. Thus, intra-operatively, the surgeon has to indicate the location of the landmarks presented in Table 2.1. Unlike the hip center, that is estimated through a kinematic method, all the other landmarks are localized using the tool marker. Additional points in the bone surface are also collected to allow the 3D reconstruction of the model [73]. Based on the acquired data, the system proposes a surgical plan for the implant positioning [74].

Table 2.1: Required landmarks for navigation in image-free TKA. Copyright by P3D.

Femur landmarks		**Tibia landmarks**	
Hip center		Ankle center	
Knee center		Knee center	
Whiteside's line		Anterior-posterior axis	
Epicondylar line		Medial third tubercle	
Anterior cortex		Plateau points	

State-of-the-art image-free surgical navigation systems for performing the TKA surgery are mostly optical tracking systems. As an example, there is the Smith&Nephew's NAVIO Surgical System [75], which is robotics-assisted by using a robotic handheld instrument that aids the surgeon in executing bone resections. More recently, Intellijoint launched Intellijoint KNEE^{TM} [76], that innovated with a mini-optical tracking device that is portable.

As previously mentioned, all image-free navigation systems require the explicit pinpointing of several bony landmarks, yet, this task is time-consuming and error-prone. A solution for this problem is given in Chapter 5, where we present a proof-of-concept method to assist in this task.

3

Markerless video-based surgical navigation

This chapter provides a detailed description of an improved system based on the prototype from [2]. The diagram in Figure 3.1 illustrates the markerless video-based surgical navigation pipeline. The pipeline consists of four main stages: (1) A depth camera is used to capture RGB and depth data during the surgery; (2) Bone segmentation is performed in the RGB images; (3) The 3D surface of the bone is reconstructed by applying the segmentation to the point cloud from the depth sensor; (4) The reconstructed point cloud is aligned with the 3D pre-operative model to retrieve the relative pose.

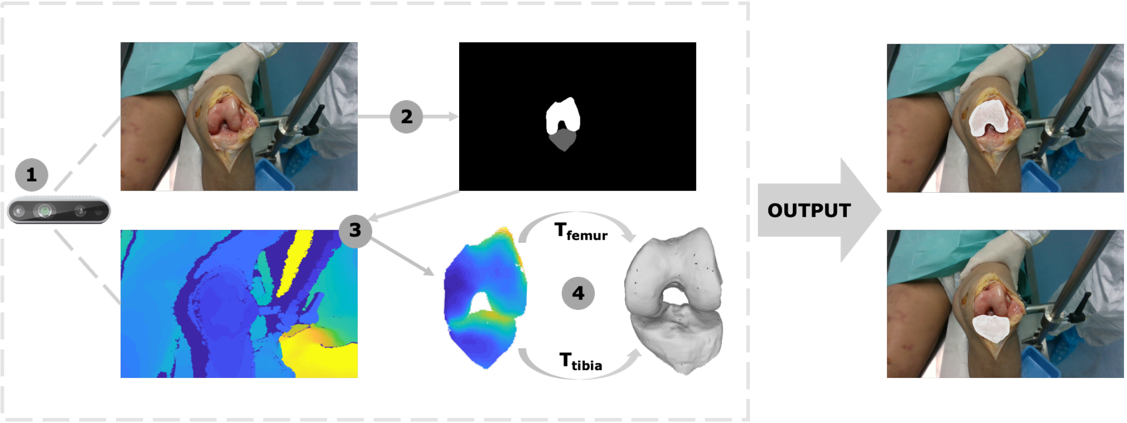


Figure 3.1: Markerless video-based surgical navigation pipeline, with the main steps represented: (1) data acquisition, (2) bone segmentation, (3) extraction of bone point cloud, and (4) bone registration.

The following sections 3.1, 3.2, 3.3 and 3.4 describe all the methods used in each step. In the final section, 3.5, the experiments conducted to validate the proposed system are presented, along with the obtained results.

3.1 Data acquisition

3.1.1 System for data acquisition

For training the segmentation DL architecture, only RGB data is necessary, thus any standard RGB camera can be used. The acquisition is in video, and frames are then extracted, at 10 frames per second, using the open source library FFmpeg [77].

For evaluating the proposed TKA navigation solution, RGB and depth data need to be captured. The platform setup in [2] was composed of a video camera and the Occipital Structure Core Depth Sensor, with these components fixed together and calibrated. Considering that the depth data from the sensor was very noisy, which affected the registration results, research was done to find a new, more effective, depth sensor.

The chosen equipment was the Intel RealSense Depth Camera D435i, a compact camera that offers high RGB and depth resolution. Comparison between the specifications of both cameras is presented in Table 3.1, as well as depth maps from both sensors in Figure 3.2.

Table 3.1: Specifications from the Occipital Structure Core Depth Sensor and the Intel RealSense Depth Camera D435i [78, 79].

Specifications	**Occipital Structure Core**	**Intel RealSense D435i**
Depth resolution	Up to 1280 x 960	Up to 1280 x 720
Depth frame rate	Up to 54 fps	Up to 90 fps
Depth Min Z distance	30 cm	10,5 cm
RGB resolution	640 x 480	1920 x 1080
RGB frame rate	Up to 100 fps	30 fps

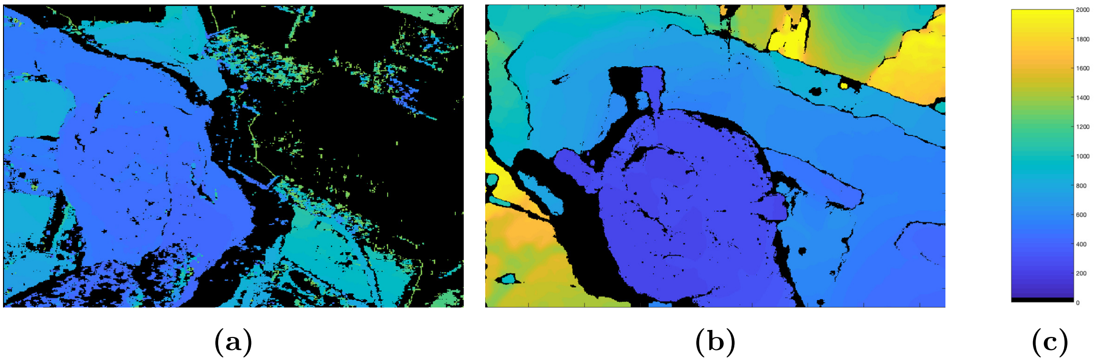


Figure 3.2: Depth maps from **(a)** Occipital Structure Core Depth Sensor and **(b)** Intel RealSense Depth Camera D435i. **(c)** Color bar represents the Z-values in millimetres.

The Intel RealSense SDK [80] was used to record and save the data through the RealSense Viewer. The settings for data acquisition were RGB resolution of 1920 x 1080 px and depth resolution of 848 x 480 px, at 30 fps.

The acquisition of data in each experiment followed a protocol to capture video sequences under different lighting conditions, from various perspectives and with occlusions from hands and surgical tools, to represent the intra-operative scenarios in a realistic manner. Figure 3.3 shows examples of frames in different scenarios.

3.1.2 Dataset description

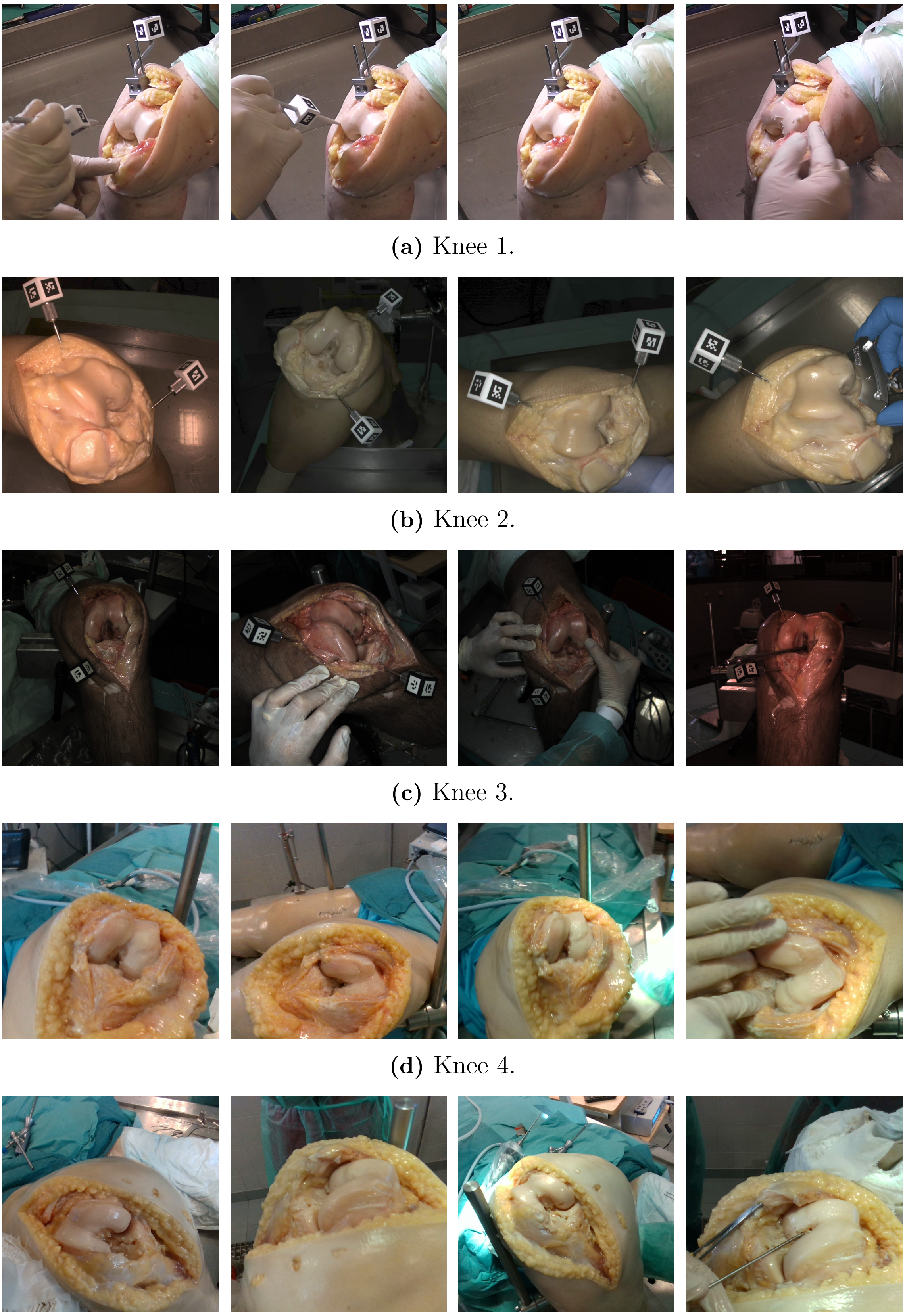
The full dataset is a combination of video sequences captured from 8 different TKA ex-vivo surgeries. During the execution of this project, I was given the opportunity to attend some of these cadaver trials to collect data following the process described in Section 3.1 (Knees from 4 to 8). The remaining data was taken from Perceive3D's database.

Table 3.2 presents the data used in this chapter. Some knees belong to the same individual, not presenting much variability other than side difference. Differences in knee anatomy and body structure across individuals contribute to inter-subject variability. The images from each experiment also differ in the background and lighting conditions. Examples of frames from each knee are given in Figure 3.3.

In some images, the femur and tibia have markers attached to the bones, that are removed (as in Figure 3.4) before being introduced to the learning algorithm in order to avoid any relationships between the location of the markers and the bones. This is done automatically by detecting the markers through their pose, removing the pixels corresponding to the area around them and inpainting that area.

Table 3.2: Information about each knee, such as knee side, individual's gender and data type.

Subject	Dataset	Side	Gender	Data type
1	Knee 1	Left	Female	RGB
2	Knee 2	Right	Female	RGB
3	Knee 3	Left	Male	RGB-D (Structure Core Sensor)
4	Knee 4 Knee 5	Left Right	Female	RGB-D (RealSense Depth Camera)
5	Knee 6 Knee 7	Right Left	Female	RGB-D (RealSense Depth Camera)
6	Knee 8	Right	Male	RGB-D (RealSense Depth Camera)



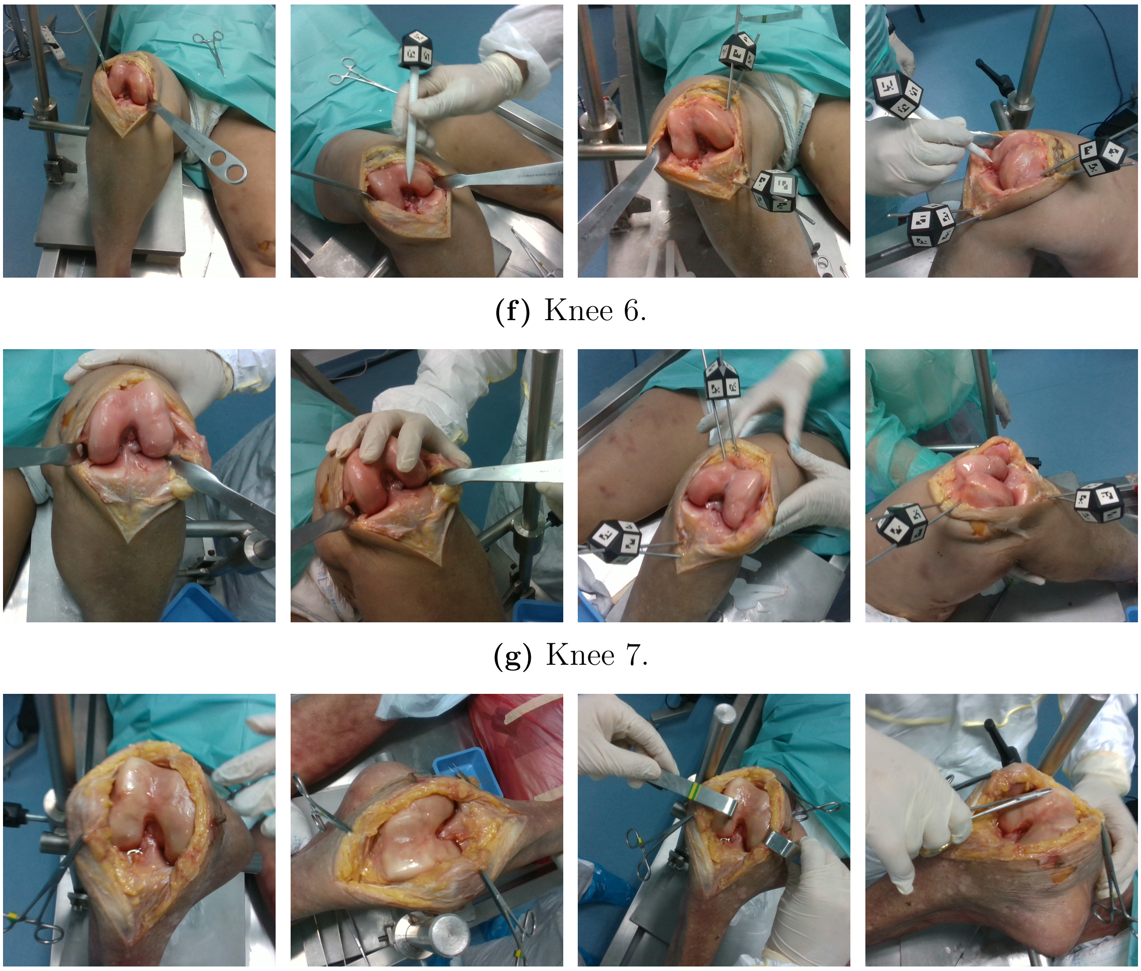
(a) Knee 1.

(b) Knee 2.

(c) Knee 3.

(d) Knee 4.

(e) Knee 5.

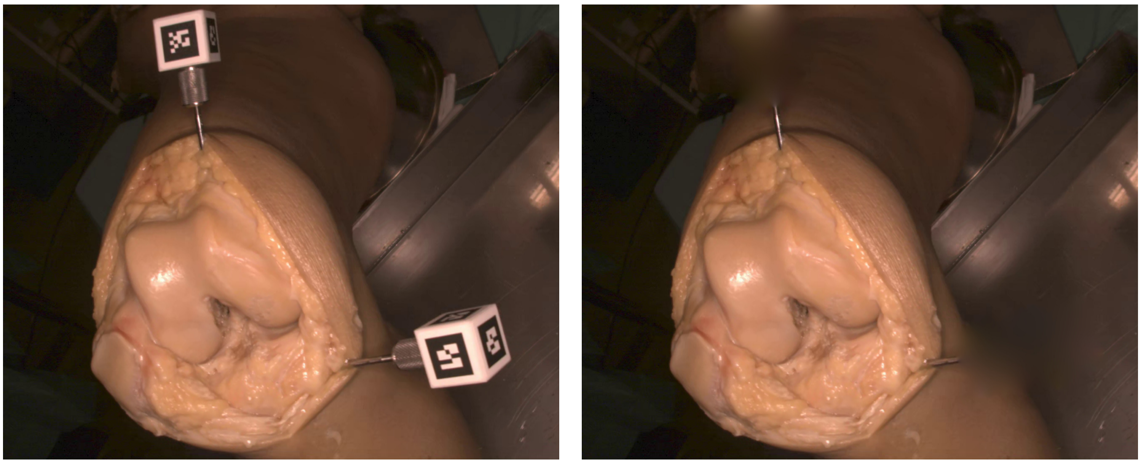


(f) Knee 6.

(g) Knee 7.

(h) Knee 8.

Figure 3.3: Examples of images from each knee.



(a) **(b)**

Figure 3.4: Image with **(a)** markers and **(b)** markers inpainted.

3.2 Bone segmentation

3.2.1 Deep Learning architecture

As in [2], the chosen network to perform bone segmentation was the TernausNet [81], an adaptation of the U-net network architecture (presented earlier in Section 2.1.1). In this architecture, the encoder is initialized with VGG11 neural network, pre-trained on ImageNet [82].

To the original implementation provided by [81], Rodrigues *et al.* [2] suggested adding dropout to prevent overfitting and freezing the encoder weights to optimize only the decoder portion. Both suggestions are taken in our implementation as hyperparameters. Hence, the hyperparameter search space for optimization contains the number of epochs, learning rate, dropout ratio, the possibility to freeze the encoder weights and the number of filters in the decoder portion of the network. The number of filters influences the number of feature maps in the convolutional layers of the decoder. The mini-batch size was set to 10 and the Adam algorithm was used as an optimizer.

In this dataset, only a small percentage of the pixels in the images correspond to the target anatomies, identifying a very common problem in the medical imaging field that is class imbalance [83]. The TernausNet architecture applies the cross-entropy loss function in the optimization process of the network [81]. Although this loss is the most used for classification purposes, it is not the best choice when handling segmentation tasks in datasets with unbalanced classes, since it averages the pixel-wise error. Inspired by [84], we implemented a loss function ($loss$) that incorporates the cross-entropy and dice losses, L_{CE} and L_{DICE} respectively (Equation 3.1, complete definition of both losses in [84]), to help in achieving the optimal pixel-wise accuracy and segmentation metrics. The dice loss measures the overlap between the predicted output and the label.

$$loss = L_{CE} + (1 - L_{DICE}) \tag{3.1}$$

This time, in the final stage, a convolutional layer with softmax activation was used to create a pixel-wise mask of three classes (background, femur and tibia), to predict simultaneously at each instant both anatomies, instead of the sigmoid activation used in [2] to predict only two classes (background and femur).

The network implementation was done in Pytorch [85], a Python library for Deep

Learning, that allows to use GPU for fast computation. Thus, it was used a computer with 4 GPUs, which accelerated the training process.

The final network architecture is illustrated in Figure 3.5.

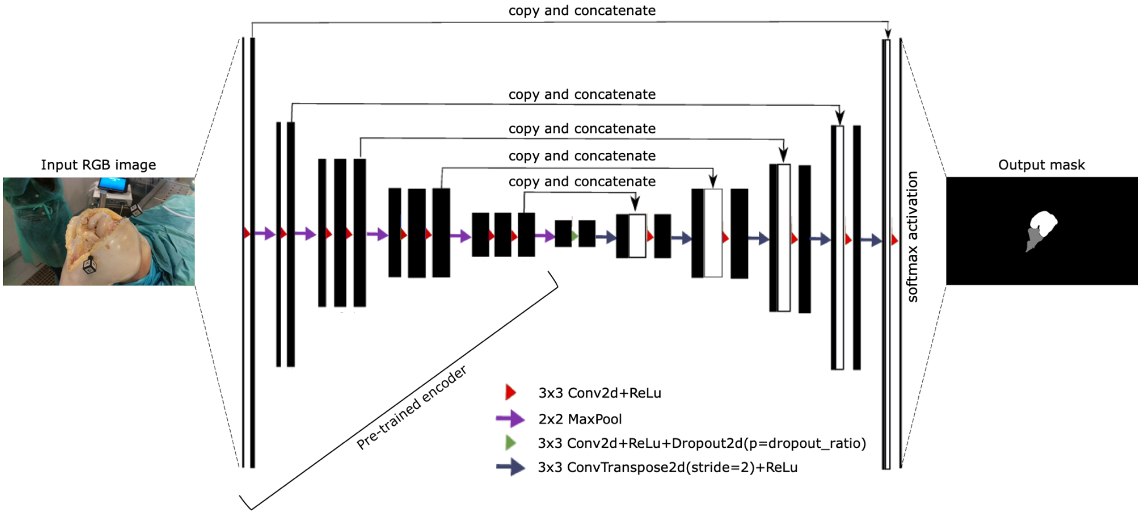


Figure 3.5: Deep Learning network architecture used for bone segmentation, based on TernausNet [81]. The output mask predicts simultaneously the femur and tibia.

3.2.2 Generating label images

The network in Figure 3.5 requires to be trained with thousands of images in which each pixel has a label corresponding to one of the classes (background, femur or tibia). Manually labeling every image would be extremely tedious and time-consuming, and infeasible in an acceptable amount of time.

The approach presented by [2] to perform the labelling task of a similar dataset was accomplished with the manual segmentation of some images and propagating the segmentation to the neighboring frames using the initial pose registration, the detected marker pose at that frame and the 3D bone model. The method proved to be effective in relieving the extensive manual segmentation, however, it still needed constant supervision, since it was not sensitive to occlusions and shifts in the viewpoint.

To overcome this issue, we developed a scheme for the automatic labeling of images that only requires the manual labelling of 100 images per bone instance. For each bone instance, the images to be manually labeled are temporally sampled to make a good discretization of the variability of poses and occlusions. Then, the manually labelled data is modeled using the network presented in Section 3.2.1 and, in this specific case, we aim that the model overfits the data so it can predict the remaining labels for the same bone. Without the need of any tracking instrumentation using this method, it's much easier to gather training data intra-operatively and grow the dataset to improve generalization capabilities.

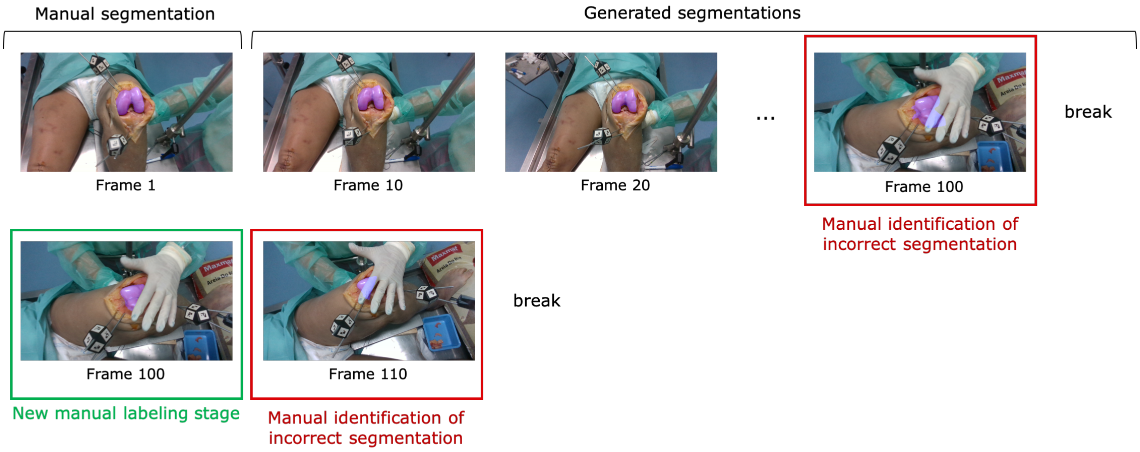

Manual segmentation
Generated segmentations
Frame 1
Frame 10
Frame 20
...
Frame 100
break
Manual identification of incorrect segmentation
Frame 100
New manual labeling stage
Frame 110
Manual identification of incorrect segmentation
break

Figure 3.6: Representation of the labelling approach of [2]. The method needs constant supervision to define when new manual segmentations need to be performed.

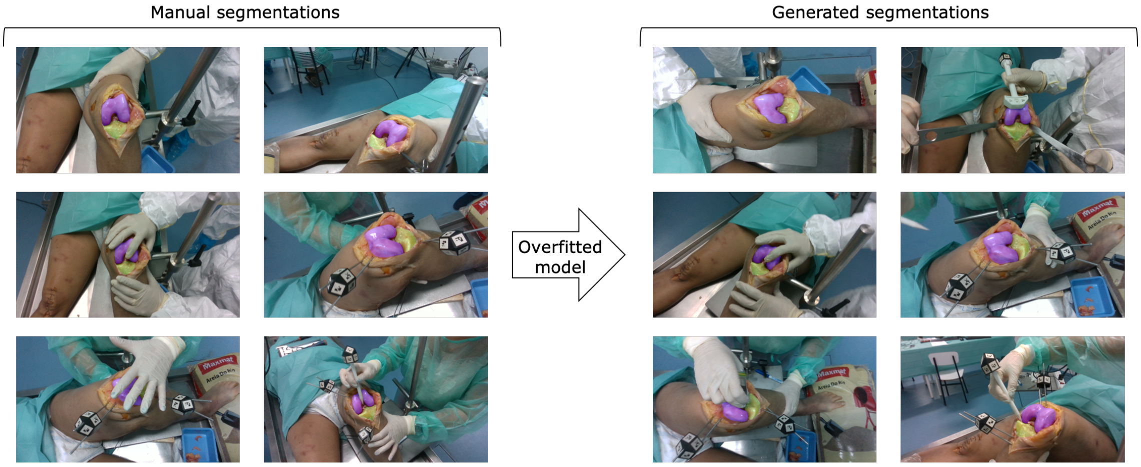


Figure 3.7: Representation of our labelling solution. A small sample of the dataset is manually labeled at once and introduced to the network. The overfitted model allows to generate accurate segmentations for the rest of the dataset, without requiring constant supervision and tracking instrumentation.

3.2.3 Data augmentation techniques

To provide robustness to the model, training data was augmented by applying random vertical/horizontal flips (Fig. 3.8b), shift, scale and rotation transformations (Fig. 3.8c) and by randomly adjusting brightness and contrast (Fig. 3.8d), resorting to Albumentations [86], an open source library for fast implementation of data augmentation operations.

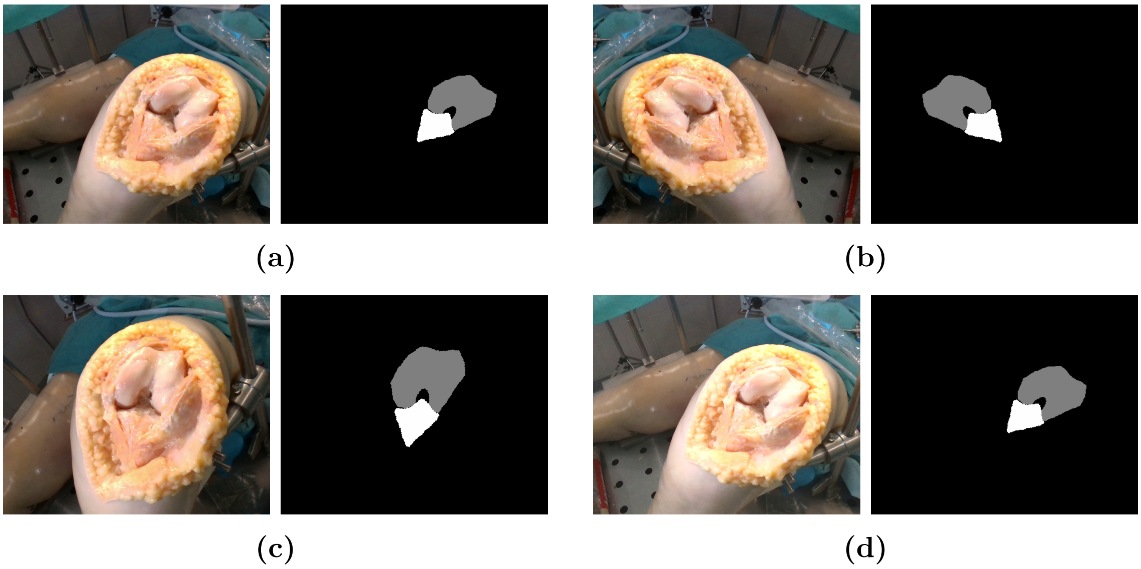


Figure 3.8: Data augmentation techniques in **(a)** original image and mask: **(b)** Flip, **(c)** ShiftScaleRotate and **(c)** RandomBrightnessContrast.

3.2.4 Evaluation metrics

To evaluate the segmentation performance, two metrics often used in image segmentation tasks were adopted: Intersection over Union (IoU) and Dice coefficient. These measure the overlap between the label and the predicted mask, for each class individually. Formulation of the metrics are denoted as shown:

$$IoU = \frac{TP}{TP + FP + FN} \tag{3.2}$$

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3.3}$$

with TP being the number of pixels correctly classified as femur or tibia, FP the number of pixels incorrectly classified as femur or tibia and FN the number of pixels incorrectly classified as background.

3.3 Extraction of bone point cloud

Color and depth data are captured from different components in the depth camera (RGB color sensor and depth module, respectively), translating in data associated to different coordinate systems. A set of transformations need to be performed in order to align the depth stream with the color stream.

The calibration of each component is represented in Equation 3.4:

$$K_d = \begin{bmatrix} f_x^d & 0 & c_x^d \\ 0 & f_y^d & c_y^d \\ 0 & 0 & 1 \end{bmatrix} \qquad K_{rgb} = \begin{bmatrix} f_x^{rgb} & 0 & c_x^{rgb} \\ 0 & f_y^{rgb} & c_y^{rgb} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where f_x and f_y describe the focal length of the image; c_x and c_y are the pixel coordinates of the principal point (center of projection). These parameters are provided by the manufacturer.

Back-projection From the pixel coordinates of the depth image, $p_d(x,y)$, considering the depth camera intrinsics, K_d, the 3D point in the depth coordinate space is $P_d(X,Y,Z)$, as follows:

$$P_d = d \begin{bmatrix} \frac{x - c_x^d}{f_x^d} \\ \frac{y - c_y^d}{f_y^d} \\ 1 \end{bmatrix} \tag{3.5}$$

with $d = p_d(x,y)$, being the depth value obtained with the depth sensor.

Transformation The 3D point in the depth coordinate space, $P_d(X,Y,Z)$, is transformed to the RGB coordinate space, $P_{rgb}(X,Y,Z)$, using the extrinsic parameters, also given by the manufacturer, defined as a rotation matrix, R, and a translation vector, t:

$$P_{rgb} = RP_d + t \tag{3.6}$$

Projection The final step is recovering the pixel coordinates of the RGB image, $p_{rgb}(x,y)$, from the projection of the 3D point in the RGB coordinate space, $P_{rgb}(X,Y,Z)$, this time considering the RGB camera intrinsics, K_{rgb}:

$$p_{rgb} = \begin{bmatrix} \frac{f_x^{rgb} \cdot X}{Z} + c_x^{rgb} \\ \frac{f_y^{rgb} \cdot Y}{Z} + c_y^{rgb} \end{bmatrix} \tag{3.7}$$

Once the color and depth frames are aligned (Figure 3.9), the obtained segmentation mask from the previous step (Section 3.2) is applied to the depth map in order to extract only the points corresponding to the bones.

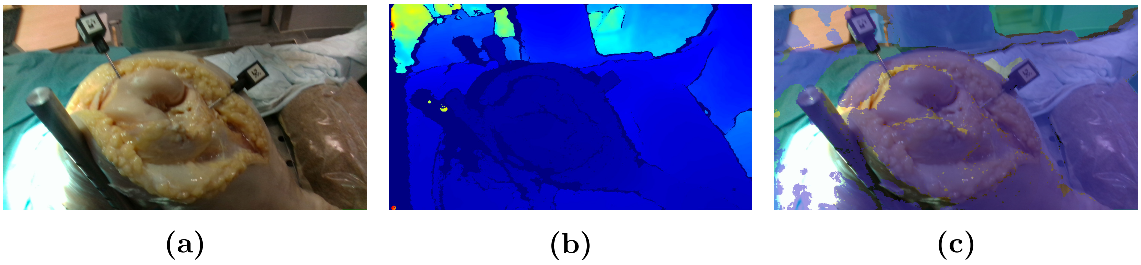


Figure 3.9: **(a)** Color frame with resolution 1920x1080 px. **(b)** Depth frame with resolution 848x480 px. **(c)** Aligned color and depth frames. Representation not to scale.

3.4 Bone registration

3.4.1 Registration algorithm

Similarly to [2], we decided to employ the method proposed in [51] to solve the registration task, which consists in aligning the 3D surface of the bone reconstructed by making use of the depth sensor with the pre-operative model. This method has proved to be faster and more accurate than the family of algorithms 4PCS (presented in 2.2.1) [51]. It is also resilient to very high levels of outliers, which is desirable in our case because small errors in the segmentation step (Section 3.2) can cause a significant amount of outliers. Such characteristics of speed and robustness to outliers are achieved by extracting pairs of points and their normals in one point cloud, finding congruent pairs of points in the other to establish alignment hypotheses and testing them in a RANSAC scheme.

Experiments on the registration of the femur using this method had already been performed in [2], presenting good accuracy. However, and to the best of our knowledge, attempts on registering the tibia have never been performed. Comparing to the femur, performing markerless alignment of the tibia with a pre-operative model is much more challenging due to the significantly smaller area of exposed bone. The registration parameters were tuned for accommodating the noise in the data and the small area of exposed bone.

3.4.2 Evaluation metrics

In the test set, the femur and tibia have markers attached to the bones, that are tracked for evaluation of pose estimation. The quantitative analysis of the 6D pose estimation is performed in images where the markers are visible and can be tracked using [22]. Using the marker's pose in each image, the reconstructed point cloud is represented in the marker's reference frame and registration is afterwards performed. Registrations are then considered successful if the algorithm converges to a stable solution, and if they have at least 80% inliers, which means that at least 80% of the points of the reconstructed surface are at a maximum distance of 2 mm from the model.

In theory, since the marker is rigid with respect to the bone, the registration result should provide the same transformation for all images. Thus, in order to assess the registration performance, the ground truth, T_{gt}, is deemed as the median transformation computed from the set of all considered registrations and compared with each estimation:

$$dT_i = T_i^{-1} T_{gt} \tag{3.8}$$

Consider that dT_i is composed of a rotation matrix, R_i, and a translation vector, t_i. The relative rotation errors, e_i^{rot}, are taken as the angular magnitude computed from the the Rodrigues' rotation formula (Equation 3.9), and the relative translation errors, e_i^t, are taken as the norm of the translation component (Equation 3.10):

$$e_i^{rot} = cos^{-1}(\frac{1}{2}(tr(R_i) - 1)) \tag{3.9}$$

$$e_i^t = \|t_i\| \tag{3.10}$$

The final rotation and translation errors, e^{rot} and e^t, are reported as the median of the relative errors, in degrees and millimeters, respectively.

3.5 Experiments and results

This section reports results of the segmentation and registration stages of the proposed pipeline in ex-vivo data (in Sections 3.5.2.2 and 3.5.2.3), and provides a comparison with the baseline method [2] (Section 3.5.2.4). It begins with an experiment for analysing the accuracy of the depth sensor is performed (Section 3.5.1).

3.5.1 Maximum registration accuracy

As previously mentioned, in a TKA procedure, it is medically required that the alignment between the patient's anatomy and the pre-operative model presents an error below 3 deg and 4 mm. Since it is known that the accuracy of consumer depth sensors is typically in the range of 2 to 5 mm [87], which is considerable when compared to the medical accuracy requirements, we decided to measure the maximum possible accuracy achieved when performing registration with our depth sensor. For this, we used the experimental setup illustrated in Figure 3.10 to acquire a dataset of around 200 RGB-D images of a dry femur model in a controlled environment without occlusions, light changes, specularities or any other sources of error. The pixels corresponding to the femur were manually segmented in the RGB images for generating 3D point clouds that were then registered with the virtual model. Under these controlled circumstances, the median registration error was 1.13 deg and 1.78 mm, which can be considered as the amount of error induced solely by the depth sensor.

Prior to acquiring the RGB-D images, it was experimentally observed that the white surface of the dry knee model caused specularities that significantly increased the error in the measured depth. This source of error was eliminated by painting the model with a matt color, as can be seen in Figure 3.10. However, in the ex-vivo sequences, the reflection of light on bone surface occurs (refer to Figure 3.11) and cannot be eliminated. This is, besides the depth sensor's low accuracy, another relevant source of error, which is, unlike the first, difficult to quantify.

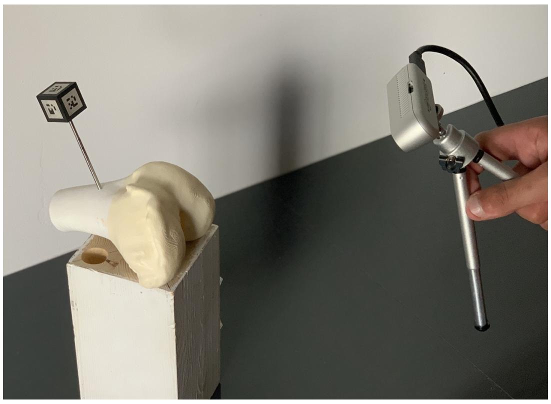


Figure 3.10: RGB-D camera (right) used for the experiments presented in this thesis, and the 3D printed femur model with a marker attached (left) used for analyzing the depth measurement errors of the sensor.

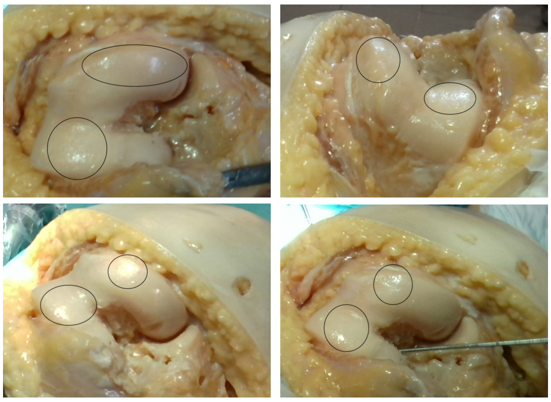


Figure 3.11: Examples of images with reflection of light on bone surface (identified with circles), which affects the depth quality.

3.5.2 Ex-vivo experiments

3.5.2.1 Training the network

The DL network was trained considering the data from 6 ex-vivo experiments (Table 3.3). The experiments to tune the hyperparameters were conducted using leaveone-patient-out cross-validation, resulting in 5-fold cross-validation (remember from Table 3.2 that Knee 4 and 5 are from the same subject). Table 3.4 presents the division of the train and validation sets into each fold. This validation approach allows to evaluate the behavior of each model when tested to unseen data from different individuals.

Table 3.3: Dataset distribution for bone segmentation.

Dataset		**Number of images**
Train/ Validation	Knee 1	600
	Knee 2	5023
	Knee 3	9486
	Knee 4	6588
	Knee 5	7918
	Knee 8	10518
Test	Knee 6	8376
	Knee 7	8273

Table 3.4: 5-fold cross-validation division of the train and validation sets.

Fold	**Train set**	**Validation set**
1	Knee 1, 2, 3, 4, 5	Knee 8
2	Knee 2, 3, 4, 5, 8	Knee 1
3	Knee 1, 3, 4, 5, 8	Knee 2
4	Knee 1, 2, 4, 5, 8	Knee 3
5	Knee 1, 2, 3, 8	Knee 4, 5

Several models were trained with different combinations of hyperparameters, randomly chosen from a defined range of values:

- **Number of epochs**: 3, 4, 5, 6, 7, 8, 9, 10
- **Learning rate**: [0.00001, 0.001]
- **Dropout ratio**: [0, 0.9]
- **Freeze encoder weights**: 0, 1
- **Number of filters in the decoder**: 16, 32

The hyperparameters selected to train the final network were the ones that yielded the best results in the cross-validation process (Table 3.5). Thus, the final Deep Learning model was trained only in the decoder weights of the network with the number of filters defined to 16, a learning rate of 2e-4 and a dropout probability of 52% for 6 epochs.

Table 3.5: U-net models trained for bone segmentation with different hyperparameter combinations. Mean Dice and Mean IoU are the average metrics of all classes in the images from the validation sets of all folds. The best results (model 003) are highlighted in bold.

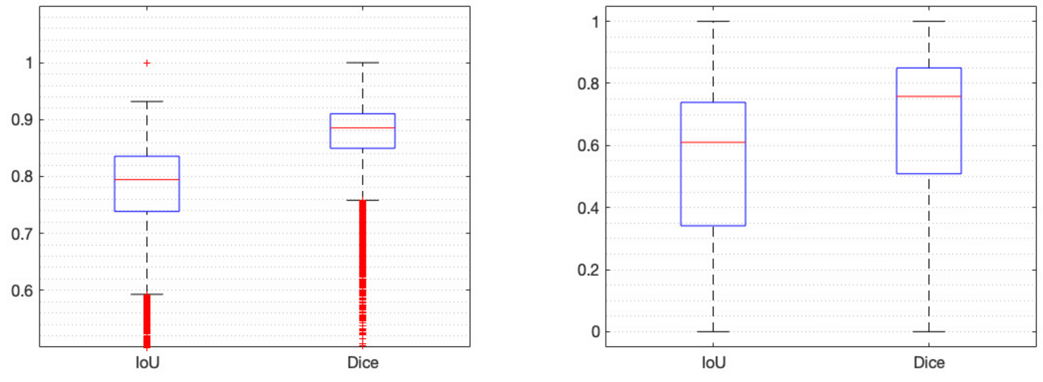
Model	Number of epochs	Learning rate	Dropout ratio	Freeze encoder	Number of filters	Mean IoU	Mean Dice
000	10	0,000218	0,309	1	32	0,512	0,629
001	9	0,000028	0,263	1	16	0,505	0,624
002	4	0,000015	0,563	1	32	0,490	0,613
003	**6**	**0,000213**	**0,516**	**1**	**16**	**0,532**	**0,648**
004	10	0,000012	0,018	1	16	0,505	0,622
005	10	0,000044	0,752	0	16	0,475	0,575
006	7	0,000087	0,083	0	16	0,409	0,493
007	4	0,000052	0,876	0	16	0,460	0,558
008	4	0,000030	0,541	0	16	0,525	0,632
009	4	0,000014	0,862	0	32	0,484	0,593
010	7	0,000131	0,609	0	32	0,431	0,528

3.5.2.2 Quantitative results

The final model was tested individually in two test datasets: Knee 6 and Knee 7. The quantitative results for each dataset are presented below.

Knee 6

The Knee 6 dataset is composed of 8376 images. Figure 3.12 shows the distribution of the segmentation metrics in this dataset. The Deep Learning model achieved a median IoU of 0.794 and a median Dice coefficient of 0.885 for femur segmentation, with the outliers displayed in the distribution corresponding to 8% of the total number of images. As for the segmentation of the tibia, the metrics obtained were IoU of median 0.610 and Dice coefficient of median 0.758.



(a)

(b)

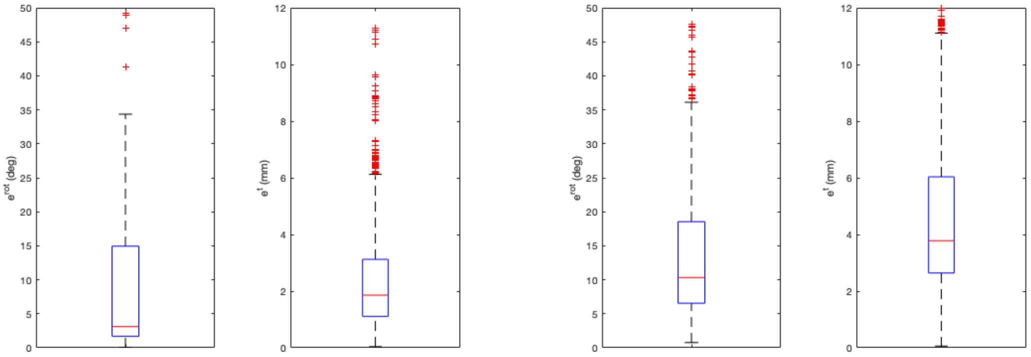
Figure 3.12: Metric distribution for the **(a)** femur and **(b)** tibia segmentation in the Knee 6 dataset.

Following the constraints explained in Section 3.4.2, Table 3.6 presents the percentage of the dataset that was used for registration (images with anatomy's marker detected) and the percentage of the dataset with successful registrations.

Table 3.6: Percentage to the total number of images of the Knee 6 dataset considered in the registration step.

Bone	**Point clouds for registration**	**Successful registrations**
Femur	84,5%	36,4%
Tibia	80,5%	39,3%

Distribution of the errors calculated from the registrations is showed in Figure 3.13. Considering the distribution of successful registrations, for the femur registration, the obtained median rotation error was 3.13 deg and the median translation error was 1.85 mm, and for the tibia registration, the results were 10.34 deg of median rotation error and 3.78 mm of median translation error.



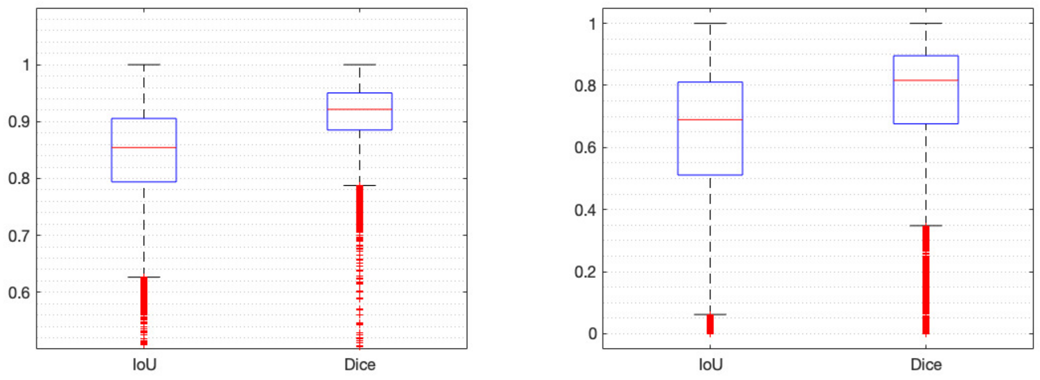
(a) **(b)**

Figure 3.13: Distribution of registration errors for the **(a)** femur and **(b)** tibia in the Knee 6 dataset.

Knee 7

The Knee 7 dataset is comprised of 8273 images. Figure 3.14 shows the distribution of the segmentation metrics in this dataset. The femur segmentation results were IoU of median 0.854 and Dice coefficient of median 0.921, where around 4% of the segmentations are outliers in the distribution. Regarding the tibia segmentation, the model obtained a median IoU of 0.689 and a median Dice coefficient of 0.816, with the outlier percentage being 8% of the images in the dataset.

Table 3.7 specifies the percentage of the dataset that is used for registration and the percentage of the dataset that has successful registrations.



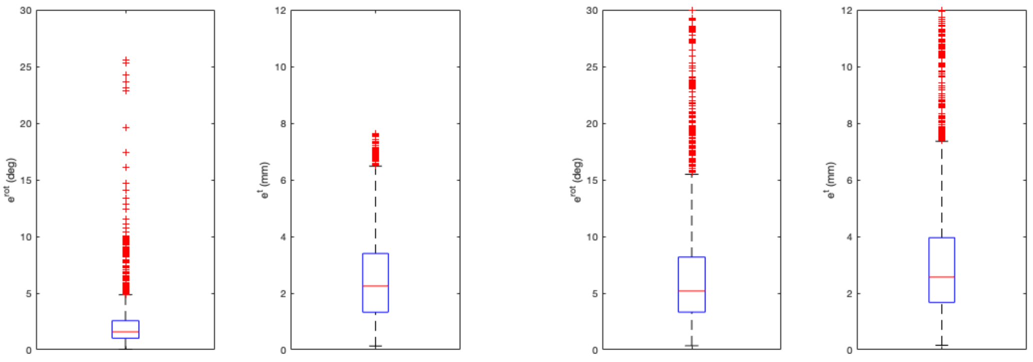
(a) (b)

Figure 3.14: Metric distribution for the **(a)** femur and **(b)** tibia segmentation in the Knee 7 dataset.

Table 3.7: Percentage to the total number of images of the Knee 7 dataset considered in the registration step.

Bone	Point clouds for registration	Successful registrations
Femur	80,3%	33,9%
Tibia	81,9%	67,9%

Distribution of the errors assessed from the registrations is showed in Figure 3.15. Regarding the distribution of successful registrations, for the femur registration, the results were 1.58 deg of median rotation error and 2.25 mm of median translation error. As for the tibia registration, the achieved median rotation error was 5.18 deg and the median translation error was 2.57 mm.



(a) (b)

Figure 3.15: Distribution of registration errors for the **(a)** femur and **(b)** tibia in the Knee 7 dataset.

Results summary

Table 3.8 summarizes the segmentation and registration results for each test dataset.

Table 3.8: Segmentation and registration results in the test datasets.

Dataset	Bone	Number of images	Median IoU	Median Dice	Successful registrations	Median e^{rot} (deg)	Median e^{t} (mm)
Knee 6	Femur	8376	0.794	0.885	3052	3,13	1,86
	Tibia		0.610	0.758	3295	10,34	3,78
Knee 7	Femur	8273	0.854	0.921	2801	1,58	2,25
	Tibia		0.689	0.816	5620	5,18	2,57

From the quantitative analysis, some observations can be made:

- Considering the segmentation results, we can verify that the Deep Learning model generalizes to unseen knees from different individuals, presenting good results for the femur and reasonable results for the tibia.
- In both datasets, the segmentation and registration accuracies of the tibia are inferior than for the femur. This is due to the fact that only a small portion of the tibia is exposed, with large occlusions caused by the patella and surrounding tissue.
- In all cases except the registration of the tibia in Knee 7, it can be seen that a significant amount of registration results are considered unsuccessful. This can be explained by the noise of the depth sensor that, according to our experiment in Section 3.5.1, is in the same order of magnitude as the threshold for selecting inliers, causing many registration attempts to be discarded because significant percentages of reconstructed points are considered as outliers. By relaxing this threshold, more registrations would be accepted but the medical accuracy requirements would be compromised.
- For the case of the registration of the tibia in Knee 7, almost 68% of the dataset has registrations considered successful. Since the segmentation accuracy is high, it can be assumed that there are few outliers caused by incorrectly classified pixels. However, the fact that the visible area of the tibia is small and considering the existence of other sources of occlusions, such as objects obstructing the line-of-sight and the camera's viewpoint, it is common for the reconstructed point cloud to be very local, causing the registration algorithm to converge to incorrect solutions with high percentages of inliers. Although the percentage of registrations of the tibia in Knee 6 is larger than of both femurs, this difference is not so evident because i) there are more outliers caused by incorrect segmentations and ii) due to occlusions, the area of reconstructed

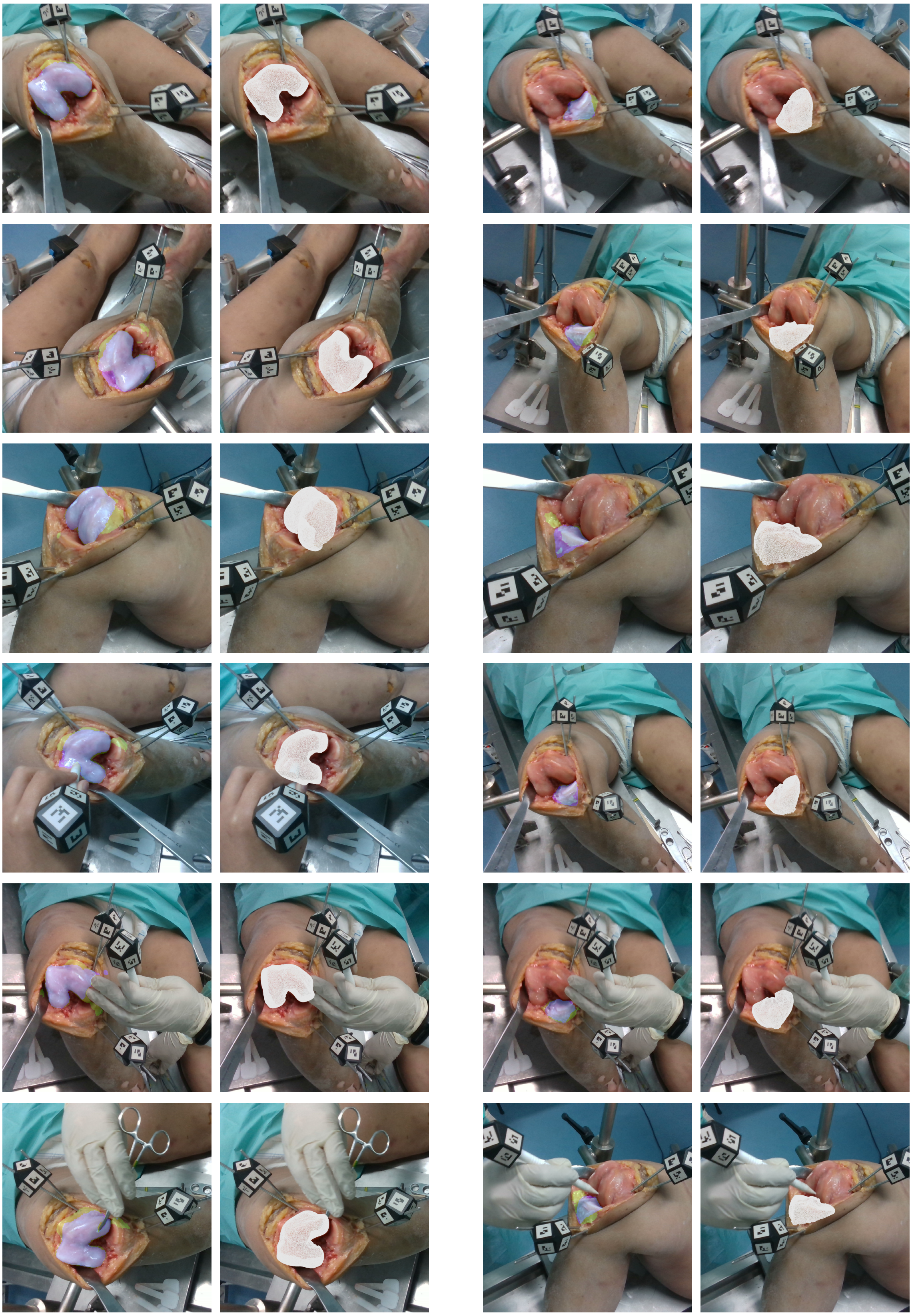
points becomes so small that the algorithm does not even attempt to perform registration.

- Another observation is that the registration accuracy obtained in Knee 7 is higher than that of Knee 6. This comes from the fact that the segmentation results in Knee 7 were also better and, in the case of the tibia, it was better exposed in this dataset.

3.5.2.3 Qualitative results

Figures 3.16 and 3.17 show qualitative examples for segmentation and registration of femur and tibia in the Knee 6 and Knee 7 datasets, respectively.

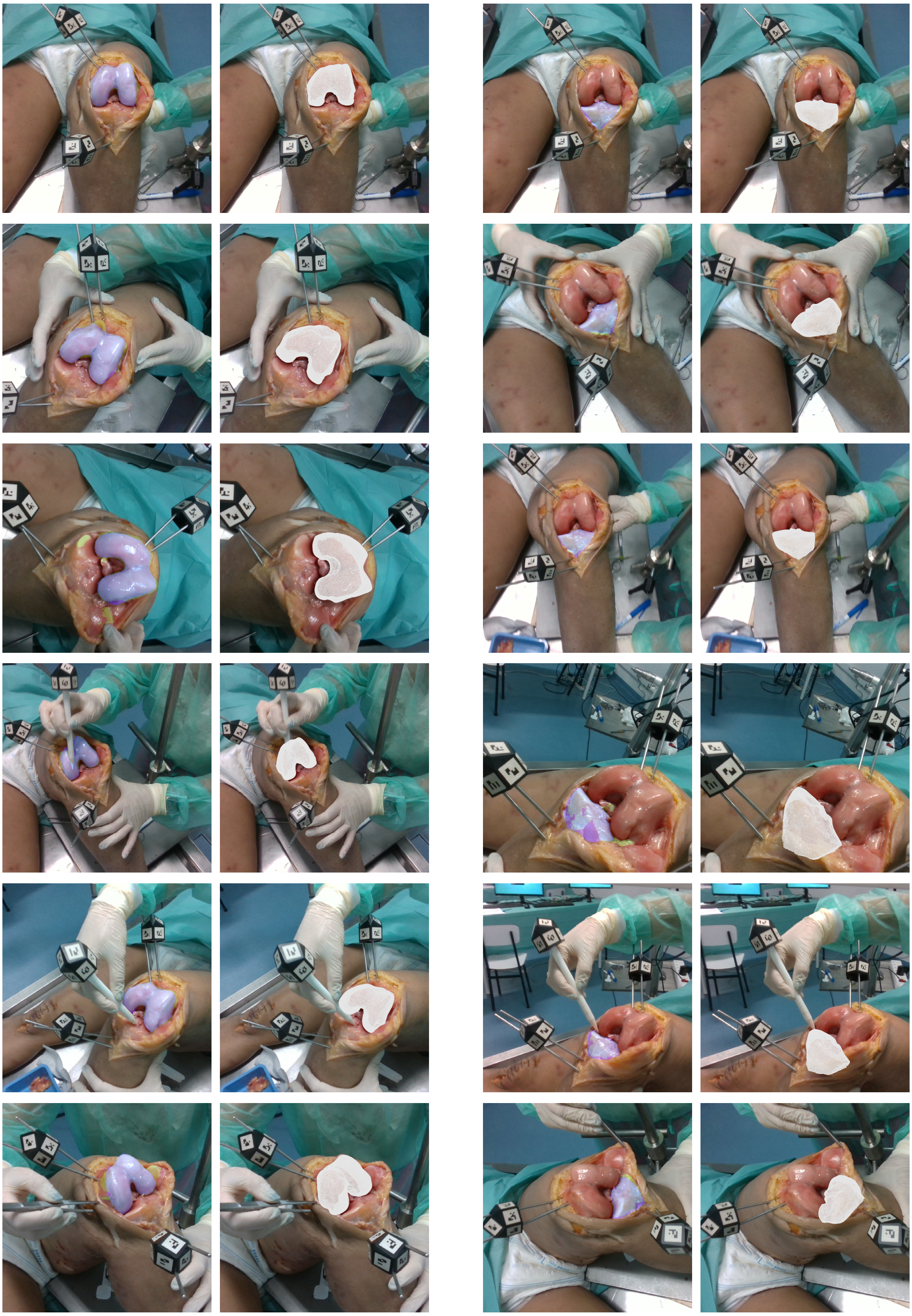
These examples prove that it is feasible to track both bones for different poses with accurate results and even handle small occlusions. Note that for Knee 6 the tibia is less exposed than in Knee 7, which is related to the higher error displayed in Table 3.8.



(a) Femur results.

(b) Tibia results.

Figure 3.16: Images from the Knee 6 dataset with overlaid segmentation masks (ground truth: purple; predicted: green; both: cyan) and projection of the registered 3D model.

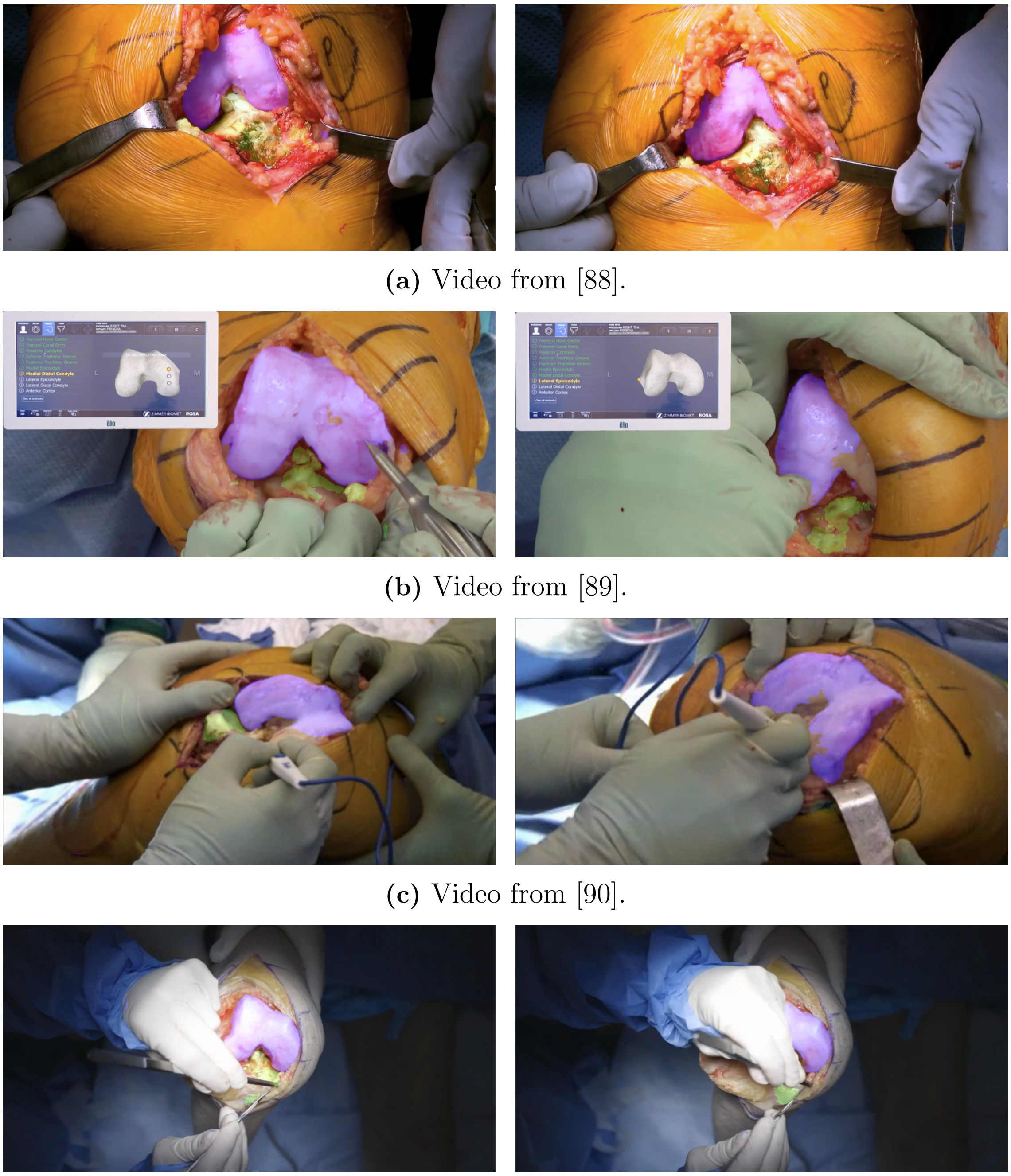


(a) Femur results.

(b) Tibia results.

Figure 3.17: Images from the Knee 7 dataset with overlaid segmentation masks (ground truth: purple; predicted: green; both: cyan) and projection of the registered 3D model.

To further test the Deep Learning model, we ran it in images from videos found in the VuMedi [1] library, to predict femur and tibia segmentations. From the examples presented in Figure 3.18, it is possible to verify that the model generates accurate predictions, being able to generalize to different surgeries.



(a) Video from [88].

(b) Video from [89].

(c) Video from [90].

(d) Video from [91].

Figure 3.18: Images from videos found in the VuMedi library with overlaid segmentation predicted masks (femur: purple; tibia: green).

[1] `www.vumedi.com`

3.5.2.4 Comparison with the baseline method

For direct comparison with the work of [2], the author provided the trained model to run in our test set. The model was trained on approximately 9000 images from the datasets of Knee 2 and 3. However, it was also tested in images from Knee 3, not presenting additional experiments to analyse the model performance on unseen data. In these conditions, the model scored an IoU of median 0.853 and a Dice coefficient of median 0.921 for femur segmentation.

Table 3.9 presents the segmentation results using the trained model from [2] in our test datasets. Note that with this model, it is only possible to predict femur segmentation masks.

Table 3.9: Segmentation results comparing our model with the model from [2] in our test datasets.

			Our results		**Results with [2]**	
Dataset	**Bone**	**Number of images**	**Median IoU**	**Median Dice**	**Median IoU**	**Median Dice**
Knee 6	Femur	8376	0.794	0.885	0.350	0.461
Knee 7	Femur	8273	0.854	0.921	0.488	0.616

Considering the results obtained by the model from [2] in our test set, we can state that it doesn't perform as well on unseen bones. Comparing with the results achieved with our final model, we can establish that the addition of new data to the training set and the adjustments made to the network led to an increase in the generalization power of the method, improving the segmentation step of the system. With these segmentation results, we decided not to perform the registration step since it is obvious that the errors would be worse than ours.

4

Evaluation of video-based surgical navigation system

In order to evaluate how a system with a structure similar to the one in Chapter 3 performs as a 6D pose estimation algorithm, and to simultaneously compare it with the state-of-the-art, a method was submitted to the BOP Challenge 2019, under the name of *Félix&Neves-ICRA2017-IET2019*. This chapter documents its implementation and reports the achieved results.

4.1 BOP Challenge 2019

BOP stands for Benchmark for Object Pose Estimation, whose goal is to capture the state-of-the-art in estimating the 6D pose. The BOP Challenge 2019 consisted in the task of recovering the 6D localization of a varying number of instances of a varying number of objects (ViVo for short) in a single RGB-D image.

4.1.1 Method

The method follows the pipeline presented in Chapter 3:

1. **Segmentation of the object in the image**

   Since U-net is a semantic segmentation network (refer to Figure 2.2), is not suitable for the ViVo task, which requires a segmentation mask for each instance of an object. Therefore, an alternative Deep Learning architecture was implemented to perform the segmentation step of the pipeline. Section 4.1.1.1 describes the chosen architecture.

2. **Extraction of the targeted object's point cloud**

   The provided test RGB-D data was already aligned, so it was only necessary to apply the segmentation mask of each instance to the depth map in order to retrieve its point cloud.

3. **Registration of the reconstructed point cloud with the object's 3D model**

   The registration algorithm from [51] was used to deliver the pose estimation for each instance resulting from the previous step.

4.1.1.1 Deep Learning architecture

The chosen DL architecture to accomplish instance segmentation was Mask-RCNN [92]. This network is introduced as an extension to the R-CNNs [93, 94], that already successfully performed the task of object classification and detection. Considering the Faster R-CNN architecture [94], two different CNN can be used as the backbone, either a ResNet or a Feature Pyramid Network (FPN), for feature extraction. Then follows a Region Proposal Network (RPN) to find Regions of Interest (ROI) and finally a ROIPool extracts feature maps from each region and performs classification and bounding-box regression. The Mask R-CNN architecture adds a branch to this last stage, with a Fully Convolutional Network (FCN) that generates a mask to each region [92].

The implementation was provided by [95], which uses Keras [96], a deep learning interface that runs on top of Tensorflow [97], a machine learning platform. The implementation was adapted to the BOP datasets format.

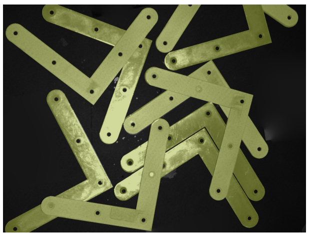
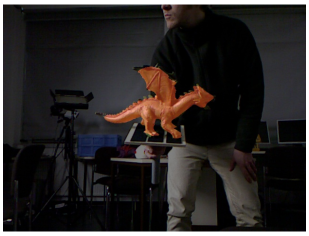
Our implementation considered the ResNet backbone, since it reported better performance in [92]. Additionally, the network was initialized with pre-trained weights on the MS COCO dataset.

4.1.2 Experimental setup

4.1.2.1 Datasets

BOP combines different datasets that were previously introduced in the literature for the evaluation of methods for 6D object pose estimation. Table 4.1 shares details about each of the datasets, such as the number of objects, number of test images, type of training data provided and conditions on the test images.

Table 4.1: Description of the datasets from the BOP Challenge 2019. Figures from [3].

	LM-O [69] 15 objects 200 test images Synthetic training data Objects with occlusions in cluttered scenes
	T-LESS [98] 30 objects 1000 test images Synthetic training data Varying complexity with cluttered scenes
	**ITODD** [99] 28 objects 721 test images (grayscale) Synthetic training data Several instances of the objects
	HB [100] 16 objects 300 test images Synthetic training data Varying complexity with cluttered scenes
	YCB-V [66] 21 objects 900 test images Real training data Limited clutter
	IC-BIN [101] 2 objects 150 test images Synthetic training data Several instances with heavy occlusion
	**TUD-L** [27] 3 objects 600 test images Real training data Moving objects and different light conditions

4.1.2.2 Training images

Besides the YCB-V and TUD-L datasets, which had real training data, all the other datasets had only synthetic training data available, that was generated by projecting the 3D models at different poses on a black background (Fig. 4.1a).

Training the network with the provided synthetic images would not allow generalization for the test sets. Hence, we generated 30 000 new synthetic images per dataset for training. This was accomplished by adding randomly different objects/ instance of objects with different poses to images from the NYU Depth dataset [102] as the background. Figure 4.1 shows some examples of generated images.

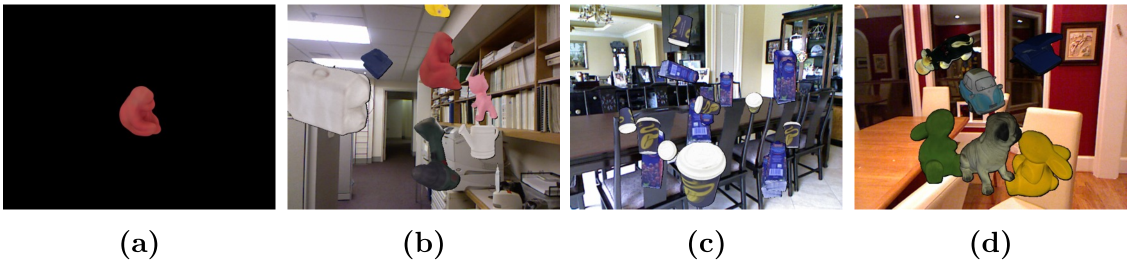


Figure 4.1: Synthetic training images of the BOP datasets: **(a)** provided; **(b-d)** generated.

4.1.2.3 Training the network

A specific model was considered for each dataset. Real training images were used, if available, otherwise, the network was trained with the generated synthetic images.

The network architecture was trained with the configurations of the original implementation [95], across all datasets, since tunning of hyperparameters was not allowed by the challenge rules ("each method has to use an identical set of hyperparameters across all objects and datasets" [3]).

4.1.3 Results

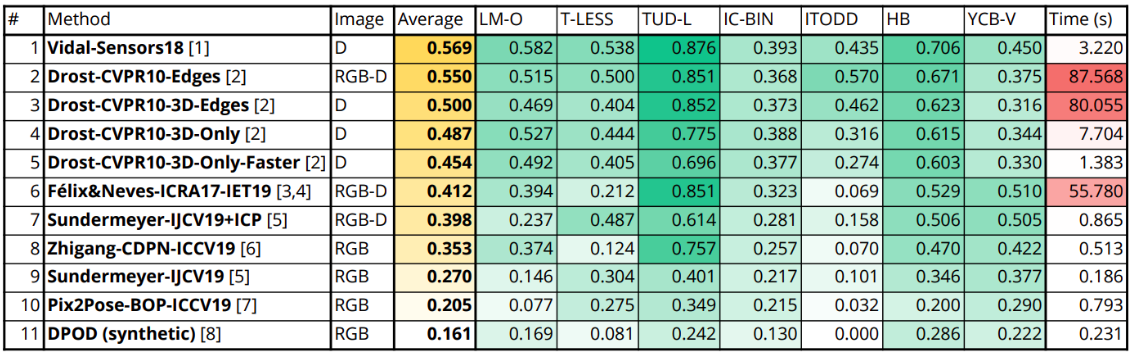
The test images annotated with ground-truth are publicly available for most datasets, except for the HB and ITODD that only have validation images. The instance segmentation performance is assessed using the Mean Average Precision (mAP) metric, for detections with IoU > 0.5. The performance score for the pose estimations in each dataset is measured by the Average Recall (AR) (explained in [3]). Table 4.2 shows the results of our method in the test set of each dataset (validation set for HB and ITODD).

Table 4.2: Results of our method for the BOP datasets.

Dataset	LM-O	T-LESS	ITODD	HB	YCB-V	IC-BIN	TUD-L
mAP	0.483	0.314	0.374	0.667	0.845	0.381	0.974
AR	0.394	0.212	0.065	0.526	0.529	0.510	0.851

From Table 4.2 some conclusions can be drawn: as expected, the segmentation network works better when trained with real data, which leads to better pose estimations; datasets with occlusions and cluttered scenes have increased difficulty.

The final results for the BOP Challenge 2019 are presented in Figure 4.2. Considering the overall results of all participants, our method ranked 6th place.



#	Method	Image	Average	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Time (s)
1	**Vidal-Sensors18** [1]	D	**0.569**	0.582	0.538	0.876	0.393	0.435	0.706	0.450	3.220
2	**Drost-CVPR10-Edges** [2]	RGB-D	**0.550**	0.515	0.500	0.851	0.368	0.570	0.671	0.375	87.568
3	**Drost-CVPR10-3D-Edges** [2]	D	**0.500**	0.469	0.404	0.852	0.373	0.462	0.623	0.316	80.055
4	**Drost-CVPR10-3D-Only** [2]	D	**0.487**	0.527	0.444	0.775	0.388	0.316	0.615	0.344	7.704
5	**Drost-CVPR10-3D-Only-Faster** [2]	D	**0.454**	0.492	0.405	0.696	0.377	0.274	0.603	0.330	1.383
6	**Félix&Neves-ICRA17-IET19** [3,4]	RGB-D	**0.412**	0.394	0.212	0.851	0.323	0.069	0.529	0.510	55.780
7	**Sundermeyer-IJCV19+ICP** [5]	RGB-D	**0.398**	0.237	0.487	0.614	0.281	0.158	0.506	0.505	0.865
8	**Zhigang-CDPN-ICCV19** [6]	RGB	**0.353**	0.374	0.124	0.757	0.257	0.070	0.470	0.422	0.513
9	**Sundermeyer-IJCV19** [5]	RGB	**0.270**	0.146	0.304	0.401	0.217	0.101	0.346	0.377	0.186
10	**Pix2Pose-BOP-ICCV19** [7]	RGB	**0.205**	0.077	0.275	0.349	0.215	0.032	0.200	0.290	0.793
11	**DPOD (synthetic)** [8]	RGB	**0.161**	0.169	0.081	0.242	0.130	0.000	0.286	0.222	0.231

Figure 4.2: BOP Challenge 2019 results [3], where the performance is measured by the Average Recall (AR) for each dataset and the overall score is calculated as the average of the per-dataset scores. Our method ranked 6th place.

Methods classified above ours are based in point pair features and use mostly depth information [58, 57]. We were the first classified that used a DL-based method. Actually, this year, the challenge opened again with the addition of photorealistic synthetic training data for all datasets to reduce the entry barrier of DL-based solutions. This made a huge difference and currently the DL-based methods surpassed the previous leaders. We did not had the opportunity to submit this year, however, we intend to in the future, since we believe the new training data would help improve the method's performance.

5

Automatic detection of anatomical landmarks for image-free navigation

This chapter overviews the concept for automatic detection of landmarks in RGB images. At this early stage of work, validation of the method will be performed only for one specific landmark of the femur (whiteside's line). However, the same methodology can be extended to the remaining landmarks of the femur and tibia.

5.1 Method overview

The U-net architecture was already explored to perform the task of landmark localization in medical image data, more specifically applied to hand images [103], achieving great results. Therefore, we decided to adapt the implementation previously presented in Section 3.2 for this purpose.

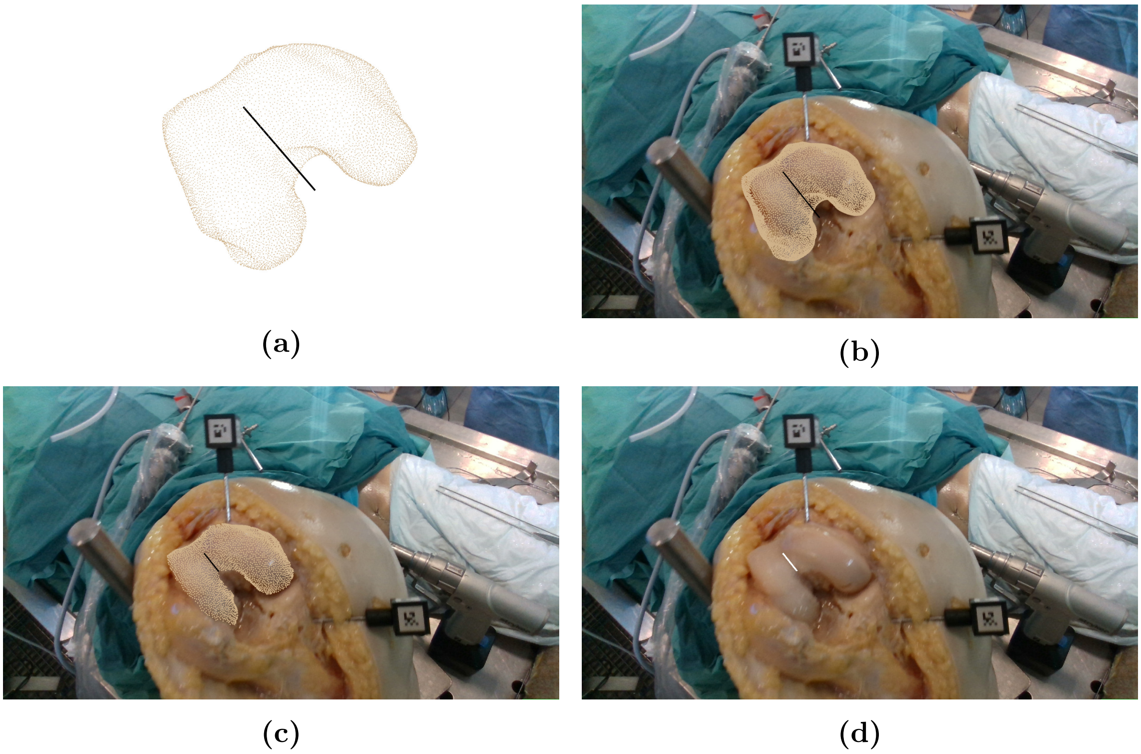
In this case, the network has to be trained individually for each landmark, since some of the landmarks are located in the same pixels, e.g. knee center of the femur with the whiteside's line and knee center of the tibia with the anterior-posterior axis. Thus, the DL network is trained considering only one positive class and the resulting output is a binary mask (with each pixel classified into landmark or background), that is created by using a sigmoid activation in the last convolutional layer.

A new labelling scheme had to be developed for generating the landmark segmentation masks, required to train the deep learning model (described in Section 5.1.1).

The segmentation metrics are considered for the training process of the network, however, these are not the metrics used to assess the performance of the method. Instead, the network's prediction masks undergo a post-processing step in order to deliver the final results (all the details in Section 5.1.2).

5.1.1 Generating label images

The process to generate labels to the images is automatic, divided into four steps, illustrated in Figure 5.1. The landmarks were acquired intra-operatively by an orthopedic surgeon and saved in the model's reference frame (Fig. 5.1a). The model and the landmarks are projected to the RGB image using the initial model-to-anatomy registration and the detected marker pose (Fig. 5.1b). The labeled segmentation of the bone earlier performed (Section 3.2.2) is applied to define the points of the model that are exposed (Fig. 5.1c). This step allows to deal with occlusions, if applicable. Finally, the region related to the landmark is used to create a pixelwise mask (Fig. 5.1d).


(a)

(b)

(c)

(d)

Figure 5.1: Steps for generating landmarks segmentation masks: **(a)** whiteside's line depicted in black in the model's reference frame; **(b)** Model and landmark projected into one image; **(c)** Projection after applying the bone segmentation; **(d)** The generated mask for the landmark is overlaid in the image (in white).

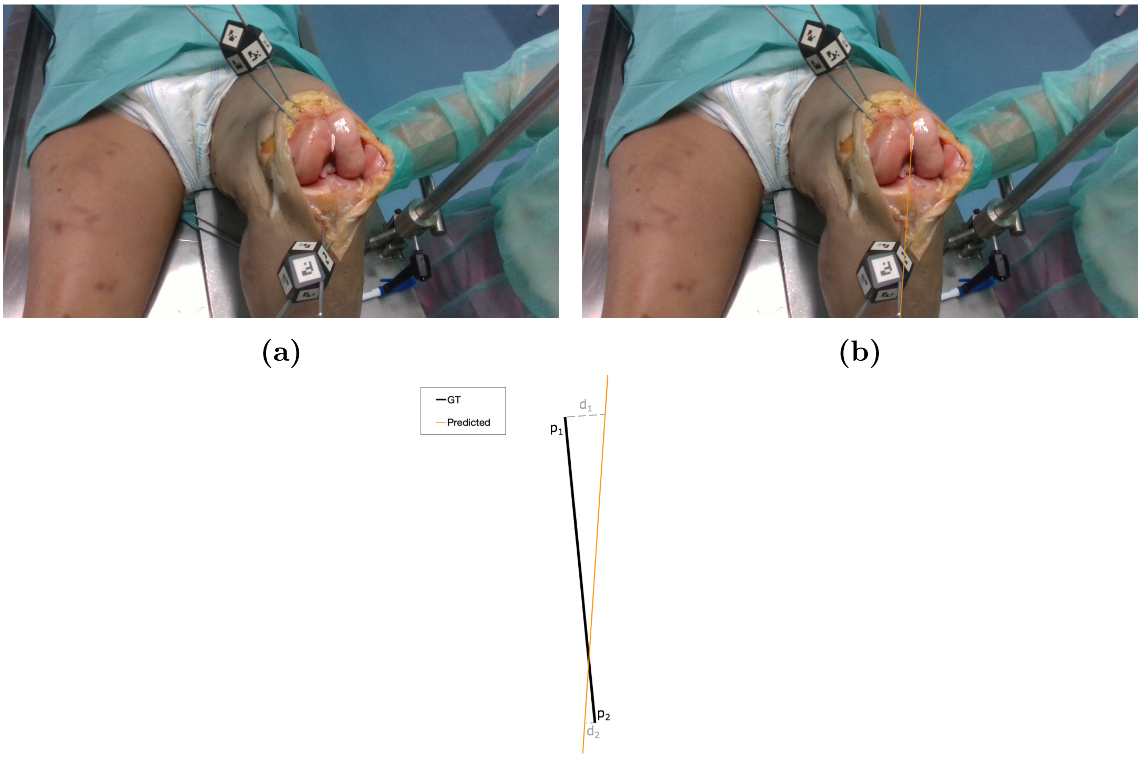
5.1.2 Evaluation protocol

Some post-processing steps need to be executed after retrieving the DL model's prediction masks, in order to assess its performance. The post-processing is implemented in MATLAB, resorting to the *geom2d* toolbox [104].

Regarding the landmark detection task, we aim at two different types of outputs: either a point, for landmarks such as the knee center and anterior cortex; or a line when referring to the whiteside's line and the anterior-posterior axis.

Line output The pixels classified as positives in the predicted segmentation mask are taken as inputs to the *lineFit* function [104], that fits a straight line to this set of points. Then, the orthogonal distance of each endpoint of the ground-truth line segment (p_1 and p_2) to the predicted line is calculated, with the function *distancePointLine* [104], resulting the values d_1 and d_2, that are averaged to measure the error, in px. For a better comprehension, these operations are represented in Figure 5.2.

Point output For this type of output, it is determined the centroid of the pixels classified as positives in the predicted segmentation mask. Then, the distance between the ground-truth point and the predicted corresponds to the error.



(a)

(b)

(c)

Figure 5.2: Evaluation protocol for line outputs: **(a)** Image with overlaid predicted segmentation mask (in white); **(b)** Line fitted to the set of predicted points (in orange); **(c)** Operation to measure the error between the ground-truth line segment and the predicted line.

5.2 Experiments and results

This section reports the experiments for testing the validity of the proposed method and its results. As mentioned before, the experiments were only conducted for the whiteside's line.

5.2.1 Training the network

From the ex-vivo experiments described in Section 3.1.2, only four had ground-truth available (landmarks digitalized intra-operatively): Knee 4, 5, 6 and 7. Since the data we have is limited, the same dataset is used for both validation and testing purposes. Additionally, instead of considering the total images of these experiments, only a small set that avoids the presence of occlusions was used, in order to facilitate the process and test whether the method works in optimal conditions. The dataset distribution is presented in Table 5.1.

Table 5.1: Dataset distribution for landmark detection.

Dataset		**Number of images**
Train	Knee 4	1077
	Knee 5	2026
	Knee 6	1858
Validation/ Test	Knee 7	2147

Once again, a search for the best hyperparameters combination was carried (Table 5.2).

Table 5.2: U-net models trained for landmark detection with different hyperparameter combinations. The best results (model 002) are highlighted in bold.

Model	Number of epochs	Learning rate	Dropout ratio	Freeze encoder	Number of filters	Mean IoU	Mean Dice
001	5	0,000019	0,451	1	32	0,168	0,265
002	**6**	**0,000047**	**0,630**	**0**	**32**	**0,287**	**0,425**
003	10	0,000142	0,578	0	32	0,274	0,408
004	7	0,000021	0,143	0	32	0,249	0,376
005	10	0,000017	0,296	0	32	0,259	0,393
006	10	0,000995	0,239	0	32	0,108	0,170
007	6	0,000018	0,690	1	32	0,163	0,258
008	8	0,000055	0,301	1	16	0,149	0,235
009	8	0,000018	0,793	1	16	0,142	0,226
010	10	0,000349	0,822	1	16	0,228	0,350

Even though the segmentation metrics are not representative of our final results, they are considered to choose the Deep Learning model from Table 5.2. Therefore, the selected model was trained in the complete network (encoder and decoder weights), with the number of filters set to 32, a learning rate of approximately 5e-5 and a dropout probability of 63% for 6 epochs.

5.2.2 Quantitative results

During the experiment of the Knee 7, two trials were done intra-operatively by an ortopedic surgeon to acquire the bone landmarks (Figure 5.3a). Thus, one of the measurements was chosen to be the ground truth, and the other will be considered to compare the amount of error introduced by the proposed method with the intra-observer error.

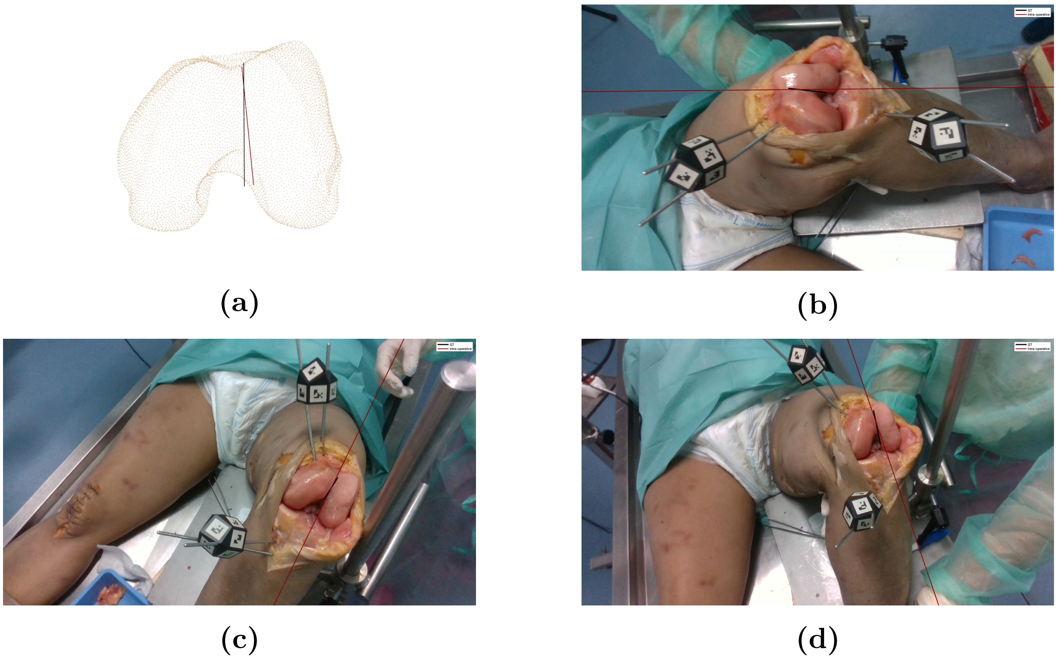


Figure 5.3: Measurements for the whiteside's line acquired intra-operatively in Knee 7 **(a)** represented in the model's reference frame and **(b-d)** projected into images. The measurement chosen as the ground truth is depicted in black.

The model is tested in the images of Knee 7 and the predictions are evaluated following the directions presented previously (Section 5.1.2). Additionally, the evaluation methodology is also performed to the projection of the second intra-operative acquisition of the same landmark (represented in red in Figure 5.3). Figure 5.4 shows the error distribution for both cases.

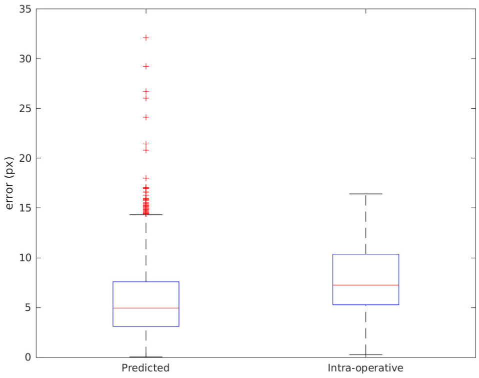


Figure 5.4: Error distribution for the predicted landmark location and for the projection of the landmark acquired intra-operatively in the test images.

For our method predictions, the obtained median error was 4.94 px, while the test for the landmark acquired intra-operatively achieved a median error of 7.25 px. Our method can generate outliers in some images (50 images in the presented distribution), however, these correspond to only a small percentage of the dataset. From these results it's clear that the automatic method is closer to the ground truth than the other measurement acquired intra-operatively.

5.2.3 Qualitative results

Figure 5.5 allows visualization of the obtained results in the test dataset.

Given that the model was validated and tested in one knee that is similar to one that is present at training time (since they belong to the same individual), further tests were needed to establish that the proposed method can generalize. Therefore, the method was tried in another knee (Knee 8) and the qualitative results are provided in Figure 5.6. Although this knee doesn't have ground truth acquired intra-operatively, it's possible to see the dashed lines physically marked in the bone that were made by an experienced surgeon and compare with our results, which have proven to be very accurate.

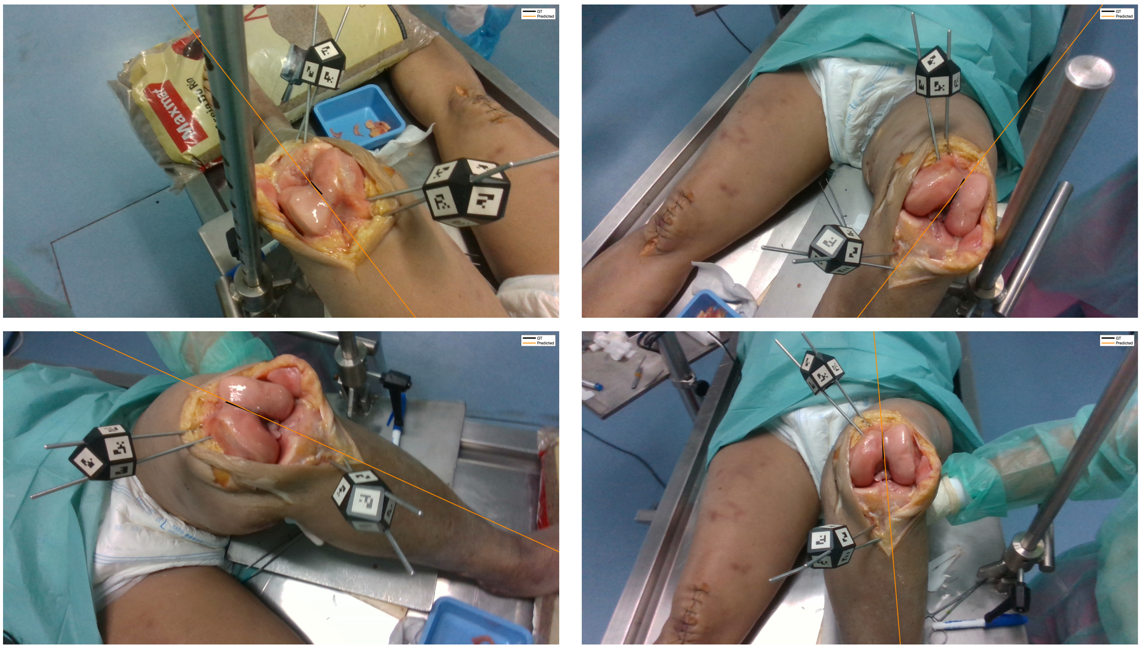


Figure 5.5: Images from the test dataset with the ground-truth (black) and the predicted (orange) whiteside's line.

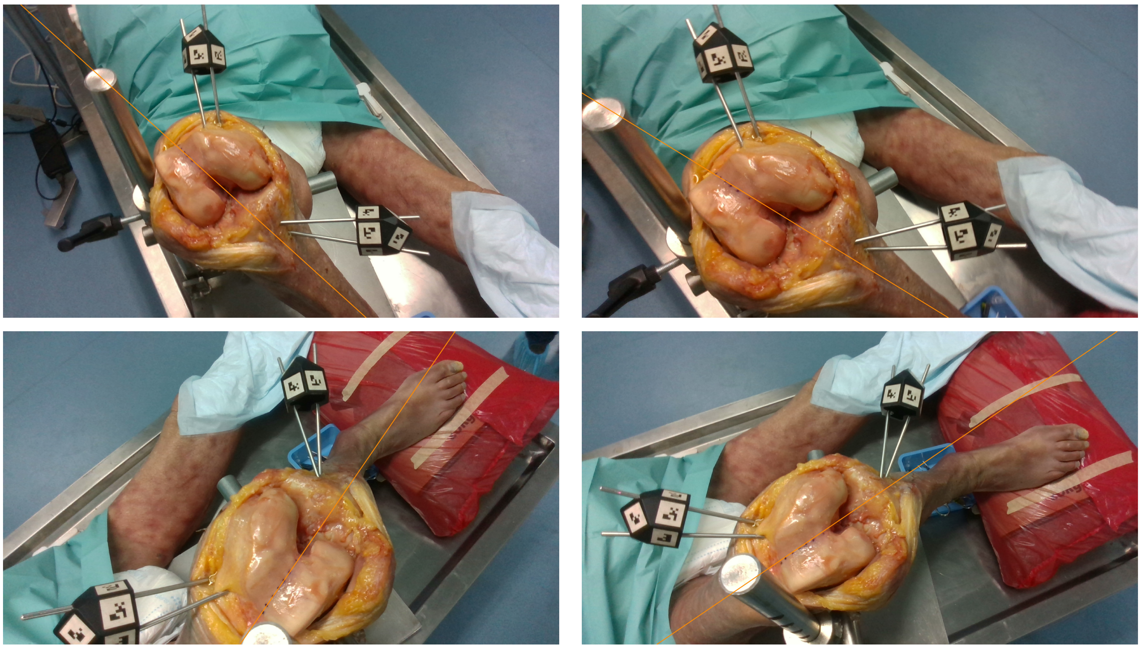


Figure 5.6: Images from another dataset with the predicted (orange) whiteside's line. The dashed lines physically marked in the bone were made by an experienced surgeon to localize the landmark.

6

Discussion and future work

Total Knee Arthroplasty (TKA) is a surgical procedure performed in patients suffering from knee arthritis. Several computer-based navigation systems have been developed in order to assist the surgeon in performing the surgery with optimal outcome. This thesis focused in navigation for TKA and addressed two problems that are pointed by many as fundamental for its broader acceptance.

The main part of this thesis was committed to surpassing the limitations of the system proposed by [2] to allow video-based navigation in TKA surgery without the help of fiducial markers attached to the knee bones:

1. The proposed algorithm works well for the femur and has difficulties in handling the tibia. The main problem of the tibia is that only a small portion of the bone is exposed, which complicates both the segmentation and registration stages. Nonetheless, results in tracking the tibia are encouraging and can be improved with additional training data;

2. It improves considerably the performance over the work of [2] in terms of accuracy and generalization, as shown in Section 3.5.2.4;

3. We achieved results within the accurate requirements of clinical practice, however, these are highly dependent on the dataset. Still, as analyzed in section 3.5.1, the main limitation currently is the depth sensor that contributes to a considerable portion of the rotation and translation errors. Nevertheless, and considering the constant improvements in depth sensing research and development [29], it is reasonable to consider that our pipeline will fulfill the medical requirements across all datasets in the near future. This will be a cornerstone for CAS and will make accurate surgical navigation possible, solely based on video, without requiring the attachment of markers to the bones.

As future work, we plan on adding more training data, ideally from in-vivo surgeries, and combine end-to-end the segmentation and the geometric based registration algorithms to allow real time test on real individuals. Additionally, improve for working with bones that were already resected - one possible strategy would be to update the 3D model with the cuts actually performed.

Considering the proposed system as a 6D pose estimation algorithm, additional evaluation of the method's performance was accomplished in the BOP Challenge 2019 (Chapter 4). Comparing with other state-of-the-art approaches, our method obtained competitive results. In the future, a new submission should be developed with the new photorealistic training data to assess possible improvements. Also, these state-of-the-art methods should be tested in our original dataset from the clinical environment.

In the final chapter, a proof-of-concept algorithm is introduced for automatic detection of landmarks using Deep Learning. The proposed method was validated for one landmark and the preliminary results are promising, showing that the characteristic variability associated with this task can be minimized. Extension to the remaining landmarks must be implemented and different DL architectures should be tested.

Overall, the solutions presented in this project are a novelty and future research directions can be derived, that allow the implementation of navigation systems in TKA to become an increasingly accepted reality.

Bibliography

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[2] P. Rodrigues, M. Antunes, C. Raposo, P. Marques, F. Fonseca, and J. P. Barreto, "Deep segmentation leverages geometric pose estimation in computer-aided total knee arthroplasty," *Healthcare Technology Letters*, vol. 6, no. 6, pp. 226–230, 2019.

[3] T. Hodaň, E. Brachmann, D. Bertram, F. Michel, M. Sundermeyer, J. Matas, and C. Rother, "Bop challenge," 2019.

[4] R. Siston, N. Giori, S. Goodman, and S. Delp, "Surgical navigation for total knee arthroplasty: A perspective," *Journal of biomechanics*, vol. 40, pp. 728–35, 02 2007.

[5] K. Deep, S. Shankar, and A. Mahendra, "Computer assisted navigation in total knee and hip arthroplasty," *SICOT-J*, vol. 3, p. 50, 07 2017.

[6] T. Tjardes, S. Shafizadeh, D. Rixen, T. Paffrath, B. Bouillon, E. S. Steinhausen, and H. Baethis, "Image-guided spine surgery: state of the art and future directions," *European Spine Journal*, vol. 19, pp. 25–45, sep 2009.

[7] C. Raposo, J. P. Barreto, C. Sousa, L. Ribeiro, R. Melo, J. P. Oliveira, P. Marques, F. Fonseca, and D. Barrett, "Video-based computer navigation in knee arthroscopy for patient-specific ACL reconstruction," *International journal of computer assisted radiology and surgery*, vol. 14, p. 1529—1539, September 2019.

[8] G. Zheng and L. Nolte, "Computer-Assisted Orthopedic Surgery: Current State and Future Perspective," *Frontiers in Surgery*, vol. 2, 12 2015.

[9] A. Mavrogenis, O. Savvidou, G. Mimidis, J. Papanastasiou, D. Koulalis, N. Demertzis, and P. Papagelopoulos, "Computer-assisted Navigation in Orthopedic Surgery," *Orthopedics*, vol. 36, pp. 631–42, 08 2013.

[10] S. Kurtz, K. Ong, E. Lau, F. Mowat, and M. Halpern, "Projections of Primary and Revision Hip and Knee Arthroplasty in the United States from 2005 to 2030," *The Journal of bone and joint surgery. American volume*, vol. 89, pp. 780–5, 04 2007.

[11] J. Van der List, H. Chawla, L. Joskowicz, and A. Pearle, "Current state of computer navigation and robotics in unicompartmental and total knee arthroplasty: a systematic review with meta-analysis," *Knee Surgery Sports Traumatology Arthroscopy*, vol. 24, pp. 3482–3495, 11 2016.

[12] H. Bäthis, L. Perlick, M. Tingart, C. Lüring, D. Zurakowski, and J. Grifka, "Alignment in total knee arthroplasty. A comparison of computer-assisted surgery with the conventional technique," *The Journal of bone and joint surgery. British volume*, vol. 86, pp. 682–7, 08 2004.

[13] M. S. Zihlmann, A. Stacoff, J. Romero, I. Kramers-de Quervain, and E. Stüssi, "Biomechanical background and clinical observations of rotational malalignment in TKA:: Literature review and consequences," *Clinical biomechanics*, vol. 20, no. 7, pp. 661–668, 2005.

[14] E.-K. Song, J. Seon, J.-H. Yim, N. Netravali, and W. Bargar, "Robotic-assisted TKA Reduces Postoperative Alignment Outliers and Improves Gap Balance Compared to Conventional TKA," *Clinical orthopaedics and related research*, vol. 471, 06 2012.

[15] M. H. L. Liow, Z. Xia, M. K. Wong, K. J. Tay, S. J. Yeo, and P. L. Chin, "Robot-assisted total knee arthroplasty accurately restores the joint line and mechanical axis. A prospective randomised study," *The Journal of arthroplasty*, vol. 29, no. 12, pp. 2373–2377, 2014.

[16] M. P. Bonnin, A. Schmidt, L. Basiglini, N. Bossard, and E. Dantony, "Mediolateral oversizing influences pain, function, and flexion after TKA," *Knee Surgery, Sports Traumatology, Arthroscopy*, vol. 21, no. 10, pp. 2314–2324, 2013.

[17] A. Todesca, L. Garro, M. Penna, and J. Bejui-Hugues, "Conventional versus computer-navigated TKA: a prospective randomized study," *Knee surgery,*

sports traumatology, arthroscopy : official journal of the ESSKA, vol. 25, p. 1778—1783, June 2017.

[18] M. R. Stulberg SD, Saragaglia D, "Total knee replacement: Navigation technique intraoperative model system," in *Computer and Robotic Assisted Hip and Knee Surgery* (P. R. N. P. e. DiGioia AM, Jaramaz B, ed.), ch. 14, pp. 157 – 178, Oxford University Press, 2004.

[19] A. M. Franz, T. Haidegger, W. Birkfellner, K. Cleary, T. M. Peters, and L. Maier-Hein, "Electromagnetic tracking in medicine—a review of technology, validation, and applications," *IEEE transactions on medical imaging*, vol. 33, no. 8, pp. 1702–1725, 2014.

[20] R. A. Siston, N. J. Giori, S. B. Goodman, and S. L. Delp, "Surgical navigation for total knee arthroplasty: a perspective," *Journal of biomechanics*, vol. 40, no. 4, pp. 728–735, 2007.

[21] E. K. Song, J. K. Seon, S. J. Park, and T. R. Yoon, "Accuracy of navigation: a comparative study of infrared optical and electromagnetic navigation," *Orthopedics*, vol. 31, no. 10, p. 76, 2008.

[22] R. Simões, C. Raposo, J. P. Barreto, P. S. Edwards, and D. Stoyanov, "Visual tracking vs optical tracking in computer-assisted intervention," *P3D technical report*, 2018.

[23] D. Yang, S. Zhang, Z. Yan, C. Tan, K. Li, and D. Metaxas, "Automated anatomical landmark detection ondistal femur surface using convolutional neural network," in *2015 IEEE 12th international symposium on biomedical imaging (ISBI)*, pp. 17–21, IEEE, 2015.

[24] N. Xue, M. Doellinger, C. P. Ho, R. K. Surowiec, and R. Schwarz, "Automatic detection of anatomical landmarks on the knee joint using mri data," *Journal of Magnetic Resonance Imaging*, vol. 41, no. 1, pp. 183–192, 2015.

[25] K. Subburaj, B. Ravi, and M. Agarwal, "Automated identification of anatomical landmarks on 3d bone models reconstructed from ct scan images," *Computerized Medical Imaging and Graphics*, vol. 33, no. 5, pp. 359–368, 2009.

[26] J. Beldame, P. Boisrenoult, and P. Beaufils, "Pin track induced fractures around computer-assisted tka," *Orthopaedics & Traumatology: Surgery & Research*, vol. 96, no. 3, pp. 249–255, 2010.

[27] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," *European Conference on Computer Vision (ECCV)*, 2018.

[28] T. Hodan, D. Barath, and J. Matas, "Epos: Estimating 6d pose of objects with symmetries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11703–11712, 2020.

[29] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with rgb-d cameras," in *Computer graphics forum*, vol. 37, pp. 625–652, Wiley Online Library, 2018.

[30] E. G. Haritinian and A. L. Pimpalnerkar, "Computer assisted total knee arthroplasty: does it make a difference?," *Maedica*, vol. 8, no. 2, p. 176, 2013.

[31] M. Robinson, D. G. Eckhoff, K. D. Reinig, M. M. Bagur, and J. M. Bach, "Variability of landmark identification in total knee arthroplasty.," *Clinical Orthopaedics and Related Research (1976-2007)*, vol. 442, pp. 57–62, 2006.

[32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[33] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354 – 377, 2018.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[36] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017.

[37] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, IGI global, 2010.

[38] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659*, 2017.

[39] T. Yu and H. Zhu, “Hyper-parameter optimization: A review of algorithms and applications,” 2020.

[40] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.

[41] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[42] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” *CoRR*, vol. abs/1212.5701, 2012.

[43] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[44] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[45] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014.

[46] A. Shvets, A. Rakhlin, A. A. Kalinin, and V. Iglovikov, “Automatic instrument segmentation in robot-assisted surgery using deep learning,” *CoRR*, vol. abs/1803.01207, 2018.

[47] S. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and K. Khan, “Medical image analysis using convolutional neural networks: A review,” *Journal of Medical Systems*, vol. 42, p. 226, 10 2018.

[48] D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 315–337, 2000. PMID: 11701515.

[49] C. Raposo and J. P. Barreto, “3d registration of curves and surfaces using local differential information,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9300–9308, 2018.

[50] M. TKA”, “Surgical guide, stryker.” `https://www.strykermeded.com/media/2223/mako-tka-surgical-guide.pdf`. (Accessed: 30-08-2020).

[51] C. Raposo and J. P. Barreto, “Using 2 point+ normal sets for fast registration of point clouds with small overlap,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5652–5658, IEEE, 2017.

[52] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[53] Q.-Y. Zhou, J. Park, and V. Koltun, “Fast global registration,” in *European Conference on Computer Vision*, pp. 766–782, Springer, 2016.

[54] D. Aiger, N. J. Mitra, and D. Cohen-Or, “4-points congruent sets for robust pairwise surface registration,” in *ACM SIGGRAPH 2008 papers*, pp. 1–10, 2008.

[55] N. Mellado, D. Aiger, and N. J. Mitra, “Super 4pcs fast global pointcloud registration via smart indexing,” in *Computer Graphics Forum*, vol. 33, pp. 205–215, Wiley Online Library, 2014.

[56] M. Mohamad, M. T. Ahmed, D. Rappaport, and M. Greenspan, “Super generalized 4pcs for 3d registration,” in *2015 International Conference on 3D Vision*, pp. 598–606, IEEE, 2015.

[57] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3d object recognition,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 998–1005, 2010.

[58] J. Vidal, C. Lin, and R. Martí, “6d pose estimation using an improved method based on point pair features,” *CoRR*, vol. abs/1802.08516, 2018.

[59] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[60] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal Of Computer Vision*, vol. 81, pp. 155–166, 2009.

[61] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[62] M. Rad and V. Lepetit, “BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth,” *CoRR*, vol. abs/1703.10896, 2017.

[63] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, “Segmentation-driven 6d object pose estimation,” *CoRR*, vol. abs/1812.02541, 2018.

[64] Z. Li, G. Wang, and X. Ji, “Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7677–7686, 2019.

[65] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, “SSD-6D: making rgb-based 3d detection and 6d pose estimation great again,” *CoRR*, vol. abs/1711.10006, 2017.

[66] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” *CoRR*, vol. abs/1711.00199, 2017.

[67] M. Sundermeyer, Z. Marton, M. Durner, and R. Triebel, “Augmented autoencoders: Implicit 3d orientation learning for 6d object detection,” *International Journal of Computer Vision*, vol. 128, 10 2019.

[68] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, “Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes,” in *2011 International Conference on Computer Vision*, pp. 858–865, 2011.

[69] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6d object pose estimation using 3d object coordinates,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 536–551, Springer International Publishing, 2014.

[70] D. Xu, D. Anguelov, and A. Jain, “Pointfusion: Deep sensor fusion for 3d bounding box estimation,” *CoRR*, vol. abs/1711.10871, 2017.

[71] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, “Densefusion: 6d object pose estimation by iterative dense fusion,” 2019.

[72] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu, “6-pack: Category-level 6d pose tracker with anchor-based keypoints,” 2019.

[73] E. Stindel, J. Briard, P. Merloz, S. Plaweski, F. Dubrana, C. Lefevre, and J. Troccaz, "Bone morphing: 3d morphological data for total knee arthroplasty," *Computer Aided Surgery*, vol. 7, no. 3, pp. 156–168, 2002.

[74] H. Bäthis, L. Perlick, M. Tingart, C. Lüring, D. Zurakowski, and J. Grifka, "Alignment in total knee arthroplasty: a comparison of computer-assisted surgery with the conventional technique," *The Journal of bone and joint surgery. British volume*, vol. 86, no. 5, pp. 682–687, 2004.

[75] S. . Nephew, "Navio surgical system surgical technique for total knee arthroplasty." `https://www.smith-nephew.com/global/assets/pdf/navio%20tka%20surgical%20technique%200718%2014529%20v1%20500095%20revc.pdf`. (Accessed: 30-08-2020).

[76] Intellijoint, "Intellijoint knee^{TM}." `https://www.intellijointsurgical.com/knee/`. (Accessed: 30-08-2020).

[77] "FFmpeg." `https://www.ffmpeg.org/`.

[78] Occipital, "Structure Core - Specs and Data." `https://structure.io/structure-core/specs`. Accessed: 2020-05-11.

[79] I. RealSense, "Intel RealSense Depth Camera D435i." `https://www.intelrealsense.com/depth-camera-d435i`. Accessed: 2020-05-11.

[80] I. RealSense, "Intel RealSense SDK." `https://github.com/IntelRealSense/librealsense`. Version 2.0.

[81] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," *CoRR*, vol. abs/1801.05746, 2018.

[82] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[83] T. Zhou, S. Ruan, and S. Canu, "A review: Deep learning for medical image segmentation using multi-modality fusion," *Array*, vol. 3-4, p. 100004, 2019.

[84] M. Khened, A. Varghese, and G. Krishnamurthi, "Fully convolutional multi-scale residual densenets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers," *CoRR*, vol. abs/1801.05173, 2018.

[85] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, pp. 8026–8037, 2019.

[86] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, 2020.

[87] A. Z. A. Aziz, H. Wei, and J. Ferryman, "On evaluation of depth accuracy in consumer depth sensors," in *Eighth International Conference on Machine Vision (ICMV 2015)* (A. Verikas, P. Radeva, and D. Nikolaev, eds.), vol. 9875, pp. 47 – 52, International Society for Optics and Photonics, SPIE, 2015.

[88] Z. Biomet and C. J. D. Valle, "Surgical demonstration of persona® partial knee." `https://www.vumedi.com/video/personar-partial-knee-surgical-video/`. (Accessed: 05-07-2020).

[89] Z. Biomet, G. Klein, and H. Levine, "Robotic-assisted tka utilizing the rosa® knee system." `https://www.vumedi.com/video/robotic-assisted-tka-utilizing-the-rosar-knee-system-/`. (Accessed: 05-07-2020).

[90] M. A. Ritter and M. Berend, "Primary total knee replacement." `https://www.vumedi.com/video/primary-total-knee-replacement/`. (Accessed: 05-07-2020).

[91] Orthalign, "Reducing the challenges of kinematic knee arthroplasty." `https://www.vumedi.com/video/reducing-the-challenges-of-kinematic-knee-arthroplasty/`. (Accessed: 05-07-2020).

[92] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[93] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.

[94] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[95] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow." `https://github.com/matterport/Mask_RCNN`, 2017.

[96] F. Chollet *et al.*, "Keras," 2015.

[97] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[98] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 880–888, IEEE, 2017.

[99] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing mvtec itodd-a dataset for 3d object recognition in industry," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2200–2208, 2017.

[100] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[101] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6d object pose and predicting next-best-view in the crowd," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3583–3592, 2016.

[102] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[103] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Regressing heatmaps for multiple landmark localization using cnns," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 230–238, Springer, 2016.

[104] D. Legland, "geom2d." `https://www.mathworks.com/matlabcentral/fileexchange/7844-geom2d`. Accessed: 2020-08-27.