Faculty of Science
Department of Physics

Advances in biplanar X-ray imaging: calibration and 2D/3D registration

Thesis submitted for the degree of doctor of Science: Physics at the University of Antwerp to be defended by Van Thi Huyen NGUYEN

Promoters:
Prof. Dr. Jan Sijbers
Prof. Dr. Ir. Jan De Beenhouwer

Antwerp, 2022

Doctoral committee:
Prof. Dr. Joris J. J. Dirckx
Prof. Dr. Peter Aerts
Prof. Dr. Jan Sijbers
Prof. Dr. Ir. Jan De Beenhouwer

Other jury members:
Prof. Caroline Vienne
Prof. Sam Van Wassenbergh
Prof. Luis Filipe Alves Pereira

Contact information:
Van Thi Huyen Nguyen
Vision Lab, Department of Physics
University of Antwerp (CDE)
Universiteitsplein 1, Building N (N1.14)
B-2610 Antwerp, Belgium
+32 (0) 3 265 22 45
van.nguyen@uantwerpen.be
https://visielab.uantwerpen.be/people/van-nguyen

I would like to dedicate this thesis to my loving parents and brother . . .

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Prof. Jan Sijbers. Thank you for giving me the opportunity to pursue my Ph.D. at the University of Antwerp, and introducing me to a very new, yet interesting field of research on X-ray Computed Tomography. Thank you for patiently teaching me the first lessons on X-ray and X-ray imaging when I just started. I also really appreciate that I have been given both freedom to explore my ideas, and close supervision to keep me on the right tracks in my research. Thank you for being encouraging and supportive to let me present my research in the scientific conferences of the field whenever it was possible. I would also like to thank my co-supervisor, Prof. Jan De Beenhouwer, who has always openly shared his technical insights in X-ray Computed Tomography since the start of my Ph.D. Thank you for always being available to discuss and to share with me both the general knowledge about X-ray and X-ray imaging system, as well as the very detailed technical methods to formulate and analyze the problems. I have learned a lot from both of my supervisors, not only about X-Ray CT, but also the method of critical thinking and questioning to improve my knowledge every day. Thank you, Jan and Jan, for always being patient and finding time in your busy schedules to guide me through difficult problems during my Ph.D. at VisionLab. Thank you for spending countless time and effort to help me in writing the scientific articles, and my Ph.D thesis. Without your supervisions and supports, it could not have been possible for me to finish my Ph.D.
I also express my appreciation to my doctoral committee, Prof. Dr. Joris J. J. Dirckx and Prof. Dr. Peter Aerts, who were always kind and supportive throughout my doctoral study. I would also like to thank my doctoral jury members, Prof. Caroline Vienne, Prof. Sam Van Wassenbergh, and Prof.

Luis Filipe Alves Pereira for your valuable time, critical, yet constructive comments and suggestions to help me complete my Ph.D. thesis.
I also want to thank my coworker, Joaquim Sanctorum, who is always enthusiastic, motivating, and helpful. Thank you for always being available for discussions, and updating each other's research to keep a close collaboration and to exchange our knowledge. Thanks a lot to Luis Filipe Alves Pereira, Zhihua Liang for your thoughtful suggestions and discussions about deep learning research. Thank you, Sam Van Wassenbergh, Falk Mielke, Jeroen Van Houtte, for sharing your insightful knowledge in biology and 2D/3D registration. Without your collaborations and kind supports, it would not have been possible for me to finish my Ph.D.
Special thanks to Jonathan Sanctorum. I feel so lucky to have been sharing the office with you, and I really enjoyed our office chats. Thanks for always being a good hearer and giving me your great attitudes. Thanks so much, Tim Elberfeld and Árpád Marinovszki, for making me feel at home when I just arrived in Antwerp.
And nothing could have been possible without my loving family. I want to express my profound gratitude and thankfulness to my loving parents and brother. Thank you a lot for always being by my side and giving me the mental supports during my Ph.D. journey.
To my loving boyfriend, thank you a lot for being in my life, Tu Hoang. Thanks for not being tired of talking about my research, listening to me complaining about everything, and giving me a shoulder whenever I need.
And many thanks to my girls - Thanh Bui, Danh Bui, Minh Nguyen, Trang Bui. I am fortunate to have been knowing you. Thanks for sharing the nice time together and being my mental therapists when I struggled with my research. Knowing you has made my Ph.D. life easier.
Thank you a lot, Hang Bui, for always encouraging and supporting me since I started looking for a Ph.D.
Last but not least, thank you a lot to all of the VisionLab's members and colleagues at the University of Antwerp for being kind and supportive during the time I have been at VisionLab.
Thank you all again for accompanying me along my Ph.D. journey. Your presences and supports have made this Ph.D. possible to me!

Van Nguyen
Antwerp, December 2022

Table of contents

List of figures

List of tables

SUMMARY

X-ray Computed Tomography (CT) is a powerful technique for non-invasive evaluation and visualization of an object's internal structure. A CT image is computed from a set of X-ray radiographs of the object acquired from different directions relative to the object in a preset scanner geometry. To obtain an artifact-free 3D tomographic image of the object, it is crucial to establish a correct geometry description of the system configuration for CT reconstruction. Unfortunately, it is not possible to obtain accurate geometry information without repeating calibration for every new system setting in an experimental X-ray scanner. Geometry calibration typically involves describing and estimating the physical arrangement of the X-ray source, rotation stage, and detector prior to computing the CT image. If the geometry parameters are not derived correctly, the CT image is subjected to misalignment artifacts that destroy the internal detail of the object. In this thesis, a phantom-based calibration method is presented to estimate the geometry setting of a biplanar X-ray CT system. Successfully calibrating the system geometry opens possibilities for different applications using the X-ray radiographs acquired with the same X-ray system.

Chapter 1 and 2 provide background knowledge on X-ray imaging and X-ray CT, as well as foundations to build and train a deep neural model. A typical residual neural network (ResNet) is also presented in chapter 2. Chapter 3 focuses on calibrating an experimental biplanar X-ray cone-beam system using a LEGO phantom. More specifically, a procedure to construct the calibration phantom using the LEGO bricks and metal markers, followed by a deep learning-based marker tracking method and geometry alignment, will be presented. Chapter 4 discusses 2D/3D registration, one of the X-ray CT applications in biomedical imaging

for the study of animal kinematics. The registration method aligns the projections of reference 3D landmarks to a set of 2D landmarks for an estimation of the object's 3D pose. An automated 3D landmark extraction procedure and a trained deep neural network facilitate 2D landmark detection and tracking in fluoroscopy images.
The summary of each chapter is as follows.

Chapter 1 - X-ray computed tomography

Chapter 1 aims to provide background knowledge for further studies in geometry calibration of an X-ray acquisition system and 2D/3D registration using X-ray cone-beam data. First, the principle of X-ray physics and X-ray imaging are discussed. Next, a practical projection model describing X-ray beam-voxel interaction as well as a linear model for X-ray computed tomography are presented. Finally, a review on a common X-ray circular cone-beam geometry and basic geometric transformations provides foundation for the followed applications in the geometry calibration, and 2D/3D registration. This knowledge will be revisited in the subsequent chapters.

Chapter 2 - Deep neural network

Chapter 2 discusses fundamentals of machine learning and deep learning building blocks and algorithms. The basic concept of machine learning and deep learning models, followed by an overview of the current deep network architectures are presented in the first section. Furthermore, the widely-used convolutional neural network, its building blocks, as well as techniques that are commonly employed when implementing a convolutional neural network model are also discussed. Finally, ResNet architecture is presented to provide background knowledge of a deep network model that will be applied in marker and landmark tracking applications.

Chapter 3 - Biplanar cone-beam geometry calibration

Chapter 3 presents a phantom-based calibration technique that estimates the geometry setting of a modular biplanar X-ray CT system. Self-calibration techniques however do not perform well with complicated

object geometries or with objects larger than the field of view. They are time intensive due to the iterative process coupled with CT reconstruction and are not well-adapted for estimating a large number of parameters. With phantom-based calibration procedure, biplanar X-ray CT acquisition geometry can be estimated, after which the calibrated parameters can be used to correct for the geometry misalignment prior to the reconstructions of different CT datasets acquired in the same geometry. First, a method for constructing an optimal calibration phantom using LEGO bricks and spherical steel markers is discussed in detail. Due to difficulties in the extraction of the marker centers from the calibration radiographs, a ResNet-based neural network was trained to facilitate the calibration procedure. Then, a calibration objective function is formulated and a strategic geometry optimization scheme is introduced to find its global minimum, consequently returning an accurate estimate of the geometry parameters. The method is validated with both simulated and real datasets.

Chapter 4 - 2D/3D registration

Chapter 4 presents a 2D/3D registration application for the reconstruction of 3D poses of an animal from X-ray radiographs. A high quality 3D CT image is acquired and employed as the reference model to estimate the 3D pose parameters of an animal from its X-ray fluoroscopy acquisition. The registration procedure is based on geometrical landmarks (points-of-interest) that are detected in the 3D reference model and the 2D X-ray projections. 3D landmarks are extracted from the reference 3D CT model with the shortest coordinate variance scheme to keep the landmarks close to the object's surface but distant from each other. The 2D landmarks in the fluoroscopy images are detected using a ResNet-based neural network architecture. A tool is also implemented to simulate the articulated transformation of joints for generating the training dataset of the deep neural network as well as validation data of the whole procedure. The procedure is evaluated with simulated data for numerical and feasibility studies.

Conclusion

In the conclusion chapter, the main contributions of the thesis with discussions about the limitations as well as potential extensions of the work in the future will be presented.

1

X-RAY IMAGING FUNDAMENTALS

This chapter discusses the basic concept of X-ray generation as the results of interaction between high speed electrons and atoms in an X-ray tube of a common X-ray scanning system. Interaction of X-rays with material, as well as formation of X-ray images and acquisition geometry are presented in Section 1.1. Section 1.2 discusses a computed tomographic reconstruction technique that is commonly used to compute 3D reconstruction of an object from a set of X-ray images. Finally, X-ray geometry and geometry transformation are presented as a general concept. In this chapter, only the basic knowledge of X-ray and X-ray imaging are provided. A more detailed and complete discussion can be found in [**?** **?**].

1.1 X-ray physics

1.1.1 X-ray formation

X-rays, discovered by Wilhelm Conrad Röntgen in 1895, are a form of electromagnetic waves with frequencies ranging between 3×10^{16} and 3×10^{19} Hz [**?** , chapter 4]. From a quantum mechanic point of view, an X-ray is considered as the transmission of X-ray photons throughout space. Each photon carries a fixed amount of energy E, which relates to its frequency f:

$$E = hf \tag{1.1}$$

with $h = 6.62607015 \times 10^{-34}(m^2kgs^{-1})$ the Planck's constant.
X-rays are generated by X-ray sources ([**?** , chapter 2], [**?** , chapter 4]), one of which is shown in Fig. 1.1. Both the cathode filament and anode of the source are made from a high atomic number element, such as tungsten, which can easily release electrons and has a high melting temperature. The cathode filament is heated up at a temperature up to $2\,000°$C to emit thermal electrons. This heating process enables the electrons to acquire enough kinetic energy to escape the binding energy of the electrons to the filament. The electrons are then accelerated in an electromagnetic field inside a vacuum tube before hitting the positively charged anode. When the electrons interact with the anode's matter, the X-ray photons are emitted as the result of several processes taking place close to the anode surface. Most of the energy released during

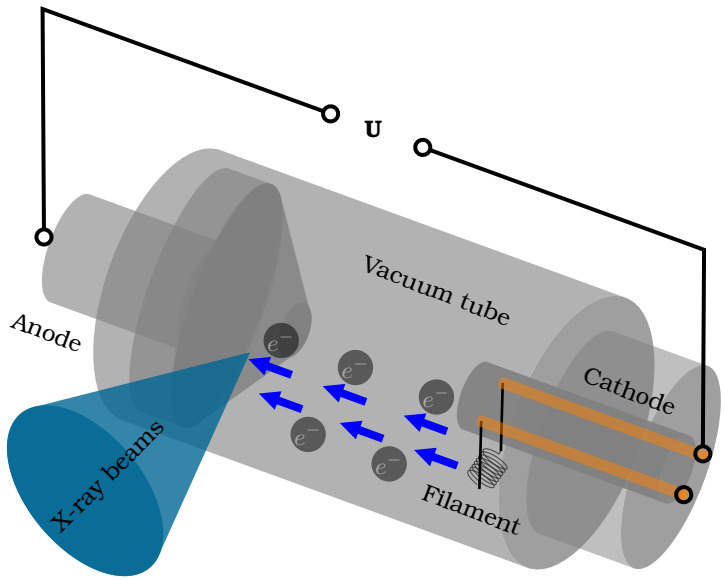


Fig. 1.1 Configuration of a typical X-ray source with the cathode being heated up to emit thermal electrons. These electrons are attracted by the positively charged anode. Interactions between emitted electrons and the anode generate X-ray photons.

the interaction between incoming electrons and the anode's atoms is converted to thermal energy. To prevent damage by heating, a rotating anode disk is used to distribute the heat over the anode.

The properties of the X-ray beam is defined by the current and voltage of the X-ray source. The current, measured in mA, refers to the number of electrons transmitted in a period of time when the X-ray source is turned on. The number of X-ray photons is proportional to the product of the current and exposure time (mAs). Voltage U measures the electric potential difference between the cathode and the anode of the X-ray source. It directly relates to the energy of the electron beam since the electrons are accelerated by this voltage after escaping the filament. In medical diagnosis, the acceleration voltage is between 25kV to 150kV, and 10kV to 300kV is usually used for radiation therapy.

X-rays are generated as the result of two interactions, namely, Bremsstrahlung ("braking radiation" in German) and characteristic radiation. Fig. 1.2 illustrates electron interactions resulting in Bremstrahlung radiation (a, b,c) or characteristic radiation (d). The Bremsstrahlung interaction occurs when a high-speed electron passes near or collides with the nucleus

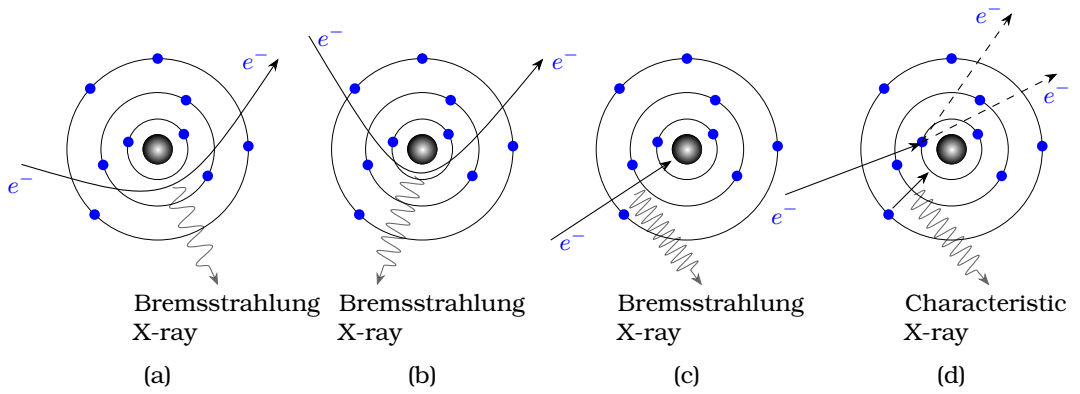


Fig. 1.2 When an electron comes close to or collides with an atom's nucleus, it is deflected (a, b) or absorbed (c), emitting X-ray photons. When an incoming electron interacts with an inner shell electron, and if an atom's electron is released, it creates a vacant position in this inner shell. This position is taken by an electron from an outer shell forming X-ray photons due to energy discrepancy between the two atom shells (d).

of an anode's atom because the positively charged nucleus attracts the electron and slows it down. During this interaction, X-ray radiation is generated with an energy proportional to the energy loss of the electron. The electron deceleration depends on the distance between the incoming electron and the nucleus. If the distance is large, the energy loss is small, and as a result, low energy X-ray will be released (Fig. 1.2a) and vice versa (Fig. 1.2b). When a high-speed electron collides with the nucleus, it looses all its energy, and consequently, a high-energy X-ray photon is released. This process corresponds to the upper spectrum (high frequencies) of the X-rays (Fig. 1.2c). Bremsstrahlung radiation produces most of the X-ray photons of an X-ray source (80%), and it has a continuous spectrum (Fig. 1.3). When a high-speed electron hits another electron at the inner shell of an anode's atom, and an atom electron is ejected from its position, the atom is ionized. When one of the outer shell electrons takes the vacant position, an X-ray photon is emitted with the energy corresponding to the energy discrepancy between the inner and outer shells. It is called characteristic X-ray, and visualized in Fig. 1.2d. During the interaction of the incoming electron beam with the anode's matter, only a fraction of the electrons' kinetic energy is converted to X-rays, and most of it (99%) is converted to thermal energy. Therefore, it

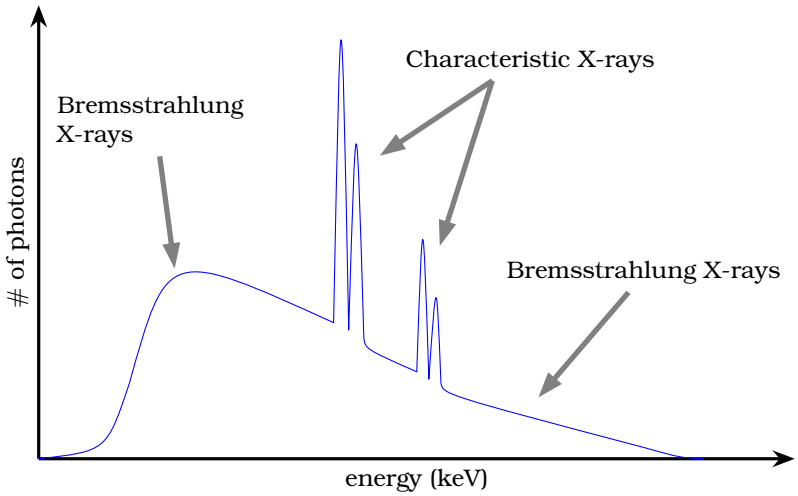


Fig. 1.3 X-ray spectrum with Bremsstrahlung X-rays generated as the results of electrons losing kinetic energy when interacting with anode's atoms (Fig. 1.2a,b,c), and characteristic X-rays emitted as outer shell electrons occupy vacant inner shell positions (Fig. 1.2d).

is important to spread the incoming electrons over the anode surface by using a rotating anode.

1.1.2 X-ray matter interaction

When X-ray photons travel through a target object, they interact with the target atoms ([**?** , chapter 2], [**?** , chapter 4]). More specifically, the X-ray photons interact with both electrons and atom's nuclei when they penetrate the object. During these interactions, they may be absorbed or scattered through the photoelectric effect, Rayleigh and Compton scattering [**?**]. When a high energy X-ray photon collides with a low binding energy electron, it results in the ejection of this electron, and the X-ray photon is completely absorbed. In contrast, when the atomic number of the material is low, the binding energy is small compared to the photon energy. Consequently, the X-ray photons penetrate through the object without losing much energy. The object becomes transparent to the incoming X-ray beam as the detector hardly detects the energy losses. Rayleigh scattering occurs when a photon hits an electron without losing its energy but the direction of the photon is diverged after the collision.

This type of scattering mainly happens when the atomic number of the material is high. Compton scattering occurs when a very high-energy photon collides with an atom electron resulting in ejection of the electron from its orbit. The remaining energy is converted to an X-ray photon that follows a different direction relative to the incoming X-ray photon.

1.1.3 X-ray image formation

When X-ray photons travel through materials, they either interact with the atoms or pass through without any interaction [**?** , chapter 4]. The interactions cause the photons to lose energy, deflect their direction, or even be absorbed completely. The scattered photons that are deflected or lose all their energies and disappear are not detected on the detector. The detector records various reductions of the incoming photons after traveling through the object, namely attenuation. X-rays that do not penetrate the object are measured with greatest intensity on the detector. The X-ray attenuation is proportional to the density of the object, the object compositions, and the materials. The intensity of the detected X-rays on the detector plane is described by the Beer-Lambert law:

$$I = I_0 e^{-\int \mu(l) dl} \tag{1.2}$$

with I_0 the intensity of the incident X-rays emitted from the anode, and I the measured incoming X-ray intensities on the detector. The linear attenuation coefficient of the object material $\mu(l)$ is represented as a function of the object thickness l.
X-ray photons can be detected through an indirect or direct detection process [**? ? ?**]. In an indirect detector, first, the incident X-rays pass a scintillator layer to convert X-ray photons to visible light photons. The visible lights are then magnified in an intensifier layer before being converted into charges, which will then be recorded by the detector elements. With a direct detection process, X-ray photons are converted directly to charges using photoconductors for recording digital X-ray images. The acquired image measures the residual intensities of the X-rays after they are attenuated through the object. This process forms object projections as 3D information of the object are accumulated and projected onto a 2D detector plane. An ideal source setting and detector design could detect,

and convert the same intensity I_0 for every detector element. However, due to the differences in the detector response, unattenuated X-rays could be recorded differently by the different detector elements. In practice, the measure of the attenuation with respect to the thickness of the object is of interest. Therefore, a log form of the projection is used for computed tomography as it represents the object density and thickness.

$$\rho = \int \mu(l)dl = -\ln \frac{I}{I_0} \tag{1.3}$$

where I/I_0 is referred to as *flat-field* correction and Eq. (1.3) is the *log* correction. In practice, I_0 is obtained by measuring the X-ray radiation on the detector plane without the object's presence.
In a fluoroscopy X-ray system, the image intensifiers may cause distortions in the acquired images. Two major distortions are pincushion distortion and radial, sinusoidal distortion [**? ? ?**]. While the pincushion distortion is due to the mapping from a curved input phosphor of the image intensifier onto a flat image plane, the later distortion is caused by interaction between electrons in the image intensifier and the homogeneous electro-magnetic field of the earth and/or inhomogeneous distribution of ferromagnetic components in the sample rotation stage [**?**]. Several techniques that relied on a rectilinear grid of known spacing and a non-magnetic metal with random pattern can be applied to correct for such distortion [**? ? ? ? ?**]. The image distortions must be corrected for prior to any further application or usage of the X-ray radiographs acquired with such systems.

1.2 X-ray projection model and geometry

1.2.1 Projection models

Each 2D X-ray image is saved as a 2D grid of digital image pixels in computer systems. Pixel values measure accumulated attenuations of the X-rays traveling through the object materials [**?**]. Fig. 1.4 demonstrates a 2×2 attenuation grid that is projected onto a 1D projector in different directions. f_i denotes the attenuation coefficient of the object material at

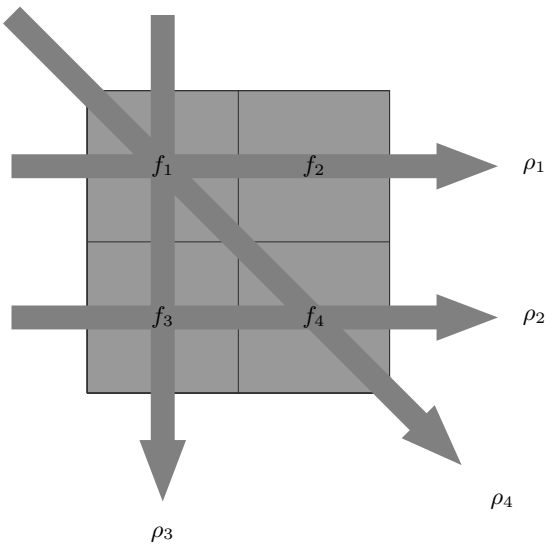


Fig. 1.4 An example of a 2×2 attenuation grid with attenuation coefficient f_i and the measured intensities ρ_j.

object grid index i^{th}.

$$\begin{aligned} f_1 + f_2 &= \rho_1 \\ f_3 + f_4 &= \rho_2 \\ f_1 + f_3 &= \rho_3 \\ f_1 + f_4 &= \rho_4 \end{aligned} \tag{1.4}$$

Eq. (1.4) only describes the accumulated attenuation when the X-ray beam traverses and assumes a binary interaction with a voxel, which is normally not the case in reality. To better handle spatial information of the object as well as the X-ray trajectories, the amount of object voxel areas covered by X-rays during acquisition is quantified as spatial weights/coefficients. The weighting scheme takes the acquisition geometry into account, and hence, better describes an X-ray system in reality. The weights can be computed using different projection models [**?**]. The following discussion considers three models applied to a 2D object, which is discretized into a grid of 2D digital pixels. A 3D object with a voxel grid is an extension of the 2D grid to 3D space.

The line model: The line model, as shown in Fig. 1.5 (a), infers attenuation weights a_{ij} as intersections of X-rays with object pixels, with i, j the pixel position in the 2D grid. The intersections are measured as the

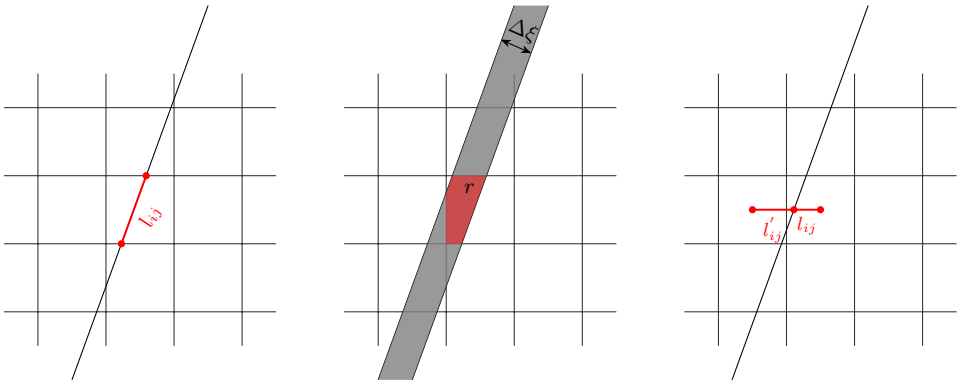


Fig. 1.5 Projection models represent different weighting schemes for X-rays traversing object voxels, i.e. line (a), strip (b), and interpolation (c) models.

lengths of the line sections within the pixels that the rays pass through. The object pixels are considered zero-width pixels in the line model.

$$a_{ij} = l_{ij} \tag{1.5}$$

The strip model: The strip model considers the X-ray beams with a width of $\Delta\xi$. Weight a_{ij} is measured as the portion of pixel area ρ_{ij} covered by ray r when it penetrates the object (Fig. 1.5b).

$$a_{ij} = \frac{r}{\rho_{ij}} \tag{1.6}$$

The interpolation model (Joseph's model): The interpolation model, first, defines the intersection between a ray with the line that connects two neighboring pixel centers. Next, the intensity of a virtual pixel, which is constructed and centered at the intersection, is computed as weighted sum intensities of the two neighboring pixels with line segments from the either center to the intersection (Fig. 1.5c). Now, Eq. (1.4) can be rewritten and generalized as below.

$$\begin{cases} a_{11}f_1 + a_{12}f_2 + \cdots + a_{1n}f_n = \rho_1 \\ a_{21}f_1 + a_{22}f_2 + \cdots + a_{2n}f_n = \rho_2 \\ \cdots \\ a_{m1}f_1 + a_{m2}f_2 + \cdots + a_{mn}f_n = \rho_m \end{cases} \tag{1.7}$$

which can be expressed under a matrix form:

$$Ax = \rho \tag{1.8}$$

with

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \tag{1.9}$$

$$x = \begin{bmatrix} f_1 & f_2 & \dots & f_n \end{bmatrix}^T \tag{1.10}$$

$$\rho = \begin{bmatrix} \rho_1 & \rho_2 & \dots & \rho_m \end{bmatrix}^T \tag{1.11}$$

with A the system matrix, x the unknown object discretized grid of n pixels, and ρ the acquired X-ray projection data of m detector elements.

1.2.2 Projection geometry

Since the weighting schemes depend on the relative orientation and position of the object grid in the acquisition geometry, it is crucial to correctly describe the system geometry prior to the CT reconstruction. A fan-beam geometry with the line projection model is demonstrated in Fig. 1.6, with the original detector position colored in black, and ρ_i a sample pixel. When a correct system geometry is taken into account, weight l_{ij} is associated with pixel ρ_i. If geometry misalignment occurs, for example, the detector is rotated and translated relative to its desired position, the detector pixel ρ_i and its new associated coefficient l_{mn} need to be correctly identified. The weights will be mapped to wrong pixel values if the displacements of the detector are not accounted for. The CT reconstruction involves solving the system of linear equations $Ax = \rho$ (Eq. (1.8)). With the incorrect equation coefficients, it is not possible to derive a correct solution. Various iterative reconstruction techniques are implemented around the fundamental idea of finding an optimal solution for this system of linear equations [**?** **?** **?**]. However, the misalignment induces reconstruction artifacts in the form of blurring in the reconstructed CT images [**?**]. Therefore, it is crucial to measure and estimate the system geometry in advance to reduce geometry misalign-

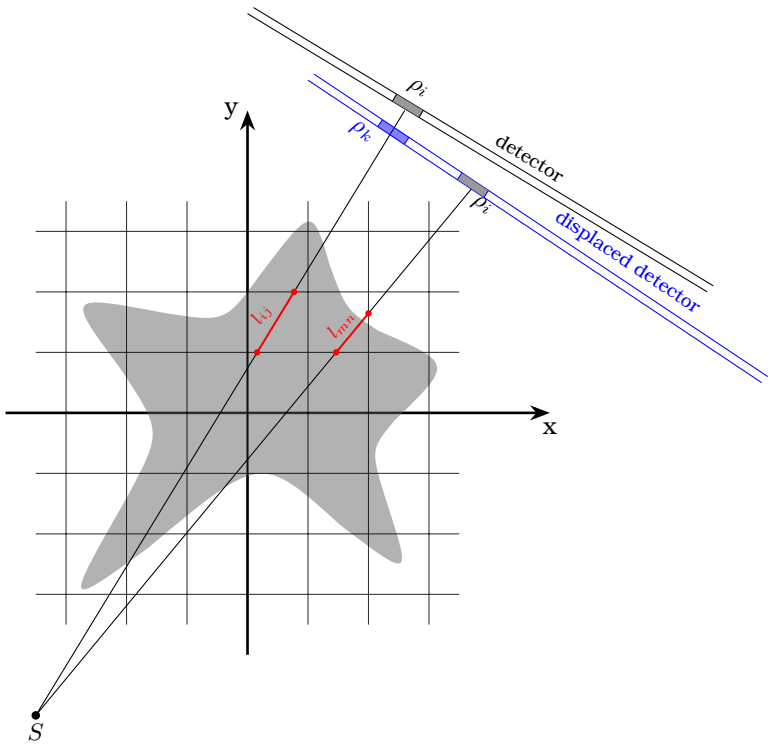


Fig. 1.6 Relation between projection geometry and system matrix coefficients. In an ideal, known geometry (dark plotted detector), system matrix coefficient l_{ij} is associated with projection pixel ρ_i. In a misaligned geometry with detector displacement, l_{ij} relates to pixel ρ_k while ρ_i is shifted to be mapped to l_{mn}.

ment artifacts in a CT reconstruction. Apart from artifact reduction in CT images, a calibrated geometry also provides valuable prior knowledge when the acquired X-ray radiographs are used in a 2D/3D registration application.

1.3 X-ray cone-beam acquisition geometry

Fig. 1.7 A conceptualization of a cone-beam X-ray acquisition system is shown in Fig. 1.7 with the X-ray cone-beam represented as a gray shade. The cone-beam geometry is usually simplified to a form of four rays connecting source point and four detector corners (dashed lines between S and the detector corners in Fig. 1.7). During an X-ray acquisition, 3D object information is projected onto a 2D plane. The process is modelled by the Beer-Lambert law (Eq. (1.2)), which relates to integration of the X-

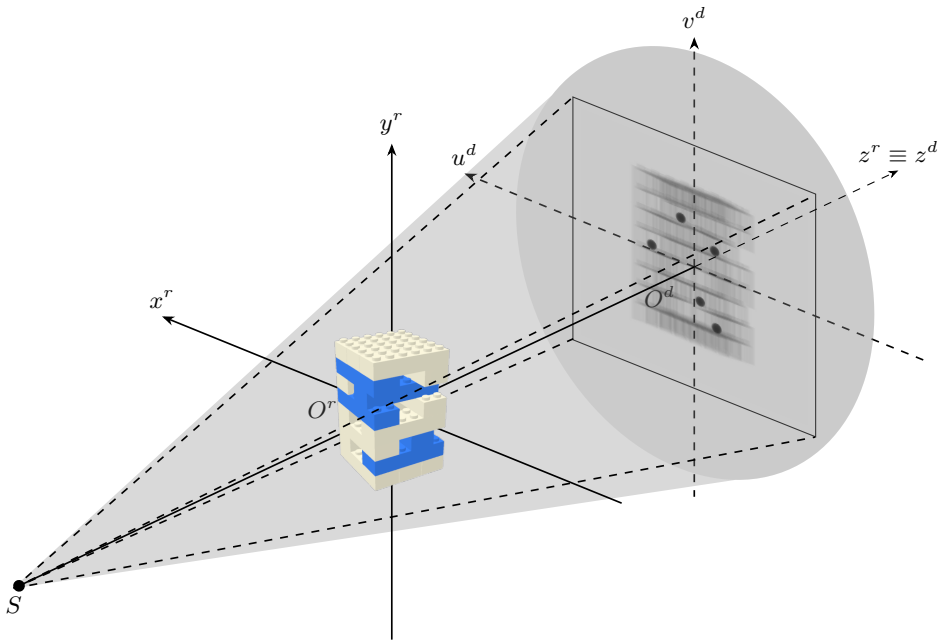


Fig. 1.7 An example of an ideal circular X-ray cone-beam geometry. In this setting, the perpendicular projection of the source S coincides with the detector center O^d. Both distances from the source S to the center of rotation O^r and the detector center O^d can be estimated correctly.

ray attenuation over the object thickness along the penetration trajectory. Multiple X-ray images from different directions relative to the object are acquired to recover the 3D information of the object. A typical circular cone-beam X-ray system is shown in Fig. 1.8, with an object mounting stage and a pair of X-ray source/detector. The circular geometry can be obtained by either stationing the object stage and rotating the source and the detector simultaneously or vice versa.

1.3.1 Geometric transformations

Geometry information is crucial for a misalignment artifact-free CT reconstruction. Due to possible displacements of the X-ray source, the detector, or the rotation axis, it is required to correctly describe the acquisition geometry in the CT reconstruction. Generally, this can be done by manipulating the system geometry based on the displacement information [**?** **?**], [**?** , chapter 2]. The following discussion provides background knowledge about geometry and geometry transformation which will be used

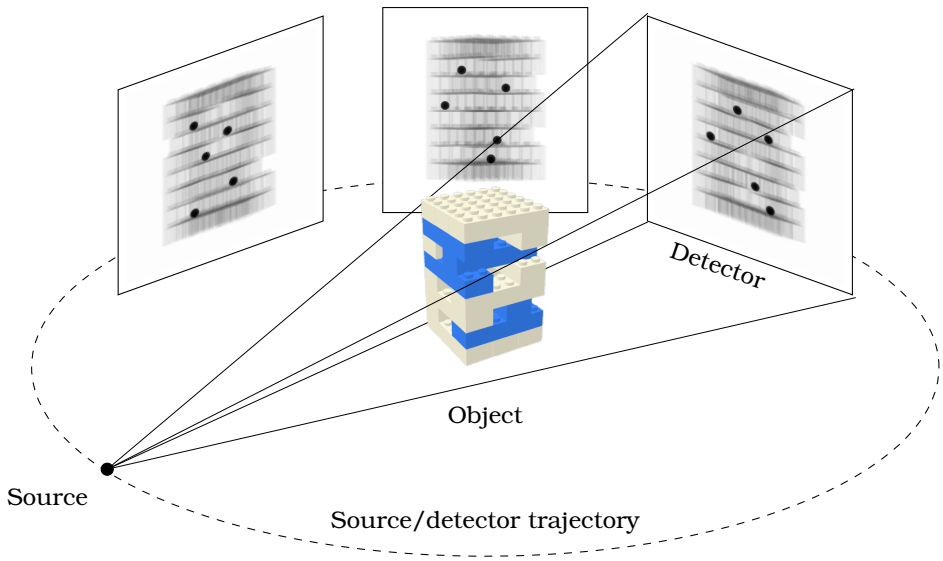


Fig. 1.8 X-ray cone-beam acquisition geometry with different acquisition angles. A dataset of X-ray radiographs is acquired at different projection angles.

throughout the calibration, and 2D/3D registration application. A point in 3D space is represented by its coordinates $(x_p, y_p, z_p)^T$ with respect to a reference coordinate system $Oxyz$. In this thesis, a right-hand rule is applied to define rotation direction in both geometry calibration and the 2D/3D registration applications. The direction of the rotation is defined as follows. The right-hand thumb points towards the positive direction of the z-axis, and the curl of the other fingers indicates the rotation from the x-axis to the y-axis (Fig. 1.9). The rotation around the z-axis following this direction is counter-clockwise and has a positive sign if we look at the coordinate system from above the z-axis. A similar rule is applied to the other axes to define the positive/negative (or clockwise/counter-clockwise) direction of the corresponding rotation angles.

A 3D object can be stored as a voxelized volume with a reference coordinate system in the computer systems. Orientation and position of the object are defined with respect to the reference coordinate system in virtual 3D space. By manipulating the reference coordinate system, the object perspective changes accordingly. In an X-ray cone-beam system, a coordinate system attached to the rotation center is usually chosen as the reference coordinate system. Therefore, it is possible to perform a rigid transformation of an object in an X-ray acquisition geometry by

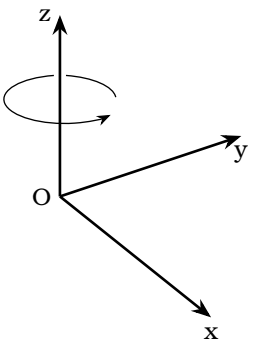


Fig. 1.9 Cartesian coordinate system with the right-hand rule.

simply manipulating the rotation center coordinate system. A non-rigid transformation requires computation of displaced voxel positions and voxel intensity after applying a transformation model to the voxel coordinates. Affine transformations include scale, shear, translation, and rotation [**?** **?**], [**?** , chapter 4], [**?** , chapter 3]. However, only translation and rotation are sufficient to model geometry misalignment of an X-ray acquisition system as scaling factor can be achieved by adjusting the object, source, or detector position with respect to the reference coordinate system (Section 3). For a 2D/3D registration of an animal object, which will be discussed in Section 4, a more complex transformation will be modeled to replicate a limb motion. The basic recipes remain the same as the translation and the rotation of an object are defined with respect to a reference coordinate system.

1.3.1.1 Translation

A translation involves movements of a given point $(x_p, y_p, z_p)^T$ along the three axes (x, y, z) of the 3D coordinate system $Oxyz$ with respective distances of $\{\Delta x, \Delta y, \Delta z\}$. The translated point coordinates $(x_p^t, y_p^t, z_p^t)^T$ are computed by Eq. (1.12).

$$\begin{cases} x_p^t = x_p + \Delta x \\ y_p^t = y_p + \Delta y \\ z_p^t = z_p + \Delta z \end{cases} \tag{1.12}$$

1.3.1.2 Rotation

Rotations of the point about the three axes x, y, and z with respective angles $\{\theta, \phi, \eta\}$ are modeled under the matrix form as presented below.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} \tag{1.13}$$

$$R_y(\phi) = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \tag{1.14}$$

$$R_z(\eta) = \begin{bmatrix} cos(\eta) & -sin(\eta) & 0 \\ sin(\eta) & cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.15}$$

An arbitrary rigid rotation of a point or object in the 3D space is obtained by matrix multiplication, for example, $R = R_x(\theta)R_y(\phi)R_z(\eta)$.

1.3.1.3 Homogeneous coordinate system

A homogeneous coordinate system is usually used to represent translation under a matrix form. It is obtained by adding an extra coordinate to the 3D point representation [**?**]. Point p is now represented in the homogeneous coordinate system as $p = (x_p, y_p, z_p, 1)$. The translation of the point p can be rewritten as shown in Eq. (1.16).

$$\begin{bmatrix} x_p^t \\ y_p^t \\ z_p^t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \tag{1.16}$$

with translation matrix T defined as:

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.17}$$

And the rotations matrix in the homogeneous coordinate system are reformulated in Eq. (1.18), Eq. (1.19), and Eq. (1.20) for rotations around (x, y, z) axes, respectively.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.18}$$

$$R_y(\phi) = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\phi) & 0 & cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.19}$$

$$R_z(\eta) = \begin{bmatrix} cos(\eta) & -sin(\eta) & 0 & 0 \\ sin(\eta) & cos(\eta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.20}$$

The translation and rotation can now be combined by a matrix multiplication to have a transformation matrix in the homogeneous coordinate system. For example:

$$TR_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & cos(\theta) & -sin(\theta) & \Delta y \\ 0 & sin(\theta) & cos(\theta) & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.21}$$

1.3.1.4 Forward transformation

Forward transformation refers to a direct computation of the voxel coordinate displacements with respect to the transformation parameters ([**?**], [**?** , chapter 3]). An example of a 2D forward transformation applied to a 2D pixel grid is shown in Fig. 1.10. The new coordinates of the object pixels (x, y) are obtained by simply multiplying the original coordinates with the transformation matrix. As a transformation matrix almost always contains float coefficients, transformed pixel coordinates are likely non-integer numbers (x^t, y^t). However, due to the discretiza-

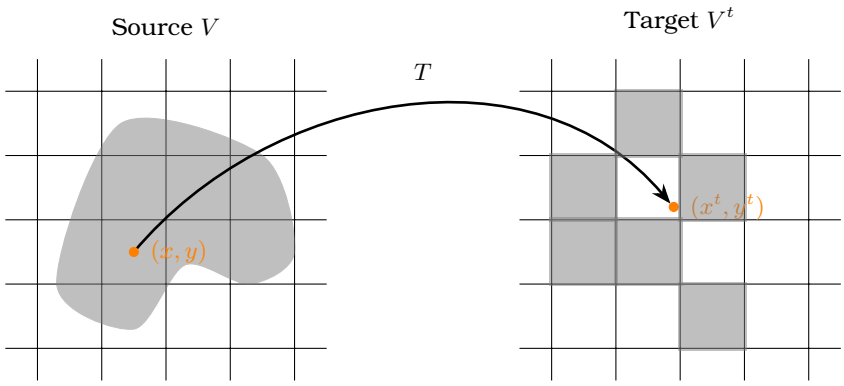


Fig. 1.10 Visualization of a 2D forward transformation.

tion of the digital image pixels stored in the computer systems, all the pixel coordinates must be represented by integer numbers. Therefore, multiple pixels from the original image are likely mapped to the same pixel in the target transformed image. There are also pixels in the target image that are not linked to any pixels in the source image, which create discontinuities in the target image as shown in Fig. 1.10 (blank pixels in between surrounding gray pixels). The same scenario is applied to the 3D voxels and volumetric image forward transformation.

1.3.1.5 Inverse transformation

Inverse transformation is a commonly used technique to tackle discontinuities in the transformed image [**?**], [**?** , chapter 3]. In general, an inverse transformation maps each target voxel with a voxel in the source volume and therefore eliminate discontinuities. Visualization of inverse transformation is shown in Fig. 1.11. Like forward transformation, an inverse transformation matrix contains non-integer coefficients. Each integer pixel (x^t, y^t) in the target volume is likely to be mapped to a single non-integer source pixel (x, y). Since the non-integer pixel does not hold an intensity, an interpolation is followed to derive the corresponding value for the target pixel. Extended to 3D space, it is possible to obtain inverses of translation and rotations by applying opposites of the translations Eq.

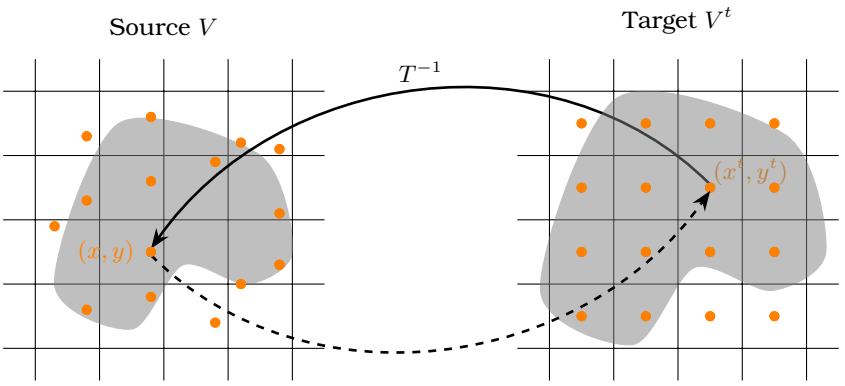


Fig. 1.11 Visualization of a 2D inverse transformation.

(1.22) and the rotations Eq. (1.23).

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -\Delta x \\ 0 & 1 & 0 & -\Delta y \\ 0 & 0 & 1 & -\Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.22}$$

$$R_x^{-1}(\theta) = R_x(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(-\theta) & -sin(-\theta) & 0 \\ 0 & sin(-\theta) & cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) & 0 \\ 0 & -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.23}$$

1.3.2 Interpolation

Inverse transformation usually faces non-integer voxel mapping, i.e., a mapped source voxel is often non-integer, and therefore requires using neighboring voxel intensities to derive its expected grayscale. This process is referred to as interpolation, in which intermediate data points are obtained based on given existing neighbors [**?** **?**]. Trilinear and tricubic spline interpolation are among the common techniques used to infer the intensity value of a float voxel based on its neighboring voxels in a regular Cartesian grid.

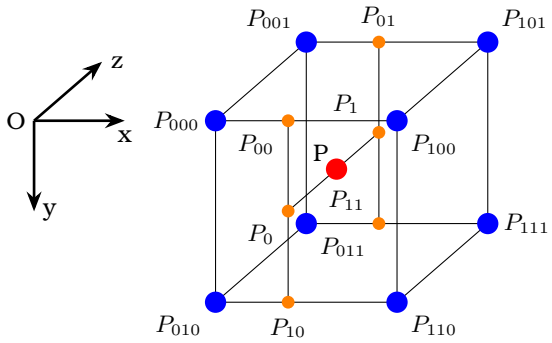


Fig. 1.12 Neighboring points (blue) in a regular Cartesian grid that are used for the interpolation of an intermediate point at the middle (red).

Trilinear interpolation algorithm relies on a grid of eight neighboring voxels (blue) to compute intensity of an unknown voxel P (red) (Fig. 1.12). A straight line is fitted to a pair of intermediate points along each dimension to compute intermediate anchor points (orange). These intermediate anchors are then used to calculate the value of the target point (blue).

Tricubic spline is a form of interpolation where intermediate points are computed by fitting a polynomial of a certain degree to its neighboring points. The value of the unknown point is obtained by evaluating the polynomial at the unknown point's coordinates. In this thesis, a polynomial of degree 3 will be used for the interpolation, which is usually regarded as cubic spline interpolation [**?**]. Tricubic spline interpolation is the 3D extension for the cubic spline method that is applied to 3D data. A 3×3-neighboring voxels are used for inferring intermediate data points. A cubic spline is fitted to three neighboring voxels along each volume dimension to derive an anchor point for the later computation. In other words, along each dimension, two polynomials $y = g_i(x), i = 1, 2$ are interpolated for each of a pair of control points (x_{i-1}, y_{i-1}) and $(x_i, y_i), i = 1, 2$ Fig. 1.13.

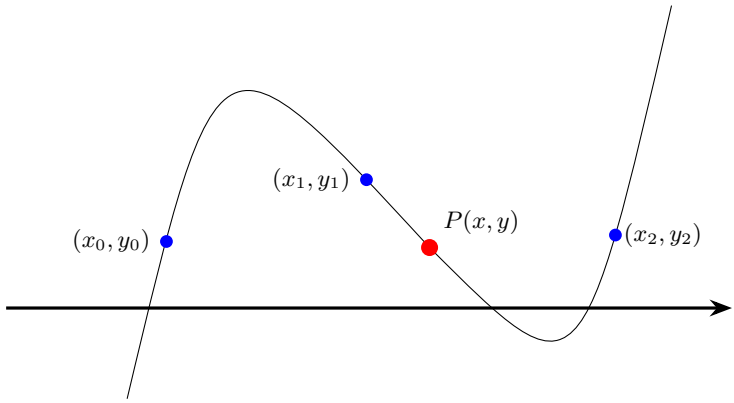


Fig. 1.13 A visualization of a 3^{rd}-order polynomial (cubic spline) interpolation with three control points (blue) for an unknown point (red).

2

DEEP NEURAL NETWORKS

This chapter discusses the fundamentals and applications of deep learning. Section 2.1 focuses on the basic concept and architectures of machine learning and deep learning models. Next, computational operators and techniques in convolution neural networks will be presented in Section 2.2. Finally, Section 2.3 provides an overview of a deep residual network (ResNet), which will be applied in chapter 3, 4. A more detailed and complete discussion can be found in [**? ?**].

2.1 Fundamentals of deep learning

2.1.1 Machine learning

Machine learning describes the computation techniques that extract and reveal meaningful information from input data [**?** , chapter 1]. Data refers to anything that can be recorded, measured, and stored in the form of raw numbers, pictures, sounds, language characters, etc. The meaningful information is any information that can be of interest in the study, or that can serve a particular purpose. Typically, the meaningful information is defined prior to designing a machine learning algorithm so that the algorithm is built and customized for desired details from the data.

The data contains data points that map each measurement (temperature, price, wind speed, etc.) to a specific value. Each data point is referred to as *sample* with the corresponding measurement (*feature*) and mapped value (*label/ground truth*). A machine learning algorithm is usually described as a mathematical model under the form of a *learning function* with learnable coefficients. The model represents the expected behaviors of input features, which are then compared to recorded or measured (ground truth) values. A *cost/loss function* is formulated to measure difference between the expected values and the provided labels.

A *training* refers to a process in which the learning function evaluates every data point and update the mathematical model coefficients so that the expected data behavior (*prediction*) would be close to the measured data. An example of a machine learning process is shown in Fig. 2.1. First, the learning function is computed at each sample feature data to get the corresponding prediction. The predicted value is then compared to the respective label, and the loss is derived. After that, coefficients

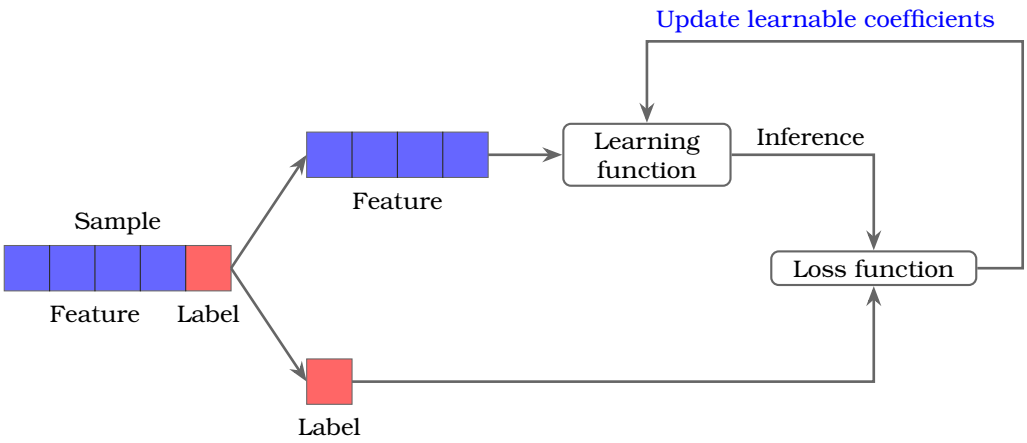


Fig. 2.1 An example of a machine learning system with input containing a feature vector and a ground-truth label. The learning function computes the prediction based on the input feature and compares it to the given label. Learning function coefficients are then updated based on the loss between the prediction and label.

of the learning function are updated accordingly so that the predictions evolve towards the true labels. The dataset that is used for training is often referred to as *training data*. In a machine learning application, there are often the values that need to be predefined, such as the rate to update the learning function coefficients or the number of times that the data evaluation will be performed. These types of values are regarded as *hyperparameters*, and preset for the training. The coefficients of a machine learning model that are updated and finetuned during the learning process are referred to as *parameters*, and are adjusted by following a particular scheme.

There are two major categories of machine learning techniques, supervised and unsupervised learning. *Supervised learning techniques* include classification and regression. Classification techniques focus on training a model that can recognize and distinguish different types of objects in the training data and is capable of inferring an object from a sample that is not included in the training. Regression techniques learn a general "trend" of the training data to predict possible "future" data points. Regression usually involves fitting a linear (straight line) or non-linear (curved) mathematical model to the training data. A prediction results from evaluating the fitted model at the new data points. *Unsupervised*

learning techniques include clustering, noise reduction, and dimension reduction. Clustering involves sorting data into groups of similar items or characteristics. Noise reduction refers to the techniques that suppress the unwanted signal from original data to enhance interested characteristics or meaningful information in the data. Features of the data are often highly-dimensional. However, not all features provide useful information, as some could be redundant. Dimension reduction techniques reduce the number of features, or discard unwanted features to simplify data, enhance the features that convey meaningful information, and increase the quality of output results.

2.1.2 Deep learning

Deep learning is a rapidly growing subset of machine learning in recent years. It has been applied to various problems in computer vision, natural language processing, or data analysis due to its robustness, simplicity in implementation with end-to-end training capabilities [**?** , chapter 1], [**?** , chapter 20]. Instead of predesigning a mathematical model to describe data behavior, deep learning uses specialized computation layers stacked together to create a "deep" computational architecture to extract meaningful information from data [**?** , chapter 20]. The word "deep" implies the multi-layer architecture of this learning model. The specialized computation layers are usually the simple computational operators that encode input data into abstract representations. A deep learning model is built by stacking the simple operators in a particular mechanism to perform complex computation and representation of input data. An example of a deep learning model with four layers (an input, two hidden, and an output layer) is shown in Fig. 2.2. Each of the two hidden layers contains four computation elements called *artificial neurons* (or *perceptrons*), and an output layer has three neurons. The neurons and layers can be constructed in particular methods, creating various deep neural network architectures to solve specific problems. There are several major deep neural network architectures [**?**], including:

- Unsupervised networks:
  - Autoencoders learn and compress input x to an abstract form $f(x)$ and then reconstruct the input into another version (usu-

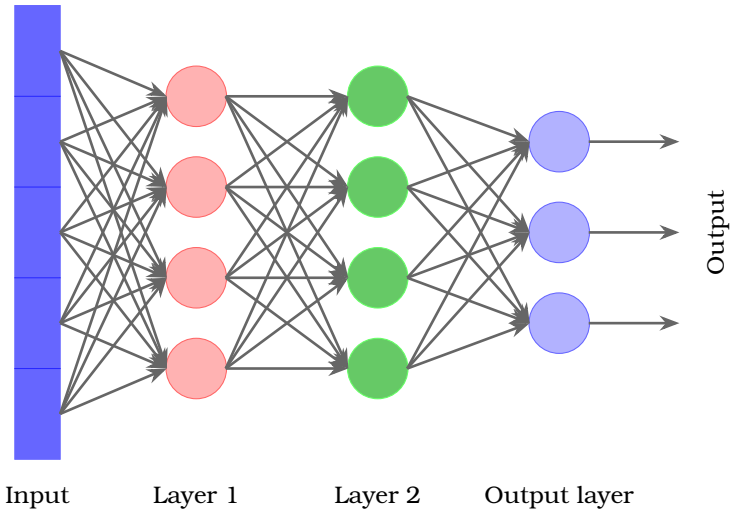


Fig. 2.2 An example of a deep learning system with an input, two hidden, and an output layers.

ally a derived version) of it as the output $g(f(x))$. An ideal autoencoder would be the network that produces $g(f(x)) = x$ but would not describe the training data too closely or exactly. Autoencoders are usually applied in denoising and data dimension reduction ([**?** , chapter 14], [**?** , chapter 25]).

- Generative Adversarial Networks (GANs) are trained based on a clever strategy that two deep networks compete with each other ([**?**], [**?** , chapter 27]). One network learns to create new samples that do not exist in the training samples but are very similar to them so that the other network cannot identify whether the newly generated samples belong to the training data or not. The network that is responsible for generating the synthetic samples is called *generator*, while the other network is referred to as *discriminator*.

- Convolutional Neural Networks (CNNs) refer to a group of deep neural networks that are constructed based on convolution operators [**?** , chapter 9]. A CNN usually contains multiple convolution layers that perform feature extractions on the input data. CNNs are widely used in image understanding applications and will be discussed in detail in Section 2.2.

- Recurrent Neural Networks (RNNs) refer to the deep neural models that are designed specifically to process sequence data $x^{(t)}$, with t ranging between 0 and τ. RNNs generally try to learn from received sequence data to predict future data ([**?** , chapter 10], [**?** , chapter 22]). The sequence data is the type of data that carries information throughout time, i.e., current data information depends on previous or future data, such as an audio sequence, human language, any kind of data recorded over time, etc. During training, RNNs compute a new state $h^{(t)}$ of the data based on the previous state $h^{(t-1)}$ and the new incoming input sequence $x^{(t)}$ as $h^{(t)} = f(h^{(t-1)}, x^{(t)})$. The idea behind it is that predicting future information does not require storing the sequence data $x^{(t+1)}$ from the beginning of time. RNNs only generate a fixed and meaningful length of data for the future state vector $h^{(t)}$, which will then be compared to ground-truth training state y and update the training model parameters.

2.1.3 Training a deep learning model

A deep learning model is designed and built based on the result or information that is expected to to be extracted/learned from the data. The training process is initialized with default or random values of the network parameters. Training is an iterative process in which the training data is evaluated multiple times in order to adjust the parameters accordingly. This repetition is called *epoch*. Depending on the size of the training datasets as well as the model inference results, the number of training epochs ranges from several hundred to thousands. During the training, we also need to evaluate whether the model is performing correctly on a separate, independent dataset (*validating dataset*) apart from the training data through a *validation*. When the training completes, the network parameters are tuned towards the expected outputs. The trained model will be tested or evaluated on one or different study datasets, which must not contain samples in the training or validation data. This process is regarded as *testing* and the datasets is referred to as *test dataset*. It is recommended to split the original dataset into 60%, 20%, and 20% [**?** , chapter 8] for training, validation, and test set. In practice, the ratios might differ as upto 88% of the data can be put into traning, and only 4% and 8% are used for validation and test set, respectively [**?**]. The

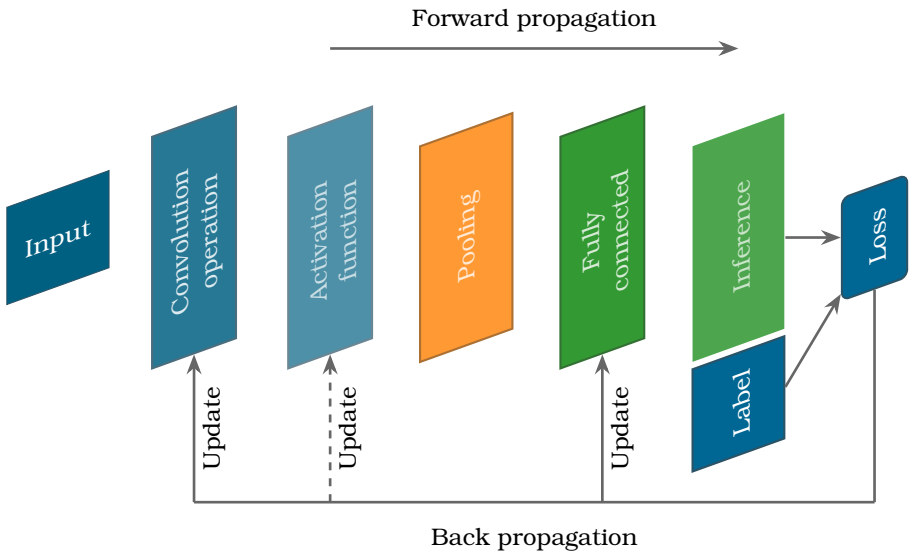


Fig. 2.3 An example of a convolutional neural network with typical building blocks.

optimum split of the test, validation, and train set depends upon factors such as the use case, the model architecture, the number of samples in the orginal data set, or the dimension of the data [**?**].

2.2 Convolutional neural networks

2.2.1 Foundations of the convolutional neural networks

A convolutional neural network is constructed from different building blocks, including convolution, activation, pooling, and fully connected layers [**?** **?**]. The building blocks can be combined and arranged in many different ways to create various depths and architectures for specific training purposes and data. CNNs are usually applied to process grid-like data, such as images, since the convolutional operator is robust and efficient in extracting useful information from this type of data. An example of a CNN with the typical building blocks is shown in Fig. 2.3. A CNN model can be represented as a function f of input x and learnable parameters ω Eq. (2.1).

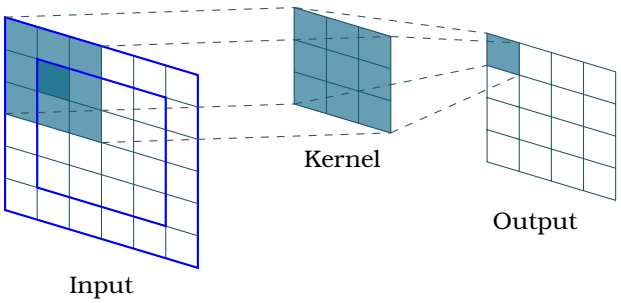


Fig. 2.4 Convolution of a 3×3-kernel with a 4×4 input array padded by 1 in both horizontal and vertical direction.

$$\mathcal{F} = f\,(x, \boldsymbol{\omega}) \tag{2.1}$$

During a training process, predictions/inferences $\mathcal{F}$ are computed with respect to input data via a *forward-propagation* $f(x, \boldsymbol{\omega})$. The training process minimizes the differences between the predictions and the truth values y, provided as training labels. The learnable parameters are updated via a *back-propagation* process that is based on a gradient descent algorithm with a certain *learning rate*. During such process, the gradient of the training loss with respect to individual parameters is computed, and the network parameters $\boldsymbol{\omega}$ are updated using (Eq. (2.2)). A more detailed discussion will be presented in Section 2.2.2.

$$\boldsymbol{\omega} = \boldsymbol{\omega} - \frac{\partial f}{\partial \boldsymbol{\omega}} \tag{2.2}$$

Convolution operation perform linear convolutional operations on input data. Convolution kernel or filter mask refers to a small matrix used in a convolution operator . The convolutional process computes the sum of the elementwise product of image pixel intensities with the kernel coefficients (see Fig. 2.4) after the filter is rotated by $180°$ [**?** , chapter 3]. The kernel sizes are predefined as hyperparameters while the kernel coefficients are initialized randomly or assigned to scalar values at the beginning of the training. The kernel coefficients are updated during the training based on the back-propagation using different optimization algorithms (Section 2.2.2).

With any kernel size k that is larger than 1, it is not possible to compute the convolution for the outermost pixels of the image grid as kernel

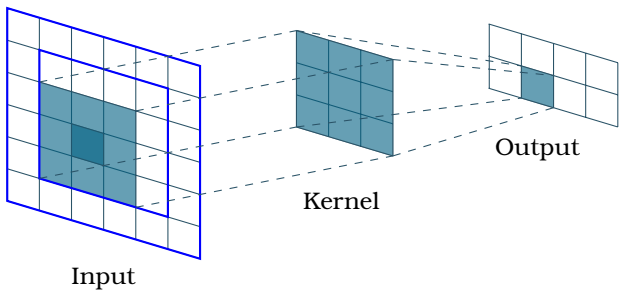


Fig. 2.5 Convolution of a 3×3-kernel with a 4×4 input array padded by 1, stride 1 and 2 in horizontal and vertical directions, respectively.

centers cannot be placed at these pixels. As a result, the convolution image has a smaller size than the input. Padding is the technique that lines extra pixels to the outer borders of an image enabling computation of the outermost pixels' convolution hence increasing the effective size of the convolution result. Padded pixels are usually set to zero, and the padding size is set prior to the training. An example of a padding of 1 with a convolution of 3×3-kernel size is shown in Fig. 2.5 with padded pixels highlighted in between the two blue rectangles.

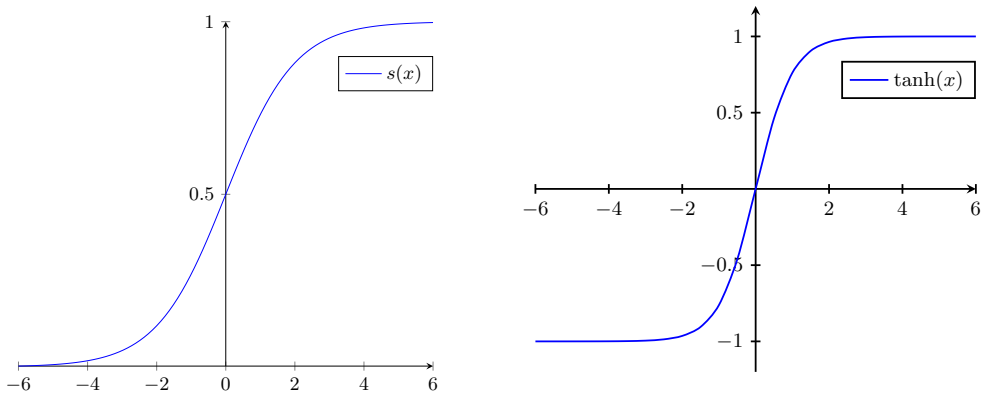
When computing the convolution of a kernel with an image, the kernel center is virtually placed at different pixel positions across the input grid. Distance between the two consecutive kernel center positions is referred to as *stride*. In other words, the stride is the number of pixels skipped in horizontal or vertical directions when performing the convolution operator over an image [**?**]. The stride step is another hyperparameter that is chosen before the training.

Activation functions non-linearize the results from convolution operation. Previously, the activation functions such as sigmoid or tangent hyperbolic (tanh) functions Fig. 2.6 were widely used. The Eq. (2.1) can be rewritten as follows:

$$\mathcal{F} = f\left(\sigma\left(x, \boldsymbol{\omega}\right)\right) \tag{2.3}$$

with $\sigma\left(x, \boldsymbol{\omega}\right)$ an activation function. The parameters $\boldsymbol{\omega}$ are updated based on the gradient of the loss function with respect to the parameters using the chain rule:

$$\frac{\partial f}{\partial \boldsymbol{\omega}} = \frac{\partial f}{\partial \sigma} \frac{\partial \sigma}{\partial \boldsymbol{\omega}} \tag{2.4}$$



(a) Sigmoid.

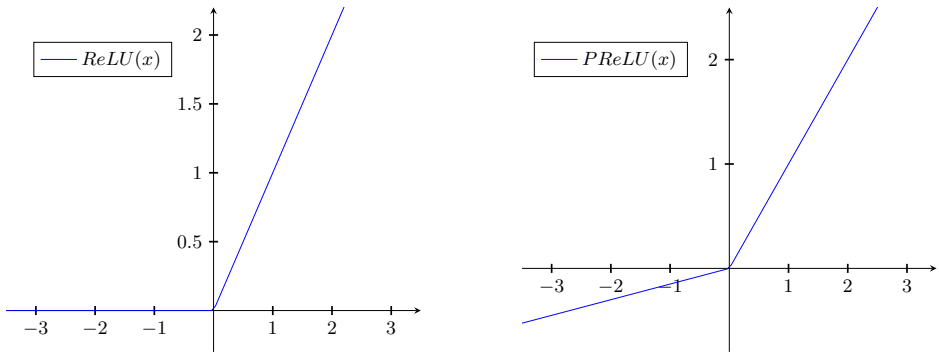
(b) Tangent hyperbolic (tanh).

Fig. 2.6 Activation functions.

As a result, if sigmoid (Fig. 2.6a) or tanh (Fig. 2.6b) are applied, the partial derivative of the activation function (Eq. (2.4)) approaches zero when the function saturates. The parameters are therefore not updated properly to make the training progress. Recent CNNs exploit a rectified linear unit (ReLU) as the activation function since it suppresses gradient vanishing and increases the learning speed of a deep neural model. ReLU simply finds the value $f(x) = \max(x, 0)$ (Fig. 2.7). A parametric ReLU (PReLU) [**?**] with learnable parameters α was also introduced to increase convergence rate with $f(x) = \max(x, 0) + \alpha \min(0, x)$. Leaky ReLU is a variance of PReLU where α is fixed as a hyperparameter (Fig. 2.7). At the last layer of a neural network, typical activation functions are usually chosen to return expected outputs. For example, when dealing with classification problems, a dense layer is often placed at the end of the networks with the same number of outputs as the number of categories. For this type of network, a *softmax* is often followed to simply scale the outputs so that they add up to 1.

Pooling layer provides a simple down sampling operation over convolution outputs. Max pooling is a typical pooling operator that extracts maximum values of feature patches. Global average pooling performs extreme down sampling as pixel values of a whole image/feature map are averaged to get 1×1 matrix. There are no learnable parameters in this pooling layer, however patch sizes are predefined for the training.

Fully connected (FC) layer is the neuron that receives an input from every neuron of the previous layers. FC is also called a dense layer. A



(a) ReLU **(b)** PReLU

Fig. 2.7 Rectified linear unit (ReLU) (a) and PReLU with parameter α (b).

fully connected network or multi-layer perceptron is constructed from a stack of only dense layers.

Loss functions (cost functions) measure differences between network inferences (expected values) and the given labels (ground truths). The loss function is defined prior to the training based on the deep learning problems, expected outputs, etc.. Cross-entropy or softmax loss function is usually employed in multi-class classification problems. First, a dense layer with a softmax activation function is implemented at the output layer to obtain expected probability $p \in [0, 1]$. Then cross-entropy loss is computed using Eq. (2.5).

$$\mathcal{L}(p, y) = -\sum_{i} y_i \log(p_i) \tag{2.5}$$

where y_i, p_i the true labels and predicted values from the network outputs, respectively, and $i \in [1, N]$, N the number of network outputs. Euclidean loss function or mean-squared error is a typical loss function used in regression problems. Computation of the Euclidean loss is as follows:

$$\mathcal{L}(p, y) = \frac{1}{2N} \sum_{i=1}^{N} (p_i - y_i)^2 \tag{2.6}$$

The L1-norm can also be employed as a loss function which measures absolution different between predictions and ground-truth values Eq.

(2.7).

$$\mathcal{L}(p, y) = \sum_{i=1}^{N} |p_i - y_i| \tag{2.7}$$

Regularization is applied to suppress overfitting in training a CNN model [**?** , chapter 20]. Overfitting occurs when the trained model has a low error when evaluating on training data but results in a high loss on other validation/testing datasets. This phenomenon is also regarded as high variance, and is often caused by a network architecture that is too deep. There are several common techniques to tackle this issue:

- Dropout is the technique in which random neurons of the network are dropped during each training epoch. This technique allows feature learning to be distributed across the whole network. Dropout also simplifies the network architecture, hence preventing overfitting to the training data.
- Drop-weights is another method involving adding a regularization factor into the formation of the loss function. By adjusting the regulation factor, the weights are zeroed-out in hidden units, and the network architecture is simplified.
- Data augmentation effectively increases the amount of data for training by employing various techniques. For example, geometric transformation or data simulation can be used as an image data augmentation tool.
- Batch normalization normalizes intermediate outputs of hidden layers allowing faster convergence, avoiding gradient vanishing, overfitting, etc.

2.2.2 Optimization algorithms

During training, the learnable parameters of a deep neural network are updated to minimize the cost function through a learning algorithm (optimizer). Therefore, it is vital to choose a suitable optimizer for a converged training [**?**]. The optimizers use partial derivatives of the loss function with respect to the learnable parameters as the backbone for the parameter updates.

Gradient descent algorithm

The parameters are updated throughout the training by a gradient descent mechanism. A learning rate is defined as the step size of parameter updates, while the training epoch represents the number of repetitions in which the network parameters are updated. The gradient descent algorithm is as follows. First, the 1^{st}-order derivative of the lost function $\mathcal{L}$ is computed with respect to individual parameters. Then the parameters are updated using Eq. (2.8).

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_{e-1} - \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}} \tag{2.8}$$

with α the learning rate, $\boldsymbol{\omega}$ the learnable parameters, and e the current epoch. Training usually performs on a large amount of data, i.e. batch. Often, due to memory limitations, training data is divided into multiple smaller *mini-batches*. Different parameter update strategies can be applied depending on the types of data and network performance.

Batch gradient descent is the technique where the loss function is evaluated on the whole training dataset at once. Subsequently, the parameters are updated in a stable manner as the gradient is computed using the whole dataset. This method can only be applied to small training datasets that fit the memory of the training computer system.

Stochastic gradient descent, in contrast, computes the gradient for each sample in the training data. The parameters are therefore updated sample-wise. Although this technique allows faster gradient computation than batch gradient descent, it could create an unstable and noisy convergence.

Mini-batch gradient descent exploits the advantages of both batch and stochastic gradient descent techniques as the training dataset is divided into multiple smaller mini-batches. The parameters are updated based on the gradients computed with respect to these mini-batches. Therefore the technique has a faster computation than the batch gradient descent and more stably converges than the stochastic gradient descent method. For these reasons, the mini-batch gradient descent has been widely employed in training a deep neural network, especially with a large amount of data.

Simply applying gradient descent in a training could experience a very slow convergence or a fast decay of the learning rate. Different optimizers are proposed to improve the speed and the occurrence of training conver-

gences [**?**].

Gradient descent with momentum utilizes the gradient decent technique with the weights computed based on the result from the previous training step via a momentum factor β_1, with the moving average of the weight (moving weight velocity) initialized to zero. The moving weight velocity $\Delta\boldsymbol{\omega_e}$ in the current epoch e is computed based on the previous weight update and the current gradient descent with momentum factor β_1 and learning rate α Eq. (2.9).

$$\begin{aligned} \Delta\boldsymbol{\omega}_e &= \beta_1 \Delta\boldsymbol{\omega}_{e-1} + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}} \\ \boldsymbol{\omega}_e &= \boldsymbol{\omega}_{e-1} - \alpha\Delta\boldsymbol{\omega_e} \end{aligned} \tag{2.9}$$

The momentum factor is set in the range $[0, 1]$. With a low momentum factor, the moving weight velocity tends to adapt quickly to the gradient of the cost function (large $(1-\beta_1)$), which could trigger slower convergence when the optimizer navigates through a plateau landscape of the cost function as the gradient is small. In contrast, a high momentum factor allows fast optimization convergence as the weights move towards the extrema at stable high speeds when it is in the direction of the extrema.

Root mean squared propagation - RMSprop employs squares of gradient as regularization factor to compute propagation rate Λ of the parameters.

$$\begin{aligned} \Lambda\boldsymbol{\omega}_e &= \beta_2 \Lambda\boldsymbol{\omega}_{e-1} + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}}\right)^2 \\ \boldsymbol{\omega}_e &= \boldsymbol{\omega}_{e-1} - \alpha\frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}}\frac{1}{\sqrt{\Lambda\boldsymbol{\omega_e}} + \epsilon} \end{aligned} \tag{2.10}$$

with $\epsilon \approx 0$ a factor to prevent division by zero. RMSprop can prevent too slowly or fast decay of the learning rate by adjusting β_2. A low β_2 allows slower learning rate adaptation to the gradient direction and vice versa. However, it takes up more memory than the gradient descent with momentum due to extra computation of the propagation factor Λ for every network parameter.

Adaptive moment estimation - adam exploits the advantages of both momentum and RMSprop to adaptively adjust the updating step for each parameter in the model [**?**]. Apart from the moving average $\Delta\boldsymbol{\omega}_e$, adam also utilizes the square of the gradient as a correction factor $\Lambda\boldsymbol{\omega}_e$ for

adaptive moment computation Eq. (2.11).

$$\begin{aligned}
\Delta\boldsymbol{\omega}_e &= \beta_1 \Delta\boldsymbol{\omega}_{e-1} + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}} \\
\Lambda\boldsymbol{\omega}_e &= \beta_2 \Lambda\boldsymbol{\omega}_{e-1} + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}}\right)^2 \\
\Delta\boldsymbol{\omega}_e^{corr} &= \frac{\Delta\boldsymbol{\omega}_e}{(1-\beta_1)^e} \\
\Lambda\boldsymbol{\omega}_e^{corr} &= \frac{\Lambda\boldsymbol{\omega}_e}{(1-\beta_2)^e} \\
\boldsymbol{\omega}_e &= \boldsymbol{\omega}_{e-1} - \alpha\frac{\Delta\boldsymbol{\omega}_e^{corr}}{\sqrt{\Lambda\boldsymbol{\omega}_e^{corr}} + \epsilon}
\end{aligned} \tag{2.11}$$

with e the current training epoch.
When applying adam, hyperparameters such as exponential decay rate β_1, β_2 are usually fixed to the default values. In contrast, the learning rate α is tuned based on the target training model and data. Adam also requires most memory usage in comparison to the other optimization methods. However, it remains a widely default optimizer in training a deep learning model due to its learning efficacy. In practice, hyperparameters $\beta_1, \beta_2, \epsilon$ are usually set to default values of 0.9, 0.999, and 10^{-8}, respectively, and α is tuned for specific model and data.

2.3 Deep residual network - ResNet

2.3.1 Introduction

As deep neural networks are designed deeper, the networks face gradient vanishing when going through the back-propagation for the parameter updates. This phenomenon occurs when more layers are added to the network, and the training accuracy gets saturated quickly, so we observed a higher training error in a deeper network. This phenomenon also occurs when the added layers are identity mapping of existing shallow models. Deep residual network (ResNet) [**?**] was introduced to tackle this issue by introducing a residual block. An example of a residual block is presented in Fig. 2.8 with a connection that carries out identity mapping of input x to the output of the block. The residual block should have more than

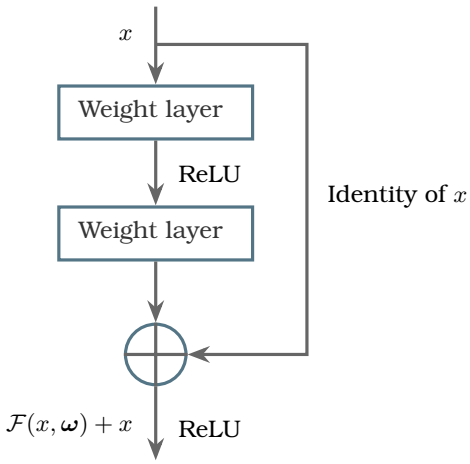


Fig. 2.8 An example of a residual building block with two convolution layers.

one weighting layer so that the deep residual network can benefit from the residual block building [**?**].

2.3.2 Network architecture

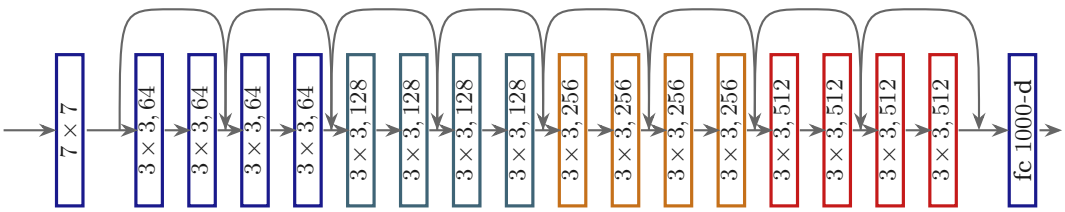


Fig. 2.9 ResNet architecture with four residual building blocks marked by different colors (blue, dark cyan, orange, red). Each contains two stacked convolution layers.

An example of a deep residual network with four residual building blocks (ResNet18) is shown in Fig. 2.9. Each building block contains two stacked residual layers (represented by a shortcut from input to output of each layer). ResNet can also be implemented deeper up to 152 layers. Table 2.1 shows different ResNet architectures with the corresponding number of residual building blocks, kernel sizes, as well as expected input and output size of the networks. Since the networks are evaluated on a dataset of 1000-class-dataset, the output layer is a 1000-dimension-fully-connected layer. For each residual layer (first column), size of the output

Table 2.1 Variants of the ResNet architecture. Residual building blocks are in the square brackets with followed corresponding number of output channels, and number of residual blocks.

layer	output size	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
conv1	112×112	$7 \times 7, 64$, stride 2				
conv2_x	56×56	3×3, max pooling, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv3_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1\,024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1\,024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1\,024 \end{bmatrix} \times 36$
conv3_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2\,048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2\,048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2\,048 \end{bmatrix} \times 3$
pooling	1×1	average pooling, 1 000-d fc, softmax				

features are shown in the second column ($[\cdot \times \cdot]$). In the third column, kernel sizes ($[\cdot \times \cdot,]$) and the number of output channels ($[, \cdot]$) are in the square brackets with the corresponding stacked layers ($[\ ] \times \cdot]$) next to it. Due to the robustness in feature learning and transferable trained models, ResNet has been applied to various research topics such as 2D landmark detection and tracking [**? ? ?**] or abnormality/disease studies [**? ? ?**].

3

Biplanar X-ray cone-beam geometry calibration

3.1 Introduction

Biplanar X-ray cone-beam CT refers to an X-ray acquisition system with two source/detector pairs positioned at different viewing angles relative to a target object. With such a setting, data can be acquired simultaneously from two, e.g., orthogonal directions or in dual-energy mode. The biplanar X-ray systems are widely used in image-guided radiotherapy applications as they provide a fast acquisition, and reduce the exposure of the patients to ionizing radiation [**? ? ?**]. The biplanar acquisition also allows reconstruction of 3D object motion from only a few X-ray radiographs when combined with a pre-recorded CT volume [**? ? ?**]. A biplanar circular cone-beam X-ray CT system ($3D^2$YMOX - 3-Dimensional DYnamic MOrphology using X-rays) [**?**] was built for morphological studies of the living animals. The system is highly modular, and geometry misalignment appears in every new acquisition setup. To obtain a high-quality tomographic reconstruction and exploit the benefits of a biplanar X-ray CT setup, the system needs to be calibrated by estimating the geometric relationship between the X-ray source and the detector pairs with respect to the rotation axis prior to image reconstruction.
Many studies have dealt with single cone-beam X-ray CT system calibration. They include self-calibration methods [**? ? ? ?**], which calculate the geometry parameters of the acquisition system directly from the acquired radiographs of the target objects, and calibration phantom-based techniques [**? ? ? ?**]. Several self-calibration methods estimate the geometry parameters by using an iterative alignment of simulated and measured projection data [**?**], or refining the sharpness of the reconstructed CT images [**?**]. Both approaches are computationally expensive due to iterative CT reconstruction being required. Some other self-calibration techniques effectively reduced calculation costs with the projection-based procedures. For example, Wang et al. [**?**] and Kyungtaek et al. [**?**] correct for the geometry misalignments of a parallel 3D geometry based on the acquired projections. However, self-calibration with a projection-based procedure depends on the object's orientation and position with respect to its projection's coordinate system. Therefore calibration prior to each scan is required, even with an unchanged geometry.

Most X-ray CT geometry calibration methods that rely on fiducial markers employ specifically designed phantoms in which the position of the markers is measured accurately using Coordinate Measuring Machines (CMM). While Liu et al. [**?**] introduced a phantom that carried 12 spherical zirconia markers placed on a triple helix glass phantom, Cho et al. and Chetley et al. [**? ?**] designed a phantom with two rings of evenly placed steel markers on an acrylic cylinder. Efforts have been made to reduce the complexity of a calibration phantom with a phantom of vertically arranged markers proposed by Gross et al. [**?**], or a 14-marker phantom presented by Mennessier et al. [**?**].

Only a few studies have been reported on estimating the geometry of a biplanar X-ray CT system with a calibration phantom, [**? ?**]. Chang et al. presented a method to calibrate a dual-axis tomosynthesis using an acrylic plate phantom holding non-solid and solid spheres. Sawall et al. [**?**] method did not require knowing the sphere marker's position but needed a well-measured nominal system geometry, which is not fulfilled in the $3D^2$YMOX system.

A LEGO phantom-based calibration technique was presented for a single cone-beam X-ray system [**?**]. However, the relative angle between the two systems was not taken into account. It is necessary to have a comprehensive calibration procedure that estimates the relative position of the two cone-beam X-ray CT systems. If this biplanar angle is known, it could be possible to exploit the benefits of a biplanar X-ray CT setup. For example, the system field-of-view can be extended beyond the physical size of a single system detector [**?**] or enhancing CT reconstruction quality as extra data can be acquired in an acquisition from two orthogonal X-ray systems. By choosing two different source energies, more object detail may be revealed in the CT images. Furthermore, when the biplanar geometry is fully calibrated, the two CT volumes obtained from two single systems can also be registered automatedly.

This chapter presents a complete procedure to calibrate the geometry of a modular biplanar cone-beam CT system with a LEGO phantom containing metal markers strategically placed in designated bricks [**? ?**]. The LEGO phantom is easy to build and customize based on the size of target systems. The marker position relative to the phantom can be calculated from the dimensions of LEGO bricks at a reliable accuracy

as LEGO bricks are molded with a dimensional tolerance of 5 µm [**?**]. Furthermore, the proposed calibration method requires no pre-calibrated geometry information, and is capable of calibrating a modular biplanar cone-beam X-ray CT system such as the 3D^{2}YMOX system. The chapter is structured as follows. Section 3.2 presents the proposed methodology to build a low-cost calibration phantom using LEGO bricks, and metal markers along with a deep learning-based procedure to estimate the bead centers accurately. Section 3.3 discusses the experiments that were performed to validate the proposed method. Finally, further discussions and conclusions are presented in Section 3.4.

3.2 Methods

3.2.1 3D^{2}YMOX system

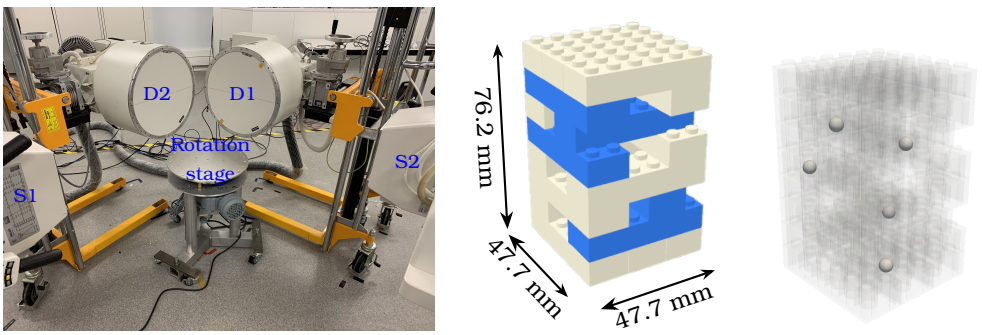


Fig. 3.1 The 3D^{2}YMOX system (a), LEGO calibration phantom with embedded metal markers in the blue bricks (b) and its transparent view (c).

Fig. 3.1a shows the 3D^{2}YMOX system (3-Dimensional DYnamic MOrphology using X-rays system) [**?**] used for morphological and biomechanical research on living animals. The system consists of two X-ray source/detector pairs $\{S1, D1\}$ and $\{S2, S2\}$, and a rotation stage, which is mounted on a wheeled tripod (Fig. 3.1a). The sources $S1$ and $S2$ are mounted on two ceiling gantries that allow to easily position them in 3D space. The orientation of each source around three principal axes is controlled by attached side handlebars. In addition, the two detectors $D1$ and $D2$ are put on two trolleys with hydraulic lifts to adjust their

horizontal and vertical position. Moreover, each detector has a steering wheel that manipulates its orientation in 3D space. Consequently, each device is positioned independently from the others. Therefore, in any new installation, the position and orientation of the source and the detector as well as the position of the stage may change dramatically. With such a setup, it is challenging to align the sources, detectors, and rotation stage properly and accurately measure the geometry. It is therefore essential to perform a calibration to estimate the system geometry as accurately as possible.

3.2.2 LEGO calibration phantom

Phantom-based calibration methods make use of marker (metal bead) positions in the measured X-ray projections to estimate the geometry parameters. The phantom must be built to maximize the contrast between the LEGO structure and the metal markers in the radiographs to facilitate the markers extraction from the X-ray projections. Steel markers with a diameter of $4\,950 \pm 10$ µm are embedded in the hollow cylinders of the bricks by pushing the LEGO bricks on a flat surface to press the metal markers exactly one diameter deep into the cylinders. Then these marker-bearing bricks are placed such that no two markers are within the same vertical bricklayer in the phantom (blue LEGO bricks in Fig. 3.1). This design avoids overlapping markers in the projections.
Moreover, the markers are placed close to the phantom's facets to maximize the coverage area of their projection trajectories on the detector field of view. As studied by Ferrucci et al. [**?**], the coordinate changes due to geometry misalignments are dependent on the marker positions relative to the rotation axis. A strategic design of the phantom can address these coordinate deviations in the misaligned geometry. The phantom dimensions and the number of markers can be adjusted to the size of the system field-of-view. The dimensions of the LEGO bricks and the metal markers were measured by an electronic caliper with 10µm accuracies. The 3D positions of the metal markers in the phantom are calculated relatively to the dimensions of a single LEGO brick.
According to the company disclosure [**?**], the LEGO bricks are molded with a dimensional tolerance of 5µm. With a detector pixel size of 142 µm and a magnification factor of ≈ 1.3 [**?**], the effective voxel size of the

3D^{2}YMOX system is around 100 µm. Therefore, the dimensional tolerance of the LEGO bricks is ≈ 20 times smaller than the effective voxel size of the target system. With the current 292×292 mm image intensifier screens of the 3D^{2}YMOX system, the LEGO calibration phantom was built with only eight bricklayers tall (76.2 mm), a width of 47.7 mm (single 6×2-brick width), and five metal markers. This structure prevents accumulated dimensional errors of the bricks horizontally. Vertical accumulated tolerance is still below the effective voxel size of the 3D^{2}YMOX system. Two identical calibration phantoms were built to validate the feasibility of reproducing the LEGO phantom and to study the effect of the dimensional tolerance on the geometry calibration of a real biplanar X-ray CT system. In addition to the calibration phantoms, a test phantom was built from the LEGO bricks and metal markers with a different structure and size from the calibration phantom for evaluating CT quality as well as calibration accuracy.

3.2.3 Extraction of the bead centers

In the template matching-based method [**?**], the center of each bead is estimated from the center-of-mass (CoM) of its corresponding region of interest (ROI), which is extracted from the calibration projections. However, cone-beam effects and the overlapping projection of the holding structure on the ROIs complicate the CoM calculation. It calls for a robust method that can handle different cone-beam geometry effects as well as asymmetric ROIs and derive center locations more accurately.
The center estimation procedure can be described as finding a mapping model $\mathcal{F}$ that takes the marker ROIs $\boldsymbol{x}$ as inputs with parameters $\mathcal{W}$ and returns the corresponding center coordinates. $\mathcal{W}$ is obtained through an optimization process that minimizes the difference between $\mathcal{F}(\boldsymbol{x}, \mathcal{W})$ and ground-truth center coordinates (u^{gt}, v^{gt}).

$$\hat{\mathcal{W}} = \arg\min_{\mathcal{W}} \left\{ \left[\mathcal{F}(\boldsymbol{x}, \mathcal{W}) - \left(u', v'\right) \right]^2 \right\} \tag{3.1}$$

In Eq. (3.1), F represents any deep learning model that learns abstract features from the input ROIs and maps them to the center coordinates of the bead in the ROI considering t being the hyperbolic tangent function and $t(u^{gt}) = u'$, $t(v^{gt}) = v'$. Two deep learning models (BeadNet) were

trained separately for each center coordinate regression. The goal of BeadNet is to find abstract features of a marker ROI that map to corresponding center coordinates. Choosing a feature learning model is important to have accurate center inference. ResNet50 [**?**] emerges as a deep learning model that is trained on more than a million images for object classifications. ResNet50 is 50 layers deep and is divided into five convolution layers. We exploited the robustness of the pre-trained ResNet50 model in learning abstract object features to continue training for our marker center extraction. The output layer of the model is replaced by a hyperbolic tangent function to adapt to the regression of the markers' coordinate offsets. To this end, two BeadNets were trained at a learning rate of 0.01 using adam adaptive learning rate optimizer with a batch size of 40 ROIs.

The generation of training data is one of the critical steps for deep learning applications. The X-ray energy spectrum can be different for each acquisition. Hence, the training dataset included projections of the calibration phantom simulated with different X-ray source spectra. Moreover, as the cone-beam X-ray CT system was parameterized using 12 degrees-of-freedom, the training dataset needs to replicate possible geometry configurations, which are common settings of the 3D^{2}YMOX system. A set of 400 projections were simulated for each of 120 angles covering $360°$ rotation. For each set, the object and detector orientations and translations were modified by a random value generated from a uniform distribution in the intervals of between $-10°$ and $10°$ and between -30 and 30 mm, respectively. The object yaw ϕ^o was generated randomly up to $200°$ simulating varied orientations of the calibration phantom. A $1\,100$ mm source-detector distance (SDD) along with varied source-object distances (SODs) were simulated for typical positions of the sources, the object, and the detectors of the 3D^{2}YMOX system.

Reference marker center orbits that correspond to the simulated geometries were calculated analytically, with which 25 ROIs were extracted around each marker from every simulated radiograph using reference marker positions. The training dataset contains the marker ROIs and their corresponding ground-truth center coordinate offsets. The xy-coordinate offsets of a ROI are computed as signed differences between the ROI's center and the correct coordinates of the marker's center. Then,

tangent hyperbolic function is applied to the offsets converting them to the output ranges of the deep learning model. In this work, the size of the ROIs is 39×39 pixels and they mainly cover the center patches of the marker projections.

3.2.4 Biplanar geometry parameters

The geometry of a biplanar X-ray CT system can be calibrated separately for each single-source system. However, the biplanar angle α between the two systems is not estimated in this procedure. A comprehensive calibration procedure is needed to fully estimate the geometric parameters of a dual-source cone-beam X-ray CT setup.

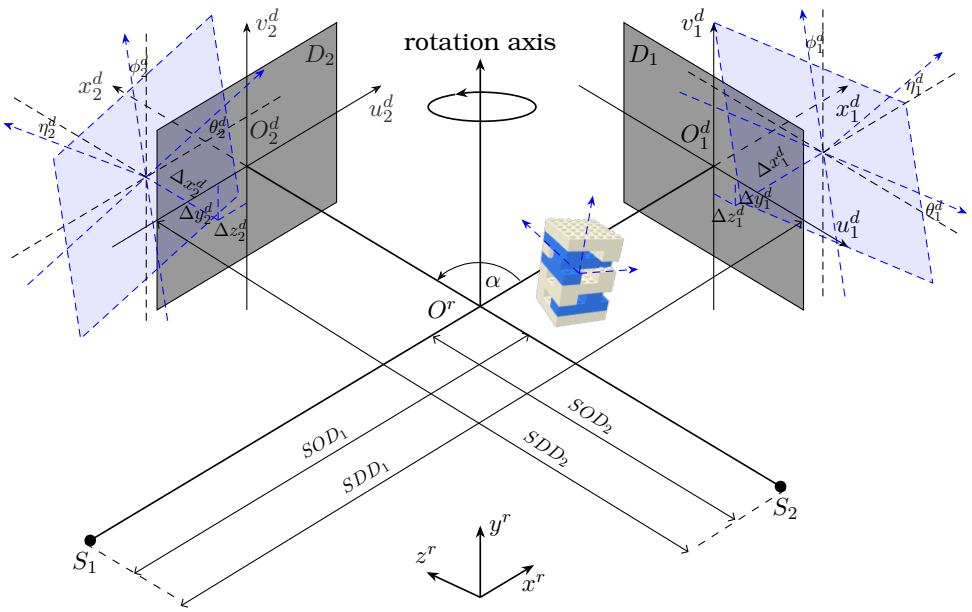


Fig. 3.2 A biplanar cone-beam geometry of an X-ray CT system.

Fig. 3.2 shows an aligned biplanar cone-beam geometry in the black plot. Two perpendicular source-detector pairs with reference to their projection axes are hereafter referred to as $S1, D1$ and $S2, D2$. The sources and the detectors are stationary during acquisition while the object is rotated around the rotation axis. The distances from either source to its corresponding detector (SDD) and the rotation axis (SOD) are known for each acquisition. Two 3D coordinate systems $O^r x^r y^r z^r$ and either $O_1^d x_1^d u_1^d v_1^d$ or $O_2^d x_2^d u_2^d v_2^d$ that originate at the center of rotation, and the center of the

detector panel, are aligned.
To calibrate the system geometry, the position and orientation of the calibration phantom need to be defined accurately, for which they are described by six DOFs with respect to the $O^r x^r y^r z^r$. The DOFs include three distance coordinates $\{\Delta x^o, \Delta y^o, \Delta z^o\}$ and three orientation angles roll θ^o, yaw ϕ^o, pitch η^o about (x^r, y^r, z^r) axis, respectively. The position and the orientation of each detector are defined by six DOFs $\{\Delta x_i^d, \Delta y_i^d, \Delta z_i^d, \theta_i^d, \phi_i^d, \eta_i^d\}$ with respect to the $O_i^d x_i^d u_i^d v_i^d$, $i = \{1, 2\}$. The misaligned detectors, and corresponding geometry parameters are demonstrated in Fig. 3.2, blue plot.
The distance from the sources to the rotation axis and to the detectors are measured after a new acquisition setup. The measurement uncertainties can be modeled by two more parameters Δsod_1 and Δsod_2. The angle between the two optical axes of the two systems is parameterized by biplanar angle α. The biplanar cone-beam geometry setup is therefore described by 21 DOFs $\boldsymbol{\beta} = \{\Delta x^o, \Delta y^o, \Delta z^o, \theta^o, \phi^o, \eta^o, \Delta sod_i, \alpha, \Delta x_i^d, \Delta y_i^d, \Delta z_i^d, \theta_i^d, \phi_i^d, \eta_i^d\}$, $i = \{1, 2\}$.

3.2.5 Geometry calibration

Calibration datasets were acquired with the biplanar cone-beam system, and the technique presented in [**?**] was followed to extract the centers of the markers from the radiographs. Marker-based calibration methods make use of the markers' positions on the detector to estimate the geometry parameters. The reference and the corresponding 2D measured coordinates of marker k in projection n on the detector plane are denoted as $(u_{nk}^{ref}, v_{nk}^{ref})$ and $(u_{nk}^{mea}, v_{nk}^{mea})$, respectively. For every projection angle, the vector that represents the system geometry is transformed with respect to the misalignment parameters. The reference $(u_{nk}^{ref}, v_{nk}^{ref})$ coordinates are defined as the intersections of the rays from the source through the 3D marker centers (x_k, y_k, z_k), with the detector plane.
The measured $(u_{nk}^{mea}, v_{nk}^{mea})$ coordinates are extracted from the calibration data with template matching technique [**?**] and fine-tuned using deep learning (BeadNet) [**?**]. BeadNet is trained from the predefined neural network model ResNet50 [**?**] using a simulated dataset containing X-ray bead projections from different geometry configurations.
The geometry parameter set $\boldsymbol{\beta}$ is estimated by the interior point optimiza-

tion [**?**] of the calibration cost function. The loss is computed as the total Euclidean distance between the reference $(u_{nk}^{ref}, v_{nk}^{ref})$ and the measured $(u_{nk}^{mea}, v_{nk}^{mea})$ coordinates across all projections p_n $(n = 1, ..., N)$, and marker centers $k = 1, ..., K$ with respect to $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left[\left(u_{nk}^{ref}(\boldsymbol{\beta}) - u_{nk}^{mea} \right)^2 + \left(v_{nk}^{ref}(\boldsymbol{\beta}) - v_{nk}^{mea} \right)^2 \right] \right\} \tag{3.2}$$

where $\boldsymbol{\beta} = \{\Delta x^o, \Delta y^o, \Delta z^o, \theta^o, \phi^o, \eta^o, \Delta sod_i, \alpha, \Delta x_i^d, \Delta y_i^d, \Delta z_i^d, \theta_i^d, \phi_i^d, \eta_i^d\}$, $i = \{1, 2\}$. By iteratively adjusting the geometry parameters to align the reference coordinates $(u_{nk}^{ref}, v_{nk}^{ref})$ to those on the calibration radiographs $(u_{nk}^{mea}, v_{nk}^{mea})$, the geometry parameters are estimated.

In the experiment with real datasets, all the geometry parameters are initialized to 0 as no prior knowledge of the geometry parameters is available in the 3D^{2}YMOX system. A complete biplanar geometry calibration procedure is as follows. The phantom orientation around the vertical axis ϕ^o and its position Δy^o are estimated first to align the object vertically. Next, the phantom translations $\{\Delta x^o, \Delta y^o, \Delta z^o\}$ are calibrated, followed by its orientation parameters $\{\theta^o, \phi^o, \eta^o\}$. Then, the calibration phantom parameters along with the biplanar angle are estimated using both datasets, followed by Δsod_1 and the orientation and translation of $D1$ optimization. Finally, Δsod_2 and $D2$ geometry parameters are estimated before all 21 DOFs are fine-tuned. This whole procedure is iterated until the calibration cost function shown in Eq. (3.2) converges. The procedure is iterated 30 times for parameter fine-tuning as either cost function residual or parameter updates are less than 10^{-6} ($^\circ$ or mm) during iterative optimization. The calibration took around 450 seconds to finish on an Intel(R) Core(TM) i7-6800K CPU @ 3.40 GHz PC, with six CPU cores multithreading.

3.3 Experiments and results

3.3.1 Simulation of experimental datasets

The training and validation datasets were generated using the LEGO phantom STL models, and the ASTRA CAD projector toolbox [**?**]. The system vector geometry was calculated with respect to the geometry misalignment parameters for every projection angle using the ASTRA Toolbox

[**?** **?**]. Then, the ASTRA CAD projector simulated X-ray radiographs of the phantom with a 150 keV polychromatic spectrum, the predefined vector geometries, and a detector pixel size of 142 µm, which corresponds to the pixel size of the 3D^{2}YMOX system.
The 44 validation datasets were generated replicating different X-ray cone-beam geometries. Corresponding marker ROIs along with the initial marker center coordinates were extracted, which are later corrected by the trained BeadNets. Two more simulated datasets, with the same detector translation and orientation parameters but different randomly generated object positions and orientations, were generated for 3D CT reconstruction evaluations, including the calibration phantom and a test phantom projections.

3.3.2 Experiments with simulated data

3.3.2.1 BeadNet evaluation using simulated datasets

The bead centers were extracted from the validation dataset with conventional Normalized Cross-Correlation (NCC) method [**?**] and BeadNet (Table 3.1). As can be seen in the Table 3.1, the bead center coordinates are estimated more accurately using BeadNet with a factor of two in comparison to the NCC method. In order to further evaluate the impact of the bead center coordinate errors on the geometry calibration, a preliminary calibration experiment is performed for a biplanar source/detector X-ray system using extracted bead centers of the testing datasets. Fig. 3.4 shows the calibrated geometry parameter errors when employing results from the BeadNet model (orange), and the NCC-based method (blue).

Table 3.1 Errors of marker center detection using NCC template matching and deep learning methods.

(pixel)	u	v
NCC	0.62 ± 0.49	0.54 ± 0.36
BeadNet	0.27 ± 0.16	0.25 ± 0.15

Employing the BeadNet estimated coordinate centers, the translation parameters are calibrated with highest median errors of around 2.5 mm, and the third quartile of 5 mm (center and upper bar of the orange boxes,

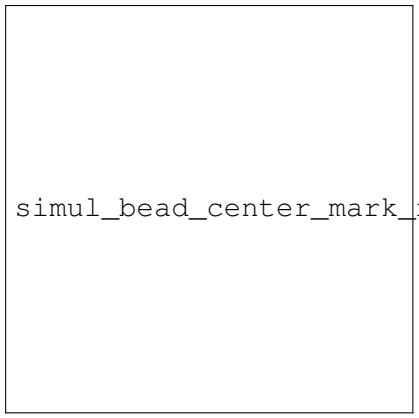


Fig. 3.3 An example of a marker center's coordinate extraction using the NCC (red), BeadNet (white), and ground-truth (green). BeadNet (white) derives closer coordinates to the ground-truth (green) than the NCC method (red).

Fig. 3.4). In contrast, using the conventional NCC method, the parameter median errors are up to 10 mm, with the third quartile of 16 mm. A similar result can be observed for the orientation parameter estimations. Using the conventional NCC bead centers, the calibration errors of the orientation parameters in 75% of the testing datasets are up to $0.8°$ while it is only $0.3°$ when using the BeadNet estimates.

3.3.2.2 Reconstruction from simulated phantom projections

Fig. 3.5 shows the cross-sections of a real phantom reconstruction before (Fig. 3.5a), and after (Fig. 3.5b, 3.5c) geometry alignment. The vertical translation of the detector Δy^d was estimated more accurately using the marker centers obtained with BeadNet. This is why the two axial slices shown in Fig. 3.5b and in Fig. 3.5c are not aligned but are shifted vertically in the 3D CT volumes. Apparent artifacts can be observed in the reconstructed slices before correcting the geometry (Fig. 3.5a), while the brick structure is clearly revealed in Fig. 3.5b and Fig. 3.5c. These images demonstrate that the misaligned geometry was substantially compensated in the reconstruction. However, a closer view at a brick region in the reconstructed volumes (Fig. 3.6) shows that residual blurring is still clearly visible in the reconstruction with the results from NCC bead center extraction method Fig. 3.6a. In contrast, the artifact is significantly reduced in the result with BeadNet Fig. 3.6a. This artifact reduction demonstrates that the more accurately derived bead centers

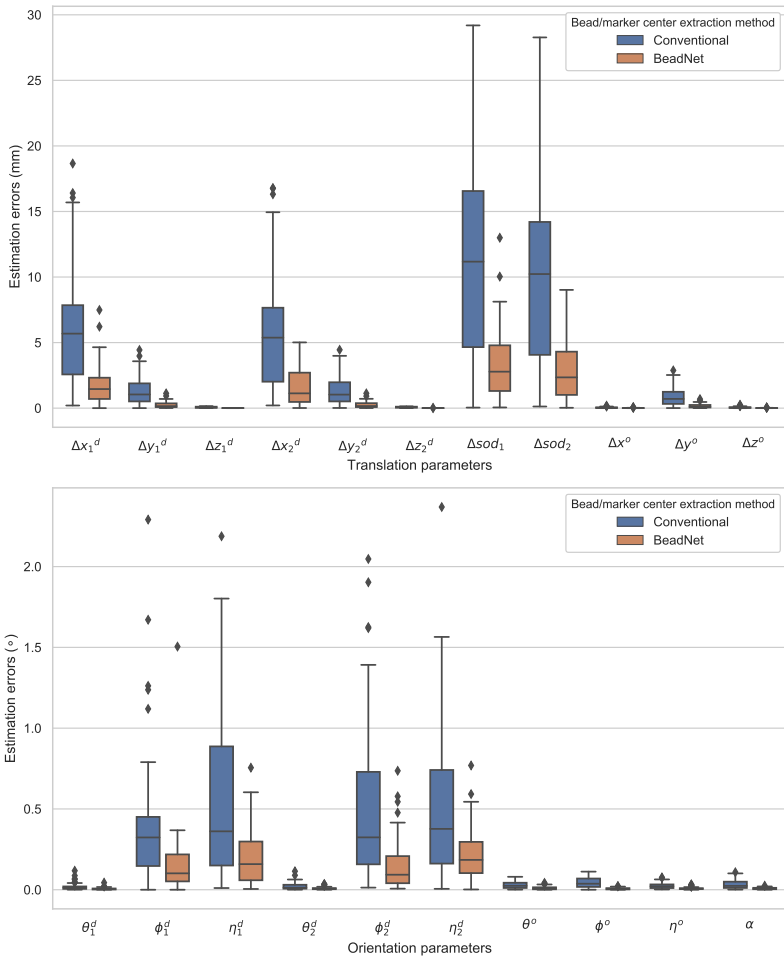
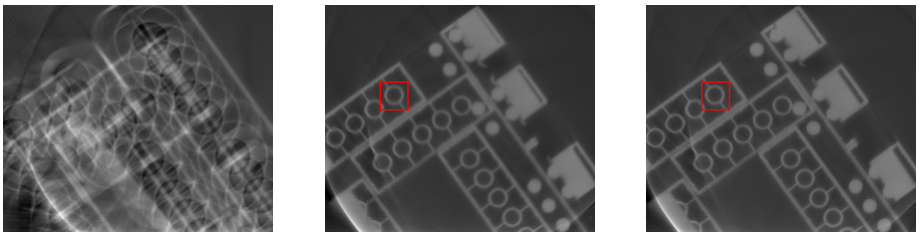


Fig. 3.4 Estimation errors of the translation (a) and the orientation (b) parameters using the NCC method (red) and BeadNet (blue) to extract the marker centers.

using BeadNet have a positive impact on the CT reconstruction quality.

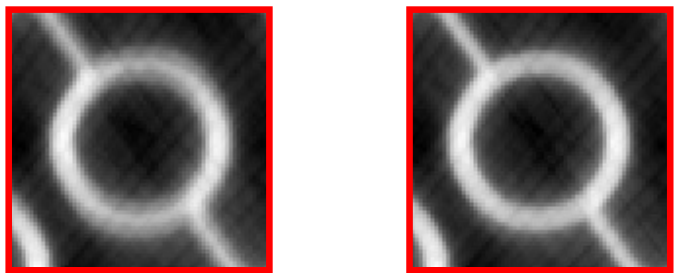
The noiseless simulated biplanar datasets included a calibration and a test dataset of 61×2, and 600×2 radiographs generated with a calibration phantom, and the test phantom, respectively. The ASTRA CAD projector [**?**] casts the X-ray beams through the CAD models of the phantoms in the biplanar cone-beam geometry, which was modified with the geometry parameters, to generate the simulated radiographs of the phantoms. The detectors were simulated as two square flat panels with $2\,048 \times 2\,048$-pixels resolution and pixel size of 142 µm.

As shown in Table 3.2, optical translations (translations along the pro-



(a) (b) (c)

Fig. 3.5 Cross-sections of the reconstructed volume using simulated dataset before (a) and after geometry calibration with the NCC technique (b) and BeadNet (c). The red squared regions are shown in Fig. 3.6.



(a) (b)

Fig. 3.6 With the NCC method, misalignment artifacts still appear at the edges of the bricks in the reconstruction (a), while the artifacts are substantially reduced in the CT slice using BeadNet (b).

jection axes) $\{\Delta x^d, \Delta sod\}$ are estimated with errors on the order of several millimeters as the differences are $470, 3\,600, 820$ and $8\,200$ µm for $\{\Delta x_1^d, \Delta x_2^d, \Delta sod_1, \Delta sod_2\}$, respectively. However, the other translation parameters, including $\{\Delta x^o, \Delta y^o, \Delta z^o, \Delta y^d\}$, and Δz^d, are estimated with a maximum deviation from the ground-truth values of 170 µm. Orientation parameters are calibrated with a precision below $0.1°$.

In a calibration procedure, the biplanar angle α, and all other geometry parameters were initialized to 90^o and 0, respectively. SOD and SDD were fixed to the simulated ground-truth values of $\{779, 1\,123\}$ mm and $\{783, 1\,141\}$ mm for $\{S1, D1\}$ and $\{S2, D2\}$, respectively. To further evaluate the impact of the calibration errors on the CT reconstruction quality, the geometry of the $\{S1, D1\}$ and $\{S2, D2\}$ systems were modified with the initialized, and calibrated geometric parameters prior to the reconstructions of the test phantom, and a piglet specimen. A SIRT algorithm was used to reconstruct the datasets with high-performance

Table 3.2 Calibration errors of the geometric parameters for a simulated biplanar X-ray CT system. Optical translations Δx^d and Δsod are estimated with errors on the order of millimeters (red) due to the high correlation between them.

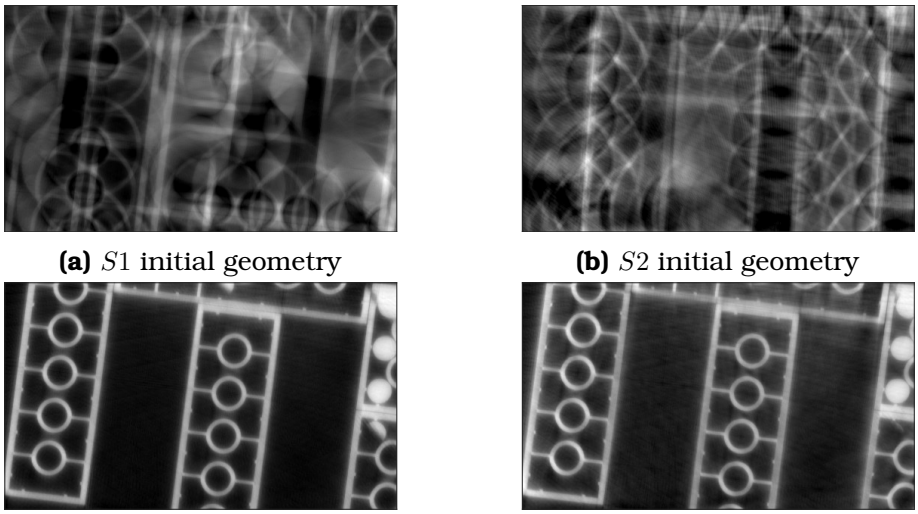
(mm)	Δx^o	Δy^o	Δz^o	Δx_1^d	Δy_1^d	Δz_1^d	Δsod_1	Δx_2^d	Δy_2^d	Δz_2^d	Δsod_2
Init.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GT	−7.96	12.6	−12.6	19.7	−18.5	−10.1	7.21	−10.2	−17.8	14.0	11.9
Err.	0.014	0.12	0.002	0.47	0.036	0.002	0.82	3.6	0.17	0.001	8.2
(°)	θ^o	ϕ^o	η^o	α	θ_1^d	ϕ_1^d	η_1^d	θ_2^d	ϕ_2^d	η_2^d	
Init.	0.00	0.00	0.00	90.0	0.00	0.00	0.00	0.00	0.00	0.00	
GT	4.08	187	4.29	94.1	3.81	2.02	2.08	4.24	2.01	0.67	
Err.	0.004	0.015	0.010	0.008	0.006	0.031	0.021	0.007	0.098	0.070	

GPU primitives ASTRA toolbox [**?** **?**]. Four transverse images of the reconstructed phantom with $S1$ and $S2$ datasets are shown in Fig. 3.7. Without calibration, the geometry misalignments induce severely blurry edge in the LEGO bricks (see Fig. 3.7 a,b). After applying the transformation to the geometry vector with estimated parameters, the misalignment artifacts are corrected (see Fig. 3.7 c,d). We obtain sharp and clear LEGO bricks as well as phantom structures. The two slices are also aligned as the biplanar angle was accounted for.

3.3.3 Experiments with real data

The geometry of the 3D^{2}YMOX system was calibrated with a real LEGO phantom. The calibration data were firstly flatfield and log corrected before they were undistorted to remove the pincushion distortion due to the intensifier curvature [**?**], and the sigmoidal distortion caused by the magnetic field generated during the stage rotation [**?**]. The bead center trajectories were extracted by the NCC method and BeadNet, and used to estimate the geometry parameters.

In this validation, the geometry was corrected for the misalignments before reconstructing a real test dataset obtained in the same geometric configuration with that the calibration projections were acquired. Fig. 3.8 shows the reconstructed slices of the test phantom before (Fig. 3.8a), and after (Fig. 3.8b, 3.8c) calibration. Without compensating the ge-

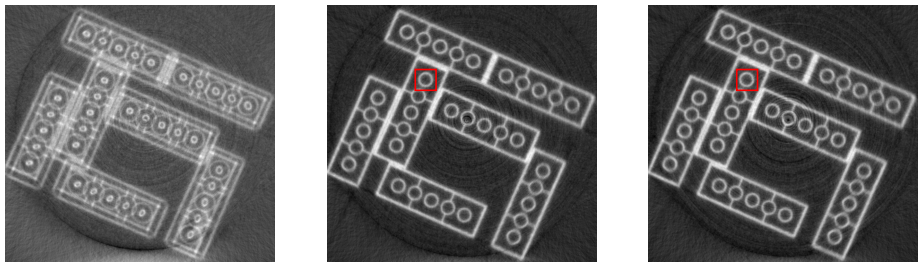


(a) $S1$ initial geometry **(b)** $S2$ initial geometry

(c) $S1$ calibrated geometry **(d)** $S2$ calibrated geometry

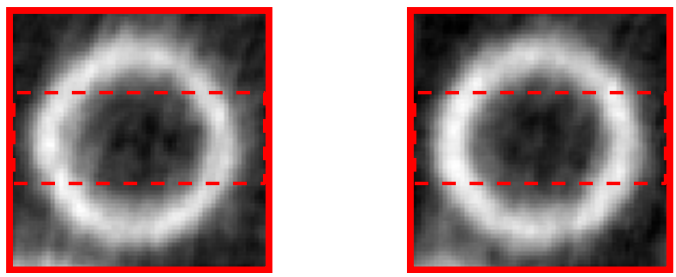
Fig. 3.7 Reconstructions of the test phantom with initial and calibrated biplanar geometry parameters for simulated biplanar datasets. The LEGO bricks are sharply recovered, and misalignment artifacts are eliminated in the CT slices with calibrated geometry (c,d) compared to without applying misalignment correction (a,b).

ometry misalignment, the reconstructed volume suffered from severe artifacts in the form of blurred edges of the LEGO bricks (Fig. 3.8a). In Fig. 3.8b and Fig. 3.8c, however, the shapes and edges of the bricks are well recovered in the reconstruction. This artifact reduction demonstrates that our estimation algorithm is capable of calibrating a real X-ray CT system. Moreover, the artifacts are better suppressed in Fig. 3.8c than in Fig. 3.8b, as highlighted in red, and displayed in Fig. 3.9a, and Fig. 3.9b, respectively. Additionally, Fig. 3.10 shows the accumulated intensity profiles that were plotted through the center rows (dashed red) in the ROIs Fig. 3.9a, and Fig. 3.9b from the conventional NCC method (orange) and BeadNet (blue), respectively. The line plots indicate that the contrast was slightly improved in the reconstruction from the BeadNet method.
Along with the test phantom, a piglet dataset was used to evaluate the quality of the CT images with calibrated geometry. Real X-ray radiographs of the calibration phantoms, the test phantom, and the piglet were acquired by the 3D^{2}YMOX [**?**] system at a resolution of $2\,048 \times 2\,048$-pixels. In this acquisition, the datasets were acquired with two source energies and currents of 57 kV, 40 mA, and 59 kV, 40 mA for $S1$ and $S2$, respectively.



(a) (b) (c)

Fig. 3.8 Cross-sections extracted from the reconstructions of a LEGO test phantom before (a) and after geometry calibration using the NCC method (b), and BeadNet (c) to extract the marker centers. Without compensating for the geometry misalignment, the internal brick structures are distorted (a). In (b, c), the artifacts are considerably reduced, revealing sharp edges and apparent shapes of the LEGO bricks. Moreover, the artifacts are better suppressed in (c) than in (b) as highlighted in red and shown in Fig. 3.9b and Fig. 3.9a, respectively.



(a) (b)

Fig. 3.9 Misalignment artifacts persist at the LEGO bricks' edges and distort the shapes in the NCC method (a). In contrast, the edges are better recovered and sharper in the CT slice using BeadNet (b).

The X-ray tubes were limited to six seconds of continuous radiation to avoid overloading the X-ray tube. Four modes of acquisition are available with a maximum of 900 X-ray frames per rotation. With these technical constraints, it is beneficial to incorporate both datasets into a single CT reconstruction to enhance the CT quality in either single or dual-energy mode. In these experiments, each test phantom dataset contains 360 biplanar projections, while 450 projections of the piglet were acquired from each cone-beam X-ray system for CT reconstruction.

The X-ray image intensifiers in the 3D^{2}YMOX system cause two major geometric distortions, namely pincushion and sigmoidal distortion [**?**]. The pincushion distortion is the result of the incident X-ray to be detected

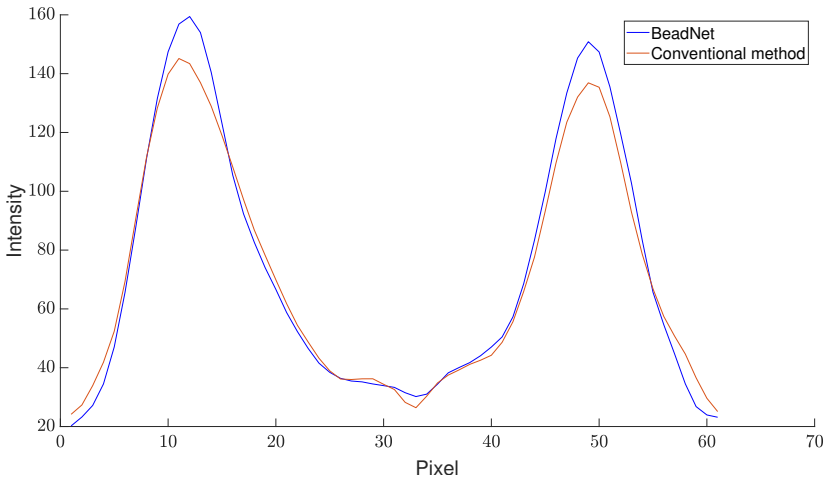


Fig. 3.10 Intensity profiles plotted through the rows (dashed boxes) in Fig. 3.9a and Fig. 3.9b.

on a curved input phosphor, while the latter is due to the magnetic interaction of the produced photo-electrons inside the image intensifier. The projection-dependent distortion correction described in [**?**] was applied to correct for these distortions. Flatfield and log correction were applied to the acquired radiographs to compensate for the different responses in the detectors.

As shown in the simulated experiments, Δsod_i along with the detector position on the optical axis (Δx_i, $i = \{1, 2\}$) are highly correlated, and all impact the magnification of the object projection. Taking into account both parameters induces significant redundancy and error in the calibration. Therefore, Δsod_i is eliminated in the followed experiments with the real datasets. Only detector displacements along two optical axes Δx_i^d, $i = \{1, 2\}$ are accounted for calibration.

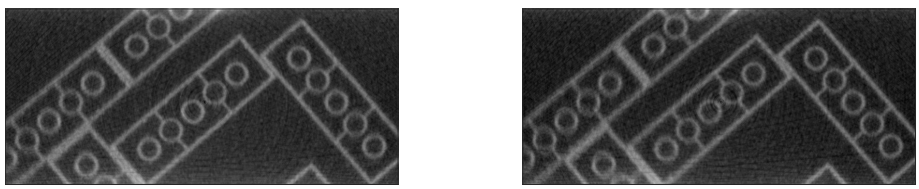
The biplanar geometry was calibrated with 90 radiographs acquired from each single-source system. In these experiments, all the geometry parameters were initialized to 0. SOD and SDD were fixed to the measurements of $\{1\,002, 1\,303\}$ mm and $\{978, 1\,226\}$ mm for $S1, D1$ and $S2, D2$, respectively. The procedure was repeated with both calibration phantom datasets, and the initializations along with the estimated geometry parameters are shown in Table 3.3. As can be seen in the table, the detector translation and orientation parameters are calibrated with maximum differences of

3.5 mm and 6.2°, respectively. Both calibrations with the two phantoms derive the same value of the biplanar angle α.

Table 3.3 Calibrated geometry parameters of the 3D^{2}YMOX system with two identical calibration phantoms.

(mm)	Δx^o	Δy^o	Δz^o	Δx_1^d	Δy_1^d	Δz_1^d	Δx_2^d	Δy_2^d	Δz_2^d	
Inits.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Phantom 1	−16.6	66.1	−10.1	−5.31	−31.6	−3.35	−4.67	−30.63	−0.06	
Phantom 2	16.7	65.4	−22.3	−1.85	−30.8	−3.61	−6.08	−29.6	−0.05	
(°)	θ^o	ϕ^o	η^o	α	θ_1^d	ϕ_1^d	η_1^d	θ_2^d	ϕ_2^d	η_2^d
Inits.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Phantom 1	0.401	−28.9	0.612	89.6	−0.049	−3.41	−0.89	0.459	−2.00	−1.12
Phantom 2	0.118	42.6	0.372	89.6	−0.424	2.79	−4.90	0.441	−1.44	−1.02

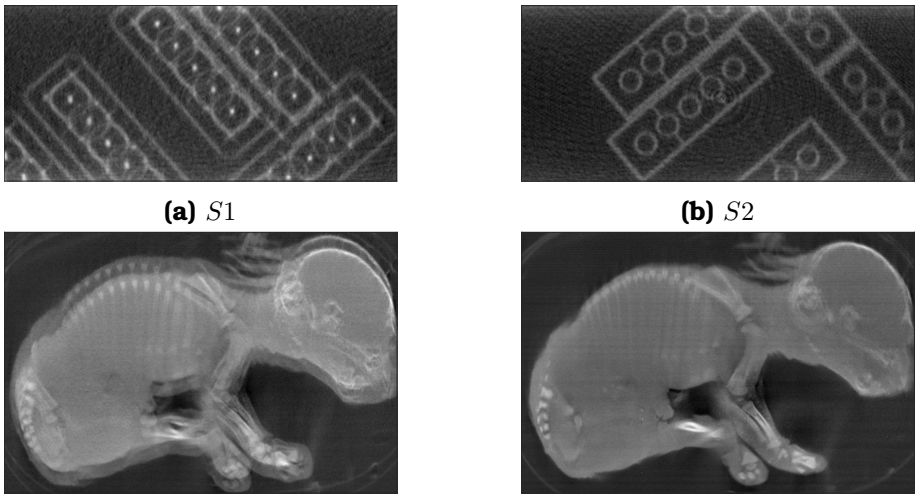
To study the impact of these differences of the calibrated parameters with two calibration phantoms in the quality of the reconstructed images, a test phantom dataset acquired in the 3D^{2}YMOX system was reconstructed with two sets of calibrated parameters by ASTRA toolbox [**? ?**] SIRT algorithm. Two CT slices of the test phantom are shown in Fig. 3.11. As shown in the figures, more apparent misalignment artifacts appear in the reconstruction with calibrated geometry by phantom 2 (Fig. 3.11b, lower-left corner), compared to the result with phantom 1 (Fig. 3.11a).



(a) Calibration with phantom 1 **(b)** Calibration with phantom 2

Fig. 3.11 Reconstructions of the test phantom acquired by the 3D^{2}YMOX system after biplanar geometry calibration with real calibration phantoms.

Fig. 3.12 shows four CT slices from the reconstructed volumes of the test phantom (a,b), and the piglet (c,d) biplanar dataset without geometry calibration. The reconstructed slices a of the test phantom Fig. 3.12



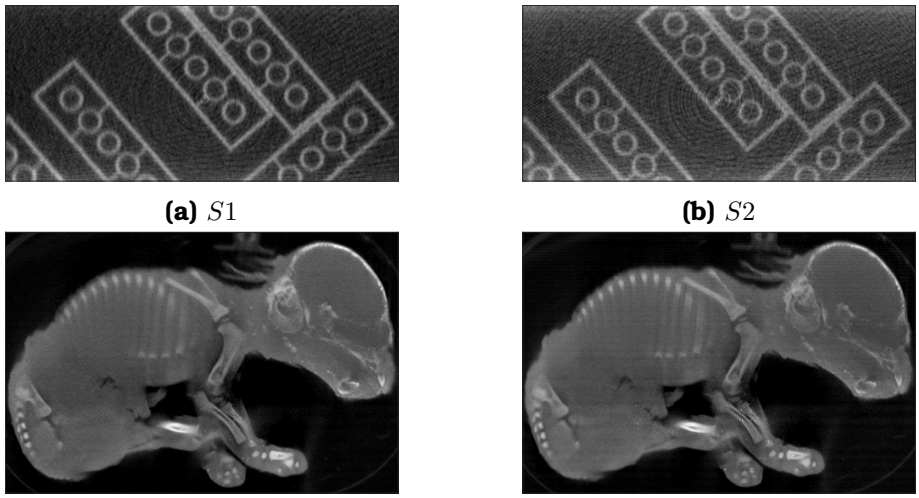
(a) $S1$ **(b)** $S2$

(c) $S1$ **(d)** $S2$ - rotated

Fig. 3.12 Reconstructions of the test phantom and the piglet with real datasets acquired by the 3D^{2}YMOX system and the initializations of the biplanar geometry parameters. The misalignment artifacts are less severe in the reconstruction with $\{S2, D2\}$ datasets (b, d) compared to (a, c) due to the initializations of the geometric parameters turning out to be more precise.

with dataset from $S1$ is in a different orientation compared to the slice from the $S2$ dataset (Fig. 3.12b). In Fig. 3.12d, the CT slice of the piglet specimen with the $S2$ dataset was rotated to a similar orientation as in Fig. 3.12c for a better visualization. This orientation difference is mainly due to the uncalibrated biplanar angle. Moreover, without compensating for the geometry misalignment, the LEGO bricks and the piglet's internal structure are blurry due to the misalignment artifacts. The effect of misalignment is more severe in the reconstruction with the $S1$ datasets as shown in Fig. 3.12a,c compared to the CT slices from $S2$ (Fig. 3.12b,d) due to less accurate initial measurements of SOD_1 and SDD_1.

With a corrected geometry, the misalignment artifacts are significantly suppressed, revealing a clear image of the LEGO bricks, and the piglet skeleton as shown in Fig. 3.13. The CT slices obtained with the $S1$ and $S2$ datasets are aligned in the same orientation when the biplanar angle was accounted for. This result suggests a possibility of having a dual-energy view of an object acquired with the 3D^{2}YMOX system. To study the benefit of the dual-source acquisition, three reconstructions of the test phantom were performed using two single-source and a dual-source datasets. The single-source datasets are a subset of the full rotation

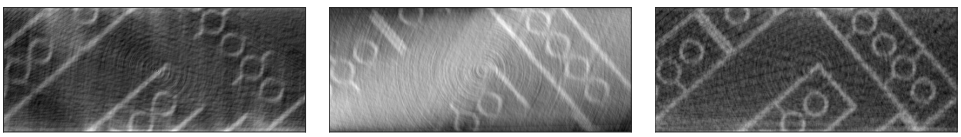


(a) $S1$ **(b)** $S2$

(c) $S1$ **(d)** $S2$

Fig. 3.13 Reconstructions of the test phantom (a, b) and the piglet (c, d) datasets acquired by the 3D^{2}YMOX system with calibrated geometry. The LEGO bricks and the piglet structure are clearly visible in the CT slices with geometry correction.

dataset with a missing angular range of $60°$. Two single-source datasets are concatenated to generate a dual-source dataset as if it was acquired with a biplanar angle, which is a sequence of the projection angles of $\{S1, D1\}$ and their shifts by the biplanar angle α. The geometries of both systems were corrected with calibrated parameters for the reconstruction of the dual-source dataset. As shown in Fig. 3.14, the CT slices reconstructed with the single-source datasets (a, b) are subjected to missing wedge artifacts. The LEGO bricks were only partly visible in both slices, while in the dual-source slice (c), the LEGO bricks are well reconstructed. This experiment demonstrated that, with a calibrated biplanar geometry of the 3D^{2}YMOX system, it is possible to reconstruct two datasets acquired simultaneously and/or in dual-energy mode from the two single cone-beam X-ray systems.



(a) $S1$ **(b)** $S2$ **(c)** $S12$

Fig. 3.14 Reconstructions of the test phantom datasets from single (a, b), and dual (c) X-ray source of the 3D^{2}YMOX system with simulated $60°$-missing wedge of either single system. The reconstruction of the dual-source dataset (c) shows that the missing wedge artifact can be corrected by incorporating both single datasets into the biplanar reconstruction.

3.4 Discussion and conclusion

In this chapter, a comprehensive method to calibrate the 3D^{2}YMOX system was presented, a highly modular biplanar X-ray CT system, with a LEGO phantom. The simulation experiments demonstrated that a LEGO phantom could be used to accurately calibrate the geometry of a biplanar X-ray CT system, except for the optical translation parameters. This can be explained by a high correlation between these two parameters, due to which the errors cancel each other out in the reconstruction. The translation of the rotation center along the optical axis was excluded from the calibration with real datasets to verify this correlation. The piglet CT reconstructions show misalignment artifact significantly reduced after biplanar geometry alignment.

When the biplanar angle is accounted for in the reconstruction, the two CT volumes obtained from the individual X-ray systems are aligned. Moreover, the two datasets acquired with each X-ray cone-beam system can be combined for a biplanar reconstruction, opening up the possibility of a dual-energy and/or faster scan. Experiments with two real LEGO phantom datasets demonstrated that our proposed method could be applied to practical dual X-ray CT systems. Further study on the difference between the geometry parameters estimated with two identical calibration phantom datasets and its impact on the reconstruction quality needs to be done.

In the experiment with the real datasets, the differences in CT reconstruction quality between the datasets acquired with $S1$ and $S2$ can be explained by the fact that the two single cone-beam X-ray systems are not identical. The X-ray projections, and the flatfield images acquired

from the two systems, differ in terms of intensity and noise level. They, therefore, result in unequal reconstruction quality and contrast. A further study needs to be done on acquisition settings in terms of hardware and software configurations to optimize the CT reconstruction quality.
In conclusion, the proposed LEGO calibration procedure can be a valuable solution to calibrate the biplanar geometry of dual cone-beam X-ray CT systems. In future work, thoroughly evaluation the CT reconstruction quality with the calibrated biplanar geometry parameters is necessary. Further study on quantifying calibration accuracy in terms of voxel resolution also needs to be done.

4

AUTOMATIC LANDMARK DETECTION AND MAPPING WITH BONENET

4.1 Introduction

Understanding 3D kinematics of an animal has long been a topic of interest in veterinary research [**? ? ?**]. Such motions can be reconstructed by aligning a 3D reference model to a series of X-ray projection images, which is generally known as 2D/3D registration [**?**]. Intensity-based and feature-based methods are the two major approaches of 2D/3D registration [**? ?**].

Intensity-based 2D/3D registration methods rely on the pixel/voxel gray values to reconstruct the 3D poses of an object from 2D images with reference to a 3D model. A similarity measure (SM) is computed as the intensity or gradient difference between the acquired 2D projections of the object and simulated projections of the 3D reference model [**? ? ? ?**]. The object's pose parameters are then estimated by minimizing the SMs. These methods, however, usually require a good initialization of the pose parameters to avoid the optimizations converging to local minima. Khamene et al. [**?**] dealt with this problem by pre-calibrating the system geometry, and Varnavas et al. [**?**] pre-registered the target pose to a broad range of possible poses within a 2D library generated from a 3D CT object model. The intensity-based registration accuracy also depends on the SM robustness, which is sensitive to the different gray value distributions across image modalities or acquisition setups. To tackle this issue, Birkfellner et al. [**?**] presented stochastic rank correlation as an intensity invariant SM with stochastic sampling, while Munbodh et al. [**?**] calculated SM from Poisson and Gaussian distribution models of CT and X-ray images, respectively. Intensity-based methods also involve computationally expensive simulations of the 2D radiographs during parameter estimation. Finally, projecting a 3D CT volume onto a 2D plane suffers from the loss of depth information [**?**].

Feature-based registration techniques circumvent the computational cost of the intensity/gradient-based methods [**? ?**]. The object's geometric features, such as curves, surfaces, landmarks, etc., are extracted and mapped to the corresponding features on the 3D model to obtain the orientation and translation parameters of the object. Feature-based registration methods allow fast estimation of the pose parameters as no reconstruction or simulation of the 2D radiographs is required during

optimization. Baka et al. [**?**] and Ito et al. [**?**], for instance, estimated the 3D motion model of an object by matching the simulated and measured object curves. However, obtaining corresponding curves proved to be challenging as they are subject to the image's dynamic range and contrast. Geometrical landmarks have been suggested to represent a bone for kinematics registration [**? ? ?**]. Joint kinematics are usually modeled as a combination of articulated transformations of individual bones, and geometric landmarks are manually annotated by experienced operators. Haase et al. [**?**] applied an active appearance model to track the geometrical landmarks of birds of different species. However, manual landmark annotation and tracking relies on the acquisition setup and experts' experience, such as that from a radiologist. Annotating the landmarks or automatically detecting them while maintaining the mapping for registration is non-trivial, raising the need for an automated and robust landmark detection method. Cai et al. [**?**] automated the landmark candidate selection based on Harris corner detection, which relies on the local intensity of image patches and does not account for global correlations, hence reducing its robustness.

Recently, following the advance of deep learning techniques in solving a wide range of computer vision problems, deep networks have been proposed for automated landmark detection [**? ? ? ?**]. Since deep learning models can learn and generalize abstract features from a large amount of data, they are robust for landmark detection. Liao et al. [**?**] applied a Siamese network to detect a set of points of interest (POIs) in an input X-ray image. Although the POIs selected from CT models by a random method result in convergence during training, the randomization might induce overlapping POIs in 2D projections. DeepLabCut [**?**] is a well-known deep network for automatic landmark detection and tracking in optical images, which requires relatively few (hundreds) of labeled images to fine-tune a ResNet-based neural network for a new type of data or object. The method was applied to marker tracking on an X-ray videography scene that followed the positions of the markers attached to animals during their feedings [**?**]. However, DeepLabCut requires manual landmark annotation in video frames that are used to generate the training dataset. This procedure is non-trivial and prone to human errors, especially with multiple landmarks usually distributed densely on

each bone in biological X-ray data. PVNet [**?**] is another deep learning model recently proposed to automatically detect nine 2D landmarks in optical images. To tackle the complexity of 3D pose reconstruction from a single X-ray radiograph of a biological object, PVNet requires customizations for inference of more landmarks and application to X-ray images.
In this chapter, the challenges of automatic landmark detection and tracking are tackled by a strategic approach that consists of two building blocks: an automated 3D landmark extraction technique, and a deep neural network for 2D landmark detection [**?**]. For 3D landmark extraction, a technique based on the shortest voxel coordinate variance is proposed to extract the 3D landmarks from the 3D tomographic reconstruction of an object. For 2D landmark detection, a customized ResNet18-based neural network, BoneNet, is proposed to automatically detect geometrical landmarks on X-ray fluoroscopy images. It relies on a simulation module to generate well-labeled training, validation, and test dataset to eliminate human errors in manual landmark annotation. The module simulates different articulated poses of an animal using a single high-resolution 3D CT model. 3D reference landmarks are then extracted automatically using the same CT model. To this end, two techniques based on a shortest coordinate variance to define two types of 3D landmarks: bounding and SIFT (Scale-Invariant Feature Transform) landmarks are presented. The bounding landmarks [**?**] are selected from the object voxels, while the SIFT landmarks are obtained from 3D SIFT keypoints extracted for conventional image matching [**?**]. Finally, BoneNet, inspired by PVNet [**?**], is trained to detect 2D landmarks in fluoroscopy images automatically. The network architecture is customized to better extract abstract features from complex X-ray image data with more landmarks.
The chapter is organized as follows. Section 4.2 presents the proposed methodology for 3D landmark extraction from the reference model of the object, along with the process of detecting the 2D landmarks accurately with deep learning and reconstructing the object poses using a least-squares optimizer [**?** **?**]. A technique to simulate realistic 3D articulated motions of the object is also presented in this section. Then, experiments using simulation data to validate the feasibility of the proposed method are discussed in Section 4.3. Finally, further discussion and

the conclusion are presented in sections Section 4.4 and Section 4.5, respectively.

4.2 Methods

4.2.1 3D pose parameterization

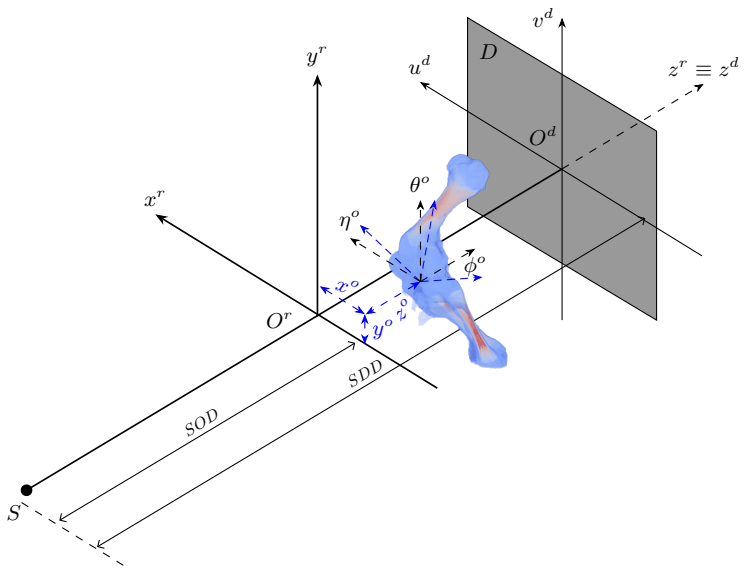


Fig. 4.1 The geometry of a cone-beam acquisition system with an X-ray source S and a detector plane D. Object position and orientation with reference to the acquisition coordinate system $O^r x^r y^r z^r$ are represented by six parameters $\{x^o, y^o, z^o, \theta^o, \phi^o, \eta^o\}$. (Color map represents voxel intensities.)

The animal motion in an X-ray video can be described by a rigid motion for representing its body's position and orientation with respect to the acquisition geometry, and articulated transformations of bones of interest and soft tissues relative to individual joints. 2D/3D registration involves both estimation of the animal body's transformation in the acquisition geometry and its 3D pose with respect to the reference model. Fig. 4.1 shows the geometry of an X-ray cone-beam acquisition system that is used to acquire animal fluoroscopy images. The system is assumed to be calibrated and the X-ray radiographs are corrected for pincushion and sinusoidal distortions in advance using the techniques presented in [**? ?**]. In other words, the perpendicular projection of the X-ray source on

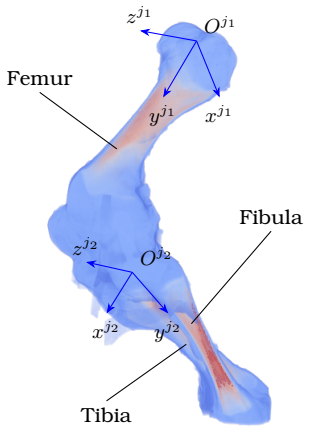


Fig. 4.2 An example of joint coordinate systems of a piglet hindlimb with two major bones (femur and tibia). Each local coordinate system is represented by three axes $(x^{j_i}, y^{j_i}, z^{j_i})$ which are the vertical, longitudinal, and transverse axis of joint $j_i, i = 1, 2$, respectively.

the detector plane $O^d u^d v^d$ coincides with the detector center O^d. Also, the distances from the source to the acquisition system's isocenter (SOD) and the detector plane (SDD) are assumed to be known. In this setting, the 3D position and orientation of the animal are represented by six parameters $\{x^o, y^o, z^o, \theta^o, \phi^o, \eta^o\}$ about three axes (x^r, y^r, z^r).

As the locomotion of an animal involves a chain of contraction and relaxation of different muscles and tendons [**?**], the articulated transformation of bone j_i can be modeled by rotations around the bone's principal axes. The axes include the vertical x^{j_i}, longitudinal y^{j_i}, and transverse axis z^{j_i} (Fig. 4.2) with three corresponding rotations, namely yaw θ^{j_i}, roll ϕ^{j_i}, pitch η^{j_i}. The three axes form the bone local coordinate system originating at the joint O^{j_i}. In the scope of this work, only clockwise and counterclockwise rotations of the bones about their transverse axes are considered, i.e., the rotation η^{j_i} around the z^{j_i} axis. As joint j_1 is chosen as a parent joint for articulated transformation, the orientation η^o about the horizontal axis x^o is equivalent to the joint rotation η^{j_1}, therefore, η^o is suppressed to avoid redundancy in the pose reconstruction. In total, $5 + N$ parameters $\boldsymbol{\tau} = \{x^o, y^o, z^o, \theta^o, \phi^o, \eta^{j_i}\}$ are reconstructed, with $i = 1 \ldots N$ and N is the number of joints under consideration.

4.2.2 Landmark-based 2D/3D registration

The goal is to align 2D detected landmarks from acquired fluoroscopy images with projections of their 3D reference landmarks to estimate $\boldsymbol{\tau}$. In other words, the registration parameters are the result of minimizing the total distances between 2D detected landmarks (u^m, v^m), and the computed projections of 3D reference landmarks (u^r, v^r) using $\boldsymbol{\tau}$ across all N joints and K landmarks. The estimated parameters $\hat{\boldsymbol{\tau}}$ are defined in Eq. (4.1):

$$\hat{\boldsymbol{\tau}} = \arg\min_{\boldsymbol{\tau}} \left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} \omega_{ik} \left((u_{ik}^r(\boldsymbol{\tau}) - u_{ik}^m)^2 + (v_{ik}^r(\boldsymbol{\tau}) - v_{ik}^m)^2 \right) \right\} \tag{4.1}$$

where the distances between the measured and reference landmarks are penalized by different weights ω_{ik} based on their hypothesis covariances [**?**], which will be further discussed in Section 4.2.4.

To avoid local minima during estimation of the parameters, the object's position and orientation with respect to the acquisition coordinate system are estimated before the joint parameters are reconstructed. The detailed process is as follows. First, the projection angle ϕ^o is adjusted to align the object orientation to the acquisition angle. Next, the object coordinate along the vertical axis y^o is estimated prior to the reconstruction of the three offsets $\{x^o, y^o, z^o\}$. After that, the two joint articulation angles $\{\eta^{j_1}, \eta^{j_2}\}$ are estimated. Finally, the object orientations with respect to the world coordinate system $\{\theta^o, \phi^o\}$ are estimated. This process is iterated until the loss function evaluation or all the parameter updates are less than 10^{-8}.

4.2.3 3D landmarks

3D reference landmarks should be key points characterizing the shape of the bones and should be easily distinguishable in the 3D reference model as well as in the 2D radiographs of the whole object. Several methods define 3D reference landmarks based on the 3D model of the object. One of the commonly used methods in computer vision finds a bounding box around the object and uses its vertices as the 3D reference landmarks for registration [**?** **?**]. Peng et al. [**?**] introduced a technique based on Euclidean distance between voxels and the object's center (C) to define 3D landmarks of an object given its 3D model. The

method avoids involving inaccurate bounding box vertices as the 3D landmarks are drawn from the voxels that belong to segmentation of the 3D object. Although the method showed its advantages over the conventional shape description based on bounding box, there is a risk of choosing 3D landmarks that are too close to each other, resulting in overlap in the 2D radiographs. The reason behind this is that a new landmark was defined as the object voxel with the largest distance to the C of the already selected landmarks. The C therefore starts to overlap with the original object's C, and new landmarks may gather close to the existing landmarks. To solve this problem, a comprehensive scheme based on the shortest voxel coordinate variance to keep the landmarks distant from the C and from each other was introduced. Two types of landmarks are determined, namely bounding (similar to [**?**]) and SIFT (Scale-Invariant Feature Transform) landmarks [**?**]. While the bounding landmarks are selected from ordinary bone voxels, the SIFT landmarks are selected from 3D SIFT keypoints of the bone volume. The 3D SIFT keypoints are the local extrema of the image's Laplacian of Gaussian $d(x, \sigma)$ within a sliding window of 2n-connected l^1 neighborhoods [**?**] that is computed with Eq. (4.2):

$$d(x, \sigma) = \mathcal{I}(x) * \Delta g_\sigma(x) \tag{4.2}$$

where $d(x, \sigma)$ is approximated by the difference of Gaussian $g(x)$ at two different scale spaces $g_\sigma(x)$ and $g_{\sigma+\delta}(x)$, with σ and $\sigma+\delta$ the two Gaussian standard deviations, and $\mathcal{I}(x)$ the 3D object image. The SIFT keypoints were selected if they satisfy Eq. (4.4), with a scale threshold α [**?**].

$$d(x, \sigma) = \mathcal{I}(x) * (g_{\sigma+\delta}(x) - g_\sigma(x)) \tag{4.3}$$

$$|d(x, \sigma)| \geq \alpha \max_{x,\sigma} |d(x, \sigma)| \tag{4.4}$$

The landmarks should distribute near/over the bone surface to better characterize its shape and avoid overlapping 2D projections. The shortest coordinate variance scheme is applied to draw the bounding and SIFT landmarks from their initial bone voxels and SIFT keypoints sets, respectively. The scheme to select a list of landmarks from their initial set is as follows:

1. Compute voxel center coordinates (C) as the mean of all voxel coordinates of the bone segment.

2. Compute 3D coordinate variances of the bone voxels. These variances correspond to the eigen values obtained from principal component analysis (PCA) of the bone voxel coordinates. The smallest and the largest eigen values imply the bone minor and major dimensions, respectively. The landmarks should spread closely to the bone surface to better describe its shape. Therefore, the smallest eigen value σ_{min} is used to compute a distance threshold in the later step.

3. Choose the first landmark with the largest Euclidean distance to the C. Add the landmark to the list.

4. All the other landmarks l are added if their distances to the C are largest, and their distances to the existing landmarks m in the list satisfy $d_{lm} \geq \lambda\sigma_{min}$, with a scale threshold λ chosen heuristically depending on the bone shape and size.

4.2.4 Automatic detection of 2D landmarks with BoneNet

To correctly reconstruct the 3D pose parameters of an animal, 2D landmarks must correspond to 3D landmarks of the reference model and be detected with the lowest possible coordinate errors. Peng et al. [**?**] trained a deep neural network (PVNet) to automatically detect 2D landmarks in an optical image scene. The 2D coordinates of each landmark were encoded by a voting vector field that points toward the landmark position in the 2D image. PVNet was based on the ResNet18 architecture [**?**], and obtained by first discarding subsequent pooling layers of ResNet18 when the feature maps were 1/8 size of the original input samples. Then, the fully connected layer was replaced by a convolution layer at the network output. Finally, up-sampling (interpolation) combined with skip connections and convolutions were applied to reconstruct the original image sizes for the bone segments and voting vector fields of the landmarks.

PVNet inherited from ResNet18 four main convolution blocks, which were constructed from sequences of basic blocks in the feature encoding stage. Each basic block is formed by two 2D convolutions followed by batch normalization and a ReLU unit. PVNet was designed for accurate inference of only nine landmarks in optical images [**?**], which resulted

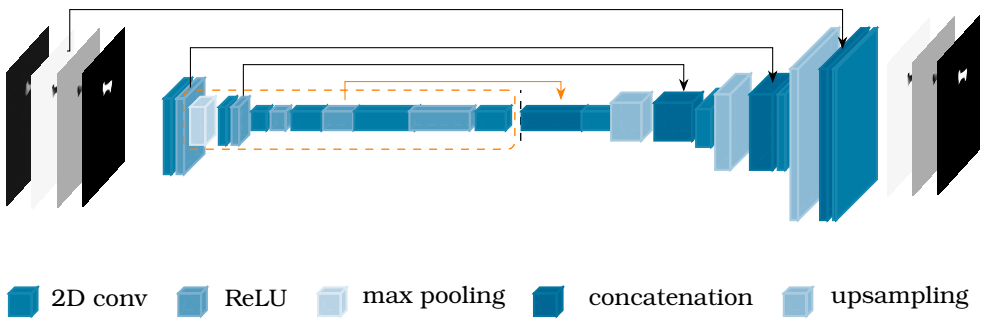
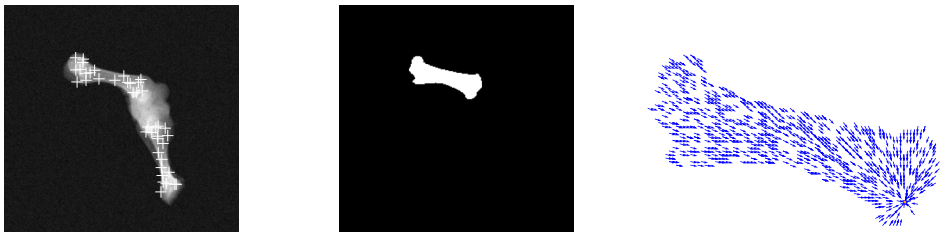


Fig. 4.3 BoneNet, the proposed customized network architecture for automatic 2D landmarks detection of a complex biological object with newly added basic blocks in the convolution blocks (orange dashed lines) and connection (orange arrow). The vertical, dashed line separates the feature learning (left) and the interpolation stage (right) of the deep network.

in an unstable and slow convergence when applying to X-ray images with a higher number of landmarks. Therefore, the model needs to be adapted to such data. We customized PVNet as follows. New 1,1,2, and 1 basic blocks were added to the four convolution blocks in the original PVNet model, respectively. A new connection from the 3^{rd} convolution block replaced the shortcut from the 2^{nd} convolution block to the first up-sampling layer. The number of features in the subsequent layers were also adjusted accordingly. Fig. 4.3 shows a simplified architecture of the customized network, named BoneNet, with the new basic blocks in the convolution blocks marked by the orange dashed line. The new connection is highlighted with the orange arrow. The rest of the network is identical to the original PVNet architecture. The convolution blocks (left hand side of the dark dashed line in Fig. 4.3) learn and optimize network parameters for image feature extraction. The interpolation layers (right hand side of the dark dashed line in Fig. 4.3) propagate the extracted features and reconstruct original dimensions for the outputs.

BoneNet is trained with a dataset that contains X-ray projections of the bone, corresponding bone binary masks, and 2D ground-truth coordinates of the bone's landmarks. Landmark 2D coordinates are then converted to 2D vector fields as in [**?**]. Fig. 4.4 shows samples of the BoneNet training dataset with 2D landmarks of the femur and the tibia marked by white crosses (a), a ground-truth femur segment (b), and the



(a) Projection **(b)** Ground truth segment **(c)** Ground truth vector field (xy)

Fig. 4.4 Visualization of an input projection (a) with the femur's and tibia's 2D landmarks, a femur segment (b), and the vector field (blue arrows) of a landmark (orange) (c).

corresponding vector field of a landmark (orange star) under the vector form (blue arrow) (c). Like PVNet, BoneNet predicts the bone segment and a voting vector field for each landmark of a given input image. The exact coordinates of each landmark are computed from its voting vector field using the voting scheme described in [**?**]. A set of pixel hypotheses is voted for each landmark with corresponding voting scores. Each landmark is then represented by the weighted mean of its hypothesis coordinates $\hat{\mu}$, and a coordinate covariance σ computed as weighted mean squared Euclidean distances between the hypotheses and the mean coordinates $\hat{\mu}$. The Mahalanobis weight ω of the corresponding landmark in Eq. (4.1) is penalized with the inverse of the covariance σ as a higher σ represents a less accurate estimation of the corresponding landmark [**?**]. In general, a training dataset contains input images $\mathcal{I}(x, y)$ with the ground-truth bone segments $\mathcal{M}_{gt}$, and the ground-truth 2D landmark coordinates (x_{gt}, y_{gt}). The learning loss is composed of smooth $\mathcal{L}1$ and cross entropy loss $\ell(\cdot)$ for vector field and segment training, respectively [**?**]. The smooth $\mathcal{L}1$ loss is computed as the differences between the 2D predicted $f\left(\mathcal{I}(x, y), \boldsymbol{\omega_c}\right)$ and the ground-truth vector fields. The cross entropy loss $\ell(\cdot)$ is computed from the predicted segments $g\left(\mathcal{I}(x, y), \boldsymbol{\omega_m}\right)$ and the ground-truth segments. BoneNet then optimizes the parameters $(\boldsymbol{\omega_c}, \boldsymbol{\omega_m})$ to minimize the learning cost $\mathcal{L}(\boldsymbol{\omega_c}, \boldsymbol{\omega_m})$ (Eq. (4.5)).

$$\mathcal{L}(\boldsymbol{\omega_c}, \boldsymbol{\omega_m}) = \left\| [\mathcal{M}_{gt} \odot f\left(\mathcal{I}(x, y), \boldsymbol{\omega_c}\right)] - [\mathcal{M}_{gt} \odot \mathcal{I}(x, y) - (x_{gt}, y_{gt})] \right\|_{smoothl1} - \ell\left(\mathcal{M}_{gt}, g\left(\mathcal{I}(x, y), \boldsymbol{\omega_m}\right)\right) \quad (4.5)$$

with $\boldsymbol{\omega_c}, \boldsymbol{\omega_m}$ the learnable weights.

4.2.5 Simulation of articulated transformation

A training dataset comprises X-ray radiographs of the bone, the 2D ground-truth bone segments that contain the landmarks, and the 2D ground-truth coordinates of the landmarks. Training BoneNet requires an extensive, well-labeled dataset, which must be diverse in terms of the landmark relative positions and orientations in the image plane. X-ray images can be simulated from 3D CT volumes of the animal using the ASTRA Toolbox volumetric projector [**?** **?**].

In principle, one could manually manipulate the joint configuration of the animal sample for every 3D CT scan to generate realistic representations of the animal articulation poses. However, the scanning procedure is time-intensive as a large number of CT scans is needed to cover possible joint configurations. Additionally, the 3D landmarks extracted from each 3D CT scan are inconsistent across the scans due to changes in the object's orientation and position with respect to the scanning volume geometry. A simulation of both rigid and articulated transformations of the animal sample can facilitate this manual procedure. It also maintains the mapping of the 3D landmark coordinates throughout the 3D models as they can be computed with respect to the transformation parameters. In this work, a 3D CT volume of a piglet hindlimb acquired with a high-quality X-ray imaging system, FlexCT [**?**], is used as the base model for the simulation. The 3D model is with a size of $1\,416 \times 1\,416 \times 416$ voxels, and voxel size of $45\mu m$. It was then downscaled to the size of $850 \times 850 \times 250$ voxels for a more efficient data processing. Then, the rigid transformation of the object with respect to the acquisition geometry is simulated using the ASTRA toolbox vector geometry [**?** **?**].

Finally, in the articulation transformations, the voxels in the joint areas might undergo more than one affine transformation as the result of consecutive rotations of individual bones relative to the joint local coordinate systems. The resulting transformation is modelled as a weighted fusion of the separate rotations. The weights $\omega_f(x)$ are obtained as a convolution of a 3D Gaussian kernel with a standard deviation σ_f and width of k_f sampling rate with the segment volumes of individual bones (Eq. (4.6)).

$$\omega_f(x) = g(\sigma_f, k_f) * \mathcal{V}(x) \tag{4.6}$$

The 3D bone segments are obtained by the following morphological operations in Matlab [**?**]. First, Otsu threshold is applied to remove soft

tissues from the original 3D CT model of the limb. Next, small segments with a few voxels are excluded. Only the segments of the bones of interest are retained. Finally, a morphological closing scheme (a dilation followed by an erosion of $25 \times 25 \times 25$ structuring element window) is applied to fill the empty holes inside each segment. The segments are then labeled with a 3D 6-connected component technique.

The Gaussian weights are used in a fuzzy polyaffine fusion scheme introduced by Arsigny et al. [**?**] to combine individual transformations that occur in a small interval of time $1/S$ with S the fusion time scale. Affine transformation of an individual bone includes rotations about its local coordinate system. Principal component analysis (PCA) of non-weighted voxel coordinates is used to define the bone local coordinate system. Three orthogonal eigen vectors $(\hat{e_x}, \hat{e_y}, \hat{e_z})$ represent three bone principal axes, namely the vertical x^{j_i}, longitudinal y^{j_i}, and transverse axis z^{j_i} (Fig. 4.2). A bone origin is then defined by sliding its C along the major semi-axis by the axis length, followed by a visual verification to ensure the origin is at the expected end of the bone. Given a rotation matrix $\mathcal{R}_t$, $t = 1 \dots m$, with m the number of rotations, rotation angle α_t is computed by:

$$\alpha_t = \arccos\left(\frac{tr(\mathcal{R}_t) - 1}{2}\right) \tag{4.7}$$

with $tr(\mathcal{R}_t)$ the trace of $\mathcal{R}_t$. Arsigny et al. [**?**] defined the transformation speed $\mathcal{A}_t$ of rotation $\mathcal{R}_t$ as $\mathcal{A}_t = \log(\mathcal{R}_t)$, with $\log(\mathcal{R}_t)$ computed by:

$$\log(\mathcal{R}_t) = \begin{cases} 0 & \text{if } \alpha_t = 0 \\ \frac{\alpha_t}{2\sin\alpha_t}(\mathcal{R}_t - \mathcal{R}_t^T) & \text{if } \alpha_t \neq 0 \text{ and } \alpha_t \in (-\pi, \pi) \end{cases} \tag{4.8}$$

The 2^{nd}-order scheme [**?**] that computes fusion of m individual transformations $\mathcal{R}_t$ occur in the time interval $1/S$ is simplified to:

$$\mathcal{T}_2^{1/S}(x) = x + \frac{\sum_t^m \omega_{f_t}(x)\left(e^{\mathcal{A}_t/S} - I\right)x}{\sum_t^m \omega_{f_t}(x)} \tag{4.9}$$

with x the object voxel coordinate, $w_{f_t}(x)$ the fusion weight applied to transformation t^{th} of the voxel x, and I the 3×3 identity matrix.

Finally, polyaffine transformation of x at k^{th} point in time is obtained by taking compositions ($\circ$) of k sub-transformations $\mathcal{T}_2^{1/S}(x)$ (Eq. (4.10)).

$$\mathcal{T}_2^{k/S}(x_k) = \underbrace{\mathcal{T}_2^{1/S}(x_{k-1}) \circ \cdots \circ \mathcal{T}_2^{1/S}(x_0)}_{k \text{ compositions}} \tag{4.10}$$

with $1 \leq k \leq S$, and x_0 the initial position of x.
Inverse transformation fusion can be obtained by simply taking the opposite of rotation angle α_t. Target voxels are then mapped to source voxels by applying the inverse warping model. As the mapped source voxels are usually non-integer-coordinate-voxels, an interpolation scheme is needed to derive the target voxel intensities afterward. In this work, a 3D cubic-spline interpolation tool is implemented that fits a 3^{rd}-order polynomial to the known integer neighboring voxels of an unknown floating voxel to compute its intensity [**?**]. The method is deployed on a GPU infrastructure to increase computational performance as the interpolation is voxel-wise, and a volume usually contains millions of voxels.

4.3 Experiments and results

4.3.1 Training data

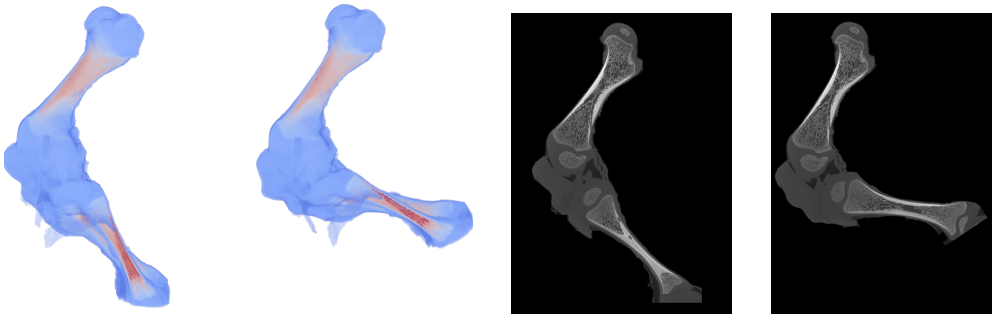
A large dataset is needed to train BoneNet. The dataset must contain the X-ray images of the hindlimb in different configurations of the bones as well as various limb's positions and orientations with reference to the 2D image space. In the following experiments, all simulation data was generated from a single 3D CT model of a piglet hindlimb sample with muscle removed by dissection. The articulation poses of the limb are simulated using the fuzzy polyaffine fusion scheme discussed in the Section 4.2.5 with a fusion scale S of 18. Fusion weights $\omega_f(x)$ were computed by the convolution of a Gaussian kernel with $\sigma_f = 13$ and width k_f of 23 with the bone segments.

(a) Femur (pastel) and tibia (red) segments.

(b) Gaussian weight maps.

Fig. 4.5 The femur and the tibia segments (a) of a piglet hindlimb and their Gaussian weight maps (b).

The femur (pastel) and tibia (red) segment of the piglet hindlimb are shown in Fig. 4.5a. Their Gaussian weight maps (Fig. 4.5b) were normalized for fusion of the polyaffine transformations of the individual bones. This allows deformation of the 3D CT model of a piglet hindlimb (Fig. 4.6a) via two rotations around the femur and tibia transverse axes (Fig. 4.6b). As shown in Fig. 4.6, the transformed slice structure Fig. 4.6d is similar to the source slice Fig. 4.6c. It must be noted that the two slices are not exactly corresponding as the femur and the tibia are rotated around their principal transverse axes, and these axes are not parallel to the volume axis. The smooth transition in the femur-tibia joint area (Fig. 4.6d) demonstrates that our polyaffine fusion and tricubic interpolation can be used for further simulation of the articulated motions of the limb.
To cover possible poses of the limb, the femur and the tibia were rotated around their transverse axes with six and five equally spaced angles in the range from $-30°$ to $35°$ and from $-20°$ to $35°$, respectively. In total, 30 polyaffine transformed volumes of different articulated poses of the limb were generated. Fig. 4.7 shows 20 bounding (green) and 20 SIFT (orange) landmarks extracted for a piglet femur by applying the shortest coordinate variance scheme presented in Section 4.2.3. The number of landmarks was chosen heuristically to be 20 for each bone with λ adjusted to $(2.1, 3.2)$ and $(2.4, 4.2)$ for the 3D bounding and SIFT landmark detection of the femur and tibia, respectively. As shown in Fig. 4.7, the landmarks are easily distinguished and distributed close to the surface of the bone. The 3D bounding and SIFT landmarks of the femur and the



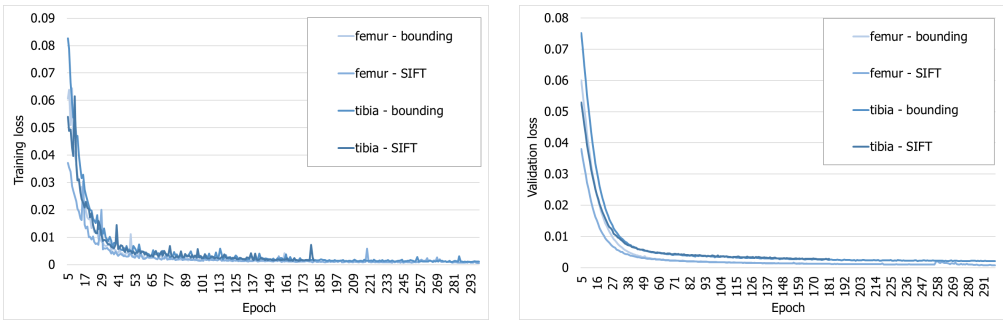
(a) Original volume. **(b)** Transformed volume. **(c)** Original slice. **(d)** Transformed slice.

Fig. 4.6 Original volume (a) and a the polyaffine transformation (b) for rotations of the femur and tibia around their principal transverse axes, with two slices from the original (c) and polyaffine (d) transformed volume. The two slices are not aligned as the transverse axes are not parallel to the volume coordinate axis.

Fig. 4.7 The bounding (green) and SIFT (orange) landmarks of a bone with random bone voxels in blue. The landmarks are distinguishable and at a distinctive distance from each other.

tibia were also transformed to obtain the corresponding coordinates in the deformed volumes. These volumes were then used for simulation of the 2D X-ray radiographs of the limb with the following scheme.

Rigid positions and orientations of the whole limb with reference to the 2D image space were simulated using the ASTRA Toolbox vector geometry [**? ?**]. Forty angle intervals were equally sampled from two ranges between $-30°$ and $30°$ and between $150°$ and $210°$, which replicate the projection angles of a practical acquisition. With each of these angle intervals of $\pm 1.5°$, 13 X-ray projections were generated using the ASTRA volumetric projector whose vector geometry is computed with the distances SOD, SDD of $6\,550 \pm 180$ and $10\,000 \pm 540$ voxels, respectively. The



(a) Training losses. **(b)** Validation losses.

Fig. 4.8 Evolution of training and validation losses over training epochs. All the losses decrease stably and plateau at the final epochs.

limb 3D positions (x^o, y^o, z^o) along the horizontal, vertical and projection axis were modified with ± 180, ± 120, and ± 180 voxel units, respectively. The rigid rotations around two horizontal axes were adjusted in the range of $\pm 15°$, and binning factor is set to 6. The 2D projections of the bounding and SIFT landmarks as well as the 2D masks of the corresponding bone segments were also computed using the same geometry and volumetric projector. Additionally, each noiseless projection was scaled to a maximum intensity I_0 randomly generated in a range of $2\,000 \pm 350$ to diversify the noise level in the simulated dataset. Then, Poisson distributed X-ray projections were simulated by replacing each projection pixel by a random draw from a Poisson distribution with a mean corresponding to the noiseless projection pixel value. In total, the generated dataset contains $15\,600$ simulated X-ray radiographs of the piglet limb with ground-truth masks of the bones and the 2D ground-truth coordinates of the 20 bounding and 20 SIFT landmarks of the corresponding bones. The dataset was then shuffled and divided into training, validation, and test sets of $11\,700$, $3\,120$, and 780 samples, respectively. This test set was used to examine the prediction loss after the training completed.

4.3.2 Train BoneNet

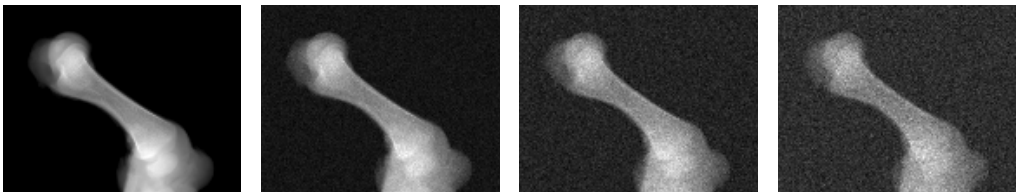
To find out whether the customized BoneNet is capable of predicting accurate 2D landmarks in X-ray radiographs, it was trained, validated, and tested on a simulated dataset. BoneNet was trained with a maximum of 600 epochs or until the training and validation losses plateau. Like

PVNet [**?**], the adam optimizer [**?**] minimizes the smooth L1 loss, which is equivalent to the Huber loss [**?**], and cross entropy loss (chapter 9, Murphy 2012 [**?**]) for the vector fields and object segment learning, respectively. A multistep learning rate scheduler [**?**] that adjusts the base learning rate of 10^{-5} by a multiplication rate of 0.5^{e} (e the current epoch) was applied for the first five training epochs. Four models were trained individually for 20 bounding and SIFT landmarks of the femur and the tibia. As can be seen in Fig. 4.8a, the training losses descend rapidly over the first 30 epochs and steadily decrease over the rest of the training. The validation losses (Fig. 4.8b) were computed for $3\,120$ samples of the corresponding dataset. Although intermittent spikes of the losses throughout the training epochs can be observed, overall, both the training and validation losses plateau over the last epochs. The models also do not overfit to the training data as both the losses gradually and stably descend. This is further demonstrated in the numerical evaluation of the 2D landmark detection for a test dataset in Section 4.3.3. The validation curves (Fig. 4.8b) evolve smoother in comparison to the training losses as the validation points are generally computed after training epoch backpropagations, namely after updates of the model parameters with respect to training batches.

4.3.3 2D landmarks detection

To study 2D landmark detection accuracy using the BoneNet predicted segments and voting vector fields, a numerical evaluation using a simulated dataset was performed. A study dataset was generated independently from the training set by following procedure. Using the same initial CT volume, nine articulated volumes with different femur and tibia rotations from the training set were simulated. More specifically, three femur (η^{j_1}) and tibia pitches (η^{j_2}) were equally sampled from two ranges between $-19°$ and $30°$ and between $-16°$ and $30°$, respectively. The X-ray images were also generated with a different sampling rate of eight $\pm 7.5°$-angle-intervals in the ranges from $-30°$ to $30°$ and from $150°$ to $210°$. The distances SOD, SDD were also manipulated with $6\,450 \pm 180$ and $10\,200 \pm 540$ voxels, respectively. The other rigid parameters including $\{x^o, y^o, z^o\}$, $\{\theta^o, \phi^o, \eta^o\}$ were randomly sampled in the same ranges with the training set.

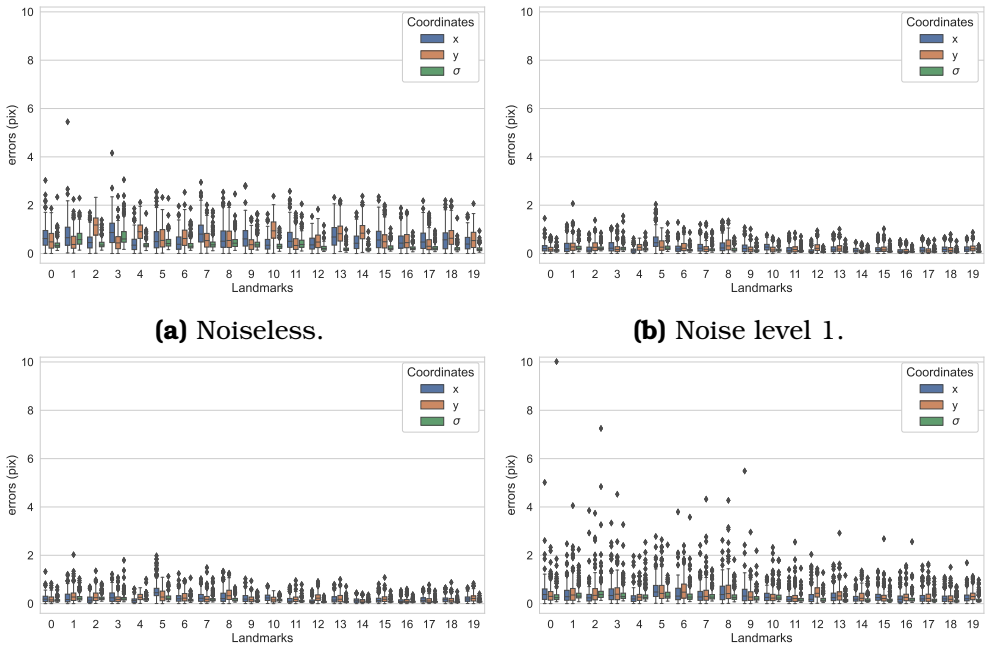
In the first experiment, the noise sensitivity of the BoneNet model that was



(a) Noiseless. **(b)** Noise level 1. **(c)** Noise level 2. **(d)** Noise level 3.

Fig. 4.9 Four sample projections from each of the simulated datasets with noiseless (a), low (b), moderate (c), and high noise level (d).

trained with the low noise training dataset was studied. Four datasets of 200 projections were generated with the aforementioned parameters, and the ASTRA toolbox. In addition to a noiseless dataset, three different noise levels were introduced to generate datasets with noise levels 1, 2, 3 corresponding to I_0 of 2000 ± 350, 650 ± 150, and 250 ± 50, respectively. Four sample projections are shown in Fig. 4.9 to illustrate the effect of different noise levels on the projection data. Next, 2D landmark detection was performed on the four datasets with the BoneNet model that was trained with the low noise data, (I_0 of 2000 ± 350). The method presented by Peng et al. [**?**] was applied to compute the exact coordinates of each landmark based on its masked voting vector field. Each landmark is represented by a mean 2D coordinate hypothesis $\hat{\mu}$ and a covariance σ. The coordinate errors were calculated as the absolute differences between the ground-truth values, and the inferred mean hypotheses $\hat{\mu}$ for each landmark. The landmark detection errors are summarized in Fig. 4.10. Since the BoneNet model was trained with a dataset of noise level 1, following discussion will use the results obtained for noisy dataset 1 (Fig. 4.10b) as a base line to assess the 2D landmark detection errors. The 2D landmarks in the noiseless dataset were estimated less accurately in comparison to the three noisy datasets as 75% of the samples are estimated with the errors up to 1.4 pixels (upper bars of the blue/orange boxes in Fig. 4.10a). More specifically, for the noise levels 1 and 2, third-quartile error levels of around 0.6 pixels were obtained, while these approximate 0.8 pixels for the level 3 dataset. The higher errors for the noiseless dataset are likely caused by the absence of noiseless samples in the training data. That is, during a training epoch, the forward evaluation of the network learning function (Eq. (4.5)) was computed using the noisy



(a) Noiseless. **(b)** Noise level 1.

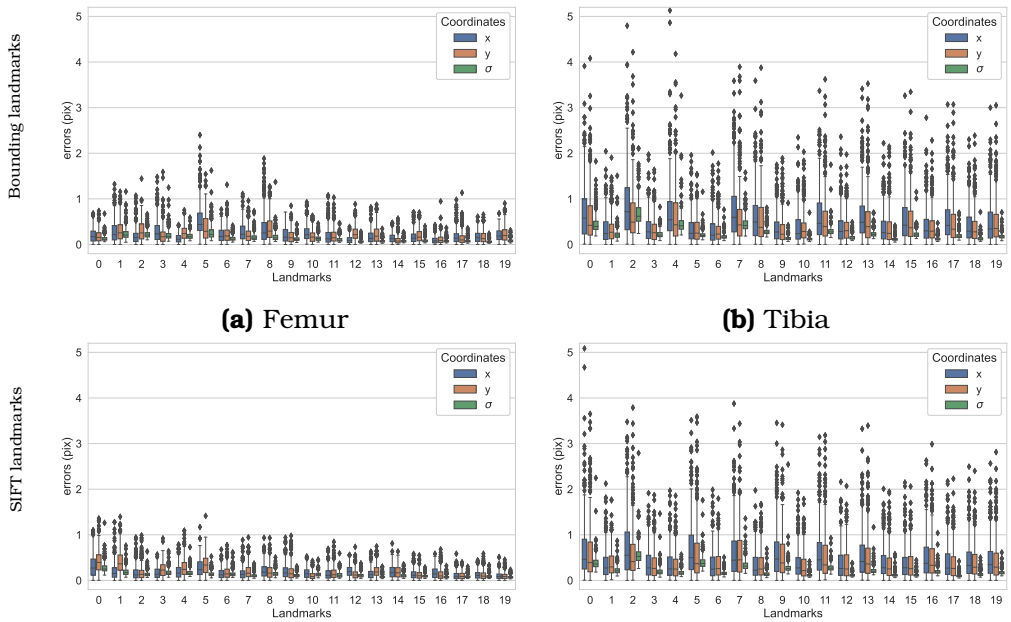
(c) Noise level 2. **(d)** Noise level 3.

Fig. 4.10 2D landmark extraction errors (x,y) and the covariance σ for four datasets with different noise levels using BoneNet trained with a dataset of noise level 1. With a higher third-quartile of around 1.4 pixels, the noiseless dataset was estimated less accurately than the noisy datasets as they have the third-quartiles of less than 0.8 pixels. However, there are higher numbers of outliers in the results of noise level 3 (black diamond points in d) in comparison to the other three datasets (black diamond points in a,b,c).

data. The learnable network parameters were then updated through the back-propagation process to derive the output feature vectors that best describe landmark positions in a noisy scene. When the trained network was used to infer a landmark in a noiseless image, the output feature map was computed using the same learned parameters. Therefore, it is possible that the feature vector is not mapped correctly to the expected position of the landmark. The relatively low inference accuracy of a deep neural network (trained with noisy data) on a noiseless or less noisy testing dataset has also been reported in other studies [**? ? ?**]. More experiments are needed to analyze BoneNet's performance on noiseless data and data with different noise levels in both training and testing dataset. This experiment also demonstrates that, although having a relatively higher noise level (Fig. 4.9d) compared to the level applied to

the training dataset, the BoneNet model is still capable of detecting 2D landmarks at noise level 3, albeit with a slightly reduced accuracy. We also observe outliers with larger coordinate inaccuracies for noise level 3 as the error levels are up to 10 pixels, and more number of landmarks detected with errors of around and above 4 pixels (black diamond points in Fig. 4.10d) in comparison to the results for noise levels 2 and 3 (black diamond points in Fig. 4.10b, 4.10c). However, the results generally indicate that, if BoneNet is trained with a similar noise level to the testing or real data, the model would be robust to noise, and could tolerate a broad range of noise levels. Furthermore, adding noise to the training data is also considered as a data augmentation technique that could reduce overfitting, and help the model cope with noise in the real data [**?** , chapter 7], [**? ?**].

To test how accurately the landmarks were detected for the different



(a) Femur **(b)** Tibia

(c) Femur **(d)** Tibia

Fig. 4.11 Visualization of the estimated 2D landmark coordinate errors. The femur landmarks are detected with lower coordinate errors as the third quartiles and maxima are around 0.5 and 1.2 pixels and both lower than 1.1 and 2.2 pixels of the tibia landmarks.

bones and landmark types, another dataset containing 200 X-ray projections was simulated using ASTRA toolbox volumetric projector. The

four trained BoneNet models infer the landmark voting vector fields and the bone binary masks in the study X-ray radiographs for two type of bones (femur and tibia) and landmarks (SIFT and bounding). The femur's landmarks are estimated more accurately than the tibia's as the respective upper whiskers (vertical, black lines of blue/orange boxes) extend to 1.1 pixels and 2.5 pixels, and inter-quartile ranges (blue/orange box areas) are around 0.1–0.6 and 0.2–1.1 pixels (Fig. 4.11). Median coordinate errors are up to 0.3 and 0.6 pixels for the femur and tibia landmarks, respectively, demonstrating that 50% of the landmark samples are estimated lower than these errors. Although all landmarks are detected with a median error of less than 0.6 pixels, several landmarks tend to be less accurately estimated than the rest, such as the fifth of the femur bone (Fig. 4.11b). As the covariance measures (σ) are proportional to the error levels of the corresponding landmarks (blue/orange), the higher the covariance, the less confident the estimated landmark coordinates. Consequently, the less accurately detected landmarks are weighted less in the pose reconstruction cost function Eq. (4.1).

4.3.4 3D pose reconstruction

The final experiment is to study how the predicted 2D landmarks perform in 3D pose reconstruction for the study samples. The voted landmarks were used to estimate the 3D pose parameters with two joint rotations (the femur and the tibia) $\boldsymbol{\tau} = \{x^o, y^o, z^o, \theta^o, \phi^o, \eta^{j_1}, \eta^{j_2}\}$. SOD, SDD, and η^o are fixed to the ground-truth values, and all the other parameters are initialized to 0. A numerical study was performed for the reconstruction of 3D poses of the 200 simulated samples. The results are summarized in Fig. 4.12. The offsets of the limb with reference to the horizontal axis parallel to the detector plane (x^o), and the vertical axis (y^o) are estimated with median errors of around 20 voxel units indicating that errors of 50% of the samples lower than this value.
Since, the magnification factors were simulated around 1.5, the projection of a point can be 30 voxels units offset from the correct position. However, the binning factor is 6, so the offset approximates to five pixels. The limb position along the projection axis z^o is estimated with a median of 60 voxels, and 75% of the samples having z^o error of less than 125 pixels (middle and upper bars of the green boxes in Fig. 4.12a, 4.12c, respectively). If

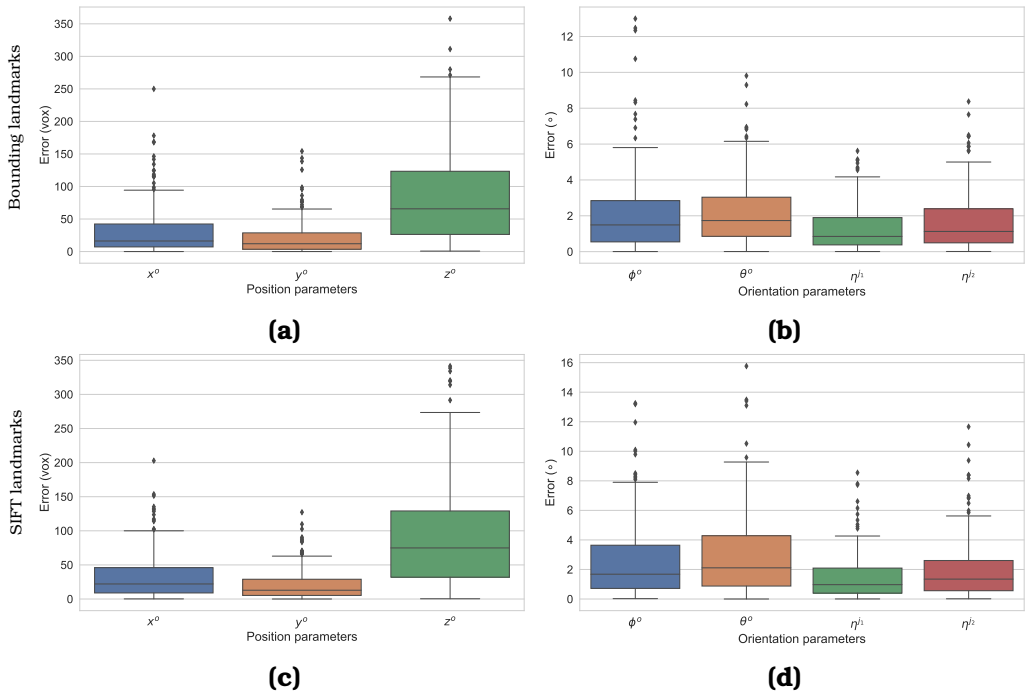
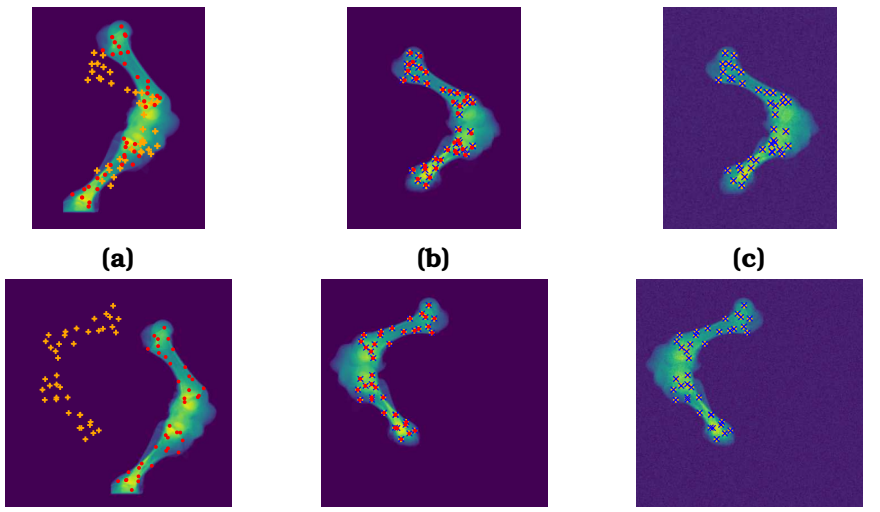


Fig. 4.12 Estimation errors of the pose parameters using the bounding (a,b) and SIFT (c,d) landmarks. The estimations using the bounding return lower error ranges in comparison to the results using SIFT landmarks, especially for the rotation parameters (b,d).

the respective simulated distances of SOD, SDD, which are $6\,450 \pm 180$ and $10\,200 \pm 540$ voxels, are accounted for, the computed error makes up around 2% of the projection magnification. Therefore, this gap is hardly visible in the projected image in terms of pixel positions. As shown in Fig. 4.12b, 4.12d, 50% of the samples are with the rigid $\{\theta^o, \phi^o\}$ and the articulated $\{\eta^{j_1}, \eta^{j_2}\}$ rotation errors below $1.9°$ and $0.9°$, respectively. The rigid rotations $\{\theta^o, \phi^o\}$ of 75% of the test samples are reconstructed more accurately using the bounding landmarks, with an error of $3°$ in comparison to $4°$ for the SIFT landmarks (upper bar of the blue/orange boxes in Fig. 4.12c, 4.12d. The articulated rotations $\{\eta^{j_1}, \eta^{j_2}\}$, have lower third quartile levels of around $2°$ using either the bounding or SIFT landmarks as demonstrated in Fig. 4.12c, 4.12d, upper bars of the green/red boxes. In general, the rotation parameters reconstructed with the bounding landmarks are more accurate as the upper whiskers and interquartile ranges are lower than the results using the SIFT landmarks (Fig. 4.12c, 4.12d).

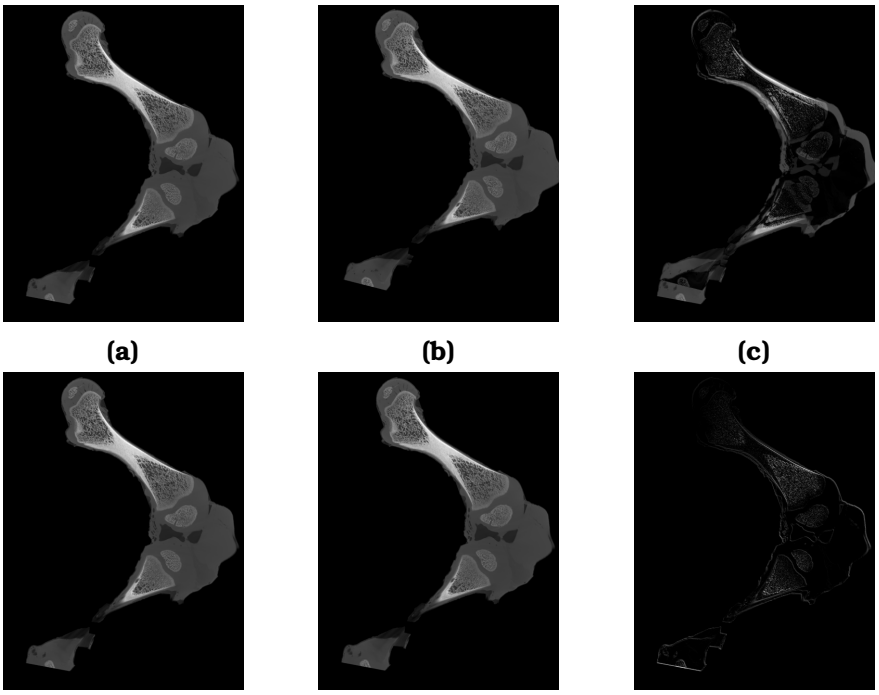
As the parameters were estimated with notable numerical errors, a further visual inspection was conducted for two typical test samples whose errors situated in the upper (high error), and lower (low error) whisker areas of Fig. 4.12. The results associated to the high and low error sample are shown in the first and second row of Fig. 4.13–4.14, respectively. First, 2D views of initial (a,c), reconstructed (b,d), and ground-truth (c,f) poses for the two test samples are visualized in Fig. 4.13. The ground-truth, detected, and registered landmarks are highlighted in blue, orange and red, respectively. As can be seen in the first column of Fig. 4.13, both the initial orientation and position of the limb do not match the detected landmarks (orange). After the registration (Fig. 4.13b, 4.13e), the detected landmarks (orange) are aligned with the bone and close to the reconstructed landmarks (red). While the landmarks computed with high-error parameters do not always overlap the detected and ground-truth landmarks (Fig. 4.13b), the low-error computed landmarks are well-aligned with the detected ones (Fig. 4.13e). The inaccurate parameters also pose a visible gap in the tibia projection between Fig. 4.13b and Fig. 4.13c. With well reconstructed parameters, no difference can be seen in the estimated and ground-truth projections shown in Fig. 4.13e and Fig. 4.13f.
To further inspect the visual impact of the registration errors, two slices were extracted at the same position from the ground-truth and reconstructed volumes for each of the two test samples. Registration residuals were computed between these slices and the results are summarized in Fig. 4.14. The gap is clearly visible with high magnitude of misalignment in the residual slice of the high error sample Fig. 4.14c. However, a marginal residual is observed in the accurate pose reconstructed sample Fig. 4.14f.
Finally, registration errors are shown in 3D to give an insight into the estimation of the femur and tibia rotation around their transverse axes $\{\eta^{j_1}, \eta^{j_2}\}$. The corresponding 3D views for the two testing samples are shown in Fig. 4.15 with the reference, ground-truth (target) are in blue and orange, respectively. Before 3D registration, the orientation of the limb (blue) is misaligned with the target pose (orange) (Fig. 4.15c). Surface-to-surface distances were computed between ground-truth and registered volumes Fig. 4.15b, 4.15d . The registration errors are clearly visible as hot color regions in Fig. 4.15b. Furthermore, the magnitudes of the



(d) Initial pose **(e)** Estimated pose **(f)** Ground-truth pose

Fig. 4.13 Visualization of the 2D views for the initial (a,d), estimated (b,e), and ground-truth (c,f) poses of the two test samples with predicted (orange), ground-truth (blue), and reconstructed (red) landmarks. The 2D detected landmarks (orange) are not aligned with the computed landmarks (red) in the initial poses (a,d). After registration, the estimated and detected landmarks align with the bones. However, the reconstructed landmarks (red) of the high error sample do not always overlap the detected landmarks (orange) (b). This is not the case with the accurate registered sample as the ground-truth (blue), detected (orange), and registered (red) landmarks are aligned correctly (e).

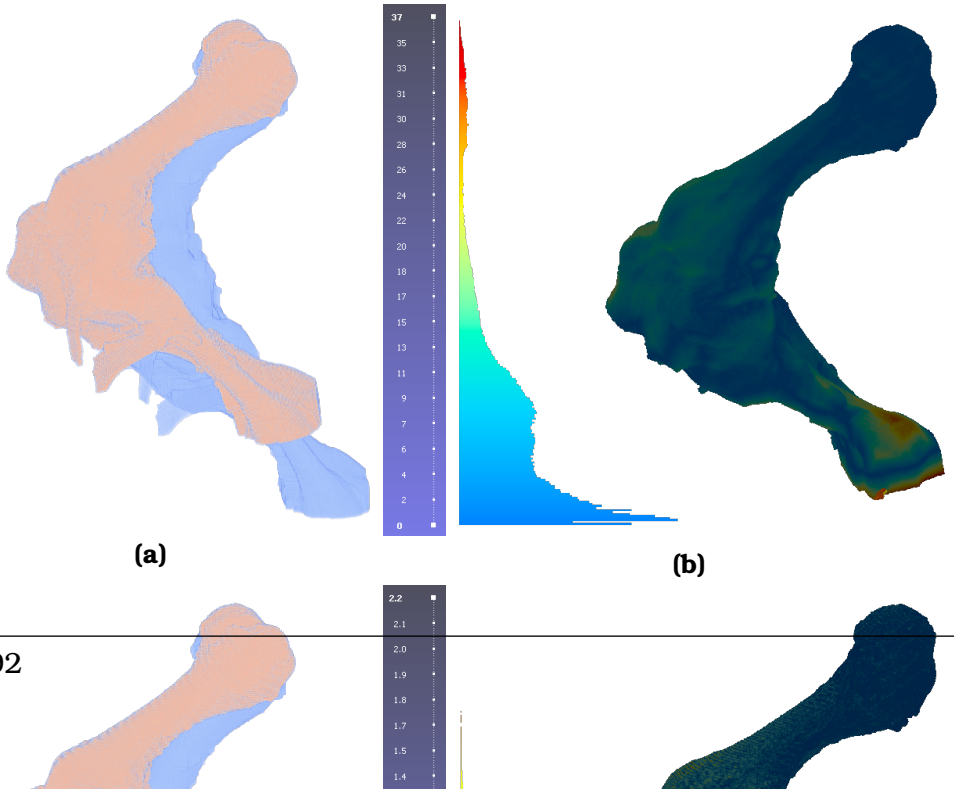
registration errors are as high as 37 voxel unit (red regions). In comparison, gaps of around 15 voxels are scattered over the surface of the bone (dark to light cyan regions in Fig. 4.15b). With an accurate estimate of the parameters, there are only marginal gaps (≈ 2 voxels) between the reconstructed and the ground-truth 3D poses marked by cyan regions in Fig. 4.15d.



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

(d) Ground-truth slice **(e)** Estimated slice **(f)** Residual slice

Fig. 4.14 Visualization of the slices extracted from the same positions in the 3D ground-truth (a,d) estimated (b,e) volumes of the two testing samples. The difference between the corresponding ground-truth and estimated slices are shown as residual images in (c,f). The residual of the high error sample (c) is more apparent with a substantial magnitude in comparison to a minor and less visible gap for the accurate registered sample (f).



4.4 Discussion

In this chapter, a comprehensive landmark-based method was introduced for 2D/3D registration to reconstruct the 3D pose of an object using its fluoroscopic X-ray image and a 3D reference model. The method aligns the 2D detected landmark positions in the X-ray image with the 2D projections of corresponding 3D landmarks. As previous 3D landmark selection methods are prone to overlapping projected landmarks, a shortest coordinate variance scheme was developed to detect the potential 3D reference landmarks. With the shortest coordinate variance threshold, the 3D landmarks were distributed over the object surface and at a distinctive distance from each other. This scheme facilitated distinguishing the 3D landmarks in the reference models as well as detecting the 2D landmarks in the 2D fluoroscopy images.
The conventional landmark extraction methods do not allow to easily map the 2D detected landmarks to the 3D reference landmarks for an accurate alignment of the object. Therefore, a deep learning method was introduced to overcome this obstacle. In general, a trained deep learning model with a well-labeled dataset can predict the positions of the 2D landmarks in a 2D X-ray radiograph. Although there are various deep learning models introduced for landmark detection and registration, a deep neural network that fits our specific object (piglet limb) and the number of landmarks to be detected was not available off the shelf. One of the most relevant models is PVNet [**?**], which was introduced to detect 2D landmarks in optical images. PVNet originally tackled occlusion in visible light photography. This model was designed to handle only nine 2D landmarks in the scenes. However, our preliminary experiments for the limb data suggested having less than 20 landmarks is insufficient to reconstruct the 3D poses of the limb using a single X-ray radiograph. Therefore, BoneNet, which is inspired by PVNet was presented, to adapt to a higher number of landmarks and a more complex biological object. By adding five more convolution basic blocks to the feature encoding stage in the original PVNet, BoneNet was capable of robustly extracting feature vectors from the X-ray imaging data and propagating the features towards upscaling layers. A shortcut from a feature encoding layer to an interpolation layer was replaced to transfer more feature vectors to the

output and derive more landmarks. The simulation results show that BoneNet was able to detect the 2D landmark positions in 2D fluoroscopy images accurately. The numerical evaluation for pose reconstruction using the detected landmarks demonstrated promising rigid and articulated parameter estimations. However, further study is needed to clarify the source of errors as well as to minimize the residual errors in both 2D landmark detection and 3D pose reconstruction.

The neural network training requires a large amount of diverse labeled data in terms of the object's positions, orientations, and articulation poses. Therefore, the polyaffine fusion scheme [**?**] was also applied for a realistic data simulation. An inverse transformation and a 3D tricubic spline interpolation module were also implemented for a smooth and continuous 3D volume transformation. This module and the ASTRA Toolbox [**?** **?**] served as a data curation tool to prepare the BoneNet training dataset, and the validation and test data to evaluate our whole registration method as the ground-truths were known. The 3D landmark positions were also computed consistently across the transformed 3D volumes by using the same transformation model and parameters.

In the scope of this thesis, only a single piglet limb from which the muscle was dissected was considered as a test object. Such simulation neglected the presence of muscle and other types of soft tissue in a real animal model that would certainly complicate the 2D landmark detection. Therefore, in future work, an evaluation of our proposed method with more complex objects, including limbs with muscles, soft tissues, and ultimately, a whole piglet model, is needed. A whole limb study would include acquiring CT scans of the limb to use as a reference model, followed by 3D landmark extraction, simulation of the 2D X-ray datasets, as well as training, and evaluation with the new data. Moreover, the current noise simulation considers neither X-ray source model nor detector responses. Although the preliminary results indicate a high robustness to noise, a further study is necessary to train and to evaluate the performance of BoneNet at the noise level of a real X-ray fluoroscopy system. Such studies are the prerequisite steps towards the evaluation of BoneNet on 2D landmark detection in real X-ray fluoroscopy radiographs. In the current implementation, a deep neural model was trained specifically for each landmark type (bounding, SIFT) and bone (femur, tibia). This

training technique is inefficient as a more complex object requires numerous models to be trained. Therefore, the current BoneNet architecture needs to be improved to learn and predict different types of landmarks and bones using a single training model. The current technique uses only a single cone-beam X-ray radiograph for 3D pose reconstruction. In the future, X-ray images from a biplanar X-ray scanner, e.g. [**?**], could be employed to gain the accuracy of the 3D pose parameter estimation as more geometric information is taken into account. The method also need to be evaluated with real X-ray fluoroscopy images for a complete reconstruction of the piglet 3D locomotion.

4.5 Conclusion

In general, the proposed method tackled the difficulties in generating a well-labeled training dataset for 2D landmark detection using a manual approach. The method employed an automated procedure to robustly detect 3D landmarks compared to the CoM-based technique [**?**]. The computed 3D landmark coordinates across the transformed volumes allowed computing the 2D landmark positions accurately for the training dataset. This procedure also eliminates human errors in manual landmark annotations. The customized PVNet architecture (BoneNet) showed stable convergences over the training with two types of landmarks and a biological sample. The inferences of the bone segments and landmark vector fields with BoneNet resulted in accurate detection of the 2D landmarks in X-ray data from which the 3D poses of the object could be accurately reconstructed.

Conclusion

In this thesis, two major problems are addressed: geometry calibration for a modular biplanar X-ray cone-beam system and 2D/3D registration in which a 3D CT image is mapped onto fluoroscopic images. For geometry calibration of a modular biplanar X-ray cone-beam system, the geometry information that is estimated or compensated based on the acquired data of an object itself is known as self-calibration. Self-calibration techniques do not perform well with complex object geometries or with objects larger than the field of view. Moreover, their iterative optimization processes are coupled with CT reconstruction that are not well-adapted to estimate a large number of parameters as in a biplanar X-ray cone-beam system. This thesis presents a simple and effective method to construct and employ a LEGO phantom to calibrate an X-ray cone-beam system. The calibration phantom allows multiple datasets sharing the same geometry calibration results for the misalignment compensation in their CT reconstructions. For 2D/3D registration using fluoroscopic images and 3D CT images, among a broad range of studies dealing with 2D/3D registration problems, several methods estimate 3D pose parameters based on the intensity profile of entire input images. In an X-ray data application, an intensity-based method faces difficulties as it usually requires simulation of X-ray images with unknown X-ray source model and spatial information loss. A landmark-based registration method could tackle the difficulties of the intensity-based method as only landmark positions are involved in the registration process.

Biplanar X-ray cone-beam geometry calibration

Geometry misalignment occurs in various X-ray acquisition systems. Therefore, it is crucial to have the geometry information estimated prior to the CT reconstructions to suppress misalignment artifacts in CT images. Geometry calibration with a LEGO phantom is easy to reproduce and customize to fit the geometry of an arbitrary X-ray system. Hence, the estimated geometry based on the scan of the calibration phantom can be used to compensate for the geometry misalignment of scans of other objects acquired with the same system configuration. Experiments with the datasets acquired with the $3D^2YMOX$ system showed significantly reduced misalignment artifacts in the CT reconstructed volumes. The main contributions to geometry calibration are as follows.

- Effortless building of a LEGO calibration phantom from off-the-shelf available materials (LEGO bricks and metal markers). Both materials are readily available and require no dedicated design and production. Both of the materials also come in various sizes and shapes. Therefore, the phantom structure can be easily customized to the target system.
- Application of ResNet50 neural network facilitates extraction of calibration marker centers and increases extraction accuracy. The ResNet50 model learns abstract features and maps marker regions of interest to accurate marker centers. By training ResNet50 with a large amount of well-labeled data, the model could generalize region of interest features and infer correct center positions of given markers.

Compared to single-source systems, biplanar X-ray CT systems allow acquiring projection data within a reduced amount of time for an extended field of view or dual X-ray energies. More geometry parameters are required to fully characterize the configuration of the $3D^2YMOX$ system, as it consists of a dual-source/detector pair. The calibration procedure was extended to a biplanar cone-beam X-ray system by exploring the effectiveness of a LEGO phantom. Simultaneous, joint estimation of 21 parameters for the dual-source/detector systems are presented. Various applications arise with the success of calibrating a biplanar X-ray CT

system, including dual-energy CT reconstruction and biplanar reconstruction by combining datasets from both X-ray systems.

2D/3D registration

The 3D musculoskeletal motion of animals is of interest in various biological studies. It can be derived from X-ray fluoroscopy acquisitions by means of image matching or manual landmark annotation and mapping. Often, a 3D model of an animal is aligned with 2D X-ray images of the same living animal to estimate the 3D pose parameters of the object in the 2D images. In this thesis, a proof of concept application was presented for a preliminary study in the reconstruction of 3D poses of an animal movement during X-ray fluoroscopy image acquisition. The main contributions of 2D/3D registration are as follows.

- A comprehensive 3D landmark extractions method was implemented with the shortest coordinate variance threshold. The 3D landmarks were distributed over the object's surface and at a distinctive distance from each other. This scheme facilitated distinguishing the 3D landmarks in the reference models as well as detecting the 2D landmarks in the 2D fluoroscopy images.

- A ResNet-based model was applied for automated 2D landmark detection on X-ray images. While the conventional landmark extraction methods do not allow to easily map the 2D detected landmarks to the 3D reference landmarks for an accurate alignment of the object, a deep learning method can overcome this obstacle. In general, a trained deep neural network model with a well-labeled dataset can infer the positions of the 2D landmarks in a 2D X-ray radiograph accurately. The original PVNet model based on ResNet18 architecture was customized for X-ray data and a custom number of landmarks to be detected.

- A realistic simulation tool of articulation transformation allows simulation of a large amount of diverse labeled data in terms of the object's positions, orientations, and articulation poses for evaluation and deep network training. An inverse transformation and a 3D tricubic spline interpolation module were also implemented

for a smooth and continuous 3D volume transformation. The 3D landmark positions can be computed consistently across the transformed 3D volumes by using the same transformation model and parameters. This procedure also eliminates human errors in manual landmark annotations.

Although significant misalignment artifact reduction can be seen in the CT reconstructions using the calibrated geometry parameters, there are residual artifacts in the reconstruction images. Further studies are required to address the source of these artifacts, as well as suppress them in the CT volumes. Additionally, calibration marker centers are not directly inferred from the marker region of interest with the current approach as BeadNet only inferred the center offsets of the marker centers. The offsets are later used to correct for the extracted marker coordinates obtained from the NCC method. Therefore, an end-to-end method is necessary to facilitate the center extraction process as well as increase extraction accuracies. As a preliminary application, ResNet50 is employed to learn and infer the center offsets of the calibration markers. However, applying a shallower variant of the ResNet architecture could also be possible. Further evaluations need to be done to find a successful model candidate in terms of training and inference performance.
The numerical evaluation for pose reconstruction using the detected landmarks demonstrated promising rigid and articulated parameter estimations in the proposed 2D/3D registration method. However, further studies are necessary to clarify the source of errors as well as to minimize the residual errors in both 2D landmark detection and 3D pose reconstruction. In the scope of this thesis, a single muscle-dissected piglet limb was considered a test object. For future work, an evaluation with more complex objects, including limbs with muscle and, ultimately, a whole piglet model, is needed. On top of that, in the current implementation, a deep neural model is trained specifically for each landmark type (bounding, SIFT) and bone (femur, tibia). This training technique is inefficient as a more complex object requires numerous models to be trained. Therefore, improving the current BoneNet architecture is essential to learn and predict different landmark types and bones by a single training model. Furthermore, the current technique uses only a single cone-beam X-ray radiograph for 3D pose reconstruction. In the

future, X-ray images from a biplanar X-ray scanner can be employed to gain the accuracy of the 3D pose parameter estimation as more geometric information is taken into account. Eventually, evaluation of the method with real X-ray fluoroscopy images completes the reconstruction of the piglet 3D locomotion.

Future work

This thesis presented a phantom-based calibration procedure to estimate the geometry of a highly modular, biplanar X-ray CT system. With a successful calibration of the system geometry, two CT volumes acquired in a biplanar X-ray system are automatically registered. The biplanar CT application creates a possibility for dual-energy CT reconstructions of a single object. Such application will allow to study objects composed of multiple materials with substantially different X-ray attenuation coefficients, as it is very challenging to obtain the detailed internal structures for these kinds of objects by a single source energy acquisition. In practice, first, each X-ray source spectrum must be calibrated in order to perform a dual-energy acquisition with the biplanar X-ray CT system. Next, based on the X-ray attenuation coefficients of the materials in the study object, different source energies are applied to obtain respective high contrast images for each of the target materials. Finally, a high quality CT reconstruction is a prerequisite requirement for studying object internal structure, therefore, common CT artifacts such as ring artifact or residual misalignment artifact must be corrected for.

The implementation of a proof-of-concept for 2D/3D registration for the pig limb in this thesis could be extended to perform 4D CT for a pig locomotion reconstruction. Eventually, it could also be applied to the different kinds of animals. Furthermore, if two CT models are simultaneously acquired by a biplanar X-ray system with a dual-energy setting, they are auto-registered. The model with a detailed skeleton of an animal could be used to reconstruct 3D motion of the animal from a fluoroscopy videography. Then, the 3D CT model with better soft tissue contrast structures could be transformed with respect to the reconstructed motions of the skeleton model. As a result, it is possible to obtain 4D CT reconstruction of both the skeleton and muscle/soft tissues of an animal

through the 2D/3D registration and biplanar X-ray CT. This data could be used to study the motion patterns of animals, as well as the development of different types of animals' muscles/soft tissues over time. The challenges of this application lie on three major issues. First, a complete articulated transformation of the whole animal anatomy must be derived from the animal skeleton 3D model. Next, a deep learning model must be customized and trained for a large number of bones and geometrical landmarks inference. The current BoneNet model can be applied to multiple bones and their landmarks, however, further experiments are needed to find its maximum capabilities as well as to customize for an optimal network architecture for a whole skeleton's landmark detection. Finally, the deep neural network requires a realistic and a large amount of data for the training process, therefore, the articulation simulation method applied in this thesis must be accommodated for more number of joints, and a complex whole animal body articulation. A realistic dataset for such training also requires simulating soft-tissues deformation with respect to the motion of the bones in the animal body. Solving all three problems simultaneously could pose difficulties in debugging when any experiments fail. Therefore, one could take several steps toward the final goal. First, performing a 2D/3D registration with a simulation data of an animal's limb with muscles intact. Then, successful experiments in this stage could lead to a next study with motion reconstruction of a real limb in a real X-ray fluoroscopic video. In the next stage, simulation of several limbs or even a whole animal's body can be taken into account before performing final experiment with the whole animal's body motion reconstruction in the real X-ray fluoroscopic images.

List of publications

Journal articles

1. **Nguyen V**, De Beenhouwer J, Sanctorum JG, Van Wassenbergh S, Bazrafkan S, Dirckx JJJ, Sijbers J, "A low-cost geometry calibration procedure for a modular cone-beam X-ray CT system". *Nondestructive Testing and Evaluation* 2020;35:3, 252–265. doi:10.1080/1058-9759.2020.1774580.

2. **Nguyen V**, Sanctorum JG, Van Van Wassenbergh S, Dirckx JJJ, Sijbers J, De Beenhouwer J, "Geometry Calibration of a Modular Stereo Cone-Beam X-ray CT System". *Journal of Imaging*. 2021 Mar; 7(3): 54. doi:10.3390/jimaging7030054.

3. **Nguyen V**, Alves Pereira LF, Liang Z, Mielke F, Van Houtte J, Sijbers J, De Beenhouwer J, "Automatic landmark detection and mapping for 2D/3D registration with BoneNet". *Frontiers in Veterinary Science Journal* doi: http://dx.doi.org/10.3389/fvets.2022.923449.

4. Sanctorum JG, Van Wassenbergh S, **Nguyen V**, De Beenhouwer J, Sijbers J, Dirckx JJJ, "Projection-angle-dependent distortion correction in high-speed image-intensifier-based x-ray computed tomography", *Meas Sci Technol.* 2021;32:3, 035404. doi:10.1088/1361-6501/ABB33E.

5. Sanctorum JG, Van Wassenbergh S, **Nguyen V**, De Beenhouwer J, Sijbers J, Dirckx JJJ, "Extended imaging volume in cone-beam x-ray tomography using the weighted simultaneous iterative reconstruction technique", *Phys Med Biol.* 2021;66:16, 165008. doi:10.1088/-1361-6560/ac16bc.

Conference proceedings

1. **Nguyen V**, De Beenhouwer J, Sanctorum J, Van Wassenbergh S, Aerts P, Dirckx J J J, and Sijbers J, "A low-cost and easy-to-use phantom for cone-beam geometry calibration of a tomographic X-ray system", in *9th Conference on Industrial Computed Tomography*, Padova, Italy, 2019.

2. **Nguyen V**, De Beenhouwer J, Bazrafkan S, Hoang A-T, Van Wassenbergh S, and Sijbers J, "BeadNet: a network for automated spherical marker detection in radiographs for geometry calibration", in *6th International Conference on Image Formation in X-Ray Computed Tomography*, 2020.

3. Sanctorum JG, Van Wassenbergh S, **Nguyen V**, De Beenhouwer J, Sijbers J, Dirckx JJJ, "Projection-angle-dependent image intensifier distortion correction in high-speed tomography", in *6th International Conference on Image Formation in X-Ray Computed Tomography*, 2020.