Université de Montréal

Analysis of 3D Human Gait Reconstructed with a Depth Camera and Mirrors

par

Trong Nguyen Nguyen

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Août 2019

Université de Montréal
Faculté des arts et des sciences

Cette thése intitulée:

Analysis of 3D Human Gait Reconstructed with a Depth Camera and Mirrors

présentée par

Trong Nguyen Nguyen

a été évaluée par un jury composé des personnes suivantes:

Sébastien Roy,	président-rapporteur
Jean Meunier,	directeur de recherche
Huu Hung Huynh,	codirecteur
Max Mignotte,	membre du jury
Alexandra Branzan Albu,	examinateur externe
Sébastien Roy,	représentant du doyen de la FES

Novembre 2019

Résumé

L'évaluation de la démarche humaine est l'une des composantes essentielles dans les soins de santé. Les systèmes à base de marqueurs avec plusieurs caméras sont largement utilisés pour faire cette analyse. Cependant, ces systèmes nécessitent généralement des équipements spécifiques à prix élevé et/ou des moyens de calcul intensif. Afin de réduire le coût de ces dispositifs, nous nous concentrons sur un système d'analyse de la marche qui utilise une seule caméra de profondeur. Le principe de notre travail est similaire aux systèmes multi-caméras, mais l'ensemble de caméras est remplacé par un seul capteur de profondeur et des miroirs. Chaque miroir dans notre configuration joue le rôle d'une caméra qui capture la scène sous un point de vue différent. Puisque nous n'utilisons qu'une seule caméra, il est ainsi possible d'éviter l'étape de synchronisation et également de réduire le coût de l'appareillage.

Notre thèse peut être divisée en deux sections: reconstruction 3D et analyse de la marche. Le résultat de la première section est utilisé comme entrée de la seconde. Notre système pour la reconstruction 3D est constitué d'une caméra de profondeur et deux miroirs. Deux types de capteurs de profondeur, qui se distinguent sur la base du mécanisme d'estimation de profondeur, ont été utilisés dans nos travaux. Avec la technique de lumière structurée (SL) intégrée dans le capteur Kinect 1, nous effectuons la reconstruction 3D à partir des principes de l'optique géométrique. Pour augmenter le niveau des détails du modèle reconstruit en 3D, la Kinect 2 qui estime la profondeur par temps de vol (ToF), est ensuite utilisée pour l'acquisition d'images. Cependant, en raison de réflections multiples sur les miroirs, il se produit une distorsion de la profondeur dans notre système. Nous proposons donc une approche simple pour réduire cette distorsion avant d'appliquer les techniques d'optique géométrique pour reconstruire un nuage de points de l'objet 3D.

Pour l'analyse de la démarche, nous proposons diverses alternatives centrées sur la normalité de la marche et la mesure de sa symétrie. Cela devrait être utile lors de traitements cliniques pour évaluer, par exemple, la récupération du patient après une intervention chirurgicale. Ces méthodes se composent d'approches avec ou sans modèle qui ont des inconvénients et avantages différents. Dans cette thèse, nous présentons 3 méthodes qui traitent directement les nuages de points reconstruits dans la section précédente. La première utilise la corrélation croisée des demi-corps gauche et droit pour évaluer la symétrie de la démarche, tandis que les deux autres methodes utilisent des autoencodeurs issus de l'apprentissage profond pour mesurer la normalité de la démarche.

Mots-clés: *optique géométrique, distorsion de profondeur, creusage de l'espace, nuage de points, miroir, Kinect, normalité de la démarche, symétrie de la démarche, modèle de démarche, adverse, auto-encodeur, histogramme cylindrique, corrélation croisée.*

Abstract

The problem of assessing human gaits has received a great attention in the literature since gait analysis is one of key components in healthcare. Marker-based and multi-camera systems are widely employed to deal with this problem. However, such systems usually require specific equipments with high price and/or high computational cost. In order to reduce the cost of devices, we focus on a system of gait analysis which employs only one depth sensor. The principle of our work is similar to multi-camera systems, but the collection of cameras is replaced by one depth sensor and mirrors. Each mirror in our setup plays the role of a camera which captures the scene at a different viewpoint. Since we use only one camera, the step of synchronization can thus be avoided and the cost of devices is also reduced.

Our studies can be separated into two categories: 3D reconstruction and gait analysis. The result of the former category is used as the input of the latter one. Our system for 3D reconstruction is built with a depth camera and two mirrors. Two types of depth sensor, which are distinguished based on the scheme of depth estimation, have been employed in our works. With the structured light (SL) technique integrated into the Kinect 1, we perform the 3D reconstruction based on geometrical optics. In order to increase the level of details of the 3D reconstructed model, the Kinect 2 with time-of-flight (ToF) depth measurement is used for image acquisition instead of the previous generation. However, due to multiple reflections on the mirrors, depth distortion occurs in our setup. We thus propose a simple approach for reducing such distortion before applying geometrical optics to reconstruct a point cloud of the 3D object.

For the task of gait analysis, we propose various alternative approaches focusing on the problem of gait normality/symmetry measurement. They are expected to be useful for clinical treatments such as monitoring patient's recovery after surgery. These methods consist of model-free and model-based approaches that have different cons and pros. In this dissertation, we present 3 methods that directly process point clouds reconstructed from the previous work. The first one uses cross-correlation of left and right half-bodies to assess gait symmetry while the other ones employ deep auto-encoders to measure gait normality.

Keywords: *geometrical optics, depth distortion, space carving, point cloud, mirror, Kinect, gait normality, gait symmetry, gait model, adversarial, auto-encoder, cylindrical histogram, cross-correlation.*

Acknowledgements

First of all, I would like to express my gratitude to Prof. Jean Meunier, my supervisor, for giving me the best support during the 4-year period of my Ph.D. program. I might not finish the project in-time without his suggestions and/or corrections for our experiments as well as paper submission/revision, and machine provision for diving into expensive studies even when some works were beyond the subject of my Ph.D. project.

I am grateful to Prof. Sébastien Roy for the useful discussion regarding to the ToF distortion that plays an important role in the success of my work. Besides, I have also benefited a lot from his interesting course on 3D Vision (IFT6145) that was very useful for the 3D reconstruction in my Ph.D. project.

I am also thankful to Hoang Anh Nguyen who helped me to realize important things related to modern models in machine learning as well as current research trends. He also spent nearly 16 hours on flights to present my work in a conference when my visa application was delayed.

In addition, I would like to give my thanks to Prof. Huu Hung Huynh, Anh Tien Mai, Duc Thanh Nguyen and other friends for their support regarding to both academic and life aspects.

The last appreciation goes to my family, especially my sweet wife, Thi Bich Tuyen Nguyen, for her exhortation during my study despite the long distance between our places. She is always my motivation as well as an important part in my life.

Contents

Page

List of Figures

List of Tables

List of Abbreviations

AAE	Adversarial Auto-Encoder
AE	Auto-Encoder
AEI	Active Energy Image
AUC	Area Under Curve
bSHI	backward Single-step History Image
DEI	Depth Energy Image
DoF	Degree of freedom
DLT	Direct Linear Transform
EER	Equal Error Rate
EMD	Empirical Mode Decomposition
FPR	False Positive Rate
fSHI	forward Single-step History Image
GAN	Generative Adversarial Network
GEI	Gait Energy Image
GHI	Gait History Image
GMM	Gaussian Mixture Model
HMM	Hidden Markov model
HOG	Histogram of Oriented Gradient
ICP	Iterative Closest Point
IR	infrared
ISA	Interacting Simulated Annealing
LoPS	Level of Posture Symmetry
MEI	Motion Energy Image
MGCM	Mean Gait Cycle Model
MHI	Motion History Image

MSE	Mean Squared Error
MSI	Motion Silhouettes Image
PF	Particle Filter
PNN	Probabilistic Neural Network
PoI	Point of Interest
PS	Pictorial Structure
RANSAC	RANdom SAmple Consensus
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RQA	Recurrence Quantification Analysis
SDK	Software Development Kit
SFS	Shape from silhouette
SL	Structured light
SPF	Smart Particle Filter
SVD	Singular Value Decomposition
SVM	Support Vector Machine
ToF	Time-of-Flight
TPR	True Positive Rate

Chapter 1

Introduction

Walking is a daily activity that is acquired at an early age, but involves many complex processes. In particular, the movements involved in walking are among the most studied in clinic, because of the possibility of diagnosing numerous pathologies. In the medical context, the walk is considered as a sequence of hierarchically successive phases as illustrated in Fig. 1.1. According to such phase separations, typical gait characteristics (e.g. stride length, half-cycle duration, or walking speed [70]) are able to be efficiently estimated for analysis.

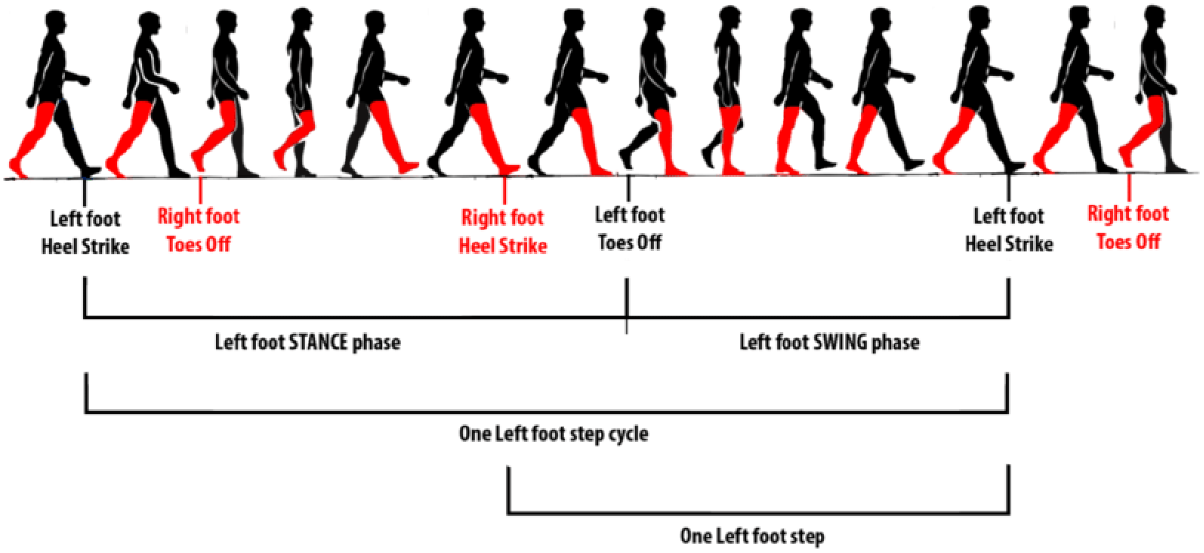


Figure 1.1: Hierarchical walking gait phases [50].

This dissertation fits into this field of gait analysis by going further with the design of an affordable computer vision-based system for the real-time 3D reconstruction and motion analysis of the walking human.

1.1 Overview of gait analysis

Gait analysis plays an important role in detecting and diagnosing human neurological and musculoskeletal problems. According to [13], there are 4 major objectives for performing clinical gait analysis: (1) diagnosing disease entities, (2) assessing disease or injury, (3) monitoring progress, and (4) predicting progress outcome.

The purpose of diagnosis is to distinguish between disease entities. This is usually simplified as a categorization of normal and abnormal movement patterns (e.g. neurological disorder diagnosis [110]). Besides, the determination of a specific disease entity (e.g. Parkinson [8, 144, 163]) given a collection of different gait types is also included in this objective category. To do such tasks, measurements and/or assessments of the disease or injury under various aspects are necessary.

The assessment is expected to provide helpful characteristics of the considering entity for supporting the diagnostic. Depending on particular clinical scenarios, different gait-related measurements may be considered for specific purposes. For example, Schwartz and Rozumalski [127] proposed the Gait Deviation Index (GDI) as a measure of gait pathology estimated from kinematic data. This index was then improved to the Gait Profile Score (GPS) in [14]. Marks *et al.* [90] indicated that the GDI and GPS are not appropriate for all gait-related problems (such as abnormality estimation) and then presented another measure. Recently, many walking gait indices have been proposed to deal with specific gait assessment tasks.

Monitoring/screening progress can be considered as a tracking of measurement results to see whether they are stable or tend to change. Since such values indicate the patient condition, the step of monitoring can help to select appropriate management options such as applying other treatments and/or giving support timely in emergency cases. Besides, typical kinematic data can also be monitored depending on the objective of the work.

In order to assess possible risks when using specific treatments, a prediction of progress outcome is necessary. This provides an overall understanding of which treatments are preferable and how the patient condition changes in the future. In addition, the prediction can also be applied directly on medical data in some problems without treatment, e.g. elderly fall risk prediction [62].

Our approaches presented in this dissertation focus on the first two mentioned objectives and can be extended for the others.

1.2 The measurement of human gait

Kinematic data used in medical researches/treatments are usually obtained from motion capture (mocap) systems. In detail, the patient has to wear some markers at his/her joints so that the segmentation stage can localize each marker position. Infrared reflective markers or even accelerometer-based ones are commonly used in such systems as well as in other fields such as film-making, sport, or anatomy. They provide very high precision in human gait estimation but are expensive. Such high device cost can be considered as a limitation. Another obvious drawback is that the user spends much time for mounting markers on the body. In addition, the operator has to know how to operate and control such complex systems. Therefore other approaches have been proposed, especially automatic vision-based methods, with the goal of reducing the system cost and directly dealing with a specific problem without requiring kinematic data. In recent years, according to the strong development of computer hardware (e.g. high-speed graphic card), marker-less systems, which integrate vision-based algorithms, have achieved promising results in the problem of analyzing human gait. In this dissertation, we focus on proposing vision-based approaches which automatically perform walking gait normality/symmetry assessment.

Our works focus on a low-cost and easy-to-use gait analysis system for a clinical setting. This system is fully automated, with no markers or sensors on the patient's body and no manual intervention. In addition to neurological/musculoskeletal disorder screening, it could enable clinicians to perform a follow-up of patient's recovery after surgery, treatment (e.g. joint replacement) or after a stroke.

While recent vision-based studies mostly process data acquired from a single camera [12, 16] or skeletons [17, 27], our system attempts to obtain 3D information of the human body instead of employing only 2D image since the projection is considered as a lossy (of details) transformation. Concretely, the input of our processing is a sequence of depth images captured by only one depth camera and two mirrors. In the captured scene, the user whose gait needs to be analysed, is walking on a treadmill. Contrary to other studies using multiple cameras (e.g. [10]), only one depth camera appears in our setup while the others are replaced by mirrors. Beside the reduction of device cost compared with multi-camera and mocap systems, another advantage of such combination of devices is that object's images in all viewpoints are captured at the same time by a single camera, the requirement of synchronization can thus be avoided.

1.3 Dissertation structure

Our work consists of two main stages: (1) reconstructing 3D point cloud of subject's body, and (2) performing gait assessment given a sequence of such reconstruction results. The dissertation is structured as follows.

- **Chapter 2** presents a literature review that includes two main sections. In the first one, we introduce briefly typical approaches for 3D reconstruction, in which some mirror-related methods are also presented. The next section discusses some recent marker-less studies working on human gait analysis. Some basic concepts used in next chapters including camera calibration and deep neural network are also presented.

- **Chapter 3** presents our preliminary method for estimating 3D object point cloud using a Kinect 1 and two mirrors. Since the depth map provided by a Kinect 1 is measured according to stereo-pair images, there is almost no depth distortion occurring in captured information. This method, however, needs to be adapted when working on the next generation of Kinect.

- **Chapter 4** gives our approach for dealing with depth distortion when we apply the method in Chapter 3 on a Kinect 2, which employs the time-of-flight technique to measure depth. The processing stages in this chapter are more complicated compared with the work on Kinect 1 since the steps of checking and solving depth distortion have been added.

- **Chapter 5** describes our preliminary approach for the task of gait analysis using the point clouds acquired according to the reconstruction in Chapter 4 under some constraints. This method is model-free and directly estimates a gait symmetry index based on cross-correlation of left and right half-bodies given a sequence of 3D point clouds.

- **Chapter 6** presents a model-based gait normality index estimation based on deep auto-encoder given the same input as the previous chapter. The model, that was carefully designed, can be adapted to provide useful information related to common characteristics of human walking gaits.

- **Chapter 7** focuses on a method based on adversarial auto-encoder that has a great potential for our gait analysis objective but does not require a careful consideration of model architecture. However, there is a trade-off between this advantage and the optimization stability. This model can be extended to apply for other purposes, e.g. generating walking gait samples.

- **Chapter 8** concludes our presented works and suggests some specific applications as well as possible extensions/research directions.

An overview of the dissertation structure is shown in Fig. 1.2.

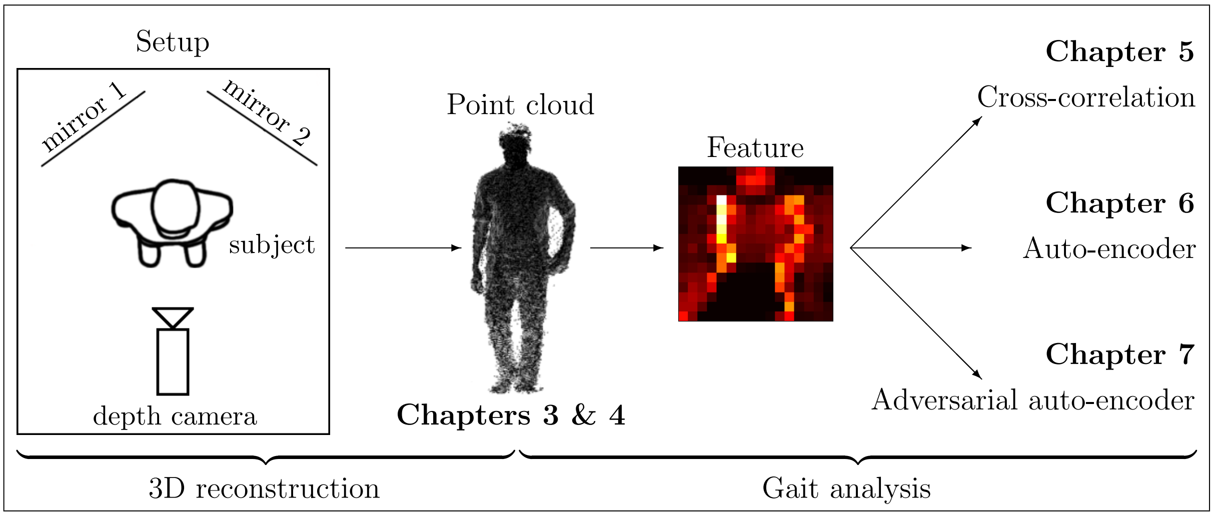


FIGURE 1.2: Schematic diagram of the structure of the dissertation.

Chapter 2

Literature review

This chapter presents a brief literature review of two domains including 3D reconstruction and human gait analysis since they are problems this dissertation is dealing with. For each part, popular approach trends as well as state-of-the-art methods are described together with their advantages and limitations.

2.1 Basic concepts

This section introduces two categories of important concepts that are used throughout next chapters. The first one is camera calibration, which has been employed as a preliminary step in many studies as well as applications in computer vision. The second category consists of typical concepts related to deep neural networks that have been adapted to our gait analysis approaches.

2.1.1 Camera calibration

In computer vision, the term *camera projection* indicates the projection of a 3D point onto an image, which is the basic mechanism of photography. There are three distinct coordinate systems involved in this projection with a specific order: world, camera, and image. The transformation of a 3D point from the world system to the camera one is called *rigid transformation* and is decomposed into a rotation and a translation. The parameters performing this transformation are named *external parameters* and do not depend on mechanical/optical structure of the employed camera. The other transformation, which transforms a 3D point in the camera coordinate system to a 2D point on the image, is performed based on *internal parameters* that involve the camera properties.

A projection of a 3D world point with homogeneous coordinates $\tilde{p}_w$ can be represented by a matrix multiplication as

$$\tilde{p} \propto \mathrm{K}_{3\times3}\,[\mathrm{I}_{3\times3}\ 0]\ \mathrm{R}_{4\times4}\ \mathrm{T}_{4\times4}\ \tilde{p}_w \tag{2.1}$$

where $\tilde{p}$ is the homogeneous coordinates of $\tilde{p}_w$'s projection, K indicates the internal parameters, I is an identity matrix, and R and T denote the rotation and translation, respectively. Let $(X, Y, Z, 1)^\mathsf{T}$ denote the homogeneous coordinates $\tilde{p}_w$ in the world system, eq. (2.1) can be represented in detail as follows

$$\tilde{p} \propto \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \tag{2.2}$$

where f is the focal length, (c_x, c_y) is the principal point that is the intersection between the optical axis and the image plane, $\{r_{ij}\}$ $(1 \le i, j \le 3)$ are elements of the rotation matrix, and $(t_x, t_y, t_z)^\mathsf{T}$ is the translation vector.

Camera calibration is the estimation of internal parameters and sometimes external ones. In this section, we introduce two common methods that respectively employ 3D and 2D

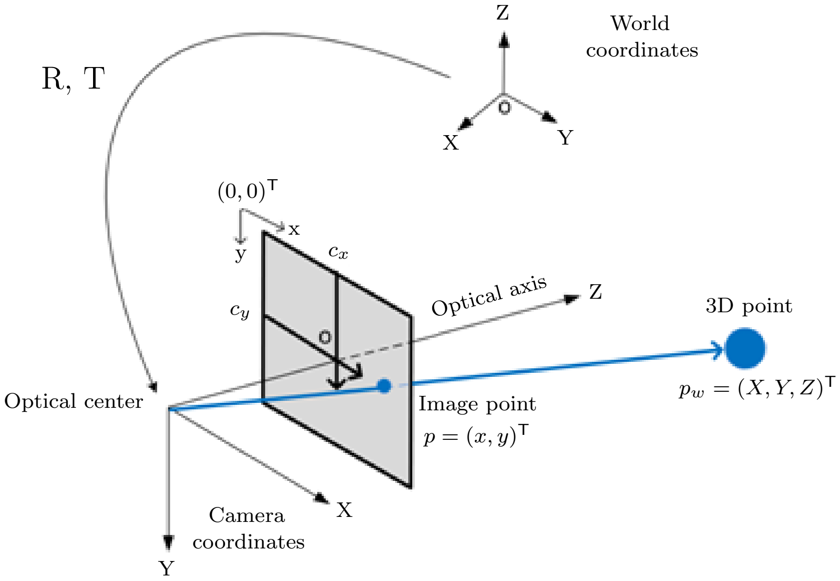


FIGURE 2.1: Relationship between world, camera, and image coordinate systems [111]. The notations R and T respectively indicate rotation and translation involved in the rigid transformation, and (c_x, c_y) is the principal point.

patterns to perform camera calibration. Examples of such patterns are shown in Fig. 2.2.

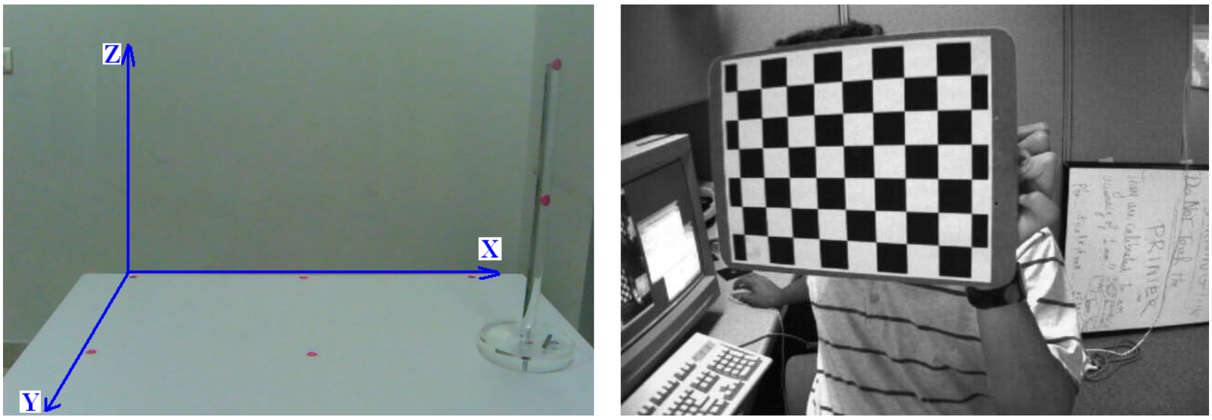


Figure 2.2: Left: a scene with 3D markers for calibration in [66]; Right: an image capturing a 2D pattern for calibration in OpenCV [23].

2.1.1.1 Calibration from 3D points

This method performs the calibration directly on the projection of a collection of 3D points onto an image. According to eq. (2.1) and (2.2), the projection can be represented by a 3×4 matrix P which maps a world point with homogeneous coordinates $\tilde{p}_w$ to an image point with homogeneous coordinates $\tilde{p}$. In general, the matrix P has 11 degrees of freedom (dof) together with a scaling factor. The internal parameters of the camera, such as focal length and principal point, can be extracted from the 3×3 matrix K which is determined from P by applying a decomposition. Concretely, since we have the projection $\tilde{p} \propto \mathrm{P}\tilde{p}_w$, an equation describing the correspondence between $\tilde{p}$ and $\tilde{p}_w$ can be formed as

$$\tilde{p} \times (\mathrm{P}\tilde{p}_w) = 0 \tag{2.3}$$

Eq. (2.3) shows that each correspondence between a 3D point and its image gives three linearly dependent equations, i.e. each correspondence leads to two equations. Therefore, at least $5\frac{1}{2}$ equations are required to solve for P which has 11 dof. The number $\frac{1}{2}$ indicates that in the sixth correspondence, only one equation is needed. Given at least 6 correspondences between 3D world points and image points, the homogeneous linear system (2.3) with the form $Ax = 0\,(x \neq 0)$ can be solved by various algorithms, such as Direct Linear Transform (DLT) or Singular Value Decomposition (SVD) [59]. Let us denote H and p_4 as the left hand 3×3 submatrix and the fourth column of the determined projection matrix P, the camera position can be calculated as $-\mathrm{H}^{-1}p_4$, and the internal matrix K as well as the rotation R are estimated by applying QR decomposition on H.

2.1.1.2 Calibration from 2D pattern

An inconvenience of calibration using 3D points is the requirement of known point coordinates in the 3D world system, that one may spend a lot of time to locate. When we focus only on the internal parameters of the camera, the calibration using a 2D pattern is an appropriate choice. The pattern is a planar surface with known Euclidean geometry, e.g. angles between lines or distances between points. The most commonly used pattern is a chessboard consisting of same-size squares. The camera's internal parameters are estimated from several images capturing the 2D pattern at different viewpoints, in which each view gives a relative pose, i.e. external parameters, between the camera and the pattern. Concretely, with each particular view, the calibration pattern represents the world coordinate system with the origin being one of the corners. Since the pattern is planar, this plane can be fixed at $Z = 0$ without loss of generality. The camera projection then becomes

$$\tilde{p} \propto \mathrm{KR}[\mathrm{I}_{3\times 3} \mid -c] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathrm{KR} \begin{bmatrix} 1 & 0 & \\ 0 & 1 & -c \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2.4}$$

The chessboard in each view thus provides a homography H that involves the internal matrix K. Once all homographies are determined based on 3D corners in the pattern and their image points, the *image of absolute conic*, ω, is then calculated, and the internal matrix K is finally estimated by applying the Cholesky decomposition on ω. The details of this computation are presented in [59].

2.1.2 Deep network

In this section, we briefly introduce deep network, a core parametric function approximation that has various applications in computer vision and natural language processing.

2.1.2.1 Feedforward network

A feedforward network can be represented as a function f with parameters θ mapping an input x to an output y, i.e. $y = f_\theta(x)$. Such networks are constructed as a chain of layers, in which each layer contains a number of units. Each unit in a layer (except for the input layer) performs a weighted summation on all units in the previous layer and followed by a non-linear operation. An example of feedforward network containing 4 layers is presented in Fig. 2.3, in which $x_j^{(i)}$ indicates value of the unit j at layer i. The connection between a pair of units in two successive layers represents the weight used in the summation. The value of a specific unit is calculated as

$$x_k^{(i+1)} = \delta\left(\sum_j w_{jk} x_j^{(i)} + b_k\right) \tag{2.5}$$

where w_{jk} indicates the connection weight between the unit k (that needs to be estimated) and a unit j in the previous layer, b_k is a bias value and δ is a non-linear activation function such as sigmoid, tanh, or ReLU [88].

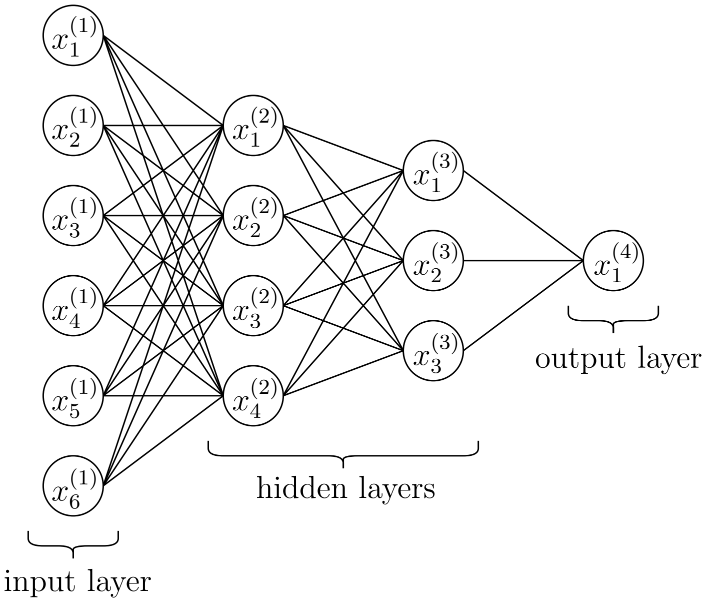


FIGURE 2.3: Example of a typical feedforward network.

2.1.2.2 Optimization

A feedforward network can be designed to perform various tasks. For example, the model in Fig. 2.3 is appropriate for regression and binary classification. The desired task is defined by an objective (or loss) function $\mathcal{L}$ (that is usually non-convex) involving the output. Since it is difficult to determine a closed-form solution of θ due to the complexity of f, the common solving way is estimating an approximation based on a local optimum of $\mathcal{L}$. By optimizing that loss, an approximation of θ is empirically obtained as

$$\theta^* \approx \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\big(f_\theta(x_i), y_i\big) \tag{2.6}$$

where n is the number of training samples x and y indicates their desired outputs.

This optimization phase is also called *learning* where there are two common schemes including supervised and unsupervised learning. Their difference is the definition of y in eq. (2.6) in the training stage.

The training is performed by gradient-based learning using back-propagation algorithm. The parameters of f, e.g. weights w and biases b in eq. (2.5), are randomly initialized and then iteratively modified according to the descending direction of gradients estimated in the parameter space, i.e.

$$\theta_{t+1} = \theta_t - \eta \frac{\partial \mathcal{L}}{\partial \theta}(\theta_t) \tag{2.7}$$

where t and η respectively indicate the counter of iterations and learning rate. Mathematically, the gradient corresponding to each unit is recursively calculated from the output layer through the chain rule. An example of such gradient and the convergence of θ is shown in Fig. 2.4, in which the gradient points upward and the convergence performs according to the opposite direction.

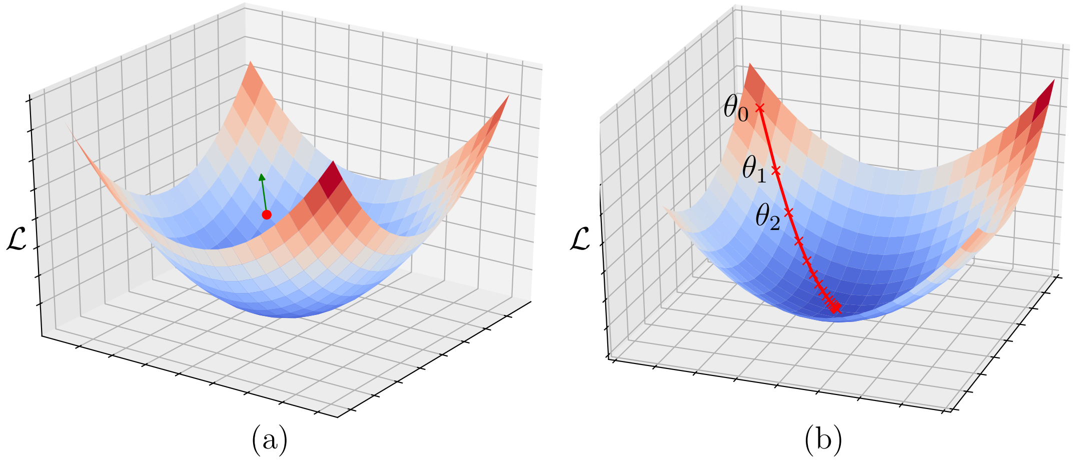


FIGURE 2.4: Example of (a) gradient and (b) θ convergence by gradient descent.

2.1.2.3 Auto-encoder

Auto-encoder is a family of networks that focuses on learning efficient data representation in an unsupervised manner. Such models can be used for determining meaningful underlying features of samples by designing a network, with a bottleneck in the middle, that attempts to reconstruct its input. Due to the reduction of the number of data dimensions, the network is forced to emphasize most useful characteristics so that the difference between an input and its reconstruction is minimal. Besides, an auto-encoder can also approximate a transformation from the input space to another one with similar structural representation, e.g. [94, 142].

Typically, an auto-encoder can be split into two parts including an encoder $h = E(x)$ and a decoder $\hat{x} = D(h)$, in which h is a hidden layer that contains emphasized characteristics of x. The decoder's output $\hat{x}$ is defined depending on the task of interest. For example, the desired value of $\hat{x}$ may be x for reconstruction, or a map of pixel-level labels for segmentation. The idea of auto-encoder can also be generalized as stochastic mappings as $p_E(h|x)$ and $p_D(\hat{x}|h)$.

2.1.2.4 Generative adversarial network

The term generative adversarial network (GAN) was firstly introduced in [53] to indicate estimation of generative models using an adversarial process. The general objective of GAN is learning an empirical distribution of training patterns so that the model has an ability to generate similar samples. Concretely, a GAN consists of two components: generator G and discriminator D. Given training data x, G attempts to perform a mapping $G_{\theta_g}(z)$ from a predefined prior distribution $p_z(z)$ to the distribution of x where θ_g indicates the parameters of G. In other words, G's output is expected to be similar to x. On the contrary, the objective of D is to distinguish real samples x from the outputs of G according to $D_{\theta_d}(x)$ representing the probability that x was sampled from the data distribution. By simultaneously optimizing G and D, the model is expected to generate samples that are similar to x from the explicit distribution $p_z(z)$ and the mapping $G_{\theta_g}(z)$. The general loss can be represented as a two-player minimax game:

$$\min_G \max_D \mathcal{L}(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.8}$$

2.2 3D reconstruction

Nowadays, estimating a 3D model of a scene is one of most popular research fields because of the fast development of hardware (e.g. CPU, GPU) as well as a wide variety of practical applications. In recent decades, many methods have been proposed to solve the problem of building 3D model of an object. We first introduce basic concepts and typical methods for camera calibration (Section 2.1.1), a principal step for 3D reconstruction. We then mention methods that employ simple inputs (Section 2.2.1), i.e. a collection of images. The next section investigates a number of techniques which work on depth information (Section 2.2.2). This content is finally closed by descriptions of some approaches employing mirrors for the reconstruction task (Section 2.2.3).

2.2.1 Typical multiview reconstruction

This category refers to algorithms that reconstruct the 3D model of an object based on its images captured at different viewpoints. There are two well-known methods named *shape from silhouette* (SFS) and *space carving*. Their shared property is that the collection of object silhouettes corresponding to the set of input images plays an important role. The obtained result after applying either of these two methods may be significantly different compared with the real object. Indeed, the former approach creates a *visual hull* [83] of the object, while the latter may contain some redundancies.

The idea of the algorithm SFS is quite simple. When the camera geometry is determined, i.e. all cameras are calibrated, each object silhouette in an image can provide a bounding cone by re-projecting it. In detail, such cone is formed by straight lines connecting the optical center and contour points of the silhouette. The number of re-projected cones is up to the number of input images. The intersection of these cones is considered as the reconstructed object model. This overall process is illustrated in Fig. 2.5. This figure shows that the number of images (views) affects the quality of reconstructed model, i.e. the larger size of the image set is, the less difference between the ground-truth object and the obtained model is. In other words, a large number of cameras are required when the object has complicated surfaces. Another factor may reduce reconstruction accuracy is bad calibration (see Fig. 2.6). Some studies dealt with this issue and proposed algorithms to refine the calibration by optimizing certain constraints [117].

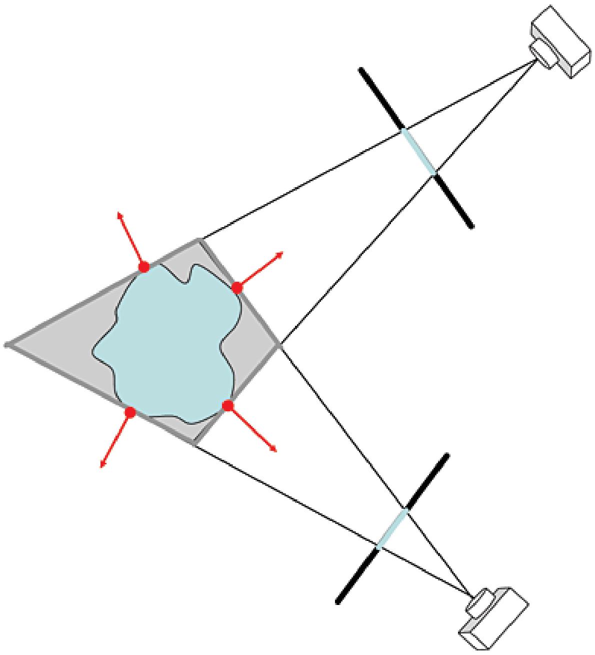


Figure 2.5: Reconstructing the cyan-color object with a system of two cameras. The obtained result (visual hull) consists of gray regions together with the object itself. The red points indicate intersections between re-projected cones and object boundary. Source [40]

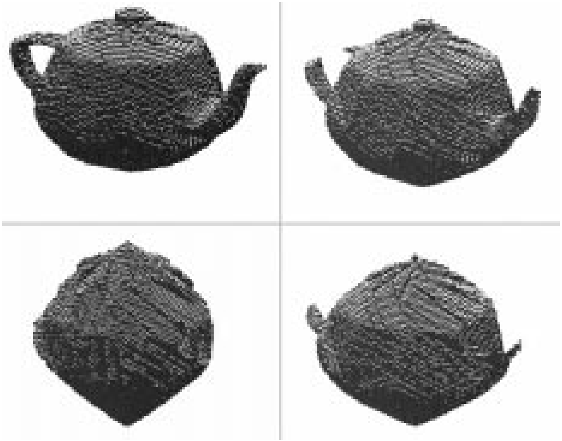


Figure 2.6: Reconstruction results of a teapot using SFS with input images captured from 9 views [117]. In these 4 images, rotation errors occur with different levels. The error values clockwise from the top-left image are: 1°, 5°, 10°, 20°.

The space carving can be imagined as sculpture. In this technique, a space region containing the object, which is needed to be reconstructed, is defined and separated into small volumes called *voxel*. Differently from SFS, the role of silhouette is to check if a voxel should be kept or removed. In other words, the 3D model is formed by removing some voxels outside the object. According to known camera geometry, each voxel is projected onto all images. A voxel is removed from the defined region if any of its projections is outside the corresponding silhouette. Some additional constraints can be considered to improve the carving quality, e.g. checking color consistency of voxel's projections. A simple illustration is shown in Fig. 2.7. Similarly to the method SFS, space carving also provides bad reconstruction result in the case of bad calibration. Another drawback of space carving is the high computational cost.

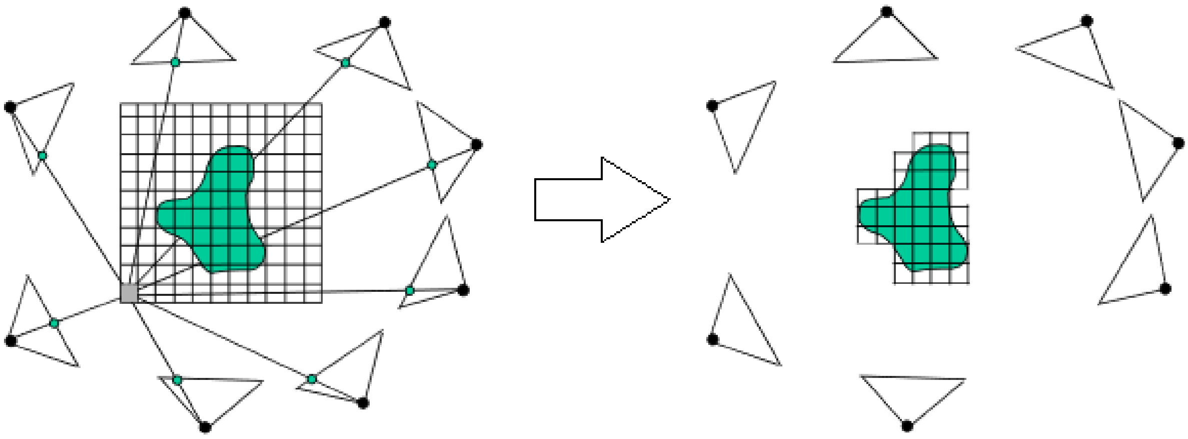


Figure 2.7: An illustration of reconstructing a 3D object with images captured by 8 cameras around [73]. The left image indicates an initialization of space carving, in which each voxel is represented by a square and the green object is inside the entire volume. The right image shows the volume after being *carved*. In order to reduce the difference between the obtained result and the real object, i.e. smoothing the 3D model, the size of voxel should be decreased.

2.2.2 Reconstruction with depth

An obvious limitation of the two mentioned methods is that reconstruction quality depends on viewpoints. For example, in order to model a surface with a concave region, images captured at appropriate positions are required. In practical applications, possible positions for placing cameras are usually limited. Therefore, techniques estimating depth information have been developed for a long time. The mentioned concave region can be modeled by placing the suitable device at only one position. An important term in these techniques is *depth map* that indicates relative depths to pixels in the input images. Depth cameras, i.e. self-contained devices which directly measure the depth of a scene, are created according to one of three techniques including stereo vision, structured light (SL), and time-of-flight (ToF).

Stereo vision. The typical approach for reconstructing depth map is stereo vision, which is similar to the human binocular system. The basic principle is to measure a distance, called *disparity*, between projections of a point in two or more cameras. The coordinates of a world point can be determined by triangulation given its corresponding disparity and geometrical relationship between the cameras. The problem thus becomes finding pixel correspondences between input images which capture a common scene. An important employed assumption is that the appearance of the world point's projections is identical in every viewpoint. According to epipolar geometry, searching the correspondence given a pixel is performed on its epipolar line on the other image (see Fig. 2.8). In order to simplify this task, rectification is often employed to make epipolar lines horizontal [138]. The details of stereo vision techniques are described and evaluated in [128]. An obvious limitation of stereo vision is that it is difficult to find correspondences when reconstructing an object with homogeneous surface (e.g. large region with same color or uniform texture), the obtained 3D model would thus have poor quality. Therefore, reconstruction result significantly depends on the scene in practical situations. In order to overcome this drawback, structured light has been employed.

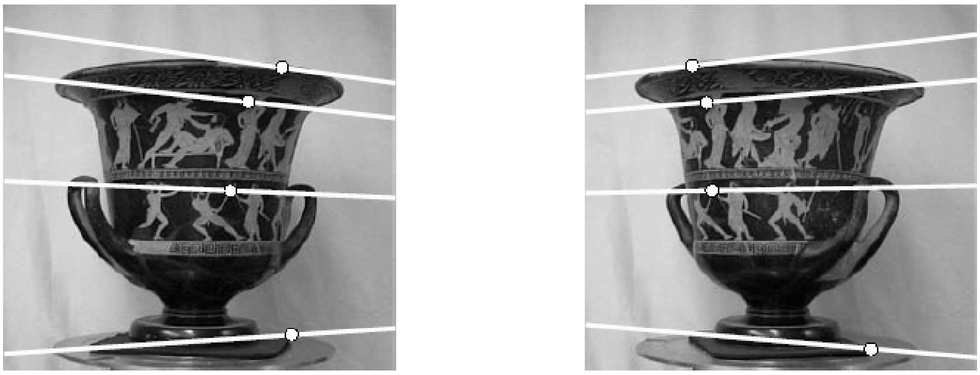


Figure 2.8: An example of point correspondences and their epipolar lines (in white) [59]. The motion between two views is a translation and rotation. In each image, the direction of the other camera may be inferred from the intersection of epipolar lines.

Structured-light. The principle of this technique is to project images, where each pixel is easily recognized, to the scene, and then infer depth based on the deformation of the captured image. Similarly to stereo vision, two or more images are employed, but one of the cameras is replaced by a projector that projects a known image, called *pattern*, to the scene. The task of finding pixel correspondences is expected to be simpler since neighborhoods of the pattern and captured images can be matched with less dependency on the texture of the object surface, and the depth is estimated by triangulation. Therefore, pattern selection plays an important role in this technique. In a pattern, every pixel has its own codeword directly mapping to the coordinates of this pixel. Various patterns have been used in recent studies, and in many cases, a set of patterns is employed to improve the matching accuracy.

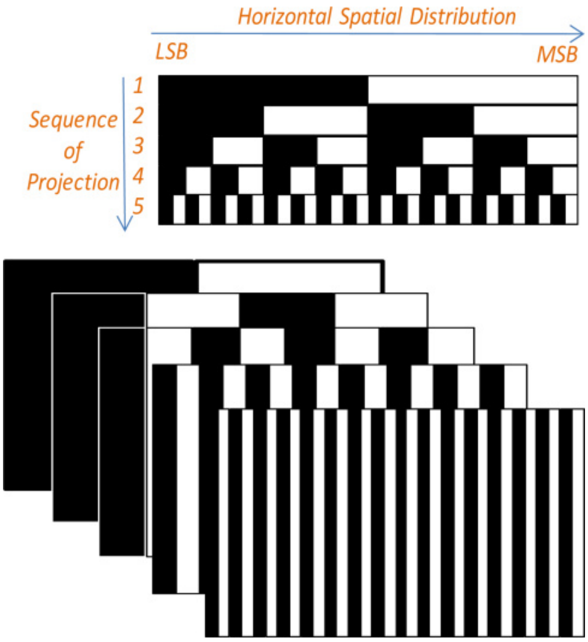


FIGURE 2.9: A sequential binary coded pattern used for 3D imaging [49]. The codeword of a pixel is determined by concatenating its binary values after projecting all patterns.

In basic structured light techniques, codewords were generated by projecting a set of patterns along a certain order, the structure of each pattern can thus be simple. Therefore such methods are called *time-multiplexing*. Binary and Gray codes are two of the most popular patterns. An example of binary coded pattern is shown in Fig. 2.9. In order to reduce the number of patterns, some studies (e.g. [25, 61]) tried to increase the number of codes in each one, so called n-ary codes. There is a trade-off between such techniques and codeword determination accuracy due to the fact that the task of pattern segmentation is more complicated.

A common disadvantage of time-multiplexing techniques is the large number of required patterns, thus many other methods attempted to overcome this drawback by concentrating all the coding scheme in a unique pattern. The codeword corresponding to a certain point of such pattern is indicated by its neighborhood. Visual features described in a neighborhood may be a statistic (e.g. histogram) of intensity and/or color, or just simply a group of raw pixels. Pattern designed based on the Bruijn code is a typical approach that has been applied in many studies [85, 165]. The advantage of a Bruijn sequence is the good windowed uniqueness property, i.e. each subsequence of the window size appears only once, which helps to minimize the ambiguity occurring when finding pixel correspondences. An example of using a pattern designed based on the Bruijn code for 3D reconstruction is shown in Fig. 2.10.

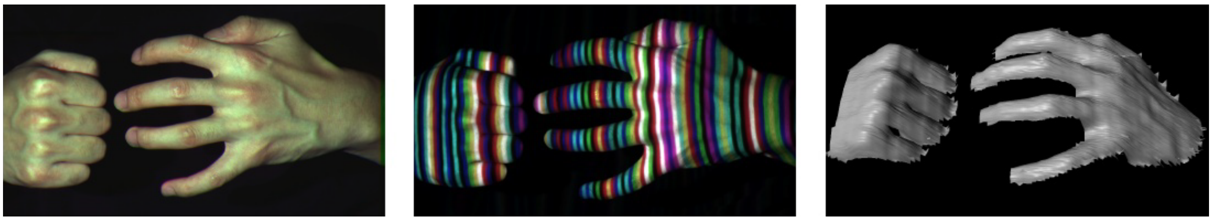


Figure 2.10: Reconstructing the shape of two hands in the study [165]. From left to right: real scene with two hands in front of a dark background, the scene under illumination with color-striped pattern designed based on the Bruijn code, reconstructed model shown at a different viewpoint.

Based on the principle of neighborhood-based matching, a device was created and is widely used in many applications, named Kinect (version 1). An overall description is presented in [166]. This device measures depth with the support of an infrared (IR) projector, which projects a speckle pattern (see Fig. 2.11), and an IR receiver. In this pattern, each point has its unique signature estimated according to relative positions of points in the vicinity, it thus simplifies the task of point identification. An advantage of this device is that the depth measurement is almost independent of the surface texture and can be performed in real-time. Many applications have been created based on this device, e.g. large-scale 3D reconstruction [68, 96] and pose estimation [132, 133]. The neighborhood codification, however, has its own limitation. The decoding stage may be difficult because there are some cases where spatial neighborhoods cannot be recovered and consequently the matching stage might then yield errors.

Time-of-Flight. This term indicates a variety of methods that estimate distance according to time-related factors. A ToF depth camera employs two principal devices including an IR emitter and an IR receiver. A signal is emitted by the former and then captured by the latter. The depth measurement is performed depending on the type of such signals. In practice, two popular types which have been used are *high-speed pulse* and *continuous wave*, and the corresponding depth values are calculated based on traveled length or phase shift of the signal, respectively. Concretely, a ToF camera with a pulsed modulation determines the distance to a 3D point based on the measured absolute time the pulse

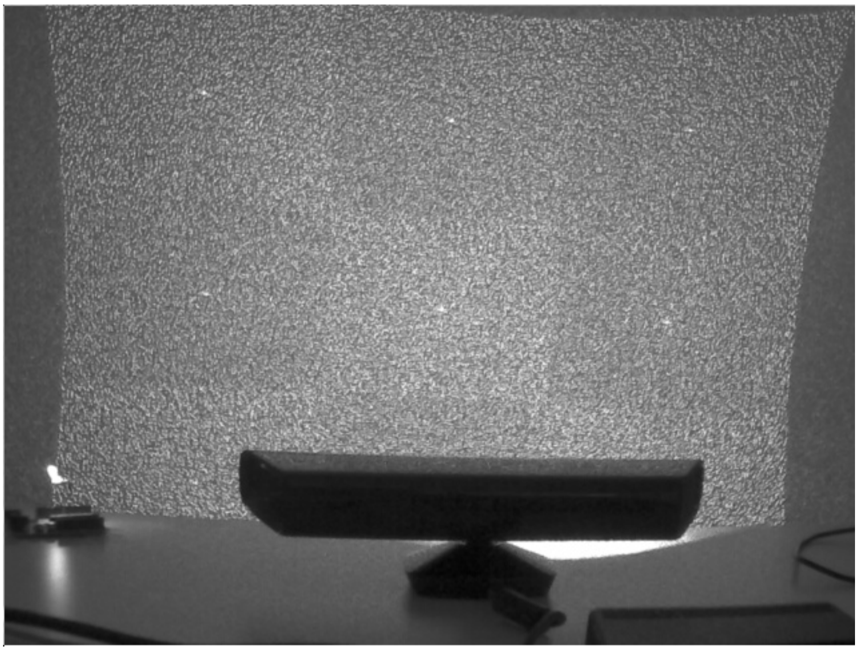
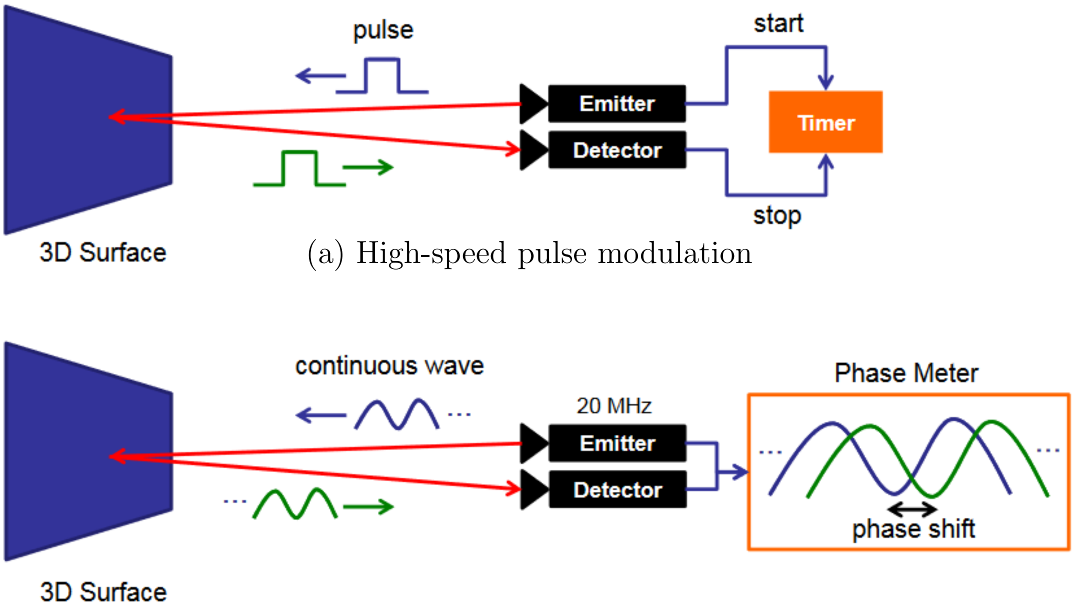


FIGURE 2.11: Image of the speckle pattern projected by a Kinect 1 [19].

travels along the emitter-scene-receiver path and the known speed of light. The other ToF type employs continuous sinusoidal waves instead of pulses, in which the distance from the camera to a 3D point is measured based on the phase shift between the emitted and received signals corresponding to this point. An overview of these two depth estimation schemes is shown in Fig. 2.12. The next generation of the mentioned Kinect employs



(a) High-speed pulse modulation

(b) Continuous wave modulation

FIGURE 2.12: Depth estimation schemes of two common modulation types employed in ToF depth camera [26].

ToF techniques to measure the scene's depth. The Kinect 2 is also a cheap device and provides depth map with higher quality compared with the previous version [164]. In this dissertation, this device plays an important role in data acquisition.

2.2.3 Reconstruction using mirror

Similarly to our work, mirrors were also important in some other approaches for reconstructing 3D objects. The principle of employing mirrors together with only one camera is to gather object images captured at different viewpoints into a single image. The task of synchronization is thus avoided, and the device cost is also reduced. In [63], geometrical constraints on real and in-mirror object silhouettes were used to perform the 3D reconstruction. Two algorithms were proposed working on two types of input including silhouette and depth map. For the former, the object model is formed by intersecting back-projected cones corresponding to object silhouettes that are extracted from real scene and mirror regions. The other algorithm reconstructs 3D model as the intersection of depth ranges. Another approach was proposed by Epstein *et al.* [39] employing structured light to reconstruct an object model according to its directly captured image and images in mirror regions. The interactive structured light reconstruction system introduced in that work is shown in Fig. 2.13. The color landmarks on the stand and on the mirror non-reflective contour serve the task of detecting, tracking and pose estimation. In order to overcome the problem that a portion of object image in the mirror may be occluded because of the real object, the researchers define a 3D bounding box enclosing the object and then project it onto mirrors using OpenGL to obtain reliable regions of object images as well as light patterns. By moving a mirror to different positions, the entire object can be completely reconstructed.

In our works, a depth camera and two flat mirrors are used for scene acquisition. Our system can thus capture object's depth maps at 3 different viewpoints. Differently from [39], mirrors in our system are placed at fixed positions because we do not require a very detailed model for the task of gait analysis. Besides, the processing complexity is expected to be reduced compared with [39] since the task of depth estimation is integrated into our employed cameras. However, some depth distortions may occur depending on the depth estimation scheme. Our studies therefore also propose a way for dealing with them. The details of our works on 3D reconstruction are presented in Chapters 3 and 4.

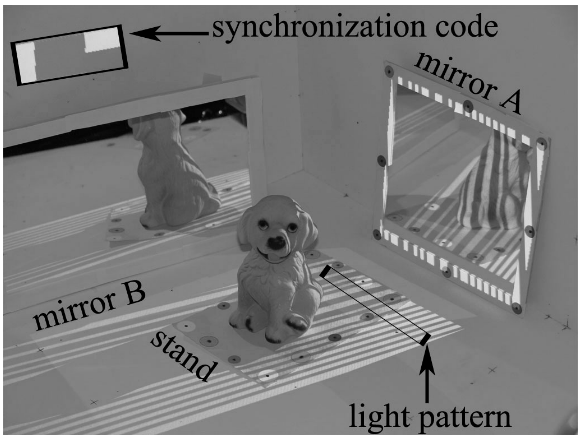


FIGURE 2.13: The interactive reconstruction system introduced in [39]. The synchronization code is used to combine each pair of projected light pattern and captured image.

2.3 Human gait analysis

In recent decades, many studies on gait analysis have been proposed with a wide variety of gait-related applications such as gender and/or person identification, health assessment, and action detection in surveillance systems. Researchers usually classify approaches of gait analysis into two categories: model-based and model-free.

2.3.1 Model-based methods

The term *model-based* indicates approaches measuring or fitting parameters related to kinematic data to given human models, i.e. estimating human pose from observations. Such explicit models are usually formed by a person's kinematics, shape, and/or appearance. An important advantage of such methods is the low dimension of feature space. Besides, the ambiguity occurring due to occlusion can be overcome once the model is fitted to observed data. However, this process requires a high computational cost because of the complexity of the underlying structure.

The ways for solving the problem of pose estimation can be categorized into three types including global optimization, filtering and/or prediction, and local optimization [46]. The first one can provide high accuracy estimations since such techniques search for the best solution in the search space. Simulated annealing [48] can be considered as the most popular method for global optimization since it has been applied successfully in many vision-related studies, e.g. image segmentation [129] and object recognition [116]. This algorithm does not require a good initialization. However, its application in practice is limited by low convergence speed. In filtering approaches, the body pose is estimated from noisy observations. Temporal coherence is widely employed to predict body parts in a specific frame. Such techniques only give good results when the human pose is simple or predefined. In the case that a human model has a high number of degrees of freedom, the motion analysis may be inaccurate. The remaining, local optimization, can be considered as the simplest since it does not require predefined complicated models. Such techniques, e.g. Iterative Closest Point (ICP) [82], can provide high (even best) accurate result if they have a good initialization.

Some approaches that estimate human pose from multiple views have been proposed in many studies since the position of object points can be recovered from images captured at different viewpoints. Gall *et al.* [46] presented a two-layer framework for estimating human pose from multiple images. An initial pose is created by the first layer and then will be refined by the second layer. Concretely, the interacting simulated annealing (ISA) [47] is employed to perform pose initialization based on silhouettes, colors, and geometrical constraints between cameras in the system. The second layer reduces jitters from the result of the previous layer and then uses local optimization to fit the model in order to increase the accuracy of pose estimation. This study also shows that the ISA provides the best initialized pose compared with some other optimization and filtering approaches including local optimization (ICP), standard particle filter (PF) [6], annealed particle filter (APF) [33] and another variant algorithm (SPF) [24]. An example of model fitted based on the two-layer framework is shown in Fig. 2.14.

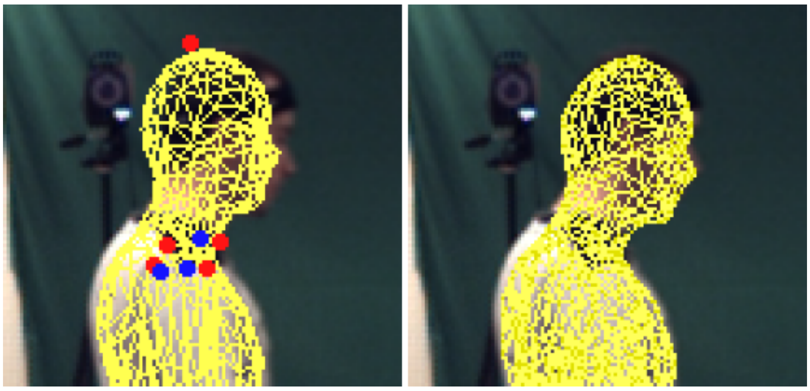
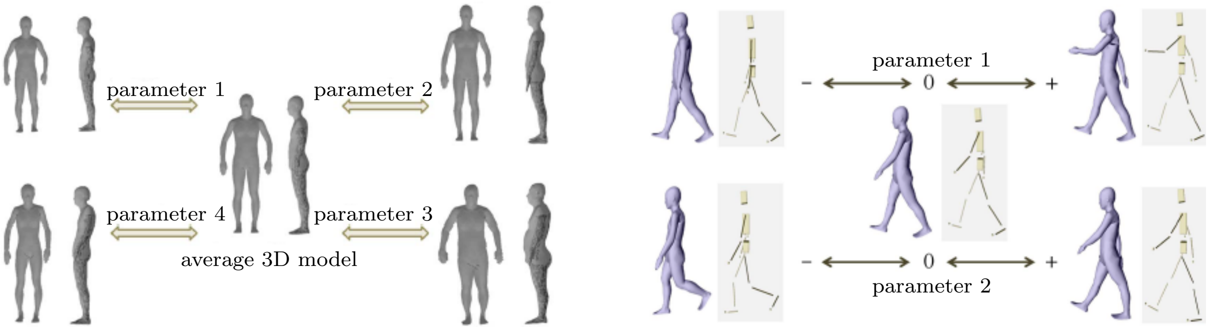


FIGURE 2.14: From left to right: initialized model (with biased head) in the first layer and better fitting resulted in the second layer [46].

Some other studies employed directly a set of given statistical 3D models for estimating human pose. For instance, a method proposed by Shinzaki *et al.* [131] attempted to fit a 3D human model to an observed subject in order to overcome the limitation of view-point in the problem of silhouette-based person identification. The researchers employed a model including two statistical ones called 3D shape and gait motion, in which each one consists of an average model together with some adjustable parameters (see Fig. 2.15). This study assumed that the Sun's position with respect to the camera as well as the subject position throughout the duration of one gait cycle are both known. In the stage of finding the best appropriate model for an observed subject, there are three steps executing in loops until convergence. First, an initial position of the model is set according to the position of the observed object, the system then synthesizes a virtual image containing object's silhouette and shadow using the known Sun's position. Second, contours corresponding to the silhouettes and shadows in both observed and synthesized images are extracted. Third, the steepest descent method is used to minimize an evaluation value measured based on the comparison of obtained contours in the two images. By repeating the three steps, the subject's sequential 3D models can be reconstructed.



(a) Statistical shape model (b) Statistical gait motion model

FIGURE 2.15: Two statistical models employed in [131]. Each model consists of an average model and adjustable parameters: (a) changing parameters leading to different body shapes (e.g. thinner, fatter, taller), (b) adjusting parameters providing various postures of typical walking.

Instead of 3D shape model, skeleton-related one is also considered in many studies. For example, Simo-Serra *et al.* [134] presented an approach estimating 3D human pose which can work well on noisy observations. Since state-of-the-art 2D detectors [7, 41, 153] are usually employed to detect human body parts from an image, resulting regions may be inaccurately estimated or not cover entirely some of the body parts. The work [134] attempted to overcome this drawback by propagating possible noise determined from the image to the shape space using a stochastic sampling strategy. A set of ambiguous 3D shapes, whose projections on the image are indistinguishable, would be then obtained. A 3D human shape was finally achieved by imposing kinematic constraints on the set for picking an accurate 3D pose. The basic idea of this study is shown in Fig. 2.16.

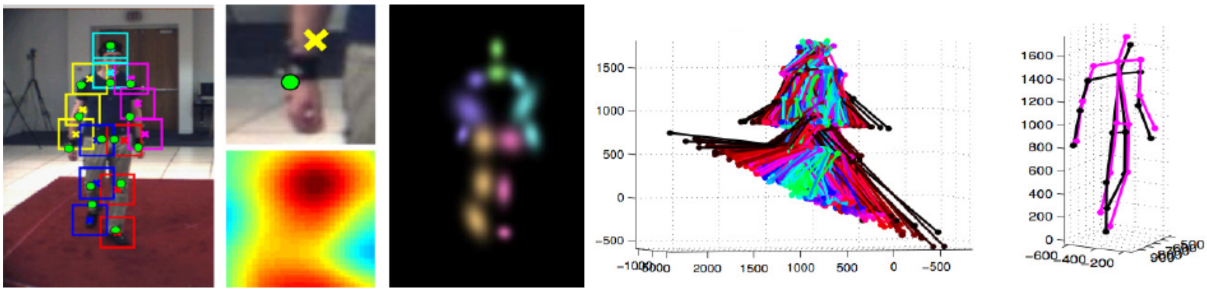


FIGURE 2.16: Estimating 3D human pose from noisy images [134]. From left to right: the image with bounding box results of a body part detector, inaccurate detection since the bounding box does not match the joint position (the green dot indicates true position of the joint), heat map scores corresponding to output of the 2D detector as Gaussian distributions, sampling the solution space and initializing a set of ambiguous 3D human poses, and the ground truth (black) together with the resulted accurate pose (magenta) selected by simultaneously imposing kinematic and geometric constraints.

2.3.2 Model-free approaches

The methods in this category consider the motion of overall human body instead of focusing the underlying structure. Compared to model-based approaches, the computational cost of model-free ones is significantly lower. However, a trade-off should be considered since the feature space is more complicated with more dimensions. Therefore techniques reducing the number of dimensions are usually employed, e.g. feature selection and dimensionality reduction.

Some state-of-the-art features have been proposed to describe human gait in a temporal sequence, i.e. gait accumulation. Their principle is to accumulate a sequence of gait frames into an image describing the gait signature. The computational cost for temporal matching and storage requirement can thus be significantly reduced. Gait Energy Image (GEI) is one of the simplest gait signatures and has been proven to give high accuracy in gait analysis [37, 57]. The GEI feature, $G(x, y)$, is calculated as the average of a sequence of pre-processed binary silhouettes, $B(x, y, t)$, corresponding to a human body as follows

$$G(x, y) = \frac{1}{N} \sum_{t=1}^{N} B(x, y, t) \tag{2.9}$$

where N is the number of frames of the input sequence and t is the frame index. Since GEI is an average template, this is not sensitive to possible noise randomly appearing in some frames of the input sequence. As mentioned in [57], the robustness could be improved by removing pixels with low energy values compared with a threshold. In addition, the silhouette sequence does not need to be separated into gait cycles. Another gait signature which is also widely applied is Motion History Image (MHI). Differently from GEI, the MHI can visually describe the way a motion performed. Concretely, the intensity of a pixel in MHI is a function of the motion history at its position, in which brighter value indicates more recent motion. The MHI function $H_\tau(x, y, t)$ is defined as

$$H_\tau(x, y, t) = \begin{cases} \tau & D(x, y, t) = 1 \\ max(0, H_\tau(x, y, t-1) - 1) & D(x, y, t) \neq 1 \end{cases} \tag{2.10}$$

where τ is a fixed duration, and $D(x, y, t)$ indicates the image of motion regions which is determined as the result of frame differencing [69] between two consecutive frames at time t and $t - 1$. We also employed the MHI in a previous study to describe the change of walking velocity [101]. The GEI signature is appropriate for person identification while the MHI is useful for action recognition. Beside GEI and MHI, some other gait signatures formed according to a sequence of binary silhouettes have been proposed such as Motion Energy Image (MEI), Motion Silhouettes Image (MSI), Gait History Image (GHI), forward Single-step History Image (fSHI), backward Single-step History Image (bSHI), and Active Energy Image (AEI) (see Fig. 2.17). The calculation of these features is summarized in [87].

In order to get more details about human gait from observations, some researchers attempted to estimate subject's pose in individual frames and represent it as *probabilistic assemblies of parts* [93]. Concretely, these studies first attempt to detect likely locations corresponding to distinct body parts and then combine them to obtain a configuration which best matches the considering observation. Such approaches can thus overcome occlusion-related limitations of tracking-based methods since the pose estimation can

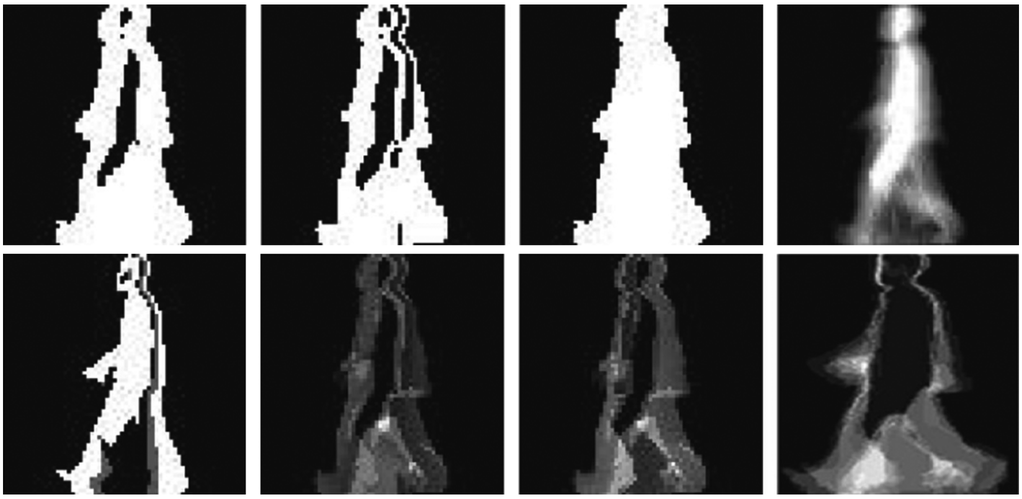


Figure 2.17: Examples of mentioned gait signatures [87]. From left to right and top to bottom: MEI, MHI, MSI, GEI, GHI, fSHI, bSHI, and AEI.

be independently performed on each frame. The pictorial structure (PS) model, which was first proposed in [42], has been employed to estimate human pose in many studies. The PS model principally represents an object as a collection of parts, in which certain pairs of them have connections. This model is naturally expressed by a undirected graph $G = (V, E)$ with vertices $V = \{v_1, ..., v_n\}$ representing n object parts and each edge $(v_i, v_j) \in E$ corresponding to the connection between parts v_i and v_j. Each object instant is given by a flexible configuration $L = (l_1, ..., l_n)$ specifying parameters of n object parts such as position and orientation. The pose estimation task is thus matching a PS model to an image by minimizing an energy function. An optimal match can be defined as

$$L^* = \arg\min_L \big[\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j)\big] \tag{2.11}$$

where functions $m_i(l_i)$ measuring the mismatch when v_i is at location l_i, and $d_{ij}(l_i, l_j)$ measuring model deformation when v_i and v_j are placed at l_i and l_j, respectively. In order to solve this problem, the posterior probability of a configuration L given a single image I and a model θ can be estimated according to Bayesian rule as

$$p(L|I, \theta) \propto p(I|L, \theta)p(L|\theta) \tag{2.12}$$

As mentioned in [43], it is difficult to determine a prior distribution of the Bayesian formulation, $p(L|\theta)$, so that this prior is both informative and generally applicable. In pictorial structure, this prior can be initialized based on the relative positions of object parts (e.g. head, torso, and arms for upper body pose estimation). In order to localize possible object parts in practice, many studies trained corresponding detectors based on different features, such as shape context [7] or histogram of oriented gradient (HOG) [31]. In the test stage, such parts are detected with different probabilities by filtering the input image, e.g. [126]. Some other researchers attempted to reduce the search space by applying a generic detector with a large sliding window to localize human locations, and part detection is then performed within resulting windows [38]. An extension of pictorial structure named *deformable structure* was proposed by Zuffi *et al.* [167] to capture the non-rigid shape deformation of object parts since some human body parts could deform non-rigidly. An example of the two structures is shown in Fig. 2.18. In some practical situations, the pose estimations is employed once and then a tracking technique is performed over time.

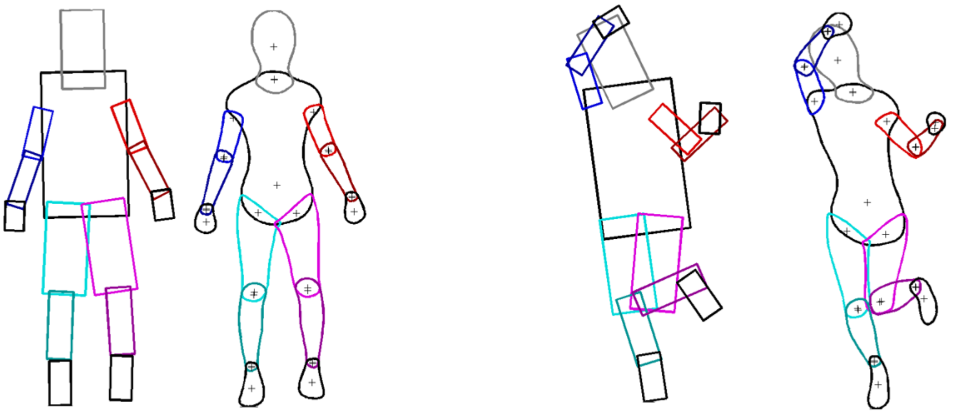


Figure 2.18: Two pairs of pictorial structure and similar deformable structure (right model in each pair) capturing 2D body shape deformation [167].

With cheap depth sensors such as Kinect, some studies tend to perform pose estimation on a depth frame. A state-of-the-art approach proposed by Shotton *et al.* [132] has been integrated into the Kinect for localizing human joints. This technique provides a human skeleton corresponding to the subject that appears in the scene with high accuracy while the pose estimation and skeleton tracking are performed in real-time. The key feature describing each body pixel involves calculating the depth differences between just a few pixels. In the training stage, a very large dataset with about 1 million synthetic image pairs (see Fig. 2.19) was employed to train a random forest (RF) [30] using the standard entropy minimization, in which each pair includes a depth image and its ground truth labels corresponding to body parts. The advantages of RF consist of its efficiency, inexpensive computational cost, and ability of parallelization. Once the RF is trained,

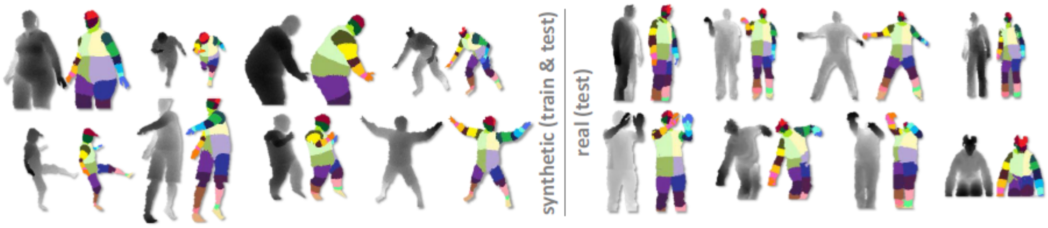


FIGURE 2.19: Synthetic and real data for training the random forest integrated into the Kinect [132]. Each pair of images consists of a depth image and ground truth labeled body parts.

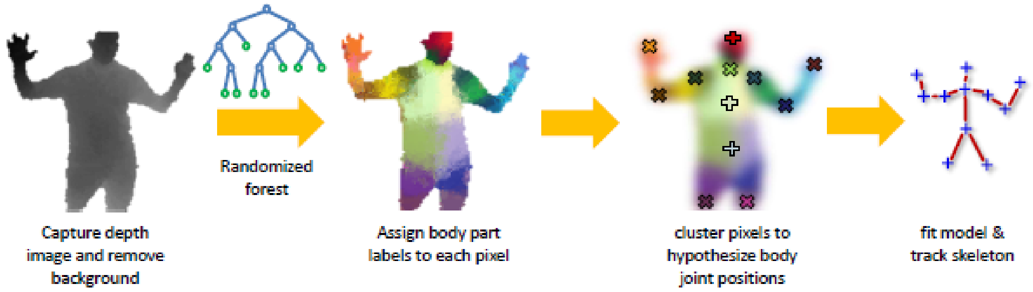

Capture depth image and remove background
Randomized forest
Assign body part labels to each pixel
cluster pixels to hypothesize body joint positions
fit model & track skeleton

FIGURE 2.20: Basic stages of the Kinect skeletal determination and tracking [79].

every pixel of a unknown depth frame traverses down all decision trees to provide a distribution of body parts associating to the pixel. The posterior probability corresponding to a pixel computed over the forest then assigns a body part label to this pixel. These per-pixel label distributions of the entire body are finally clustered together to give the position hypotheses of predefined joints. The pipeline of the described process is shown in Fig. 2.20. The model learnt by this approach is largely invariant to visual factors such as body shape, pose, and clothing.

2.4 Studies related to this dissertation

To the best of our knowledge, there are very few major studies on the problem of gait index estimation employing 3D body reconstruction. Researchers instead (1) focus on other tasks (e.g. human posture classification [54], person identification [67]) given the 3D body, or (2) estimate human walking gait index using other common inputs (e.g. depth map [12], skeleton [17]). We thus present in this section two independent parts related to the two main stages of our works.

2.4.1 3D gait reconstruction

A common method for 3D reconstruction is using a multi-camera system. Iwashita *et al.* [67] built a studio with 16 cameras mounted around a specific region where walking gaits are performed (see Fig. 2.21). The reconstruction was performed according to the volumetric intersection technique given 16 binary silhouettes extracted by background subtraction. The 3D models were employed to synthesize subject's silhouettes corresponding to arbitrary camera directions supporting the problem of person identification. This system requires a synchronization protocol for acquiring 16 images of the same posture.

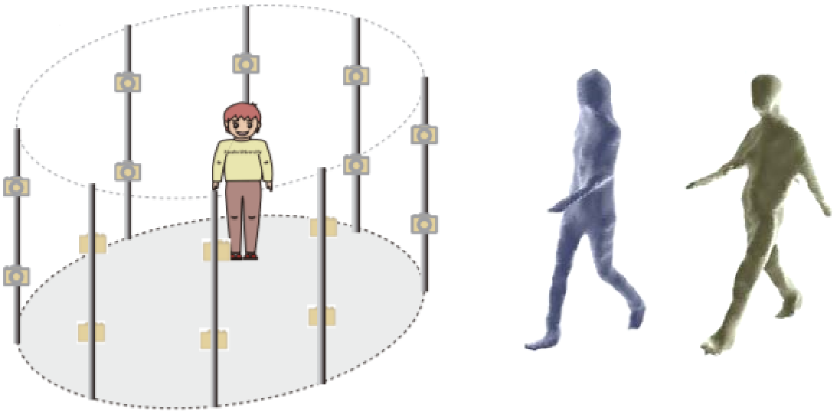


FIGURE 2.21: Multi-camera system for walking gait acquisition in [67] and reconstructed 3D models.

Instead of model with surface, the work [54] represented 3D body as a volume of voxels as shown in Fig. 2.22. The volume is formed according to the space-carving technique given a subject's silhouette and cast shadows of infrared lights. The computational cost is the main drawback of this reconstruction. Therefore, the system is inappropriate for practical applications that require fast (or even real-time) execution or a powerful machine must be used. In addition, such volumes may contain redundancies (illustrated in Fig. 3.1) that increase the complexity of gait index estimation.

Such redundancies can be reduced with the increment of the number of cameras. Some recent studies deal with this problem using depth cameras since they can localize points lying on the object surface. For example, Auvinet *et al.* [10] employed a collection of depth images to reconstruct 3D volumes of human postures while Kim *et al.* [77] performed alignments on point clouds captured from multiple Kinects to form an unified body.

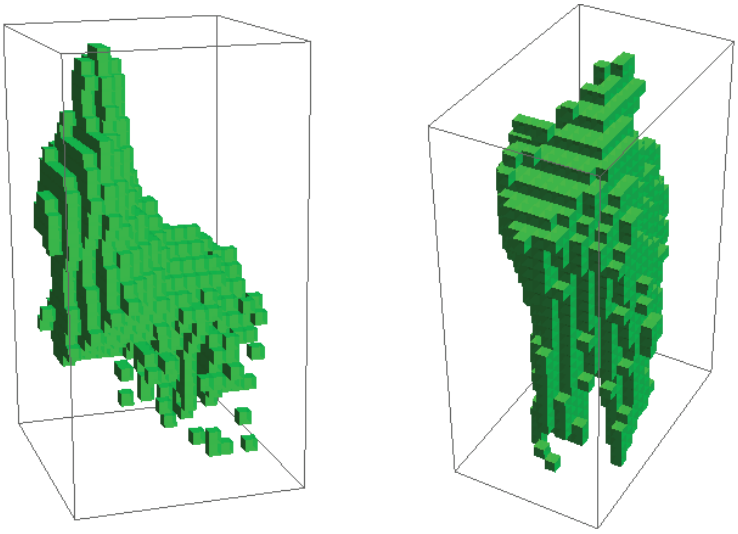


Figure 2.22: Voxel volume representing human gait in [54] (left: sitting on chair, right: standing).

Unless using simulation as [10], simultaneously capturing depth maps from multiple Kinects may lead to scene deformation. Concretely, the IR signal emitted from a camera (see Figs. 2.11 and 2.12) can affect the depth estimation of the others. A schedule of camera acquisition might thus be required.

In order to avoid these mentioned problems, our works employ a novel system configuration including a depth camera and two mirrors. It can be considered as a collection of 3 depth sensors but does not require any synchronization and takes lower-cost devices.

2.4.2 Gait index estimation

Differently from our perspective on the input of gait analysis, recent vision-based studies employ typical data such as depth map and skeleton. The use of 3D skeleton is especially popular since it can be determined in real-time and is provided in low-cost devices such as Kinect. It can be considered as a bridge connecting the medical and vision research fields since some typical kinematic parameters can be approximated from 3D coordinates of skeletal joints. For example, gait characteristics can be represented under medical viewpoint such as step length and gait cycle in [17] or vision one such as skeletal concatenation using sliding window in [27]. An example of gait cycle determination is shown in Fig. 2.23. A major drawback of skeleton processing is that the body joint localization is easily noisy when applied on pathology walking gaits.

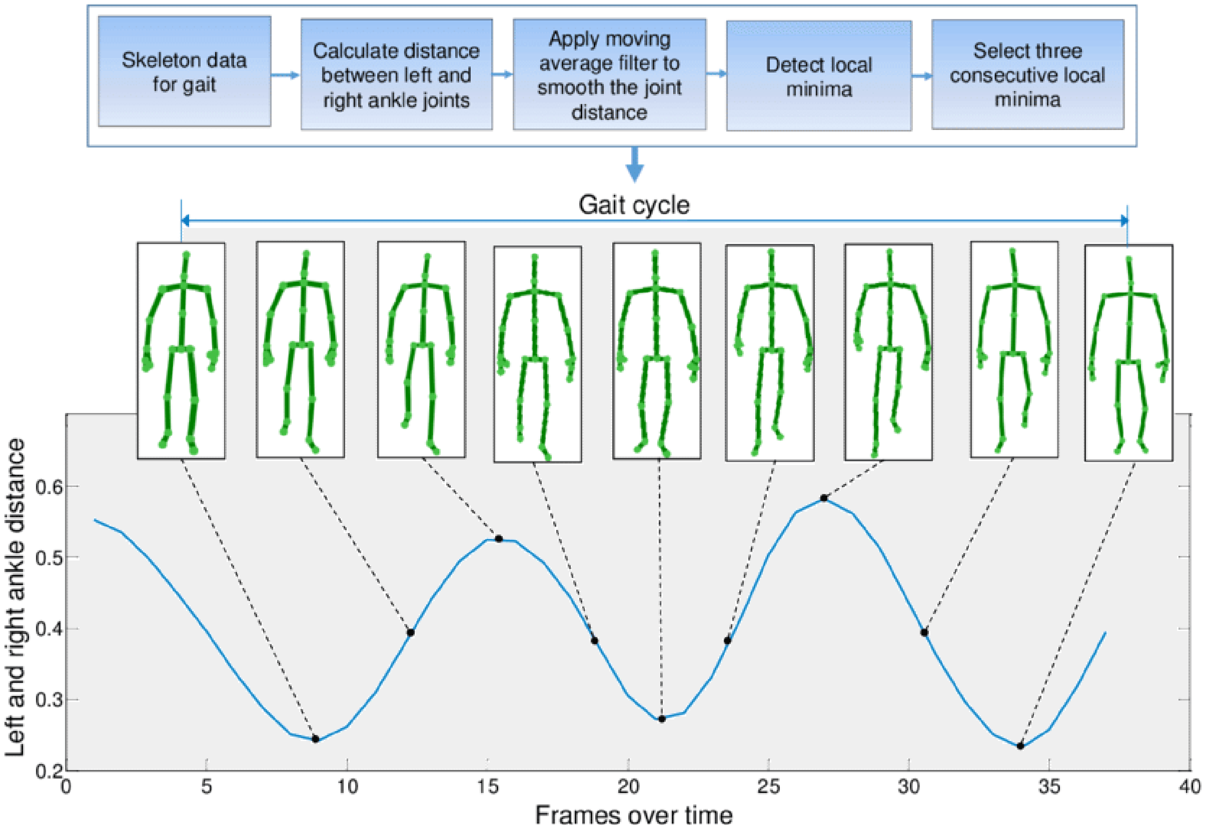


FIGURE 2.23: Gait cycle separation based on distance between left and right ankles [2].

Regarding to the depth map, a common approach is representing accumulated walking gaits by a single image. Such single representation may be an average depth body (named Depth Energy Image) within a gait cycle [121] or a key depth map corresponding to a specific walking stage [11] (see Fig. 2.24). A common difficulty of such methods is that they significantly depend on gait cycle determination. Performing automatically this step usually provides noisy results. Besides, preprocessing is also necessary to smooth each depth map.

The work of Auvinet *et al.* [9] can be considered as the one closest to ours since it also estimates a gait index from 3D body (voxel volume) reconstructed from 3 depth cameras. That study, however, focuses only on step length and requires a manual operation of gait phase separation. Our approaches aim to avoid such steps of input enhancement and work automatically only on raw point cloud data while still guarantee to obtain promising results. Besides gait analysis approaches working on sequences of 3D point clouds in this dissertation, our side-works regarding to skeleton and depth map can be found in the studies [97, 102, 104, 105, 109].

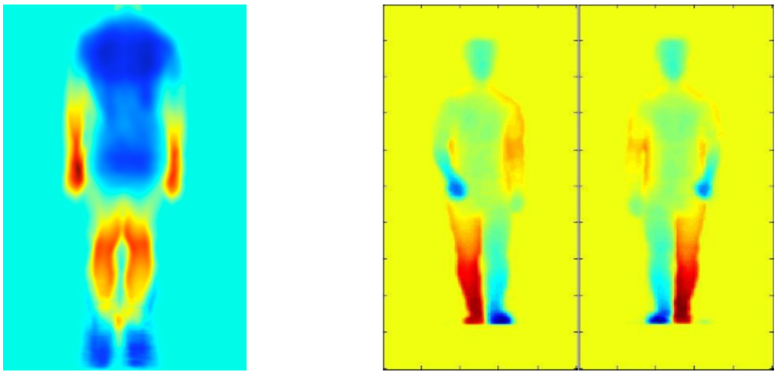


Figure 2.24: Depth maps used for gait analysis in two related studies. Left: Depth Energy Image [121]. Right: key depth maps corresponding to left and right step heel strikes [11], respectively.

Chapter 3

Reconstruction with Kinect 1 (structured light) and mirrors

As mentioned in previous chapters, 3D reconstruction with mirrors is a principal work in this dissertation. Our preliminary attempt for this task is to reconstruct an object using a depth camera, that uses matching-based depth estimation, together with two mirrors. Our method has been published as the following conference paper:

Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Matching-based depth camera and mirrors for 3d reconstruction. In *Three-Dimensional Imaging, Visualization, and Display 2018, SPIE conference on*, volume 10666, pages 10666 – 10666 – 10, Orlando, FL, USA, April 2018. SPIE. doi: 10.1117/12.2304427. URL https://doi.org/10.1117/12.2304427

3.1 Abstract

Reconstructing 3D object models is playing an important role in many applications in the field of computer vision. Instead of employing a collection of cameras and/or sensors as in many studies, this chapter proposes a simple way to build a cheaper system for 3D reconstruction using only one depth camera and 2 or more mirrors. Each mirror is equivalently considered as a depth camera at another viewpoint. Since all scene data are provided by only one depth sensor, our approach can be applied to moving objects and does not require any synchronization protocol as with a set of cameras. Some experiments were performed on easy-to-evaluate objects to confirm the reconstruction accuracy of our proposed system.

3.2 Introduction

Compared with 2D image, processing 3D information usually requires more computations as well as more resources such as memory and storage capacity. With the strong development of electronic devices in term of processing speed, many vision-based applications are now focusing on 3D data in order to exploit more information. Some researchers performed the reconstruction based on a sequence of images captured by a camera at different positions [113]. An obvious drawback of such methods is that the object of interest has to be static. Therefore in order to deal with moving objects, many recent studies employed a system of multiple color cameras [75] and/or depth sensors [10]. The main disadvantage of such approaches is that they require a synchronization protocol (e.g. [10, 36]) when working on moving objects, and sometimes each camera and/or sensor has to be connected to a unique computer. The latency of system as well as cost of devices are thus increased. In order to overcome these problems, our approach employs only one depth camera together with 2 or more mirrors for building a system for 3D reconstruction. Synchronization is not necessary since all captured data are provided by only a single device, and the complexity and cost of equipments can thus be decreased.

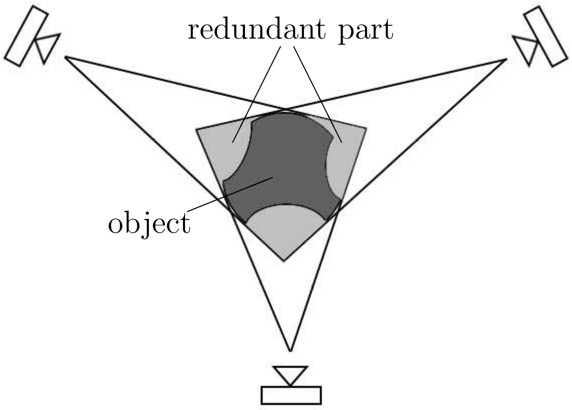


Figure 3.1: Redundancy when reconstructing a 3D object using shape-from-silhouette or space carving techniques in which the inputs are three color images. The overall gray region is the reconstruction result.

As mentioned, a 3D reconstruction could be performed with a system of basic color cameras (e.g. convex hull). However, there are some advantages for using a depth sensor in this work. The most important one is that a depth map could indicate details on the object surface such as concave regions while a combination of object silhouettes provides a convex hull with redundancies (see Fig. 3.1). Another reason is that our approach requires mirror calibrations, i.e. estimating mirror planes, a depth sensor thus reduces the complexity of this stage.

Depth cameras which are popularly used in vision applications could be categorized into two types: matching-based, e.g. stereo and structured-light (SL), and time-of-flight (ToF). Let us introduce briefly these two depth estimation mechanisms to explain why a depth sensor using the former technique is preferred in our approach. A matching-based approach generates a depth map by matching input images. A stereo camera captures two color images of a scene at different viewpoints while a SL-based device projects a template of light and then matches it with the corresponding image captured with a camera. Since this mechanism is related to the human vision system, we can also expect a good depth estimation of the object behind mirrors. A ToF camera uses infrared (IR) emitter and receiver to measure depth of scene based on the traveled time of a high-speed pulse or the phase shift of continuous wave. Both measurements depend on traveled trajectories of IR signals which are more difficult to predict with high-reflection surfaces such as mirror. The depth of reflected objects could thus become significantly deformed. In summary, the depth map provided by a matching-based depth camera is easier to manage than a ToF sensor in our configuration. An illustration of our setup is presented in Fig. 3.2.

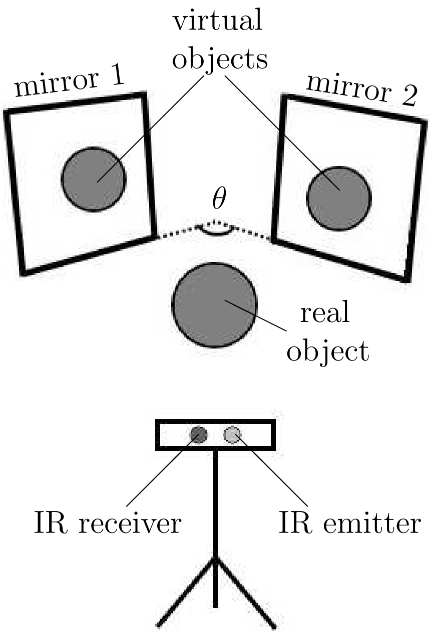


FIGURE 3.2: An overview of our setup including a camera with structured-light depth estimation, two mirrors, and a sphere. The notation θ indicates the angle between the two mirror's surfaces.

The remaining of this chapter is organized as follows. The reliability of depth map measured by SL matching with mirrors is analyzed in Sec. 3.3. Section 3.4 mentions the way of calibrating planes of mirror surfaces. Reconstructing object point cloud from depth map is presented in Sec. 3.5. Our experiments and evaluation are shown in Sec. 3.6, and Sec. 3.7 presents the conclusion.

3.3 Reliability of SL matching with mirrors

According to geometrical optics, image of an object is reversed when seen in a mirror. As mentioned in Sec. 3.2, matching-based depth measurement usually employs a passive approach such as stereo or an active one such as SL. Although stereo images of an object-behind-mirror are reversed, the matching process can be expected to provide a reliable depth map. However, we can guess that a SL-based camera may give an ambiguity since there are reflected regions in the captured image while the corresponding light pattern is still unchanged. Fortunately, this ambiguity does not happen as explained below. Let us consider a configuration example in Fig. 3.3, in which the light pattern is characterized by the order of two different colored lines emitted from the projector. It is obvious to see that this pattern is flipped on the real object surface. This swap happens after the light

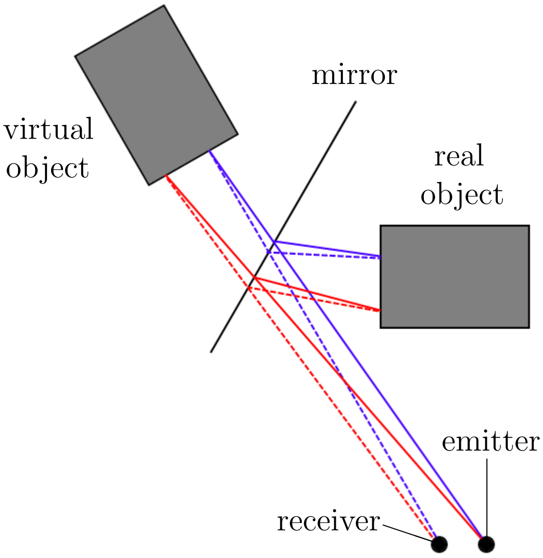


Figure 3.3: Example of emitting and receiving a structured-light pattern in a mirror. Emitter (or projector) is the source which emits the light pattern, and receiver captures the illuminated scene. The received pattern is not reversed because the rays are reflected twice.

rays touch the mirror surface. The received pattern, however, is similar to the original one (in term of order) since it touches the mirror surface twice when traveling from the emitter to the receiver. Therefore the matching result with a SL-based camera will be unaffected and reliable.

3.4 Mirror calibration

Calibration is considered as the primary step in most vision-based applications. When dealing with a system of cameras, researchers typically perform the calibration for estimating not only the internal camera parameters but also external relationship between these cameras [10]. Even when working on a configuration which is similar to ours, researchers also consider it as a collection of a realistic and virtual cameras [3]. The proposed solutions in such studies thus employ external calibrations. Our work avoids this redundancy by estimating only internal camera matrix together with mirror surfaces based on captured depth data. The idea of using mirror planes is quite simple: object parts which are seen in a mirror will be reconstructed by reflecting them through this mirror. Since the camera calibration process has been dealt with by many approaches for color input [59] and monochromatic depth images (e.g. [10]), this section only mentions the latter problem.

There are many ways for estimating the mirror plane based on the depth map provided by the camera. An indirect method could be employed by putting one or some easy-to-locate calibration objects (e.g. simple marker, cylinder, cube) in front of the mirror. The plane is then determined based on 3D coordinates of these real objects together with corresponding virtual ones behind the mirror. Another method, that directly estimates the mirror surface, is also possible. The depth map of the mirror's frame could be used to assess the position of the plane if it is large enough. In our setup, the frame is too small, thus the plane equation of each mirror surface was estimated based on 3D coordinates of some markers placed on it. Since the depth Z of a pixel (x, y) is given by the depth image, the corresponding point (X, Y, Z) in 3D space can be localized using internal parameters of the depth camera as

$$[X, Y, Z]^\top = Z \cdot diag(f_x^{-1}, f_y^{-1}, 1)[x - c_x, y - c_y, 1]^\top \tag{3.1}$$

where (c_x, c_y) is the principal point on the image, f_x and f_y are focal lengths. These values can be easily estimated based on standard camera calibration techniques [138].

Given a set of n markers, the mirror plane, which is characterized by a collection of 4 parameters (a, b, c, d), is determined by solving the equation

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0 \tag{3.2}$$

where (X_i, Y_i, Z_i) is the 3D coordinates of the i^{th} marker. A solution could be approximated by performing singular value decomposition (SVD) on the first matrix [138]. Depth information estimated in practical environments, however, is usually affected by noise. The obtained mirror plane thus may have a significant deviation, especially when working on low-cost devices. Therefore, we applied a combination of RANSAC [44] and SVD to reduce the effect of outliers (noise) in order to get better results. The next section describes in detail the use of mirror surfaces in reconstructing a 3D point cloud.

3.5 3D reconstruction

According to our configuration, which consists of an object directly seen by a depth camera and 2 or more mirrors around, the object is represented in captured images as a collection of object's pieces including a real one and some virtual ones, i.e. behind mirrors. As mentioned, the whole object is formed by combining the real points directly seen in front of the camera with reflections of virtual pieces obtained via corresponding mirror planes.

In detail, given the internal matrix K including camera parameters in Eq. (3.1), a set of n object pixels $\{\tilde{p}\}_n$ in the captured image and corresponding depth values $\{Z\}_n$, m mirror planes $\{\pi\}_m$ and 2D object boundaries $\{\hat{b}\}_m$, our method for reconstructing a point cloud $\{P\}$ which represents the object is as follows:

Algorithm 3.1: Reconstructing a raw point cloud of the object from a depth image.

```
  Data: $K, \{\tilde{p}\}_n, \{Z\}_n, \{\pi\}_m, \{\hat{b}\}_m$
  Result: $\{P\}$
1 $\{P\} \leftarrow \emptyset$;
  for $i \leftarrow 1$ to $n$ do
2     $P_i \leftarrow$ Reproject($\tilde{p}_i, Z_i, K$);
      for $j \leftarrow 1$ to $m$ do
          if $\tilde{p}_i$ inside $\hat{b}_j$ and $P_i$ behind $\pi_j$ then
3             $P_i \leftarrow$ Reflect($P_i, \pi_j$);
              break;
          end
      end
4     $\{P\} \leftarrow \{P\} \cup P_i$;
  end
```

In the Algorithm 3.1, the reprojection at line 2 is performed based on Eq. (3.1), and the reflection at line 3 is done according to the following equation [29]

$$P_r = P - 2\|\hat{n}\|^{-1}(P^\top \hat{n} + d)\hat{n} \tag{3.3}$$

where P_r is the point reflected from P via a plane of parameters (a, b, c, d), and $\hat{n}$ is the corresponding normal vector, i.e. $\hat{n} = [a, b, c]^\top$.

When working on moving objects, the Algorithm 3.1 is an appropriate choice because it can run in real-time with a low computational cost. However, in some situations, one could want to reconstruct an object point cloud with a higher density. The space carving technique [81] is a suitable approach in these cases, especially with static objects. The overall idea of our workflow could be summarized in Algorithm 3.2 by following steps performed on each voxel of a predefined volume. In detail, we first compute the projected pixel based on the voxel coordinates and the calibrated internal camera matrix. The corresponding estimated depth $\|P\|$ is then compared with the voxel's depth $\|v\|$, and a deviation is calculated. The voxel is kept in the volume if such deviation is less than a predefined threshold, i.e. this voxel corresponds to a real point captured directly by the depth camera. Otherwise, the voxel is reflected through each mirror and the mentioned checking is repeated on each virtual reflection result. The voxel is removed from the volume if the deviation condition is not satisfied with at least one mirror. Beside the space carving approach presented here, a high-density cloud could be obtained by using an additional high-resolution camera and employing a registration between its images and

Algorithm 3.2: Reconstructing a volume of voxels representing the object, in which the assignment of Boolean values *true* or *false* to each voxel indicates that this voxel is kept or removed, respectively.
Notation:
th: a threshold related to the thickness of the reconstructed object boundary
$Z_{\tilde{p}}$: measured depth value at pixel $\tilde{p}$

```
Data: $V_{init}, K, \{\tilde{p}\}_n, \{Z\}_n, \{\pi\}_m, \{\hat{b}\}_m$
Result: $V_{carved}$
$V_{carved} \leftarrow V_{init}$;
$th \leftarrow t_0$;
foreach $voxel\ v \in V_{carved}$ do
    $\tilde{p} \leftarrow$ Project($v, K$);
    if $\tilde{p} \notin \{\tilde{p}\}_n$ then
        $v \leftarrow false$;
        continue;
    end
    $P \leftarrow$ Reproject($\tilde{p}, Z_p, K$);
    $v \leftarrow true$;
    if Abs($\|v\| - \|P\|$) $< th$ then
        continue;
    else
        for $j \leftarrow 1$ to $m$ do
            $v_j \leftarrow$Reflect($v, \pi_j$);
            $\tilde{p}_j \leftarrow$ Project($v_j, K$);
            if $\tilde{p}_j \notin \{\tilde{p}\}_n$ or $\tilde{p}_j$ not inside $\hat{b}_j$ then
                $v \leftarrow false$;
                break;
            end
            $P_j \leftarrow$ Reproject($\tilde{p}_j, Z_{\tilde{p}_j}, K$);
            if Abs($\|v_j\| - \|P_j\|$) $\geq th$ then
                $v \leftarrow false$;
                break;
            end
        end
    end
end
```

captured depth frames. This method is not described in this chapter since one objective of our work is to reduce the device price.

Given a voxel volume V_{init} in front of the mirrors and input terms similar to the Algorithm 3.1, the space carving is applied to create the corresponding object volume V_{carved}

as in Algorithm 3.2. Let us notice that the origin of the coordinate system in this algorithm is the camera center. With another 3D space, a rigid transformation [84] between it and the camera space is required, and vector terms in the Algorithm 3.2 (e.g. voxel v, point P) thus need to be recalculated with respect to the camera center.

In some cases, the collection of object pixels $\{\tilde{p}\}_n$ may be defined as a group of points representing a region which contains the object instead of a set of true pixels. Depending on each application as well as visual properties of the object of interest, some additional conditions could be integrated into the two algorithms to reduce noise, i.e. reconstructed points which are not object's parts. In our experiments, our system employed such constrains including background subtraction and color filtering. This content is not described in this chapter since it does not play a principal role in our proposed algorithms.

3.6 Experiment

3.6.1 Configuration and error measurement

In order to evaluate our approach in reconstructing 3D object point clouds, we built a configuration of a depth camera and two mirrors. The camera employed in our experiments is a Microsoft Kinect 1, which provides depth information by emitting an IR dot pattern and matching it with the corresponding captured IR image. This device was selected because of its cheap price and good SDK with many functionalities [148]. There were two objects used in our experiments consisting of a cylinder and a sphere. Reconstruction accuracy was estimated by fitting each resulting point cloud according to its true shape and then calculating an error based on the cloud and fitted geometric parameters. Root mean square error (RMSE) was determined according to fitted center and radius in the case of a sphere, and the main axis and radius for the cylinder. The mean value of such deviations was also estimated in order to provide another error type which is easy to visualize.

3.6.2 Test on sphere

With the spherical object, we performed the reconstruction at different angles between the two mirrors. The object shape was fitted by applying the RANSAC technique on the obtained cloud. The RMSE error was then estimated based on the equation

$$\epsilon_{sphere} = \sqrt{\frac{\sum_{i=1}^{n}[dist(P_i,\tilde{c}) - \mathrm{Re}]^2}{n}} \tag{3.4}$$

where *dist* is a function measuring Euclidean distance between two input coordinates, P_i is the i^{th} element of n 3D points, $\tilde{c}$ and Re are the fitted sphere's center and radius, respectively. A simple mean error was also calculated as average of deviation values, i.e. $dist(P_i,\tilde{c}) - \mathrm{Re}$. Our experimental results corresponding to the test on the sphere are shown in Fig. 3.4.

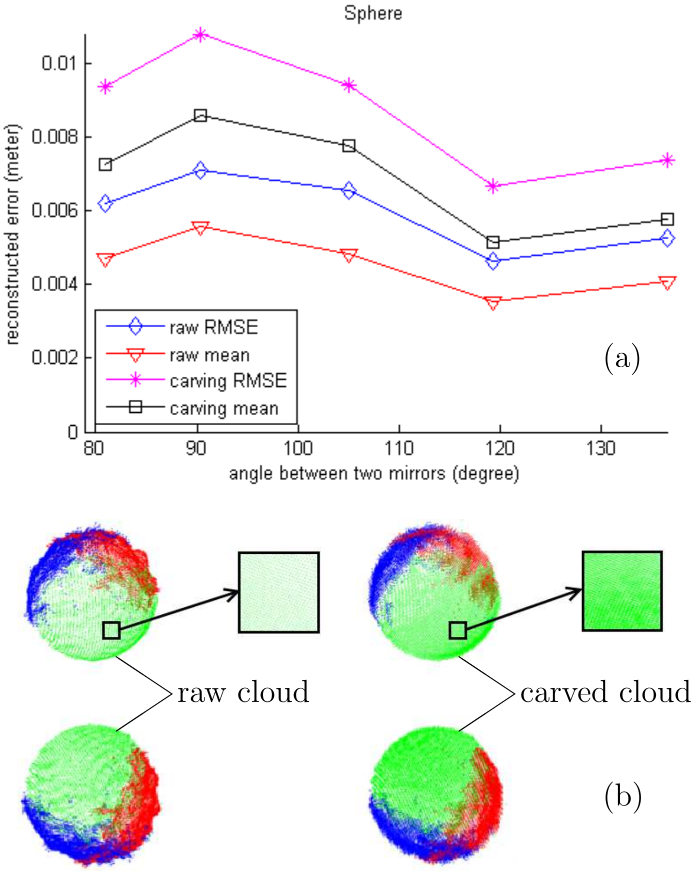


FIGURE 3.4: (a) Estimated fitting errors when reconstructing a sphere with different angles between the two mirrors, and (b) reconstructed point clouds which are seen at different viewpoints. The two terms "raw" and "carved" indicate the two point clouds reconstructed by Algorithm 3.1 and 3.2, respectively. Different colors in cloud indicate points obtained from different sources, i.e. a depth camera and two mirrors.

Both measured errors were less than 1 centimeter. The average length of estimated radii was 117 millimeters while the true value, which was manually measured, was 115 millimeters. The errors corresponding to the space carving approach were always greater than the other because of its higher cloud density and thicker surface. According to all four curves in Fig. 3.4, reconstruction errors tend to be lowest at a specific degree between mirrors (about 120° in our experiment). We can thus expect that in an arbitrary configuration (in terms of distance between object and camera and/or mirrors) with two mirrors, there exists an angle between them which provides reconstructed object point clouds with lowest errors. This value can be estimated by trial-and-error.

3.6.3 Test on cylinder

When working on the sphere, we focused on its center coordinates and radius. With cylinder, the error measurement was performed based on the line equation of its axis and radius length. The experiment was done under different average distances between the object and the two mirrors. RANSAC was also employed for fitting the point cloud. The error was measured as

$$\epsilon_{cylinder} = \sqrt{\frac{\sum_{i=1}^{n}[dist(P_i, \ell) - \mathrm{Re}]^2}{n}} \tag{3.5}$$

where ℓ is the straight line corresponding to the cylinder axis, $dist$ calculates the distance from a 3D point to a line, and Re is the fitted cylinder radius. The mentioned mean error is estimated by computing mean of deviations $dist(P_i, \ell) - \mathrm{Re}$. The obtained results are presented in Fig. 3.5 together with visualization of a pair of reconstructed clouds for top and side viewpoints. The true radius was 150 millimeters.

Similarly to our previous experiments, errors measured on raw clouds are less than on space carving results. These charts also show that when the distance between the tested object and mirrors increases, fitting errors of the former approach tend to slightly decrease while the latter one go in the opposite direction. This property can be explained based on captured 2D depth images. As usual, we can guess that depth information, which is directly measured, is usually more reliable than the reflected one. When increasing the mentioned distance, the number of pixels representing the object's part directly seen by the camera is also larger, and such quantity in mirror regions is reduced. Reconstruction

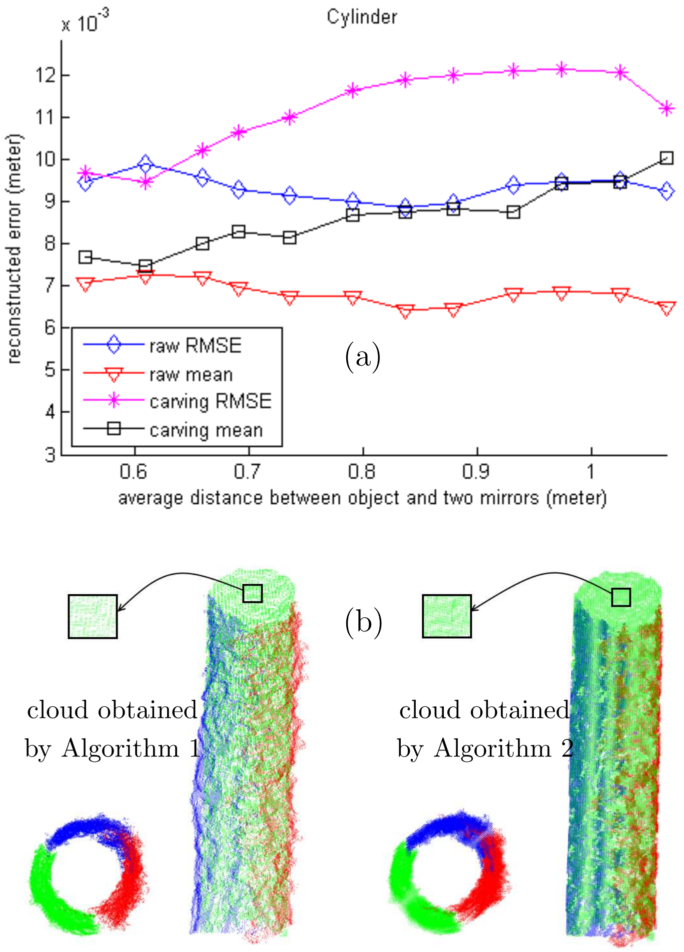


FIGURE 3.5: (a) Fitting errors when applying our approach on a cylinder at different (average) distances from the two mirrors, and (b) visualization of the clouds (top and side viewpoints), in which three colors correspond to points generated from the depth camera and 2 mirrors.

errors thus become lower because of the increased proportion of more reliable information. This change, however, reduces the accuracy of the space carving approach because mirrors generate real 3D clouds in which points are more sparse. Sub-volumes of such regions thus could be carved wrongly producing larger errors. This drawback can be overcome by performing an interpolation on depth images to provide a sub-pixel level for 2D projection from each voxel. According to these properties, we can obtain good results when using either of both proposed algorithms by creating a configuration in which all devices are near each other around the object. In our experiments, the distance between the Kinect and tested objects was about 2 meters.

As an illustration of a practical application, we also tried to reconstruct a 3D point cloud representing a human body based on the experimental configuration. The obtained results are shown in Fig. 3.6. We believe that these clouds are acceptable for realistic applications such as human gait or shape analysis.

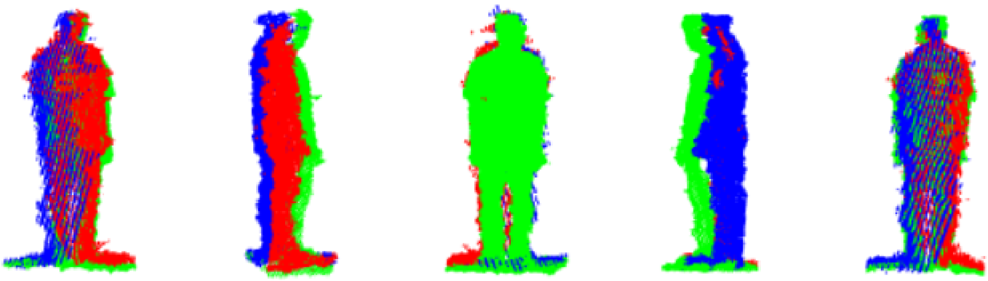


FIGURE 3.6: Reconstructed point clouds of a human body with the same posture. This process was done by applying the proposed Algorithm 1 on noisy depth information. The points corresponding to ground can be easily removed as a post-processing step based on the ground calibration.

3.6.4 Implementation

Our experimental system was executed on a medium-strength computer based on non-optimized C++ code and the two popular open source libraries Point Cloud Library [122] and OpenCV [23]. Depth images in our work were captured with the largest possible resolution (640 × 480 pixels) by the SDK version 1.8. Our system could be expected to reconstruct object point cloud using the Algorithm 3.1 in real-time since our non-optimized code processed each frame in about 0.2 seconds. The execution speed can even be increased by optimizing the source code of memory allocation and management as well as employing the power of parallel processing and/or multi-threading. Our approach could thus be integrated into vision-based systems without affecting significantly computational time.

3.7 Conclusion

Throughout this chapter, an approach which overcomes problems of synchronization has been proposed for reconstructing a 3D object point cloud. Our system can run with a low computational cost with low-cost devices since the proposed configuration employs only a matching-based depth camera together with a few mirrors. The two described algorithms, i.e. combination of reflected points and space carving, are appropriate for working on dynamic (e.g. a walking person on a treadmill for health analysis) as well as static objects, respectively. In summary, our approach can play a significant role in a low-price 3D reconstruction system and can provide acceptable intermediate object models for a wide variety of practical applications in many research fields. In future work, we intend to integrate our method into problems of human gait analysis for health assessment.

Chapter 4

Reconstruction with Kinect 2 (Time-of-Flight) and mirrors

The previous chapter described our approach for 3D reconstruction using the Kinect 1 together with two mirrors. In this chapter, we replace this camera by the next generation one, which integrates the Time-of-Flight technique for depth measurement, to obtain depth maps with higher level of details. Our work proves that the depth estimated by the ToF would be distorted due to the use of mirrors. We then propose a geometry-based method to reduce it in order to achieve a more reliable reconstruction result. This chapter presents the following published journal article:

Nguyen, T.-N., Huynh, H.-H., and Meunier, J. 3d reconstruction with time-of-flight depth camera and multiple mirrors. *IEEE Access*, 6:38106–38114, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2854262. URL https://doi.org/10.1109/ACCESS.2018.2854262

4.1 Abstract

In order to extract more detailed features, many recent practical applications work with 3D models instead of 2D images. However, 3D reconstruction usually requires either multiple cameras or a depth sensor and a turntable. This chapter proposes an approach for performing a 3D reconstruction using only one depth camera together with 2 or more mirrors. Mirrors are employed as virtual depth cameras placed at different positions. All measured depth data are provided in only one frame at each time. Significant depth distortion behind a mirror, which occurred with a standard time-of-flight (ToF) depth sensor, is reduced by removing unreliable points and/or re-estimating better positions for these points. The experiments on easy-to-evaluate geometric objects show that the

proposed approach could play a basic role in reconstructing intermediate 3D object models in practical applications using only cheap devices.

4.2 Introduction

Reconstructing 3D models is an important process in a wide variety of fields including computer animation, medical imaging, computer graphics, etc. A typical strategy for that matter is using a depth camera combined with a turntable where the object is placed on (e.g. [140]). An obvious limitation is that such system is not appropriate to work on dynamic objects (e.g. a walking person) as well as requires prior knowledge such as rotation speed of the turntable. Other studies perform the shape-from-silhouette approach with the support of multiple cameras to retrieve the object visual hull. To overcome the main drawback of this method, i.e. missing concave regions in reconstructed model, other researchers employ a collection of depth sensors [10] and/or stereo cameras [28]. Considering the good accuracies obtained in these experiments, this chapter proposes an approach which reduces the cost of devices as well as avoids unnecessary resource redundancies. In detail, only one depth sensor is required while the others are replaced by mirrors. This work guarantees obtaining depth information from different view points and does not need a synchronization solution as when using multiple depth sensors (e.g. a time server using NTP protocol in [10]). In addition, using multiple depth cameras may cause severe IR interferences.

There are wide varieties of depth sensors together with different estimation techniques such as stereo matching and ToF. In this work, a Microsoft Kinect 2, which uses ToF, is employed because of its cheap cost, good manufactured calibration, and good depth estimation. An approach for 3D reconstruction using mirrors has been performed in [106] with the previous generation of Kinect. The depth map provided by a Kinect 1 is measured based on a structured light technique. Such depth map thus contains less details compared with the one obtained by ToF [147]. Therefore, the Kinect 2 with ToF depth estimation is considered in our work. However, with ToF camera, we need to solve depth measurement ambiguities which occur from unwanted multiple reflections [45]. Such solutions usually require prior knowledge of the ToF camera characteristics (e.g. modulation frequency [35]) or performing low-level modifications as well as using additional devices (e.g. a projector [95]). This chapter presents a simple solution for reducing such ambiguities based on some basic assumptions. Although this method may not solve all depth distortions, it still provides an obvious improvement versus the raw initialized model. It is important to recall that our approach focuses on providing an acceptable 3D model for practical applications instead of reconstructing a detailed object or absolutely removing all depth distortions. Using mirror for 3D reconstruction has been introduced in related works such as [52] and [158]. Unlike our work, these studies focused on alternative implementations of silhouette-based reconstruction using multiple cameras.

Let us introduce briefly the way a ToF sensor measures depth information to provide an overview of possible depth distortions. A ToF depth sensor contains two important parts that are infrared (IR) emitter and receiver. A signal is emitted by the former device and is then received by the latter one. There are two common types of such signal: *high-speed pulse* and *continuous wave*. Distance between the sensor and an object point is approximated as a half of traveled length based on time delay of the pulse or the phase shift between retrieved and emitted waves. Because of this measurement way, if such signal travels in a multipath trajectory, the obtained depth may be significantly changed. This scenario occurs in our configuration with mirrors under several conditions. The details of such depth distortion and our solution are presented in next two sections. Because the Kinect 2 employs a continuous wave modulation, the remaining content of this chapter only mentions this technique.

4.3 Depth distortion behind a mirror

Let us consider a scenario using only one mirror without any environment reflection (e.g. a white wall), an overview of possible returned signals corresponding to a pixel in the depth image is illustrated in Fig. 4.1, in which C and C$_m$ are the real and mirrored camera centers, P and P$_m$ are the considered point and its reflection behind the mirror, P$_K$ is the estimated result of the Kinect, and M is the point where the emitted signal touches the mirror. The term *mirrored point* indicates the image (behind a mirror) of a real point.

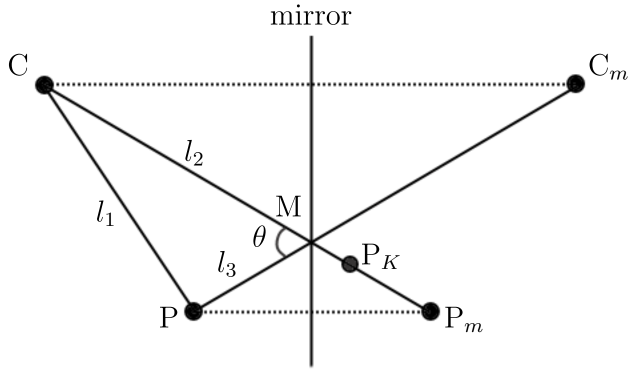


FIGURE 4.1: Depth estimation of a point in front of a mirror and distortion of corresponding mirrored point depth.

As mentioned in the previous section, the distance between the depth sensor and a point is approximated by half of the traveled distance of the signal, i.e.

$$distance(\mathrm{C},\mathrm{P}) = \frac{1}{2}(\|\overrightarrow{\mathrm{CP}}\| + \|\overrightarrow{\mathrm{PC}}\|) = l_1 \tag{4.1}$$

With the reflected point P_m, the trajectory of the corresponding signal is $\overrightarrow{\mathrm{CM}}+\overrightarrow{\mathrm{MP}}+\overrightarrow{\mathrm{PM}}+\overrightarrow{\mathrm{MC}}$, thus the expected distance is $l_2 + l_3$. The value measured by the Kinect, however, is significantly decreased, and a unreliable point P_K is obtained instead of the true point P_m. This distortion occurs because of another signal, which travels along the following way $\overrightarrow{\mathrm{CP}}+\overrightarrow{\mathrm{PM}}+\overrightarrow{\mathrm{MC}}$. We empirically found that if there is a significant difference of length between these two trajectories, the obtained depth value is approximated by the shorter one. This is indicated by the term *geometrical distortion* in this chapter. In the other case, i.e. if the difference is small, the measured depth is affected by multipath ambiguity. We use the term *phase distortion* to denote this effect. In Fig. 4.1, the estimated distance between C and P_m becomes

$$distance(\mathrm{C},\mathrm{P}_m) = \frac{1}{2}(l_1 + l_2 + l_3) \tag{4.2}$$

Due to this distortion, a shape behind a mirror could be very different compared with the original one (e.g. a planar surface becomes curved, see Fig. 4.2 and Appendix A.1). Thanks to the relation between the camera and the mirror, the estimated distance between C and P_K can be used to approximate a better position of P_m.

First, the equation of the mirror surface is determined using some markers placed on it, with their 3D coordinates measured by the Kinect. The position of M is then localized by intersecting the direction $\overrightarrow{\mathrm{CP}_K}$ with the mirror plane to get the length l_2. Let us consider the triangle $\triangle$CMP. The angle θ is determined based on the two vectors $\overrightarrow{\mathrm{MC}}$ and $\overrightarrow{\mathrm{C}_m\mathrm{M}}$. With the estimated depth of P_K, we have:

$$l_1 + l_2 + l_3 = 2\|\overrightarrow{\mathrm{CP}_K}\| \tag{4.3}$$

$$\Leftrightarrow l_1 + l_3 = 2\|\overrightarrow{\mathrm{CP}_K}\| - l_2 = k \tag{4.4}$$

The law of cosines in the triangle $\triangle$CMP leads to the relation

$$l_1^2 = l_2^2 + l_3^2 - 2l_2l_3cos\theta \tag{4.5}$$

By combining eq. (4.4) and (4.5), the length l_3 is obtained as

$$l_3 = \frac{1}{2}.\frac{l_2^2 - k^2}{l_2\mathrm{cos}\theta - k} \tag{4.6}$$

Finally, the point P_m along the straight line CM can be localized together with its real point P. This solution will be tested in Section 4.5.1.

In practical situations, e.g. reconstructing an object with several mirrors, the depth measurement is slightly different. The described depth distortion, however, is useful for removing unreliable measured points. The details of our practical configuration together with the reconstruction of object's point cloud are presented in the next section.

4.4 Unreliable point removal

Let us consider a practical scenario with a Kinect and two mirrors as in Fig. 4.2. According to geometrical optics, the object model can be formed by combining the front part, which is directly seen by the depth sensor, and reflected parts of the back through corresponding mirrors. The 3D cloud measured by a ToF depth sensor, however, contains a lot of unreliable points due to the *geometrical* and *phase* distortions defined in Section 4.3.

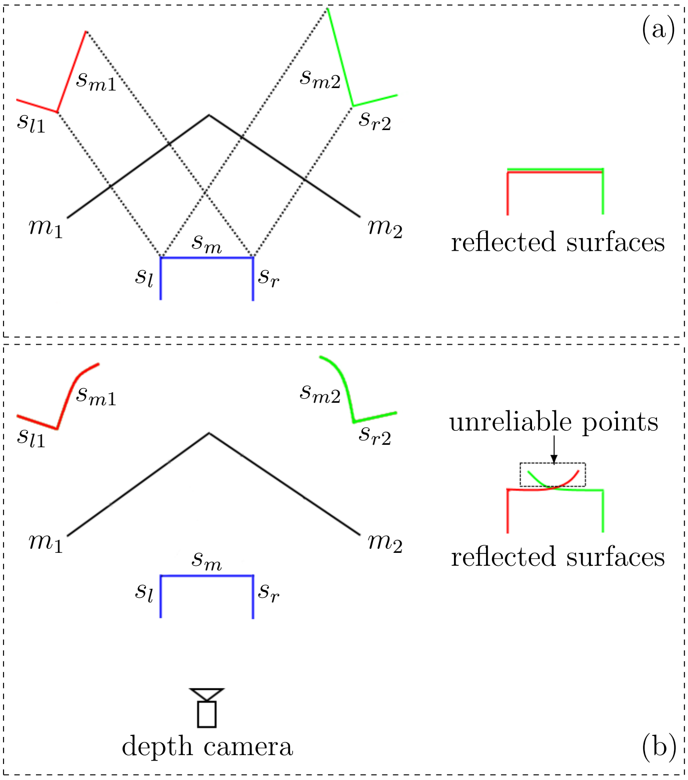


FIGURE 4.2: Practical situation of two reflections with mirrors m_1 and m_2: (a) physical reflection of an object in two mirrors, (b) depth information measured by a ToF sensor. The camera is placed in front of the object and the 3 illustrated object parts (i.e. the 3 surfaces s_l, s_m, and s_r) are not directly seen by the depth sensor (e.g. occluded by front parts (not shown) of the object). The mirrored surfaces of s_r in m_1 as well as s_l in m_2 do not appear in the figure because the depth camera cannot see them due to occlusions.

4.4.1 Geometrical distortion

With a given 3D point P on the back of the object and two mirrors m_1 and m_2 as in Fig. 4.3, the camera provides depth measurements of two mirrored points P_1 and P_2. Because the depth camera C does not directly see the point P, the measured distances of P_1 and P_2 are expected to be $l_1 + l_2$ and $l_3 + l_4$, respectively. The obtained values, however, are only exact for the point P_1, while the corresponding depth of P_2 decreases to P_{K2} with a significant deviation. This distortion occurs because there are two returned signals in the direction $\overrightarrow{P_2C}$ with traveled length $2l_3 + 2l_4$ and $l_1 + l_2 + l_3 + l_4$. The depth information is thus estimated based on the shorter way. In summary, a 3D point P, which is not seen by the depth camera, can create two mirrored points P_1 and P_2 containing at least one reliable point (e.g. P_1 in Fig. 4.3 because $l_1 + l_2 < l_3 + l_4$).

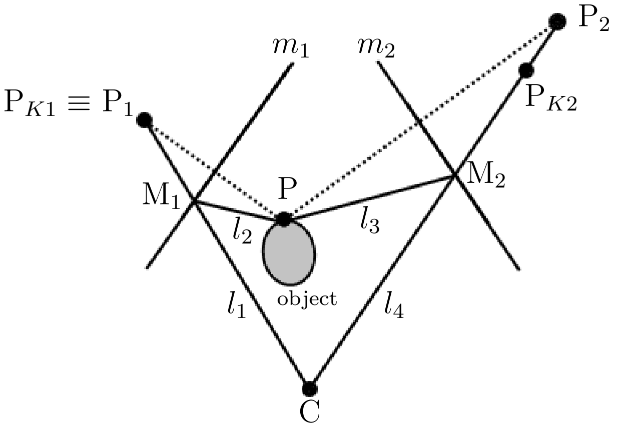


FIGURE 4.3: Depth measurement of a 3D point P in two mirrors m_1 and m_2. Let us note that P is not seen by the depth camera C. Two points P_{K1} and P_{K2} are Kinect measured points of P_1 and P_2, respectively.

4.4.2 Phase distortion

We empirically found that most mirrored points were affected by geometrical distortion, thus our restoration approach for the other distortion is presented as an additional post-processing (see Section 4.4.4 and Appendix A.1).

4.4.3 Reconstructing raw point cloud

In the scenario illustrated in Fig. 4.2, the raw estimated point cloud of the object is obtained by combining two components:

- Points (in front of the object) which are directly seen by the depth camera (not shown in the figure)
- Points (on the back of the object) which are reflected through corresponding mirrors m_1 and m_2

First, a 3D region of the reconstructed object is defined. Let us consider a point P in the cloud mentioned above. If P comes from the first component, i.e. P can be directly seen by the depth sensor, it is a *reliable point* lying on the object surface. If the camera sees a mirrored point P_m of P in an arbitrary mirror, the measured depth of P_m is significantly reduced, but P_m is always behind the mirror. The reflection of P_m is thus in front of this mirror. Our experiments (see Section 4.5.1) show that the distance between this

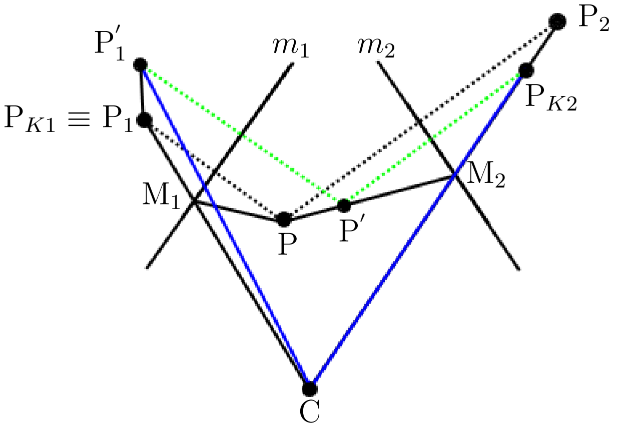


FIGURE 4.4: Reconstruction of a bad-measured Kinect point and its images corresponding to the two mirrors.

reflected point and the corresponding mirror is very small, thus P_m can be easily removed by checking if its reflection is outside of the defined 3D object region. Therefore, there remains two cases which need to be focused on: a 3D point can be seen in only one mirror (e.g. point on surfaces s_l and s_r in Fig. 4.2) or in both mirrors (e.g. point on s_m).

In the first case, the signal corresponding to such point always travels along the shortest way, thus the reflected point is reliable. In the second one, it is important to recall that we have proved that a 3D point, which is not seen by the depth camera, can create two mirrored points containing at least one reliable point. Our goal thus becomes simpler since we just need to remove these false-estimated points.

Our idea for deciding a point in the raw reflected cloud to be removed or be kept is quite simple. Assume that a point P in cloud is recovered (i.e. reflected) from a mirrored point P_i through a mirror m_i with $i \in \{1, 2\}$, the corresponding mirrored point P_j of P in the other mirror is localized. According to the given coordinates of the camera center C, the point P is kept in the cloud if $\mathit{distance}(C, P_i) \leq \mathit{distance}(C, P_j)$, and otherwise is removed. This idea can be proved with the illustration of Fig. 4.4 (extended from Fig. 4.3). Let us assume that P' is the reflected point of P_{K2} through m_2, and P'_1 is the image of P' in m_1. As presented in Section 4.3, the distance between camera center C and estimated point P_{K2} satisfies the following condition:

$$2\|CP_{K2}\| = \|CP_1\| + \|CP_2\| \Rightarrow \|CP_{K2}\| = \|CP_1\| + \|P_2P_{K2}\| \tag{4.7}$$

The three segments $\mathrm{P}_2\mathrm{P}_{K2}$, $\mathrm{PP}^{'}$, and $\mathrm{P}_1\mathrm{P}_1^{'}$ have the same length, thus eq. (4.7) is equivalent to

$$\|\mathrm{CP}_{K2}\| = \|\mathrm{CP}_1\| + \|\mathrm{P}_1\mathrm{P}_1^{'}\| \tag{4.8}$$

According to the triangle inequality [74] in $\triangle\mathrm{CP}_1\mathrm{P}_1^{'}$, we have

$$\|\mathrm{CP}_1\| + \|\mathrm{P}_1\mathrm{P}_1^{'}\| > \|\mathrm{CP}_1^{'}\| \tag{4.9}$$

By combining eq. (4.8) and (4.9), the length of CP_{K2} is always greater than the distance between C and $\mathrm{P}_1^{'}$. In other words, a point in the raw reflected cloud can be considered to be a reliable or unreliable one by checking distances between the camera center to mirrored points behind the two mirrors.

In summary, given a 2D array *pts* (depth image) of 3D points measured by the Kinect, two mirror plane equations mir_1 and mir_2, position of camera center C, and a predefined 3D object region of interest reg, our algorithm for reconstructing a point cloud representing an object is as the Algorithm 4.1.

4.4.4 Increasing point density by space carving

An obvious limitation of the reconstructed object point cloud in Section 4.4.3 is that the farther the object is from a mirror, the larger is the distance between two neighbor 3D points corresponding to this mirror in the obtained cloud. To increase the density of such points, the space carving approach can be applied together with the algorithm described in the previous section. Given a voxel volume V and input components of the algorithm of unreliable point removal, the overall processing is performed as the Algorithm 4.2.

In practical applications as well as when working on specific objects, some additional operations can be integrated into the two presented algorithms to improve reconstruction accuracy such as color filtering and defining object boundary.

As mentioned in the end of Section 4.4.2, a post-processing could be applied to improve the model quality. This processing requires a correspondence of two mirrored points which are created based on one real 3D point, thus it is appropriate to apply the post-processing in the presented space carving approach. This stage can be easily performed based on the eq. (4) (see Appendix A.1). However, let us recall that most estimated points are

Algorithm 4.1: Unreliable point removal

```
Data: $pts$, $mir_1$, $mir_2$, $C$, $reg$
Result: $cloud$
$cloud \leftarrow null$
foreach $point\ P \in pts$ do
    if $P$ inside $reg$ then
        $cloud \leftarrow$ Push ($P$)
    else if $P$ behind $mir_1$ then
        $P_r \leftarrow$ Reflect ($P$, $mir_1$)
        if $P_r$ not inside $reg$ then
            continue                          /* check another point */
        end
        $P_2 \leftarrow$ Reflect ($P_r$, $mir_2$)
        if $CP < CP_2$ then
            $cloud \leftarrow$ Push ($P_r$)              /* reliable point */
        end
    else if $P$ behind $mir_2$ then
        $P_r \leftarrow$ Reflect ($P$, $mir_2$)
        if $P_r$ not inside $reg$ then
            continue                          /* check another point */
        end
        $P_1 \leftarrow$ Reflect ($P_r$, $mir_1$)
        if $CP < CP_1$ then
            $cloud \leftarrow$ Push ($P_r$)              /* reliable point */
        end
    end
end
return $cloud$                       /* Return object point cloud */
```

not affected by this distortion, thus this post-processing is not necessary if our goal is to provide an acceptable intermediate model for practical applications. Moreover, the method in Section 4.4.3 could be integrated into real-time systems while it takes much time to perform the space carving technique.

4.5 Experimental results

This section demonstrates the results of solving depth distortion in the cases of using one and two mirrors. The former experiment was performed by comparing distances between a real 3D point and its raw reflected point as well as the one relocated by our proposed approach [Section 4.3, eq. (4.6)]. In order to obtain a high generalization, a set of points,

Algorithm 4.2: Space-carving-based reconstruction

```
Data: $pts$, $mir_1$, $mir_2$, $C$
Result: $V$
$th \leftarrow th_0$          /* define a threshold of distance deviation */
foreach voxel $P \in V$ do
    $pixel \leftarrow$ Project ($P$)                    /* 3D to 2D projection */
    $P_K \leftarrow$ Get3Dpoint ($pts$, $pixel$)              /* 3D Kinect point */
    if $\|CP_K - CP\| < th$ then
        $V \leftarrow$ Keep ($P$)
        continue                                  /* check next voxel */
    else
        $P_1 \leftarrow$ Reflect ($P$, $mir_1$)
        $P_2 \leftarrow$ Reflect ($P$, $mir_2$)
        $pixel_1 \leftarrow$ Project ($P_1$)
        $pixel_2 \leftarrow$ Project ($P_2$)
        $P_{K1} \leftarrow$ Get3Dpoint ($pts$, $pixel_1$)
        $P_{K2} \leftarrow$ Get3Dpoint ($pts$, $pixel_2$)
        if $CP_1 < CP_2$ and $\|CP_{K1} - CP_1\| < th$ then
            $V \leftarrow$ Keep ($P$)
            continue                              /* check next voxel */
        end
        if $CP_2 < CP_1$ and $\|CP_{K2} - CP_2\| < th$ then
            $V \leftarrow$ Keep ($P$)
            continue                              /* check next voxel */
        end
    end
    $V \leftarrow$ Remove ($P$)
end
return $V$                                  /* Return voxel volume */
```

which consists of markers located on a small flat board, was employed to calculate the distance deviation instead of using only one point at a time, and the board was also placed in front of the mirror at different tilt angles. The latter experiment was evaluated by fitting a surface based on raw reconstructed point cloud as well as voxel volume and then estimating the corresponding error according to prior knowledge of the object shape. In order to simplify the calculation, this work employed two simple objects including a flat board and a cylinder. The testing process was also performed with different distances between the object and the two mirrors.

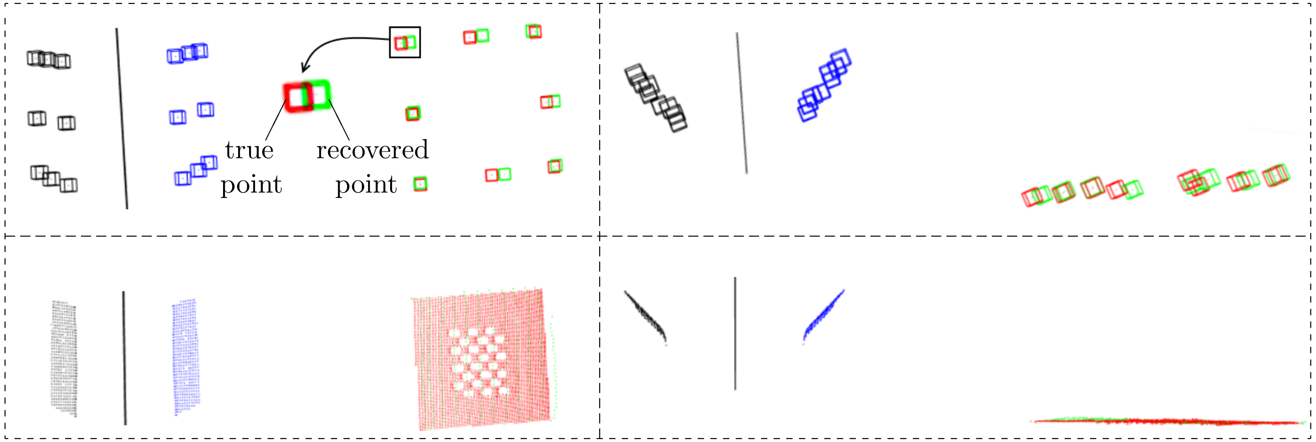


Figure 4.5: From left to right: tested scenes with top view and side view. The top row shows result of solving one-mirror distortion on a set of 8 markers lying on a flat board of size $30cm \times 30cm$ while the second row is the result of recovering all points in the board. In each sub-figure, the mirror is represented by a straight segment, and each marker position is shown as a point bounded by a cube-shape-wireframe for better visualization. The red points are real 3D markers and the black ones are their mirrored points determined in the captured depth map. The blue and green markers respectively indicate the reflection of mirrored points and our recovered ones. Notice that in the bottom-left sub-figure, the board contains some holes because there was a chessboard on the surface, and processing of black pixels was avoided due to low-reflection of black regions.

4.5.1 Solving depth distortion with one mirror

For each real marker P on a flat pattern placed in front of the mirror, our processing flow in this experiment consists of the following steps (see Fig. 4.1): (a) reflecting P to get the true position of its image P_m behind the mirror, (b) determining the corresponding measured Kinect point P_K, (c) re-estimating a corrected point P_C of P_K ($\mathrm{P}_C \equiv \mathrm{P}_m$ in the ideal case), and (d) calculating $distance(\mathrm{P}, \mathrm{P}_{Cm})$ and $distance(\mathrm{P}, \mathrm{P}_{Km})$ where P_{Cm} and P_{Km} are reflections of P_C and P_K through the mirror, respectively. In summary, a set of n corners provides n pairs of such values. Finally, average distances are compared together to evaluate the proposed solution.

Processing a set of markers as well as all points on the flat board are illustrated in Fig. 4.5. In the top row, the recovered points were almost at their corresponding true points though there were significant distance deviations in the Kinect measurement. In the bottom row, the points provided by our solution and the true points also fit a plane. The small position deviations of our recovered points in Fig. 4.5 come from the following reasons. First, the mirror was not an absolute planar surface, a point displacement might thus occur. Besides, this experiment was performed on raw captured data without any improvement (e.g. depth smoothing or enhancement). In addition, different 3D positions could be mixed into one point by Kinect due to the low resolution of the IR camera (512 × 424 pixels). To overcome these limitations, a depth improvement procedure could be applied (e.g. [92]), and a high-resolution camera could also be employed as an additional view (e.g. mapping between color and depth cameras of Kinect to investigate a higher density of recovered points).

Figure 4.6 shows experimental results corresponding to 12 different pattern poses in front of the mirror. It is obvious to see that distance deviations between true points and reflected ones were significantly reduced by our proposed solution.

4.5.2 Reducing distortion in the case of two mirrors

In this experiment, the angle between two mirrors was about 120 degrees. The distance from a tested object to mirrors was defined as the mean of all distances between the final reconstructed object points and the two mirrors. Given knowledge about the object

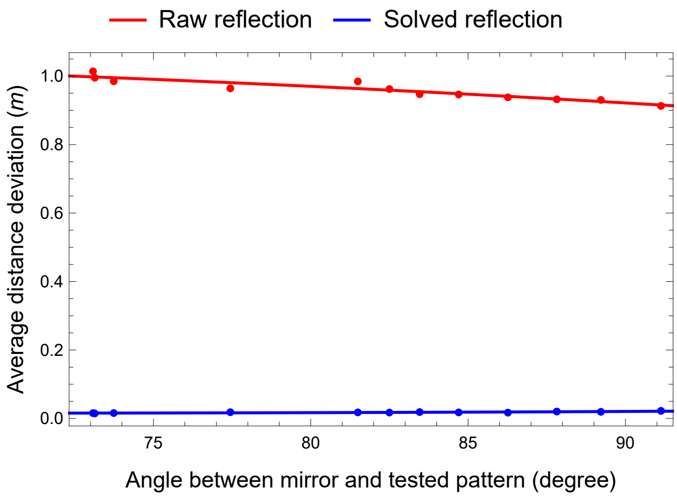


FIGURE 4.6: Measured reflection errors before and after applying our solution, in which deviation values were decreased about 53 times (0.959 and 0.018 on average, respectively).

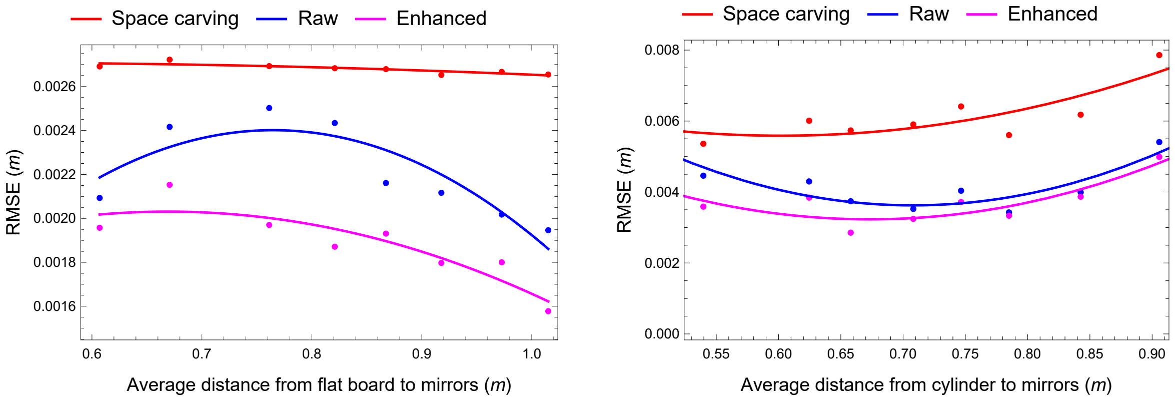


FIGURE 4.7: Reconstruction errors [when experimenting with a flat board (left) and a cylinder (right)] of three types of clouds: raw, distortion removal, and space carving. The cylinder radius, which was manually measured, was 150 *mm* and the average radius of the reconstructed clouds was 147.4 *mm*.

shape (either plane or cylinder), the evaluation was performed by fitting a surface based on RANSAC [59] and estimating root-mean-square errors (RMSE). Our experimental results when testing these two objects are shown in Fig. 4.7. Fitting errors were reduced after applying our approach on raw reconstructed point cloud. Notice that the error corresponding to the space carving method was always larger than the two others because of object's thicker borders. Measured errors were less than 1 *cm*. The cylinder radius was 150 *mm*.

A visual comparison of reconstructed point clouds of a cylinder before and after performing our method is also presented in Fig. 4.8. The proposed approach removed a large number of noisy points from the raw reconstructed models.

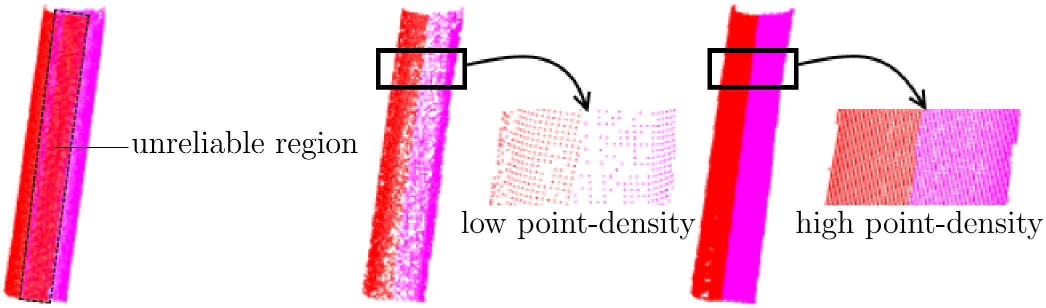


FIGURE 4.8: From left to right: raw cloud, cloud after removing unreliable points, and space carving. Points directly seen by the Kinect are not shown in this figure since they are not affected by any of mentioned distortions.

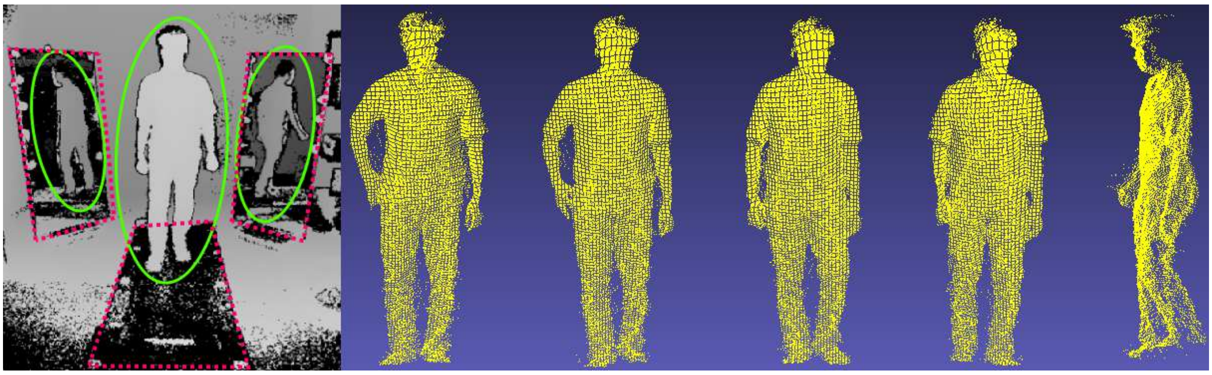


FIGURE 4.9: Left: our realistic setup of a 3D reconstruction system for the task of gait analysis including a treadmill and two mirrors (highlighted by dotted red rectangles). Right: reconstructed point clouds corresponding to 4 nearby poses of a walking gait, and the last cloud is the 4^{th} one seen from side view. These point clouds were acquired at 13 fps using the computer mentioned in Section 4.5.3. These clouds are extracted from our huge dataset (nearly 100,000 postures) of human walking gaits. Details of data acquisition is clearly described in [98].

A visualization of point clouds representing a human body with different postures is also presented in Fig. 4.9. These clouds are reconstructed by the algorithm of unreliable point removal presented in Section 4.4.3. The figure shows that it is reasonable to expect that our approach could be used to provide intermediate (real-time) models in systems which process 3D information. A huge dataset (nearly 100,000 samples) of such point clouds representing human walking gaits performed on a treadmill is also available online[1].

Figure 4.10 shows a comparison of reconstruction error (RMSE) between our system and the similar setup in [106], where a Kinect 1 with structured-light depth estimation was employed instead of a Kinect 2. Both reconstructions were performed on the same cylinder, and the statistical information presented in Fig. 4.10 was calculated on different

[1] http://www.iro.umontreal.ca/~labimage/GaitDataset

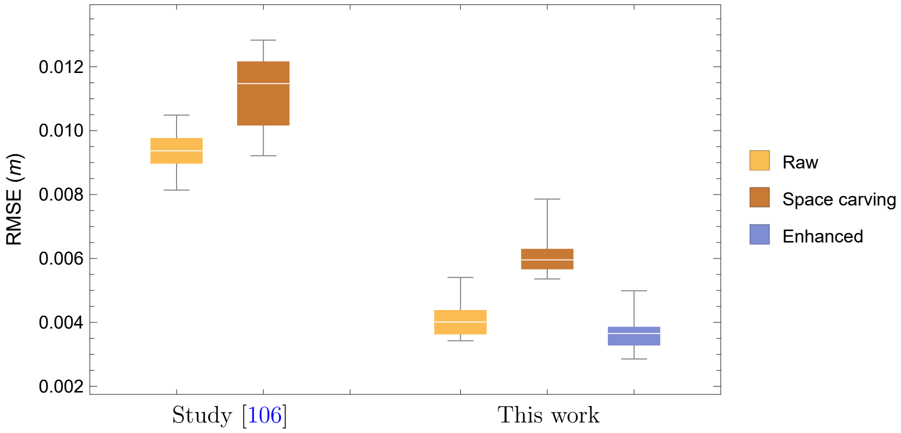


FIGURE 4.10: Reconstruction errors corresponding to our work and the study [106]. The comparison is performed on three types of clouds: raw reflection, distortion removal (only our work), and space carving.

distances between the cylinder and mirrors. The study [106] also provided point clouds corresponding to raw reflection and space carving. Notice that the depth distortion, which has been dealt with in our study, does not occur in the setup [106]. This comparison shows that our system with a Kinect 2 provided better point clouds. This is because the depth map of Kinect 1 is noisier and has less details compared with the next generation [147].

Finally, let us note that our algorithm makes a trade off between the simplicity of processing flow and a constraint in scene configuration. For example, in the case where the object in Fig. 4.2(b) is placed nearer the mirror m_1 (large deviation of distances between the object and each mirror), the proposed algorithm might fail to reconstruct the surface s_r from s_{r2}. In detail, the idea of checking point reliability in Section 4.4.3 is sometimes not appropriate for object points which are seen in only one mirror. This drawback, however, could be easily avoided by placing the object near the center of a *balanced* (approximately) configuration. All our experiments satisfy this constraint without any complicated additional processing. In addition, we should notice that if the setup contains more than 2 mirrors, the depth distortion would be more complicated due to the increasing number of unwanted reflections. Such setup may even reduce the quality of reconstructed 3D point clouds.

4.5.3 Implementation

Our system was built on a medium-strength laptop using C++ (non-optimized code) and the two open source libraries OpenCV [23] and Point Cloud Library [122]. All Kinect depth images in our experiments were captured with a resolution of 512 × 424 pixels. The process of reconstructing raw point cloud (as in Section 4.4.3) was performed with an average speed of 0.07 seconds per frame. This processing time could be significantly reduced with the support of parallel (and multi-threading) programming. The proposed approach thus could be expected to be appropriate for creating a real-time reconstruction system.

4.6 Conclusion

Throughout this chapter, a new approach for reconstructing a 3D object using only one ToF depth sensor together with mirrors has been presented. An overview of depth distortion occurring with one and two mirrors and corresponding solutions are also mentioned. Beside avoiding the problem of synchronization (i.e. all depth data from different viewing directions are provided by only one Kinect) and possible severe IR interferences caused by multiple depth cameras, our method can be applied on dynamic objects (e.g. a walking person). The experiments and evaluations show that the proposed approach improves significantly the quality of Kinect depth estimation. In summary, our method can serve as a basic system for cheap 3D reconstruction as well as for providing intermediate object models in practical applications. In future work, we intend to use the reconstructed data for various applications, such as human gait analysis and assessment.

Chapter 5

Gait Symmetry Assessment based on Cross-Correlation

This chapter presents our preliminary approach for gait symmetry assessment given a sequence of 3D point clouds reconstructed using the method in the previous chapter. This work has been published as the following journal article:

Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Human gait symmetry assessment using a depth camera and mirrors. *Computers in Biology and Medicine*, 101:174 – 183, 2018. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2018.08.021. URL http://www.sciencedirect.com/science/article/pii/S0010482518302415

5.1 Abstract

It is proposed in this chapter a reliable approach for human gait symmetry assessment using a Time-of-Flight (ToF) depth camera and two mirrors. The setup formed from these devices provides a sequence of 3D point clouds that is the input of our system. A cylindrical histogram is estimated for describing the posture in each point cloud. The sequence of such histograms is then separated into two sequences of sub-histograms representing two half-bodies. A cross-correlation technique is finally applied to provide values describing gait symmetry indices. The evaluation was performed on 9 different gait types to demonstrate the ability of our approach in assessing gait symmetry. A comparison between our system and related methods, that employ different input data types, is also provided.

5.2 Introduction

The problem of assessing human gait has received a great attention in the literature since gait analysis is a key component of health diagnosis. Marker-based and multi-camera systems are widely employed to deal with this problem. Collections of wearable devices (e.g. inertial systems using accelerometer [22, 34], gyroscope [55, 56], and/or magnetometer [21, 80]) are also considered to provide information about pre-selected body parts. However, such systems are less accessible due to their cost, size, need for accurate sensors/markers placement on the body and/or the necessity of trained staff to operate them. To alleviate these issues, we focus on a system of gait analysis which employs only one depth sensor. The principle is similar to a multi-camera system, but the collection of cameras are replaced by one depth sensor and mirrors. Each mirror in our setup plays the role of a camera which captures the scene at a different viewpoint. Since we use only one camera, the task of synchronization when working with multi-camera systems can thus be avoided, and the cost and complexity of devices are reduced. Our approach is especially appropriate for non-hospital settings (e.g. small clinics) and may complement more precise instruments (motion capture or inertial systems). Our system could enable clinicians to perform more frequent screening or follow-up of patient prior to more sophisticate tests involving gold standard systems in a specialized gait analysis lab or hospital when necessary.

In order to simplify the setup, recent vision-based studies used a color or depth camera to perform gait analysis. The input of such systems is thus either the subject's silhouette or depth map. Many gait signatures have been proposed based on the former input type such as Gait Energy Image (GEI) [57], Motion History Image (MHI) [32], or Active Energy Image (AEI) [87]. Typically they are computed based on a side view camera and are usually applied on the problem of human identification. In order to deal with pathological gaits, the input sequence of silhouettes needs more elaborate processing. In the work [101], the input sequence of silhouettes was separated into consecutive sub-sequences corresponding to gait cycles. The feature extraction was applied on each individual silhouette and the gait assessment was performed based on a combination of such features in each sub-sequence. Instead of capturing a side view of the subject, the authors in [15, 16] put the camera in front of a walking person and tried to detect unusual movement. The balance of the subject was encoded based on a sequence of lattices applied on the captured silhouettes. A feature vector was then estimated for each lattice according to a predefined set of points, and the characteristic representing the whole motion was formed by concatenating such vectors. This step of concatenation is to incorporate the temporal context into the classification with a Support Vector Machine (SVM). A common limitation of such silhouette-based approaches is the reduction of data dimension since the 3D scene is represented by 2D images. In order to overcome this drawback, a depth camera is often employed. One of the devices that are widely used is the Microsoft Kinect. Beside its low price, this camera provides a built-in functionality of human skeleton localization, estimated in each single depth frame [132, 133]. Such skeletal information is useful for gait-related problems such as abnormal gait detection [102], gait-based recognition [72], and pathological gait analysis [20]. A limitation of skeleton-based approaches is that the skeleton may be deformed due to self-occlusions in the depth map. Unfortunately, such problem usually occurs in pathological gaits [12, 112].

For that reason, other researchers have used depth images without skeleton fitting to perform gait assessment. Auvinet *et al.* [12] proposed an asymmetry index obtained with a depth camera (Microsoft Kinect). It is based on the longitudinal spatial difference between a specific zone of the left and the right legs at comparable times within their respective step cycle. Mean depth images representing the most representative (averaged) gait cycle for each subject are used to decrease the influence of noise. However, this method is limited to a small part of the lower limbs and requires the detection of gait cycles. Nguyen *et al.* [97] have also employed successfully (enhanced) depth maps for gait assessment using a weighted combination of a PoI-score, based on depth map key points, and a LoPS-score describing a measurement of body balance from the body silhouette. However, their method was still limited to a partial view of the body and basic features.

Taking all this into account, we present an original approach that estimates an index of human gait symmetry without requiring skeleton extraction or gait cycle detection. To improve the performance, the input of our system is a sequence of 3D point clouds of the whole body obtained with a combination of a depth camera and two mirrors. Cylindrical histograms corresponding to point clouds are then computed and analysed for left-right symmetry for subjects walking on a treadmill to obtain their symmetry index. The remaining of this chapter is organized as follow: Section 5.3 describes details of our method including the setup, point cloud formation, feature extraction, and gait symmetry

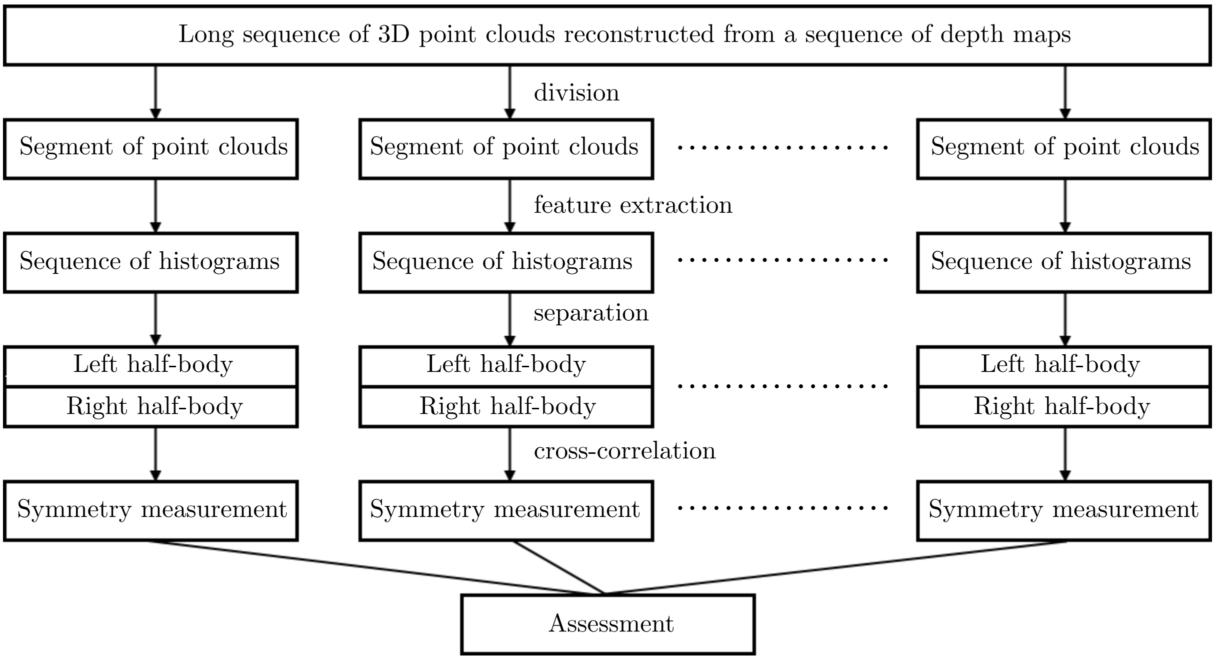


FIGURE 5.1: Flowchart of our processing.

assessment; our experiments, evaluation, and discussion are presented in Section 5.4, and Section 5.5 gives the conclusion.

5.3 Proposed method

In order to give a visual understanding, an overview of the proposed approach is shown in Fig. 5.1.

5.3.1 Point cloud formation

Beside a ToF depth camera and two mirrors, our setup also employs a treadmill where each subject performs his/her walking gait. The ToF camera is put in front of the subject and the two mirrors are behind so that the walking person nearly stands at the center [see Fig. 5.2(a)]. An example of such captured depth map is presented in Fig. 5.3.

There are two popular types of depth sensor that are distinguished based on the scheme of depth estimation: structured light (SL) and Time-of-Flight (ToF) [58]. In our work, the second type was employed because it is more accurate [147] and consequently its point cloud has a higher level of details compared with the first one.

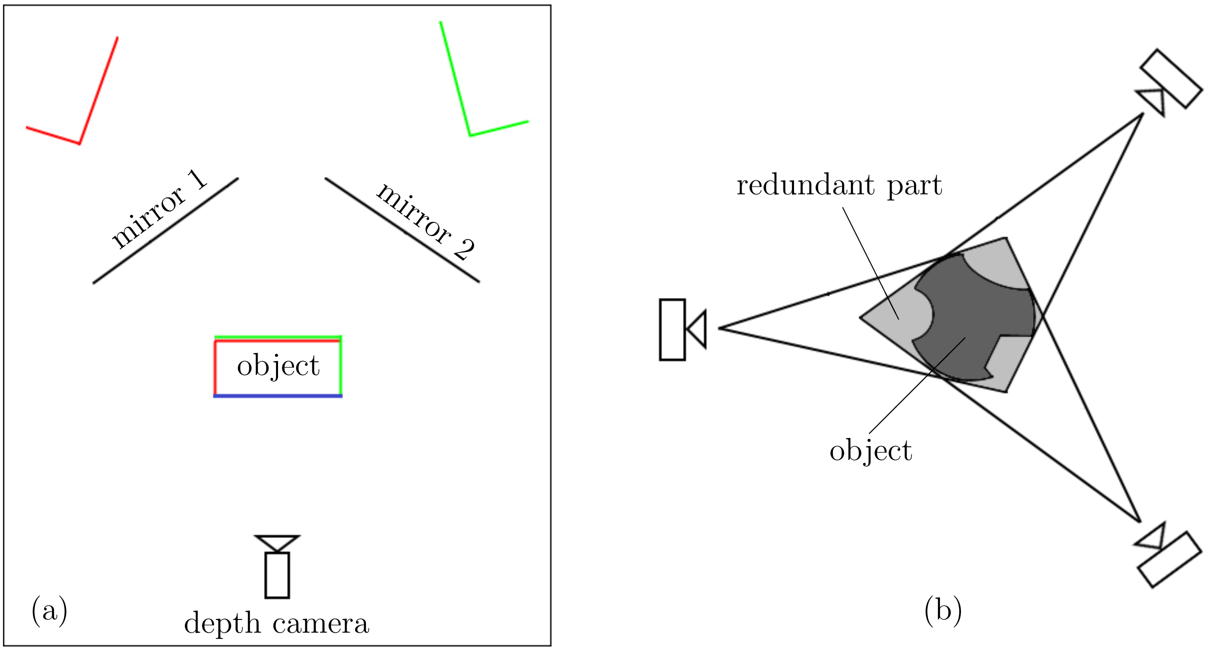


FIGURE 5.2: (a) Basic principle of the depth camera system with mirrors. The depth information visible by the depth camera (blue surface of the object) is complemented by the reflected depth information from the two mirrors (red and green surfaces) to obtain the full 3D reconstruction of the object. Notice that in practice, some unreliable points must be removed due to multiple reflections with ToF camera (see [107]). (b) Visual hull reconstructed from silhouettes by 3 cameras. Beside the true object (dark-gray region), the obtained result also contains redundant parts (light-gray regions). These redundancies could be removed when performing the reconstruction according to 3 depth maps (adapted from [10]).

As shown in Fig. 5.3, each captured depth map provides subject's images from 3 different view points. In practice, the 3D reconstruction of a point cloud representing a subject's posture could also be performed when the depth camera is replaced by a color one. However, the process of reconstruction based on such data produces an object (visual hull) that is bigger, less accurate and contains redundancies as illustrated in Fig. 5.2(b). Therefore employing a depth camera in our setup is advantageous to provide a better model of 3D information.

Let us briefly describe the formation of a 3D point cloud from each depth map captured by a depth camera in our work. According to the example shown in Fig. 5.3, a depth map contains 3 partial surfaces of the subject. A point cloud representing the walking person can thus be formed by combining (a) the direct cloud (highlighted by the middle ellipse) and (b) reflections of two indirect ones (smaller ellipses), which are behind the mirrors [106, 107]. The reflection of the two clouds is performed based on the equations of the two mirror planes that are determined from the positions of markers mounted on

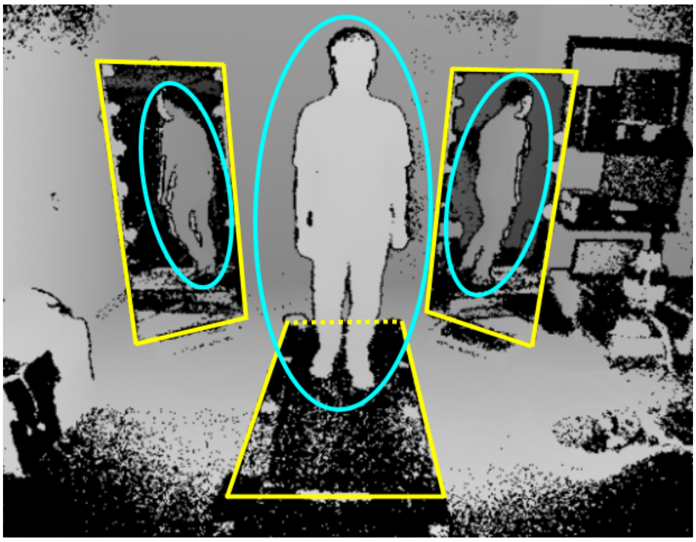


FIGURE 5.3: A depth map captured by our system, in which there are 3 collections of subject's pixels (highlighted by cyan ellipses). The two mirrors and the treadmill are highlighted with yellow rectangles.

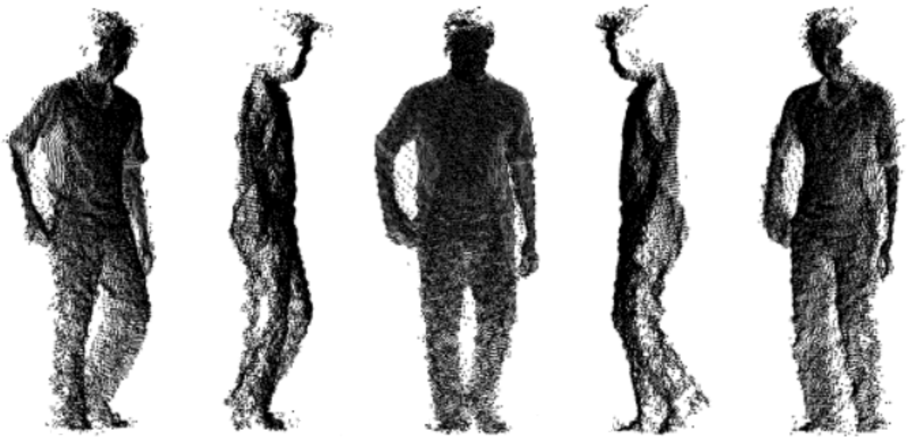


FIGURE 5.4: A point cloud obtained in our setup seen from different view points.

the mirror surfaces. We used the method described in [107] because it was specifically designed for ToF camera and is robust to unreliable points caused by unwanted multiple reflections. The reported reconstruction RMS errors obtained when experimenting on geometric objects were less than 5 *mm*. Figure 5.4 illustrates an example of a 3D point cloud obtained with the setup in Fig. 5.3.

5.3.2 Feature extraction

In order to perform gait symmetry assessment, we separate the entire point cloud with a sagittal plane (perpendicular to the z-axis (coordinate system in Fig. 5.5) and passing through the point cloud centroid) into two non-overlapping half-point-clouds corresponding to the left and right half-bodies. In practice, each individual point cloud is processed to obtain a cylindrical histogram, and then the histogram is vertically split into two sub-histograms representing two half-bodies (see below).

5.3.2.1 Coordinate system transformation

Let us notice that the point cloud is initially computed in the camera space (x_c, y_c, z_c). Therefore, to facilitate the computation of the cylindrical histogram, we need a rigid transformation from the camera coordinate system to the object (body) coordinate system. The latter is defined by its origin assigned to the centroid of the body 3D point cloud, the y-axis normal to the ground (treadmill), the x-axis along the walking direction and the z-axis in the left to right direction (see Fig. 5.5). The y-axis is easily estimated as the normal to the treadmill plane obtained during calibration using a few markers (a set of 4 markers was employed in our experiments in Section 5.4). The walking direction (x-axis) is determined from the vector between two appropriate markers on the treadmill. The remaining dimension (z-axis) is estimated by performing a cross product.

5.3.2.2 Cylindrical histogram estimation

Once the subject's point cloud corresponding to each depth frame has been transformed, its symmetrical characteristic is then extracted with a cylindrical histogram. In detail, a cylinder is estimated with the main axis coinciding with the y-axis of the body coordinate system, and the top and bottom surfaces going through the highest and lowest points along this dimension. The cylinder's radius is long enough to guarantee that the entire point cloud is within the cylinder.

Given a cloud P of n 3D points and the size $h \times w$ of a target cylindrical histogram (see Fig. 5.6), the sector's zero-based index of each point $P^{(i)}$ is determined as

$$\begin{cases} h^{(i)} = min\Big(\Big\lfloor h(max_y - P_y^{(i)})(max_y - min_y)^{-1} \Big\rfloor, h - 1\Big) \\ w^{(i)} = \Big\lfloor \frac{w}{2\pi}\{[2\pi + sgn(\vec{v}_z^{(i)})cos^{-1}(\frac{\vec{v}_x^{(i)}}{\|\vec{v}^{(i)}\|})] \bmod (2\pi)\} \Big\rfloor \end{cases} \tag{5.1}$$

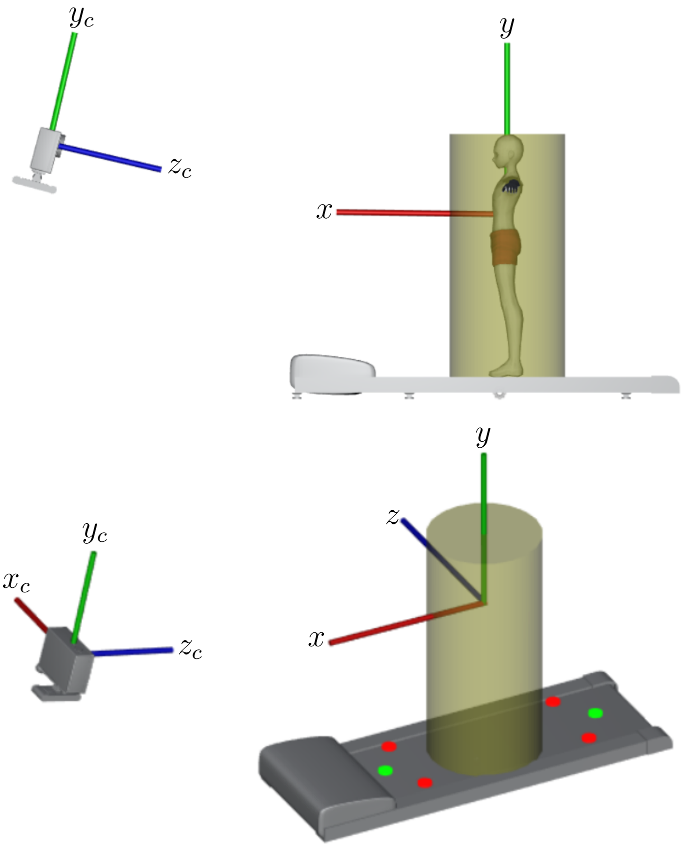


FIGURE 5.5: Visualizations of our scene from two different view points that show the camera coordinate system and the body coordinate system used for matching a cylinder with a point cloud. They are right-handed. The four red circles indicate the markers used to estimate the treadmill plane, and the two green markers are to determine the unidirectional belt motion.

where max_y and min_y respectively indicates the y-coordinate of highest and lowest points in the cloud P along the y-axis, $\lfloor \circ \rfloor$ is the floor function, $P_y^{(i)}$ is the y value of point $P^{(i)}$, $sgn(\circ)$ is the sign function, and $\vec{v}^{(i)}$ is a 2D vector computed from the y-axis to the point $P^{(i)}$. Notice that the notation $\vec{v}_z^{(i)}$ in eq. (5.1) is the z coordinate of $\vec{v}^{(i)}$. The subscript z is to indicate the axis used in this calculation. The *min* function in eq. (5.1) is to guarantee that the output index is in the range $[0, h-1]$.

Although a cylinder is employed to estimate a histogram for each point cloud, the representation of such histogram is flat, i.e. a matrix of size $h \times w$. The correspondence between a histogram's bin and its original cylinder's sector is illustrated in Fig. 5.6. As illustrated in Fig. 5.7, the head is aligned at the center of the cylindrical histogram after performing the estimation. Notice that a slight *rotation* of the cylinder might be necessary to ensure that the body is well centered in the cylindrical histogram depending on the camera-to-body rigid transformation accuracy (see above).

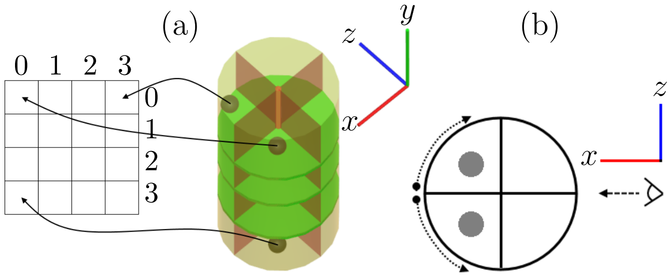


FIGURE 5.6: Mapping from cylindrical sectors to histogram's bins. The sub-figure (a) shows a 3D visualization. The histogram can be considered as a flattened cylinder seen from a specific view point as the sub-figure (b). In this simplified representation, the histogram's size is 4 × 4 corresponding to 16 sectors.

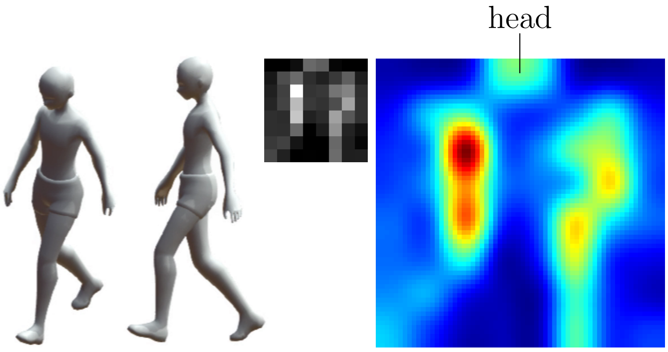
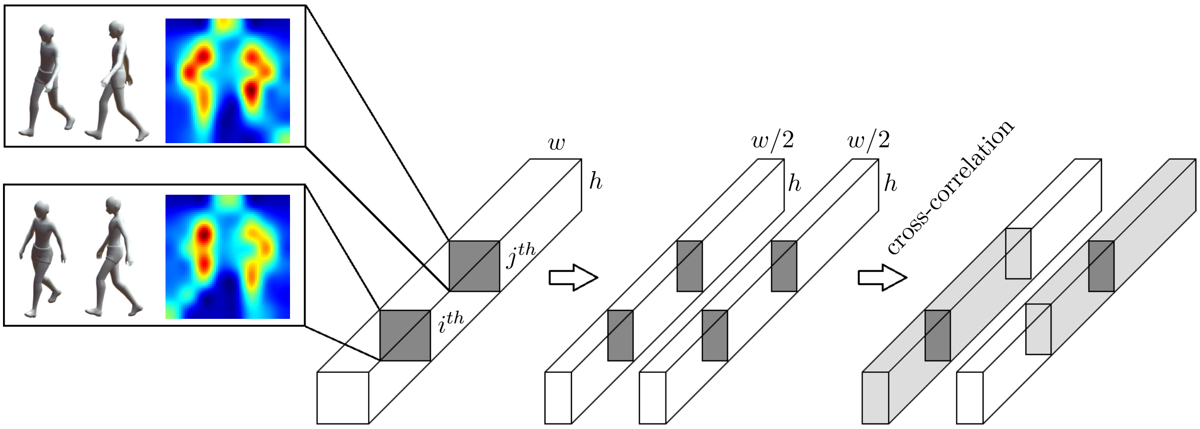


FIGURE 5.7: Example of flattened cylindrical histogram. The original histogram (gray image) of size 8×8 is scaled and is represented as a heat map for a better visualization. We can explicitly see the posture's self-symmetry since the head is at the center of the histogram.

5.3.3 Gait symmetry assessment

Similarly to related studies on gait analysis (e.g. [12, 16, 97, 102]), the assessment of gait symmetry in our system also considers the temporal factor. In detail, the value measuring the gait symmetry is estimated on a sequence of consecutive histograms. Symmetry can thus be measured by vertically separating (equivalent to a sagittal plane passing through the point cloud centroid) each histogram into two sub-histograms corresponding to two half-bodies (left and right). In other words, a sequence of histograms of size $h \times w$ becomes two sequences of sub-histograms of size $h \times 0.5w$. According to the nature of normal walking gait, there is a shifting along the time axis between a left sub-histogram and its corresponding symmetric right one. Therefore our method employs a cross-correlation technique [137] to measure the gait symmetry index. A good symmetry occurs if each left sub-histogram is similar to the horizontal flip version of the (shifted) right sub-histogram (Fig. 5.8).



(a) Sequence of histograms (b) Half-body sequences (c) Best matching of different shiftings

FIGURE 5.8: Symmetry assessment for a sequence of histograms. We say that the i^{th} and j^{th} histograms have a good symmetry since each one and the horizontal flip version of the other are quite similar. The heat maps in this figure are enhanced (for visualization) from actual histograms estimated from 3D point clouds in our experiments, and the 3D models are used for illustrating the corresponding postures. Instead of performing the cross-correlation on the input sequence and its clone, we process directly on two sequences corresponding to half-bodies to reduce the number of calculations and memory requirement. Notice that an input sequence may contain similar histograms because walking is a periodic motion.

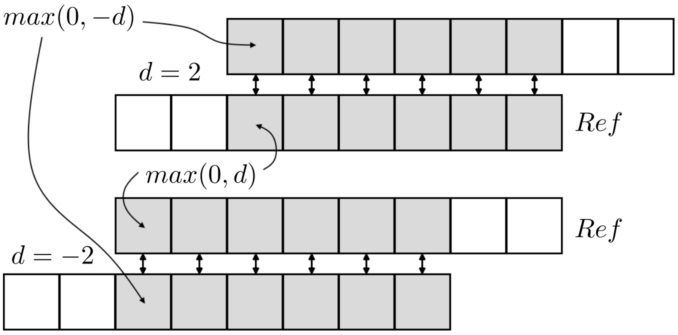


FIGURE 5.9: Correlation between two sequences corresponding to positive and negative shifting values d, and indices of beginning positions. The notation Ref indicates the reference (left sequence in our work). In these two examples, the lengths of each input sequence and the common one are 8 and 6, respectively.

The processing of this stage is as follows. The input is a sequence of histograms. Although many related studies tried to process on gait cycles, our assessment is performed on consecutive (i.e. non-overlapping) sub-sequences (or segments) that have the same length. There are several reasons leading to our choice: (1) gait cycle determination would be difficult to perform when working on pathological gaits, (2) the symmetry can be measured well by dealing with the mentioned shifting on an arbitrary (long enough) sequence of histograms, and (3) sub-sequences do not need to have common properties (e.g. similar beginning and ending postures as in [102] or [12]) because we do not focus on training a model representing the gait. Each sub-sequence is then separated into two sequences of left and right sub-histograms. We can expect that by assigning a sequence as the reference and shifting the other with an appropriate delay, the two registered sub-sequences would have a good symmetry (see Fig. 5.8). Because such suitable delay is various with different subjects, we perform the shifting with a set of delays and choose the best match. Given two sequences of sub-histograms L and R^f (R horizontally flipped) of length l representing two half-bodies, a set of shifting delays D, the symmetry index S is measured as

$$S\big(L, R, D\big) = min\Big(\Big\{\frac{1}{l-|d|}\sum_{i=0}^{l-|d|-1}\mathrm{Diff}\big(L_{max(0,d)+i}, R^f_{max(0,-d)+i}\big) \mid d \in D\Big\}\Big) \tag{5.2}$$

Since the Diff function estimates the distance between two sub-histograms (L_1 norm in our experiments), the *min* function thus provides the best matching. Notice that the left segment is assigned as the reference, and the set D contains both negative and positive values indicating the shifting direction of the other segment (see Fig. 5.9). In the implementation, L and R^f can be defined as arrays of histograms, and their subscript in eq. (5.2) indicates the index (starting at zero). At the end of this stage, the system provides a sequence of scores measuring the gait symmetry corresponding to consecutive segments.

5.4 Experiments

5.4.1 Data acquisition

Our experiments were performed on 9 different gait types consisting of normal walking gaits and 8 simulated asymmetrical (so-called abnormal) ones. These abnormal gaits were simulated by either padding a sole with a thickness of 5/10/15-centimeters under one foot or attaching a weight (4 kilograms) to one ankle. We use the notations L|5cm, L|10cm, L|15cm, and L|4kg to indicate these abnormal gaits with left leg, and so on for the other leg. Such set up can provide gaits having a higher level of asymmetry compared with normal walking ones. A Kinect 2 was employed for data acquisition since it uses ToF for depth measurement and had a low price. There were 9 volunteers that performed the 9 mentioned walking gaits, in which each motion was captured as 1200 continuous frames with a frame rate of 13 fps. The treadmill speed was set at 1.28 km/h. In order to provide a comparison with related approaches, we also captured other data types including skeleton and silhouette using built-in functionalities of the Kinect 2. Therefore, each walking gait of a volunteer is represented by 1200 point clouds, 1200 skeletons, 1200 depth maps, and 1200 silhouettes [98][1]. These experimental procedures involving human subjects were approved by the Institutional Review Board (IRB).

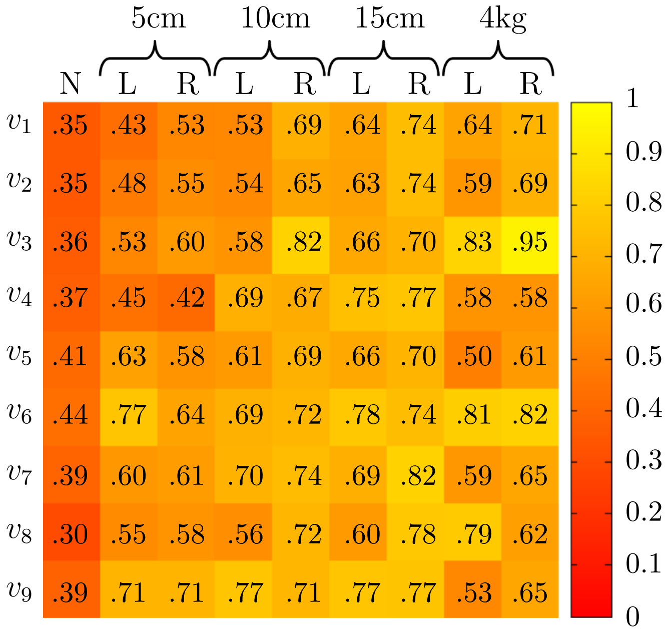
5.4.2 System parameters

As mentioned in Section 5.3.3, the input sequence of point clouds is segmented into non-overlapping segments. In our experiments, each input sequence was separated into 10 segments of length 120 (about 9 seconds), the corresponding output was thus a vector of 10 elements measuring the gait symmetry. The size of cylindrical histogram was 16×16, so each half-body volume in Fig. 5.8 had a size of $[16 \times 8 \times 120]$. The L_1 norm was used for measuring the distance [the term Diff in eq. (5.2)] between two normalized histograms, i.e. dividing each bin value by the sum. The shifting delays $d \in D$ were in the range $[-50, 50]$ to guarantee that the length of the common sub-sequence would be greater than a half of input length. Let us notice that 16×16 is *not* the optimal size of cylindrical histograms. This is just an arbitrarily selected value for our experiments. The effect of that hyperparameter will be discussed in Section 5.4.5.

5.4.3 Testing results

Since our system returned 10 measurement values (corresponding to 10 segments of length 120) for each input sequence of point clouds, their mean can be used as an index of gait symmetry. The experimental results are shown in Fig. 5.10. The mean values were in the range between 0.30 and 0.44 for normal gaits, and higher measures for the asymmetrical ones. Therefore, considering the returned estimation of an arbitrary gait and that range may allow gait symmetry assessment. However, that range is formed from a set of volunteers, an asymmetrical gait of a subject may thus have an estimation falling inside the normal range of other subjects though this value is still higher than the measure of normal gait with the same subject. This case happened for the R|5cm gait of

[1] http://www.iro.umontreal.ca/~labimage/GaitDataset



		5cm		10cm		15cm		4kg	
	N	L	R	L	R	L	R	L	R
v_1	.35	.43	.53	.53	.69	.64	.74	.64	.71
v_2	.35	.48	.55	.54	.65	.63	.74	.59	.69
v_3	.36	.53	.60	.58	.82	.66	.70	.83	.95
v_4	.37	.45	.42	.69	.67	.75	.77	.58	.58
v_5	.41	.63	.58	.61	.69	.66	.70	.50	.61
v_6	.44	.77	.64	.69	.72	.78	.74	.81	.82
v_7	.39	.60	.61	.70	.74	.69	.82	.59	.65
v_8	.30	.55	.58	.56	.72	.60	.78	.79	.62
v_9	.39	.71	.71	.77	.71	.77	.77	.53	.65

FIGURE 5.10: Mean values of 10 measurements provided by our system for each gait of each volunteer. The notation N indicates normal gaits, L and R respectively represent left and right legs, and v_i is the i^{th} volunteer.

the 4^{th} volunteer which was lower than the normal gait of the 6^{th} volunteer. Therefore, within-subject analysis should be considered to increase the confidence of the symmetry assessment. Let us see more details of our experimental results in Fig. 5.11 instead of only mean values. With most subjects, the measured values tended to decrease when the asymmetry reduces (e.g. L|10cm compared with L|15cm). This means that our system could be used to assess the recovery of patients after a (knee, hip, etc.) surgery, during a musculoskeletal treatment or after a stroke for instance. In summary, the assessment of gait symmetry can be performed by checking estimated measures with a specific range and confirming the decision based on recent changes of these values (e.g. day by day). Let us notice again that considering only the normal range may not be sufficient since the actual gait symmetry depends on various factors such as health, physical body, and even walking habit. Therefore checking the convergence of symmetry measurements helps us to confirm the normality of patient's gaits.

5.4.4 Comparison with other related methods

In order to compare the gait-related information gained when exploiting 3D point clouds with other data types, we also performed experiments on the skeletons and silhouettes

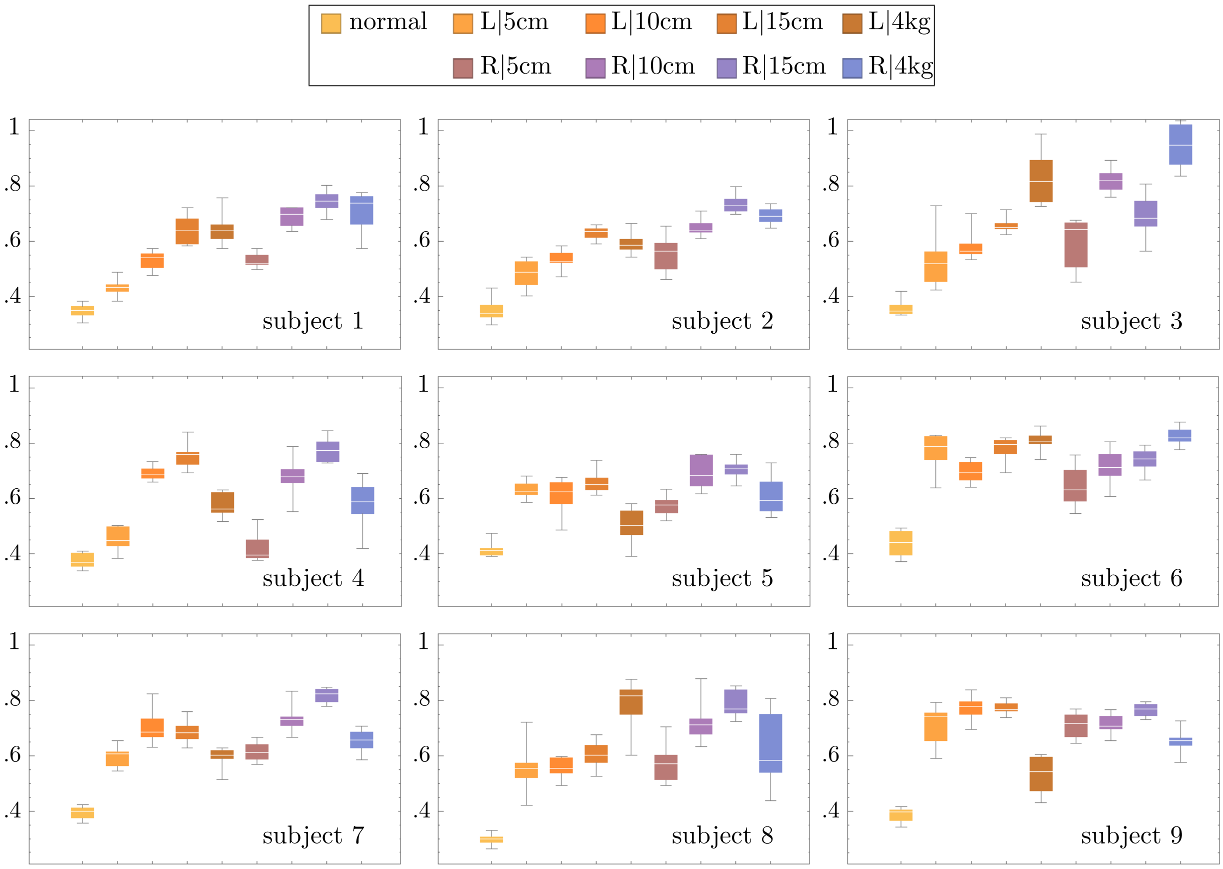


FIGURE 5.11: Statistic of the gait symmetry measurement in our experiments. The horizontal and vertical axes represent respectively gait types and corresponding measurements shown as box and whisker charts. The notation L|5cm indicates the simulated gait in which a sole with 5cm of thickness was padded under the left foot, while L|4kg means that a 4kg-heavy object was mounted to the left leg, and so on.

TABLE 5.1: Errors in distinguishing between normal (symmetric) and abnormal (asymmetric) gaits with different approaches

Test subjects	v_2, v_4, v_7, v_8		all subjects		leave-one-out	
Evaluation	short-term	full seq.	short-term	full seq.	short-term	full seq.
HMM [102]	0.335	0.250	-	-	0.396 (±0.117)	0.198 (±0.250)
One-class SVM [16]	0.227	0.139	-	-	0.274 (±0.183)	0.136 (±0.070)
Binary SVM [16]	0.157	0.139	-	-	0.152 (±0.058)	0.111 (±0.000)
MGCM [12]	-	0.250	-	0.222	-	0.125 (±0.125)
Our method	0.042	0.000	0.051	0.037	0.025 (±0.038)	0.000 (±0.000)

mentioned in Section 5.4.1. We also projected the 3D point clouds to provide depth maps as another data type. Sequences of such depth maps were used to evaluate the recent study [12] that proposes the longitudinal depth difference between left and right legs of averaged gait cycles as an indicator of gait asymmetry. Besides, method [102] was employed to deal with the skeletons. That study separated an input sequence of skeletons into consecutive gait cycles detected using the distance between two foot joints. A hidden

Table 5.2: The ability of our method indicated by ROC-based quantities estimated based on different sizes of cylindrical histogram (evaluated on all subjects)

Measure on	Quantity	Histogram size							
		Increasing of width				Increasing of height			
		16×8	16×16	16×24	16×32	8×16	16×16	24×16	32×16
Segments	AUC	0.989	**0.989**	0.988	0.987	0.989	0.989	0.989	**0.989**
	EER	**0.043**	0.050	0.044	0.044	**0.046**	0.050	0.050	0.050
Mean	AUC	**0.998**	0.997	0.995	0.995	0.997	0.997	0.997	0.997
	EER	**0.014**	0.028	0.028	0.028	**0.014**	0.028	0.028	0.028

Markov model (HMM) with a specific structure was employed to build a model of normal walking gait cycles as well as to provide a likelihood for each input cycle. The categorization was finally performed by comparing such log-likelihoods with a predefined threshold. For the silhouette input, we used the approach [16], in which the feature extraction was performed on each frame, the temporal context was embedded by vector concatenation, and a support vector machine (SVM) was employed for the task of classification. Both latter methods aim to classify each input sequence into two categories: normal and abnormal gaits. Their ability was evaluated based on different measures: the Area Under Curve of a Receiver Operating Characteristic (ROC) curve for [102] and typical classification accuracy for [16]. We decided to use the Equal Error Rate (EER) as the measure for comparison because this is estimated according to the ROC curve and its meaning is related to the classification accuracy. Such ROC-based measures have been employed in many problems of binary classification.

The HMM in [102] was built with only normal gaits. Therefore, beside the typical binary SVM, we also modified the model in approach [16] to have a one-class SVM. That unsupervised learning is reasonable in practical situations because there are numerous walking gaits that have abnormality, collecting a dataset of such gaits with a high generality is thus difficult. In our experiments, the HMM and one-class SVM were trained with the same dataset consisting of normal gaits of 5 (over 9) subjects (v_1, v_3, v_5, v_6, v_9 in Fig. 5.10 as suggested in [98]), and the (normal and abnormal) gaits of the remaining subjects were the test set. The binary SVM was also trained on all gaits of those 5 volunteers, and the test set included all gaits of the other 4 volunteers. In order to have a more general evaluation, we also performed the experiments using leave-one-out, i.e. 9-fold cross-validation where each fold contains all 9 gaits of a subject. The assessment was thus represented as mean (± std) of the evaluation quantity. The experimental results are presented in Table 5.1. The notation *short-term* has different meanings: a segment of 120 point clouds in our method, an automatically detected gait cycle in [102], and a temporal context of $\Delta = 20$ in [16] (i.e. per-frame classification based on vector concatenation of features in 21 recent frames). The notation *full sequence* indicates the classification based on mean values in our work (as shown in Fig. 5.10), lowest averages of log-likelihoods computed on three consecutive cycles in each sequence in [102], and alarm triggers on whole input sequences in [16].

According to Table 5.1, the classification errors resulting from our method are much lower compared with the others. Table 5.1 also shows that in all the 3 methods, the decision provided based on the whole input sequence had a higher confidence compared with short segments. In other words, the mean values in Fig. 5.10 were better than individual segment measures in indicating the gait symmetry embedded inside a sequence of point clouds. During our experiments, we observed that the binary SVM [16] always classified sequences of normal gaits (according to alarm triggers) into the category of anomaly. This property was clearly showed in the leave-one-out cross validation where the error was 0.111 for all 9 folds. This problem might be due to the large ratio between abnormal and normal gaits (8:1), and a binary (i.e. supervised) SVM was thus not really appropriate for the task of detecting abnormal gaits where there are numerous types of abnormal walking. Another drawback of using SVM in gait-related problems is the high computational cost. Since an SVM attempts to linearly classify input patterns in a high-dimension space, the increasing number of support vectors (especially when concatenating features over a sequence of frames) requires a large amount of computations. Employing the HMM as in study [102] may also have another limitation. Since an HMM can be considered as a chain of posture's states, a bad-determined skeleton may cause a disturbance to the state transformation and the outputted likelihood could thus be significantly affected. It was also noticeable that the approach [102] could be improved to get better results by modifying the width of sliding window since the frame rate of our data acquisition was lower than the system in [102]. Finally, the high error obtained from the evaluation of [12] showed the risk of estimating asymmetry index according to step cycles since a bad cycle separation may significantly impair the averaged gait cycle. Furthermore, the method works over a limited region of the lower limbs and consequently could lose relevant information available elsewhere on the body.

5.4.5 Sensitivity to size of cylindrical histogram

The cylindrical histogram plays the main role in our approach and also affects the gait symmetry assessment. By changing the histogram's size, i.e. number of sectors, the range of mean values in Section 5.4.3 would be different. The ability of distinguishing two gait types would also change. We can guess that a histogram with small resolution can reduce the computational cost of the entire system but may not have enough details for describing body postures. On the contrary, using a histogram formed from a large number of cylindrical sectors may also reduce the system's efficiency. In that case, each sector covers a small volume with low numbers of 3D points, the result of eq. (5.2) is thus sensitive to noise in the input 3D point clouds. In summary, the system accuracy can be improved by a careful selection of histogram size. Table 5.2 shows the abilities (according to AUCs and EERs of ROC curves) of our system for various histogram resolutions in distinguishing symmetrical and asymmetrical walking gaits. In this table, we focus on the mean-based measurement because it describes the gait symmetry better than segments (according to Table 5.1). The ability of our method tended to reduce, i.e. increasing of EER and decreasing of AUC, when we set a high value for the histogram width. The height of cylindrical histograms had a lower effect since the AUC and EER (for both segments and means) were almost unchanged when the height exceeded a particular threshold.

5.4.6 Discussion

The completeness and accuracy of motion capture or high-end inertial systems are unquestionable. The proposed system does not have the ambition to be as accurate as these gold-standard systems capable of specific measurements such as joint kinematics. However, motion capture or inertial systems rely on data provided by sensors or markers that are placed on the body and require expertise for marker/sensor placement, calibration and manual editing of the data, which could involve recruiting trained staff and requires time for measurement preparation and analysis. Inversely, our system has the advantage of being low cost, requiring a small space and is easy to use, without markers or sensors on the patient's body, without run-time calibration and without manual editing. Therefore, it can be deployed more easily in small clinics which could be a significant advantage. Our system may therefore complement more precise instruments (motion capture or inertial systems). For instance, our system could enable clinicians to perform more frequent screening or follow-up of patient before more elaborate analysis with gold standard systems if needed.

Let us notice that the measurement of the x-axis in Section 5.3.2.1 must be carefully performed since it directly affects the cylindrical histogram estimation and the left-right separation. A bad determination of the coordinate system may lead to a significantly impaired cylindrical histogram, and cross-correlation on the left- and right-histogram sequences could thus not be as accurate for measuring the body asymmetry. It is also important to remember that normal gait is different for every individual and therefore within-subject analysis should be considered to increase the performance of the method.

A noticeable feature of our approach is that local body parts (e.g. hips, arms,...) and their motion are not directly considered since we focused on the patient's global walking. In order to increase the application range of this method (e.g. measurement in neurological and/or musculoskeletal disorders), a cloud-based analysis on human body part locomotion and/or joint kinematics could be performed in future work.

Finally, let us notice that there is another dimension for increasing the resolution of our proposed cylindrical histogram: the radial dimension. By additionally segmenting the cylinder according to radial sectors (see Fig. 5.12), the obtained histogram becomes a 3D volume. We performed experiments on such 3D cylindrical histograms in order to evaluate the usefulness of such dimension. The AUCs estimated according to segment (of 120 frames) and sequence (average index of 1200 frames) are shown in Fig. 5.13. The use of only 1 radial sector corresponds to our described 2D cylindrical histogram. In Fig. 5.13(a) where the evaluation was performed on the entire 9 subjects, increasing the number of radial sectors tended to enlarge the deviation of symmetry indices, the ambiguity region between value ranges of normal and abnormal gaits was thus expanded and consequently the averaged AUCs decreased. This effect was demonstrated again with segment-based indices in Fig. 5.13(b). However, when using less than 5 radial sectors, within-subject analysis still provided good results since the normal gait and abnormal ones were perfectly distinguished (i.e. AUC = 1) for each subject. In summary, considering cylinder radius may be an extension for our method but requires further investigation with a larger dataset including more variability of abnormal and asymmetric gaits.

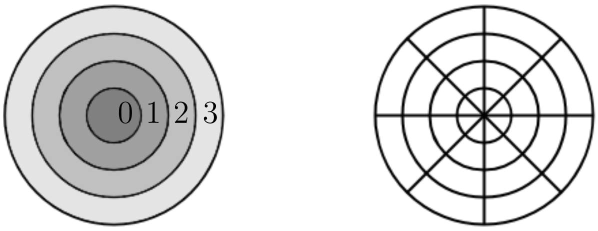


FIGURE 5.12: Illustration of splitting a cylinder by a radial grid. **Left:** A separation with 4 radial sectors that are indexed from 0 to 3. **Right:** Top-view of a cylindrical histogram of size $h \times 8 \times 4$ (h is not shown in the figure).

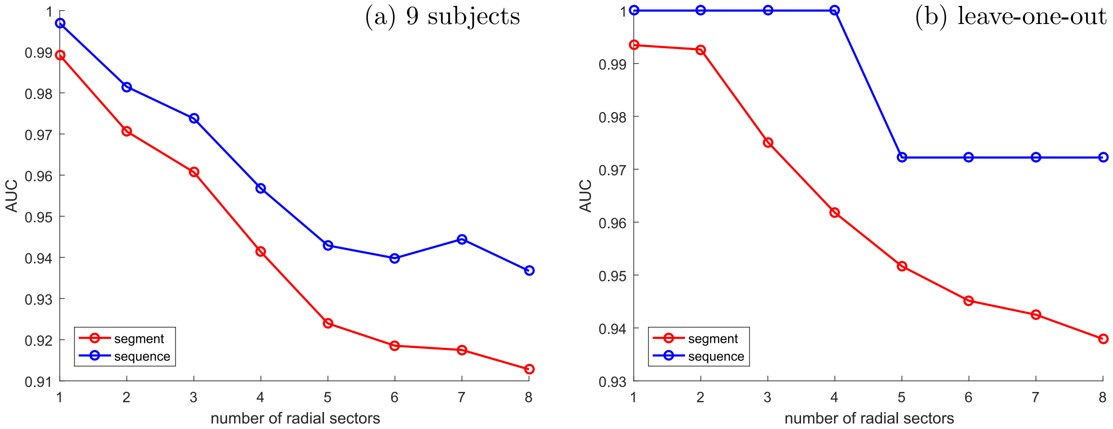


FIGURE 5.13: AUCs evaluated on symmetry index measured from 3D cylindrical histograms. The resolution of each histogram was $16 \times 16 \times r$ where r is the number of radial sectors ($1 \leq r \leq 8$).

5.5 Conclusion

In this chapter, we have presented an original and efficient low-cost system for assessing gait symmetry using a ToF depth camera together with two mirrors. The input of the proposed method is a sequence of 3D point clouds representing the subject's postures when walking on a treadmill. By fitting a cylinder on each point cloud, a cylindrical histogram is formed to describe the corresponding gait in the manner of self-symmetry. Cross-correlation is then applied on each pair of sequences of half-body sub-histograms to measure the gait symmetry along the movement. The ability of our method has been demonstrated via a dataset of 9 subjects and 9 gait types. Our approach also outperforms other vision-based methods that employ skeletons, frontal view silhouettes or depth maps as the input, in the task of distinguishing normal (symmetric) and abnormal (asymmetric) walking gaits. The resulting system is thus a promising tool for a wide range of clinical applications by providing relevant gait symmetry information. Patient screening, follow-up after surgery, treatment or assessing recovery after a stroke are obvious applications that come to mind. As future work, the proposed method will be modified focusing on particular pathological gaits such as diplegic, hemiplegic, choreiform, and Parkinsonian [141] in order to support the gait diagnosis on patients.

Chapter 6

Estimation of Gait Normality Index through Deep Auto-Encoder

This chapter presents a model-based approach for gait normality assessment given a sequence of 3D point clouds of human walking gaits. Compared with the work in the previous chapter, this model is promising for further objectives beyond assessing gait normality such as exploring common characteristics of typical walking gaits or checking the effect of specific body regions. This work has been published as the following journal article:

Nguyen, T.-N. and Meunier, J. Estimation of gait normality index based on point clouds through deep auto-encoder. *EURASIP Journal on Image and Video Processing*, 2019(1): 60, May 2019. ISSN 1687-5281. doi: 10.1186/s13640-019-0466-z. URL https://doi.org/10.1186/s13640-019-0466-z

6.1 Abstract

This chapter proposes a method estimating an index that indicates human gait normality based on a sequence of 3D point clouds representing the walking motion of a subject. A cylinder-based histogram is extracted from each cloud to reduce the number of data dimensions as well as highlight gait-related characteristics. We propose a deep auto-encoder that learns common features of gait normality based on histograms of point clouds and then provide a discussion on cloud-oriented deep networks for gait analysis. The ability of our approach is demonstrated using a dataset of 9 different gait types performed by 9 subjects and two other datasets converted from mocap data. The experimental results are also compared with other related methods that process different input data types

including silhouette, depth map, and skeleton as well as state-of-the-art deep learning approaches working on point cloud.

6.2 Introduction

Gait normality index estimation is one of the most common studied problems to support healthcare systems. Many researchers employed complex marker-based and multi-camera systems to acquire more details for gait analysis. One of their drawbacks is that they require specific devices with high price and/or have high computational cost. Therefore, some recent studies employed a single camera to deal with gait analysis problems. Depending on the used sensors, the input of those methods is either subject's silhouette or depth map. The former information has been used to propose numerous gait signatures such as Motion History Image (MHI) [32], Gait Energy Image (GEI) [57], and Active Energy Image (AEI) [87]. Each signature is a compression of a sequence of consecutive 2D silhouettes and is represented as a single grayscale or binary image. They were usually applied for the task of person identification. However, in the case of gait normality index estimation, using only the gait signature is not enough. Nguyen *et al.* [101] employed MHI to estimate 4-dimensional features. They processed each individual silhouette as well as segmented each input sequence of frames into gait cycles where the temporal context was embedded in. The gait assessment was performed on each gait cycle using a one-class model that was trained with normal gait patterns, i.e. unsupervised learning. Bauckhage *et al.* [15] also proposed an approach detecting unusual movement. They put a camera to capture the frontal view of a walking subject. Each silhouette was encoded by a flexible lattice that followed a vector conversion of coordinates corresponding to a set of predefined control points. The temporal characteristic was then integrated into each feature vector by concatenating vectors of consecutive frames. Differently from [101], the gait normality decision was determined based on a binary SVM where both normal and abnormal gait samples appeared in the training set. However, in many applications, using only a sequence of silhouettes as the input would lose important gait information because of the missing depth.

In order to deal with that limitation, depth sensors replaced color cameras in some studies. A popular device is the Kinect, which is provided by Microsoft with a low price and a

SDK containing the functionality of per-frame 3D human skeleton localization [132, 133]. Such skeletons played the main role in some recent studies of gait-related problems such as pathological gait analysis [20], gait recognition [72], and abnormal gait detection [102]. These approaches, however, still have a drawback since each skeleton is determined based on a depth frame. Concretely, self-occlusions in depth maps might lead to unusual skeleton postures, embedded gait characteristics would thus be deformed.

In this chapter, we present an approach dealing with the problem of gait normality estimation. We focus on a setup of cheap equipments to capture the motion from different view points. We employ a Time-of-Flight (ToF) depth camera together with two mirrors so that the system can work in the manner of a collection of cameras while keeping the cost much lower than multi-camera systems [107]. A subject performs her/his walking gait on a treadmill at the center of the setup. A depth map captured by our setup is presented in Fig. 6.1. As shown in the figure, there are 3 regions (highlighted with ellipses) corresponding to partial subject's surfaces seen from different view points. A point cloud representing the subject can thus be easily formed as a combination of 3 collections of reprojected points (from 2D to 3D) including (a) the real cloud in the middle and (b) reflections (through mirror planes) of virtual clouds that are behind the two mirrors. An example of such reconstructed 3D point cloud is presented in Fig. 6.2. More details on this reconstruction method are given in [107]. The input of our method is a sequence of these 3D point clouds that are formed based on consecutive depth frames captured by the depth camera. The output is gait normality indices provided by a model of normal walking postures. To our knowledge, this is the first work that performs gait normality index estimation on a sequence of 3D point clouds representing a walking person.

Our contributions are summarized as follows:

- Proposing a deep auto-encoder that learns common features of gait normality based on histograms of point clouds and a discussion on cloud-oriented deep networks for gait analysis.

- Demonstrating the potential of point cloud in gait analysis problems compared to typical input data types such as skeleton, depth map and silhouette.

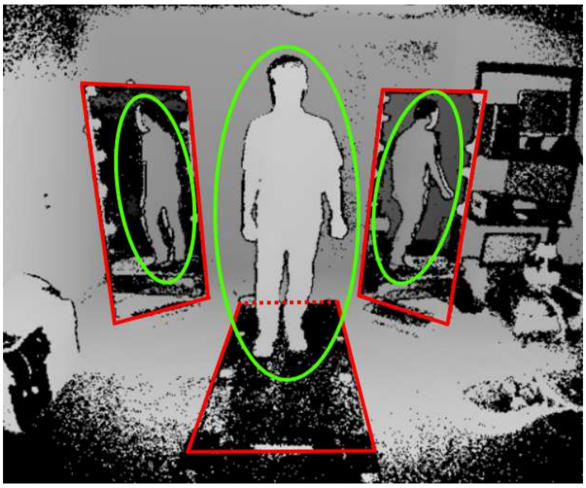


FIGURE 6.1: A depth map captured by our setup that shows 3 devices including two mirrors and a treadmill where each subject performs her/his walking gait. Three collections of subject's pixels are highlighted by ellipses.

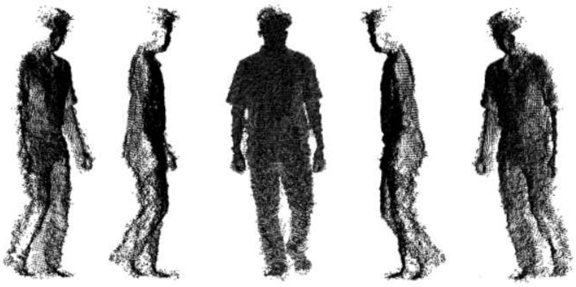


FIGURE 6.2: The point cloud reconstructed from a depth map using the method [107].

6.3 Proposed method

Our method consists of three main steps. First, a 2D histogram of each point cloud is formed to normalize the data dimension as well as highlight gait-related characteristics. Then, the second stage generates a model representing postures corresponding to normal walking gait based on a collection of 2D histograms. Finally, this model is used to compute a normality index for gait analysis.

6.3.1 Cylindrical histogram estimation

There are some inconveniences when performing gait assessment on 3D point clouds: (1) the number of points inside each cloud is not normalized, (2) such cloud may contain redundant information that are not useful for gait-related tasks, and (3) there may be some noises in each cloud, i.e. points reconstructed from depth values containing noise

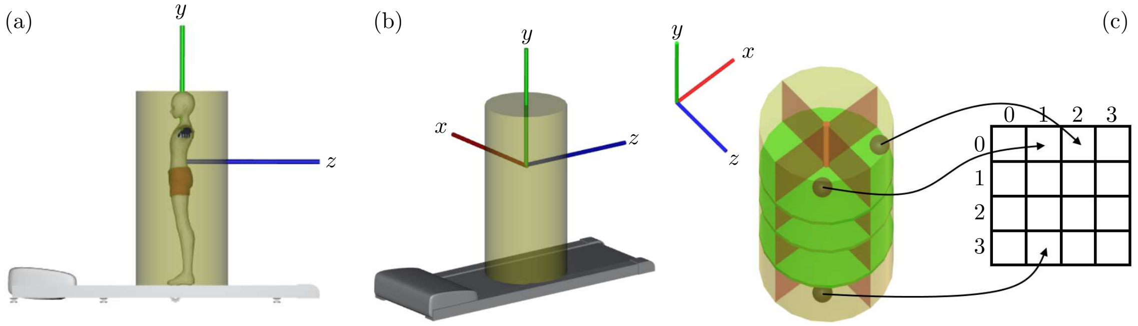


FIGURE 6.3: Visualizations of (a, b) fitting a cylinder onto a 3D point cloud and (c) the conversion from 16 cylinder's sectors to a 2D histogram with size of 4 × 4. The coordinate system in the three sub-figures is to present the mapping between each cylindrical sector and the corresponding elemental index in the histogram.

in the depth map. Therefore, each 3D point cloud is converted into a 2D histogram by fitting a cylinder with equal sectors. It is worth noting that this step of normalization also plays an important role when working with neural networks since such models require inputs of fixed dimensions. Its axis coincides with the normal vector of the treadmill surface and goes through the cloud's centroid. Illustrations of the cylinder fitting and histogram formation are shown in Fig. 6.3.

Let us notice that the coordinate system in that figure is flexible. The only constraint is that the y-axis must be normal to the treadmill surface. The coordinate system in Fig. 6.3 is to show the relation between cylindrical sectors and their mapped elements in the corresponding 2D histogram. Such arrangement of elements inside a histogram is to highlight the balance of human posture embedded in the point cloud. In other words, our cylindrical histogram is considered as a smart projection of a 3D point cloud onto a frontal (or back) grid. The element values of each histogram are finally scaled to give a grayscale image of 256 levels. This representation is convenient for data range normalization and for storing. An example of grayscale histogram and the corresponding human posture is given in Fig. 6.4.

6.3.2 Model of normal gait postures

Many recent studies embedded the temporal context into features that were then employed to create a model supporting gait classification. Our model, however, considers only individual postures. The temporal factor can then be integrated by extracting statistical quantities based on a sequence of posture assessments. An unsupervised learning

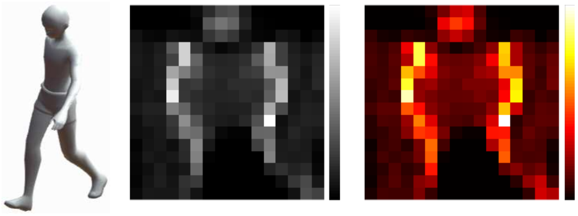
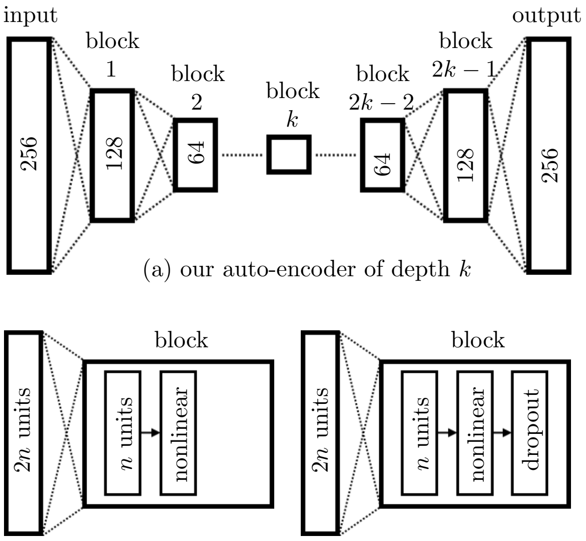

(a) (b) (c)

Figure 6.4: Example of 2D histogram estimated by fitting a cylinder onto a 3D point cloud: (a) posture, (b) grayscale histogram, and (c) pseudo-color histogram for better visualization. The size of this histogram is 16 × 16.

is appropriate since we are focusing on estimating gait normality index. A model that is formed from a training set containing both normal and abnormal gaits may have a low generalization. The reason is that patterns of abnormal gaits would significantly affect the classifier because there are too numerous possible types of walking postures with abnormality in practical situations. Therefore, we attempt to create a model describing common characteristics of normal gait postures. A typical way of performing this task is learning a vocabulary of code words extracted from histograms of normal gait. Recently, such approaches have demonstrated good performance on common problems such as content-based image retrieval [4, 162] and image classification [5, 160, 161]. Another approach is the use of pretrained deep networks for feature extraction such as [124, 135]. These methods, however, are applied on natural images with an appropriate resolution, in which each code word is formed from an image patch. Therefore, vocabulary learning is not suitable to deal with our histograms of small size 16 × 16. Since deep learning has provided very good results in recent studies, we decide to employ such structures that can automatically determine useful features itself and work as a one-class classifier. The deep auto-encoder [123] is thus chosen in our approach to model normal gait postures.

Our model structure is similar to a typical neural network but has some specific constrains. First, the model is a stack of blocks with the same layers inside. The only difference between these blocks is the number of input and output connections. Each block contains a fully connected layer, a non-linear activation layer, and an optional dropout layer. The dropout layer is considered to reduce the risk of overfitting [136]. We selected 3 popular activation functions including sigmoid, tanh, and leaky ReLU (rectified linear unit) for the middle (or last if no dropout) layer in each block. The original ReLU function is not considered because it may cause the problem of dead neuron [88] when embedded into a deep fully connected neural network where the learning rate is not small enough.



(a) our auto-encoder of depth k

(b) blocks used in our model

FIGURE 6.5: Structure of our auto-encoder that models characteristics of normal gait postures: (a) an example of model of block-level depth k with the number of units indicated inside each block, (b) two possible block structures used in our auto-encoder.

Let us consider a block l where its fully connected layer is parametrized by weights $W^{(l)}$ and biases $b^{(l)}$, the output of an i^{th} unit given an input $x^{(l)}$ is computed as

$$\begin{cases} y_i^{(l)} = W_i^{(l)} x^{(l)} + b_i^{(l)} \\ z_i^{(l)} = f(y_i^{(l)}) \\ \hat{z}_i^{(l)} = \mathcal{U}_i(p, N(x^{(l)})) * z_i^{(l)} \end{cases} \tag{6.1}$$

where f indicates one of the three mentioned activations, $N(x^{(l)})$ is the number of units connected from the previous block, and $\mathcal{U}(p, n)$ is a function that produces n binary values where p is the probability of zero ones. The block output $\hat{z}^{(l)}$ is the input of the next block, i.e. $x^{(l+1)} \leftarrow \hat{z}^{(l)}$.

The second constrain is that when the data is propagated from one block to the next, the number of dimensions is reduced by half. This property is reasonable since auto-encoders are to compress and highlight useful features inside the input. These two constrains are illustrated in Fig. 6.5. Since we consider one of three activation functions including sigmoid, tanh, and leaky ReLU, there are thus 6 different structures that can be employed for constructing our model. Notice that in the partial network of decoder, the number of units in a next block is doubled but the order of layers inside each block is the same. The auto-encoder structure in our work is symmetric, i.e. we stack $k - 1$ blocks with increasing data dimension after using k blocks to encode an input histogram. We use the term *block-level depth* (or simply *depth*) to indicate such value of k, a model of depth k will thus have $2k - 1$ hidden blocks. The input of our network is a vector of 256 elements that is vectorized from each 16×16 histogram. The loss function used in our work is the Mean Squared Error (MSE) combined with a L2-regularization to prevent the model from overfitting:

$$\mathcal{L}(\mathcal{H}, \hat{\mathcal{H}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left\|\mathcal{H}_i - \hat{\mathcal{H}}_i\right\|_2^2}{256} + \lambda \sum_{l} \left\|W^{(l)}\right\|_2^2 \tag{6.2}$$

where $\mathcal{H}$ and $\hat{\mathcal{H}}$ respectively denote a batch of n input vectorized histograms of 256 elements and their reconstruction, $W^{(l)}$ indicates weights of the fully connected layer in block l and λ is the regularization rate that controls the effect of Ws on the total loss $\mathcal{L}$.

6.3.3 Normality index

Since the input and output of our auto-encoder are the same in the training stage, we expect that the model can learn common characteristics embedded in normal walking gait. We also expect that the loss of information in case of abnormal posture inputs will be significantly higher compared with normal gaits. The normality index is computed for each individual posture as the MSE loss between the input and output vectors of the same size, i.e.

$$\mathcal{I}(h) = \frac{1}{256} \left\|h - \mathcal{M}(h)\right\|_2^2 \tag{6.3}$$

where h is an input vectorized cylindrical histogram and $\mathcal{M}$ denotes the model estimating a reconstruction from h. The gait assessment can be performed with or without considering the temporal factor depending on specific problems. Recent studies working on time series data (e.g. action recognition or video retrieval) embedded this factor into their processing in various fashions such as by considering the variance among successive key frames [152], concatenating consecutive frames [145] or using specific neural network layers [146]. In our work, we directly measure a normality index given a sequence of n cylindrical histograms by simply averaging their frame-level indices:

$$\mathcal{I}(h_{1..n}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}(h_i) \tag{6.4}$$

This measure is appropriate for the task of gait normality index estimation because of the following reason. A sequence of walking postures can be considered as a hierarchy: it is a collection of walking cycles and each cycle is a group of poses. Unlike related tasks such as action classification or behavior understanding, walking movement tends to be periodic. Given an input sequence that is long enough to cover a number of gait cycles, the average of frame-level normality indices is expected to implicitly indicate the overall measure through the gait cycles.

The details of our model parameters and the ability of measuring gait normality index for distinguishing normal and abnormal walking gaits are shown in the next section.

6.4 Experiments

6.4.1 Dataset

Our approach was experimented on a dataset that includes normal walking gaits and 8 simulated abnormal gaits [98]. The abnormal gaits were created by embedding asymmetry into walking postures. Concretely, this task was performed by one of the following actions: (a) padding a sole with 3 possible heights (5/10/15 centimeters) under the left or right foot, or (b) attaching a 4 kilograms weight to the left or right ankle. There are thus 8 possible walking gaits with anomaly. The normal and abnormal gaits were performed by 9 volunteers using a Kinect 2. Each gait was represented by a sequence of 1200 consecutive point clouds. They were formed by applying the method proposed in [107] at a frame rate of 13 fps. The speed of the treadmill was set at 1.28 kph. Beside 3D point clouds, our data acquisition also captured corresponding skeletons and silhouettes using existing functionalities in the Kinect SDK. These two data types were employed for a comparison between our method and two other related studies. In summary, the dataset contains 1200 point clouds, 1200 silhouettes, and 1200 skeletons for each gait type of a subject. Our experimental procedures involving human subjects were approved by the Institutional Review Board (IRB). The experiments focus on assessing the efficiency of the proposed models and demonstrating the potential of point cloud in gait normality index estimation compared with typical inputs such as skeleton, silhouette and depth map.

The dataset was split into two sets according to the suggestion in [98]. The first one including gaits of 5 subjects was used in the training stage. The gaits of the 4 remaining subjects were tested to evaluate the ability of our trained models. The same split was also used in our experiments on related works in order to provide a comparison. Beside that data separation, the leave-one-out cross validation (on subject) was also considered to evaluate our method in a more general fashion.

6.4.2 Auto-encoder hyperparameters

This section presents our selection for typical hyperparameters and the strategy for finding a reasonable value for the block-level depth k of our auto-encoder.

6.4.2.1 Typical hyperparameters

First, we consider the algorithm that performs the weight update after each iteration. We employed the RMSProp [139] since the learning rate is adaptively changed instead of being a constant value. An initial learning rate of 0.0001 was thus reasonable. The momentum that controls convergence speed was set to 0.9 according to the suggestion in [139].

Such selection of learning rate leads to the choice of the constant that affects the negative slope of the element-wise nonlinear activation leaky ReLU, i.e. α in the equation $f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x)$. This parameter was set to 0.1 in our model because a too small value (such as 0.01) still sometimes causes the problem of dead neuron.

Another layer that also requires a predefined parameter is dropout. In our model, the probability of forcing input elements to zero was set to 0.3. Using a larger value may cause difficulties for the model in attempting to recover meaningful information during iterations in the training stage.

The λ coefficient controlling the L2-regularization was set to 0.25 after evaluating some randomized generating values. For the training process, we used a batch size of 512 and 800 epochs for each possible network without dropout layer. The number of epochs used for training the models with dropout was higher (1600 in our work) as suggested in [136]. The model weights were initialized according to the method proposed by [51].

Table 6.1: Empirically selected hyperparameters in our auto-encoders

training algorithm	RMSProp
loss function	MSE
initial learning rate	0.0001
λ (L2-regularization)	0.25
momentum	0.9
batch size	512
α (leaky ReLU)	0.1
number of epochs (without dropout)	800
number of epochs (with dropout)	1600
dropout probability	0.3
weight initialization	Xavier [51]

Many traditional auto-encoders initialized their weights based on greedy layer-wise pre-training [18, 60]. Our model, however, is considered as a typical deep neural network where the input is a hand-crafting feature, our selection of weight initialization is thus reasonable. The collection of such hyperparameters is summarized in Table 6.1.

6.4.2.2 Depth determination

An important factor that is not considered in the previous section is the block-level depth of network [i.e. k in Fig. 6.5(a)]. This is the last parameter which needs to be determined in order to form a specific network structure. We selected an appropriate value using a cross-validation strategy applied on the training data consisting of gaits of 5 subjects.

Concretely, the cross-validation was performed with 5 folds, in which each one corresponds to the gaits of a subject. For each value k, we tested 6 networks [3 nonlinear activations with/without dropout layer]. Since an auto-encoder is considered as a lossy compression, it is obvious that increasing the number of blocks will increase the loss, i.e. the distance between an input and its reconstructed image. Therefore, we need a more meaningful criterion for depth selection instead of simply performing a loss comparison. Let us recall that our auto-encoder would be trained with the goal of modeling normal walking gait, the ability of providing gait indices that can well distinguish normal and abnormal gaits is thus suitable for assessing the optimal value of k. For a problem of binary decision, the Area Under Curve (AUC) of a Receiver Operating Characteristic (ROC) curve is an appropriate measurement and was used here.

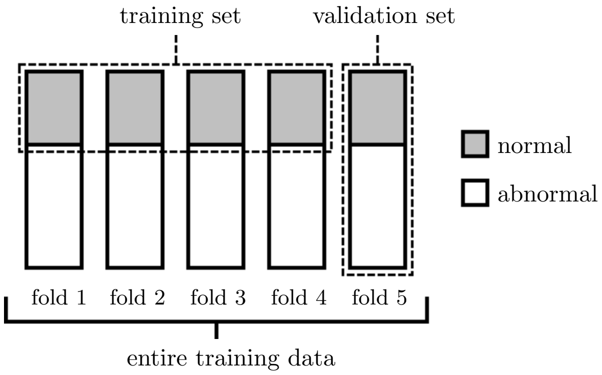


FIGURE 6.6: The formation of training and validation sets for one of 5 models corresponding to a specific network structure in the stage of cross-validation.

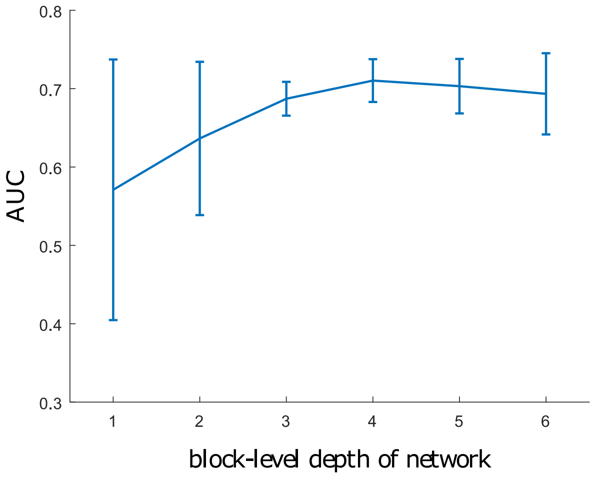


FIGURE 6.7: AUCs estimated in our cross-validation stage with different choices of network depth.

The stage of our 5-fold cross-validation was performed as follows. Given a block-level depth value k_0, we constructed 6 networks with $2k_0 - 1$ hidden blocks. Each network would provide 5 applicable models since the training data was separated into 5 folds. Each model was trained with the normal gaits of 4 folds (4800 histograms) to get a collection of 10800 MSE loss values when evaluating both normal and abnormal gaits (1200 and 9600 frames, respectively) of the remaining fold. A visualization of this separation is shown in Fig. 6.6. An AUC was finally estimated from such sequence of losses to represent the model's ability. Therefore, each of the 6 networks provided 5 AUCs in the stage of cross-validation given a specific depth. The mean AUC was calculated to represent the strength of each network for different depths in Fig. 6.7. Notice that we did not consider the choice of block structure, the cross-validation is just to find a reasonable depth for our auto-encoders.

According to Fig. 6.7, assigning 4 as the network block-level depth is a good choice since it provided the highest mean AUC and a relatively small standard deviation (that can be considered as a stability criterion). Our final network was thus trained with 7 hidden blocks (i.e. depth of 4) with hyperparameters in Table 6.1 using all normal gaits in the training data. The overall architecture of our model can be represented as a sequence of blocks F128AD-F64AD-F32AD-F16AD-F32AD-F64AD-F128AD-F256, in which FxAD indicates a block where F is a fully-connected layer that outputs x units, A is a nonlinear activation (sigmoid, tanh or leaky ReLU), and D is a dropout layer. When performing experiments on the models of non-dropout blocks, we simply set the dropout probability to 0.

There were 6 possible auto-encoders corresponding to 6 block structures. They were employed independently in our evaluations. Our networks were implemented in Python with the use of TensorFlow [1].

6.4.3 Reimplementation of related methods

In order to provide a comparison with other related works that employed different input data types, we also performed experiments on skeletons and silhouettes using the methods proposed in [102] and [16], respectively. The recent study [97] was also considered since it represents features of interest as an intermediate between 2D (silhouette) and 3D (depth map) information. Let us describe briefly these three approaches. The researchers in [102] directly employed the position of lower-limb joints in skeletons provided by a Kinect to extract feature vectors representing subject's walking postures. A sequence of such vectors was then converted into a sequence of codewords using a clustering technique in order to simplify the feature space. The sequence was segmented into gait cycles by considering the change of distance between two feet. This step is necessary since the researchers focused on building a model of normal walking gait cycles using a specific Hidden Markov Model (HMM) structure. The gait normality index was finally estimated for each input cycle as the log-likelihood provided by the trained HMM. Similarly to [102], the authors in [16] also performed the feature extraction on each silhouette using a lattice and embedded the temporal factor by concatenating vectors estimated from a number of consecutive frames. A difference of this method from [102] and ours is that the researchers employed a supervised learning (binary Support Vector Machine (SVM)) with two-class training dataset to distinguish normal and abnormal walking gaits. The method [97] estimated a gait-related score as a weighted sum of two scores corresponding to 2D and 3D information. Concretely, the researchers measured a LoPS (level of posture symmetry) score using a cross-correlation technique to describe the symmetry of 2D subject's silhouette, and simultaneously employed a HMM to compute a PoI (point of interest) score according to key points determined from the corresponding depth map. A combination of those two scores provided good results in distinguishing between normal and abnormal walking gaits. In our experiments, we reimplemented a HMM of normal walking gait cycle for the study [102], a binary SVM for [16], and a combination model of HMM and cross-correlation for [97]. We also slightly modified the SVM to create a one-class SVM where the training stage only dealt with samples of normal gaits. These models and ours were trained and evaluated on the same dataset split but with different input types, i.e. point cloud, skeleton, and silhouette. Notice that depth maps for experimenting the study [97] were formed based on a projection of 3D point clouds according to the calibration information.

6.4.4 Evaluation metric

The ability of each proposed network was measured according to an Equal Error Rate (EER) estimated based on the collection of MSE loss values. Since some related works attempted to embed the temporal context into their measurement, we also consider it by computing a simple average EER over a short segment (length of 120 in our experiments) of histograms as well as over the entire sequence (i.e. length of 1200) corresponding to each walking gait. Since we did not focus on selecting the best block structure in this work, the average loss of the 6 networks (with $k = 4$) was also computed. We also need to consider the measure for comparison since the three related works employed different quantities: the AUC for [102], the classification accuracy for [16], and the EER for [97]. We selected the EER estimated from the ROC curve to represent the evaluation result of all models because this measure is related to both AUC and classification accuracy.

6.5 Results

The experimental results on the suggested data split (5 training subjects and 4 test subjects) and the leave-one-subject-out cross-validation are respectively presented in Table 6.2 and 6.3. The last seven models are proposed in our work, in which the term *multi-network* indicates the assessment of gait normality indices estimated as the average of the losses resulting from the 6 other models. Notice that the notation *segment* has different meanings: a sub-sequence of 120 histograms in our approach, a gait cycle that was automatically determined in [102], a per-frame feature that embedded the temporal context of $\Delta = 20$ recent frames in [16] and $\Delta = 9$ recent frames in [97]. These values were suggested by the authors in their original works. The term *entire sequence* indicates EERs calculated based on the average loss over 1200 histograms in our method, lowest mean of log-likelihoods estimated on 3 consecutive walking cycles of a sequence in [102], alarm triggers in [16], and the average score in [97].

Table 6.2: Classification errors (≈ EERs) resulting from experiments on our auto-encoders and related studies with different data types.

Model	Training data	Data type	Classification error (4 test subjects)[†]		
			per-frame	segment	entire seq.
HMM [102]	normal only	skeleton	-	0.335	0.250
One-class SVM [16]	normal only	silhouette	0.399	0.227	0.139
Binary SVM [16]	normal + abnormal	silhouette	**0.104**	0.157	0.139
HMM [97]	normal only	depth map	-	0.396	0.281
cross-correlation [97]	normal only	silhouette	-	0.381	0.250
HMM + cross-correlation [97]	normal only	silhouette + depth map	-	0.377	0.218
(Our) Sigmoid	normal only	point cloud	0.332	0.264	0.250
(Our) Sigmoid + dropout	normal only	point cloud	0.328	0.261	0.250
(Our) Tanh	normal only	point cloud	0.298	0.158	0.111
(Our) Tanh + dropout	normal only	point cloud	0.289	0.136	0.111
(Our) Leaky ReLU	normal only	point cloud	0.326	0.125	**0.028**
(Our) Leaky ReLU + dropout	normal only	point cloud	0.296	**0.103**	**0.028**
(Our) Multi-network	normal only	point cloud	0.288	0.125	0.083

[†] Our system was originally implemented in Mathematica [151]. The models without dropout provided better results compared with the ones performed by TensorFlow [1] in this table. This may be because of the underlying algorithm implementation.

According to Table 6.2 and 6.3, employing the temporal factor improved the accuracy in estimating the gait normality index compared with *per-frame* (i.e. without considering recent frames) estimation except for the binary SVM which is a supervised learning. Therefore, we should focus only on the assessment performed on *segment* and *entire sequence*. The classification errors almost always significantly decreased when the gait normality index was estimated over the input sequence instead of short segments. Let

Table 6.3: Average classification errors (≈ EERs) resulting from our leave-one-subject-out cross validation.

Model	Training data	Data type	Classification error (leave-one-out)		
			per-frame	segment	entire seq.
HMM [102]	normal only	skeleton	-	0.396	0.198
One-class SVM [16]	normal only	silhouette	0.418	0.274	0.136
Binary SVM [16]	normal + abnormal	silhouette	**0.110**	0.152	0.111
HMM [97]	normal only	depth map	-	0.473	0.431
cross-correlation [97]	normal only	silhouette	-	0.321	0.097
HMM + cross-correlation [97]	normal only	silhouette + depth map	-	0.319	0.083
(Our) Sigmoid	normal only	point cloud	0.362	0.240	0.160
(Our) Sigmoid + dropout	normal only	point cloud	0.363	0.241	0.148
(Our) Tanh	normal only	point cloud	0.298	**0.144**	**0.049**
(Our) Tanh + dropout	normal only	point cloud	0.301	0.168	0.074
(Our) Leaky ReLU	normal only	point cloud	0.297	0.173	0.099
(Our) Leaky ReLU + dropout	normal only	point cloud	0.311	0.185	0.123
(Our) Multi-network	normal only	point cloud	0.303	0.178	0.086

us notice that our method measures the index of a sequence as a simple average of per-frame losses while the studies [16] and [102] used nonlinear computations, i.e. decisions respectively based on triggers and minimum 3-cycles means of log-likelihoods. In other words, those two methods assume that segment-based estimation possibly contains noises (or outliers), a post-processing is thus required to provide a decision. Our method directly calculates the index considering every measured loss. There were also several noticeable factors related to the approach [97]. First, the combination of silhouette and depth map in [97] has a lack of generalization compared with our method. Since our dataset (with 8 abnormal gaits) is an extended version of the one in [97] (without gaits with a 4 kilograms weight attached to the left or right ankle), Table 6.2 showed that the system [97] encountered difficulty in distinguishing those two additional abnormal gaits from normal ones. Another possible factor affecting the accuracy of method [97] is the size of training set (5 subjects in our experiments vs. 6 subjects in the original paper [97]). This was clearly demonstrated in Table 6.3, in which the method [97] provided good results when there were 8 training subjects in each fold. It also showed that the generalization ability of our deep neural network is better compared with the combination of HMM and cross-correlation given a small training set.

In order to demonstrate the effect of the length of input walking postures, i.e. n in Eq. (6.4), we provide the assessment on various values of the temporal factor in Fig. 6.8. These assessment results of default split and leave-one-out cross-validation schemes were respectively obtained from the models with leaky ReLU and tanh activations that provided best results in Tables 6.2 and 6.3. Figure 6.8 shows that the gait normality index

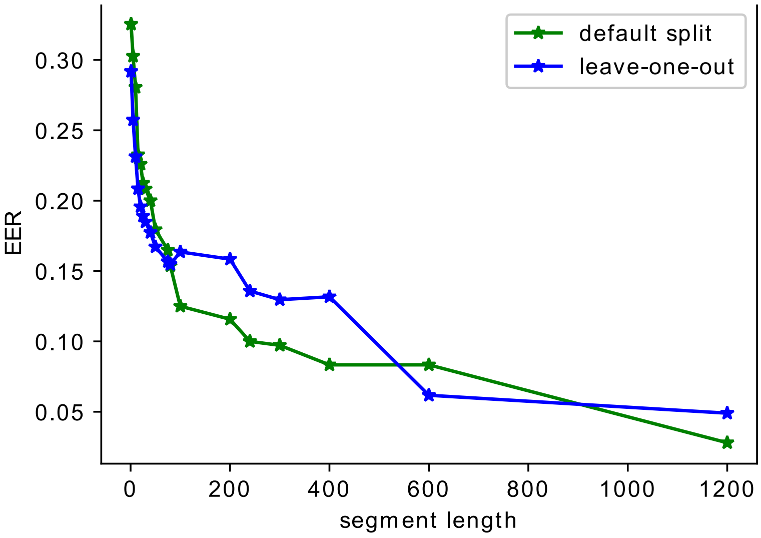


Figure 6.8: EERs obtained when the gait normality index was estimated on different lengths of posture sequence.

estimation tent to be improved with the increasing number of successive postures. Therefore, estimating gait index on a pre-assigned sufficiently large number of frames is an appropriate choice besides the typical consideration of walking gait cycle.

6.6 Comparison with deep learning models

With the fast development of deep learning, some networks have been proposed to deal with 3D point cloud for popular objectives such as classification, reconstruction and segmentation. We adaptively modified[1] three recent models including FoldingNet [157], PointNet [115] and RSNet [65] to obtain auto-encoder structures supporting the task of gait normality index estimation in the same fashion as ours. The former network is an auto-encoder while the two others are segmentation networks. Details of the reimplementation and experimentation are as follows.

First, each model requires its inputs having the same shape, i.e. a fixed number of points. Therefore, we employed random sampling [143] to downsample the number of points in each input cloud to 2048 for FoldingNet and PointNet, and 4096 for RSNet. Second, we adapted the last layer and the objective function of PointNet and RSNet to obtain new architectures of point cloud reconstruction. Concretely, the number of channels in their last layer (corresponding to the number of segmentation categories) was replaced by the

[1] The modification was performed on official public resources of these studies.

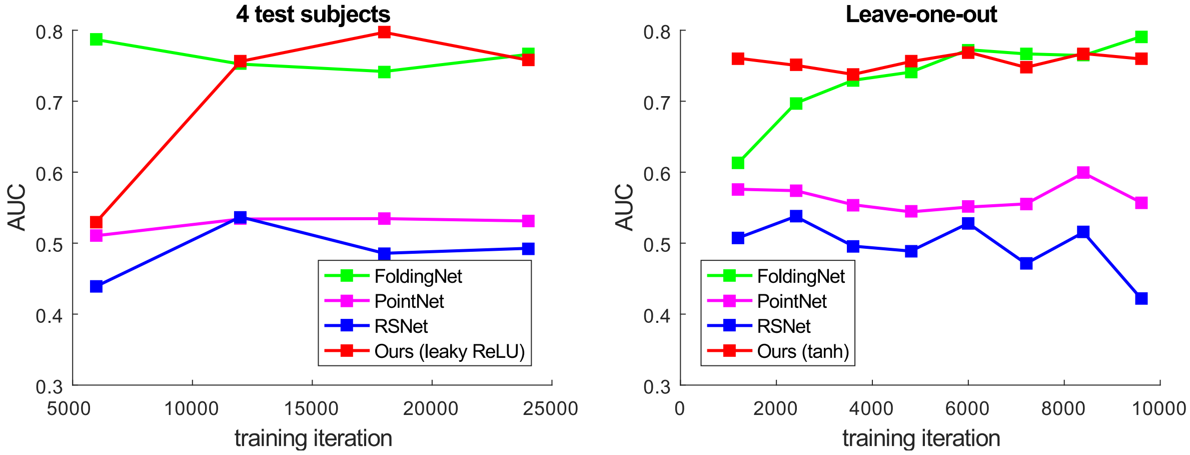


FIGURE 6.9: AUCs estimated from our evaluation on deep learning models.

number of input channels (i.e. 3 for the coordinates). The softmax loss was changed into MSE loss to force the models learning a way of reconstructing point position instead of performing point classification. The FoldingNet originally uses Chamfer distance for the reconstruction since its input and output clouds have different sizes, we thus did not perform any modification on this model structure. The loss of these models were used to indicate the gait normality index. In order to provide a comparison on processing time, we converted the framework of FoldingNet from Caffe [71] to TensorFlow [1].

Similarly to previous experiments, we evaluated the three networks using two schemes: the suggested data split and the leave-one-subject-out cross-validation. These models were respectively trained for 24000 and 9600 iterations with batch size of 1 for the two schemes. Notice that these numbers of iterations are just to evaluate the potential of models instead of guaranteeing a convergence. We also retrained our best networks (according to Table 6.2 and 6.3) in the same fashion for comparison. Since there was no classification model in this evaluation, we used AUC as the performance measure. The AUCs estimated on the gait indices outputted from all networks are shown in Fig. 6.9. Notice that we consider only per-frame index.

The experimental results show that our method and FoldingNet have a similar potential for estimating gait normality index. There are some possible reasons for the efficiency of FoldingNet. First, it considers local property of each point via the k-NN point-graph and local covariance of its neighborhood. This consideration would thus lead to a good feature extraction/description as typical convolutional neural networks. Second, the reconstructed cloud contains just a small number of outlier points since it is warped from

Table 6.4: Average processing time of basic operations in experimented models. The preprocessing indicates the cylindrical histogram formation in our method and the cloud downsampling in the others. The time is reported in second and millisecond.

Model	Framework	Preprocessing (using C++)	Forward & backward (in training stage)	Forward (in inference stage)
FoldingNet [157]	TensorFlow	**0.262** (ms)	1.639 (s)	0.446 (s)
PointNet [115]	TensorFlow	**0.262** (ms)	1.308 (s)	0.102 (s)
RSNet [65]	Torch	0.311 (ms)	0.202 (s)	0.058 (s)
Our 6 models	TensorFlow	1.126 (ms)	**0.014** (s)	**0.002** (s)

a 2D point grid. Therefore, the use of Chamfer distance in gait index calculation is not significantly affected by noise in the input cloud. Recall that there was no enhancement step performed on clouds in our experiments. On the contrary, PointNet and RSNet were directly designed for predicting point's label instead of explicitly emphasizing informative hidden attributes to support the cloud reconstruction. Besides, the point neighborhood is determined using a small network in PointNet and a pooling layer in RSNet while FoldingNet directly considers the distance-based point graph. We believe that this is a reason for the large efficiency gap between FoldingNet and the two others in the task of cloud reconstruction.

A summary of single-cloud processing time corresponding to basic steps in our experiments is given in Table 6.4. The evaluation was performed on a single GTX 1080 using Torch 0.4.1 (for RSNet) and TensorFlow 1.10.1 (for the others) with Python 3.5. It is obvious that FoldingNet takes very long times in both training and inference stages compared with our models. This is because we represent each input cloud by a 16×16 matrix and this size does not increase during propagation in the network. On the contrary, FoldingNet operates on cloud coordinates together with the distance-based graph, performs multiple concatenations, and uses the costly Chamfer distance as the loss function. It should also be noticed that RSNet may be slightly slower when using TensorFlow since the study [130] reported that Torch is faster than TensorFlow.

6.7 Experiments on additional datasets

In addition to the dataset used for experiments in previous sections, we also performed some testing on two smaller datasets formed from mocap data. In detail, some mocap walking sequences including normal and looking-like-abnormal gaits (unbalance, hobble,

Table 6.5: Number of frames and walking sequences in additional datasets. Each pair of values u (v) indicates a collection of v sequences containing a total of u frames.

Dataset	Training set	Test set	
	(only normal gait)	Normal	Abnormal
CMU	540 (5)	769 (8)	2224 (7)
SFU	1082 (5)	1295 (6)	3086 (13)

Table 6.6: EERs obtained from experiments on two additional datasets. The two methods [17, 118] are not adaptive to perform per-frame assessment.

Method	CMU		SFU	
	frame	sequence	frame	sequence
K-means [17]	-	0.133	-	0.474
Bayesian GMM [17]	-	0.133	-	0.231
One-class SVM [118]	-	0.400	-	0.356
Bayesian GMM [118]	-	0.267	-	0.350
Ours (leaky ReLU)	0.233	**0.067**	0.253	**0.158**

skipping, swaggering) were sampled from the CMU[2] and SFU[3] databases. These mocap data were converted to point clouds by fitting a 3D model (created with MakeHuman[4]) and using the set of 3D vertices as the point clouds. A summary of the two additional datasets used in this experiment is given in Table 6.5.

In order to provide a comparison, we also reimplemented two recent studies [17, 118] that perform gait analysis on human movement. The method [118] decomposes gait input signals into an ensemble of intrinsic mode functions to extract gait frequency properties and then analyzes their association and inherent relations. The study [17] also considers periodical factors, but the gait features were manually estimated from 3D skeletons including average step length, mean gait cycle duration and leg swing similarity. Both methods focus on efficient gait characteristics and employ simple learning algorithms for the assessment.

The experimental results (EER) are presented in Table 6.6. It shows that our gait normality index was improved over a walking sequence instead of on each frame. Notice that these two datasets were selectively collected from mocap databases focusing on action recognition. Table 6.6 also shows that the cylindrical histogram can be appropriate for describing various gaits.

[2] http://mocap.cs.cmu.edu/

[3] http://mocap.cs.sfu.ca/

[4] www.makehumancommunity.org

6.8 Discussion

First, let us explore in more detail the classification errors provided by the proposed auto-encoders. When embedding the temporal context into the estimation of gait normality index, the model which employed the leaky ReLU activation together with dropout layers provided the best results according to Table 6.2. In the leave-one-out cross-validation stage, replacing such combination by tanh activation gave the lowest classification errors. Therefore, more experiments as well as an extension of the dataset are needed to confirm the best block structure. However, the two tables show that using the tanh and/or leaky ReLU is preferred to sigmoid activation. In addition, the average of indices resulting from the 6 auto-encoders corresponding to 6 block structures (last row of Table 6.2 and 6.3) demonstrated the potential of auto-encoder compared with the three other related methods.

Second, it is worth noting that our cylindrical histogram provides a good visual understanding (see Fig. 6.3) while intermediate features extracted from a cloud-oriented deep neural network would be much more difficult to interpret. Therefore, our method is more appropriate for practical applications where users/operators are not familiar with the more difficult interpretation of intermediate features in deep networks.

Another important factor is the coordinate system that is illustrated in Fig. 6.3. A setting that does not satisfy this constraint might significantly affect the ability of extracted histograms in reasonably representing gait postures. In that case, a rigid transformation [59] is an appropriate solution to guarantee the constraint.

Finally, the local motion of body parts (e.g. limbs) is not explicitly considered in a sequence of cylindrical histograms. A further investigation of such local descriptions is expected to increase the applicability of the method to specific gait problems.

6.9 Conclusion

This chapter proposes an approach that estimates the gait normality index based on a sequence of point clouds formed by a ToF depth camera and two mirrors. Using such system not only reduces the price of devices but also avoids the requirement of a synchronization protocol since the data acquisition is performed by only one camera. This work introduces a simple hand-crafting feature, cylindrical histogram, extracted from raw input clouds that efficiently represents characteristics of walking postures. Auto-encoders with a specific block-level depth and various block structures are then employed to process such sequence of histograms, and the resulting losses are considered as gait normality indices. The efficiency of our method was demonstrated in the experiments using a dataset of 9 subjects with 9 different walking gaits. The quality of 3D point clouds provided by our setup was also highlighted in a comparison with other related works that employed different input data types (skeleton, silhouette, and depth map). Our method could be appropriate for many gait-related tasks such as assessing patient recovery after a lower-limb surgery for instance.

In further works, elaborate experiments will be performed to select the block that is best appropriate with our model structure. Besides, sparsity constraints will be considered to give visual understanding about characteristics embedded inside the cylindrical histograms that are useful for gait-related tasks. Finally, modeling specific pathological gaits using our auto-encoders is also an interesting future study.

Chapter 7

Gait Abnormality Index Estimation using Adversarial Auto-Encoder

This chapter presents an alternative model-based approach for gait normality assessment where the efficiency is comparable to the method in the previous chapter. Its main advantage is that the model was formed with a simple architecture instead of requiring a careful consideration as the previous work. However, the optimization may encounter difficulty for determining a convergence state. This work has been published as the following journal article:

Nguyen, T.-N. and Meunier, J. Applying adversarial auto-encoder for estimating human walking gait abnormality index. *Pattern Analysis and Applications*, Feb 2019. ISSN 1433-755X. doi: 10.1007/s10044-019-00790-7. URL https://doi.org/10.1007/s10044-019-00790-7

7.1 Abstract

This chapter proposes an approach that estimates a human walking gait abnormality index using an adversarial auto-encoder (AAE), i.e. a combination of auto-encoder and generative adversarial network (GAN). Since most GAN-based models have been employed as data generators, our work introduces another perspective of their application. This method directly works on a sequence of 3D point clouds representing the walking postures of a subject. By fitting a cylinder onto each point cloud and feeding cylindrical histograms to an appropriate AAE, our system is able to provide different measures that may be used as gait abnormality indices. The combinations of such quantities are also investigated to obtain improved indicators. The ability of our method is demonstrated

by experimenting on a large dataset of nearly 100 thousands point clouds and the results outperform related approaches that employ different input data types.

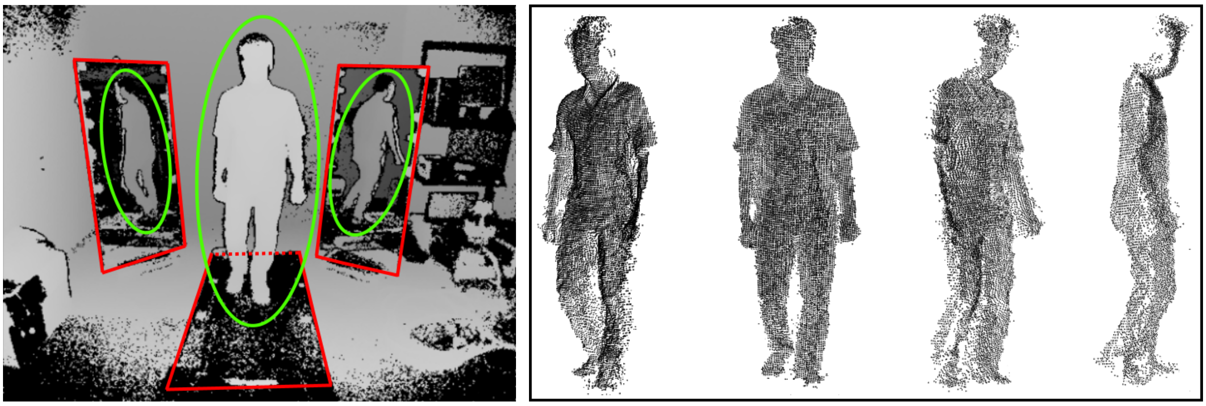
7.2 Introduction

Gait analysis has a wide variety of applications in medicine, person identification or activity recognition. In healthcare, many gait measurements can be done for the precise identification of locomotion problems and the planning of an appropriate treatment. However there are many situations where an overall measurement of the quality of gait would be useful to the clinician. In this work, we propose such gait index using a computer vision approach and adversarial auto-encoder to detect abnormal gait.

7.2.1 Common computer vision approaches for gait analysis

In order to deal with problems of gait analysis with computer vision methods, researchers employed different data types. Early studies started with a color camera that captures subject silhouettes under a specific view point. Many gait signatures have been introduced to describe various properties of each individual. For example, the Motion History Image (MHI) [32] used the pixel intensity to represent the motion history at the corresponding location. Another gait signature, Gait Energy Image (GEI) [57], focused on person identification by calculating an average image of consecutive aligned silhouettes. Beside such characteristics, researchers also proposed some problem-oriented features describing the movement. By proposing a 4-d vector that employed the MHI to indicate subject posture in each frame, Nguyen *et al.* [101] measured a walking gait index for each gait cycle as the log-likelihood provided by a hidden Markov model (HMM). Differently from that work, Bauckhage *et al.* [16] captured the walking silhouettes under the frontal view in order to detect abnormal gaits via the balance deficiency of motion. A common drawback of such silhouette-based gait analysis is the significant dependency on the camera view point and self-occlusion in captured silhouettes.

Another popular input of gait analysis systems is 3D skeleton. Since the Kinect 1 and 2 were released by Microsoft with low prices and SDK for skeleton localization [132, 133], these devices have been applied in many studies surpassing previous approaches using a



(a) Depth map captured by our system (b) Reconstructed point cloud

Figure 7.1: Data acquisition of our system: (a) a depth map showing our setup that includes a treadmill and two mirrors (highlighted by rectangles), each depth map captures three subject's surfaces (marked by ellipses) under different view points, (b) a reconstructed point cloud of a similar posture.

2D skeleton or other 2D model. Such skeletons have been demonstrated to be useful for a wide variety of applications such as recognizing predefined gaits [72], human-machine interaction [76, 119] and action recognition [154, 155, 156]. The skeletal input was also employed for healthcare related studies such as analyzing pathological gaits [20], and detecting abnormal gaits [102]. However, these skeletons that are detected based on depth maps may have a higher risk of posture deformation with pathological gait e.g. due to self-occluded parts.

To alleviate the previous problems, our method attempts to represent a subject pose by 3D information collected from different view points. The effect of view point dependency (including self-occlusion) would thus be reduced. Instead of employing a system of multiple cameras as in [10, 86], we use only one Time-of-Flight (ToF) depth camera together with two mirrors. Each mirror plays the role of a virtual depth camera where its position is symmetric with the real one through the corresponding mirror plane. A depth map captured by the ToF camera in our setup is presented in Fig. 7.1. Since the scene is captured by only one device, the task of camera synchronization is thus avoided. Furthermore, the system is not expensive and does not require precise placement of sensors or markers on the body of the patient (e.g. motion capture). Our system provides a 3D point cloud of a subject walking on a treadmill for each depth frame using the method proposed in [107, 108]. These point clouds are then fed to the AAE (next section) to obtain gait abnormality indices.

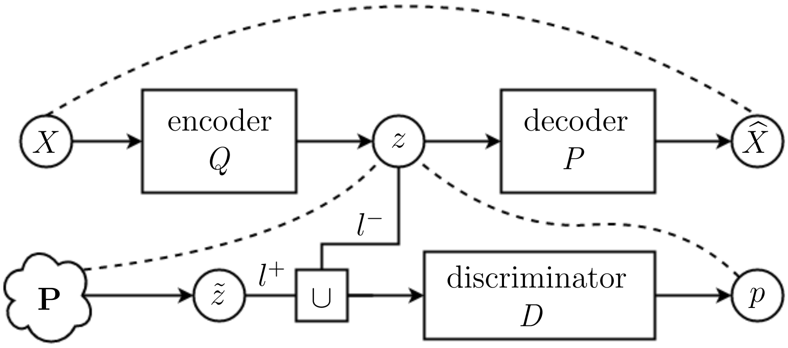
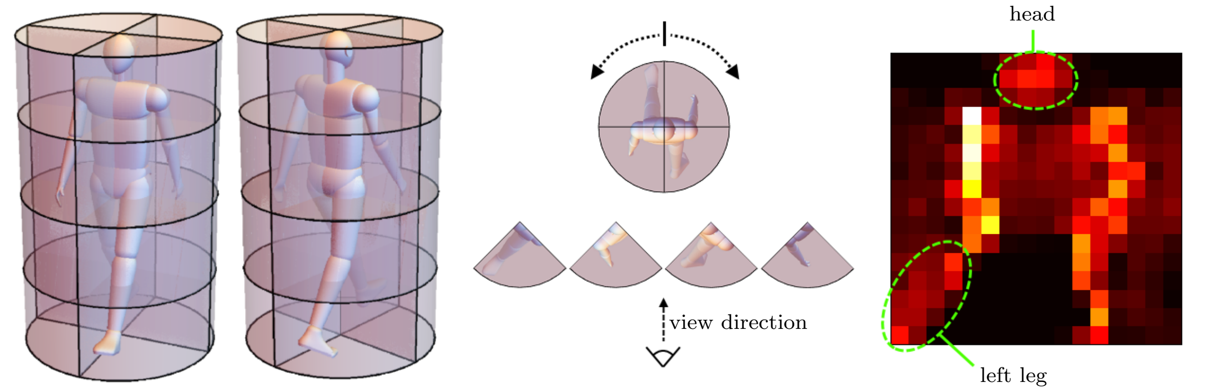


FIGURE 7.2: A typical AAE where X and $\widehat{X}$ are respectively an input and its reconstruction provided by the AE, z is the representation of X in latent space, $\mathbf{P}$ is a predefined prior distribution that draws samples $\tilde{z}$, l^+ and l^- respectively indicate the assigning of positive and negative labels, and p is the probability that an input is from $\mathbf{P}$, i.e. its label is positive (l^+). The operation $\cup$ represents the union of labeled samples z and $\tilde{z}$. In this diagram, the dash lines indicate components that may provide partial measures.

7.2.2 Adversarial auto-encoder

An AAE can be considered as a combination of an auto-encoder (AE) and a generative adversarial network (GAN) [53]. The AAE was introduced in [89] to perform variational inference so that the aggregated posterior distribution of latent variables is similar to a given prior distribution. That model focuses on supporting the task of sample generation that is currently a research trend. Our work, however, considers the AAE under another perspective. Inspired by recent works [97, 159] where a weighted combination of partial measures helped to improve the final assessment, we believe that an AAE could be applied in the same fashion since it contains multiple partial networks that can provide input-oriented measures. Our system does not focus on evaluating generated samples, the objective instead is to tune model weights so that such partial measures are reasonable to indicate a gait index for each input of point cloud. An overview of the AAE used in this work is presented in Fig. 7.2.

The remainder of this chapter is organized as follows: Section 7.3 describes the processing flow of our approach; the experiments on a large dataset, a comparison with related methods and an investigation of model input size are given in Section 7.4; Section 7.5 presents the conclusion together with possible extensions that may improve the current work.



(a) Fitting a cylinder of 16 sectors onto a body (b) Flattening a cylinder (c) A real histogram of 256 sectors

FIGURE 7.3: Cylindrical histogram: (a) a cylinder, that contains 16 equal-volume sectors, is employed to segment a 3D point cloud (a 3D model was used in the figure to provide a better visualization), (b) the collection of cylindrical sectors is then flattened to give a 2D representation, i.e. a histogram where each bin is the number of 3D points inside the corresponding sector, and (c) a pseudo-color version of such histogram (of size 16 × 16) that was estimated from our real data. Human model created by Dano Vinson (https://grabcad.com).

7.3 Proposed method

As presented in Fig. 7.2, the input X is fed to an AE where the number of units in the input layer is fixed, the point clouds should thus be converted into an appropriate representation. In other words, such point clouds need to be normalized to vectors or images (depending on the AE structure) with a predefined length or resolution.

Differently from studies [16, 102] where the temporal factor was directly integrated into the stage of feature extraction, we first perform the gait index measurement on each individual point cloud and then consider a sequence of such measures to assess the whole gait.

7.3.1 Posture representation

Each input of our AAE is a 3D point cloud that is reconstructed from the corresponding depth map using the method [107] [Fig. 7.1(b)]. Such clouds are simply stored as ensembles of 3D points with various numbers of elements. A neural network cannot easily adapt the number of units in its input layer, we thus need a procedure that transforms each point cloud into a new representation with a predefined shape. In order to perform this task, we use a cylinder with same-size 3D sectors to fit the point cloud (see Fig. 7.3). The main axis of the cylinder goes through the cloud centroid and is normal to the ground plane (or treadmill surface in our experiments). The top and bottom bases respectively go through the highest and lowest points (along the main axis) of the cloud. The cylinder's radius is large enough to guarantee that every point is inside the cylinder volume. The collection of such 3D sectors can be flattened to obtain a 2D histogram where each bin value indicates the number of 3D points belonging to the corresponding sector.

An illustration of our histogram formation is shown in Fig. 7.3. First, a cylinder is employed to fit the input 3D point cloud according to the mentioned constraints (i.e. main axis, top and bottom bases, and radius). The cylinder is then separated into same-size sectors using horizontal and vertical slices as shown in Fig. 7.3(a). It is obvious that the cylinder's main axis is normal to the horizontal slices and is the intersection of the vertical ones. In the next step, the number of 3D points inside each sector is counted, the input point cloud thus becomes a cylindrical histogram. In order to get an appropriate representation, the collection of sectors is flattened to a typical 2D array. The flattening also provides a visual understanding since body parts can be easily localized on the histogram (see Fig. 7.3(c) where the head and the left leg are indicated). Let us notice that in our work and experiments, this histogram is seen from the back as shown in Fig. 7.3(b). Such arrangement of sectors is not strictly a constraint because our model does not consider this factor. In the implementation stage, such cylindrical histogram can be formed by performing a loop on 3D points and determining the corresponding sectors based on geometric calculations.

After estimating the histogram, an enhancement is performed for the following reasons. First, the value assigned to each bin is the number of points belonging to the corresponding sector, measuring gait index directly on such data is thus significantly affected by the subject's shape properties. For example, the cloud that is formed with a fat subject should contain much more points than a thin one. Therefore, a normalization is necessary. Each histogram is thus scaled to the range [0, 1]. This operation is also useful for further processing where neural networks are employed. Beside the scaling, the output range is also separated into 256 levels. The histogram can thus be stored and directly visualized as a typical image. In our work, the selected size of cylindrical histogram is 16×16. Notice that this is just an arbitrary choice, not necessarily the optimal one. The histogram size can be considered as a hyperparameter of our model. The effect of this factor was considered in our experiments on various histogram sizes in Section 7.4.5.

Table 7.1: Structures of the 3 partial networks in our AAE.

encoder $Q(z\|X)$		**decoder $P(\widehat{X}\|z)$**		**discriminator $D(z)$**	
layer	no. of units	layer	no. of units	layer	no. of units
input	256	input	16	input	16
fc	96	fc	96	fc	96
lrelu	-	lrelu	-	lrelu	-
fc	16	fc	256	fc	1
		sigmoid	-	sigmoid	-

Abbreviation: fc = fully-connected, lrelu = leaky ReLU

7.3.2 Model components

In this work, the AAE is our choice for building the model because we focus on unsupervised learning. Since there are numerous possible walking gaits, collecting patterns of every type of gait for a supervised learning is nearly impossible. On the other hand, the unsupervised learning does not consider the data label and is appropriate for a training set that contains samples belonging to only one class. Our idea is to create a model that provides the score measuring the similarity between an input and known gaits. Another reason for the choice of unsupervised learning is that gait indices are usually used to assess the normality of a subject walking, a one-class classifier is thus appropriate. In our experiments, the AAE was trained using only normal walking gaits.

As visualized in Fig. 7.2, our model contains 3 main partial networks: the encoder and decoder that belong to the AE, and the discriminator that estimates the probability that an input is drawn from the given distribution $\mathbf{P}$. Each network is simply designed as a stack of fully-connected layers. Unlike popular deep learning models, we do not use any convolutional layer in our AAE because of the following reason. The input X is a normalized histogram instead of a natural image. Different inputs have a similar structure (e.g. body part position, body orientation), a convolutional layer (as well as a pooling layer) is thus not necessary to highlight common low-level features. In our work, each input sample X contains 256 elements (corresponding to a histogram of size 16×16), and the latent space (i.e. z) has 16 dimensions. The structures of the three partial networks are presented in Table 7.1.

The three components in our AAE use a similar hidden layer of (experimentally selected) 96 units that are fully connected from the input and are then activated by a leaky ReLU (rectified linear unit). The output layer of the decoder P attempts to reconstruct the input X of the AE. Therefore, 256 units are contained in that layer and followed by the sigmoid activation to guarantee each outputted element asymptotically belongs to the range [0, 1]. The sigmoid in the discriminator D focuses on another objective that is to estimate a probability.

The connection between our gait abnormality index and the AE is as follows. Our auto-encoder is considered as a lossy compression since the number of latent units is much less than the input dimension. Because of such bottleneck structure, the AE attempts to determine and propagate the most emphasized features of the training data. These characteristics are expected to appear only in the inputs sampled from the distribution of training samples. Therefore, the reconstruction loss can be employed to measure the difference between an unknown gait and the trained ones. Recall that our model was trained using only normal gaits in our experiments.

Our training stage employed three different optimizers. The first one uses the Adam algorithm [78] to train the encoder Q and decoder P together as a typical AE to minimize the reconstruction error. The loss function is cross entropy as follows:

$$L_{AE} = -X\text{log}(\widehat{X}) - (1-X)\text{log}(1-\widehat{X}) \tag{7.1}$$

where the input terms are similar to the notations in Fig. 7.2. The two remaining optimizers deal with two components of the adversarial loss that has the overall form:

$$\min_{Q}\max_{D} \mathbb{E}_{\tilde{z}\sim\mathbf{P}}[\text{log}D(\tilde{z})] + \mathbb{E}_{z\sim Q(z|X)}[\text{log}(1-D(Q(z|X)))] \tag{7.2}$$

where $\mathbf{P}$ is the given prior distribution and the encoder $Q(z|X)$ plays the role of the generator in the GAN. The optimization of such minimax function can be performed by alternatively optimizing the two following losses:

$$L_D = \frac{1}{2n}\sum_{i=1}^{n}[-\text{log}D(\tilde{z}_i) - \text{log}(1-D(Q(z_i|X_i)))] + \frac{\gamma}{2}R_D(\tilde{z}, z, D) \tag{7.3}$$

$$L_Q = \frac{1}{n}\sum_{i=1}^{n}[-\text{log}D(Q(z_i|X_i))] \tag{7.4}$$

where n is the number of samples $\tilde{z}$ with positive label drawn from $\mathbf{P}$ as well as the number of normal gait postures X drawn from the training set. γ is an annealing factor that is combined with the regularization R_D in order to increase the stability when training the discriminator [120]. In detail, one reason of the difficulty in training a GAN model is the mismatch between the distributions $\mathbf{P}$ and $Q(z|X)$. The study [120] attempted to overcome this problem by adding noise to the sampled data. Mathematically, both $\mathbf{P}$ and $Q(z|X)$ were convolved with white Gaussian noise. This operation was integrated into the GAN as a regularization R_D of the objective function of discriminator D. By performing analytic approximation and simplification, R_D was estimated as

$$R_D(\tilde{z}, z, D) = \mathbb{E}\big[\big\{\big[1 - D(\tilde{z})\big]\big\|\nabla_{\tilde{z}}\text{log}D(\tilde{z})\big\|\big\}^2 + \big\{D(z)\big\|\nabla_{z}\text{log}D(z)\big\|\big\}^2\big] \tag{7.5}$$

where $\|.\|$ indicates the L2-norm.

The two losses L_D and L_Q were respectively optimized using SGD and Adam algorithms in our experiments. Both losses are opposing functions, L_D updates the discriminator to better differentiate positive samples $\tilde{z}$ generated by $\mathbf{P}$ from negative samples z computed by the encoder while L_Q updates the GAN generator, which is also the encoder of the AE, to fool the discriminator. Since L_D and L_Q update two ensembles of parameters, the use of two distinct optimizers simplifies the implementation. The choice of SGD algorithm for optimizing the discriminator D is suggested by [150], in which the researchers empirically found that SGD optimization tends to provide better results than adaptive algorithms for binary classification. Since D is also a binary classifier, the use of SGD is expected to be an appropriate choice while the standard Adam algorithm was employed for optimizing the generator $Q(z|X)$.

7.3.3 Gait index estimation

As mentioned in Section 7.2.2, our gait index is estimated as a combination of measures obtained from partial networks. The first measure is the reconstruction loss Υ_{AE} that is estimated as the Root-Mean-Square Error (RMSE) between an input X and its output $\widehat{X}$. The second operand of the combination is the probability $\Upsilon_{\mathbf{P}}$ that z is sampled from the prior distribution $\mathbf{P}$. This is a reasonable consideration since we expect that the AAE forces the distribution of trained latent variables z being similar to $\mathbf{P}$, a mapped $Q(z|X)$ of an abnormal gait posture should thus belong to a region of low probability density. The last measure, notated as Υ_D, is the output $p = D(z)$ of the discriminator. Concretely, the discriminator D should assign high values to normal walking postures and lower values to ones that are different from training samples since D has been fooled to consider the latent representation z of a normal posture as a positive sample.

It is obvious that the three terms Υ_{AE}, $\Upsilon_{\mathbf{P}}$ and Υ_D are non-negative, but the posture orders corresponding to these values are not the same. For example, a (very) normal posture should provide Υ_{AE} that tends to be near the low-end, while $\Upsilon_{\mathbf{P}}$ and Υ_D should be near the high-end of their range. The combination of the three measures is calculated according to a weighted sum as

$$\begin{aligned}\Upsilon_X &= w_{AE}\Upsilon_{AE} + w_{\mathbf{P}}\Upsilon_{\mathbf{P}} + w_D\Upsilon_D \\ &= w_{AE}\frac{\|X - \widehat{X}\|_2}{\sqrt{m_X}} + w_{\mathbf{P}} f_s(Q(z|X)|\mathbf{P}) + w_D D(Q(z|X))\end{aligned} \tag{7.6}$$

where m_X is the dimension of X and f_s is a range scaling operation that applies on a probability density function f as $f_s(Q(z|X)|\mathbf{P}) = \frac{f(Q(z|X)|\mathbf{P})}{f(0|\mathbf{P})}$. The denominator scales the output of f to the range [0, 1]. In our experiments, m_X was 256 since the size of cylindrical histograms was 16×16, and the prior distribution $\mathbf{P}$ was a multivariate normal distribution with zero mean and scalar covariance matrix. Therefore, $f(0|\mathbf{P})$ corresponds to the maximum value of f.

An unknown factor in eq. (7.6) is the weight values. We consider the combination of 2 and 3 quantities. The removal of a measure in the former case is performed by simply assigning its weight to zero in eq. (7.6). Since the three terms Υ_{AE}, $\Upsilon_{\mathbf{P}}$ and Υ_D are normalized in the range [0, 1], the weight of Υ_i is computed as $w_i = \frac{\sum_i s_i}{s_i}$ where s_i is the average value of the corresponding measure m_i calculated from training patterns (normal gaits) as in [97]. In other words, the weight calculation of a measure only depends on its values obtained in the training stage. The numerator is a constant in all the weights to facilitate the computation. After obtaining the weights, the gait index of a posture (i.e. a cylindrical histogram) is calculated according to eq. (7.6). The combination is expected to improve the gait index measure as follows. In the three measures Υ_{AE}, $\Upsilon_{\mathbf{P}}$ and Υ_D, the first one is the most significant factor since many studies demonstrated the ability of auto-encoder in anomaly detection (e.g. [91, 125]). This property is embedded into eq. (7.6) by w_{AE} that is much greater than $w_{\mathbf{P}}$ and w_D. Therefore, $\Upsilon_{\mathbf{P}}$ and Υ_D should be considered as additional factors to enhance the main indicator Υ_{AE}.

7.4 Experiments

7.4.1 Dataset

In order to evaluate the proposed method, we performed the gait index estimation [eq. (7.6)] on a dataset of 9 types of walking gaits including normal and abnormal ones that reduce the gait balance. These gait types were performed on a treadmill by 9 volunteers with the setup visualized in Fig. 7.1(a). The speed of the treadmill was 1.28 km/h that is appropriate for clinical experiments in practical situations. Beside normal gaits, the dataset includes simulations of two types of gait abnormality. The first one is frontal asymmetry where a sole with 3 different thicknesses (5/10/15 cm) was padded under one of the two feet. The second gait abnormality is the impairment of walking motion on each side of the body by attaching a weight of 4 kg to one ankle. The dataset was acquired by a Kinect 2 with a camera frame rate of 13 fps. Each gait of a subject was captured as a sequence of 1200 point clouds, 1200 frontal silhouettes and 1200 skeletons, synchronously. Details of the dataset can be found in [98, 107]. This dataset is available online at www.iro.umontreal.ca/~labimage/GaitDataset.

7.4.2 Assessment scheme

The evaluation was performed by considering gait indices in the task of distinguishing normal and abnormal gaits. The dataset was split into training and test sets under two schemes. The first one used the default separation suggested in [98] where the gaits of 5 subjects are available for the training stage, and the test set contains the 4 remaining ones. The other evaluation scheme was leave-one-out (on subject) cross-validation to get a more general assessment. We also reimplemented related works (including [12, 16, 17, 27, 97, 102, 114, 118]) that employ different data types to provide a comparison. These studies used various quantities for evaluation: classification accuracy in [16, 17, 27, 114], Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve in [102, 118] and Equal Error Rate (EER) in [97]. In our experiments, the gait index was used to decide the label (normal/abnormal) of an input sequence. The decision typically depends on a specific threshold. The ROC curve is a tool to assess the performance of a binary classifier, it is formed by plotting true positive rates (TPRs) vs. false positive rates

(FPRs) estimated from various thresholds. The AUC is calculated as the ratio between the area under the curve and the whole plotting area. Therefore, an AUC is normalized in the range [0, 1]. The higher the AUC, the better the performance of the system is. The EER is the classification error estimated from the threshold where the false positive rate is equal to the false negative rate, i.e. FPR $= 1 -$ TPR. In summary, AUC and EER are two assessment quantities that are commonly used in problems of binary classification. Our method introduces 6 possible gait indices (including Υ_{AE}, $\Upsilon_{\mathbf{P}}$, Υ_D and their combinations containing Υ_{AE}), there are thus 6 corresponding ROC curves. We used the EER to indicate the ability of each experimented method since this is related to the classification error and is estimated according to the ROC curve. Beside the per-frame assessment, the temporal factor was also considered by using the average measure over (non-overlapping) segments of consecutive frames as the gait indices. Such segment-based measure is usually considered as a better gait index indicator compared with the per-frame one as reported in [16, 97, 102].

As mentioned in [53], the GAN optimization attempts to converge to a saddle point instead of a minima, the loss is thus usually unstable during the training stage. Since there is not an obvious criterion to stop training, we performed the evaluation on a range of 100 training epochs where the GAN-related losses were sufficiently stable. Concretely, we trained the AAE for 500 epochs and selected the models in a period of 100 epochs so that the losses did not suddenly change, a collection of EERs was then estimated for each AAE based on outputted measures ($\Upsilon_{\mathbf{P}}$, Υ_D, Υ_{AE} and its 3 combinations), and the average EER of each measure was finally considered as an indicator of the method ability. Details of our weight estimation for the measure combinations were given in Section 7.3.3. A visualization of losses in our training stage is presented in Fig. 7.4. The figure shows that the GAN losses were less stable after the 370$^{\text{th}}$ epoch, a range of 200-300 was thus selected. It is also obvious that the reconstruction loss L_{AE} quickly converged after a few epochs.

7.4.3 Experimental results

First, we consider the separation where the training and test sets respectively contain 5 and 4 subjects. Remember that our AAE was trained using only normal gaits. The ability of the three measures for the task of distinguishing normal and abnormal gaits is

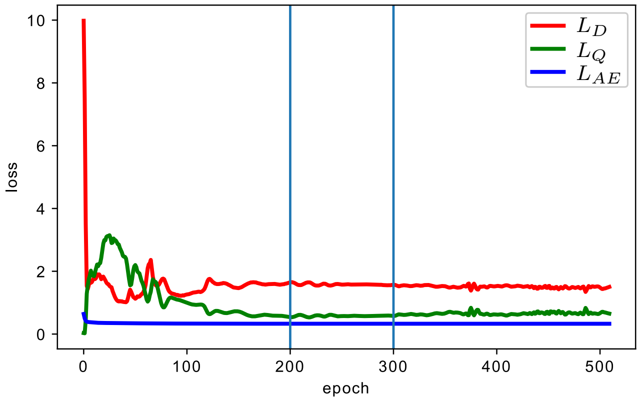
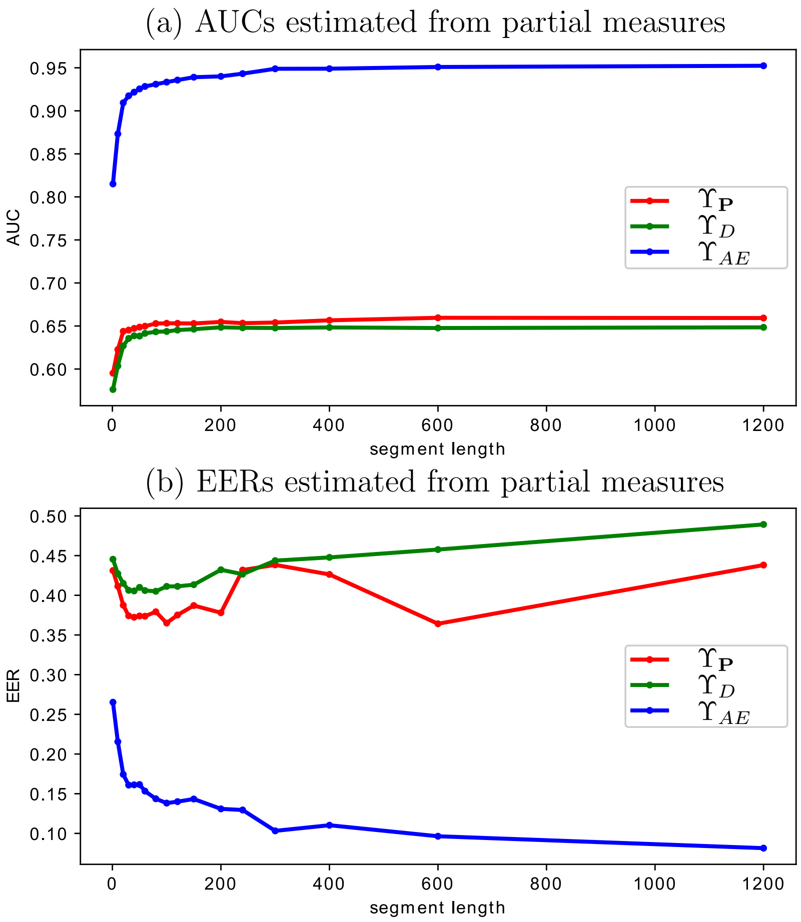


FIGURE 7.4: The change of AAE losses during first 500 training epochs. The training set includes normal walking gaits of 5 subjects. Our evaluation was performed on the epochs from 200 to 300.

indicated in Fig. 7.5. The reconstruction loss Υ_{AE} is a good measure since its EERs were low and quickly decreased when increasing the segment length. Therefore, Υ_{AE} should be used as the main factor in further combinations. The two others ($\Upsilon_{\mathbf{P}}$ and Υ_D), however, are not individually good indicators since their EERs were very high and AUCs, low.

In order to enhance Υ_{AE} using the other measures, we attempted to perform some combinations. We observed that combining Υ_{AE} and the output of discriminator Υ_D decreased the EER while the opposite is true when we replaced Υ_D by $\Upsilon_{\mathbf{P}}$. We empirically found that this unwanted effect might be avoided when $\Upsilon_{\mathbf{P}}$ was raised by a small exponent (i.e. $\Upsilon_{\mathbf{P}} \leftarrow (\Upsilon_{\mathbf{P}})^u$ where $0 < u < 1$). The exponent only changes the contribution of $\Upsilon_{\mathbf{P}}$ in its combination, while its AUC and EER are still unchanged (see Fig. 7.5) since the operation is monotonic. According to Fig. 7.6 (where $u = \frac{1}{8}$ after considering some small values), improving Υ_{AE} by both Υ_D and $\Upsilon_{\mathbf{P}}$ is recommended since its results were the best compared with the other combinations. Figure 7.6 also shows that the gait normality indicator tended to be better when using a higher value of temporal factor, i.e. estimating the gait index based on a longer sequence of point clouds.

As for the leave-one-out (on subject) cross-validation, 9 AAEs were trained and evaluated according to 9 different data separations of ratio 8:1. AUCs and EERs are shown in Fig. 7.7. When combined with Υ_D and $\Upsilon_{\mathbf{P}}$, the reconstruction-based measure Υ_{AE} was slightly improved for assessing the gait normality. Let us notice that the selected epoch ranges of the 9 AAEs in the leave-one-out cross-validation were different depending on the stability of their training losses.


(a) AUCs estimated from partial measures

(b) EERs estimated from partial measures

FIGURE 7.5: The average AUCs and EERs of the three partial measures estimated on segments of various lengths (including the per-frame assessment where the length is 1). The evaluation was performed according to the selected epoch period in Fig. 7.4.

7.4.4 Comparison

As mentioned in Section 7.4.2, related studies [16, 97, 102] were reimplemented and evaluated on our dataset under different input types. Bauckhage *et al.* [16] detected abnormal walking gaits based on a sequence of frontal silhouettes. The feature of each silhouette was extracted by fitting a lattice, and the posture was then described as a vector of some 2D corners that are pre-selected. The researchers embedded the temporal factor to improve their method by concatenating such consecutive vectors. The classification was performed using Support Vector Machines (SVMs) trained on multiple gait classes. Considering that objective under a different perspective, study [102] proposed another approach based on a sequence of 3D skeletons. The task of abnormal gait detection was performed according to an unsupervised (one-class) learning since defining specific abnormal gait types as in [16] may reduce the generalization of the system in practical

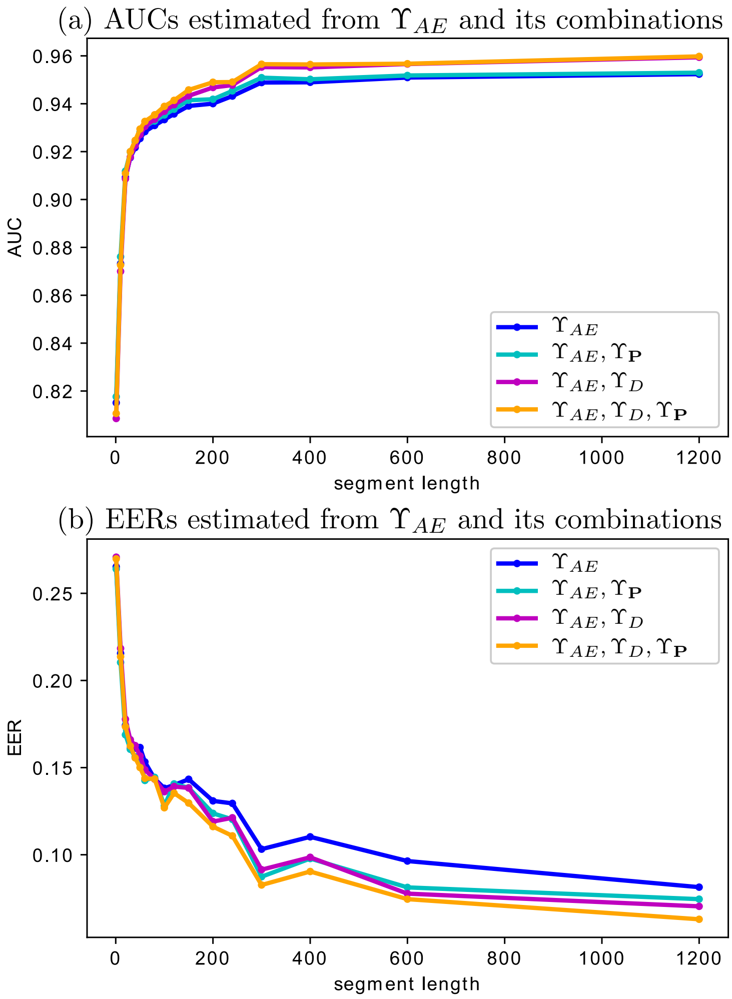
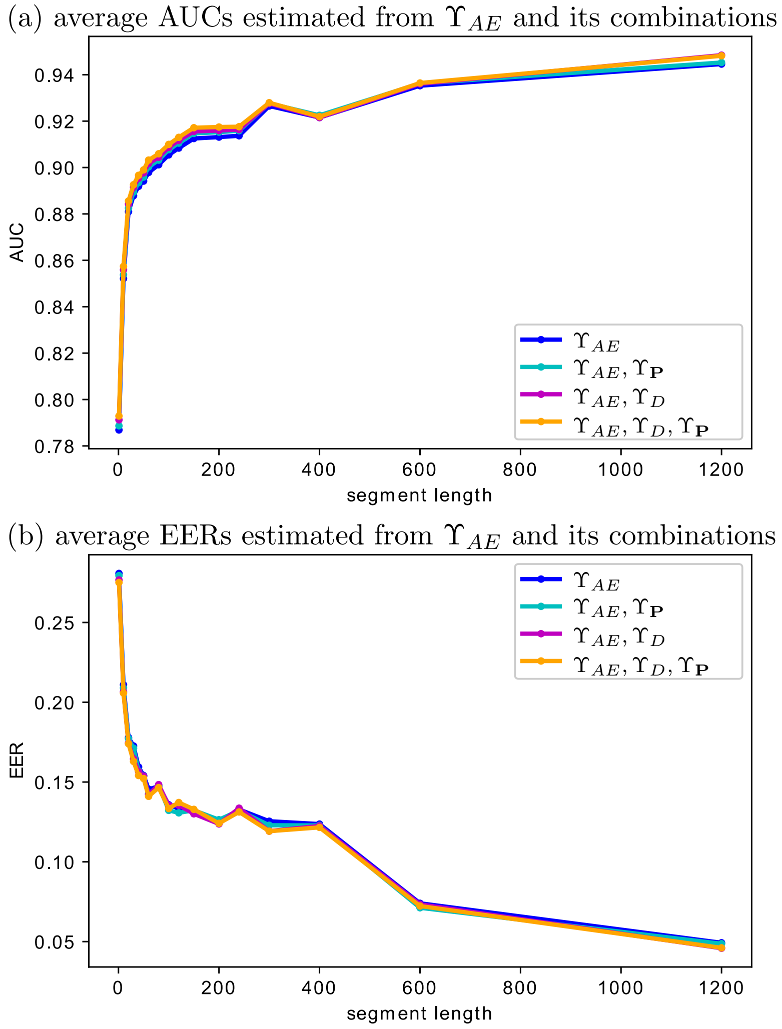


FIGURE 7.6: The average AUCs and EERs of Υ_{AE}'s possible combinations estimated with different segment lengths. The AAE was evaluated according to the suggested 5:4 separation.

applications. Besides, the temporal factor was directly embedded in the stage of feature extraction. Concretely, the 3D skeleton in each frame was described by a vector of geometric quantities, and a sequence of such vectors corresponding to a gait cycle was then employed as a unit of gait representation. The gait index was provided by a HMM that described the change of postures within normal gait cycles. The method reported in [97] estimated a gait normality index as a combination of two scores. The first one was determined by employing a HMM to measure the change of key points detected in consecutive depth maps. The second score was estimated by a cross-correlation on sequences of left and right projections of frontal silhouettes. The two scores were calculated with the support of a sliding window.

Unlike the three methods mentioned above, the remaining approaches considered in our



(a) average AUCs estimated from Υ_{AE} and its combinations

(b) average EERs estimated from Υ_{AE} and its combinations

FIGURE 7.7: The average AUCs and EERs estimated in the leave-one-out evaluation stage. The discriminator output Υ_D slightly enhanced the reconstruction-based measure Υ_{AE}.

comparison directly analyzed walking gaits on the whole sequence. Prabhu *et al.* [114] applied Recurrence Quantification Analysis (RQA) [149] to extract the recurrence nature of the walking gait signals. The determined features were then combined with typical statistical quantities to fully describe the gait information. The task of gait classification was performed and evaluated using SVMs and Probabilistic Neural Networks (PNNs). Similarly, Ren *et al.* [118] emphasized gait frequency factors by decomposing input signals into a finite set of intrinsic mode functions with the support of Empirical Mode Decomposition (EMD) [64] and then considered the association as well as the inherent relations between them. Bei *et al.* [17] also focused on periodic factors, but the features including gait symmetry, step length and gait cycle were manually determined on each sequence of skeletons. To demonstrate the potential of their proposed gait characteristics, K-means and Bayesian methods were employed for the gait categorization. A more typical approach was proposed by Chaaraoui *et al.* [27], in which each sequence of skeletons was split according to a sliding window and the classification was then performed based on the bag-of-words scheme. Differently from these studies, Auvinet *et al.* [12] compressed an input sequence of frontal depth maps into a Mean Gait Cycle Model (MGCM) to estimate a gait symmetry index. Concretely, the index was defined as the longitudinal spatial difference between two legs. However, this method considered only a particular region of lower limbs.

We reimplemented a HMM for [102], a HMM and a cross-correlation procedure for [97]. A one-class SVM was considered as a modification of the method [16] to be used for a training set of only normal gait samples (similarly to [97, 102] and our work). The evaluation was also performed on the suggested separation in [98] as well as the leave-one-out cross-validation. Beside the assessment on a short sequence of frames (called *per-segment*), i.e. feature concatenation of $\Delta = 21$ consecutive frames for [16], automatically determined gait cycle for [102], $\Delta = 10$ frames within a sliding window for [97] and $\Delta = 60$ clouds for our method, we also considered the decision over the entire sequence of 1200 frames (so-called *per-sequence*). The decision was determined by an alarm trigger in [16], smallest average log-likelihood of triple continuous cycles in [102], and simply the mean score in [97] as well as ours. In experiments of the remaining studies, we reimplemented a PNN for [114], K-means and Bayesian inference for [17], a random forest and a multilayer perceptron for [118], a bag-of-words model for [27] and finally the typical ROC-based evaluation on MGCM for [12]. The assessment of all these 5 methods was performed on entire sequence of inputs. Details of the obtained results are respectively shown in Table 7.2 and Table 7.3 for the evaluations according to the suggested data separation and the leave-one-out cross-validation scheme applied on each subject.

Let us first consider the three studies [16, 97, 102] which are capable to perform the assessment on short segments and full sequences of frames. The two tables show that gait description over a long sequence was more reliable than considering short segments in all evaluated methods. The EERs resulting from Υ_{AE} and its combination with both $\Upsilon_{\mathbf{P}}$ and Υ_D were lower than the others in the leave-one-out cross-validation as well as in the per-sequence assessment according to the suggested separation. Let us notice the difference between the sequence-based assessments in [16, 102] and ours. Those two studies proposed non-linear computations on the per-segment results to obtain a reliable

Table 7.2: Classification errors obtained from training and testing sets suggested in [98].

Data split	Model	Input type	Classification error (≈ EER)		
			per-frame	per-segment	per-sequence
5:4 separation	Probabilistic neural network [114]	gait signal (adapted)	-	-	0.167
	K-means [17]	skeleton	-	-	0.222
	Bayesian inference [17]	skeleton	-	-	0.111
	Random forest [118]	gait signal (adapted)	-	-	0.222
	Multilayer perceptron [118]	gait signal (adapted)	-	-	0.194
	Bag-of-Words [27]	skeleton	-	-	0.167
	Mean gait cycle model [12]	depth map	-	-	0.250
	Hidden Markov model [102]	skeleton	-	0.335	0.250
	One-class SVM [16]	silhouette	0.399	0.227	0.139
	Hidden Markov model [97]	depth map	-	0.396	0.281
	Cross-correlation [97]	silhouette	-	0.381	0.250
	HMM + cross-correlation [97]	depth map + silhouette	-	0.377	0.218
	Υ_{AE}	point cloud	0.265	0.153	0.081
	$\Upsilon_{AE} + \Upsilon_{\mathbf{P}}$	point cloud	**0.264**	**0.143**	0.075
	$\Upsilon_{AE} + \Upsilon_{D}$	point cloud	0.271	0.149	0.070
	$\Upsilon_{AE} + \Upsilon_{\mathbf{P}} + \Upsilon_{D}$	point cloud	0.270	0.144	**0.063**

Table 7.3: Classification errors obtained from the leave-one-out (on subject) cross-validation scheme.

Data split	Model	Input type	Classification error (≈ EER)		
			per-frame	per-segment	per-sequence
leave-one-out	Probabilistic neural network [114]	gait signal (adapted)	-	-	0.148
	K-means [17]	skeleton	-	-	0.259
	Bayesian inference [17]	skeleton	-	-	0.099
	Random forest [118]	gait signal (adapted)	-	-	0.160
	Multilayer perceptron [118]	gait signal (adapted)	-	-	0.160
	Bag-of-Words [27]	skeleton	-	-	0.198
	Mean gait cycle model [12]	depth map	-	-	0.125
	Hidden Markov model [102]	skeleton	-	0.396	0.198
	One-class SVM [16]	silhouette	0.418	0.274	0.136
	Hidden Markov model [97]	depth map	-	0.473	0.431
	Cross-correlation [97]	silhouette	-	0.321	0.097
	HMM + cross-correlation [97]	depth map + silhouette	-	0.319	0.083
	Υ_{AE}	point cloud	0.281	0.145	0.049
	$\Upsilon_{AE} + \Upsilon_{\mathbf{P}}$	point cloud	0.279	0.143	0.049
	$\Upsilon_{AE} + \Upsilon_{D}$	point cloud	0.277	0.142	0.046
	$\Upsilon_{AE} + \Upsilon_{\mathbf{P}} + \Upsilon_{D}$	point cloud	**0.275**	**0.141**	**0.046**

gait indicator. In other words, such segment-based measure might be noisy and the non-linear operations performed noise removal to keep a small piece of useful information. Unlike them, every per-frame measure in our work has an equal contribution to the index estimation. The method [97] also used the same scheme but was affected by another drawback: the lack of generalization. This was clearly shown in Table 7.2 and Table 7.3 where its per-sequence EERs were significantly reduced in the leave-one-out evaluation compared with the case of testing on 4 subjects. The number of training subjects in the two cases was 8 and 5, respectively. Therefore, it is reasonable to guess that the method [97] requires a large training dataset to provide a usable system. Recall that our

Table 7.4: Classification errors when evaluating gait index with the support of a sliding window.

Model	5:4 separation		leave-one-out	
	$\Delta = 10$	$\Delta = 21$	$\Delta = 10$	$\Delta = 21$
One-class SVM [16]	-	0.227	-	0.274
HMM [97]	0.396	-	0.473	-
Cross-correlation [97]	0.381	-	0.321	-
HMM + cross-corr. [97]	0.377	-	0.319	-
Υ_{AE}	0.211	0.174	0.207	0.169
$\Upsilon_{AE} + \Upsilon_{\mathbf{P}}$	**0.207**	**0.169**	0.206	0.168
$\Upsilon_{AE} + \Upsilon_{D}$	0.216	0.176	0.203	0.166
$\Upsilon_{AE} + \Upsilon_{\mathbf{P}} + \Upsilon_{D}$	0.213	0.171	**0.202**	**0.165**

AAE was designed with a simple architecture, we can thus expect to improve the model by carefully choosing component structures as well as tuning hyperparameters.

We next evaluate the efficiency of typical machine learning methods in the remaining approaches where the gait analysis was performed directly on input sequences. The use of K-means on manually extracted gait parameters in [17] was not an appropriate selection since the classification errors in both evaluation schemes were greater than 20%. However, replacing K-means by Bayesian inference seems promising since this reduced the error to around 0.1, the best one in this group of methods. Also, the use of neural networks and random forest in [114, 118] did not provide the desired results. This might be due to the lack of gait factor consideration in the feature extraction stage. In detail, using only time series analysis techniques such as RQA and EMD was not enough to determine distinguishable characteristics of pathological gaits. Another possible reason was that the gait signal for the experiments on these two methods was approximated from existing data. We might expect better results when combining [114, 118] with signal obtained from more sensitive devices such as inertial sensors. In the two remaining studies [12, 27], the researchers respectively considered only a portion of temporal and spatial information provided from the input sequence. Concretely, method [27] focused on combined poses, i.e. concatenations of $\tau = 35$ consecutive skeletons, and replaced them by specific key poses. This substitution was equivalent to a partial compression along the temporal axis that possibly led to the missing of informative poses. This drawback also occurred in [12] since only a small region of legs was considered for measuring the gait symmetry. We believe that further investigation on discarded features may improve the efficiency of the two approaches.

Let us notice that the choice of segment length $\Delta = 21$ and $\Delta = 10$ respectively has a significant effect in [16, 97] since these hyperparameters define the input of their models. Our approach, however, does not directly consider such temporal factor in the stage of model formation. Therefore, the per-segment evaluation of our method is an option where the segment length can be tuned depending on particular setup, objective, or application. These segments were non-overlapping to reduce the required computational cost. In order to emphasize the better ability of the proposed method compared with the others, a per-segment evaluation using sliding windows is presented in Table 7.4. This table shows that our method provided better results in describing gait index using a sliding window with small width. Notice that $\Delta = 21$ and $\Delta = 10$ were respectively recommended in [16, 97] and were not optimal values for our approach. Therefore, a careful selection of such quantity is expected to improve our results (similarly to Fig. 7.6 and 7.7). Once again, the combination of the 3 measures provided best results in the phase of leave-one-out evaluation even with a very small window's width.

7.4.5 Effect of histogram size

As mentioned in Section 7.3.1, the resolution of the cylindrical histogram is a hyperparameter that must be assigned in the model formation. It is reasonable to guess that a low resolution histogram might not be efficient to describe gait characteristics since each 3D sector could cover a large space of multiple body parts. However, a histogram of high resolution would increase the computational cost and might be easily affected by noise since its bin considers a small region. In order to evaluate the importance of this factor, we performed experiments on various sizes $h \times w$ of histogram where $(h, w) \in \{4, 8, 16, 32\}^2$. The evaluation scheme was leave-one-out cross-validation and we considered the per-sequence gait index provided by the combination of all the three measures Υ_{AE}, $\Upsilon_{\mathbf{P}}$ and Υ_D.

Since the input size was changed during these experiments, we also adapted the number of units in the remaining layers in Table 7.1 (excluding the output of discriminator $D(z)$ where only one unit was used to indicate a probability). The adaptation was performed proportionally to the histogram size, in which the reference was the values in Table 7.1. For example, when the input size was 8×16, each number of units in our AAE was also reduced half, i.e. (128, 48, 8) for encoder $Q(z|X)$, (8, 48, 128) for decoder $P(\hat{X}|z)$ and (8, 48, 1) for discriminator $D(z)$. This structure was also used for the input of sizes 16×8,

TABLE 7.5: EERs obtained in experiments on various sizes $h \times w$ of cylindrical histograms.

$h \backslash w$	4	8	16	32
4	0.194	0.301	0.067	0.059
8	0.068	0.123	0.055	0.077
16	0.124	0.102	0.055	0.094
32	0.132	0.102	**0.047**	0.103

32×4 and 4×32. The obtained EERs are given in Table 7.5. Let us notice that the result corresponding to the input size 16×16 was slightly different compared with Table 7.3 because we applied a procedure of epoch range selection instead of a manual selection as in Fig. 7.4 to automate the process and avoid any subjectivity.

Table 7.5 shows that a family of histogram sizes with small w (such as 4 and 8) is not efficient to emphasize characteristics of a walking gait since most values in the first two columns are greater than those of the two latter ones. Therefore, it is recommended to first consider the histogram width in order to find an appropriate size. It is also noticeable that the model ability tends to reduce together with increasing h when $w = 32$. This demonstrated our hypothesis of noise effect when the 3D sector was too small. Finally, the use of $w = 16$ seems to be an appropriate choice with $h = 8$ or higher. The 16×16 histogram tested in this work was thus appropriate although not optimal. Further investigation is planned in the future on this hyperparameter.

7.5 Conclusion

Adversarial auto-encoder and most GAN-based models have been employed for the task of data generation. This chapter introduces another use of AAE to deal with a practical problem, i.e. gait abnormality index estimation that can be applied for screening patients for example. The proposed approach focuses on the combination of measures provided from partial model components. The experiments demonstrate that an AAE has a great potential to work as a gait index estimator since such AAE with a very simple structure outperformed related studies that deal with various input types. The model can thus be expected to get better results by carefully tuning the architecture and related hyperparameters. Besides, finding an efficient criterion for stopping the AAE training is also a significant work to extend our study. In addition, considering other ways of combining different quantities (Υ_i) could help to improve the ability of our system for the task of gait index estimation as well as for other similar applications.

Chapter 8

Conclusion

The evaluation of human walking gait has received a great attention in the scientific and medical literature as it is one of the key elements in the diagnosis of locomotion problems in health systems. In general, highly sophisticated multi-camera motion capture systems are popularly used. However, they require specific equipments of high price, a methodical and skillful manual intervention and a high computing power.

In order to reduce the cost of these devices, we proposed a much simpler gait analysis system that uses only one depth camera. Concretely, multiple cameras are replaced by a single depth sensor and mirrors. Each mirror in our configuration plays the role of a camera that captures the scene from another point of view. Since we only use one camera, synchronization can be avoided, device costs are reduced, and the system is significantly simplified. Our system aims to perform 3D reconstruction of patient's walking postures to provide point clouds for the successive stage of gait index estimation.

In this dissertation, we have proposed a number of approaches dealing with the two sub-tasks: (1) 3D reconstruction using a depth camera and mirrors, and (2) performing gait analysis according to such reconstructed 3D point clouds of walking subjects. Each particular work was presented in a chapter and the corresponding publication was also provided.

To provide an overview of researches that are related to our objective, we briefly presented in Chapter 2 typical methods for 3D reconstruction as well as gait analysis, and some particularly close studies together with their limitations. Some recent approaches working on mirror-based reconstruction were also introduced to emphasize the difference between them and ours in following chapters.

Regarding to the 3D reconstruction task, two types of depth sensors were considered in our studies: structured light and Time-of-Flight. In detail, Chapter 3 provided our preliminary approach for reconstructing 3D point cloud using Kinect 1 of structured-light depth estimation together with two mirrors. We presented the benefit of employing a depth camera instead of a color one for redundancy avoidance. The reliability of signal obtained according to mirror reflection was also demonstrated. This method was simple and easy to implement. However, we needed to enhance reconstructed point clouds since the quality of obtained 3D bodies did not contain enough details for a valuable gait analysis. Although applying preprocessing steps on captured depth map may improve its quality, our system focuses on a fast execution directly working on raw acquired data.

We thus in Chapter 4 replaced the Kinect 1 by its next generation that employs Time-of-Flight depth estimation scheme in order to obtain better depth maps. The use of Kinect 2 led to a trade-off: the point cloud contains more details but may be distorted. The reason is *multipath interference* effect that was significantly emphasized due to the strong reflection of mirror surface. Our main contribution in this chapter was proposing a solution for reducing such distortions. In addition, we also performed data acquisition providing a huge dataset of nearly 100,000 point clouds (together with silhouettes and 3D skeletons) of walking subjects with various gaits.

Given a sequence of 3D point clouds representing human walking postures reconstructed in Chapter 4, we described a preliminary method for gait symmetry assessment in Chapter 5. A simple feature called cylindrical histogram was proposed to represent each point cloud as a matrix with a very small number of elements compared with the number of points in the original data. Beside such dimensional reduction, it can deal with noisy points appearing in the data acquisition since we did not perform any enhancement step on captured depth maps. The assessment was performed using cross-correlation applied on sequences of these histograms. Although the experiments provided very promising results, the method significantly depends on each individual gait without having any reference of expected postures. In other words, a walking gait which is periodically wobbly toward left and right sides may still get a confident score of symmetry while it should not. Therefore, we focused on model-based approaches where expected walking gaits are embedded within the model.

In Chapter 6, we introduced a method that models common typical walking gaits and supports the task of gait normality index estimation. The mentioned cylindrical histogram was still used as the representation of each instant posture. Various auto-encoders with specific constraints on their structures were built for evaluation. Differently from the cross-correlation method in Chapter 5 where a sequence of postures was considered, the networks in Chapter 6 were fed with only single histograms. The temporal factor was employed outside the networks as a post-processing, and the impact of the length of histogram sequence was also evaluated. A comparison between the proposed model and recent auto-encoders that directly process 3D point clouds was also provided to demonstrate the efficiency of cylindrical histogram in the problem of gait index estimation. In addition, our networks are potentially capable to support researchers exploring the effect of particular body areas on normal walking gait in further studies.

Regarding to the method in Chapter 6, the networks need to be carefully designed. We thus in Chapter 7 attempted to improve the previous work under a new aspect: reducing the effort (e.g. time to spend) of designing the auto-encoder structures. In other words, we focused on simplifying the architecture of previous model while still having a comparable ability for the task of gait index estimation. An adversarial auto-encoder was proposed with very simple stacks of layers. The experimental results were promising despite the network simplicity. In addition, a portion of the network can be used to generate samples of cylindrical histogram representing instant walking postures. In further studies, the network can be modified to embed the temporal factor to generate complete walking gait sequences. Besides, stabilizing the training stage would also be an improvement since the optimization of the current network may encounter difficulty to reach an optimal state.

From the presented approaches, the dissertation provides the following helpful discussions that are promising for further extension works:

- According to experimental comparisons in Chapters 5, 6 and 7, the efficiency of gait index estimation in proposed methods working on sequences of 3D point clouds was better than related studies. Therefore, the use of 3D point cloud has a great potential for dealing with other gait analysis problems compared to typical inputs such as depth map and skeleton.
- We should notice the importance of cylindrical histogram that was employed to fit each body posture. Since the input point cloud may be noisy, each sector of the histogram is able to cover a large space where the portion of noise is less significant. Therefore, such histogram simplifies the task of analyzing body point clouds. Directly processing these clouds at point-level would require very complicated models

with a huge number of computations as experimented in Section 6.6. A study focusing on the problem of selecting optimal histogram size is an appropriate extension.

- The use of cross-correlation in Chapter 5 is sensitive to the determination of body-coordinate system. A significant deviation in the estimation of the system axes may lead to a bad index measurement because it directly affects the histogram formation and left-right separation. On the contrary, the networks in Chapters 6 and 7 are less dependent on that factor since they focus on posture matching rather than inner gait comparison. Therefore, modeling walking gait is encouraged in clinical scenarios where the body-coordinate system is not guaranteed to be well calibrated.

- Temporal factor is important for walking gait analysis since a single posture may not indicate enough information about patient's condition. In our approaches, the gait was modeled at the posture-level and the temporal factor was embedded as a post-processing beyond the networks. This selection is appropriate for our experimental configuration where the subject walking velocity is controlled by fixing the treadmill speed. The design of feeding single histogram into the networks thus reduces the computational cost of the system. In gait-related problems where each subject can walk with a free speed, the network should directly embed temporal factor to model not only single postures but also their relation. This operation can be performed on the input, e.g. using sliding windows with various widths for accumulating space-time characteristics, and/or within the model architecture such as applying a recurrent neural network or long short-term memory.

- Finally, regarding to the aspect of clinical use, our system could enable clinicians to perform more frequent patient screening, follow-up after surgery, treatment or assessing recovery after a stroke. Our networks in Chapters 6 and 7 also allow scientists to investigate interesting gait characteristics. For example, a careful consideration on network units after imposing a sparse constraint may provide useful information about which histogram sectors are mostly focused by the model and how they are combined together. Generation of specific walking gaits is also a promising extension as long as samples of such expected gaits are available.

In summary, these works have contributed to the design of a unique and affordable computer vision-based gait analysis system. As the Canadian population is aging, medical care and indirect costs from musculoskeletal problems will increase. The proposed gait analysis system could eventually help to increase the efficiency and accuracy with which physicians identify and diagnose significant abnormalities for more efficient treatment and recovery of the patient; a clear social benefit and economic advantage for Canada. These works also have a great potential to be extended and/or adapted for a wide range of applications (e.g. biometric identification, activity analysis, real-time moving object reconstruction) depending on demands and/or situations.

Appendix

A.1 Analysis of phase distortion

According to [95], the depth of a point is measured based on the phase delay of optical trajectories as

$$d = \frac{c\varphi}{4\pi f} \tag{1}$$

where the constant c is the speed of light, f is the modulation frequency of the IR emitter, and φ is the phase shift. The measured phase shift in the case of multipath interference is

$$\widetilde{\varphi} = tan^{-1}(\frac{\alpha_0 sin\varphi_0 + \sum_{i=1}^{K} \alpha_i sin\varphi_i}{\alpha_0 cos\varphi_0 + \sum_{i=1}^{K} \alpha_i cos\varphi_i}) \tag{2}$$

where K is the number of signals returning to the corresponding pixel of the considering point, and α denotes the amplitude.

In our setup, there are only two signal paths: the direct way which provides a true depth and the indirect one which affects this value (e.g. the two mentioned trajectories in Section 4.4.1). Besides, we also assume that these two signal amplitudes are similar because they touch the object only once. Eq. (2) thus could be simply approximated as

$$\widetilde{\varphi} = tan^{-1}(\frac{sin\varphi_D + sin\varphi_I}{cos\varphi_D + cos\varphi_I}) \tag{3}$$

where the subscripts D and I denote parameters of the direct and indirect signals, respectively.

By combining eq. (1) and (3), the relation between the measured depth d_K and the two elementary traveled ways d_D and d_I is

$$d_K = \frac{1}{2}\big(d_D + d_I\big) + k\frac{c}{4f} \tag{4}$$

where k is an integer. By performing some experiments, we found that 0 is the most appropriate value of k. It means that in the case of phase distortion, the measured depth could be approximated as a quarter of the total traveled lengths of the two elementary signals.

Bibliography

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA, 2016. USENIX Association. ISBN 978-1-931971-33-1. URL http://dl.acm.org/citation.cfm?id=3026877.3026899.

[2] Ahmed, F., Paul, P. P., and Gavrilova, M. L. Dtw-based kernel and rank-level fusion for 3d gait recognition using kinect. *The Visual Computer*, 31(6):915–924, Jun 2015. ISSN 1432-2315. doi: 10.1007/s00371-015-1092-0.

[3] Akay, A. and Akgul, Y. S. 3d reconstruction with mirrors and rgb-d cameras. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, pages 325–334, Jan 2014.

[4] Ali, N., Bajwa, K. B., Sablatnig, R., Chatzichristofis, S. A., Iqbal, Z., Rashid, M., and Habib, H. A. A novel image retrieval based on visual words integration of sift and surf. *PLOS ONE*, 11(6):1–20, 06 2016. doi: 10.1371/journal.pone.0157428. URL https://doi.org/10.1371/journal.pone.0157428.

[5] Ali, N., Zafar, B., Riaz, F., Hanif Dar, S., Iqbal Ratyal, N., Bashir Bajwa, K., Kashif Iqbal, M., and Sajid, M. A hybrid geometric spatial image representation for scene classification. *PLOS ONE*, 13(9):1–27, 09 2018. doi: 10.1371/journal.pone.0203339. URL https://doi.org/10.1371/journal.pone.0203339.

[6] Andrieu, C., Doucet, A., and Punskaya, E. *Sequential Monte Carlo Methods for Optimal Filtering*, pages 79–95. Springer New York, New York, NY, 2001. ISBN 978-1-4757-3437-9. doi: 10.1007/978-1-4757-3437-9_4. URL http://dx.doi.org/10.1007/978-1-4757-3437-9_4.

[7] Andriluka, M., Roth, S., and Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, June 2009. doi: 10.1109/CVPR.2009.5206754.

[8] Arami, A., Poulakakis-Daktylidis, A., Tai, Y. F., and Burdet, E. Prediction of gait freezing in parkinsonian patients: A binary classification augmented with time series prediction. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(9):1909–1919, Sep. 2019. doi: 10.1109/TNSRE.2019.2933626.

[9] Auvinet, E., Multon, F., and Meunier, J. Gait analysis with multiple depth cameras. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6265–6268, Aug 2011. doi: 10.1109/IEMBS.2011.6091546.

[10] Auvinet, E., Meunier, J., and Multon, F. Multiple depth cameras calibration and body volume reconstruction for gait analysis. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 478–483, July 2012. doi: 10.1109/ISSPA.2012.6310598.

[11] Auvinet, E., Multon, F., and Meunier, J. Lower limb movement asymmetry measurement with a depth camera. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6793–6796, Aug 2012. doi: 10.1109/EMBC.2012.6347554.

[12] Auvinet, E., Multon, F., and Meunier, J. New lower-limb gait asymmetry indices based on a depth camera. *Sensors*, 15(3):4605–4623, 2015. ISSN 1424-8220. doi: 10.3390/s150304605. URL http://www.mdpi.com/1424-8220/15/3/4605.

[13] Baker, R. Gait analysis methods in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3(1):4, 2006.

[14] Baker, R., McGinley, J. L., Schwartz, M. H., Beynon, S., Rozumalski, A., Graham, H. K., and Tirosh, O. The gait profile score and movement analysis profile. *Gait & Posture*, 30(3):265 – 269, 2009. ISSN 0966-6362. doi: https://doi.org/10.1016/j.gaitpost.2009.05.020. URL http://www.sciencedirect.com/science/article/pii/S0966636209001489.

[15] Bauckhage, C., Tsotsos, J. K., and Bunn, F. E. Detecting abnormal gait. In *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 282–288, May 2005. doi: 10.1109/CRV.2005.32.

[16] Bauckhage, C., Tsotsos, J. K., and Bunn, F. E. Automatic detection of abnormal gait. *Image and Vision Computing*, 27(1):108–115, 2009.

[17] Bei, S., Zhen, Z., Xing, Z., Taocheng, L., and Qin, L. Movement disorder detection via adaptively fused gait analysis based on kinect sensors. *IEEE Sensors Journal*, 18(17):7305–7314, Sep. 2018. ISSN 1530-437X. doi: 10.1109/JSEN.2018.2839732.

[18] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 153–160, Cambridge, MA, USA, 2006. MIT Press.

[19] Bernin, A. Kinect chess board meta pattern, 2010. URL http://livingplace.informatik.haw-hamburg.de/blog/wp-content/uploads/2010/11/kinect1.png. Online; accessed February 17, 2017.

[20] Bigy, A. A. M., Banitsas, K., Badii, A., and Cosmas, J. Recognition of postures and freezing of gait in parkinson's disease patients using microsoft kinect sensor. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 731–734, April 2015. doi: 10.1109/NER.2015.7146727.

[21] Bonnet, S. and Heliot, R. A magnetometer-based approach for studying human movements. *IEEE Transactions on Biomedical Engineering*, 54(7):1353–1355, July 2007. ISSN 0018-9294. doi: 10.1109/TBME.2007.890742.

[22] Boutaayamou, M., Schwartz, C., Stamatakis, J., Denoël, V., Maquet, D., Forthomme, B., Croisier, J.-L., Macq, B., Verly, J. G., Garraux, G., and Brüls, O. Development and validation of an accelerometer-based method for quantifying gait events. *Medical Engineering & Physics*, 37(2):226 – 232, 2015. ISSN 1350-4533. doi: https://doi.org/10.1016/j.medengphy.2015.01.001. URL http://www.sciencedirect.com/science/article/pii/S135045331500003X.

[23] Bradski, G. and Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[24] Bray, M., Koller-Meier, E., and Gool, L. V. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106 (1):116 – 129, 2007. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2005.09.013. URL http://www.sciencedirect.com/science/article/pii/S1077314206001731. Special issue on Generative Model Based Vision.

[25] Caspi, D., Kiryati, N., and Shamir, J. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, May 1998. ISSN 0162-8828. doi: 10.1109/34.682177.

[26] Castaneda, V. and Navab, N. Time-of-flight and kinect imaging. *Kinect Programming for Computer Vision*, 2011.

[27] Chaaraoui, A. A., Padilla-López, J. R., and Flórez-Revuelta, F. Abnormal gait detection with rgb-d devices using joint motion history features. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 7, pages 1–6. IEEE, 2015.

[28] Cheung, G. K. M., Baker, S., and Kanade, T. Visual hull alignment and refinement across time: a 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–375–82 vol.2, June 2003. doi: 10.1109/CVPR.2003.1211493.

[29] Coxeter, H. S. M. and Greitzer, S. L. *Geometry revisited*, volume 19. Maa, 1967.

[30] Criminisi, A., Shotton, J., and Konukoglu, E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012. ISSN 1572-2740. doi: 10.1561/0600000035. URL http://dx.doi.org/10.1561/0600000035.

[31] Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

[32] Davis, J. W. Hierarchical motion history images for recognizing human motion. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 39–46. IEEE, 2001.

[33] Deutscher, J. and Reid, I. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005. ISSN 1573-1405. doi: 10.1023/B:VISI.0000043757.18370.9c. URL http://dx.doi.org/10.1023/B:VISI.0000043757.18370.9c.

[34] Din, S. D., Godfrey, A., and Rochester, L. Validation of an accelerometer to quantify a comprehensive battery of gait characteristics in healthy older adults and parkinson's disease: Toward clinical and at home use. *IEEE Journal of Biomedical and Health Informatics*, 20(3):838–847, May 2016. ISSN 2168-2194. doi: 10.1109/JBHI.2015.2419317.

[35] Dorrington, A. A., Godbaz, J. P., Cree, M. J., Payne, A. D., and Streeter, L. V. Separating true range measurements from multi-path and scattering interference in commercial range cameras. In *IS&T/SPIE Electronic Imaging*, pages 786404–786404. International Society for Optics and Photonics, 2011.

[36] Duckworth, T. and Roberts, D. J. Camera image synchronisation in multiple camera real-time 3d reconstruction of moving humans. In *Distributed Simulation and Real Time Applications (DS-RT), 2011 IEEE/ACM 15th International Symposium on*, pages 138–144, Sept 2011. doi: 10.1109/DS-RT.2011.15.

[37] Dupuis, Y., Savatier, X., and Vasseur, P. Feature subset selection applied to model-free gait recognition. *Image and Vision Computing*, 31(8):580 – 591, 2013. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2013.04.001. URL http://www.sciencedirect.com/science/article/pii/S0262885613000644.

[38] Eichner, M., Ferrari, V., and Zurich, S. Better appearance models for pictorial structures. In *BMVC*, volume 2, page 5, 2009.

[39] Epstein, E., Granger-Piché, M., and Poulin, P. Exploiting mirrors in interactive reconstruction with structured light. In *Vision, Modeling, and Visualization 2004: Proceedings, November 16-18, 2004, Standford, USA*, page 125. IOS Press, 2004.

[40] Esteban, C. H., Vogiatzis, G., and Cipolla, R. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[41] Felzenszwalb, P., McAllester, D., and Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587597.

[42] Felzenszwalb, P. F. and Huttenlocher, D. P. Efficient matching of pictorial structures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, pages 66–73 vol.2, June 2000. doi: 10.1109/CVPR.2000.854739.

[43] Felzenszwalb, P. F. and Huttenlocher, D. P. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. ISSN 1573-1405. doi: 10.1023/B:VISI.0000042934.15159.49. URL http://dx.doi.org/10.1023/B:VISI.0000042934.15159.49.

[44] Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[45] Freedman, D., Smolin, Y., Krupka, E., Leichter, I., and Schmidt, M. *SRA: Fast Removal of General Multipath for ToF Sensors*, pages 234–249. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10590-1.

[46] Gall, J., Rosenhahn, B., Brox, T., and Seidel, H.-P. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87(1):75, 2008. ISSN 1573-1405. doi: 10.1007/s11263-008-0173-1. URL http://dx.doi.org/10.1007/s11263-008-0173-1.

[47] Gall, J., Potthoff, J., Schnörr, C., Rosenhahn, B., and Seidel, H.-P. Interacting and annealing particle filters: Mathematics and a recipe for applications. *Journal of Mathematical Imaging and Vision*, 28(1):1–18, 2007. ISSN 1573-7683. doi: 10.1007/s10851-007-0007-8. URL http://dx.doi.org/10.1007/s10851-007-0007-8.

[48] Geman, S. and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, PAMI-6(6):721–741, Nov 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596.

[49] Geng, J. Structured-light 3d surface imaging: a tutorial. *Adv. Opt. Photon.*, 3 (2):128–160, Jun 2011. doi: 10.1364/AOP.3.000128. URL http://aop.osa.org/abstract.cfm?URI=aop-3-2-128.

[50] Georgiou, T. *Rhythmic Haptic Cueing for Gait Rehabilitation of Hemiparetic Stroke and Brain Injury Survivors*. PhD thesis, The Open University, 2018.

[51] Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[52] Gluckman, J. and Nayar, S. K. Catadioptric stereo using planar mirrors. *International Journal of Computer Vision*, 44(1):65–79, Aug 2001. ISSN 1573-1405. doi: 10.1023/A:1011172403203. URL https://doi.org/10.1023/A:1011172403203.

[53] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[54] Gouiaa, R. and Meunier, J. Human posture classification based on 3d body shape recovered using silhouette and infrared cast shadows. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 73–78, Nov 2015. doi: 10.1109/IPTA.2015.7367099.

[55] Gouwanda, D. and Senanayake, S. M. N. A. Periodical gait asymmetry assessment using real-time wireless gyroscopes gait monitoring system. *Journal of Medical Engineering & Technology*, 35(8):432–440, 2011. doi: 10.3109/03091902.2011.627080. URL https://doi.org/10.3109/03091902.2011.627080.

[56] Gouwanda, D. and Senanayake, S. A. Identifying gait asymmetry using gyroscopes - a cross-correlation and normalized symmetry index approach. *Journal of Biomechanics*, 44(5):972 – 978, 2011. ISSN 0021-9290. doi: https://doi.org/10.1016/j.jbiomech.2010.12.013.

[57] Han, J. and Bhanu, B. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, Feb 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.38.

[58] Hansard, M., Lee, S., Choi, O., and Horaud, R. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Publishing Company, Incorporated, 2012. ISBN 1447146573, 9781447146575.

[59] Hartley, R. and Zisserman, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[60] Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL http://dx.doi.org/10.1162/neco.2006.18.7.1527.

[61] Horn, E. and Kiryati, N. Toward optimal structured light patterns. *Image and Vision Computing*, 17(2):87–97, 1999.

[62] Howcroft, J., Lemaire, E. D., Kofman, J., and McIlroy, W. E. Elderly fall risk prediction using static posturography. *PLOS ONE*, 12(2):1–13, 02 2017. doi: 10.1371/journal.pone.0172398. URL https://doi.org/10.1371/journal.pone.0172398.

[63] Hu, B., Brown, C., and Nelson, R. Multiple-view 3-d reconstruction using a mirror. *Technical Report TR863, Computer Science Dept.*, 2005.

[64] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 454 (1971):903–995, 1998. ISSN 1364-5021. doi: 10.1098/rspa.1998.0193.

[65] Huang, Q., Wang, W., and Neumann, U. Recurrent slice networks for 3d segmentation of point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[66] Huynh, H. H. *Vidéosurveillance pour appartements intelligents : application à la détection de prise de médicaments*. PhD thesis, 2010. URL http://www.theses.fr/2010AIX22133. Thèse de doctorat dirigée par Daniel, Marc et Meunier, Jean Mathématiques, informatique, automatique Aix Marseille 2 2010.

[67] Iwashita, Y., Baba, R., Ogawara, K., and Kurazume, R. Person identification from spatio-temporal 3d gait. In *2010 International Conference on Emerging Security Technologies*, pages 30–35, Sep. 2010. doi: 10.1109/EST.2010.19.

[68] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047270. URL http://doi.acm.org/10.1145/2047196.2047270.

[69] Jain, R. and Nagel, H. H. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):206–214, April 1979. ISSN 0162-8828. doi: 10.1109/TPAMI.1979.4766907.

[70] Jean, F., Bergevin, R., and Branzan Albu, A. Human gait characteristics from unconstrained walks and viewpoints. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1883–1888, Nov 2011. doi: 10.1109/ICCVW.2011.6130478.

[71] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3063-3. doi: 10.1145/2647868.2654889. URL http://doi.acm.org/10.1145/2647868.2654889.

[72] Jiang, S., Wang, Y., Zhang, Y., and Sun, J. *Real Time Gait Recognition System Based on Kinect Skeleton Feature*, pages 46–57. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16628-5. doi: 10.1007/978-3-319-16628-5_4.

[73] Jodoin, P.-M. Vision par ordinateur imn 559, Autumn 2011.

[74] Khamsi, M. A. and Kirk, W. A. *An introduction to metric spaces and fixed point theory*, volume 53. John Wiley & Sons, 2011.

[75] Khan, S. M., Yan, P., and Shah, M. A homographic framework for the fusion of multi-view silhouettes. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408897.

[76] Kim, H., Kim, Y., Ko, D., Kim, J., and Lee, E. C. Pointing gesture interface for large display environments based on the kinect skeleton model. In Park, J. J. J. H., Pan, Y., Kim, C.-S., and Yang, Y., editors, *Future Information Technology*, pages 509–514, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-642-55038-6.

[77] Kim, Y., Baek, S., and Bae, B.-C. Motion capture of the human body using multiple depth sensors. *ETRI Journal*, 39(2):181–190, 2017. doi: 10.4218/etrij.17.2816.0045. URL https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.17.2816.0045.

[78] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[79] Kohli, P. and Shotton, J. *Key Developments in Human Pose Estimation for Kinect*, pages 63–70. Springer London, London, 2013. ISBN 978-1-4471-4640-7. doi: 10.1007/978-1-4471-4640-7_4. URL http://dx.doi.org/10.1007/978-1-4471-4640-7_4.

[80] Kun, L., Inoue, Y., Shibata, K., and Enguo, C. Ambulatory estimation of knee-joint kinematics in anatomical coordinate system using accelerometers and magnetometers. *IEEE Transactions on Biomedical Engineering*, 58(2):435–442, Feb 2011. ISSN 0018-9294. doi: 10.1109/TBME.2010.2089454.

[81] Kutulakos, K. N. and Seitz, S. M. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[82] Langis, C., Greenspan, M., and Godin, G. The parallel iterative closest point algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 195–202, May 2001. doi: 10.1109/IM.2001.924434.

[83] Laurentini, A. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, Feb 1994. ISSN 0162-8828. doi: 10.1109/34.273735.

[84] Legarda-Sa, R., Bothe, T., Ju, W. P., *et al.* Accurate procedure for the calibration of a structured light system. *Optical Engineering*, 43(2):464–471, 2004.

[85] Lim, J. Optimized projection pattern supplementing stereo systems. In *2009 IEEE International Conference on Robotics and Automation*, pages 2823–2829, May 2009. doi: 10.1109/ROBOT.2009.5152786.

[86] López-Fernández, D., Madrid-Cuevas, F., Carmona-Poyato, A., noz Salinas, R. M., and Medina-Carnicer, R. A new approach for multi-view gait recognition on unconstrained paths. *Journal of Visual Communication and Image Representation*, 38:396 – 406, 2016. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2016.03.020. URL http://www.sciencedirect.com/science/article/pii/S1047320316300232.

[87] Lv, Z., Xing, X., Wang, K., and Guan, D. Class energy image analysis for video sensor-based gait recognition: A review. *Sensors*, 15(1):932–964, 2015. ISSN 1424-8220. doi: 10.3390/s150100932. URL http://www.mdpi.com/1424-8220/15/1/932.

[88] Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[89] Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.05644.

[90] Marks, M., Kingsbury, T., Bryant, R., Collins, J. D., and Wyatt, M. Measuring abnormality in high dimensional spaces with applications in biomechanical gait analysis. *Scientific reports*, 8(1):15481, 2018.

[91] Martinelli, M., Tronci, E., Dipoppa, G., and Balducelli, C. Electric power system anomaly detection using neural networks. In Negoita, M. G., Howlett, R. J., and

Jain, L. C., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 1242–1248, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30132-5.

[92] Matyunin, S., Vatolin, D., Berdnikov, Y., and Smirnov, M. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4, May 2011. doi: 10.1109/3DTV.2011.5877202.

[93] Moeslund, T. B., Hilton, A., and Krüger, V. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2–3):90–126, 2006. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2006.08.002. URL http://www.sciencedirect.com/science/article/pii/S1077314206001263. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour.

[94] Mostajabi, M., Maire, M., and Shakhnarovich, G. Regularizing deep networks by modeling and predicting label structure. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[95] Naik, N., Kadambi, A., Rhemann, C., Izadi, S., Raskar, R., and Bing Kang, S. A light transport model for mitigating multipath interference in time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–81, 2015.

[96] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011. doi: 10.1109/ISMAR.2011.6092378.

[97] Nguyen, T. N., Huynh, H. H., and Meunier, J. Assessment of gait normality using a depth camera and mirrors. In *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 37–41, Las Vegas, NV, USA, March 2018. doi: 10.1109/BHI.2018.8333364.

[98] Nguyen, T.-N. and Meunier, J. Walking gait dataset: point clouds, skeletons and silhouettes. Technical Report 1379, DIRO, University of Montreal, April 2018. URL http://www.iro.umontreal.ca/~labimage/GaitDataset/dataset.pdf.

[99] Nguyen, T.-N. and Meunier, J. Estimation of gait normality index based on point clouds through deep auto-encoder. *EURASIP Journal on Image and Video Processing*, 2019(1):60, May 2019. ISSN 1687-5281. doi: 10.1186/s13640-019-0466-z. URL https://doi.org/10.1186/s13640-019-0466-z.

[100] Nguyen, T.-N. and Meunier, J. Applying adversarial auto-encoder for estimating human walking gait abnormality index. *Pattern Analysis and Applications*, Feb 2019. ISSN 1433-755X. doi: 10.1007/s10044-019-00790-7. URL https://doi.org/10.1007/s10044-019-00790-7.

[101] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Extracting silhouette-based characteristics for human gait analysis using one camera. In *Proceedings of the Fifth Symposium on Information and Communication Technology*, SoICT '14, pages 171–177, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2930-9. doi: 10.1145/2676585.2676612. URL http://doi.acm.org/10.1145/2676585.2676612.

[102] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Skeleton-based abnormal gait detection. *Sensors*, 16(11):1792, 2016. ISSN 1424-8220. doi: 10.3390/s16111792. URL http://www.mdpi.com/1424-8220/16/11/1792.

[103] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Human gait symmetry assessment using a depth camera and mirrors. *Computers in Biology and Medicine*, 101:174 – 183, 2018. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2018.08.021. URL http://www.sciencedirect.com/science/article/pii/S0010482518302415.

[104] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Estimating skeleton-based gait abnormality index by sparse deep auto-encoder. In *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, pages 311–315, July 2018. doi: 10.1109/CCE.2018.8465714.

[105] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Skeleton-based gait index estimation with lstms. In *2018 IEEE/ACIS 17th International Conference on Computer and*

Information Science (ICIS), pages 468–473, June 2018. doi: 10.1109/ICIS.2018.8466522.

[106] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Matching-based depth camera and mirrors for 3d reconstruction. In *Three-Dimensional Imaging, Visualization, and Display 2018, SPIE conference on*, volume 10666, pages 10666 – 10666 – 10, Orlando, FL, USA, April 2018. SPIE. doi: 10.1117/12.2304427. URL https://doi.org/10.1117/12.2304427.

[107] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. 3d reconstruction with time-of-flight depth camera and multiple mirrors. *IEEE Access*, 6:38106–38114, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2854262. URL https://doi.org/10.1109/ACCESS.2018.2854262.

[108] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Using tof camera and two mirrors for 3d reconstruction of dynamic objects. Technical Report 1380, DIRO, University of Montreal, April 2018. URL http://www.iro.umontreal.ca/~labimage/GaitDataset/reconstruct3D.pdf.

[109] Nguyen, T.-N., Huynh, H.-H., and Meunier, J. Measurement of human gait symmetry using body surface normals extracted from depth maps. *Sensors*, 19(4), 2019. ISSN 1424-8220. doi: 10.3390/s19040891. URL http://www.mdpi.com/1424-8220/19/4/891.

[110] Nonnekes, J., Goselink, R. J., Růžička, E., Fasano, A., Nutt, J. G., and Bloem, B. R. Neurological disorders of gait, balance and posture: a sign-based approach. *Nature Reviews Neurology*, 14(3):183, 2018.

[111] Ntawiniga, F. *Head motion tracking in 3D space for drivers.* PhD thesis, Université Laval, 2008.

[112] Pfister, A., West, A. M., Bronner, S., and Noah, J. A. Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis. *Journal of Medical Engineering & Technology*, 38(5):274–280, 2014. doi: 10.3109/03091902.2014.909540.

[113] Pollefeys, M., Koch, R., Vergauwen, M., and Van Gool, L. Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(4):251–267, 2000.

[114] Prabhu, P., Karunakar, A., Anitha, H., and Pradhan, N. Classification of gait signals into different neurodegenerative diseases using statistical analysis and recurrence quantification analysis. *Pattern Recognition Letters*, 2018. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2018.05.006. URL http://www.sciencedirect.com/science/article/pii/S0167865518301727.

[115] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[116] Queirolo, C. C., Silva, L., Bellon, O. R. P., and Segundo, M. P. 3d face recognition using simulated annealing and the surface interpenetration measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):206–219, Feb 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.14.

[117] Ramanathan, P., Steinbach, E. G., and Girod, B. Silhouette-based multiple-view camera calibration. In *VMV*, pages 3–10, 2000.

[118] Ren, P., Tang, S., Fang, F., Luo, L., Xu, L., Bringas-Vega, M. L., Yao, D., Kendrick, K. M., and Valdes-Sosa, P. A. Gait rhythm fluctuation analysis for neurodegenerative diseases by empirical mode decomposition. *IEEE Transactions on Biomedical Engineering*, 64(1):52–60, Jan 2017. ISSN 0018-9294. doi: 10.1109/TBME.2016.2536438.

[119] Rodriguez, S., Pérez, K., Quintero, C., López, J., Rojas, E., and Calderón, J. Identification of multimodal human-robot interaction using combined kernels. In Snášel, V., Abraham, A., Krömer, P., Pant, M., and Muda, A. K., editors, *Innovations in Bio-Inspired Computing and Applications*, pages 263–273, Cham, 2016. Springer International Publishing. ISBN 978-3-319-28031-8.

[120] Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2018–2028. Curran Associates, Inc., 2017.

[121] Rougier, C., Auvinet, E., Meunier, J., Mignotte, M., and De Guise, J. A. Depth energy image for gait symmetry quantification. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5136–5139. IEEE, 2011.

[122] Rusu, R. B. and Cousins, S. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[123] Sainath, T. N., Kingsbury, B., and Ramabhadran, B. Auto-encoder bottleneck features using deep belief networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4153–4156, March 2012. doi: 10.1109/ICASSP.2012.6288833.

[124] Sajid, M., Ali, N., Dar, S. H., Iqbal Ratyal, N., Butt, A. R., Zafar, B., Shafique, T., Baig, M. J. A., Riaz, I., and Baig, S. Data augmentation-assisted makeup-invariant face recognition. *Mathematical Problems in Engineering*, 2018, 2018.

[125] Sakurada, M. and Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2Nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, pages 4:4–4:11, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3159-3. doi: 10.1145/2689746.2689747. URL http://doi.acm.org/10.1145/2689746.2689747.

[126] Sapp, B., Toshev, A., and Taskar, B. *Cascaded Models for Articulated Pose Estimation*, pages 406–420. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15552-9. doi: 10.1007/978-3-642-15552-9_30. URL http://dx.doi.org/10.1007/978-3-642-15552-9_30.

[127] Schwartz, M. H. and Rozumalski, A. The gait deviation index: A new comprehensive index of gait pathology. *Gait & Posture*, 28(3):351 – 357, 2008. ISSN 0966-6362. doi: https://doi.org/10.1016/j.gaitpost.2008.05.001. URL http://www.sciencedirect.com/science/article/pii/S0966636208001136.

[128] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006. doi: 10.1109/CVPR.2006.19.

[129] Sharma, N., Ray, A. K., Sharma, S., Shukla, K., Aggarwal, L., and Pradhan, S. Segmentation of medical images using simulated annealing based fuzzy c means algorithm. *International Journal of Biomedical Engineering and Technology*, 2(3): 260–278, 2009.

[130] Shi, S., Wang, Q., Xu, P., and Chu, X. Benchmarking state-of-the-art deep learning software tools. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 99–104, Nov 2016. doi: 10.1109/CCBD.2016.029.

[131] Shinzaki, M., Iwashita, Y., Kurazume, R., and Ogawara, K. Gait-based person identification method using shadow biometrics for robustness to changes in the walking direction. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 670–677, Jan 2015. doi: 10.1109/WACV.2015.95.

[132] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, June 2011. doi: 10.1109/CVPR.2011.5995316.

[133] Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, Dec 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.241.

[134] Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C., and Moreno-Noguer, F. Single image 3d human pose estimation from noisy observations. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2673–2680, June 2012. doi: 10.1109/CVPR.2012.6247988.

[135] Smeureanu, S., Ionescu, R. T., Popescu, M., and Alexe, B. Deep appearance features for abnormal behavior detection in video. In Battiato, S., Gallo, G., Schettini, R., and Stanco, F., editors, *Image Analysis and Processing - ICIAP 2017*, pages 779–789, Cham, 2017. Springer International Publishing. ISBN 978-3-319-68548-9.

[136] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of*

Machine Learning Research, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[137] Stoica, P. and Moses, R. L. *Spectral analysis of signals*, volume 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[138] Szeliski, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[139] Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[140] Tong, J., Zhou, J., Liu, L., Pan, Z., and Yan, H. Scanning 3d full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18(4): 643–650, April 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.56.

[141] University of Utah and University of Nebraska. Neurologic exam. http://library.med.utah.edu/neurologicexam, 2016. Accessed: 2017-11-07.

[142] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. URL http://doi.acm.org/10.1145/1390156.1390294.

[143] Vitter, J. S. Faster methods for random sampling. *Commun. ACM*, 27(7):703–718, July 1984. ISSN 0001-0782. doi: 10.1145/358105.893. URL http://doi.acm.org/10.1145/358105.893.

[144] Wahid, F., Begg, R. K., Hass, C. J., Halgamuge, S., and Ackland, D. C. Classification of parkinson's disease gait using spatial-temporal gait features. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1794–1802, Nov 2015. doi: 10.1109/JBHI.2015.2450232.

[145] Wang, B., Liu, X., Xia, K., Ramamohanarao, K., and Tao, D. Random angular projection for fast nearest subspace search. In Hong, R., Cheng, W.-H., Yamasaki,

T., Wang, M., and Ngo, C.-W., editors, *Advances in Multimedia Information Processing – PCM 2018*, pages 15–26, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00776-8.

[146] Wang, X., Gao, L., Wang, P., Sun, X., and Liu, X. Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. *IEEE Transactions on Multimedia*, 20(3):634–644, March 2018. ISSN 1520-9210. doi: 10.1109/TMM.2017.2749159.

[147] Wasenmüller, O. and Stricker, D. *Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision*, pages 34–45. Springer International Publishing, Cham, 2017. ISBN 978-3-319-54427-4.

[148] Webb, J. and Ashley, J. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.

[149] Webber, C. L. and Marwan, N., editors. *Recurrence Quantification Analysis*. Springer International Publishing, 2015. URL http://dx.doi.org/10.1007/978-3-319-07155-8.

[150] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017.

[151] Wolfram Research Inc. Mathematica, Version 11.1, 2017. Champaign, IL, 2017.

[152] Xia, K., Ma, Y., Liu, X., Mu, Y., and Liu, L. Temporal binary coding for large-scale video search. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, pages 333–341, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4906-2. doi: 10.1145/3123266.3123273. URL http://doi.acm.org/10.1145/3123266.3123273.

[153] Yang, Y. and Ramanan, D. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392, June 2011. doi: 10.1109/CVPR.2011.5995741.

[154] Yang, Y., Deng, C., Gao, S., Liu, W., Tao, D., and Gao, X. Discriminative multi-instance multitask learning for 3d action recognition. *IEEE Transactions on Multimedia*, 19(3):519–529, March 2017. ISSN 1520-9210. doi: 10.1109/TMM.2016.2626959.

[155] Yang, Y., Deng, C., Tao, D., Zhang, S., Liu, W., and Gao, X. Latent max-margin multitask learning with skelets for 3-d action recognition. *IEEE Transactions on Cybernetics*, 47(2):439–448, Feb 2017. ISSN 2168-2267. doi: 10.1109/TCYB.2016.2519448.

[156] Yang, Y., Liu, R., Deng, C., and Gao, X. Multi-task human action recognition via exploring super-category. *Signal Processing*, 124:36 – 44, 2016. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2015.10.035. URL http://www.sciencedirect.com/science/article/pii/S0165168415003795.

[157] Yang, Y., Feng, C., Shen, Y., and Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[158] Ying, X., Peng, K., Hou, Y., Guan, S., Kong, J., and Zha, H. Self-calibration of catadioptric camera with two planar mirrors from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1206–1220, May 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.195.

[159] Yu, T.-H., Kim, T.-K., and Cipolla, R. Real-time action recognition by spatio-temporal semantic and structural forest. In *Proceedings of the British Machine Vision Conference*, pages 52.1–52.12. BMVA Press, 2010. ISBN 1-901725-40-5. doi:10.5244/C.24.52.

[160] Zafar, B., Ashraf, R., Ali, N., Ahmed, M., Jabbar, S., and Chatzichristofis, S. A. Image classification by addition of spatial information based on histograms of orthogonal vectors. *PLOS ONE*, 13(6):1–26, 06 2018. doi: 10.1371/journal.pone.0198175. URL https://doi.org/10.1371/journal.pone.0198175.

[161] Zafar, B., Ashraf, R., Ali, N., Ahmed, M., Jabbar, S., Naseer, K., Ahmad, A., and Jeon, G. Intelligent image classification-based on spatial weighted histograms of concentric circles. *Comput. Sci. Inf. Syst.*, 15(3):615–633, 2018. URL http://doiserbia.nb.rs/Article.aspx?id=1820-02141800025Z.

[162] Zafar, B., Ashraf, R., Ali, N., Iqbal, M. K., Sajid, M., Dar, S. H., and Ratyal, N. I. A novel discriminating and relative global spatial image representation with applications in cbir. *Applied Sciences*, 8(11), 2018. ISSN 2076-3417. doi: 10.3390/app8112242. URL http://www.mdpi.com/2076-3417/8/11/2242.

[163] Zeng, W., Liu, F., Wang, Q., Wang, Y., Ma, L., and Zhang, Y. Parkinson's disease classification using gait analysis via deterministic learning. *Neuroscience Letters*, 633:268 – 278, 2016. ISSN 0304-3940. doi: https://doi.org/10.1016/j.neulet.2016.09.043. URL http://www.sciencedirect.com/science/article/pii/S0304394016307261.

[164] Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., and Menegatti, E. Performance evaluation of the 1st and 2nd generation kinect for multimedia applications. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, June 2015. doi: 10.1109/ICME.2015.7177380.

[165] Zhang, L., Curless, B., and Seitz, S. M. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.

[166] Zhang, Z. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[167] Zuffi, S., Freifeld, O., and Black, M. J. From pictorial structures to deformable structures. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3546–3553, June 2012. doi: 10.1109/CVPR.2012.6248098.