3D COW IDENTIFICATION IN CATTLE FARMS

by

Ahmet Cumhur Arslan

B.S., Electrical & Electronics Engineering, Boğaziçi University, 2010

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

Graduate Program in Systems & Control Engineering
Boğaziçi University

2015

3D COW IDENTIFICATION IN CATTLE FARMS

APPROVED BY:

Prof. Mehmet Akar (Thesis Supervisor)	.....
Prof. Fatih Alagöz	.....
Prof. Yağmur Denizhan	.....

DATE OF APPROVAL: 15.06.2015

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Mehmet Akar, for his excellent guidance, and encouragement. He has supported me throughout this thesis with his patience and knowledge whilst allowing me the room to work in my own way.

I am grateful to Prof. Fatih Alagöz for his trustful and relaxing support. It was always a joy to be around him.

I am also grateful to Prof. Yağmur Denizhan for being an excellent academic advisor through my graduate studies.

I would like to thank to my friends, my professors and every one else constituting the memorable educational spirit of the Boğaziçi University.

Finally I would like to thank to my parents Mustafa Arslan and Nergiz Arslan. They are the meaning of my life and I have always felt their presence even from kilometers away. I dedicated this thesis to them as it would not be completed without their love and support.

The work of this thesis is partially supported by SAN-TEZ project #01109.STZ.2011-2.

ABSTRACT

3D COW IDENTIFICATION IN CATTLE FARMS

Animal farms have been steadily growing to meet the consumption requirements of the society in an efficient manner. This fact necessitates new monitoring and tracking systems to collect useful information about the herds in order to observe their general health and instantaneous state. However, recognizing and tracking an animal in a farm is a difficult task due to the target's similarity and hard to predict dynamics. In this thesis, a novel cow identification system is proposed. There are prominent features of this solution which differentiates it from the others in the literature, i.e., it does not need any markers or external devices placed on the animal; works in even unlighted environments; identifies even black cows without distinctive coat patterns; is relatively cheaper, and enables accurate positioning. Proposed solution is based on 3D shape analysis of the top back part of the animals captured with RGBD cameras placed at an adequate height, where two dimensional images are constructed with respect to the local surface features and are subsequently identified by using face recognition methods. To evaluate the applicability of the proposed system, a real-time prototype software has been developed and a 3D cattle dataset is acquired which, to our knowledge, is unique in the literature. This dataset is gathered from moving animals which do not have distinctive coat patterns and captured in different lighting conditions. Applicability of the proposed solution has been verified by testing with the acquired dataset. Convincing results are obtained where %88 of 50 cows are identified successfully.

ÖZET

İNEK ÇİFTLİKLERİNDEKİ HAYVANLARIN 3B KİMLİK SAPTAMASI

Hayvan çiftlikleri, toplumun artan ihtiyaçlarını etkin bir şekilde karşılayabilmek için giderek büyümektedir. Bu durum, hayvanlar hakkında faydalı bilgilerin toplanabilmesi ve anlık durumları ve sağlıklarının takibi için yeni izleme ve takip sistemlerini gerekli kılmaktadır. Ancak hayvanların kimliklendirmeleri ve takibi, benzerlikleri ve davranışlarının tahmininin kolay olmayışı sebebiyle zor bir problemdir. Bu tezde yeni bir inek tanımlama sistemi önerilmektedir. Önerilen çözümü li-teratürdeki diğer sistemlerden ayıran belirgin özellikler bulunmaktadır. Örneğin bu sistem, hayvanların üzerine koyulan işaretlere veya harici cihazlara ihtiyaç duymamaktadır, karanlık ortamlarda bile çalışabilmekte, ayırt edici görünüşleri olmayan siyah inekleri bile tanımlayabilmekte, göreceli olarak ucuz ve hassas yer tespiti sağlamaktadır. Önerilen çözüm, belirli yüksekliğe konan RGBD kameralarla çekilen hayvanların üst arka gövdesinin 3B şekil analizine dayanmaktadır. Yerel yüzey özelliklerine göre iki boyutlu imgeler oluşturulmakta ve bu imgeler yüz tanımlama algoritmaları kullanılarak kimliklendirilmektedir. Önerilen sistemin uygulanabilirliğini değerlendirmek amacıyla gerçek zamanda çalışan bir prototip yazılım geliştirilmiş ve bildiğimiz kadarıyla literatürde bulunmayan bir 3B sığır veri öbeği oluşturulmuştur. Bu veri öbeği, hareketli ve belirgin görünüşleri olmayan hayvanlardan farklı ışık koşullarında alınmıştır. Yapılan testlerde önerilen çözümün uygulanabilirliği alınan veri öbeği ile doğrulanmış olup, 50 ineğin %88'i doğru bir şekilde tanımlanabilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES ix
LIST OF TABLES xii
LIST OF SYMBOLS xiii
LIST OF ACRONYMS/ABBREVIATIONS xvii
1. INTRODUCTION 1
1.1. Indoor Animal Positioning Systems 1
1.2. Optical Positioning Systems 2
1.3. Proposed Optical Solution 4
1.3.1. Cameras 5
1.3.2. Identification Process 5
1.4. Contributions of the Thesis 8
1.5. Organization of the Thesis 8
2. METHODS AND ALGORITHMS 10
2.1. Depth Measurement of Kinect 10
2.2. Voxel Grid Filtering 12
2.3. Euclidean Clustering 12
2.4. RANSAC (RAndom SAmple Concensus) Algorithm 13
2.5. ICP (Iterative Closest Point) Algorithm 14
2.6. Eigenface Method 16
2.7. Fisherface Method 19
2.8. Summary of the Chapter 21
3. 3D CATTLE IDENTIFICATION 22
3.1. Data Acquisition and Animal Detection 22
3.2. Data Filtering 22

3.2.1. Down-sampling with Voxel Grid Filter . . . 23
3.2.2. Ground Plane Correction . . . 23
3.2.3. Ground Plane Extraction . . . 29
3.2.4. Euclidean Clustering . . . 30
3.3. Pose Normalization . . . 30
3.3.1. Rough Pose Normalization . . . 31
3.3.2. Extracting the Region of Interest . . . 34
3.3.3. Pose Normalization by Spine Recognition . . . 35
3.4. Image Creation . . . 39
3.4.1. Depth Image . . . 41
3.4.2. Normal-y Image . . . 42
3.4.3. Normal-z Image . . . 43
3.5. Identification with Images . . . 43
3.6. Summary of the Chapter . . . 45
4. IMPLEMENTATION AND DATASET ACQUISITION . . . 46
4.1. Implementation . . . 46
4.1.1. Debug Application . . . 46
4.1.2. Calibration Tool . . . 46
4.1.3. Dataset Grabber Tool . . . 48
4.1.4. Real-time Identification Application . . . 50
4.2. Dataset Acquisition . . . 53
4.3. Summary of the Chapter . . . 55
5. EXPERIMENTAL EVALUATION . . . 56
5.1. Eigenface Method . . . 58
5.2. Fisherface Method . . . 59
5.3. Effect of Spine Recognition . . . 60
5.4. Effect of Number of Samples . . . 61
5.5. Summary of the Chapter . . . 62
6. CONCLUDING REMARKS . . . 63
APPENDIX A: SAMPLE DEPTH IMAGES . . . 65

APPENDIX B: SAMPLE NORMAL-Y IMAGES 66
APPENDIX C: SAMPLE NORMAL-Z IMAGES 67
REFERENCES 68

LIST OF FIGURES

Figure 1.1. Kinect Sensor. 5

Figure 1.2. Sample configuration of the cameras. 6

Figure 1.3. Sample point cloud of a cow [1]. 6

Figure 1.4. The proposed 3D identification scheme. 7

Figure 2.1. Triangulation process [2]. 11

Figure 2.2. Example dataset and fitted line model. Blue dots are inliers, red dots are outliers. 15

Figure 3.1. Steps of data filtering process. 23

Figure 3.2. Raw point cloud, ${}^{C}P$, captured from the top of a cow. Left - view from the camera frame, Right - view from the reference frame. . . 24

Figure 3.3. Reference frame is located on the ground and is aligned with the center of the animal. 25

Figure 3.4. Normal vector ($\vec{v}$) of the detected ground plane and related angles for the rotations (θ, β). 27

Figure 3.5. Point cloud after the ground plane is corrected (${}^{R'}P$). 29

Figure 3.6. Left - Before extracting the ground, Right - After extracting the ground. 30

Figure 3.7. Left - Before clustering, Right - The biggest cluster, ${}^{A}P$. 31

Figure 3.8. Principle direction of a point cloud of a horse [3]. 32

Figure 3.9. Left - Before correction of center of mass and direction, Right - Center of mass is aligned with the origin, and the direction is parallel with the X axis. 33

Figure 3.10. Left - Point cloud of the animal (${}^{R}P$), Right - Region of interest (${}^{S}P$). 35

Figure 3.11. Steps of the proposed spine recognition and correction procedure. 36

Figure 3.12. Illustration of spine detection and correction procedure 37

Figure 3.13. Mapping from 3D Euclidean space to 2D image space. 40

Figure 3.14. Sample depth images captured from different cows. 42

Figure 3.15. Sample normal-y images captured from different cows. 42

Figure 3.16. Sample normal-z images captured from different cows. 43

Figure 4.1. Snapshot of the user interface of the debug application. 47

Figure 4.2. Manually selected keypoints. Left - scene from the first camera. Right - scene from the second camera. 48

Figure 4.3. Left - Transformed and merged point clouds with manually selected keypoints. Right - Same scene fine tuned with ICP algorithm. . . 49

Figure 4.4. Snapshots of the user interface of the real-time identification application. 52

Figure 4.5. 3D point clouds captured from two cameras placed at different points of the corridor. 53

Figure 5.1. Identification process. 58

LIST OF TABLES

Table 4.1. Number of samples of each cow in two datasets. 54

Table 5.1. Identification results of Eigenface method. 59

Table 5.2. Identification results of Fisherface method. 60

Table 5.3. Identification results of Eigenface method without spine correction. 61

Table 5.4. Identification results of Fisherface method without spine correction. 61

Table 5.5. Identification results of Eigenface method with switched datasets. . 62

Table 5.6. Identification results of Fisherface method with switched datasets. 62

LIST OF SYMBOLS

A	Set of difference images
a	Model parameter of a plane
b	Distance between the infrared camera and the laser emitter *or* model parameter of a plane
C	Covariance matrix *or* camera frame *or* number of classes
c	Model parameter of a plane
C_k	kth cluster
c_m	Point representing the mth cluster
d	Length of an image vector *or* measured disparity *or* model parameter of a plane
d_1	First principle direction
$d_{plane}(p_i)$	Distance between the ith point and the ground plane
f	Focal length
$f(\cdot)$	Mapping function
$f_{id}(\cdot)$	Function which predicts the ID of the image or image set
$f_{slice}(P, x)$	Slice of P at x
$f_{voxel}(P)$	Function which down-samples P with voxel grid filter
$f_{width}(P, x)$	Width of P at x
$g_{int}(\cdot)$	Function which converts a real number to an integer
k	Number of animals
l	Length threshold
M	Image
M_i	ith image
$m(x_m, y_m)$	Pixel located at the y_mth row and x_mth column
N	Number of samples
$\vec{n}$	Normal vector
n_x	x value of the normal vector
n_y	y value of the normal vector

n_z	z value of the normal vector
$n_{inliers}$	Number of of inliers
$n_{outliers}$	Number of of outliers
P	Point cloud
P_{inlier}	Point cloud of inliers
$P_{outlier}$	Point cloud of outliers
${}^{A}P$	Point cloud belonging to the animal
${}^{C}P$	Point cloud represented with respect to the camera frame
${}^{K}P$	Candidate key-points representing the spine
${}^{m}P$	Points mapped to the pixel m
${}^{N}P$	Neighboring points of p_i
${}^{R}P$	Point cloud represented with respect to the reference frame
${}^{S}P$	Point cloud of top back part of the animal
$p(x, y, z)$	Point whose coordinates are x, y, and z
$p'(x', y', z')$	Point $p(x, y, z)$ after transformation
p_b	Rearmost point of ${}^{R}P$ where width is more than ϵ_w
p_i	ith point
p'_i	Nearest neighbor of p_i
$\bar{p}_i$	Mean point
p_g	Any point representing the ground
p_m	Uppermost point related with the pixel m
p_t	Uppermost point of ${}^{R}P$
${}^{C}p_i$	ith point represented with respect to the camera frame
${}^{R}p_i$	ith point represented with respect to the reference frame
p_{ijk}	Centroid point of v_{ijk}
R	Rotation matrix *or* the reference frame
R_1	First elemental rotation
R_2	Second elemental rotation
R_3	Third elemental rotation
${}^{R}_{C}R$	Rotation from the camera frame to the reference frame

$r_{spine_calculation}$	Neighboring radius for spine calculation
s	Point representing the center of mass
S_B	Between-class scatter matrix
S_i	Image set of ith individual
S_t	Set of test images
$S_{training}$	Training set
S_W	Within-class scatter matrix
s_x	x coordinate of the center of mass
s_y	y coordinate of the center of mass
s_z	z coordinate of the center of mass
${}^{R}_{A}T$	Transformation which corrects the direction
${}^{R'}_{C}T$	Transformation which corrects the ground plane
t	Translation vector
t'	Translation vector
u	Translation vector
u_i	ith eigenvector of $A^T A$
V	Matrix whose columns are the eigenfaces
$\hat{v}$	Normal vector
v_i	ith eigenvector
v_{ijk}	Non-empty grid
W_i	Set of weight coefficients of ith image
w_{ij}	jth weight coefficient of ith image
W_{fld}	Projection of Fisher's linear discriminant
W_{opt}	Optimal projection
W_{pca}	Projection of PCA
$W_{training}$	Matrix whose columns are weight vectors
X	Name of the axis
X_k	Point coordinate
x_k	Image coordinate
Y	Name of the axis

Y'	Rotated Y axis
Y_k	Point coordinate
y_k	Image coordinate
Z	Name of the axis
Z'	Rotated Z axis
Z_0	Distance of the reference plane
Z_k	Distance of the object plane
β	Rotation angle
γ	Rotation angle
δ	Distance threshold
δ_m	Value of the pixel m
δ_x	Edge size of a grid *or* lens correction through X axis
δ_y	Edge size of a grid *or* lens correction through Y axis
δ_z	Edge size of a grid through Z axis
ϵ_n	Threshold
ϵ_d	Distance threshold
ϵ_w	Width threshold
θ	Rotation angle
θ'	Rotation angle
λ_i	ith eigenvalue
μ	Mean image
μ_j	Mean image of the jth class
$\sum_p$	Covariance matrix
τ	Threshold *or* image vector
τ_i	ith image vector
τ_i'	Difference vector of the ith image
τ_{ij}	jth image vector of the ith class
$\mathbb{Z}$	Integers

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
CSR	Class Success Rate
ICP	Iterative Closest Point
PCA	Principle Component Analysis
RANSAC	RANdom SAmple Consensus
SSR	Sample Success Rate

1. INTRODUCTION

Animal farms have been steadily growing to meet the consumption requirements of the society in an efficient manner. According to Turkish Statistical Institute data [4], the number of cattle, meat production and milk production have increased 33.9%, 131.4%, and 94.8% respectively in Turkey between 2001 and 2014. With new automated technologies, not only the efficiency of the farms is improving, but also the labor force needed per animal is diminishing. Due to automation and the growing size of the animal herds, the interaction between animals and caretakers is decreasing. This fact necessitates new monitoring and tracking systems to collect useful information about the herds in order to observe their general health and instantaneous state.

By tracking animals individually, valuable clues can be gathered on the health and welfare of each animal. However, recognizing and tracking an animal in a farm is a difficult task due to the target's similarity and hard to predict dynamics. Moreover, an indoor positioning system has requirements which can not be satisfied by the technologies used in outdoor positioning systems [5]. Environmental factors such as walls, ceilings, objects and obstacles cause time delays, high attenuation and signal scattering. Besides, most of the applications require high precision and accuracy in small areas [6].

1.1. Indoor Animal Positioning Systems

In [7], cow positioning and tracking systems are compared according to their accuracy, sampling frequency, number of tracking animals, battery life, environment resistance, size of tag and price. These systems can be categorized in two groups where the first group consists of indirect recognition techniques based on markers or external devices placed on the object/animal; and the second group consists of direct approaches based on biometric features of the animals. Indirect systems; such as global positioning

system (GPS) [8], radio frequency identification (RFID) [9], radio tracking [10], bluetooth [11], wireless local area network (WLAN) [12, 13], ultrasound [14], depend on external devices which send signals from the animal to the receivers. The distance of the signal emitter is estimated by using time of arrival, angle of arrival or received signal strength; and generally two techniques, triangulation and fingerprinting, are used to position the source [15].

Installation cost of the required hardware of above systems is relatively high which is an important limitation against the usability of such systems. Power consumption is another issue where the cost and weight increase when the capacity of the batteries is increased; moreover state of the batteries and devices should be controlled periodically which increases the maintenance cost of the system. Finally, the most important drawback is the negative psychological effects created by external devices which also affect the natural interactions and behavior of the animals and decrease the production [16].

1.2. Optical Positioning Systems

Alternative approaches to indirect object tracking are image based, sometimes also called optical solutions [17] which are vision based direct recognition and positioning techniques where visual features of the subjects are used for identification. Positioning is mostly done by converting relative positions of the objects with respect to the cameras, to a representation in a global reference [18]. Such solutions are cheaper and hence relatively more applicable. Moreover, its power consumption is lower than that of other systems and the location of the detected object is very precise. If the identity of the object is required, identification is done independently or simultaneously with tracking [19], where these processes can be carried out in three steps:

- detection of the object
- identification of the object

- tracking

Recognition and tracking of objects are widely investigated topics in the context of face/person recognition and pedestrian, vehicle tracking in computer vision; and relatively robust solutions exist in the literature [20, 21]. On the other hand, animal detection and identification are relatively less studied in computer vision. The main difficulty of these topics stems from the natural structure of the problem. More specifically, visual appearances of most of the animals are evolved to be camouflaged with respect to their environments. On the other hand, for some species, coat patterns and shape features can serve as unique features for identification of individuals in fair conditions. In [22], a prototype application is evaluated in a colony of African penguins and in a small scale zebra image collection. In studies on marine animals, naturally occurring distinctive shape features are used for identification. These approaches are based on patterns of nicks and notches of the dorsal fins [23] where populations of marine animals are analyzed by matching photos captured in different seasons. In [24,25], photos of bowhead whales and dolphins are selected which are in sharp focus, with good contrast and minimal perspective error; subsequently distinctiveness of the animals are evaluated.

In a related line of research, there are some studies on applying face recognition algorithms on animals. In [26], a system which acquires 3D images, is installed in a feeding station for horses. After background extraction, an ellipse-like head model is fitted and images are normalized to be used for identification. However the results of the system is arguable since captured images are not tested with an identification algorithm. In [27–29], face recognition methods are applied on dogs, great apes and sheep. For cattle identification, there is a preliminary trial [30] which uses single face images of twelve Japanese black cattle and in [31] a texture descriptor method is used to identify face images of thirty cattle. These studies are valuable to demonstrate the applicability of face recognition algorithms on animals; however capturing and normalizing face images in a natural and dynamic environment are still important

problems.

Most of the image processing algorithms, including the ones mentioned above, are affected by the external factors such as lighting and illumination changes. They need a precise view of the object for the identification step and most of the approaches are sensitive to the posture of the animal. For an applicable solution, environmental factors should be controllable; rather designed solution should be robust to such changes.

1.3. Proposed Optical Solution

The main assumption of this thesis is "in a cattle farm individual cows can be identified according to their 3D geometric features". When global geometric features of the cows (i.e., volume, area, center of mass, height, weight, width etc.) are analyzed, there is evidence for possible classification among animals, however they are not sufficient for identification of individual animals. On the other hand, most of the cow body has dynamic natures which make capture and process of such areas difficult. For example, general behavior of head and feet of an animal contains movements most of the time and shape of the lower part of the body changes with different daily factors such as feeding, pregnancy and physical interaction with other animals. In addition, shape of the stated parts is hard to fully observe with cameras put on stationary positions in a cattle farm.

On the other hand, the top of the animal contains smaller movements with respect to other parts. The shape of these areas does not change daily; besides contains characteristic shapes due to the skeletal structure (i.e., spine, backbones, tail tip), which is convenient for identification. To be more specific, it is observed that the back part of the upper surface contains most of the discriminative shape patterns which do not change significantly even when the animal is walking. Therefore, the identification process proposed in this thesis is constructed based on shape analysis of upper back part of the cow body; which can be captured smoothly with 3D cameras placed at an



Figure 1.1. Kinect Sensor.

adequate height.

1.3.1. Cameras

In recent years with the aid of the three dimensional depth cameras, such as Microsoft's Kinect, Asus's XTION, which are commercially available with relatively cheap prices, different approaches have become applicable in real time computer vision. In this thesis, Kinect cameras (see Figure 1.1) are used for data acquisition. With these cameras in addition to the color information of the scene, a depth map is also available in 30 Hz. Hence the scene can be represented with points which have x,y,z values in the Cartesian space and the related RGB color.

For the proposed solution, cameras should be placed on suitable places of the cattle farm, looking downwards to capture the top of the animals while passing through or staying in the field of the view. Since the optimal range of the sensor is between 1 to 3 meters, and average cow height is around 1.5 meters, appropriate camera height is between 2.5 - 4.5 meters above the ground level. An example camera configuration and point cloud of a cow are shown in Figure 1.2 and Figure 1.3, respectively. Height and number of cameras should be optimized with respect to the structure of the facilities and coverage area.

1.3.2. Identification Process

As shown in Figure 1.4, the proposed identification process has four main steps:

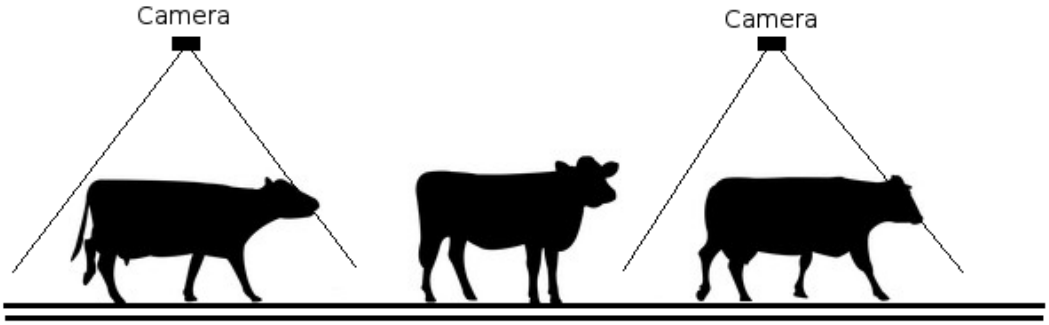


Figure 1.2. Sample configuration of the cameras.

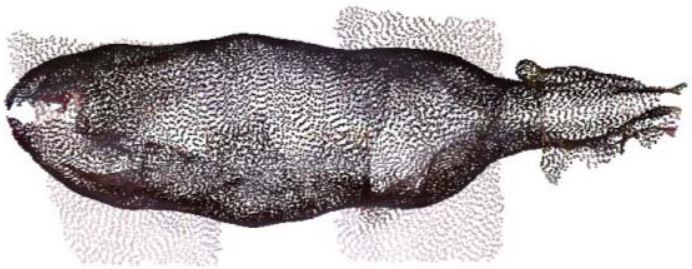


Figure 1.3. Sample point cloud of a cow [1].

- In the implementation, *detection of the animal* is simply done by analyzing the change of the centroid of points representing the scene in sequential frames. Identification process starts when the change is greater than a certain distance and the area of the entrant is more than a certain threshold.
- In the *data filtering* processes, point cloud of the scene is down-sampled, and points corresponding to the cow are extracted by using their three dimensional spatial relations.
- In the *pose normalization* step, a rough normalization is done by estimating the direction of the animal by principle component analysis (PCA) and by using the centroid of the points. Subsequently, the proposed pose normalization algorithm is applied where points corresponding to the spine of the animal are detected and a line model is fitted onto them.

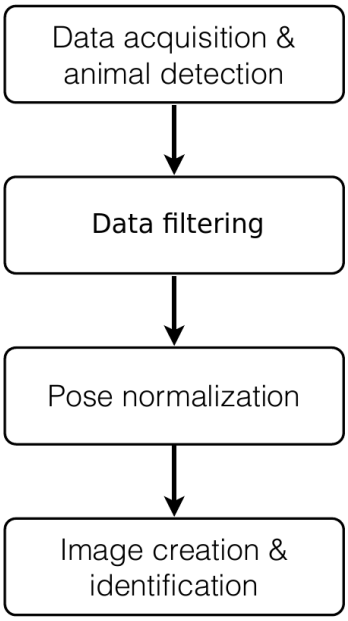


Figure 1.4. The proposed 3D identification scheme.

- In the *image creation* process, the top back part of the animal is extracted; and 2D gray-scale images are constructed by using the local geometric surface features of the related points. Then, *face recognition* algorithms are applied to these images to identify the related animal.

To test the process in real life conditions, the proposed solution is implemented in a cattle farm which has 50 cows. A dataset is acquired with two cameras installed in two different places of a corridor, where animals pass through to enter the milking area. Algorithms are implemented as a complete real time application by using C++ programming language. The first video is captured in a lighted environment and samples are automatically extracted and used for training the algorithm. The second video is captured in a dark environment where samples are used for testing the automatic identification process and successful identification results are obtained which are presented in Chapter 5.

1.4. Contributions of the Thesis

- A novel cow identification system is proposed which is based on 3D shape analysis of top back part of the animals. There are prominent features of this solution which differentiates it from the others in the literature, i.e., it
  - (i) does not need any markers or external devices placed on the animal
  - (ii) works in even unlighted environments
  - (iii) identifies even black cows without distinctive coat patterns
  - (iv) is relatively cheaper
  - (v) enables accurate positioning
- A 3D cattle dataset is acquired, which is unique in the literature. This dataset consists of three dimensional point cloud sequences of 50 different cows, captured in lighted and unlighted conditions.
- Applicability of the proposed solution is verified by testing with acquired dataset. In particular,
  - (i) dataset is gathered from moving animals and contains relatively few number of samples per animal,
  - (ii) training and testing videos are captured with different cameras in different lighting conditions (lighted and dark),
  - (iii) most of the cows are black and without distinctive coat patterns.
- A novel pose normalization method is proposed which is based on recognition of the spine.
- For identification, two popular face recognition methods, namely Eigenface and Fisherface methods, are used. Therefore, presented identification results serve a base point for future research on this topic.

1.5. Organization of the Thesis

The organization of this thesis is as follows. Chapter 2 provides brief descriptions of applied methods and algorithms with proper details and related references. Chapter

3 introduces steps of the proposed solution, such as data filtration, pose normalization and image creation, in detail with mathematical and conceptual expressions and explanatory illustrations. In Chapters 4 and 5, implementation of the solution and dataset acquisition are explained; and test results and discussions are presented, respectively. Finally in Chapter 6, concluding remarks are stated.

2. METHODS AND ALGORITHMS

In this chapter, applied methods and algorithms are introduced. First, depth measurement method of Kinect and point cloud processing algorithms; such as voxel grid filtering, euclidean clustering, RANSAC, and ICP are explained. Then, face recognition methods are introduced, which are used in the identification process. These methods and algorithms are implemented mostly by the aid of two important libraries, namely PCL (Pint Cloud Library) [32] and OpenCv [33], which contain many state of the art algorithms on point cloud and image processing as open source.

2.1. Depth Measurement of Kinect

The sensor used in this thesis, has an infrared laser emitter, an infrared camera, and an RGB camera. The laser subunit emits a constant pattern to the scene whose reflection is captured via the infrared camera. The captured pattern is correlated with a precaptured reference pattern of a plane; and depth of pixels are estimated with a process based on triangulations [34] as shown in Figure 2.1. The position of a speckle in the infrared image is shifted according to the reference plane. With a simple image correlation process, shifts are measured for all the speckles and a disparity image is obtained.

In Figure 2.1, Z_0 is the known distance of the reference plane; Z_k is the distance of the object plane to the sensor; b is the distance between the infrared camera and the laser emitter; f is the focal length and d is the measured disparity. From the similarity of the triangles, $\triangle op_1k \sim \triangle oCL$ and $\triangle p_1Ck \sim \triangle p_3Cp_2$, we have

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \tag{2.1}$$

$$\frac{d}{f} = \frac{D}{Z_k}. \tag{2.2}$$

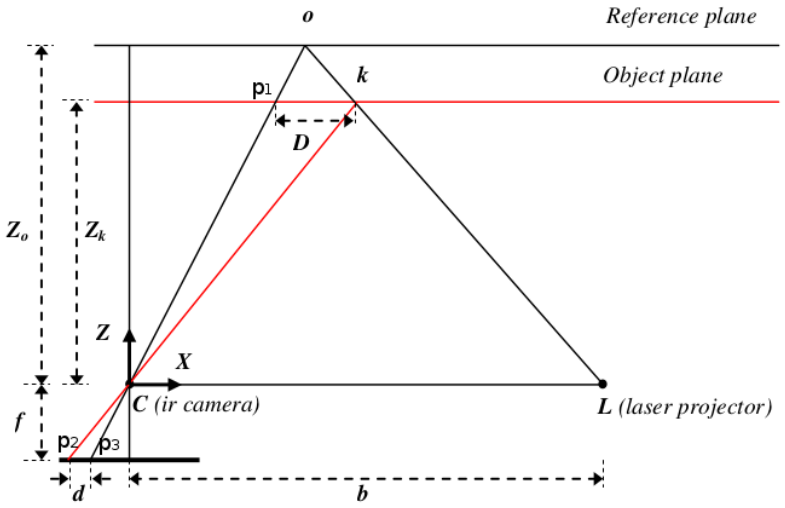


Figure 2.1. Triangulation process [2].

By combining (2.1) and (2.2), the distance of the object plane, Z_k, can be expressed as

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{fb}d} \tag{2.3}$$

where, Z_0, f and b are known a priori and d is measured.

The coordinates of the point, X_k and Y_k, are obtained by multiplying the imaging scale and the image coordinates of corresponding pixel as

$$X_k = -\frac{Z_k}{f}(x_k - x_0 + \delta x) \tag{2.4}$$

$$Y_k = -\frac{Z_k}{f}(y_k - y_0 + \delta y) \tag{2.5}$$

where x_k and y_k are the image coordinates of the point; x_0 and y_0 are the coordinates of the principal point; and δx and δy are the corrections for lens distortion.

In [2], accuracy and resolution of Kinect's depth data are analyzed. The depth resolution decreases quadratically and random error of the depth measurements in-

creases quadratically with the increasing distance from the sensor. At the maximum range, this error is as high as 4 cm and in general, the sensor should be used within the range 1 to 3 meters.

2.2. Voxel Grid Filtering

In this algorithm, three dimensional Euclidean space is divided into grids and points in the same grid are represented by a single point whose coordinates are the average of the associated points. Let

$$\begin{aligned} v_{ijk} = \{\, p(x,y,z) \in P \mid \|x - i\delta_x\| < (\delta_x/2) \wedge \\ \|y - j\delta_y\| < (\delta_y/2) \wedge \\ \|z - k\delta_z\| < (\delta_z/2) \;\} \end{aligned} \tag{2.6}$$

be a nonempty grid, where δ_x, δ_y, and δ_z are edge sizes of the grid volume. Each nonempty grid, v_{ijk}, is represented with a point, p_{ijk}, which is the centroid of grid points:

$$v_{ijk} \sim p_{ijk}\,(x,y,z)\,,\; x = \frac{\sum\limits_{m=1,.,n} x_m}{n},\; y = \frac{\sum\limits_{m=1,.,n} y_m}{n},\; z = \frac{\sum\limits_{m=1,.,n} z_m}{n} \tag{2.7}$$

Finally, down-sampled point cloud consists of centroid points of each grid:

$$f_{voxel}(P) = \{p_{ijk} \sim v_{ijk} \mid v_{ijk} \neq \varnothing\} \tag{2.8}$$

2.3. Euclidean Clustering

With Euclidean clustering algorithm, points are separated into groups according to their relative Euclidean distances. If the distance of any two point is smaller than a threshold, these two points are put into the same cluster, C_k. Therefore, for any point

in a cluster, there is at least one neighbor point near than a threshold distance:

$$C_k = \{\, p \in P \mid \exists p_k \in C_k \;\; \|p - p_k\| < \epsilon_c \} \tag{2.9}$$

With this algorithm, big point clouds can be divided into meaningful clusters which may belong to individual objects.

2.4. RANSAC (RAndom SAmple Concensus) Algorithm

RANSAC algorithm [35] is a general parameter estimation method, which fits a model to a given dataset by assuming the data have inliers and outliers. The term "inlier" refers to a sample which suits the underlying model, and "outlier" refers to an erronious sample. This algorithm is widely used for extracting primitive shapes from noisy 3D data [36], as well as finding planes [37]. With RANSAC, candidate solutions are generated by using minimum number of random samples to estimate the underlying model parameters in an iterative manner. To extract a plane from a point cloud, P, the algorithm mainly consists of the steps stated below:

(i) Randomly select minimum number of points to generate the model. Therefore, three points are adequate to define a plane. Let these not-all colinear points be

$$p_i, p_j, p_k \in P; \;\; i \neq j, \; j \neq k, \; i \neq k. \tag{2.10}$$

(ii) Fit the model to these points:

$$ax + by + cz + d = 0 \tag{2.11}$$

(iii) Evaluate all other points with respect to the candidate model. If the distance of a point is smaller than a threshold, it is considered as an inlier, otherwise an

outlier:

$$p_i \in \begin{cases} P_{inlier}, & \text{if } \|ax_i + by_i + cz_i + d\| < \delta, \\ P_{outlier}, & \text{otherwise.} \end{cases} \tag{2.12}$$

(iv) If the fraction of number of inliers and outliers is higher than a threshold, i.e.,

$$\frac{n_{inliers}}{n_{outliers}} > \tau, \tag{2.13}$$

then continue; otherwise return to the first step and reiterate. If maximum iteration number is achieved, terminate without a model.

(v) Re-estimate parameters with inliers and refine the constructed model. Terminate with the model.

In Figure 2.2, an illustration is shown where a line is fitted to 2D data. Inliers are represented with blue dots and outliers are represented with red dots. The advantage of RANSAC algorithm is that a model can be fitted accurately even when there are relatively many outliers; on the other hand, if the iteration number is limited, the solution may not be optimal and thresholds should be adjusted according to the problem and the given data.

2.5. ICP (Iterative Closest Point) Algorithm

ICP algorithm [38] is mainly used for registration of surfaces. In this algorithm, the reference point cloud is kept stationary and the second point cloud is transformed iteratively to match the reference. At the beginning of the ICP algorithm, it is important to start with a good initial rotation and translation since the algorithm may converge to a local minimum. At the end of the algorithm, according to the final value of the surface to surface distance, two surfaces are said to be congruent if this distance is smaller than a predefined threshold. Main steps of this algorithm are as follows:

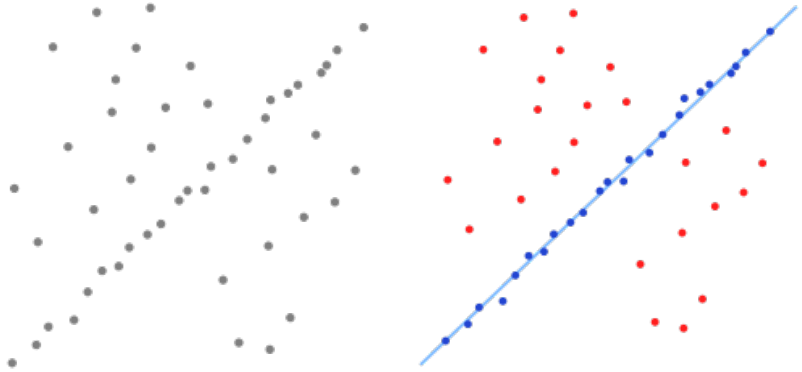


Figure 2.2. Example dataset and fitted line model. Blue dots are inliers, red dots are outliers.

(i) Associate points by the nearest neighbor criteria. For each point in the source point cloud, $p_i \in P_2$, find the closest point in the reference point cloud, $p'_i \in P_1$; where the minimum distance does not exeed a predefined threshold, i.e.,

$$p_i \Leftrightarrow p'_i, \;\; p'_i = \arg\min_{p \in P_1} \|p - p_i\|, \;\; \|p_i - p'_i\| < \delta_n \tag{2.14}$$

(ii) Estimate transformation parameters (rotation and translation) which minimize the following mean square cost function:

$$E(R,t) = \sum_i \|(Rp_i + t) - p'_i\|^2 \tag{2.15}$$

$$[R,t] = \arg\min_{R,t} E(R,t) \tag{2.16}$$

To find the best transformation, there are closed form solutions based on singular value decomposition, quaternions, orthonormal matrices and dual quaternions [39].

(iii) Transform the source point cloud.

(iv) Calculate the cost function.

(v) Iterate until the difference between two consecutive transformations is smaller

than a pre-defined threshold.

2.6. Eigenface Method

Eigenface method [40] is a widely used face recognition approach in the literature [41], which depends on principle component analysis (PCA), and is a data reduction technique where data points are projected onto a set of vectors. The most representative vectors are the eigenvectors of the covariance matrix with the highest eigenvalues, and projected data points preserve most of the information about the original data.

In eigenface method, each image, M_i, in the training set is converted to a vector, τ_i, by concatenating rows of pixels in it:

$$M_i \sim \tau_i \tag{2.17}$$

Subsequently, eigenfaces are computed and original images are represented with weighted sum of these vectors, sometimes also called eigenpictures. Here, the term "eigenfaces" refers to the set of n eigenvectors of the sample covariance matrix with highest eigenvalues. For a training set with N images, the sample covariance matrix is

$$C = \sum_i (\tau_i - \mu)(\tau_i - \mu)^T, \tag{2.18}$$

where $\mu = \sum_i \tau_i / N$ is the mean image. If v_i is a nonzero vector and λ_i is a number such that

$$Cv_i = \lambda_i v_i, \tag{2.19}$$

v_i said to be an eigenvector of C with eigenvalue λ_i. Let the ith image of the training

set differ from the mean image by a vector

$$\tau_i' = \tau_i - \mu, \tag{2.20}$$

and A be the set of these vectors:

$$A = \begin{bmatrix} \tau_1' & \dots & \tau_N' \end{bmatrix} \tag{2.21}$$

The covariance matrix, C, can then be represented as

$$C = AA^T, \tag{2.22}$$

where AA^T is a $d \times d$ matrix, and d is the length of an image vector which is relatively a big number, $d > N$. Computing the eigenvectors of C is often computationally infeasible. However, the rank of the covariance matrix is limited with the number of samples and there can be at most $N - 1$ eigenvectors with nonzero eigenvalues. Therefore, let u_i be an eigenvector of $A^T A$ which is a $N \times N$ matrix, i.e.,

$$A^T A u_i = \lambda_i u_i. \tag{2.23}$$

Multiplying each side by A, we obtain

$$AA^T(Au_i) = \lambda_i(Au_i). \tag{2.24}$$

Therefore, eigenvectors of AA^T corresponding to its non-zero eigenvalues can be computed simply by first finding eigenvectors of $A^T A$ and then multiplying it by A, i.e.,

$$v_i = Au_i. \tag{2.25}$$

This latter approach is computationally less expensive. After computing the eigenvec-

tors, they are normalized and every face image in the training set, M_i, is represented by means of weighted sum of these eigenfaces and the mean image

$$M_i = \mu + VW_i, \tag{2.26}$$

where V is a matrix whose columns are the eigenfaces and W_i is a set of weights:

$$W_i = \begin{bmatrix} w_{i1} & \dots & w_{in} \end{bmatrix}^T \tag{2.27}$$

The weight coefficient of the jth eigenface, w_{ij}, can be found by projecting the difference image onto it:

$$w_{ij} = v_j^T(\tau_i - \mu) \tag{2.28}$$

In the identification process, the weight vector of the new image is constructed with the above approach and compared with the ones in the training set. For every image in the training set, distance of the new image, M_i, is found by

$$d(M_i, M_j) = \|W_i - W_j\|, \tag{2.29}$$

and the new image is recognized with the ID of the training image with the smallest distance.

With this data reduction approach, relatively big number of images are represented with weight vectors in the face space spanned by the eigenfaces. Therefore, the memory requirement is diminished, and the speed of the system is increased significantly which enable recognition in real time. On the other hand, it is very sensitive to illumination changes, scale and translation; which makes preprocessing of images, such as pose normalization, crucial for successful identification.

2.7. Fisherface Method

In the eigenface method, we look for the projection which maximizes the total scatter across all images. This approach retains variations due to illumination changes and facial expressions, and discrimination may be prevented. On the other hand, Fisherface method is a derivative of Fisher's linear discriminant [42] and chooses projection directions which maximizes the ratio of between-class scatter to that of within-class scatter; therefore, it discounts regions with large variations [43].

Suppose that we have

$$N = n_1 + \cdots + n_C \tag{2.30}$$

samples captured from C classes and τ_{ij} corresponds to the jth image of the ith class. The within-class scatter, S_w, and between-class scatter, S_b, matrices are calculated as

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{n_j} (\tau_{ij} - \mu_i) (\tau_{ij} - \mu_i)^T \tag{2.31}$$

$$S_b = \sum_{i=1}^{C} n_i (\mu_i - \mu) (\mu_i - \mu)^T , \tag{2.32}$$

where

$$\mu = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{n_i} \tau_{ij} \tag{2.33}$$

is the total mean, and

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \tau_{ij} \tag{2.34}$$

is the mean of ith class. If S_w is not singular, an optimal projection, W_{opt}, is chosen

which maximizes the ratio of determinants as:

$$W_{opt} = \begin{bmatrix} w_1 & \dots & w_m \end{bmatrix} = \arg\max_W \frac{|W^T S_b W|}{|W^T S_w W|} \tag{2.35}$$

In the above expression, $\begin{bmatrix} w_1 & \dots & w_m \end{bmatrix}$ is the set of generalized eigenvectors of $S_w^{-1} S_b$

$$S_w^{-1} S_b w_i = \lambda_i w_i \tag{2.36}$$

where the upper bound on m is $C-1$. In [43], the image set is projected to a lower $(N-C)$ dimensional space to overcome the problem of singular S_w, where the resulting within-class scatter matrix is nonsingular. This reduction is done by applying principal component analysis (PCA), subsequently standard Fisher's linear discriminant is applied and samples are projected to $C-1$ dimensional space

$$W_{opt} = W_{fld} W_{pca}, \tag{2.37}$$

where

$$W_{pca} = \arg\max_W \left| W^T S_t W \right|, \tag{2.38}$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T S_b W W_{pca} \right|}{\left| W^T W_{pca}^T S_w W W_{pca} \right|}. \tag{2.39}$$

Identification is achieved with the same approach in the previous section by comparing the weight vectors of the training images and the test image. More information on the implementation of both eigenface and fisherface methods can be found in [44].

2.8. Summary of the Chapter

In this chapter, the underlying theory of depth measurement of the Kinect sensor is presented. Subsequently, point cloud processing algorithms such as voxel grid filtering, Euclidean clustering, RANSAC and ICP are explained; which are applied for different purposes throughout this thesis:

- *Voxel grid filtering:* down-sampling the raw data (in Section 3.2.1).
- *Euclidean clustering:* extraction of points corresponding to the animal to be identified (in Section 3.2.4).
- *RANSAC:* ground plane and spine line detection (in Sections 3.3.3 and 3.2.2).
- *ICP:* alignment of captured frames from different viewpoints (in Section 4.1.2).

Finally, two face recognition algorithms, Eigenface and Fisherface, are presented which are used in Section 3.5 for identification of animals.

3. 3D CATTLE IDENTIFICATION

In this chapter, steps of the proposed 3D cattle identification solution are discussed in detail, where an overview can be found in Section 1.3. In the first section, acquisition of the raw data and detection of the animal; and in the second section, the filtering process of these raw data are explained. Subsequently, the proposed pose normalization method is presented in Section 3.3 and captured samples are brought into canonical form for the comparison step. In the last section, three types of images are introduced which are constructed with respect to the local features of the normalized 3D data corresponding to the detected animal. Finally, identification process is explained, where face recognition methods are applied on the constructed images.

3.1. Data Acquisition and Animal Detection

The sensor gives a point cloud as an output, 30 times in a second. This point cloud consists of 480×640 points with related x, y, z Cartesian coordinates and RGB color information (see Section 2.1). The sensor frame is originated on the infrared camera with the X,Y,Z axes pointing front, right and bottom of the camera, respectively.

Detection of the animal is simply done by analyzing the change of the centroid of point cloud representing the scene in sequential frames. Filtering process starts when the change is greater than a certain distance. If the area of the entrant is greater than a certain threshold, it is considered as an animal and the identification process starts.

3.2. Data Filtering

After the acquisition, the raw data should be processed to extract meaningful parts related to the animal to be identified. As to be discussed in following subsections, firstly the point cloud is down-sampled and the view of the camera is corrected with

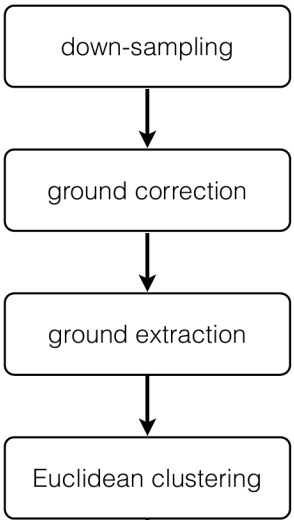


Figure 3.1. Steps of data filtering process.

respect to the ground. Subsequently, points related to the ground are filtered out and the point cloud corresponding to the animal is extracted.

3.2.1. Down-sampling with Voxel Grid Filter

Redundantly dense point clouds require excessive processing time in the algorithms. Moreover, some parts of the animal, which are visible at more than one camera, will have greater point densities than the ones visible at just one camera. To equalize the density and down-sample the data, voxel grid filtering is applied. In this down-sampling algorithm, 3D Euclidean space is divided into grids and points which are in the same grid are represented with their centroid points (see Section 2.2). By selecting a proper grid size, a smoother surface is acquired without loosing the local discriminative features.

3.2.2. Ground Plane Correction

In Figure 3.2, a scene captured from the top of a cow is shown. On the left side of the figure, the scene is represented according to the camera frame, C, and on the right side it is represented according to the reference frame, R. The reference frame is



Figure 3.2. Raw point cloud, ${}^{C}P$, captured from the top of a cow. Left - view from the camera frame, Right - view from the reference frame.

located on the ground plane and is aligned with the center of mass of the animal. The Z axis points upwards whereas the X axis points in the same direction with the animal (see Figure 3.3). As shown on the right side of Figure 3.2, the view of the animal is erroneous, since the transformation of the camera frame according to the reference frame is not known and the related point cloud is not transformed accordingly. Let

$$ {}^{C}P = [{}^{C}p_1, {}^{C}p_2, ..., {}^{C}p_n] \tag{3.1} $$

represent a point cloud of n points with respect to the camera frame, C, where each point $p_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ is three dimensional, and is to be represented with respect to the reference frame, R, as

$$ {}^{R}P = [{}^{R}p_1, {}^{R}p_2, ..., {}^{R}p_n]. \tag{3.2} $$

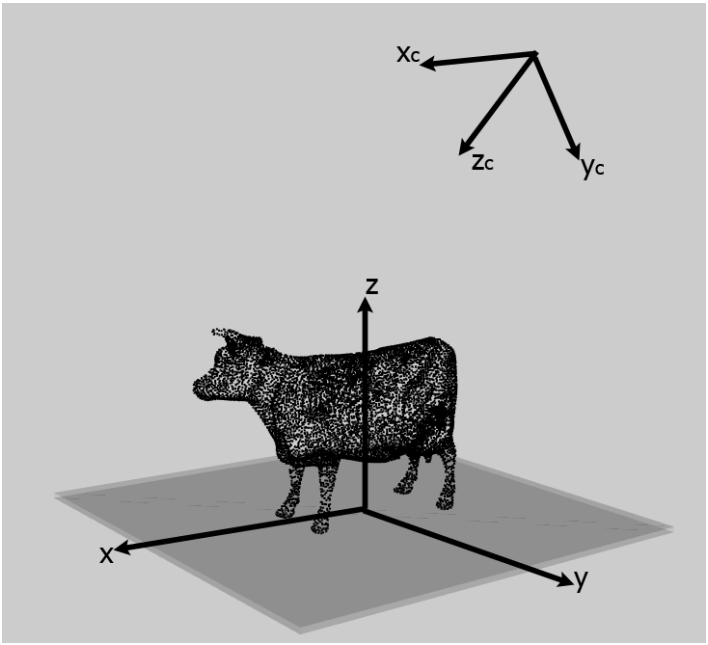


Figure 3.3. Reference frame is located on the ground and is aligned with the center of the animal.

To this end, all points should be rotated with an appropriate C to R rotation matrix, ${}^{R}_{C}R$, and translated with a vector as follows:

$$ {}^{R}P = {}^{R}_{C}R \ {}^{C}P + t \tag{3.3} $$

According to Euler's rotation theorem [45], any rotation can be decomposed as a product of three sequential elemental rotations. Therefore, rotation from camera frame to reference frame, ${}^{R}_{C}R$, can be computed as

$$ {}^{R}_{C}R = R_3 \ R_2 \ R_1. \tag{3.4} $$

R_1, R_2 and R_3 are rotations around Z, Y' and Z' axes, where Y' and Z' are rotated Y and Z axes, respectively.

R_1 and R_2 are determined by using the geometrical relations between the detected

and expected planes representing the ground. In Figure 3.3, the reference and camera frames are shown, respectively. The expected ground plane overlaps the XY plane of the reference frame with a normal vector parallel to the Z axis. Note that the orientation of the detected plane differs due to the rotation of the camera.

The ground in the scene is found by fitting a plane model to the proper subsection of the point cloud, by using the RANSAC (RAndom Sample Consensus) algorithm [35] (see Section 2.4). Let the detected ground plane be described by

$$ax + by + cz + d = 0 \tag{3.5}$$

where a,b,c,d are determined such that distance of any point, p_g, representing the ground is smaller than a certain threshold:

$$d_{plane}(p_i) = ax_i + by_i + cz_i + d \tag{3.6}$$

$$|d_{plane}(p_g)| < \epsilon \tag{3.7}$$

In Figure 3.4, the normal vector $\vec{v} = \begin{bmatrix} a & b & c \end{bmatrix}^T$ of the detected ground is shown. Our aim is to make this vector overlap with the Z axis by applying proper rotations. The first rotation is $-\theta$ degrees around the Z axis, where θ is the angle between the X axis and the projection of $\vec{v}$ onto the XY plane. This angle can be computed as

$$\theta = \arctan\left(\frac{b}{a}\right) \tag{3.8}$$

resulting in the rotation matrix

$$R_1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.9}$$

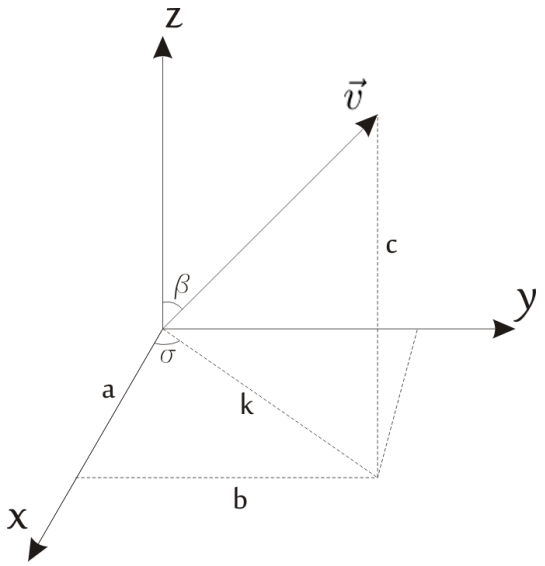


Figure 3.4. Normal vector ($\vec{v}$) of the detected ground plane and related angles for the rotations (θ, β).

The second rotation is $-\beta$ degrees around the Y axis, where β is the angle between $\vec{v}$ and the Z axis

$$\beta = \arctan\left(\frac{\sqrt{a^2 + b^2}}{c}\right). \tag{3.10}$$

The corresponding rotation matrix is obtained as

$$R_2 = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}. \tag{3.11}$$

After applying these rotations, the detected ground plane becomes parallel to the XY plane. To locate the origin on the ground, points should be moved d units along

the Z axis since the distance of the origin to the ground plane is given by

$$d_{plane}(p(0,0,0)) = a0 + b0 + c0 + d = d. \tag{3.12}$$

Let us define a 4×4 transformation matrix, ${}^{R'}_{C}T$, representing above rotations and the translation:

$${}^{R'}_{C}T = \begin{bmatrix} & & 0 \\ & R_2R_1 & 0 \\ & & d \\ 0 & & 1 \end{bmatrix} \tag{3.13}$$

By applying this transformation to the initial point cloud, the new form of the point cloud is acquired where the ground of the scene is overlapped with the XY plane. For a single point in the initial point cloud, this transformation is applied as follows:

$${}^{C}p_i \in {}^{C}P \tag{3.14}$$

$${}^{C}p_i = \begin{bmatrix} {}^{C}x_i & {}^{C}y_i & {}^{C}z_i \end{bmatrix}^T \tag{3.15}$$

$$\begin{bmatrix} {}^{R'}x_i \\ {}^{R'}y_i \\ {}^{R'}z_i \\ 1 \end{bmatrix} = \begin{bmatrix} & & 0 \\ & R_2R_1 & 0 \\ & & d \\ 0 & & 1 \end{bmatrix} \begin{bmatrix} {}^{C}x_i \\ {}^{C}y_i \\ {}^{C}z_i \\ 1 \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} {}^{R'}P \\ 1 \end{bmatrix} = {}^{R'}_{C}T \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix} \tag{3.17}$$

Figure 3.5 depicts the point cloud in Figure 3.2 that is transformed via (3.17). After the correction, the ground is overlapped with the virtual expected ground. In the application, cameras are in stationary positions; therefore it is not necessary to calculate the transformation matrix in (3.17) for all captured samples, as it can be

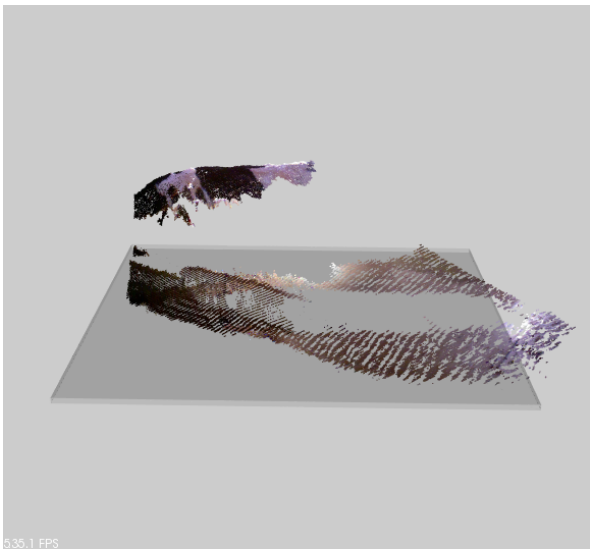


Figure 3.5. Point cloud after the ground plane is corrected (${}^{R'}P$).

used for all frames after it is calculated once initially.

To construct the final point cloud, ${}^{R}P$, the third rotation in (3.4) should be applied which corrects the rotation of the animal; and points should be translated to make the center of the animal aligned with the origin. Since the direction and the position of the animal are not stationary over time, this transformation should be calculated for every captured sample to bring the points of the animal to the canonical form. The procedure for this transformation is explained under rough pose normalization heading in Section 3.3.1.

3.2.3. Ground Plane Extraction

As shown in Figure 3.6, the points belonging to the ground are extracted with a pass through filter which extracts points having distance less than 5 cm to the ground plane.

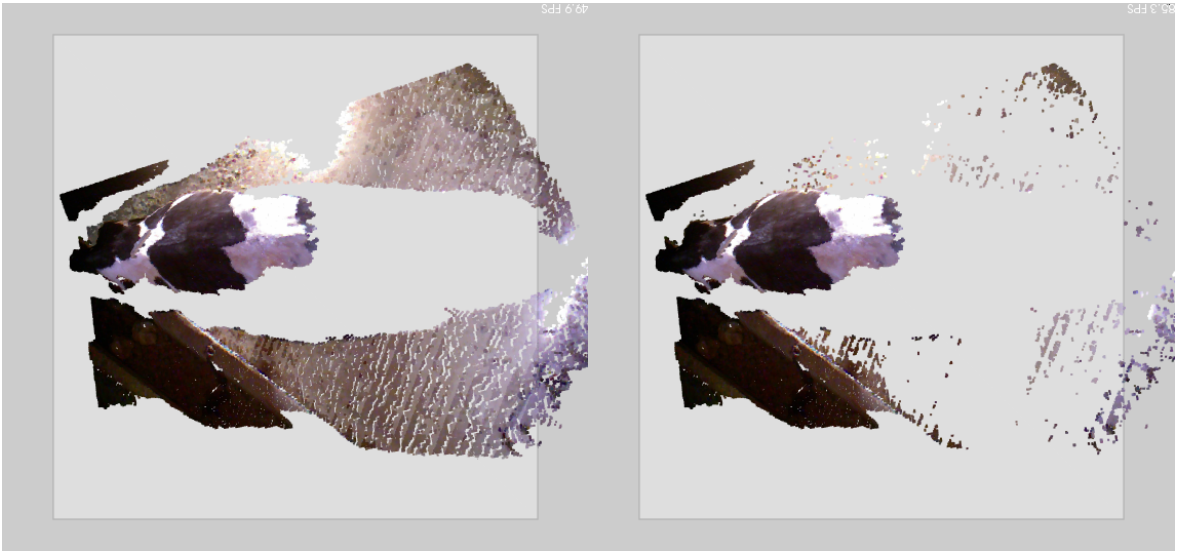


Figure 3.6. Left - Before extracting the ground, Right - After extracting the ground.

3.2.4. Euclidean Clustering

After ground plane extraction, we theoretically expect to have points belonging to the animal of interest. However, some other objects may be seen (e.g., part of an another animal, worker, container, etc.) or some erroneous points may exist in the scene. Therefore, points corresponding to the animal should be extracted. For this purpose, point cloud is clustered by using the Euclidean clustering algorithm, which clusters points according to their Euclidean distances (see Section 2.3). The biggest cluster in the scene is chosen and other clusters are filtered out and the point cloud, ${}^{A}P$, belonging to the animal is acquired as shown in Figure 3.7.

3.3. Pose Normalization

For the comparison step, samples of different animals should be brought into a canonical form. For that purpose, firstly a rough pose normalization is realized by aligning center of mass of each animal with the Z axis and rotating the point cloud accordingly to make the direction of the animal coincide with the X axis. Subsequently,

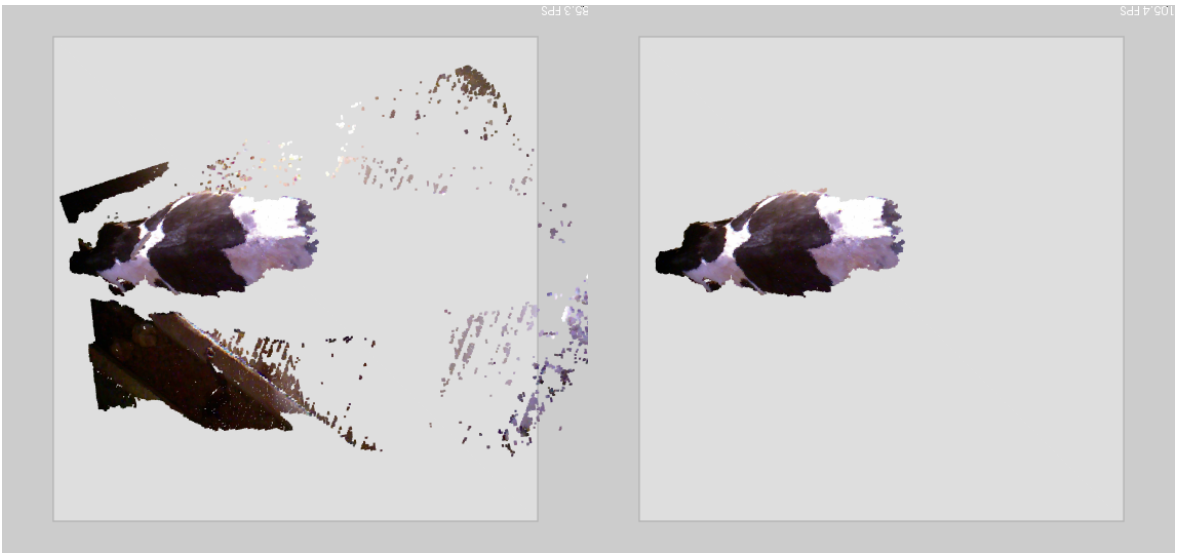


Figure 3.7. Left - Before clustering, Right - The biggest cluster, ${}^{A}P$.

a finer pose normalization is achieved by recognizing the spine of the animal and aligning it with the X axis.

3.3.1. Rough Pose Normalization

Let the point cloud of the animal, ${}^{A}P$, consist of k points:

$$ {}^{A}P = \{{}^{A}p_1, {}^{A}p_2, ..., {}^{A}p_k\} \tag{3.18} $$

$$ {}^{A}p_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{3.19} $$

and the center of mass is represented by a point $s = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}$. Coordinates of the center of mass are found by averaging the coordinates:

$$ s_x = \frac{\sum\limits_{i=1,.,k} x_i}{k}, \quad s_y = \frac{\sum\limits_{i=1,.,k} y_i}{k}, \quad s_z = \frac{\sum\limits_{i=1,.,k} z_i}{k} \tag{3.20} $$

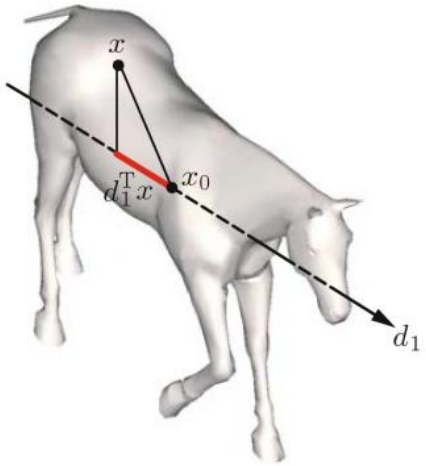


Figure 3.8. Principle direction of a point cloud of a horse [3].

To make the center of mass aligned with the Z axis, each point is translated with vector, t', as:

$$t' = \begin{bmatrix} -s_x \\ -s_y \\ 0 \end{bmatrix} \tag{3.21}$$

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + t' \tag{3.22}$$

By using the principle component analysis (PCA), the heading of the animal is determined which is the first principle direction, d_1, that maximizes the variance of the projection of the points onto it (see Figure 3.8),

$$d_1 = \begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T = \arg\max_d \left(d^T \, \Sigma_P \, d \right) \tag{3.23}$$

where Σ_P is the covariance matrix of the points. The angle between the principle



Figure 3.9. Left - Before correction of center of mass and direction, Right - Center of mass is aligned with the origin, and the direction is parallel with the X axis.

direction (d_1) and the X axis is:

$$\gamma = \arctan\left(\frac{y_d}{x_d}\right) \tag{3.24}$$

By rotating the point cloud $-\gamma$ degrees around the Z axis, the direction of the animal becomes aligned with the X axis. The related matrix of this rotation is:

$$R_3 = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

This translation and rotation can be represented with a 4×4 transformation matrix as:

$$^{R}_{A}T = \begin{bmatrix} R_3 & R_3 t' \\ 0 & 1 \end{bmatrix} \tag{3.26}$$

As shown in Figure 3.9, the canonical form of the sample is acquired as follows:

$$^{R}P = {}^{R}_{A}T \;\; {}^{A}P \tag{3.27}$$

3.3.2. Extracting the Region of Interest

After rough pose normalization, the top back part of the animal is extracted by thresholding points according to their x and z values. Let

$$f_{slice}\left(P, x\right) = \{p_i \in P \mid \|x_i - \epsilon\| \leq x\} \tag{3.28}$$

be the slice of the point cloud, P; and

$$f_{width}\left(P, x\right) = \max_{p_i, p_j \in f_{slice}(P,x)} \|y_i - y_j\| \tag{3.29}$$

be the width of this slice. p_t represents the uppermost point of ^{R}P which has the maximum z value, i.e.,

$$p_t = \arg\max_{p_i \in {}^{R}P} \left(z_i\right), \tag{3.30}$$

and p_b represents the rearmost point where the width of the point cloud is equal or higher than a certain threshold, ϵ_w:

$$p_b = \arg\min_{p_i \in {}^{R}P} \left(x_i\right), \;\; f_{width}\left(P, x_i\right) \leq \epsilon_w \tag{3.31}$$

Points are extracted as:

$$^{S}P = \{\, p_i \in {}^{R}P \mid x_b \leq x_i \leq x_b + l \;\wedge\; z_i \geq z_t - h \,\} \tag{3.32}$$

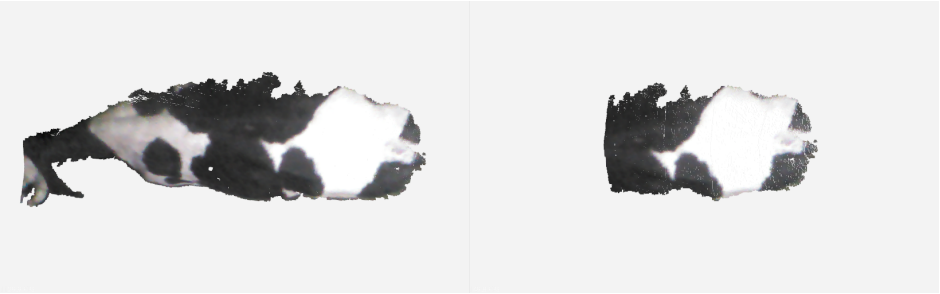


Figure 3.10. Left - Point cloud of the animal (${}^{R}P$), Right - Region of interest (${}^{S}P$).

where l and h are the length and height thresholds, respectively; and ${}^{S}P$ is the point cloud corresponding to the top back part of the animal. In Figure 3.10 an example filtering is depicted.

3.3.3. Pose Normalization by Spine Recognition

Bringing samples into canonical form is a crucial step for successful identification. As discussed in Section 3.3.1, point clouds are roughly transformed according to their center of mass and principle direction. In this section, direction and position of the point cloud are refined by recognizing the spine, which gives us better estimation of the direction and center of the sample.

The steps of the proposed spine recognition and correction procedure, which is illustrated in Figure 3.12, are depicted in Figure 3.11. In the first step, surface normals are calculated at each point by considering neighboring points in a radius of $r_{spine_calculation}$. Let ${}^{N}P_i$ be the set of points nearer than $r_{spine_calculation}$ to the point p_i:

$$ {}^{N}P_i = \{p \in {}^{S}P \mid \|p - p_i\| < r_{spine_calculation}\} \tag{3.33} $$

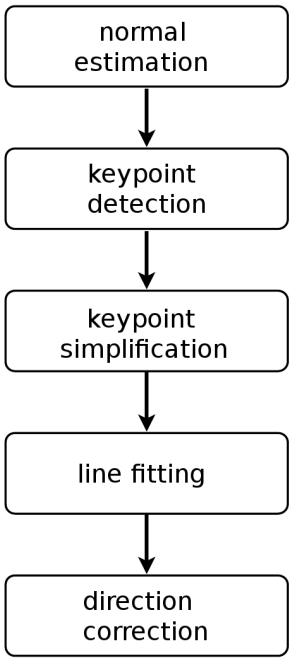


Figure 3.11. Steps of the proposed spine recognition and correction procedure.

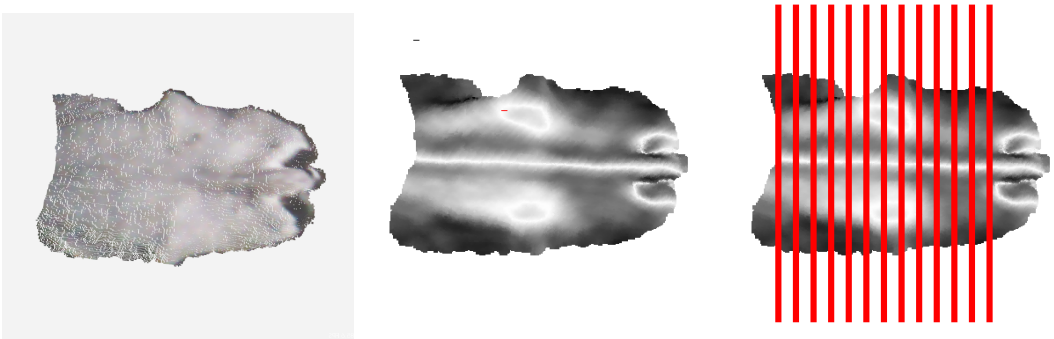
Let ${}^{N}P_i$ have k points and C be the covariance matrix:

$$C = \frac{1}{k}\sum_{j=1}^{k}(p_{ij} - \bar{p_i})(p_{ij} - \bar{p_i})^T \tag{3.34}$$

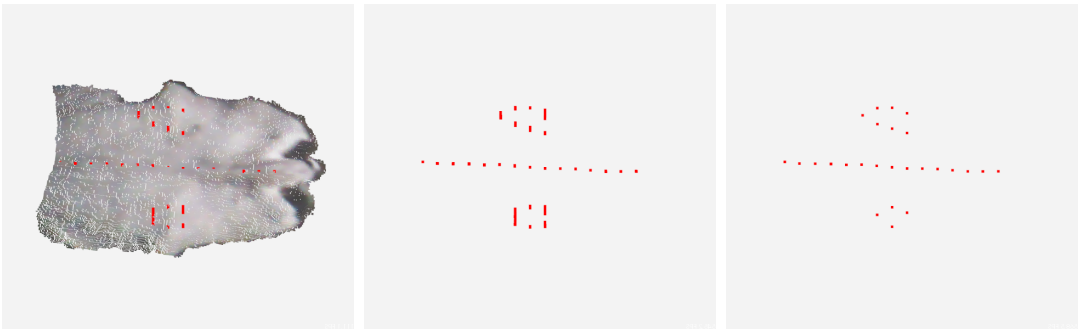
$$C.\vec{v_m} = \lambda_m.\vec{v_m},\ m \in \{0, 1, 2\} \tag{3.35}$$

where $\bar{p_i}$ is the mean of the points, λ_m is the m-th eigenvalue and $\vec{v_m}$ is the m-th eigenvector of the covariance matrix. The normal vector is simply equal to the eigenvector which has the smallest eigenvalue.

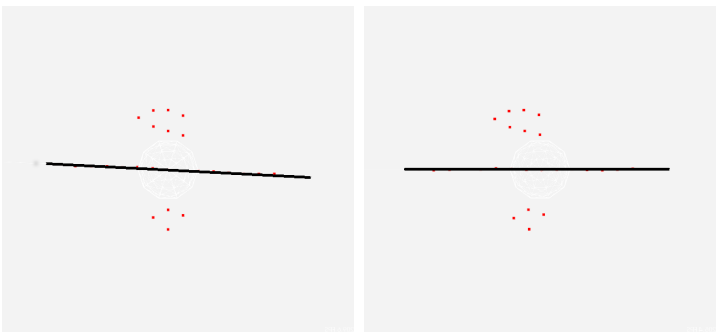
In Figure 3.12b a gray-scale normal image is shown, where pixel values are determined according to the z value of the corresponding normal vector of each point ($n_z \times 255$). As shown, spine, backbone and tail tip areas of the animal have lighter colors where direction of the normal vector of these points are almost parallel to the Z axis. Therefore, by extracting these points, keypoints are detected to recognize the



(a) Point cloud (b) Normal image (c) Slices



(d) Detected keypoints on the point cloud (e) Detected keypoints (f) Simplified keypoints



(g) Detected spine line (h) Corrected spine line

Figure 3.12. Illustration of spine detection and correction procedure

spine of the animal.

In Figure 3.12c, vertical hypothetical planes, which are orthogonal to X axis, are drawn with equal distances, d. Points, which are nearer than a certain threshold, ϵ_d, to these planes and whose normal vectors have z values greater than a threshold, ϵ_n, are selected as candidate keypoints, ${}^{K}P$, representing the spine:

$$d_{planes}(p_i) = \min_{k \in \mathbb{Z}}(\|x_i - k \times d\|) \tag{3.36}$$

$${}^{K}P = \{p \in {}^{S}P \mid d_{planes}(p) \leq \epsilon_d \ \wedge \ n_z \geq \epsilon_n\} \tag{3.37}$$

In Figure 3.12d and 3.12e, sample keypoints are shown. After these steps, keypoints are simplified by clustering based on their relative Euclidean distances and each cluster is represented by a point whose coordinates are the centroid of corresponding cluster. Let C_m be m-th cluster, where any point in this cluster has at least one neighboring point in the same cluster with smaller or equal distance of a certain threshold, ϵ_c:

$$C_m = \{p \in {}^{K}P \mid \exists p_m \in C_m \ \ \|p - p_m\| \leq \epsilon_c\} \tag{3.38}$$

Each cluster, C_m, is simply represented by a point, c_m, whose coordinates are average of the coordinates of points corresponding to that cluster.

$$c_m = \begin{bmatrix} \bar{x}_m & \bar{y}_m & \bar{z}_m \end{bmatrix}^T \tag{3.39}$$

In Figure 3.12(f) simplified keypoints are shown which are the centroids of clustered keypoints. As shown in Figure 3.12(g), a line model is fitted to the simplified keypoints with RANSAC algorithm (see Section 2.4). Let the center point of the line be $p_l(x_l, y_l, z_l)$ and the direction of the line $\vec{d_l}(x_d, y_d, z_d)$. All points are translated and rotated accordingly to make the detected spine line parallel with the X axis and cen-

ter point of the line is aligned with the origin. To align the center point, points are translated with a vector u and rotated $-\theta'$ around the Z axis to correct the direction:

$$u = \begin{bmatrix} -x_l & -y_l & 0 \end{bmatrix}^T \tag{3.40}$$

$$\theta' = \arctan\left(\frac{y_d}{x_d}\right) \tag{3.41}$$

Let $p(x, y, z)$ be a point in ${}^{S}P$. After applying the below transformation, the corrected point $p'(x', y', z') \in {}^{S'}P$ is acquired:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta' & \sin\theta' & 0 \\ -\sin\theta' & \cos\theta' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_l \\ y - y_l \\ z \end{bmatrix} \tag{3.42}$$

In Figure 3.12(h), the corrected spine line is shown.

3.4. Image Creation

To create images for the identification process, three dimensional point cloud of an animal is transformed into a two dimensional image. Then, each pixel value of the gray-scale image is determined with respect to the local features of the related three dimensional point. Let $p_i(x_i, y_i, z_i)$ be a point in ${}^{S'}P$ and $m(x_m, y_m)$ be the pixel related to this point, which is located at the y_m-th row and x_m-th column of the image M:

$$p_i(x_i, y_i, z_i) \in {}^{S'}P \tag{3.43}$$

$$m(x_m, y_m) \in M \tag{3.44}$$

As shown in Figure 3.13, $f(\cdot)$ is a function which maps each point in the point cloud to a pixel in the image space:

$$f(p_i(x_i, y_i, z_i)) = m(x_m, y_m) \tag{3.45}$$

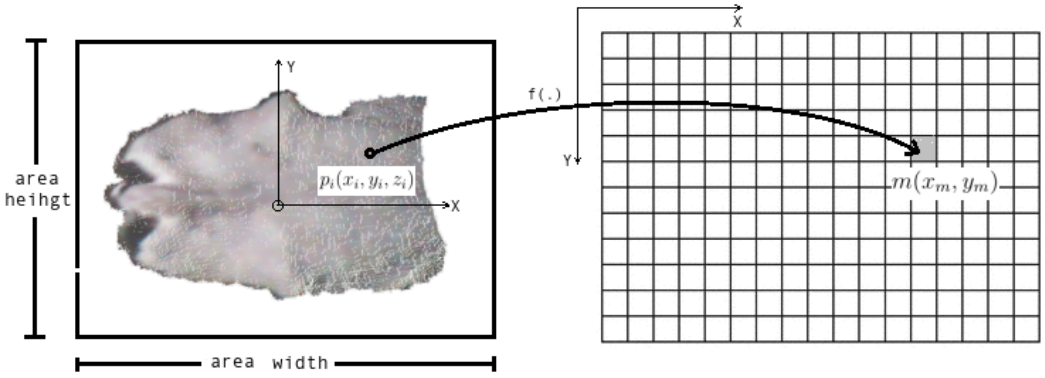


Figure 3.13. Mapping from 3D Euclidean space to 2D image space.

In 3D Euclidean space, distances are represented with real numbers, however in the image space pixel values are represented by integers. Let the function $g_{int}(\cdot)$ convert real number d to integer i as:

$$g_{int}(d) = \min_{i \in \mathbb{Z}}(i), \quad d \leq i \tag{3.46}$$

The width and height of the area to be mapped are represented with $area_width$ and $area_height$, respectively. To map three dimensional data, Euclidean distances are scaled by dividing with a scaling parameter ($image_scale$). Therefore, the height and the width of the created image are given by

$$image_width = g_{int}\left(\frac{area_width}{image_scale}\right) \tag{3.47}$$

$$image_height = g_{int}\left(\frac{area_height}{image_scale}\right) \tag{3.48}$$

and the point $p_i(x_i, y_i, z_i)$ is mapped to the pixel m at y_m-th row and x_m-th column by scaling and shifting accordingly to align the data with the center of the image:

$$x_m = g_{int}\left(\frac{x_i}{image_scale} + \frac{image_width}{2}\right) \tag{3.49}$$

$$y_m = -g_{int}\left(\frac{y_i}{image_scale} + \frac{image_height}{2}\right) \tag{3.50}$$

For a pixel in the image, there may be more than one point mapped to this pixel. Let ${}^{m}P$ represent points mapped to the pixel m:

$$ {}^{m}P = \{p \in {}^{S'}P \mid f(p) = m\} \tag{3.51} $$

Since the camera is located on the top of the animal looking downwards, the uppermost point is selected as the corresponding point (p_m) to the pixel m as:

$$ p_m = \{p(x, y, z) \in {}^{m}P \mid \underset{p}{arg\,max}\ (z)\} \tag{3.52} $$

and the point p_m is associated with the pixel m, i.e.,

$$ p_m \in {}^{S'}P \sim m \in M \tag{3.53} $$

3.4.1. Depth Image

Depth images are constructed by considering the heights of the points related to each pixel. In Figure 3.14, sample depth images captured from five different cows are shown. Let m be a pixel in image M related to point p_m in ${}^{S'}P$. The value of this pixel, δ_m, is determined according to the normalized z value of the related point with respect to the minimum and maximum heights, max_h, min_h; if there is no point related to this pixel, it is white colored:

$$ \delta_m = \begin{cases} g_{int}\left(255 \times \frac{max_h - z}{max_h - min_h}\right) & \text{if } p_m(x, y, z) \sim m \\ 255 & \text{if } \varnothing \sim m \end{cases} \tag{3.54} $$



Figure 3.14. Sample depth images captured from different cows.

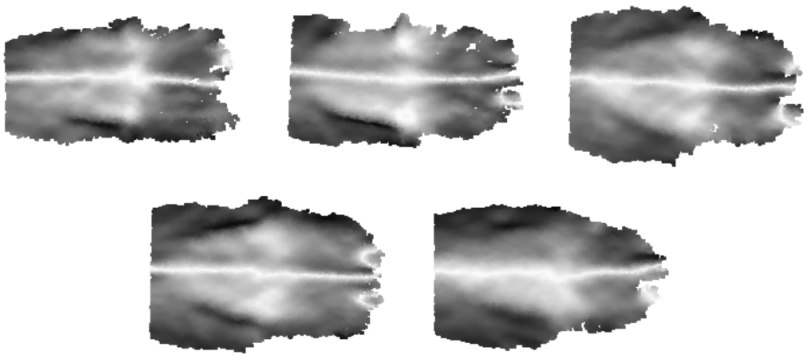


Figure 3.15. Sample normal-y images captured from different cows.

3.4.2. Normal-y Image

Before constructing the normal image, normal vectors of each point in ${}^{S'}P$ are calculated by considering neighboring points nearer than *normal_radius* with the same approach in Section 3.3.3. In Figure 3.15, sample normal-y images of five different cows are shown. For normal-y image, pixel values are colored according to y value of the normal vectors of the related points:

$$\delta_m = \begin{cases} g_{int}\left(255 \times (1 - n_y)\right) & \text{if } \vec{n}_m(n_x, n_y, n_z) \sim m \\ 255 & \text{if } \varnothing \sim m \end{cases} \tag{3.55}$$

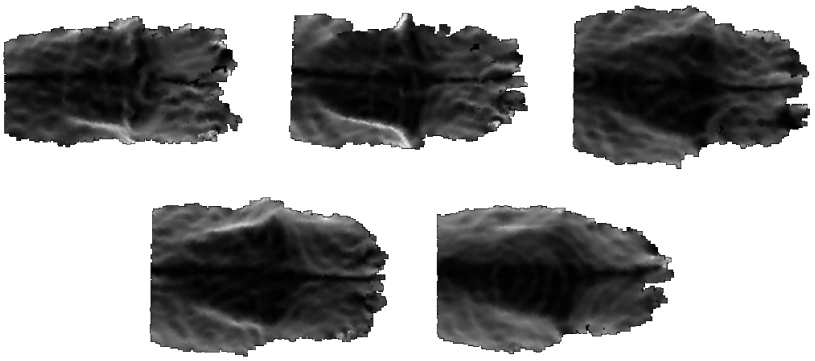


Figure 3.16. Sample normal-z images captured from different cows.

3.4.3. Normal-z Image

Similar to normal-y images, pixels of normal-z images are colored according to the z value of normal vector of each corresponding point.

$$\delta_m = \begin{cases} g_{int}\,(255 \times (1 - n_z)) & \text{if } \vec{n}_m(n_x, n_y, n_z) \sim m \\ 255 & \text{if } \varnothing \sim m \end{cases} \tag{3.56}$$

Sample normal-z images captured from five different cows are shown in Figure 3.16.

3.5. Identification with Images

In the identification process, recognition methods, namely Eigenface and Fisherface methods (see Sections 2.6 and 2.7), are applied to the constructed images. These two methods reduce the data by projecting images onto a lower dimensional space and register training images without loosing discriminative features. Eigenface method seeks a projection which preserves the most of the variance of the total data, and Fisherface projects samples where within-class scatter is minimized and between-class scatter is maximized. With these approaches, the identification process becomes applicable in real time and the memory requirement is also diminished.

In the implementation, the identifier is trained with a pre-labeled dataset

$$S_{training} = \{S_1, S_2, \ldots, S_k\}, \tag{3.57}$$

where this dataset consists of image sets

$$S_i = \{\tau_{i1}, \tau_{i2}, \ldots, \tau_{ij}\}, \tag{3.58}$$

captured from k individuals. After determination of the projection directions, each image, τ_{ij}, is represented with a weight vector, w_{ij}, and is registered in the memory:

$$\tau_{ij} \sim w_{ij} \tag{3.59}$$

$$W_i = \{w_{i1}, w_{i2}, \ldots, w_{ij}\} \tag{3.60}$$

$$S_{training} \sim W_{training} = \{W_1, W_2, \ldots, W_k\} \tag{3.61}$$

Let, τ, be a test image to be identified. First, this image is converted to a weight vector, w, with the same projection in the training procedure; subsequently, the ID of this sample is determined by the nearest neighbor criteria. Let

$$d(\tau, S_i) = \min\|w - w_{ij}\| \tag{3.62}$$

be the distance of an image to a subset of the training set corresponding to the ith individual. The test image is matched with the mth individual

$$\tau \sim S_m, \tag{3.63}$$

whose image set is nearer than the others:

$$m = \arg\min_i d\left(\tau, S_i\right). \tag{3.64}$$

If we have a set of test images

$$S_t = \{\tau_1, \tau_2, \ldots, \tau_k\}, \tag{3.65}$$

captured from the same animal, this set is matched with the mth individual

$$S_t \sim S_m, \tag{3.66}$$

where the minimum distance is found between kth image of the test set and a training image corresponding to the mth individual:

$$[m, k] = \arg\min_{i,j} d\left(\tau_j, S_i\right) \tag{3.67}$$

3.6. Summary of the Chapter

In this chapter, the proposed 3D cattle identification solution is detailed where animals are identified while they are passing through the field of view of RGBD cameras placed on stationary positions in a cattle farm. In Section 3.2, the acquired 3D points are filtered to extract points of the top back part of the animal; and in Section 3.3.1 the pose of the animal is roughly normalized by principle component analysis and brought into a canonical form. Subsequently, a finer pose normalization method is proposed in Section 3.3.3, which depends on recognition of the spine of the animal. In the last part of this chapter, different types of images such as depth, normal-y and normal-z; are constructed; and animals are identified by using face recognition methods, namely Eigenface and Fisherface.

The implementation and the experimental evaluation of the proposed solution in this chapter is to be discussed in Chapters 4 and 5.

4. IMPLEMENTATION AND DATASET ACQUISITION

To test the proposed solution in real life conditions, algorithms are implemented by using the C++ programming language and a dataset is acquired by installing cameras in a cattle farm which has 50 cows. In the first part of this chapter, developed tools and the complete identification application are explained; and in the second part, the 3D cattle dataset is introduced which consists of two videos captured with different cameras and environmental conditions.

4.1. Implementation

4.1.1. Debug Application

To debug and visually analyze the identification process, an application is developed in Qt framework, and algorithms are implemented by using the C++ programming language, where a sample snapshot of the user interface is shown in Figure 4.1. In the part labellec with “1”, there is an interactive window where the ground plane, point cloud and model elements such as direction and center of mass are shown. With buttons below the interactive window, three dimensional data can be saved or loaded with different formats and the snapshot of the view can be captured. In part “2”, there are buttons and parameter fields to apply implemented methods on the loaded model. In part “3”, there is a text output field, where information and results are printed in text format.

4.1.2. Calibration Tool

The camera calibration tool calculates the transformation between two cameras, and is developed to use more than one camera in the project. In this tool, there are two interactive windows where each of them shows the same scene captured from different

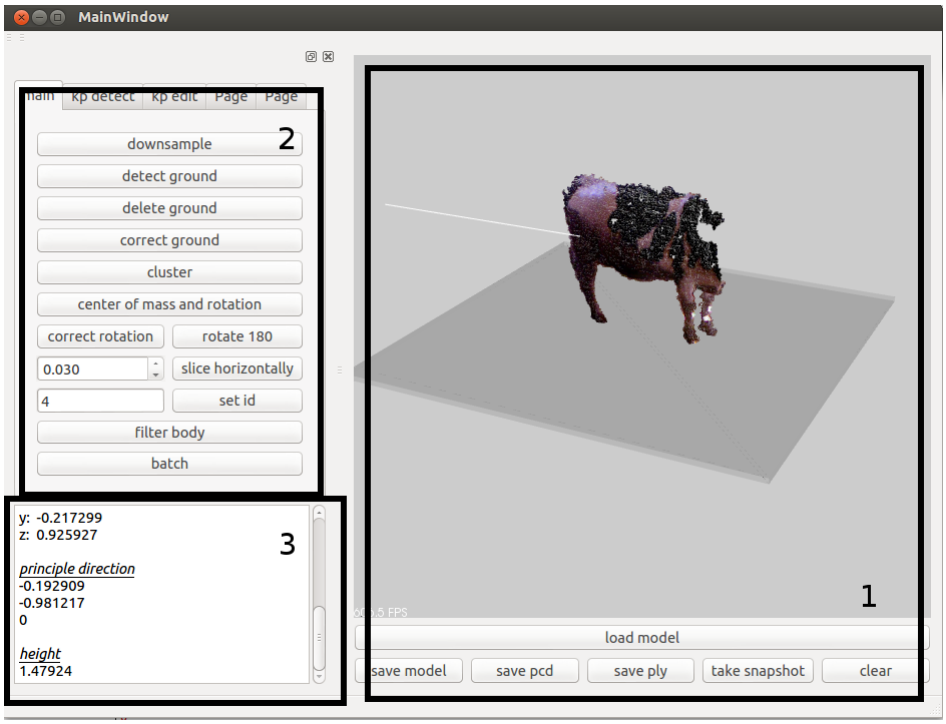


Figure 4.1. Snapshot of the user interface of the debug application.

angles. By manually selecting keypoint pairs on these samples, a rough transformation is calculated. In Figure 4.2, a sample scene is shown from different perspectives with the related keypoints. In this example, the transformation matrix is found as:

$$\begin{bmatrix} 0.923782 & -0.11887 & 0.364002 & -0.74489 \\ 0.107823 & 0.992881 & 0.0506023 & 0.0167988 \\ -0.367426 & -0.00749759 & 0.930023 & -0.0377713 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

After the first alignment, points which are further than three meters and that belong to the ground are extracted; and the transformation is fine tuned by applying the ICP (Iterative Closest Point) algorithm (see Section 2.5). In Figure 4.3 merged point clouds after the first transformation and final point cloud after applying ICP algorithm are shown. In this example the refined transformation matrix is calculated

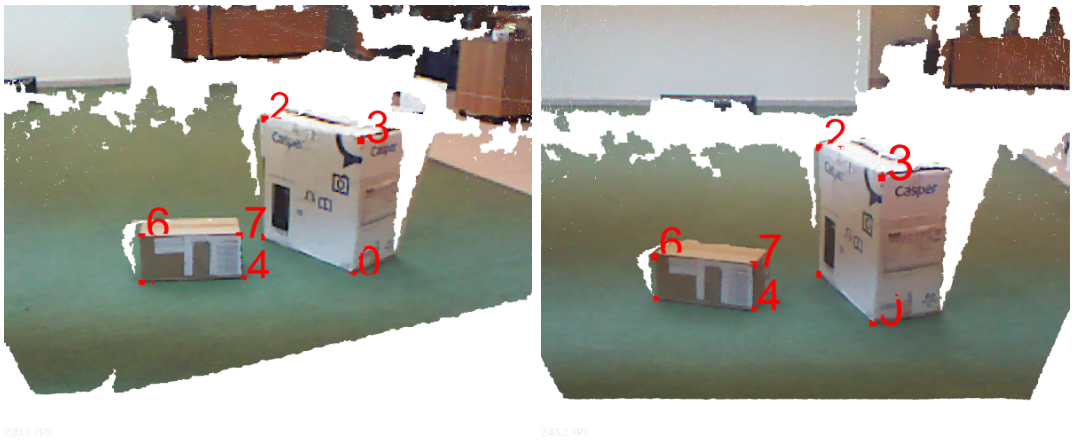


Figure 4.2. Manually selected keypoints. Left - scene from the first camera. Right - scene from the second camera.

as:

$$\begin{bmatrix} 0.934363 & -0.107881 & 0.339655 & -0.684459 \\ 0.105755 & 0.994102 & 0.02481 & 0.0700632 \\ -0.340329 & 0.0127288 & 0.940233 & -0.0600696 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

4.1.3. Dataset Grabber Tool

The dataset grabber tool is developed to capture datasets and to save them for off-line testing. It can work simultaneously with more than one camera; it captures the scene periodically and saves point clouds as "pcd" files. Every captured frame is noted in a text file with related capture information (id of the frame, camera ids, timestamps). A sample dataset file is shown below:

```
-------------------------------------------
DATASET FILE
```



Figure 4.3. Left - Transformed and merged point clouds with manually selected keypoints. Right - Same scene fine tuned with ICP algorithm.

```
Fri Nov 15 2013  14:06:13

format:
"frame id"
"camera id"    "pcd name"   "timestamp"
"camera id"    "pcd name"   "timestamp"

0
#2 0_2.pcd 40.8793
#1 0_1.pcd 40.8921

1
#2 1_2.pcd 41.9748
#1 1_1.pcd 41.9843

...
-------------------------------------------------
```

4.1.4. Real-time Identification Application

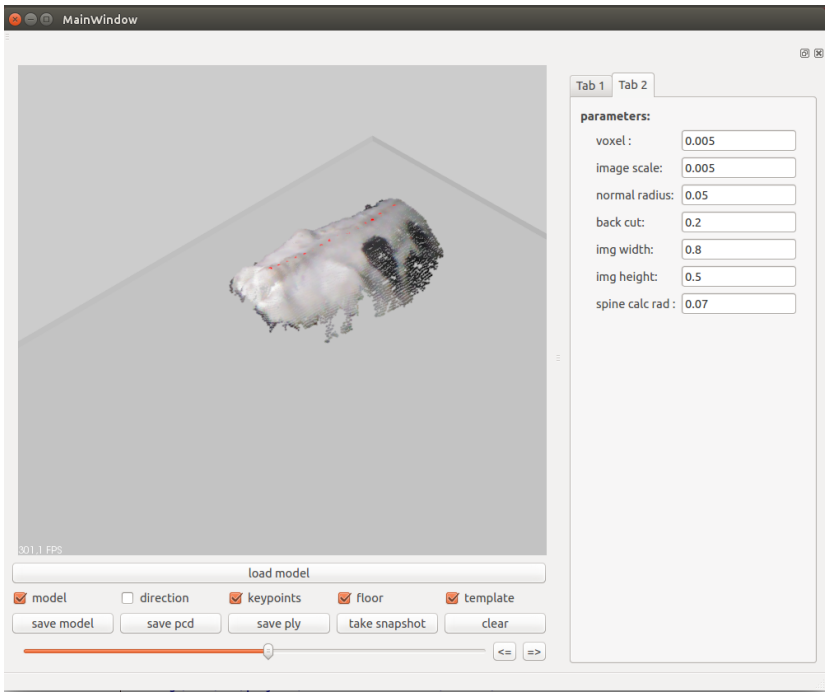
This application is developed as a prototype in order to test the identification process in real time with real data. It has "train" and "identification" modes; and can be used by directly connecting to a camera or a dataset file. When a new frame is captured, the application processes the raw data and detects whether there is a cow in the scene or not. After noticing the animal, it processes the point cloud and brings it into the canonical form, then predicts the id of the animal. In Figure 4.4, two snapshots of this application are shown. On the left side of the user interface, there is an interactive window which shows the three dimensional data and the state of the process. On the right side, there are two tabs where parameters of the processes are set. The parameters used in image creation process are as follows:

- *voxel size:* voxel grid size in meters used in down-sampling of the original data,
- *image scale:* scale factor used for transforming points from 3D Euclidian space to 2D pixel space,
- *normal radius:* neighboring radius in meters used in normal estimation process,
- *back cut width:* width threshold in meters to cut the back part of the animal,
- *image width:* width of the area in meters to be transformed to image space,
- *image height:* height of the area in meters to be transformed to image space,
- *spine calculation radius:* neighboring radius in meters used in normal estimation in spine recognition.

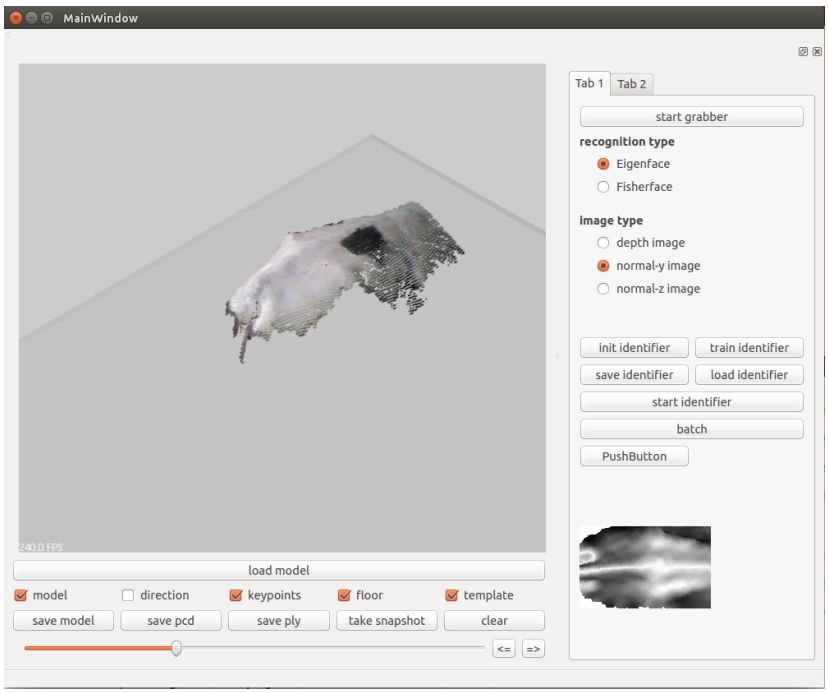
The parameters used in the identification process are:

- *identification type:* The identification method is selected among Eigenface and Fisherface algorithms.
- *image type:* The image type is selected among depth, normal-y and normal-z images.

With this application once an identifier is trained, it can be saved to be used later; or a pre-trained identifier can be loaded. It can be also used in batch mode, where the parameters are changed systematically to determine the optimal values for better recognition.

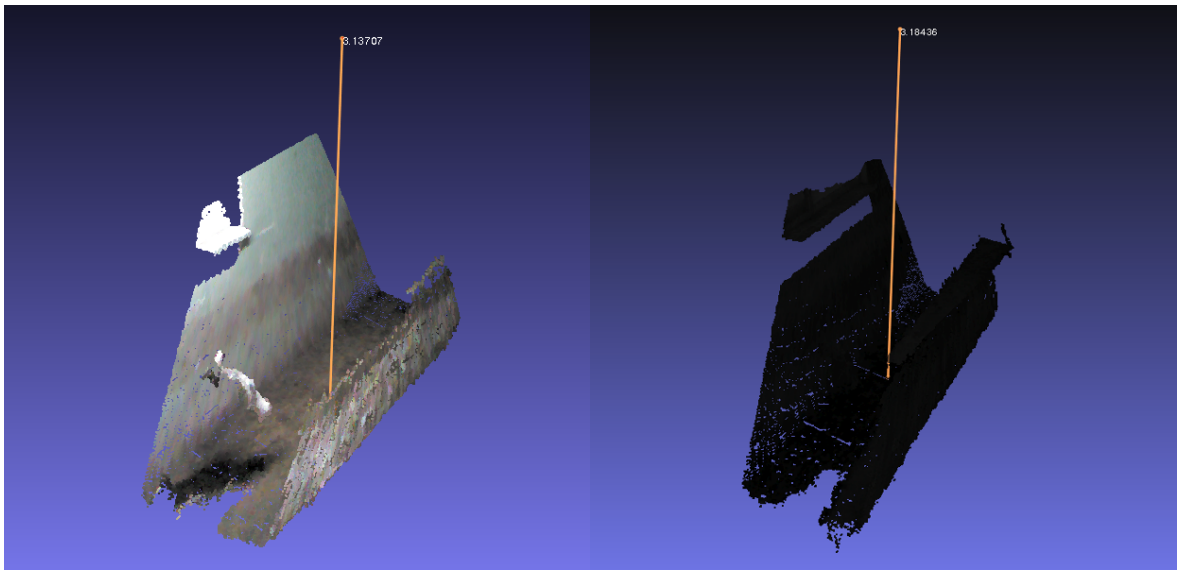


(a) Interactive window and parameters tab



(b) Interactive window and identification tab

Figure 4.4. Snapshots of the user interface of the real-time identification application.



(a) Camera #1, $h_1 = 3.13$ m, with lighting (b) Camera #2, $h_2 = 3.18$ m, no lighting

Figure 4.5. 3D point clouds captured from two cameras placed at different points of the corridor.

4.2. Dataset Acquisition

To test the developed system, a dataset is acquired at a cattle farm which has 50 cows. In this farm, there are semi-open environments with shelters where cows rest and get nourished. The dataset is acquired with two Kinect cameras which are installed at a 1 meter width corridor, where cows pass in groups twice a day to achieve the milking room:

- *Camera 1 :* The first camera is placed near the entrance of the corridor at 3.13 m height where the environment is lighted with a lamp.
- *Camera 2 :* The second camera is placed at near the end of the corridor at 3.18 m height where there is no lighting in the environment.

In Figure 4.5 two snapshots captured from the sensors are shown respectively. Three dimensional top views of each animal are captured by these cameras with 250 millisecond

Table 4.1. Number of samples of each cow in two datasets.

cow id	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Dataset 1	7	6	5	9	2	1	3	6	5	3	3	1	3	2	3
Dataset 2	4	3	8	27	22	2	4	5	5	4	4	1	3	2	2

cow id	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
Dataset 1	5	21	2	6	2	16	2	5	5	3	2	16	22	5	3
Dataset 2	5	4	2	2	3	4	3	5	4	7	2	1	2	1	2

cow id	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**
Dataset 1	4	24	8	3	5	10	4	8	5	5	4	4	6	10	2
Dataset 2	2	2	2	2	2	2	3	2	2	2	2	2	4	5	2

cow id	**45**	**46**	**47**	**48**	**49**	**Total**
Dataset 1	2	7	2	5	4	296
Dataset 2	2	4	3	6	3	197

intervals; and two videos are acquired.

These two videos are processed and healthy frames are extracted where the top back part of the related animal is in the field of the view. Two datasets are acquired where the number of samples for each animal is shown in Table 4.1.

- *Dataset 1 :* The first dataset consists of 296 samples. 11 cows have equal or less than 2 samples, and 19 cows have equal or less than 3 samples.
- *Dataset 2 :* The second dataset consists of 197 samples; where 24 cows have equal or less than 2 samples and 29 cows have equal or less than 3 samples.

Low number of samples affects identification results negatively. Therefore, the first dataset can be deemed as healthier where the total number of samples and the

number of samples for individuals are higher than those of the second dataset. On the other hand, the visual appearances of the cows are not visible in the second dataset since the second video is captured with no light source. Moreover, the fact that videos are captured while animals are passing under the cameras, and animals are not in stationary positions, should be considered in the evaluation of the test results.

4.3. Summary of the Chapter

In the first part of this chapter, developed applications are explained; where these applications are:

- *Debug Application :* Methods and algorithms are implemented and tested with this application before developing a prototype application.
- *Calibration Tool :* Estimates the relative transformations of multiple cameras to merge point clouds captured from different sensors.
- *Dataset Grabber Tool :* Captures datasets and saves them for off-line testing. It can work simultaneously with more than one camera and saves point clouds as "pcd" files.
- *Real-time Identification Application :* This application is a prototype, to test the identification process in real time with real data. It has "train" and "identification" modes; and can be used by directly connecting to a camera or a dataset file.

In the second part, the 3D cattle dataset is presented and analyzed, which is acquired with two cameras installed in two different places of a corridor, where animals pass through. Sample depth, normal-y and normal-z images from this dataset are shown in Appendices A, B and C, respectively.

5. EXPERIMENTAL EVALUATION

In this chapter, experimental identification results are presented and analyzed. Two types of identification methods, Eigenface and Fisherface; and three types of image types, depth, normal-y and normal-z images are evaluated in the tests.

As discussed in the previous chapter, the 3D cattle dataset consists of two 3D image sets captured from two cameras. These datasets are fed to the real-time identification application with respect to the related time-stamps and identification results are obtained. The first set has relatively higher number of samples, which is used for training of the identifier; and the second set is used for testing the success of the identification process.

In the identification, the trained identifier outputs the predicted id, m, given in (3.64) for the jth test image of the ith cow, i.e.,

$$m = f_{id}\left(\tau_{ij}\right). \tag{5.1}$$

Identification of this sample is said to be successful if the predicted id is equal to the actual id of the test image:

$$g\left(\tau_{ij}\right) = \begin{cases} 1, & \text{if } f_{id}\left(\tau_{ij}\right) = i, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

If we have a set of images, S_i, captured from the same animal, identifier outputs the predicted id in (3.67) as follows:

$$m = f_{id}\left(S_i\right) \tag{5.3}$$

This animal is said to be identified successfully if the predicted id is equal to the actual id of the animal:

$$g\left(S_i\right) = \begin{cases} 1, & \text{if } f_{id}\left(S_i\right) = i, \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

There are two success criteria for the identification process: sample success rate (SSR) and class success rate (CSR). Suppose that we have a test set with N samples, captured from k animals. Sample success rate is the ratio of number of successfully identified samples and the total number of samples computed by

$$\text{SSR} = \frac{\sum_i \sum_j g\left(\tau_{ij}\right)}{N}, \tag{5.5}$$

and the class success rate is the ratio of number of successfully identified animals and the total number of animals given by

$$\text{CSR} = \frac{\sum_i g\left(S_i\right)}{k}. \tag{5.6}$$

As depicted in Figure 5.1, the identifier has the following parameters

A - voxel size,
B - image scale,
C - normal radius,
D - back cut width,
E - area width,
F - area height,
G - spine calculation radius;

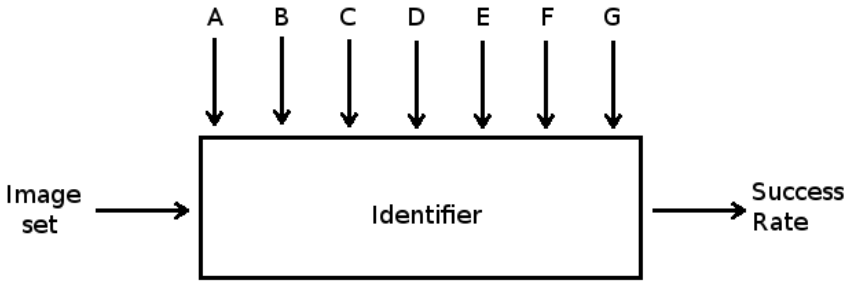


Figure 5.1. Identification process.

to which proper values should be assigned for successful identification (see Section 4.1.4).

5.1. Eigenface Method

In the tests with the Eigenface method, the following parameters are determined to be optimal for identification:

- *voxel size:* 0.005 m
- *image scale:* 0.005
- *normal radius:* 0.05 m
- *back cut width:* 0.3 m
- *area width:* 0.9 m
- *area height:* 0.52 m
- *spine calculation radius:* 0.07 m

The resulting identification rates are as shown in Table 5.1. The highest identification rates are obtained with normal-y images where 146 of 197 samples (74%), and 44 of 50 cows (88%) are identified successfully; and the lowest rates are obtained with depth images.

Table 5.1. Identification results of Eigenface method.

EIGENFACE METHOD	Sample Success Rate	Class Success Rate
Depth Image	84/197 (42%)	24/50 (48%)
Normal-y Image	146/197 **(74%)**	44/50 **(88%)**
Normal-z Image	113/197 (57%)	31/50 (62%)

5.2. Fisherface Method

For the Fisherface method, results in Table 5.2 are obtained with the following parameters:

- *voxel size:* 0.006 m
- *image scale:* 0.005
- *normal radius:* 0.05 m
- *back cut width:* 0.2 m
- *area width:* 0.9 m
- *area height:* 0.5 m
- *spine calculation radius:* 0.08 m

In general, identification rates of the Fisherface method are similar to the ones in the Eigenface method. The lowest rates are obtained with depth images and the highest identification rates are observed with normal-y images, where 159 of 197 samples (77%), and 43 of 50 cows (86%) are identified successfully.

In the tests, the highest sample success rate, 77%, is achieved with the Fisherface method and the highest class success rate, 88%, is achieved with the Eigenface method by using normal-y images. Therefore, we can conclude that normal-y image is the most appropriate image type for this solution. Although the highest class success rate

is achieved with the Eigenface method, average SSR and CSR of the Fisherface method are 16.0% and 7.1% higher than those of the Eigenface method, respectively. Therefore, Fisherface method could be more applicable in a real world application with normal-y images.

Table 5.2. Identification results of Fisherface method.

FISHERFACE METHOD	**Sample Success Rate**	**Class Success Rate**
Depth Image	116/197 (56%)	31/50 (62%)
Normal-y Image	159/197 **(77%)**	43/50 **(86%)**
Normal-z Image	123/197 (59%)	32/50 (64%)

5.3. Effect of Spine Recognition

A pose normalization method has been proposed in Section 3.3.3. After a rough pose normalization, a finer normalization is achieved by recognizing the spine of the animal. In this section, the same tests are carried out by applying the Eigenface and Fisherface methods on the samples whose poses are are roughly normalized and not fine tuned with spine recognition. The identification results are summarized in Tables 5.3 and 5.4.

In these tests, it is observed that average SSR and CSR have decreased by 12.0% and 10.7%, respectively. Therefore, we can conclude that the normalization process based on spine recognition has a significant positive effect on the identification process.

Table 5.3. Identification results of Eigenface method without spine correction.

EIGENFACE METHOD	Sample Success Rate	Class Success Rate
Depth Image	74/197 (37%)	19/50 (38%)
Normal-y Image	127/197 **(64%)**	37/50 **(74%)**
Normal-z Image	107/197 (54%)	33/50 (66%)

Table 5.4. Identification results of Fisherface method without spine correction.

FISHERFACE METHOD	Sample Success Rate	Class Success Rate
Depth Image	94/197 (45%)	25/50 (50%)
Normal-y Image	131/197 **(63%)**	39/50 **(78%)**
Normal-z Image	114/197 (55%)	30/50 (60%)

5.4. Effect of Number of Samples

Finally, the same tests are conducted by switching the training and test sets, where the results are summarized in Tables 5.5 and 5.6. Previously, the dataset which has 50.3% higher number of samples was chosen for the training and the other dataset was used for testing. In this section, the training and test sets are switched; therefore the identifier is trained with fewer number of samples. We have found that average sample success and average class success rates have decreased by 16.0% and 8.2%, respectively. This suggests that the number of training samples is an important aspect which should be considered to get higher identification rates.

Table 5.5. Identification results of Eigenface method with switched datasets.

EIGENFACE METHOD	Sample Success Rate	Class Success Rate
Depth Image	99/296 (33%)	23/50 (46%)
Normal-y Image	170/296 **(57%)**	37/50 **(74%)**
Normal-z Image	154/296 (52%)	31/50 (62%)

Table 5.6. Identification results of Fisherface method with switched datasets.

FISHERFACE METHOD	Sample Success Rate	Class Success Rate
Depth Image	146/296 (48%)	26/50 (52%)
Normal-y Image	190/296 **(62%)**	38/50 **(76%)**
Normal-z Image	177/296 (58%)	33/50 (66%)

5.5. Summary of the Chapter

In this chapter, the test results with the proposed 3D cattle identification system by using the acquired datasets has been presented. First of all, proper values have been assigned to the parameters of the real-time identification application; and experiments have been conducted with different types of images and identification methods. In summary, the following conclusions are drawn:

- The highest success rates are achieved by using normal-y images,
- The highest class success rate, 88%, is achieved with the Eigenface method,
- The highest sample success rate, 77%, is achieved with the Fisherface method,
- In average, the Fisherface method results in higher identification rates,
- The proposed pose normalization method has a significant positive effect,
- The number of training samples is an important factor for better identification.

6. CONCLUDING REMARKS

In this thesis, a cow identification system based on 3D shape analysis of top back part of the animals has been proposed. There are prominent features of this solution which differentiates it from others in the literature, i.e., it

- does not need any markers or external devices placed on the animal,
- works in even unlighted environments,
- identifies even black cows without distinctive coat patterns,
- is relatively cheaper,
- enables accurate positioning.

Another contribution of this thesis is a novel pose normalization method which is based on recognition of the spine of the animal.

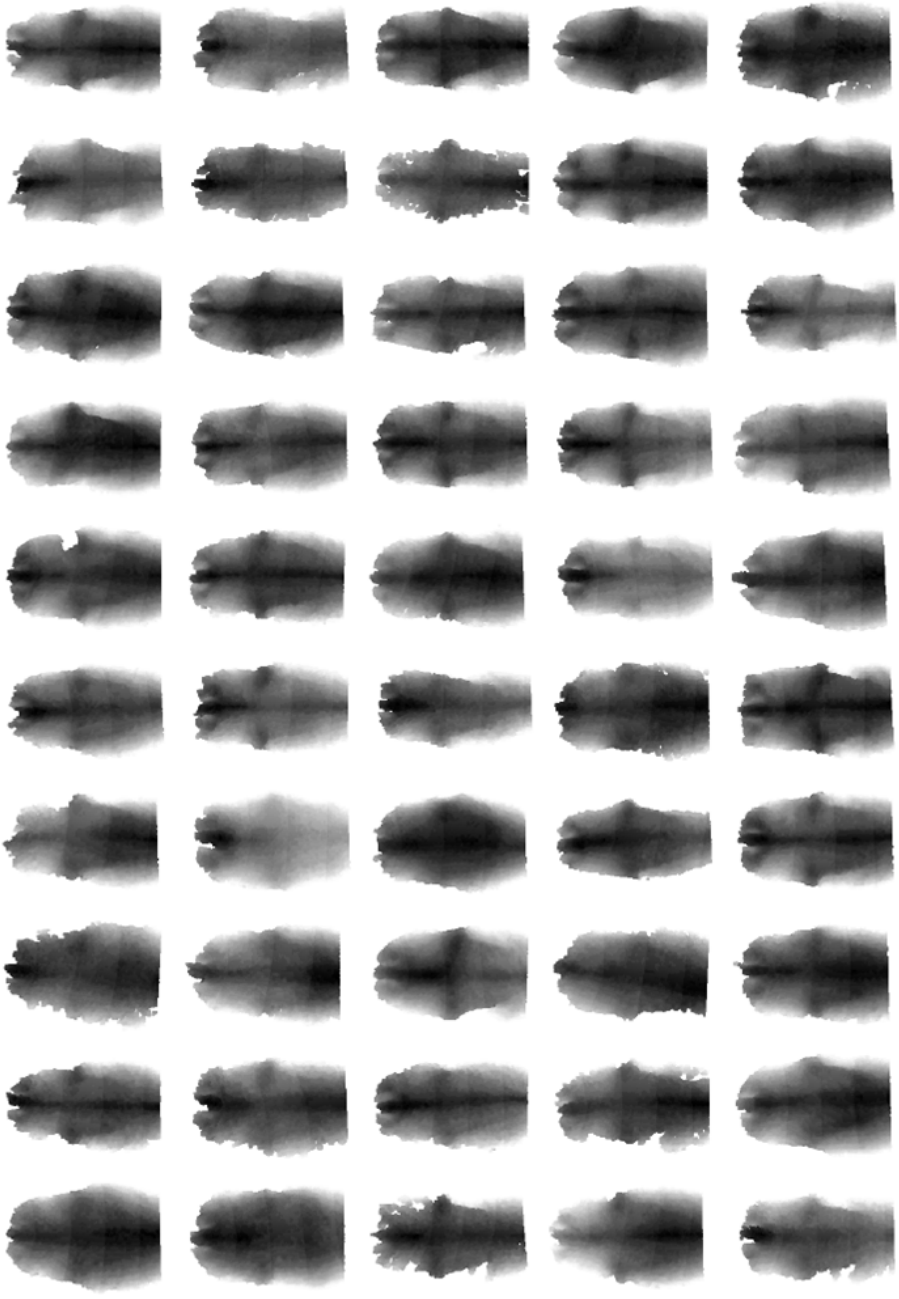
To evaluate the efficiency of the proposed system, it is implemented as a real-time prototype application and a 3D cattle dataset is acquired which, to our knowledge, is unique in the literature. This dataset consists of three dimensional point cloud sequences of 50 different cows, captured in lighted and unlighted conditions. More specifically,

- the dataset is gathered from moving animals and it contains relatively few number of samples per animal,
- training and testing videos are captured with different cameras in different lighting conditions (lighted and dark),
- most of the cows are black and without distinctive coat patterns.

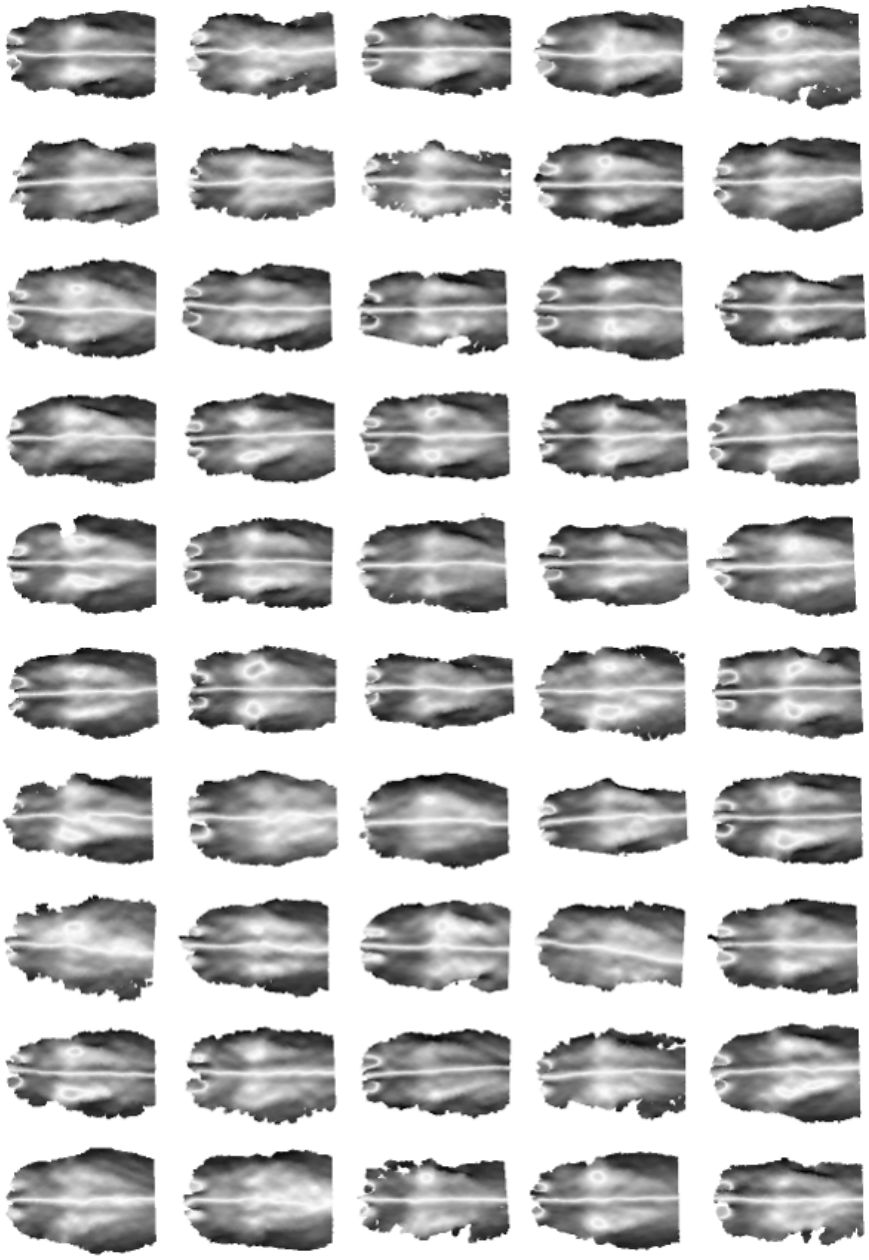
Applicability of the proposed solution has been verified by testing with the acquired dataset. Convincing results are obtained where %88 of 50 cows are identified successfully. For identification, two famous face recognition methods, namely Eigen-

face and Fisherface methods, have been used. Therefore, the presented identification results also serve as a base point for future research on this topic.

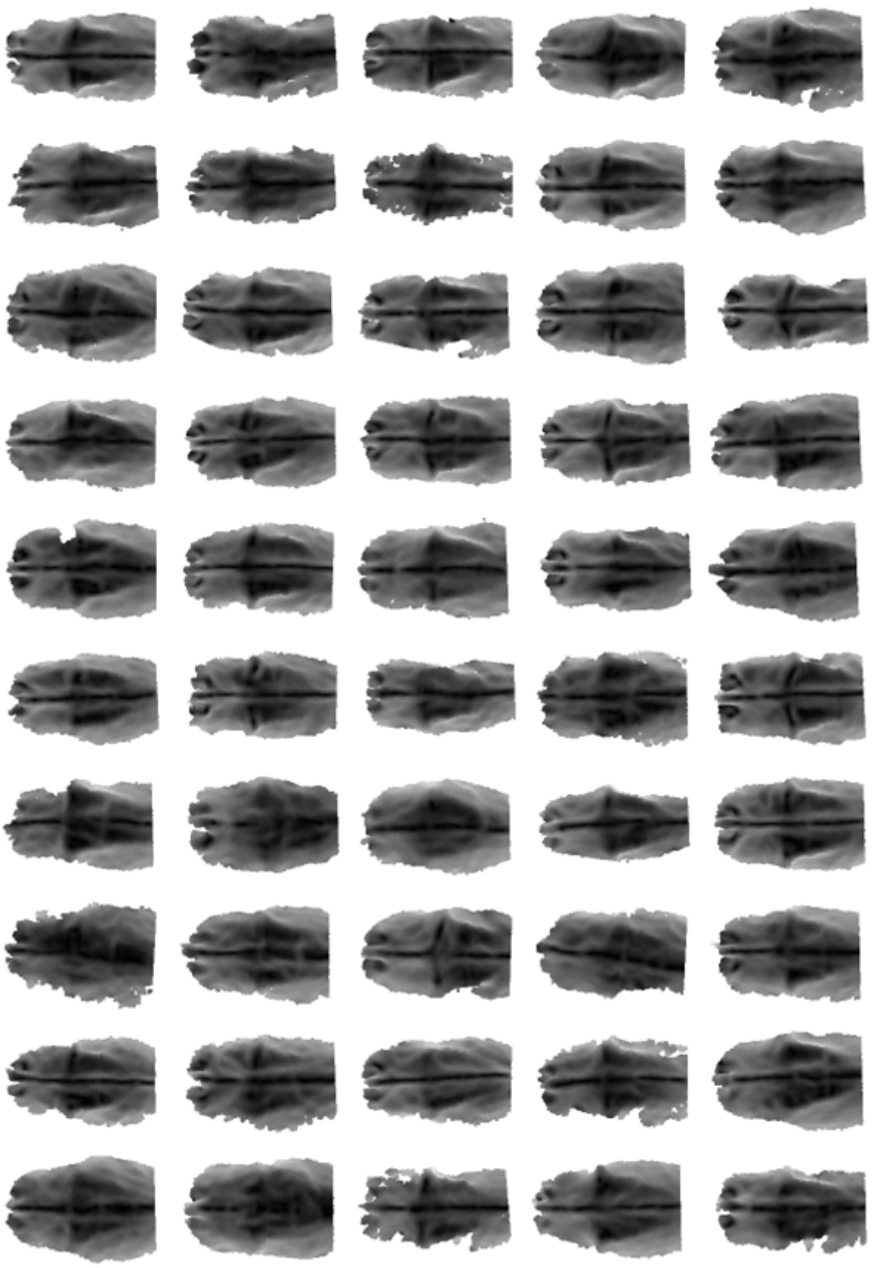
APPENDIX A: SAMPLE DEPTH IMAGES



APPENDIX B: SAMPLE NORMAL-Y IMAGES



APPENDIX C: SAMPLE NORMAL-Z IMAGES



REFERENCES

1. Kawasue, K., T. Ikeda, T. Tokunaga and H. Harada, "Three-Dimensional Shape Measurement System for Black Cattle Using Kinect Sensor", *International Journal of Circuit and Signal Processing*, Vol. 7, No. 4, pp. 222–230, 2013.

2. Khoshelham, K. and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications", *Sensors*, Vol. 12, No. 2, pp. 1437–1454, 2012.

3. Bronstein, A. M., M. Bronstein, M. M. Bronstein and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*, Monographs in Computer Science, Springer, 2008.

4. "Livestock Production Statistics", *Turkish Statistical Institute*, `http://www.tuik.gov.tr`, [Accessed January 2015].

5. Al-Ammar, M. A., S. Alhadhrami, A. Al-Salman, A. Alarifi, H. S. Al-Khalifa, A. Alnafessah and M. Alsaleh, "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms", *2014 International Conference on Cyberworlds (CW)*, pp. 245–252, IEEE, 2014.

6. Godil, A., R. Eastman and T. Hong, *Ground Truth Systems for Object Recognition and Tracking*, Vol. 7923, NISTIR, 2013.

7. Huhtala, A., K. Suhonen, P. Mäkelä, M. Hakojärvi and J. Ahokas, "Evaluation of Instrumentation for Cow Positioning and Tracking Indoors", *Biosystems Engineering*, Vol. 96, No. 3, pp. 399–405, 2007.

8. Schlecht, E., C. Hülsebusch, F. Mahler and K. Becker, "The Use of Differentially Corrected Global Positioning System to Monitor Activities of Cattle at Pasture", *Applied Animal Behaviour Science*, Vol. 85, No. 3, pp. 185–202, 2004.

9. Bouet, M. and A. L. Dos Santos, "RFID Tags: Positioning Principles and Localization Techniques", *Wireless Days, 2008. WD'08. 1st IFIP*, pp. 1–5, IEEE, 2008.

10. Briner, T., J.-P. Airoldi, F. Dellsperger, S. Eggimann and W. Nentwig, "A New System for Automatic Radiotracking of Small Mammals", *Journal of Mammalogy*, Vol. 84, No. 2, pp. 571–578, 2003.

11. Pei, L., R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen and Y. Chen, "Using Inquiry-based Bluetooth RSSI Probability Distributions for Indoor Positioning", *Journal of Global Positioning Systems*, Vol. 9, No. 2, pp. 122–130, 2010.

12. Mazuelas, S., A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas and E. J. Abril, "Robust Indoor Positioning Provided by Real-time RSSI Values in Unmodified WLAN Networks", *Selected Topics in Signal Processing, IEEE Journal of*, Vol. 3, No. 5, pp. 821–831, 2009.

13. Li, B., J. Salter, A. G. Dempster and C. Rizos, "Indoor Positioning Techniques Based on Wireless LAN", *LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, 2006.

14. Smith, A., H. Balakrishnan, M. Goraczko and N. Priyantha, "Tracking Moving Devices with the Cricket Location System", *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pp. 190–202, ACM, 2004.

15. Gu, Y., A. Lo and I. Niemegeers, "A Survey of Indoor Positioning Systems for Wireless Personal Networks", *Communications Surveys & Tutorials, IEEE*, Vol. 11, No. 1, pp. 13–32, 2009.

16. Von Keyserlingk, M., J. Rushen, A. M. de Passillé and D. M. Weary, "Invited Review: The Welfare of Dairy Cattle - Key Concepts and the Role of Science",

Journal of Dairy Science, Vol. 92, No. 9, pp. 4101–4111, 2009.

17. Mautz, R. and S. Tilch, "Survey of Optical Indoor Positioning Systems", *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pp. 1–7, IEEE, 2011.

18. Zhao, L., Z. Liu, T. Li, B. Huang and L. Xie, "Moving Target Positioning Based on a Distributed Camera Network", *Mathematical Problems in Engineering*, Vol. 2014, 2014.

19. Chi, S., C. H. Caldas and D. Y. Kim, "A Methodology for Object Identification and Tracking in Construction Based on Spatial Modeling and Image Matching Techniques", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 24, No. 3, pp. 199–211, 2009.

20. Viola, P. and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, pp. I–511, IEEE, 2001.

21. Kalal, Z., K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 34, No. 7, pp. 1409–1422, 2012.

22. Burghardt, T., *Visual animal biometrics*, Ph.D. Thesis, PhD thesis, University of Bristol, UK, 2008.

23. Hillman, G., B. Wursig, G. Gailey, N. Kehtarnavaz, A. Drobyshevsky, B. Araabi, H. Tagare and D. Weller, "Computer-assisted Photo-identification of Individual Marine Vertebrates: a Multi-species System", *Aquatic Mammals*, Vol. 29, No. 1, pp. 117–123, 2003.

24. City, K. and K. City, "Review of Bowhead Whale Aerial Photographic Studies in 2003-06", *Fisheries Science*, pp. 1–6, 2006.

25. Markowitz, T. M., A. D. Harlin and B. Würsig, "Digital Photography Improves Efficiency of Individual Dolphin Identification", *Marine Mammal Science*, Vol. 19, No. 1, pp. 217–223, 2003.

26. Stahl, Henry and Schadler, Kristina and Hartung, E., "Capturing 2D and 3D Biometric Data of Farm Animals Under Real-life Conditions", *International Workshop on Computer Image Analysis in Agriculture*, 17, 2012.

27. Kumar, S. and S. K. Singh, "Biometric Recognition for Pet Animal", *Journal of Software Engineering and Applications*, Vol. 2014, 2014.

28. Loos, A. and M. Pfitzer, "Towards Automated Visual İdentification of Primates Using Face Recognition", *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pp. 425–428, IEEE, 2012.

29. Corkery, G., U. A. Gonzales-Barron, F. Butler, K. McDonnell and S. Ward, "A Preliminary Investigation on Face Recognition as a Biometric Identifier of Sheep", The American Society of Agricultural and Biological Engineers, 2007.

30. Kim, H. T., Y. Ikeda and H. L. Choi, "The Identification of Japanese Black Cattle by Their Faces", *Training*, Vol. 1, p. e11e12, 2005.

31. Cai, C. and J. Li, "Cattle Face Recognition Using Local Binary Pattern Descriptor", *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pp. 1–4, IEEE, 2013.

32. Rusu, R. B. and S. Cousins, "3D is Here: Point Cloud Library (PCL)", *2011 IEEE International Conference on Robotics and Automation*, Vol. 29, No. 2, pp. 1–4, 2011.

33. Bradski, G., "The OpenCV Library", *Doctor Dobbs Journal*, Vol. 25, No. 11, pp. 120–126, 2000.

34. Freedman, B., A. Shpunt, M. Machline and Y. Arieli, "Depth Mapping Using Projected Patterns", Google Patents, Apr. 3 2012, uS Patent 8,150,142.

35. Fischler, M. A. and R. C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.

36. Schnabel, R., R. Wahl and R. Klein, "Efficient RANSAC for Point Cloud Shape Detection", *Computer graphics forum*, Vol. 26, pp. 214–226, Wiley Online Library, 2007.

37. Yang, M. Y. and W. Förstner, "Plane Detection in Point Cloud Data", *Proceedings of the 2nd int conf on machine control guidance, Bonn*, Vol. 1, pp. 95–104, 2010.

38. Besl, P. J. and N. D. McKay, "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239–256, 1992.

39. Rusinkiewicz, S. and M. Levoy, "Efficient Variants of the ICP Algorithm", *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.

40. Turk, M. A. and A. P. Pentland, "Face Recognition Using Eigenfaces", *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.

41. Scheenstra, A., A. Ruifrok and R. C. Veltkamp, "A Survey of 3D Face Recognition Methods", *Audio-and Video-Based Biometric Person Authentication*, pp. 891–899, Springer, 2005.

42. Fisher, R. A., "The Use of Multiple Measurements in Taxonomic Problems", *Annals of eugenics*, Vol. 7, No. 2, pp. 179–188, 1936.

43. Belhumeur, P. N., J. P. Hespanha and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 19, No. 7, pp. 711–720, 1997.

44. "Face Recognition with OpenCV - OpenCV 2.4.11.0 Documentation", `http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html`, [Accessed January 2015].

45. Goldstein, H., C. Poole and J. Safko, *Classical Mechanics*, Addison Wesley, 2002.