Newcastle

University

Live

Pig Weight Estimation

With Images

Project & Dissertation

Schools of Agriculture & Computing

Mahmoud Moshtagh

MSc 2014

Newcastle University

Abstract

Monitoring livestock using computer based technology would enable the owners to check the condition of their animals with higher efficiency meaning more frequency, less time and better organization. By having a system that can visually track animals in real-time, cost effective information such as health status, growth rate and many other can be retrieved to manage higher productivity. This research paper only focuses on pig weight estimation through analysis of visual data in real-time. Later on the common industrial approach to visual weight estimation is explained. Following that the method developed as a result of this project to achieve the same result is described step by step. The application created based on project's method only uses depth data collected from Kinect camera as an input. The depth data is collection of videos recorded by Kinect camera from the top view perspective. By the end of this paper the necessary elements of image processing for the purpose of this project are covered so that future application can concentrate on optimising and further expanding the overall capabilities of this technology.

In total 8 pigs went on data collection trial for three sessions on three different days. Manual weight of the pigs was recorded as a reference point to be compared with the final estimation results. Using Kinect camera videos were captured while the pigs were walking on an aisle. The camera was placed on ceiling shooting in a top view perspective. Throughout the course of project an application was developed to take the necessary measurements read from frames and use them for finding pig's body weight. This research is first going to clarify pig weight estimation process before simplifying it for final application at the end. Then it provides an instructional guidance for procedural approach to building real-time image processing applications including geometrical and statistical measurements. The upcoming section explains common industrial approach for body weight estimation.

Followed by the background research, design and structure of finished application is laid out. Break down of the entire architecture and implementation is drawn and explained down to every piece and its purpose. Image processing method designed here can be applied to put together other similar applications using the same steps.

The steps are described as three stages of pre, mid and post-processes. Stages are series of procedures which in order focus on providing tools and preparing the frame for data extraction, extracting data (e.g. feature detection, measurements), analysing the data to conclude result (e.g. finding average weight). Focus on expansion of each stage can improve performance in different ways.

In evaluation acheivements with the strongest and most valuble aspects of the program are presented. Statistical data such as graphs and tables are given as proof that indeed the results are accurate and worthwhile.

In conclusion routes for further development and addition of possible features are pointed out.

Keywords

Kinect, depth data, pigs, measurement, weight, estimation, growth, top view perspective, image processing, projected area, livestock control, remote management, camera monitoring, video data, real-time, convex hull, arbitrary oriented minimum bounding box, object detection, connected component labelling, linear correlation

1.INTRODUCTION

Integration of different types of technology with industry in our case the farming industry in specific can enhance the performance by undertaking tasks that originally would have had to be done by people. This project is about engaging image processing technology to facilitate management of livestock in specific for Newcastle University's pig farm which belongs to agriculture department[1]. The aim of this project is to provide tools that can estimate pig's weight based on graphic images gathered from the pigs in real-time.

In short the advantage of automating the process of weight measurement is less energy and money being wasted on hiring human forces. To keep up the continuous monitoring of potentially large number of pigs it is more effective to use a live visual analytic technology instead. This is because it takes less time for the computers to calculate results. Also a visual based technology means that the animals will not have to be moved and strained for the manual weighing process.

The visual device used in this project is Kinect camera. In summary the camera on this device has three main sensors. One records colour data similar to regular cameras. Another analyses skeletal structure of moving objects within the field of view which is usually used for dealing with body gesture as an input data. Finally the last lens records the distance of each point within the view field from the camera itself. There is a minimum and maximum range for the distances being captured. The device saves the distances as depth map, with each pixel of the image frames representing a number within the accepted range. For this application the size of recorded depth stream for each frame is 640 × 480 pixels.

[1] http://www.ncl.ac.uk/afrd/

2.BACKGROUND/RELATED WORK

In this section first common industrial approaches to weight estimation using different image processing techniques are introduced. Afterwards the most compatible one with this project is chosen for application and explained why.

2.1 Pig Weight Estimation

In image processing algorithms are used to separate the segments constituting the target object within a frame. This is done through processing of quantifiable data (e.g. colour value pixels, depth value pixels, temperature pixels etc.). [4]

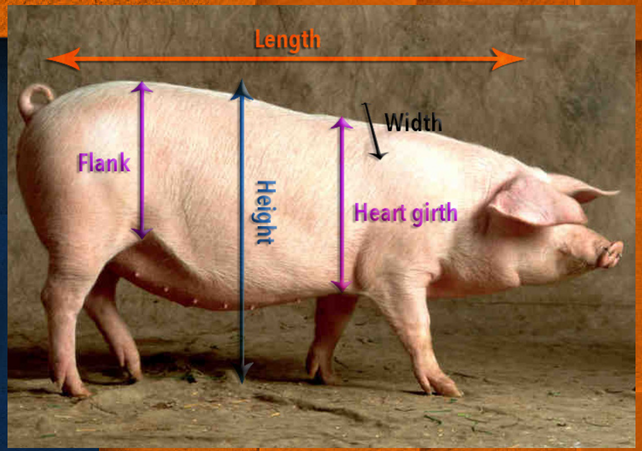


Figure 1: Example of 3D quantifiers effective in weight estimation [2]

2.1.1 Weight Estimation Approaches

3D Capture Approach: Pig's shape is an important aspect and plays a key role in weight estimation. By having 3D model of pig's body it's possible to make an accurate prediction of pig's weight due to having its volumetric measurements. Two of the common techniques for 3D shape attainment are laser range finding scanners and stereoscopic cameras. Lasers can be too slow to operate since pigs will be moving [2]. 3D cameras are good alternatives and as for Kinect devices they have the capability of creating 3D view of the recorded incidents using depth data. In this project however the samples are only recorded from one angle (i.e. top view) which makes it impossible to extract a full 3D model of pig's body.

[2] http://www.lrrd.org/lrrd26/5/walu26096.htm

2D Image Series: The second best approach after 3D capture is obtaining series of 2D images from different angles. To achieve the best result it's recommended to have at least two cameras, one positioned above the pig and one capturing a side view. Alternatively a mirror with 45 degree angle can be placed above the pig to give the side view camera both top and side perspectives as shown in figure 2. Both techniques in this approach resemble previous approach since they provide the same data that can be taken from 3D shots [7].

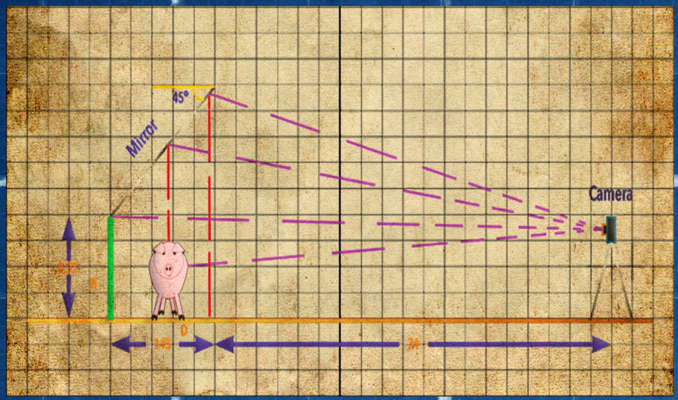


Figure 2: simultaneous top and side views with use of a mirror.

Having only a top view data available the approaches are limited to 2D measurements. In this situation the techniques to be used will not be as accurate as having volumetric data such as heart girth which is considered to be a reliable factor for accurate estimation of weight [8]. Figure 3 shows how the side dimension can play a role in constructing a geometrical model of a pig which can then help estimating weight by adding volume of the shapes.

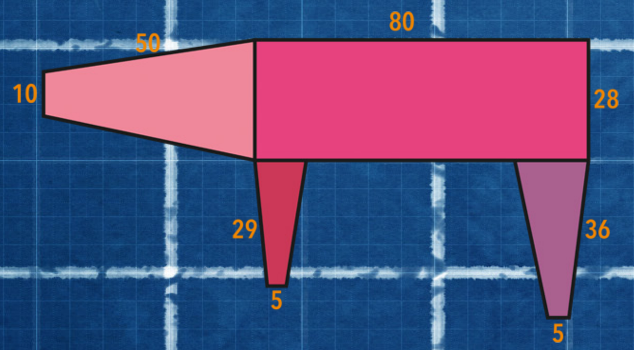


Figure 3: Model pig constructed from cones and a cylinder. Dimensions in cm were measured from the image of an 80 kg pig. [7]

2D Top-view Image: In this project so far this is the only applicable approach. With 2D images weight can be estimated through finding out correlations between the flat measurements (e.g. top surface area) and pig's actual body weight. There are several industry standard techniques that help the overall process of finding out these correlations. Plus there are systems that already use a few of the discovered correlations, highly related to the actual weight. The following are some common techniques and systems.

Generating difference image: this is a process of suppressing background and emphasizing foreground objects (i.e. pigs) in frames [9]. As well as making the foreground object salient this process makes height differences more significant and noticeable. So if calculations for 3D models were involved this would become very useful.

Binarization: This is the process of polarizing a frame into two regions. One representing pig regions and zero the background. Knowing which pixels of the frame are parts of pig's body, they can be counted to extract data such as pig's area. The way to do this is to know the real area reference for a pixel (i.e. how big each pixel is in reality usually in cm^2). [9] Alternatively the direct correlation between weight and number of pixels can be found to conclude the estimation.

Finding correlation(s): After obtaining different data sets from the video samples such as height, widths, length, heart girth etc. it is important to know how they relate to the live weight. In other words we want to find out if the actual weight is dependent on the variable candidate and if so by how much. There are various approaches to derive a formula based on a correlation which calculates the live weight. The simplest form of it is the linear regression interpretation which defines how much a single variable has influence on live weight. [8]

Live weight = A + B (predictor variable) ± E

In the simple linear regression formula presented above Live weight is the dependent. A is the live weight intercept which can be zero. B is the correlation coefficient (the slope or Pearson product moment correlation coefficient). E is the amount of error involved in the estimation. Predictor variable or the regressor can be any of data sets (e.g. length, width etc.) previously mentioned.

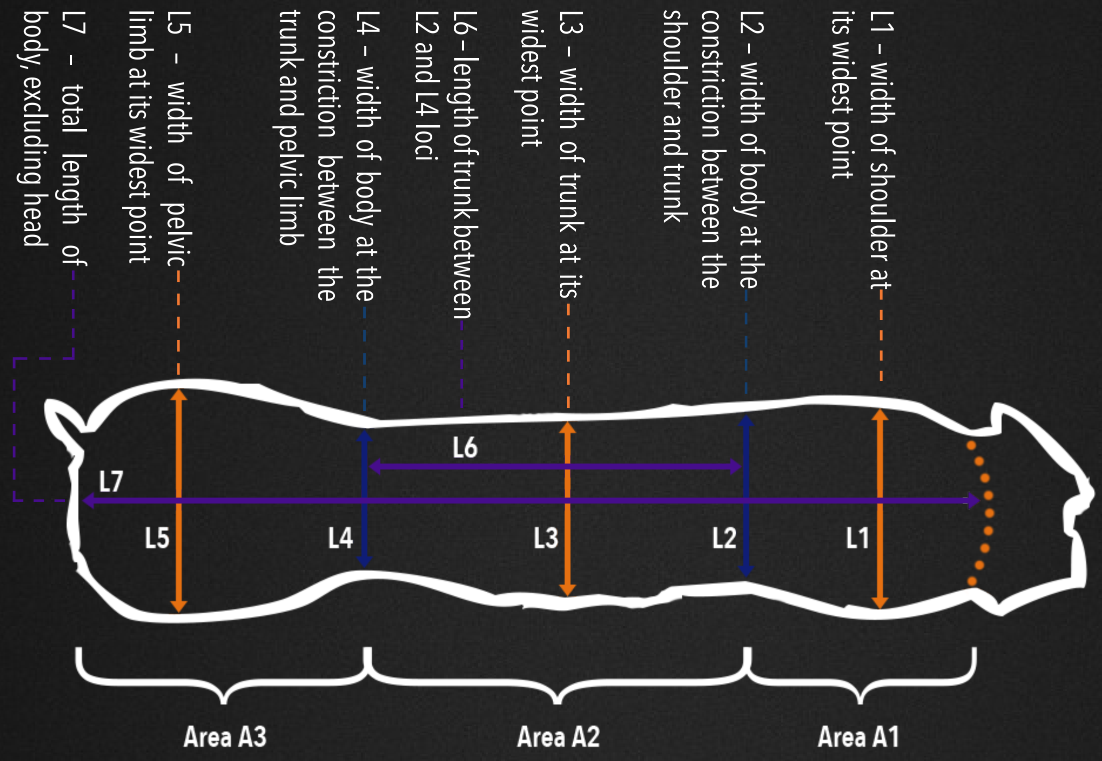
Finding coefficient of determination (COD): It is an indicator of how close the original data (e.g. actual weight) is to the prediction model (e.g. linear regression formula). For example using the heart girth predictor to achieve a linear regression model for live weight estimation, we will have linear diagram with real weight values plus a regression line (i.e. estimation values). Suppose in diagram's (x,y) data set y is the live weight and x is the heart girth then the coefficient of determination describes how much the real weight values are close to the estimation values which are based on heart girth. In other words it indicates how well data points on both diagrams overlap. The value of the coefficient ranges between 0 and 1 with 0 meaning no correlation between the dependent and the chosen predictor and 1 meaning the model describes (in this case estimates) the data 100% perfectly.

Finding multiple correlation(s): As mentioned earlier to formulate correlation we need to have our dependent variable weight and a predictor (e.g. heart girth). After finding coefficients of determination for all the possible predictors, the ones with the biggest CODs can be chosen for finding multiple correlation. This makes the overall estimation very close to the actual value. A linear regression model in this case is not much different to the simple linear model which is based on single predictor. The main difference is that instead of one predictor multiple predictors with their relative correlation coefficients are used for deriving the formula. As **for the coefficient of determination for each of the predictors, it is also calculated relative to all other predictors being used in formula.**

Qscan System

Qscan is a standard visual image analysis system which introduces a structured approach to pig weight estimation [5][6]. This approach also provides a good solution for 2D image processing. The relationship between area of the pig's back and body weight according Qscan is as shown on figure 4.

Figure4: Image shows width of different body sections, body length without head and the area measurements defined by the visual analysis system.



Total Area A4 = A1+A2+A3

A1-plan area of the shoulder, proximal to L2 locus	A2 - plan area of the trunk between L2 and L4 area	A3 - plan area of the pelvic limb, distal to the L4 locus	A4 – total plan area of body, excluding head

2.1.2 Weight Estimation Errors

With the background research information designing an effective solution sounds promising. However due to limiting data and simplicity of the technology engaged decent level error is expected. One of the challenges risen by the errors can be inability of detecting and addressing them. These errors can depend on different factors like pig's posture/stance, walking speed, movement angle, inappropriate lighting and many others. Some of these factors can contribute to significant level of error especially those directly affecting the segment measurements and visual analysis such as the head not being in a straight line as the back. As it will be seen in implementation section some of these problems can be solved such as the curved back. Another type of issue with errors is that working with live biological entities such as pigs, the data can be unstable due to visual variety in sample frames. The reason for this variety can be that animals unlike solid objects can deform, they can have different stance or posture at each moment which can be incompatible to application's standards (e.g. sitting position) etc. Due to instability of data error will always be present in the process. However certain strategies like finding average of samples plus an increase in their quantity measurement accuracy would improve statistically. [4][5]

3.DESIGN AND IMPLEMENTATION

In this section application's design & structure are broken down and explained. This also talks about the role of each section of the program and how it serves the overall processing of frames. The process of analysing each frame received by the program has been divided into three main stages of pre-processes, mid-processes and post-processes. The following subsections are about each stage and what they contain.

3.1 Pre-processes

Pre-processes are the series of operation that prepare frame data for further advanced and logical analysis. This stage provides simple tools that are based on mathematical and geometrical operations. This stage can be thought of as an engine or core for image processing. The first step for the application is to read frames (640 × 480 for this project) from the video data. After reading the samples pre-processes perform the tasks described in the upcoming subsections. The first few subsections mainly present the tools designed as the basis of operations in this stage. Followed by that in the implementation section it is explained how the designed tools are put together to perform the pre-processes (i.e. preparing the frame by transforming it to a desired state).

3.1.1 Grid, pixel data structure and lines

Frames in the application as mentioned before are 640 × 480 pixel area. However the raw depth data in each frame is extracted as straight line streams. The displaying of the streams is done by starting from top left corner of screen moving to right and row by row going down till it reaches the last value being placed at the bottom right corner pixel. In order to have a more tangible and straightforward system to deal with the frames it's better to move streams into a grid system. This way each pixel can be reached by its row and column. Lines are series of pixels that connect two pixels to each other. A common algorithm that calculates an approximation for grid based straight line between two pixels (i.e. two points) is Bresenham's line algorithm which is the same algorithm as many other computer applications (e.g. Adobe Photoshop). The function for drawing line as well as the method for getting the list of points on a line segment both take 2 points as an input and are modified to only consider the points that are within the frame. Having a drawing function can help visualize processes to test them against error and to see the results.

3.1.2 Frame modifiers

Setting frame rate: this function decreases the number of frames loaded into the buffer per second when application is performing expensive calculations in real-time. It takes a divisor as an input and divides the original frame rate (30 fps for this project) by it.
Range Selection: To separate target objects from an image the background and foreground need to be defined. In case of depth data the range containing distances that are considered ground, too close to the camera or known to not include the target(s) are set to zero to pin-point regions that need to be processed.
Region Selection: Frame regions known to not include the target(s) are removed to speed up performance.
Noise Cancellation: this operation removes the noise spots which can look like a small hole in a shape due to camera error to a random spot looking like an object or an unnecessary gap between edges of a shape.
Edge Smoothing: this operation makes the edges of shapes smooth for more accurate analysis.

3.1.3 Geometrical operations

These are foundation tools that are used in procedures such as shape recognition, area/volume calculation and object orientation determination.
LineAngle: this function given two points it calculates the angle between the line connecting them and the x axis.
Point to Line Position: this function given a point and a line finds out which side of the line the point is located.
Euclidean distance: finds the Euclidean distance between two points.
Closest point from a line to a point: given a line and a point this function finds the intersection point of the line and another line perpendicular to it which goes through the point.
Point-Line distance: this function finds the distance of a point from a line.
Cross product: calculating cross product of two vectors (i.e. segments)
Dot product: finding the dot product of two vectors (e.g. AB and BC)

3.1.4 Objects & boundaries

An object in a frame is collection of pixels that are connected and their value falls in a pre-defined range. Any two pixels taken from an object will either be neighbours or there is at least one path made of continuous series of pixels connecting the two pixels and the series of pixels also have values that fall in the same pre-defined range the two pixels. In order to detect an object within a frame a technique is required to do **connected component labelling** (CCL) [11]. CCL depending on the technique chosen has a certain way of identifying connected chunks of pixel and giving them a label. The chunks that happen to have the same label are parts of one object.

Main boundaries of an object also known as axis aligned boundaries (AAB) are first and last (i.e. lowest and highest) rows and columns that contain the entire object. The orange rectangular borders in figure 5 is an axis aligned bounding box (AABB) which is essentially a rectangle created out of object's main boundaries intersecting.

There is however another type of boundary which has more applications for image processing and that's the oriented minimum bounding box (OMBB) [13]. It's the smallest rectangle possible drawn around an object that contains it.

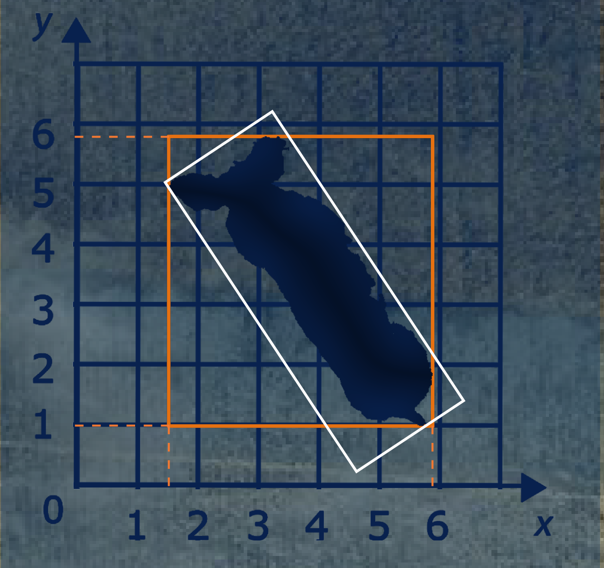


Figure5: Pig shot taken from the application. Axis aligned boundaries are shown with orange lines and OMBB in white.

One of the OMBB's advantage over AAB is showing the general orientation (facing direction) of the pig useful for identifying measurments (e.g. length & width). To generate OMBB we need another asset called convex hull. It is the minimum number of points selected from the object that if connected will enclose the entire object. It is not to be confused with object's outline.

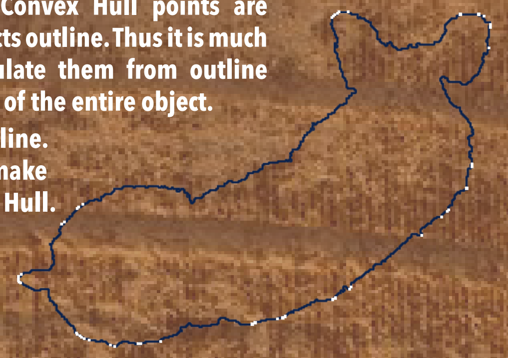


Figure6: The Convex Hull points are subset of objects outline. Thus it is much faster to calculate them from outline points instead of the entire object. Blue is the outline. White points make up the Convex Hull.

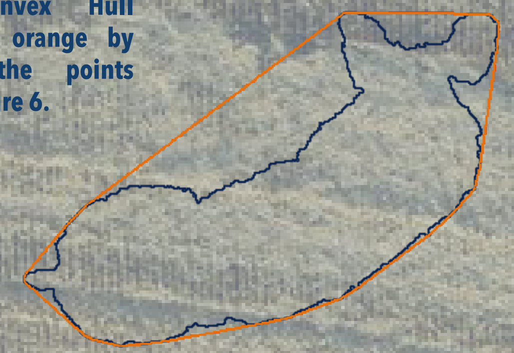


Figure7: Convex Hull is drawn in orange by connecting the points shown on Figure 6.

3.1.5 Implementation

The following are the procedures taking place at pre processesing stage of the application. In this flow chart 2 processes in same row means they are simultaneous. The advantage of CCL technique used is that the frames are scanned through only once. The rest of processes are independent of the frames. Because of this advantage the design is highly efficient for real time applications.

Range Selection

Connected Component Labeling:
Scanning through the pixels row by row, all the pixel line segments containing an object are stored in a tree data structure. As a new line segment is found it will be checked to see if it has any previously stored neighbour[1] segments. If not it will stroed as a new root node. If it only has one neighbour it will be directly connected as a child node to the root of that tree. If it has more than one neighbour it will be checked to see which one is the closest to a root node saving that root as the destination. Once figured out the line segment itself and all of the segments belonging to its other neighbour trees are set as children of destination root. After stroing all of the line segments in a frame in the described fashion. Each root node uniquely identifies an object.

Joining line segments:
Following CCL line segments in each tree are joined to have final object.

+

Outline extraction:
The advantage of using CCL technique introduced by this research is that as long as a labeled object does not have holes inside it the start and end points of the line segments during the joining process can be used to make up the outline.

Convex Hull Calculation:
With object's outline convex hull points can be found using any of the available algorithms. Gift wrapping [12] is the applied algorithm in this project because of its speed.

Finding OMBB:
First the convex hull (CH) points need to be connected clockwise or antiCW to find the longest segment amoung all pairs connected (figure 8 shows an example). On figure 9 AB is the longest segment. The rest of the points are found in alphabetic order based on AB and the rest of the CH points to create OMBB. The process of finding each point is as follows. C is the furthest point from line AB. D is the closest point to C on AB. E is the furthest point from CD. F is the closest point to E on AB. G is the furthest point from EF. H is the closest point to G on AB. I is the closest point to C on GH. J is the closest point C on EF.

Figure 8: The longest segment on Convex Hull. The image is a captured depth map

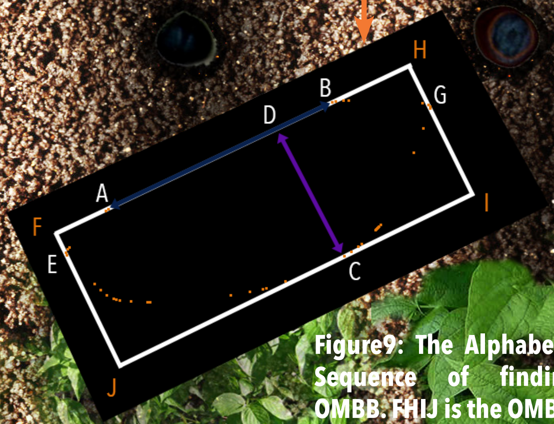


Figure9: The Alphabetic Sequence of finding OMBB. FHIJ is the OMBB.

[1] Any two segment that have overlaping column range are neighbours.

3.2 Mid-processes

This stage can be thought of as an application built upon the engine (i.e. Pre-processes). Thus Mid-processes are set of operations that use combination of different tools provided by the previous stage to extract data according to a given specification. This can include tasks such as feature detection/recognition (e.g. pig's head) and measurements (e.g. width of pelvis). First step at this stage is to define the specification. In other words what type of data is required from the frames. The strategy to retrieve the specified data is done by first rejecting what is unwanted through setting filters and then by defining how to look for the right data through designing a scanning method.
The raw data produced at this stage is used for finding the originally required data and further statistical analysis at the next stage.

Required Data: Detecting pig's head and tail. Measuring its length and the widths for shoulder, trunk, pelvis and the points where these segments meet.

Specifications: Head starts from ear's top and ends at base of the neck. Tail starts at tail top and ends at tail base. Length starts from tail base ends at neck base.

3.2.1 Filters

Filters are the logical tests using different rules to define whether a frame meets the required criteria for further analysis. It can be thought of as specification for an acceptable frame. If a frame fails a filter test at any stage not only the processing of the frame is stopped from going further but also all the progress made on this frame is dumped.

The main reasons for having filter is one it saves a lot of unnecessary computations. Two it correctly eliminates frames that hold no useful information. Finally it separates the data that can add inconsistencies to the overall averages and other statistical analysis carried out by the next stage. The filters can appear at any step of this stage. There are three basic filters used in this application.

Size filter (defined by number of pixels): as implied by the name this filter eliminates objects that are out of certain size range because the smallest pig is never going to be smaller than a particular size and the largest pig is never going to exceed a certain size.

Ratio filter: from an infant pig to a large adult pig the length is always going to be at least 2 times bigger than the width. Using OMBB this is a simple test yet effective for setting aside other objects with different ratios.

Presence filter: this filter checks whether the target object is fully present in the frame. It is done by checking if axis aligned boundaries are touching the sides of the frame.

In this application if a frame passes all the filter tests a green light appears on screen meaning the current frame is accepted.

3.2.2 Scanning

To scan pig's body for finding a point/points with certain specification the OMBB is used as the tool in this application. The way this scanning works is by starting from one of the width sides of the box moving segment by segment all the way to the mirroring side. While on a segment each of its points is checked one by one from one end to the other to see if the current point is part of the pig's body. Segments are picked one by one based on pairs of mirroring points across the box's length sides which make up the two ends of the segments. The first pair of points are always two corners of the OMBB. Each of the chosen segments has to be parallel to the width of the box. This is similar to how scanner devices take photos of objects line by line. Figure 10 is a good representation of how this process works.

Mirror test: As mentioned to pick segments we need two points & the starting pair are always two corners of the OMBB. However to know that the corners are chosen correctly from one side and not from the opposite sides of the box we need to check if they are mirroring each other. This test measures the distance between the initial points to see if it approximates to OMBB's Width.

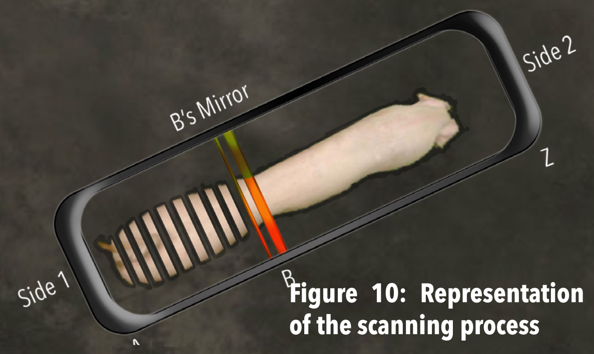


Figure 10: Representation of the scanning process

Solution to curvy stance: When pig's length from head to tail is almost on a straight line scans are going to result in accurate data. However if the pig's back is bent to the sides as in figure 8 using an OMBB will produce inaccurate results. Because of the misalignment between the length of OMBB and the pig's body the mirroring points at both ends of the picked segments are not going to represent the actual symmetry of pig's anatomy. To solve this problem the pig's length is divided in half and OMBB is calculated for each half separately. This increases the accuracy back to a standard level.

Distinguishing between head and tail: The number of points with a range close to maximum height are more on the tail half compared to the head half.

Finding neck's base: To find neck points a region close to head known to contain neck is chosen (The 2nd and 3rd quarters of OMBB from head half). Within the region the closest points from each sides of the length are the base of neck.

Finding tail base points: The tail width grows with a certain rate from its top to its base. This rate changes as the pelvis starts. The point where the width growth rate changes is the base of the tail.

Removing head tail regions: Head and tail regions can be removed once their bases are found. The remaining is the area suggested by Qscan system (i.e. A4) to be effective for weight estimation.

3.3 Post-processes

Post-processes are the operations calculating final results based on the raw data extracted from previous stage such as basic measurements. They also derive statistical conclusions, for example average of pig's weight based on series of estimations. This stage is where weight is estimated using the area measurements gathered. Tables and graphs can be generated as a presenter of results.

Weight - Area Correlation: As concluded from background research pig's projected area (excluding head and tail) has a direct relationship with its weight. First we need to explore this first-hand. Therefore the next step is to find the pig's projected area.the OMBB.

Alternative approach to area calculation: Having gathered all the parameters required to find A1, A2 and A3 from introduced by qscan A4 which is the sum the three can be found. Instead of trying to find A4 based on A1, A2 and A3 the simpler way is to count the number of pixels remained after exclusion of head and tail this way using the distance reference for a pixel A4 can be calculated in mm^2.

Distance reference: The distance reference (equivalent in reality) for the length of floor in the frames is 221cm (2210mm). Length of row in the frame is 640 pixels. Therefore a pixel is 2210 divided by 640 millimetres which is roughly 3.453125 mm and 11.924072265 mm^2. Using the distance reference and the number of pixels within pig's area we can find pig's area. Pig's area = 11.9240 * number of pixels

Measuring in pixels: It is not necessary to find the real area of the pig's projected area in order to find the weight correlation. Number of pixels enough can be representative of the area. Using linear correlation we can find and prove how much influence the area found has on weight. The next step after this is linear regression interpretation which derives a formula based on sample data that can estimate pig's weight close enough to the real weight. This is exactly the final goal of the entire project. In the application's post- processing stage the average of projected area excluding head and tail for each video sample is calculated and the results are recorded into an excel spreadsheet. Then Excel finds the correlation coefficient and the linear regression interpretation (i.e. the formula for finding weight). Once the formula for the weight is found it can be implement into the application to automatically weigh the pigs for each frame.

4.RESULTS AND EVALUATION

In this section overall achievement of each stage is shown including the challenges overcame to maintain real-time response. Next the simple alternative approach for finding the pig's projected area is pointed out. Finally some statistical data are presented at the end to justify the accuracy and value of this research and the application produced as the final product of this project.

Real-time application requires algorithms calculating accurate results and fast enough to respond before the next frame is called.

Worst case scenario for a real-time application is being stuck in a situation where frames are being loaded faster than being processed and the application cannot be optimized in any way to catch up with processing. In this case we can use set frame rate to drop frames. Although is solved but this should always be sought as a last resort option because dropping frames means losing significant amount of potentially useful data which could have increased the overall accuracy in many cases.

To support run-time speed application's main focus was on architecture's design.

Basic geometric operations & tools provided were straight forward to implement. The reason being is that most of these have only one way of being implemented and if there are possible alternations results would mostly be as fast or accurate.

Object detection has high influence on run-time speed. As a result the challenge was to limit the number of iterations through each frame down to one. Most common approaches found online suggested ways that could not fulfil this so I had to design my own.

Instead of finding convex hull using all of points making up pig's body, the outline points are used to significantly decrease number of pixels involved in calculations & in turn speeding up the process.

Finally the filters used at any point can stop the application from further processing frames not containing useful information.

Finding the average projected area (i.e. A4) directly through counting pixels instead of measuring local minimum & maximums for A1, A2, A3 not only needs less calculations but also is less susceptible to error.

Finding A4 of some of the samples included lots of errors such as wrong neck detection wrong head/tail elimination etc. The main reason for this is that in most of the videos the pigs are moving with an undesired speed instead of being steady or walking slowly. Running makes the pig's shape distorted which results in errors or inaccuracy. Another reason for the errors present beside running state is lack of strong shape recognition algorithm. Pig's head can be very diverse in shape ears can change direction and head can bend in different direction causing instability and hence inaccuracy.

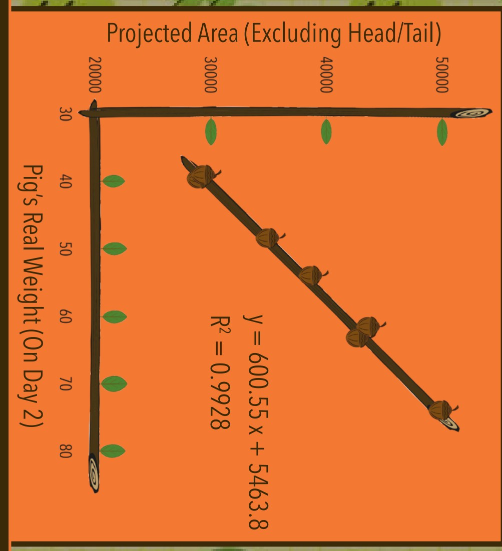
Also the larger the pig the larger its features therefore the more accurate the data.

Another factor that compromises the results accuracy in the samples is that pig's appearance in the frame is mostly for a very limited a time so there is not enough data for a robust result.

Statistical Results:

	Day 2		Day 3	
	Weight	A4	Weight	A4
Pig 1	61.6	43390.86	59.4	42847.2
Pig 2	54	38579.5	54.4	38772.5
Pig 3	39.2	29026.11	39.6	29985.29
Pig 4	39.6	28663.89	40.2	28961.67
Pig 5	63.2	42520.93	63.4	43425.3
Pig 6	48.4	34675.14	47.6	34665.2
Pig 7	74	49600		
Pig 8		37909.43	56.2	38286.33

As mentioned the video data used in this project were recorded in three different days. To prove that projected area, A4, of a pig has strong relation with the pig's body weight first the average A4 of each pig was calculated from a day's samples. Then average of all for a day is calculated. The following table has the average area of all pigs for an entire day in one column and their real weight measured on the same day in another column.



The graph above is based on day 2 data. R^2 for both days is 0.9 which indicates that there is a strong correlation between pig's projected area and its bodyweight. Using the same graphs the linear regression interpretation has given us the formula for estimating pig's weight.
y=600.55x+5463.8(for day 2)
y=611.43x+5191.4(for day 3)
since the correlation for day 2 is stronger than day 3 (i.e. 0.9928> 0.9753) the weight estimation formula is generated based on day 2 to have higher accuracy.
Estimated Weight=(A4-5463.8)/600.55

	Day 2		Day 3	
	Estimated Weight	Estimated Weight minus Real Weight	Estimated Weight	Estimated Weight minus Real Weight
Pig 1	63.15	1.55	62.24	2.84
Pig 2	55.14	1.14	55.46	1.06
Pig 3	39.23	0.03	40.83	1.23
Pig 4	38.63	-0.96	39.12	-1.07
Pig 5	61.7	-1.49	63.21	-0.18
Pig 6	48.64	0.24	48.62	1.02
Pig 7	73.49	-0.5		
Pig 8	54.02		54.65	-1.54

Table above is comparison of averages of estimations against real weights for finding error level. The biggest absolute value from error column (i.e. difference) is 2.848605 which is roughly 3 KG. So the application can estimate weight with SE = ± 3 KG. SE stands for standard error level. The statistical results express that the project has indeed been successful. Application can be twicked to reduce the error level but the improvements will not be significant because the the data is 2D based which will never be as accurate as 3D data analysis.

5.CONCLUSIONS

In conclusion the most valuable and advantageous aspects of the entire application are the three features mentioned below.

1- the simplicity and its high accuracy for creating results

2- Its adaptability to unusual situations like when pig's back is bent to the sides

3- The fact that it operates real-time

A primary importance of this research is that it can provide good guidelines for image processing in real-time. Future researches based on this can explore different aspects of the project as whole or focus on improving individual sections of it. Some examples of subjects for improvements are pointed out here.

Tracking system (using frame buffering): Certain information found for a frame can be carried to the upcoming frames instead of applying all of the procedures to every single frame. One of such knowledge can refer to the direction of the pig. A pig will not be able to turn 360 degree during one frame so knowing where the head and tail are for one frame it can be assumed that the pig's direction in the next frame is a slight variation of previous frame.

Smarter object detection: This means including factors other than just boundaries because as it can be seen in some of the samples once two object's boundaries collide with each other both are considered one. Using depth data, the curve on a pig's back can be recognised based on a rise and fall pattern in height.

Smarter & more frequent filters: as seen earlier filters are an important part of solving confusions between non-pig and pigs objects plus they drop a lot of unnecessary processing.

Smarter shape recognition: Neural network image processing is one example of shape recognition technology particularly useful for its ability to learn patterns based on references.

6.REFERENCES

[1] R. Harrell, C. Rusk, B. Richert. *Identification of Swine by Auricular Vein Patterns*. Nuclear Medicine and PET/CT, University of St. Francis and Indiana Purdue Fort Wayne, Fort Wayne, Indiana Department of Youth Development and Agricultural Education, Purdue University, West Lafayette, Indiana Swine Nutrition and Management, Purdue University, West Lafayette, Indiana, 2008

[2] R. Tillett1, N. McFarlane1, J. Wu1, C. Schofield, X. Ju, J. Siebert. *Extracting Morphological Data from 3D Images of Pigs*. Silsoe Research Institute, Wrest Park, Silsoe, Beds, MK45 4HS, U.K. Dept. of Computing Science, University of Glasgow, Glasgow, G12 8NN, U.K., 2004

[3] M. Kashiha, C. Bahr, S. Ott, C. Moons, T. Niewold, F. Ödberg, D. Berckmans. *Automatic Identification of Marked Pigs in a Pen Using Image Pattern Recognition*. Elsevier, Amsterdam, Netherlands, Apr-2013

[4] A. Frost, C. Schofield, S. Beaulah, T. Mottram, J. Lines, C. Wathes. *A review of livestock monitoring and the need for integrated systems*. Silsoe Research Instirute, Wrest Park, Shoe, Bedford, MK45 4HHs, UK, 1997

[5] J. Love. *Video Imaging for Real-Time Performance Monitoring of Growing-Finishing Pigs*. The University of Guelph, Guelph, Ontario, Canada, Aug-2012

[6] T. Banhazi, M. Tscharke. *Improved image analysis based system to reliably predict live weight of pigs on farm preliminary results*. National Centre for Engineering in Agriculture, University of Southern Queensland, Toowoomba Queensland, Jun-2011

[7] C. Schofield. *Evaluation of Image Analysis as a Means of Estimating the Weight of Pigs*. APRC Institute of Engineering Research, Wrest Park, Silsoe, Bedford MK4.5 4HS, UK, 1990

[8] B. Valentino, S. Paola, M. Raffaele, C. Claudio, S. Alberto. *Predicting Liveweight From Body Measures In Nero Di Parma Pigs*.

[9] Y. Yang, G. Teng. *Estimating pig weight from 2D images*. Department of Agricultural and Bioenvironmental Engineering, China Agricultural University, Beijing, 2008.

[10] C.SchoÞeld, J. Marchant, R. White, N. Brandl, M. Wilson. *Monitoring Pig Growth using a Prototype Imaging System*.Silsoe Research Institute, Wrest Park, Silsoe, Bedford MK45 4HS, UK, Danish Institute of Agricultural Sciences, Research Centre Foulum, P.O. Box 50, DK-8830 Tjele, Denmark, PIC Group Ltd, FyÞeld Wick, Kingston Bagpuize, Abingdon, Oxford OX13 5NA, UK, Jul-1998

[11] R. Walczyk, A. Armitage, T.D. Binnie. *Comparative Study on Connected Component Labeling Algorithms for Embedded Video Processing Systems*, School of Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, UK School of Computing, Edinburgh Napier University, Edinburgh, UK, 2010

[12] M.T. Goodrich, R. Tamassia. *Algorithm Design*, Wiley, 2001 ISBN-10: 0471383651 ISBN-13: 978-0471383659

[13] H. Freeman, R. Shapira. *Determining the minimum-area encasing rectangle for an arbitrary closed curve*.Communications of the ACM, 18 Issue 7, July 1975, Pages 409-413

Newcastle University

Agriculture & Computing Schools