Learn to Play like Minnesota Fats: Augmented Reality in the Pool Hall

Deirdre Dalton[1] and Lorcan Coyle[2]

[1] School of Computer Science and Informatics
UCD Dublin, Ireland
deirdredalton1@gmail.com
[2] Lero, the Irish Software Engineering Research Centre
University of Limerick, IE
{Lorcan.Coyle}@lero.ie

Abstract. Detecting human activities from sensors deployed in the environment is a difficult problem. Human activities are so varied that it is difficult to match sensed activities to observed actions. Instrumented homes have proved a popular platform for such sensors but experiences there show how challenging it can be. We believe that the pool hall is a more useful platform for the deployment of body-mounted sensor networks as the motions that can be captured correlate well to the activities that take place on the pool table. The problem is more narrow than detecting human activities in general. We will focus on a simple sensor, mounted on the cue, which is important as the cue is the sole actuator used to play the game. By correlating the force with which a player strikes the ball with their cue and the distance the ball travels we demonstrate that with a simple off-the-shelf sensor we can derive useful information. We conclude by describing a set of enhancements that enable a computer to "sense" the game of pool.

1 Introduction

Pool (and its variants Snooker, and Carom Billiards) are popular sports at amateur and professional level. At its most simple level, two players take turns to strike the cue-ball with a cue in order to strike other balls to pot them in some sequence[3]. There is a lot of variables for the player to consider in each strike, from the angle of the cue to the object ball, to the strike-point on the object ball, to the strength of the strike. The shot ends when all balls on the table stop moving. By aiming a camera at the table and measuring the velocities and positions of each of the balls will be possible to assess the outcome of each shot. Assuming a good model of the physics of the cue, balls, and table can be captured it will be possible to measure with some accuracy the precise type of shot that was made[4].

[3] Carom Billiards is a little more subtle... for a start, there are no pockets
[4] We will question whether this assumption can be relied upon in Section 5

We have instrumented a pool table with a camera that points downwards from above to detect the position of the balls and a cue with a 3-axis accelerometer to detect the type of shot taken. Section 3 describes our set-up in more detail. The rest of the paper is arranged as follows: We describe some prior research that has inspired our work in Section 2. In Section 4 we present some preliminary results, and in Section 5 we discuss some possible directions this research could take and some of its applications. Finally we conclude the paper in Section 6.

2 Background

This project has taken much inspiration from research in augmented reality, pool trainers, and the physics of pool itself. There has been much work on the areas of robotic pool playing [5, 4, 1] and some success in building GUIs for simulating pool [6, 2].

Gao & Mac Namee developed SE3DB, vision system for snooker, as a viewer aid in televised snooker broadcasting [6]. Their inspirations were the various software systems used in sports broadcasting such as FoxTrax Hockey Puck Tracking System, PISTE and the Hawk-eye system. PISTE minimizes the usage of sensors or cameras, so as to reduce interference, and create content at almost the same time as the event is carried out. Hawk-eye is now used officially in the Wimbledon tennis tournament, where players can appeal a call by a line judge and use Hawk-eye to determine if the ball was in or out. The primary purpose of SE3DB was to determine whether a particular shot is possible or not.

Guo & Mac Namee address many of the challenges that arise when interpreting images of the game of pool such as determining the boundaries of the table, the ball and pocket positions, and the colours of the balls. They apply edge detection using gradient operators, Hough transforms and canny edge detection to recognise and locate the balls on the table. Hough transforms may be used for circle and line detection, and thresholding is used to identify shapes by assigning, for example, black to colours below threshold and white to those above. Contour flowing may be used to detect any shaped objects. Figure 1 shows, on the left the original image, and on the right the image after image analysis has identified the centroids of the balls. Guo & Mac Namee's research highlights the utility of computer vision and demonstrates the accuracy with which vision-based software can position a pool ball on the table.

Chan et al. developed the Pool Table Trainer at Villanova University, Pennsylvania in 2009 [2]. Parallels between the Pool Table Trainer and this project are clear. The goal of their project was to develop a pool table trainer to calculate the various types of shots a player can make and different stages of a game of pool. They used MATLAB to calculate the trajectory and distance the ball travels. A camera was mounted above a pool table and the physical environment was translated into a GUI. Using some interaction from the user (i.e. select ball and pocket) the best shot was calculated and displayed on screen. They used a thresholding type technique where all pixels belonging to the pool table were

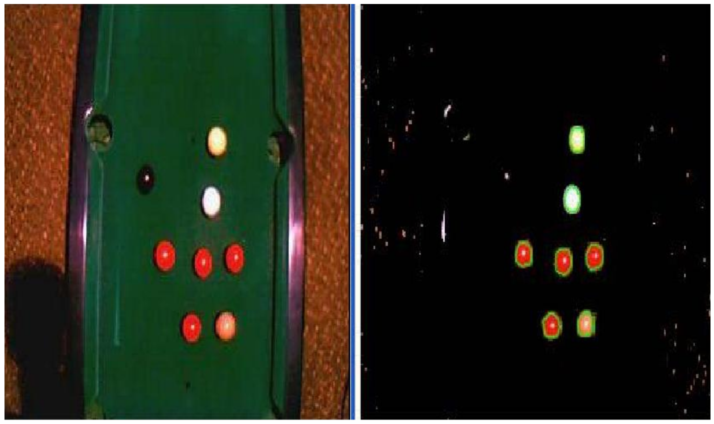

Fig. 1. Original Image and Image after Contour Flowing worked on original image, Guo & Mac Namee [6]

assigned a black colour and pixels associated with the balls, a white colour. A pixel was associated with a ball it if fell within a particular RGB range. Figure 2 is a screengrab of the GUI interface they developed.

Larsen & Brøndsted presented a pool training system called the Automatic Pool Trainer [7]. The aim of their system is to automate the learning process of a game of pool. It describes how a camera, laser and microphone will be used to interact with the player to improve their performance. The Automatic Pool Trainer is an automatic system implementing the Target Pool training scheme. The idea of Target Pool is simple: to present the trainee with a number of exercises (accompanied with detailed instructions), and record and evaluate his or hers performance after each shot. The use of a camera, laser (similar to using a projector), and microphone results in a similar set of problems that we present — best shot calculation and image processing — with the added difficulty of voice recognition. It should be highlighted that Target Pool is strictly a training programme, focussing on improving a player's skills by training them on certain shots and tasks. Figure 3 shows the equipment and set-up needed for the automatic pool trainer.

Rod Cross has presented research into the physics of pool [3]. His work describes 'squirt', which is an unintended deflection of the cue ball if the player employs too much side spin on the ball. He demonstrates how a small change in the angle at which the ball is struck can cause a mis-hit leading to a completely inaccurate shot. His research highlights the importance of correct usage of the cue — Figure 4 shows the angle a ball can squirt at. Since the angle at which a ball is struck can be measured, and since squirt is a common and misunderstood phenomenon common to novice players it would be very useful for a pool tutor to identify its occurrence and offer improvements.

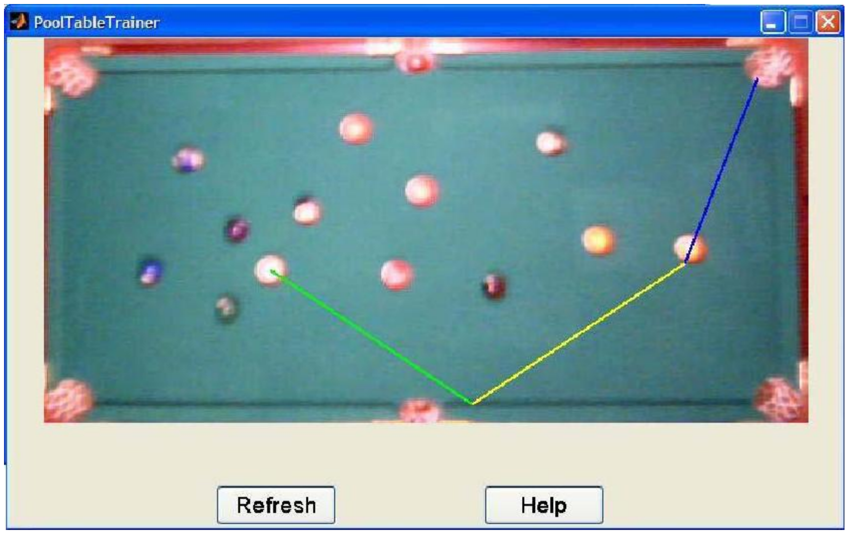

Fig. 2. GUI determining if bank shot is available (image taken from Chan et al. [2])

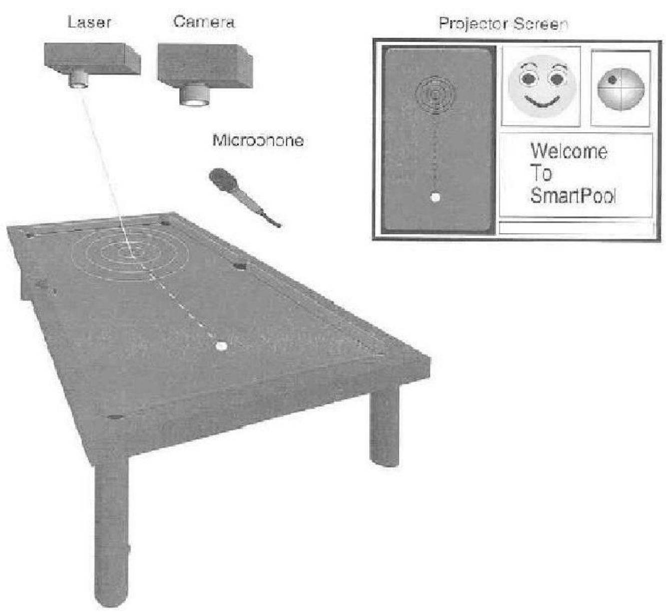

Fig. 3. The Automatic Pool Training System uses a camera and a laser to interact directly with the pool table: (image taken from Larsen & Brøndsted [7])

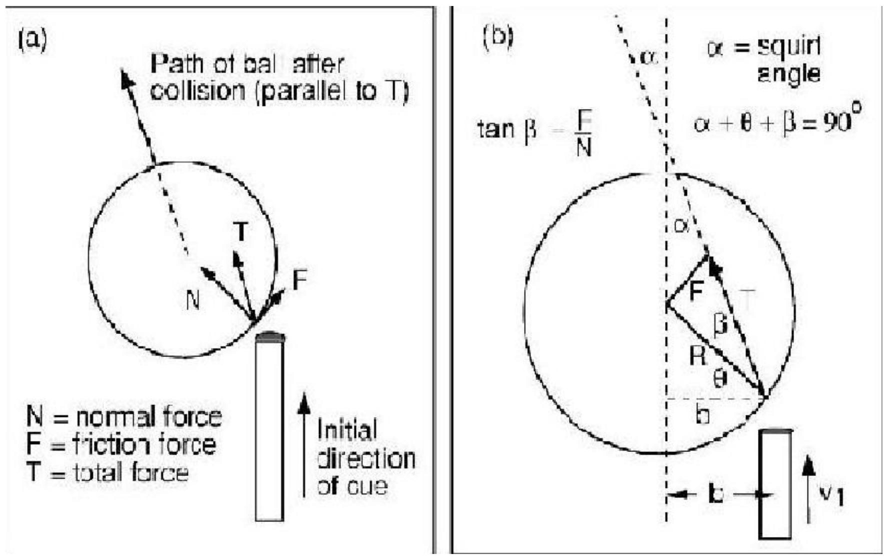

Fig. 4. View of ball and cue looking down onto the table. The ball is deflected by an angle α away from the line of approach of the cue (image take from Cross [3]).

3 Implementation

In order to interpret information from the pool table, it was important to choose suitable equipment. We took inspiration from Gao & Mac Namee and Chan et al.'s physical setup and purchased a wooden pool table and camera to locate the position of the balls on the table. We used an inexpensive Network Camera with image resolution of 320X240 and fed the images to MATLAB to locate the position of each ball. The core difference between our work and that of others is that we attached a sensor with a 3D accelerometer to the base of the cue to capture the forces at work as each shot was taken. Since the cue is in contact with a ball for a very short period of time during the shot the sensor must be able to react to sudden and small changes. We used Intel's SHIMMER (Sensing Health with Intelligence, Modularity, Mobility, and Experimental Reusability) sensor platform to capture and record the accelerations observed on the cue. Each SHIMMER contains a MicroSDcard slot (currently up to 2G Bytes of flash storage), a rechargable lithium-polymer battery, a 3-Axis accelerometer, and a 3-Axis gyroscope. We used BioMOBIUS[5], developed by the TRIL Centre[6] to interpret the accelerations captured by the SHIMMER and translate this into a prediction of the distance travelled by the pool ball.

Figure 5 shows a flow diagram of the information processed by the hardware and software components. The camera was mounted directly above the pool

[5] http://www.biomobius.org/
[6] http://www.trilcentre.org/

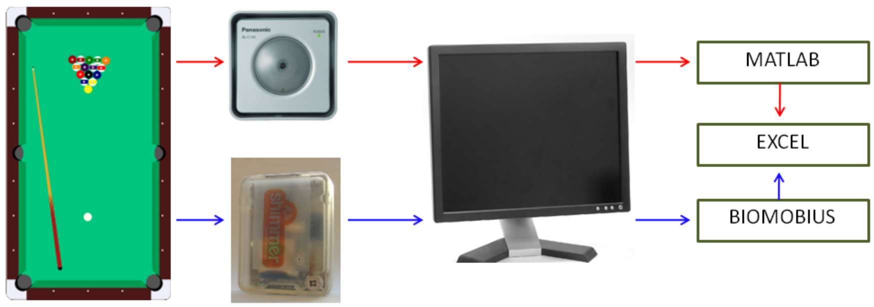

Fig. 5. Flow diagram of the interaction of our system's components

table to record and take images of the table and ball positions. The SHIMMER was strapped to the base of the cue with the y-axis aligned along the cue. This, therefore, was the axis of most interest as this is the axis through which the cue imparts force on the cue-ball. Recorded data from the camera was imported to MATLAB for analysis. A MATLAB algorithm was developed to determine the co-ordinates of the centre of each ball, edge detection and distance the ball travelled. BioMOBIUS was then used to collect the data from the SHIMMER and save it to a text file for later analysis by Microsoft EXCEL. Figure 6 shows a sample plotting of the accelerations detected in each of the axes as the cue is shaken.

4 Experimentation

In order to test the the ability to capture and interpret data gathered from the pool cue we attempted to find a correlation between the accelerations observed by the SHIMMER with the distance travelled by a struck ball. Our goal was to gather enough data to model this correlation with a general formula that could take an observed acceleration and predict the distance the struck ball would travel. In order to measure the distance a struck ball travelled we examined two images captured by the fixed overhead camera — one before any ball motion, and the other after the struck ball had ceased to travel. We made the assumption that the struck ball would not hit a cushion or any other obstacle, and so were able to calculate distance as the straight-line Euclidean distance between the before and after centroids. From experimentation we determined that in images taken with our fixed-mounted camera 1 pixel $\approx$ 0.61cm.

In order to capture a data point the camera was set to record and the BIOMOBIUS software commenced saving data to a text file. The cue was steadied and a smooth shot taken, striking a single ball along the table. Our assumption was that the peak acceleration would occur as the cue struck the ball. We

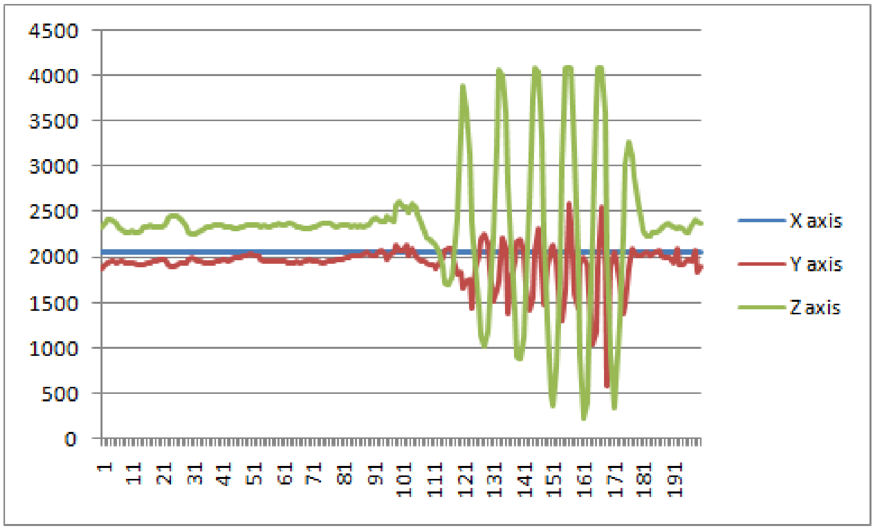

Fig. 6. This plot shows the changes in acceleration as the cue is wobbled by the user. There are relatively large accelerations on z axis as the cue was shook in a mainly vertical, or up-down, motion.

used BIOMOBIUS to process the accelerations recorded over the course of the shot and plotted the resulting accelerations. Figure 7 shows a sample plot of acceleration along the y-axis as a shot was taken. After this shot the distance travelled by the cue-ball was measured at 109cm.

In order to mechanise the process of gathering data we segmented the acceleration stream by using exaggerated shaking motion (cf. Figure 6). This allowed us to manually break the acceleration stream into a number of shots that we could correlate with the measured distances travelled by the cue-ball. Both before and after images of each shot were then extracted from the video and ran through a MATLAB algorithm to calculate the distance the ball travelled. The text file containing all information from SHIMMER was imported to EXCEL for analysis. The outputted data consisted of 3 columns, representing x, y and z axes, and over 30,000 rows, one for each sample (samples were taken at 100 Hz). After the data was segmented into shots, the segments were graphed and the magnitudes of acceleration were extracted. Shots were discarded if the cue ball struck a cushion or another ball or if it did not travel at least 100 pixels. This left us with nine "good" shots. The magnitude of the acceleration of each shot was calculated as:

$$Acc = MAX(yaxis) - MIN(yaxis) \tag{1}$$

Figure 8 shows a plot of distance, δ, (in pixels) the cue-ball travelled against Acc. As Acc increases, so too does δ, with a correlation of 0.76. From the graph

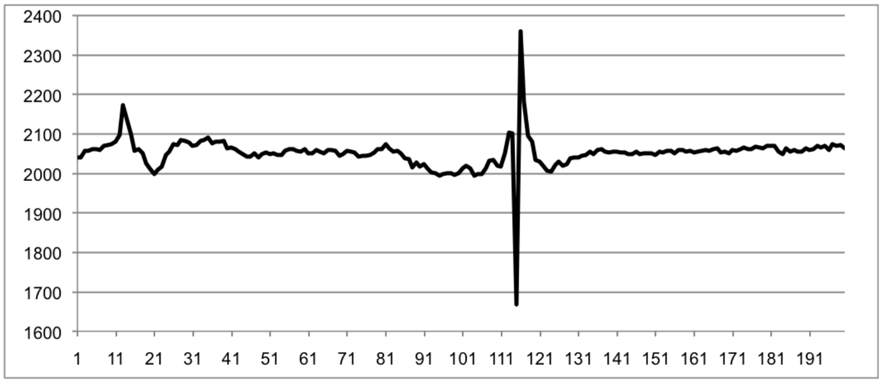

Fig. 7. The output acceleration of the SHIMMER parallel to the cue, with acceleration on the y-axis and time (in hundreds of a second) on the x-axis. A spike in acceleration can clearly be seen showing the point of contact between cue and ball. The magnitude of this spike (from peak-to-trough) is 692 units.

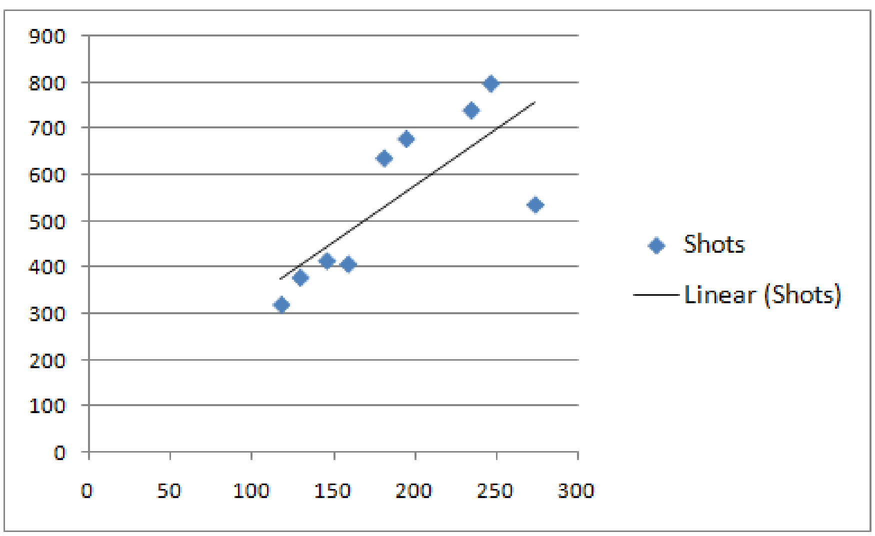

Fig. 8. Acceleration (on the y-axis) vs. Distance in pixels

the following equation can be extracted:

$$\delta = (Acc - 88.2)/2.4 \tag{2}$$

This is a general equation to estimate the distance the ball will travel upon a strike from the cue that indicates a strong linear relationship between the two sets of data points. From our test data we found the average absolute error to be 37.2 pixels, or about 22cm.

5 Discussions

The results of this experiment demonstrate a strong correlation between acceleration sensed on the cue and the distance travelled by the cue-ball. Further experiments could be designed based on this preliminary result. By combining this research with path and geometry prediction algorithms, e.g., those developed by Chan et al. [2], it would be possible to predict both the direction and magnitude of the shot. We envisage a pool tutor that integrates a feedback mechanism for the direction of the shot (such as that proposed by Larsen & Brøndsted [7]) and feedback for the strength of the shot — e.g., through the use of a vibrating cue, or LEDs indicating the strength of each shot. By using these assistants it would be possible to develop a complete pool tutor.

We believe a further development for this project would be to perform a closer examination of Squirt. According to Cross, if the ball is struck too close to its edge, then a miscue will occur, where the tip slides around the side of the ball and, the ball squirts away from its intended path by a large angle [3]. As pool players will know, a miscue is accompanied by a sharp vibration on the cue, which could be easily detected by accelerometers. The novice player could be told how badly they miscued the ball and the type of miscue could be diagnosed, e.g., whether it was to the left or right. The technology could be also used by referees to detect foul shouts, such as double-strikes, whereby the cue strikes the cue-ball twice in a shot.

6 Conclusions

Our research shows that by attaching accelerometers to the base of the cue it is possible to improve the analysis of the game. We successfully demonstrated a strong correlation between accelerations on the base of the cue to the distance the ball will travel. Further developments could be used to determine if the player is striking the ball correctly, squirting, or is having poor contact with the cue-ball. This papers concludes that by using simple, lightweight sensors, such as the SHIMMER, along with image analysis techniques it is possible to begin to “sense” the game of pool. We believe this extra information could easily be used to teach a novice player to improve their performance.

Acknowledgments

The authors would like to thank Benoit Gaudin for his contribution and help during this project as well as Gabriel Muntean, Aaron Quigley, Dan Hartnett, and everyone who worked so hard in putting together the ODCSSS internship and made this project possible.

Deirdre Dalton is supported with funding from the Online Dublin Computer Science Summer School (ODCSSS). This work is partially supported by Science Foundation Ireland under grant no. 03/CE2/I303-1.

References

1. ME Alian, S. Shouraki, M. Shalmani, P. Karimian, and P. Sabzmeydani. Roboshark: A gantry pool player robot. *ISR 2004*, 35.
2. Matthew Chan, Aaren McLucas, Chelsea Stenner, and Richard Perry. Pool table trainer, 2009. Dept of Electrical and Computer Engineering Villanova University Pennsylvania.
3. Rod Cross. Cue and ball deflection (or “squirt”) in billiards. *American Journal of Physics*, 76(3):205–212, 2008.
4. Justin Ghan, Tomas Radzevicius, Will Robertson, and Alexandra Thornton. Robotic pool player, 2006. Mechanical Engineering The University of Adelide Austraila.
5. Michael Greenspan, Joseph Lam, Marc Godard, Imran Zaidi, Sam Jordan, Will Leckie, Ken Anderson, and Donna Dupuis. Toward a competitive pool-playing robot. *Computer*, 41(1):46–53, 2008.
6. Hao Guo and Brian Mac Namee. Using computer vision to create a 3d representation of a snooker table for televised competition broadcasting. In *Proceedings of the 18th Irish Conference on Artificial Intelligence and Cognitive Science*, pages 220–229, 2007.
7. Lars Bo Larsen and Tom Brøndsted. A multi modal pool trainer. 2001. Center for PersonKommunikation Aalborg University Aalborg Denmark.