ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Automatic Control Laboratory
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Vision system for babyfoot

Professor:
Christophe Salzmann
Colin Jones

Author:
Martin SAVARY

Supervisor:
Milan Korda

Lausanne, June, 2013

Laboratoire d'Automatique

EPFL STI IGM LA
ME C2 399 (Bâtiment ME)
Station 9
CH-1015 Lausanne

Téléphone : + 41 21 693 11 71
Fax : + 41 21 693 25 74
E-mail : colin.jones@epfl.ch
Site web : http://la.epfl.ch

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Lausanne, avril 2013

Projet de semestre M2
Printemps 2013

Candidat : **SAVARY Martin**
Section de Microtechnique

Sujet : **Vision System for Babyfoot**

The LA's babyfoot has been modified to replace one human player by a mechanical "arm". The next step is to provide vision to the cybernetic opponent.

This project aims at designing the vision system for the Babyfoot. It has to be precise enough to permit an accurate ball following and fast enough to permit the real-time control of the mechanical "arm". The student will adapt the Babyfoot, select the hardware and design the software to acquire the ball position. If time allows it, the simple control of the mechanical "arm" can be designed.

Prerequisite: Interest in control, mathematics, programming, ...

Le nombre de crédits réservés au plan d'études pour ce projet est de **douze**.

Dr. Christophe Salzmann
Prof. Colin Jones

Abstract

Vision system for babyfoot

AUTOMATIC CONTROL LABORATORY (LA) SEMESTER PROJECT June 2013

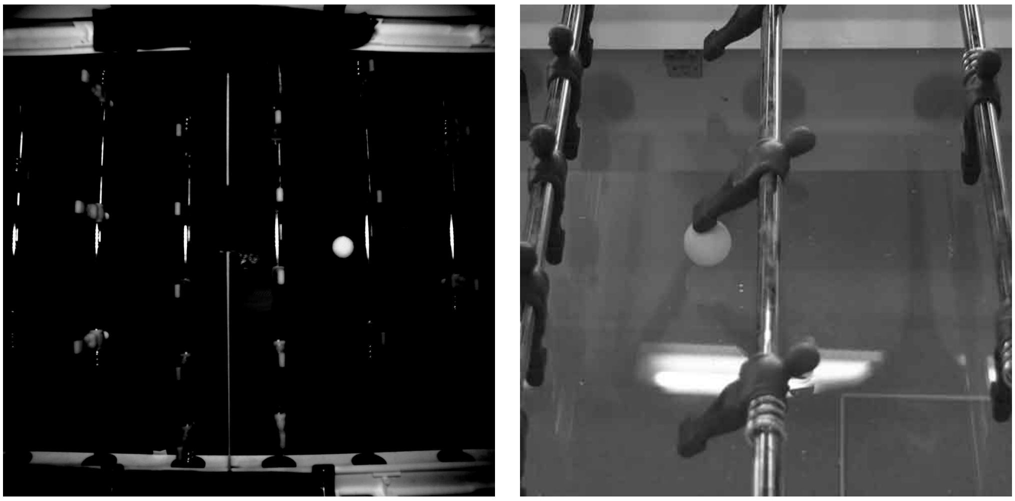


Figure 1: The camera view of a babyfoot ball and its real position on the table.

The Automatic Control Laboratory wishes to resplace a babyfoot human player by a robotic system. They have already created a mechanical arm in order to control one bar of the table. But, for now, no vision system has been implemented.

The purpose of this project is to look for a high speed camera, to modify the babyfoot table in order to set the camera and to develop a tracking algorithm.
The proposed software solution is based on an "image comparison" algorithm which subtract each part of the camera picture with a model of the ball in order to find where the model fits the best. A filter is used to limit the noise on the perceived trajectory and different optical correction is done to increase the precision of the found position.
The result of this implementation allows to track the ball with very small time of process and a good accuracy.

Acknowledgements

I would like to thank Christophe Salzmann and Milan Korda for their supervision during this research and Francis Tschantz for his presence and help.

I am also very grateful to the Automatic Control Laboratory for allowing this kind of unusual work to take place, and to the EPFL for making it accessible to students.

Contents

List of Figures

1 Introduction

1.1 Aim of the project

The Automatic Control Laboratory wishes to resplace a babyfoot human player by a robotic system. They have already created a mechanical arm in order to control one bar of the table. But, for now, no vision system has been implemented. The main purpose is to find a way to extract the position of the ball in real-time so that this information can be used by the robotic arm.

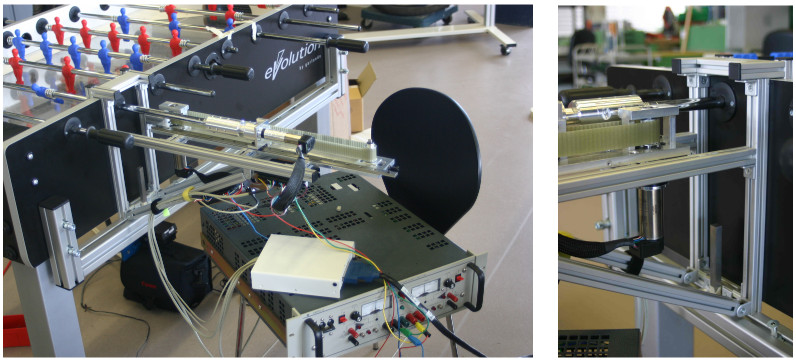


Figure 1.1: Robotic arm.

The tracking of the ball needs to be precise enough to permit an accurate ball following and fast enough to permit real-time control. Indeed, the size of the ball is only 33.5 milimeters on a table of 1.2 meters and its speed can exceed 5 meter per second, that mean the ball can pass through the whole table in less than 0.2 second.

Although several types of hardware could be used to provide an effective "vision", like a matrix of sensor on the ground for example, this project is based on the utilisation of a camera. This solution allows a good resolution and a pretty fast tracking without forgetting that there is no need to develop the hardware and it can be reused for other applications.

1.2 Steps of the work

This project includes two parts. The first one is about hardware, that means the choice of camera with an appropriate lens and the modifications made to the table.

The second part is focused on the tracking software. The different steps of the image processing, the results and the performance of the system.

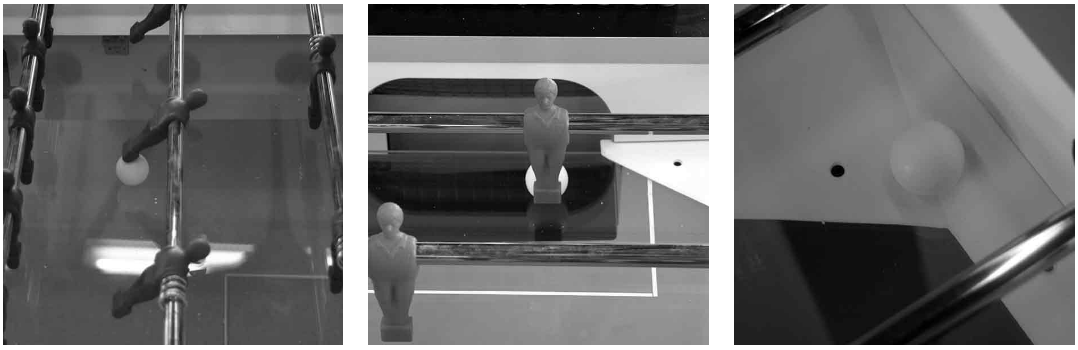
2 Hardware

2.1 Camera

2.1.1 Camera positionning

The use of a camera for tracking can be a problem in some configurations. When the ball is near or under a figurine, or when the ball is close to the border of the table, the camera can't see it. To avoid these difficulties, a solution is to place the camera under the table with a transparent tray. In this way the ball stays more often visible for the vision system.

One case is still complicated. The border of the arena is covered with a white plastic piece (figure 2.1c). This piece has a complex shape and it's difficult to replace it with a transparent material without much optic distortion, so this situation should be fixed by software and the predicting of the position.



(a) Under a figurine (b) Close to the goal keeper (c) On the border with a white background

Figure 2.1: Three different cases where the ball is difficult to see by a camera above the table

2.1.2 Choosing the camera

The most important features of the babyfoot table and the parameters of the system are undermentioned:

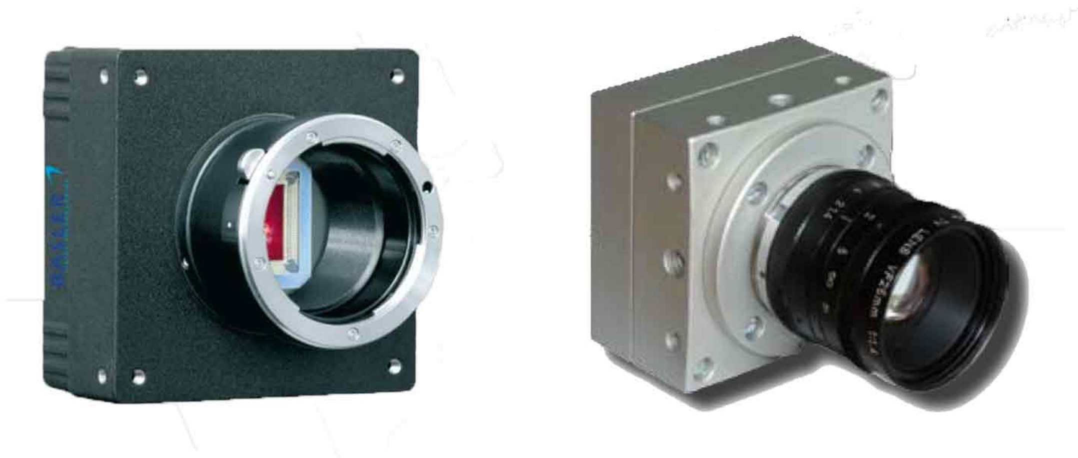
Babyfoot table	
Size tray	1220 x 720 mm
Height	750 mm

Ball	
Size	ø33.5 mm
Max. speed	>5 m/s

The ball can move very fast so the frequency of the camera must be high. With a speed between 200 and 500 frames per second, the estimation of the maximal displacement of the ball with a speed of 5m/s is about 2,5cm to 1cm. Of course, a faster camera would be better but the time between frames will not be long enough to process the picture. At 500fps, the maximum time for the ball tracking, including the acquisition of frame, is only 2ms.

The resolution of the sensor must also be taken into account. With a table size of 1220x770mm, a resolution of 1280x640 pixels gives a ratio of about 1px per millimeter. Again, the higher the resolution, the higher the final accuracy, but a large number of pixels implies a long-time process.

Finally, two high speed cameras were selected. The next table compares the principal features between them.



(a) Basler A504k

(b) Optronis CL600x2

Figure 2.2: The two selected cameras.

Basler A504k	
Sensor size	1280 x 1024 px
Pixel size	12 μm
Frame rate	500 fps
Output format	8 bits
Output type	Camera Link
Mount type	F-mount C-mount optional
Housing size	41.5 x 90 x 90 mm
Price	12'900 CHF

Optronis CL600x2	
Sensor size	1280 x 1024 px
Pixel size	14 μm
Frame rate	500 fps
Output format	8 bits
Output type	Camera Link
Mount type	C-mount F-mount optional
Housing size	56 x 56 x 44 mm
Price	7'840 CHF

These two cameras are very similar. We can expect that they present a difference in noise level because their prices are very different. Unfortunately the datasheet is not complete enough to confirm that.

This project was done with another camera. Until the laboratory ordered a new camera, an older one, already acquired by the LA, was used. These specifications are undermentioned:

Gazelle CL-41C6	
Sensor size	2048 x 2048 px
Pixel size	5.5 μm
Frame rate	150 fps
Output format	8 bits
Output type	Camera Link
Mount type	C-mount
Housing size	44 x 29 x 59.5 mm

The student who worked with this camera before reported a problem of transmission between the camera and the frame grabber. It seems that each third frame is not transmitted. This dysfunction is still there. In order to avoid frames disappearing and connection problems, the real frame rate used was 90 fps.

This speed is not fast enough to track the ball, but it allows us to test and develop the tracking software.

2.1.3 Choosing the lens

The length under the table is very short. So, a wide angle lens must be used in order to see the entire arena.

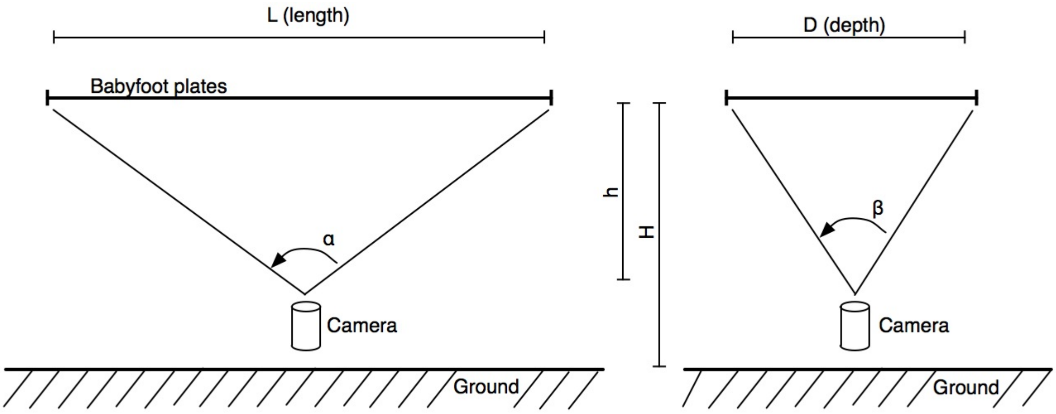


Figure 2.3: Schema of the position of the lens relative to the table

L	1220 mm	**D**	720 mm
H	750 mm	**h**	550 mm
$\boldsymbol{\alpha}$	96°	$\boldsymbol{\beta}$	66.5°

The focal length depends of the sensor size. It is calculated with the following equation.

$$\alpha = 2 \cdot \left(\arctan \frac{d}{2f} \right) \tag{2.1}$$

$$f = \frac{d}{2 \tan\left(\frac{\alpha}{2}\right)} \tag{2.2}$$

The space under the table must include the length of the camera and the lens, so, we can expect that the final distance before the table is at least 55 cm.

Here is the focal length possible for this application in function of the sensor size.

Camera Type	Sensor size	Focal
Basler A504k	1280 x 1024 12 μm	7.5 mm
Optronis Cl600x2	1280 x 1024 14 μm	8.8 mm
Gazelle CL-41C6	2048 x 2048 5.5 μm	5.5 mm

For this project, a IR lens with a focal of 8 mm was used with the Gazelle camera.

2.2 Modification made to the table

2.2.1 Transparent arena

The positioning of the camera under the table implies the use of a transparent plates for the ground of the babyfoot.

Originally the table surface was made with a tray of compressed wood and a thin tray of glass to assure a good contact with the ball. Both of these trays were replaced with plexiglas.

The final thickness of the plates is 20 mm. Because the refraction of the light through plexiglas, it will be important to implement a correction of the position of the ball during the tracking.

2.2.2 Fixation of the camera

For this project, the camera was just fixed with an articulated arm. This kind of attachment system raises difficulties. The vibration of the table during the game, and the weight of the camera with the tension of the cable move the arm. These displacements make tracking impossible. So, it's essential to establish a system that keeps the camera very stable.

Although there is a large amount of options of how to do this, this section will present only one solution. Indeed, it could be better to wait for the choice of final equipement before implementing the fixing. So, this section is not intended to present a complete solution, but just give an idea about a simple structure that requires very little material and should meet all needs.

The attachment system must satisfy certain objectives. First, it must be attached to the babyfoot table to assure that its relative position is always the same. It must be also rigid enough to avoid too many vibrations. Then, the structure must allow a positioning of the camera aligned with the center of the table and as close to the ground as possible. Ideally, the position of the camera should be able to be adjustable after the implementation of the fixing structure.

The proposed solution is based on aluminium profiles fixed to the border of the table and connected at the center with small plates that support the camera.

The position of the profiles leaves the free space for the insatllation of light. The same structure can be used as support for a light diffusion plate.

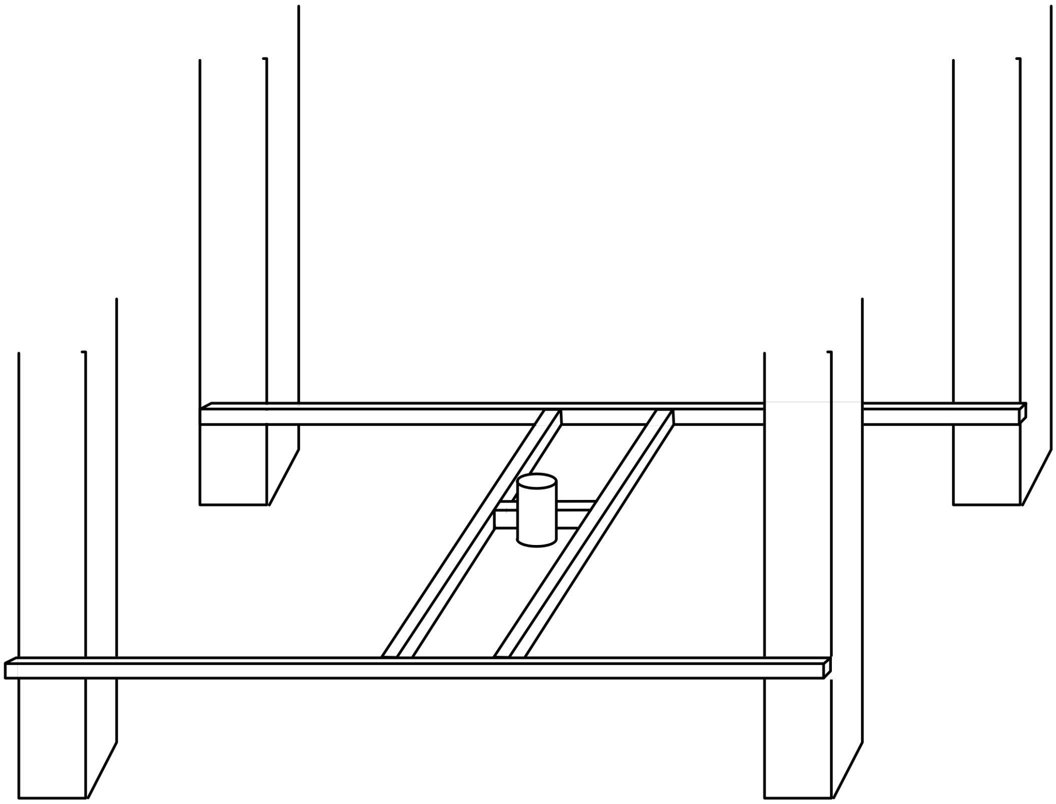


Figure 2.4: Schema of solution for the fixation of the camera

3 Sofware

3.1 Tracking algorithm

Before the implementation of the tracking software, a preliminary study was made to compare two algorithms able to find the ball in a picture. The first one, named "hough circle transform", allows us to find a specific shape in an image. The second is based on the comparison between the complete picture and an example of what we want to find.

The algorithm of hough circle transform, starts the process with an edge detection to brings out the principal shape of the picture. After that, it compares all edge pixels two by two and notes at which circle they could belong. The circle most represented should be the sought circle.

This way is very sensitive to the precision of the circle. In our application, because of the background, the edges of the circle are not always well-defined. Moreover, the distortion suffered by the border of the picture caused by the lens and the geometric properties of the situation show the ball elongated or deformed. This algorithm is not able to adapt itself to these deformities, without forgetting that the background could present a parasite circle detected by the algorithm.

So, despite the fact that this algorithm is not dependant on the brightness and is able to take into account a change in scale of the ball, it cannot be used in our application.

The second algorithm compares two pictures, a kernel that represents what the ball should look like and the frame picture taken by the camera. This kind of "template tracking" finds the part of the camera picture that fits the best with the kernel. So, it's important to choose a very realistic kernel able to represent the tracked object in all situations.

The variations of brightness are very dangerous, so, with this process it could be important to correct the loss of light at the border of the picture by software.

Finally, the process based on the image comparison fits better with our application, but different factors need to be taken into account during the development, such as the sensitivity of the change in scale of the ball and the ambient light.

3.2 Absolute difference

3.2.1 Application

The mathematic algorithm of image comparison used during the image processing is based on a difference of pixel intensity.

The idea is to substract the intensity of the frame picture to the kernel for each pixel of the entire picture. The smaller this result is, the better the kernel fits with reality and the most probably the found position is the position of the ball.

$$r(x,y) = \sum_{x'y'} |K(x',y') - I(x+x',y+y')| \tag{3.1}$$

With K the kernel and I the picture taken by the camera.

The size of the kernel calculated with the apparent size of the ball is a side of square 141 px. That means that each image comparison needs about 20'000 subtractions and absolute value calculating. This process takes time, so only one pixel out of twenty is used. Moreover, in order to limit the number of comparisons a threshold was applied before and only the part of the frame where the picture has white pixel is compared.

The application of the tracking algorithm is done in two steps. First, when the software doesn't know where the ball could be, the algorithm is applied to the entire frame picture, then when the ball is found once, the algorithm process only a little square around the last position of the tracked object. The size of the new little process window is calculated with the speed of the ball at the last step with a safety margin.

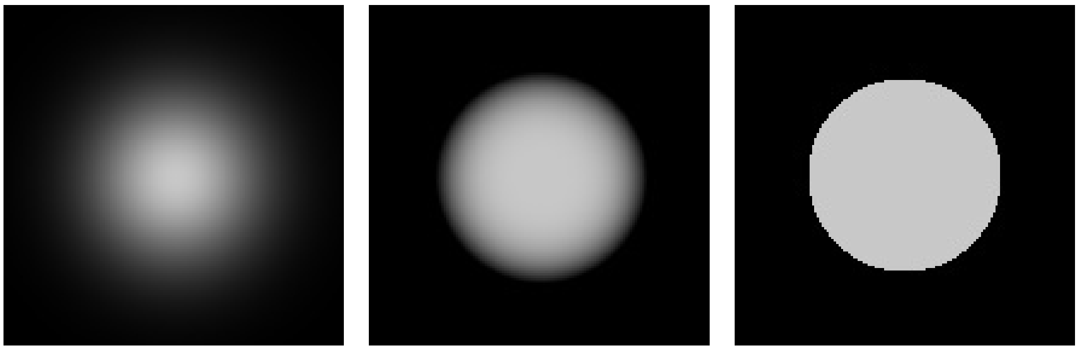
Of course, if the software loses sight the ball, it starts treating the whole frame again.

This way allows it to save a lot of time and to make the process very fast.

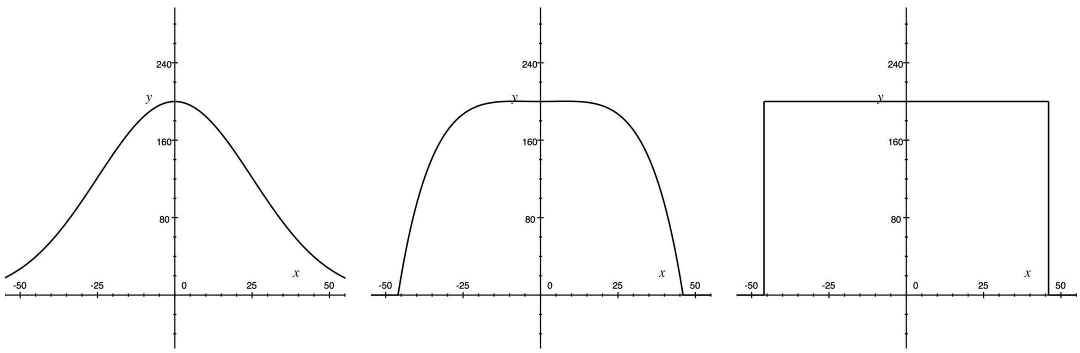
3.2.2 Kernel

As previously mentioned, the choice of the kernel is crucial. It has to fit with reality as often as possible while the appearance of the real ball can change a lot depending on its position on the table.

Three different kernels were tested.



(a) Gaussian kernel (b) Modified Gaussian kernel (c) Circle kernel



(d) Pixel value of the gaussian kernel (e) Pixel value of the modified gaussian kernel (f) Pixel value of the circle kernel

Figure 3.1: The three different kernel with the corrsponding function.

The first kernel (3.1a-d) is based on a simple gaussian function. As all the kernels tested, the maximum intensity value is 200 instead of 255 (8bits) because this value seems to fit better with real ball.

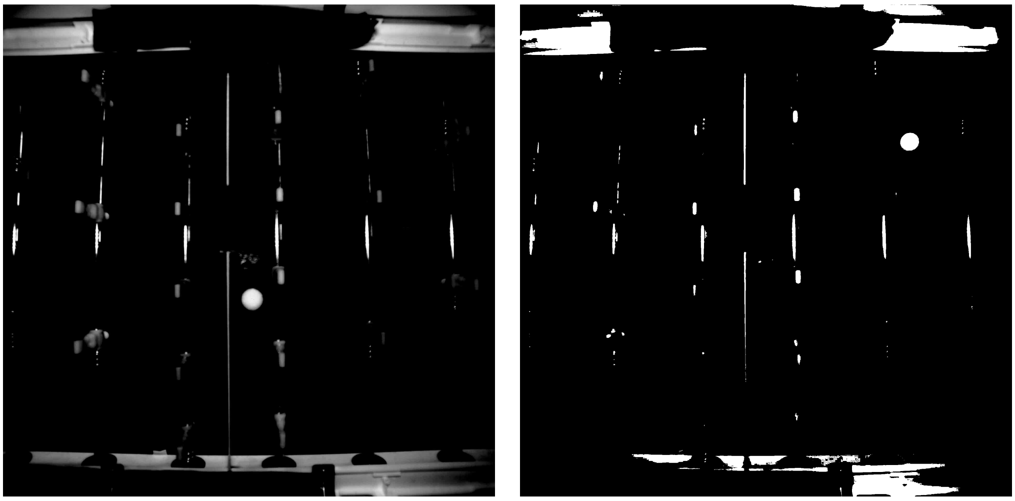
The second (3.1b-e) is a modified version of the first. Indeed, that is a mixture of two gaussian functions given by:

$$K(x,y) = -a \cdot e^{\left(-\frac{x^2}{2\cdot\sigma^2} - \frac{y^2}{2\cdot\sigma^2}\right)} + a \cdot e^{\left(\frac{x^2}{2\cdot\sigma^2} + \frac{y^2}{2\cdot\sigma^2}\right)} + 4 \cdot a \tag{3.2}$$

With $x, y \in]0; 141]$. and a the half maximal intensity of the kernel.

Finally, the third (3.1c-f) is a simple and solid circle.

These models were tested with three kinds of situations, when a ball is present on the middle of the frame picture, more on a border or when no ball was visible.



(a) Ball close to the middle

(b) Ball more on a border

Figure 3.2: Examples of ball positions tested with the kernels (frames seen by the camera).

The tables 3.1, 3.2 and 3.3 present the value of the results found with the expression 3.1 for the three tested pictures for each kernel. As a reminder, the smaller the result, the better the kernel. Moreover, the standard deviation of the results and the positions in x and y is given. The values are an average found with 22 frames (0.25 second of recording) and a motionless ball.

Table 3.1: Gaussian kernel results

	Middle	Border	No ball
Result	$\bar{r} = 797$, $\sigma = 11.8$	$\bar{r} = 535$, $\sigma = 17.1$	$\bar{r} = 1421$
Pos. x [px]	$\sigma = 3.50$	$\sigma = 1.27$	-
Pos. y [px]	$\sigma = 4.41$	$\sigma = 1.56$	-

Table 3.2: Modified Gaussian kernel results

	Middle	Border	No ball
R(x, y)	$\bar{r} = 175$, $\sigma = 28.7$	$\bar{r} = 622$, $\sigma = 11.2$	$\bar{r} = 1551$
Pos. x [px]	$\sigma = 1.02$	$\sigma = 1.18$	-
Pos. y [px]	$\sigma = 1.02$	$\sigma = 1.02$	-

Table 3.3: Circle kernel results

	Middle	Border	No ball
R(x, y)	$\bar{r} = 777,\ \sigma = 17.4$	$\bar{r} = 1292,\ \sigma = 10.2$	$\bar{r} = 2057$
Pos. x [px]	$\sigma = 2.03$	$\sigma = 1.16$	-
Pos. y [px]	$\sigma = 1.35$	$\sigma = 1.02$	-

We can notice that the model based on the modified gaussian is the best. It fits quite well with the ball even if it is not on the middle of the table. Furthermore, when the ball is not visible, the result of the comparison is big enough to avoid confusion with a part of the background.

The small standard deviation found on the position assures a good repeatability of the applying of the tracking software.

The valuea given are very dependant of the ambiant light and the background. In order to improve the quality of a kernel it's important to have a background as dark as possible and an uniform illumination.

3.3 Filtering

The position found with the solution based on the image comparison are noisy. As we saw on the table 3.2 the standard deviation of the position of a motionless ball is about 1 px. This value corresponds to about 0.5mm in the scale of the table.

In order to limit the noise, a filter could be implemented.

A test was done with a Kalman filter. This solution uses a prediction of the next most likely position and a measure of this. Then, these two pieces of data are mixed with different matrix that represent noise and inaccuracy.

Although a Kalman filter is often very effective in the filtering of trajectory, in our case a filter based on a prediction ultimately provies an unreliable result. In a babyfoot game, the ball moves fast and changes direction very often. At each change of direction, the prediction calculated with the Kalman filter goes straight during one step, and is far from the real position of the ball. Perhaps, with the use of a very high speed camera (>200fps) it is possible to use a Kalman filter because the time between two steps is short enough to avoid a crucial difference between prediction and real position after a change of direction. But, the Gazelle camera is not fast enough to assure this, and if the ball turns often, the filter is not able to follow the correct position.

Finally, a very simple filter was implemented. The position of the ball can be found by two ways, the first one is the position where the kernel matches the best with the camera picture, and the second is the pixel placed in the center of the white pixel representing ball.

In order to apply this filter, the software starts with an image comparison algorithm and when the ball is found (x_1, y_1), it calculates the centre of the average of the white pixel in a small square around the ball (x_2, y_2). Then these two values are mixed with different weight in order to optimize the noise reduction (equ. 3.3).

$$\begin{pmatrix} x \\ y \end{pmatrix} = a \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + (1-a) \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{3.3}$$

The graph 3.3 shows the noise of the position x of a motionless ball after optimization of a.

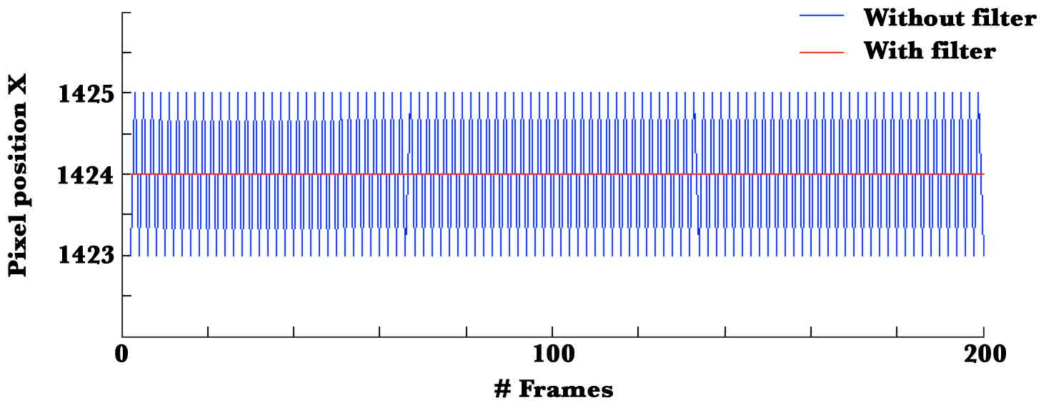


Figure 3.3: Noise of the track of a motionless ball with and without filtering.

3.4 Optic correction

3.4.1 Light refraction and angle of view

The geometry of the entire system (babyfoot, table, plexiglas plate, camera position, etc) makes it essential to correct the position of the ball found on the camera picture. Two principal things affect the difference between the real position of the ball and the perceived one: the angle of view of the camera and the light refraction through the table plate. Both of these have to be fixed by software.

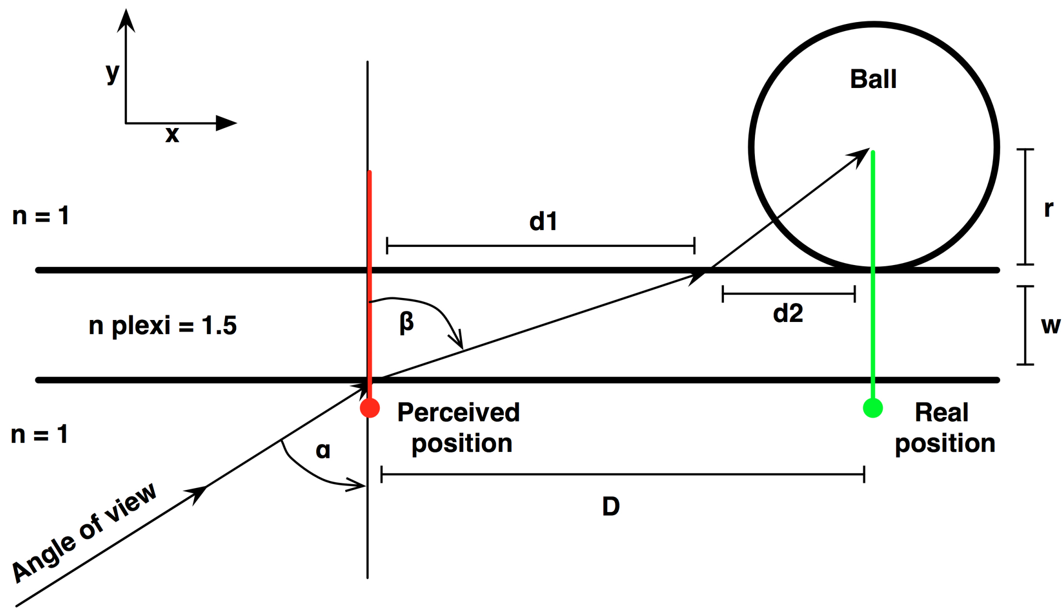


Figure 3.4: Schema of the situation between the perceived position of the ball and the real position.

With snell's law (equ. 3.4) we can find the the expression of β in function of the angle of incidence of the light α (equ. 3.5).

$$n_1 \cdot sin(\alpha) = n_2 \cdot sin(\beta) \quad (3.4) \qquad \beta = sin^{-1}\left(\frac{n_1}{n_2} \cdot sin(\alpha)\right) \quad (3.5)$$

The path travelled by the light in the plexiglas according to the coordinate x is therefore:

$$d_1 = w \cdot tan\left(sin^{-1}\left(\frac{n_1}{n_2} \cdot sin(\alpha)\right)\right) \tag{3.6}$$

The distance d_2 is merely given by the projection of the center of the ball in the direction of the angle of view:

$$d_2 = r \cdot tan(\alpha) \tag{3.7}$$

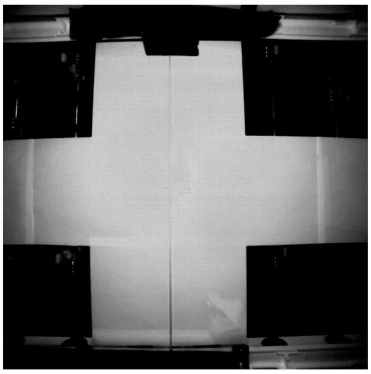
Finally, the difference between the perceived position and the real position is given by:

$$D = d_1 + d_2 = w \cdot tan\left(sin^{-1}\left(\frac{n_1}{n_2} \cdot sin(\alpha)\right)\right) + r \cdot tan(\alpha) \tag{3.8}$$

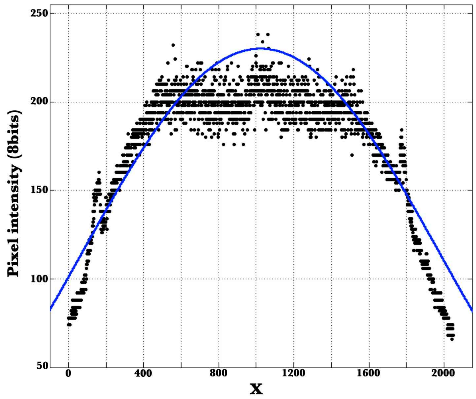
3.4.2 Brightness

Because of the angle of projection of the picture on the camera sensor and the use of a IR light which does not project uniformly, the brightness of the final frame is uneven.

A picture with a white background, taken with the camera, shows the situation clearly.



(a) Brightness analyses picture



(b) Brightness analyses graph with the approximated curve in cos^2

Figure 3.5: Example of a picture with a white background. The graph shows, the pixel light intensity of the line Y=1024 of the picture 3.5a

When the ball is on a border of the table, it becomes too dark to be tracked. So, a brightness correction of the picture is necessary.

The light intensity distribution is approximated with a cos^2 function:

$$B(x) = 230 \cdot cos\left(\frac{(x-1024)\cdot\pi}{3800}\right)^2 \tag{3.9}$$

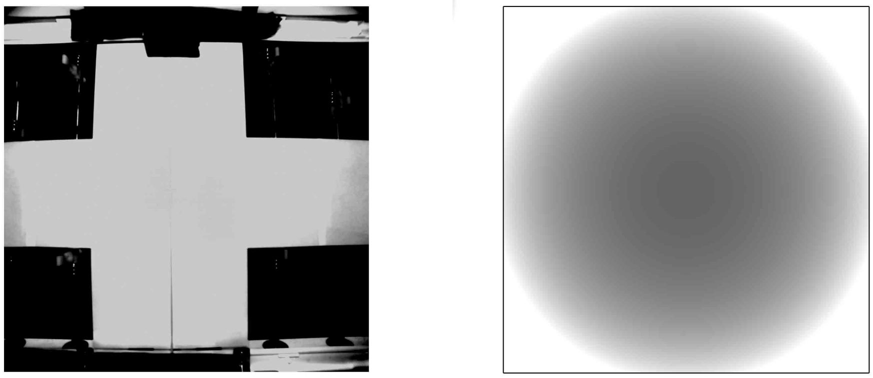
With $x \in]0;2048]$.

The inverse of the function 3.9 enables us to correct the intensity of each pixel. The 3D function is given by:

$$B(x) = \frac{1}{cos\left(\frac{\sqrt{(x-1024)^2+(y-1024)^2}\cdot\pi}{3800}\right)^2} \tag{3.10}$$

With $x, y \in]0;2048]$.

As we can see on the figure 3.6, the brightness is more uniform. The ball should appear with the same brightness everywhere on the camera picture.



(a) Camera Picture with brightness correction

(b) Visual representation of the multiplier function used for the brightness correction (equ: 3.10)

Figure 3.6: Result of the pixel intensity correction

3.5 Structure of the program

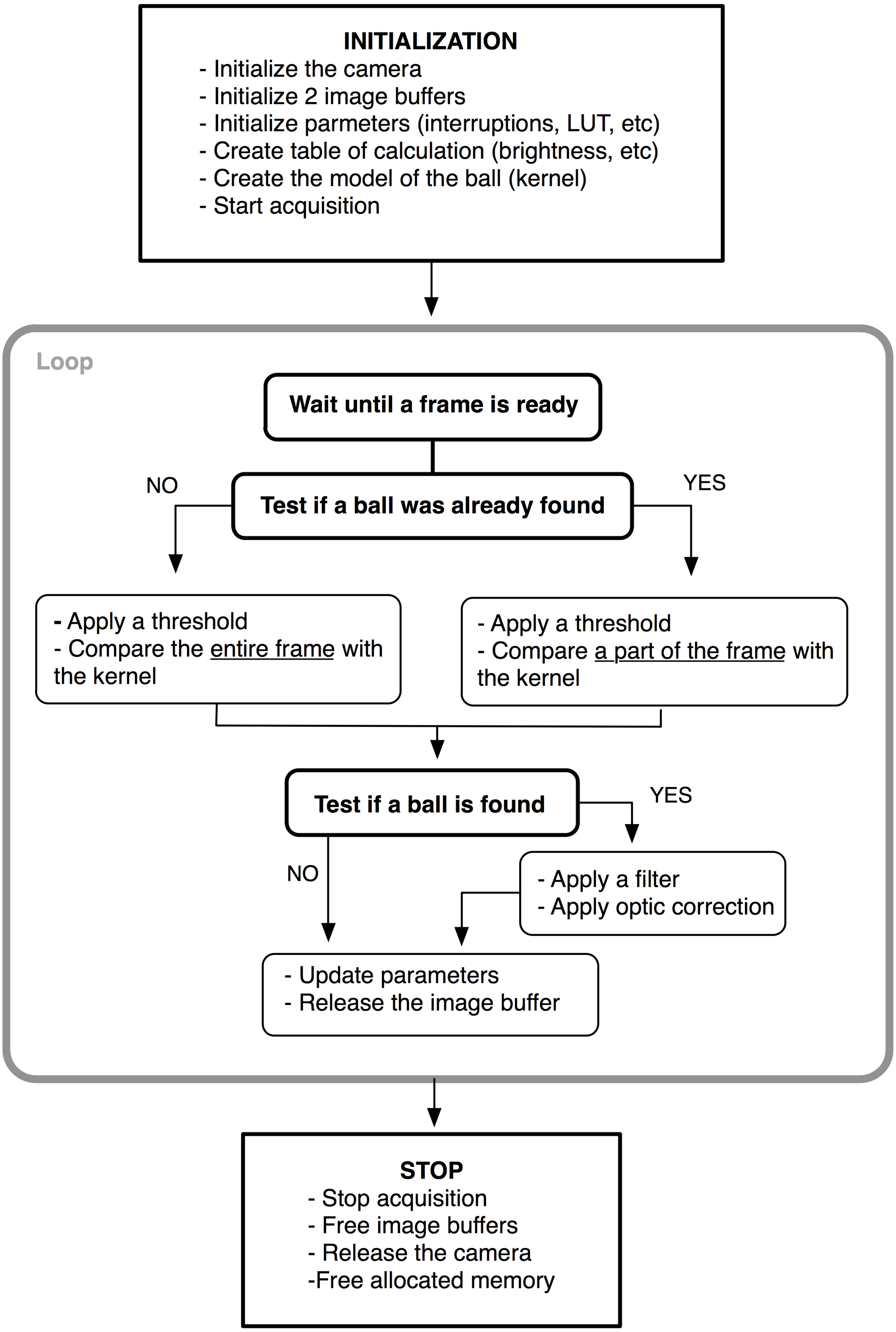


Figure 3.7: Structure of the tracking process

3.6 Time of process

The complete time of the process depends of the size of the image used. Indeed, as it is explained in the section 3.2.1 the tracking algorithm treats the entire picture if the ball was not found at the last step and only a small window if the ball was found just the step before.

The ambiant luminosity also has an important impact on the time of process. As the software uses only the light pixel for the image comparison, the quantity of white parasite pixel makes the tracking process slower.

The tests were done in the laboratory room, during the day, with pieces of black stoff on the border of the babyfoot in order to avoid light background. The pictures show in the figure 3.2 are quite representative of the brightness during the speed test.

When the image comparison was done on the entire picture, the time of process (without the image acquisition) was less than 10^{-2}s. The best result was $3.7 \cdot 10^{-3}$s, but the condition of brightness has to be very controlled. With the use of a small window around the ball the time of process is drastically lowered, which means less than 10^{-3}s with a mininum time at $3.5 \cdot 10^{-4}$s.

The time for the acquisition of a frame is about $1.1 \cdot 10^{-2}$s. That is clearly higher than the time of process. The speed is limited by the time of acquisition and not by the time of process. So, the use of a faster camera is possible knowing that a camera speed of 500fps allows a time of process of $2 \cdot 10^{-3}$s.

4 Conclusion

4.1 Conclusion of the work

The aim of this project was to install a camera system on a babyfoot game in order to track the ball. Different algorithms of tracking were tested. The process based on an image comparison between the camera picture and a model of the ball were choosen. The parameters of the algorithm as the choice of the model and the kind of data filter have been optimized. Finally, several corrections were made because of optic or geometric distortion due to the installation.

The result is that tracking software is able to find the ball and track it with a time process of less than 10^{-4}s. For now, the precision of the tracking is limited by the resolution of the camera that corresponds to 0.5mm in the position of the ball with a good level of brightness.

4.2 Future work

The use of a faster camera without IR light will involve modification of the babyfoot table. A good uniform light should be placed under the table with a diffusion plate in order to avoid light reflection on the plexiglas. Moreover, The background should be hidden, for example with the installation of tinted glass above the table. Then, a stable fixation for the final camera should also be built.

Finally, the trajectory of the ball could be predicted in the part of the picture where the ball is not visible, in the corner for example. So, work can still be done in order to complete this tracking software until we can be certain to know the exact position of the ball everywhere on the babyfoot table.