DIPLOMA THESIS

Object Touchdown Position Prediction

A Stereo Vision Based Approach

Submitted at the Faculty of Electrical Engineering, Vienna University of Technology
in partial fulfillment of the requirements for the degree of
Master of Sciences (Diplomaingenieur)

under supervision of

O. Univ. Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich
Dipl.-Ing. Dr.techn. Friederich Kupzog

by

Martin Pongratz
Matr.Nr. 0326131
Sebastian Kneipp Gasse 9/20, 1020 Wien

19.09.2009 _____

Kurzfassung

Individualisierung macht auch vor Produktionsprozessen nicht Halt. Diese müssen immer flexibler und rekonfigurierbarer gestaltet sein. Deswegen müssen Produktionslinien hohe Anpassbarkeit aufweisen. Eine Möglichkeit, um die auftretenden Herausforderungen zu lösen, stellt der sogenannte "Wurftransport-Ansatz" dar. Dabei werden die zu transportierenden Güter von einer Station zur nächsten geworfen, statt mit Förderbändern transportiert zu werden. Die vorliegende Arbeit basiert auf diesem Ansatz und greift die Frage nach der Prognose der Flugbahn auf. Diese ist eine Grundvoraussetzung, um erfolgreich Objekte zu fangen, da der Fangmechanismus rechtzeitig in die richtige Position gebracht werden muss. In dieser Arbeit werden zur Prognose der Endposition gemessene Positionsdaten verwendet. Die Positionsdaten werden dabei mit Hilfe eines Stereokamerasystems ermittelt. Ein Vergleich verschiedener mathematischer Vorhersagemodelle für die Flugbahn an Hand verschiedener Faktoren (Bildaufnahmefrequenz, Auflösung) ist ebenfalls Teil dieser Arbeit. Auf die Abhängigkeit der Prognosegenauigkeit von der Bildaufnahmefrequenz und der Auflösung der Bilder wird dabei speziell eingegangen. Zuletzt wird diejenige Lösung vorgestellt, welche im Vergleich am besten geeignet scheint. Dabei handelt es sich um ein vereinfachtes Modell, welches auf der Flugphysik basiert und die präzisesten Vorhersagen ermöglicht. Es zeichnet sich durch eine ausgezeichnete Stabilität der Prognose und gute Skalierbarkait im Bezug auf Aufnahmefrequenz und Auflösung der Kamera aus.

Abstract

Flexible production is one of the key factors to pay account to the needs for individual products. Besides the flexibility of the productions systems for industrial production themself also the transportation systems need to be flexible, fast and easily reconfigurable. Transport-by-throwing is one suitable approach where goods are thrown from one station to the next station. This work deals with the prediction of the flight trajectory which is necessary to enable reliable catching. The prediction is based on measured positions of the object in the early flight phase. The setup and realization of such a position acquisition system, consisting of two cameras, is presented. Three prediction models are introduced and their prediction accuracy is analyzed and compared. In addition the dependency of the prediction accuracy on the frame rate and resolution of the vision system is discussed. The model based on simplified flight physics shows the best prediction accuracy and also scales well when changing frame rate or resolution of the stereo vision system. Increasing resolution is found to be superior to increasing frame rate within the same bandwidth envelope.

To those who helped
and the others who didn't
for their help
and the challenges they created.

Table of Contents

Abbreviations

1D	one dimension(al)
2D,3D	two dimension(al),three dimension(al)
AFD	Adjacent Frame Difference
BM	Background Model
CCTB	Camera Calibration Toobox for Matlab
DST	Dispersive Signal Technology
EB	Error Bound
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
IP	Interception Position
IPP	Interception Position Prediction
LOG	Laplacian Of Gaussian
PIP	Predicted Interception Position
PDF	Probablility Density Function
TD	Throwing Device

1 Introduction

Modern welfare in the western world has increased the demand for more individuality. Ranging from individual-exotic holidays to an individual life-style, people use every possible way to set themselves apart from the crowd. Every manufacturer and service provider has to adapt to this situation and the wide range of similar but not exactly identical products offered shows how deep this demand has already influenced the market. In the field of service providers this behavior requires a more and more individual treatment of customers while the effects on the manufacturers are not only influencing the employees but also the production lines. Years ago machines and robots were introduced to increase the output of a production line. In the long term this has leveled the quality of the end products as every single production step which is repeatedly done by a machine is more exact than a human could do. In 1914, when Henry Ford [9] introduced the conveyor belt first in car manufacturing, humans were doing their fraction of the whole production process for every car during their shift. Today robots have replaced them — but they still only do one small manufacturing step twenty four hours a day. The demand for more individual products had and will have a big influence on this as well. More personalalized products demand a flexible manufacturing process. A flexible manufacturing process requires flexible production lines with flexible machines and robots. But the key to this process of individualization is that the backbone for the manufacturing process, the transportation system, has to be more flexible and individual as well.

1.1 Motivation

In current production halls the more than 200 years old idea of Oliver Evans [10] to use continuous transport systems like a conveyor belt is widely used. In some cases also transport carts or trollies are employed. A common attribute of both systems is that they cannot be reconfigured very fast. Conveyor belts are heavy and require a lot of space as well. Changing their routes requires a complete production stop of the facility along this transport system for the time of rerouting and reconfiguring. This also means costs for the operator of the facility. In times of individualization different products or similar products with small varieties have to be produced within one facility, if possible on one production site/line. However, long production breaks have to be avoided in order to keep the facility at a high load and the final price of the product low. Arising from this request, fast reconfigurable transportation systems have to be found. The transportation system of the future is able to handle every object on its own. Reconfiguration time is close to zero and the order of processing on different working stations can be changed in order to provide a

possibility for load balancing. A suitable solution for this requirements is a transportation system which is based on throwing and catching. Assembly stations could be equipped with a throwing and catching device and handle each piece individually. Fieldbusses would serve as a backbone for communication but the tasks of throwing and especially catching have to be solved decentralized in a distributed architecture, where each assembly station has to catch the piece thrown towards it on it's own.

1.2 Problem Statement

In the proposed transportation system the task of transportation is divided into throwing and catching. These two tasks can be processed on their own but finally the crucial attributes of a transportation system are speed, reliability and drop out rate. In order to achieve or surpass common transportation systems in theses regards both components of the system have to be optimized. The throwing system should throw all objects towards the same position at the catching device. This would lead to a minimal moving range of the catching device and minimize the mechanical requirements for the positioning of the catching device. Due to different leverages like local air flow, different air density, different flying properties of thrown objects and so on even a perfect throwing system's variation of the interception position at the catching system would still be big enough to demand the tracking of the thrown object and a precise prediction of the interception position. This circumstance puts focus on the tracking and interception position prediction system. The tasks of these systems is to observe the thrown object through out the flight and to position the catching device in a position to catch the workload. This interception position prediction has to work through out the flight of the object in order to position the catching device early improve gradually with the time of observation. This also avoids spikes in the necessary force to move the catching device. The requirements for the prediction system are low cost, suitable-high precision, object independent application. The early stage of work in this field of research invalidates the last requirement of object independancies the object thrown in though out all the work presented is a tennis ball. This alliation simplifies the task in many domains while the main task of object tracking and interception position prediction can be processed extensive. In order to evaluate the process of prediction the accuracy and the temporal progress of the interception position has to be benchmarked. This on one hand can be achieved by using a gantry robot and a simple catching device [FBWW+07] and judging the system based on the catching rate of the device. A second possibility is to develop a object position detection [DBP09] system which leads to a higher quality of evaluation information as not only the binary information (object caught/object not caught) is available but also the deviation between the predicted interception position and the real position of the object in the catching plane is obtainable.

1.3 Proposed Methodology

The problem outlined requires two semi-independent systems: the prediction system and the verification system. Both can be developed and optimized on their own but only their interaction allows feasible research for the given task.
The prediction system has to be designed and realized. A binocular vision system serves as the input sensor. In analogy to hunters in nature who also use two eyes track their haul, this approach

might lead to a higher prediction accuracy than a monocular approach [BFK08]. The selection of suitable cameras that also allow to downscale the system in terms of frame rate and image resolution is necessary. Downscaling is necessary in order to determine the lowest cost setup for a given accuracy requirement. This stereo vision system has to be calibrated — also in relation to the verification system. Image acquisition of both cameras has to be done synchronous and the moving tennis ball has to be identified in each frame of both cameras. The center of the ball has to be localized in each picture and the information of both cameras has to be merged in order to obtain the related position of the ball in space. Based on the sequence of the ball positions a suitable trajectory has to be fitted into the measured data and the prediction of the future flight trajectory has to be done. This prediction has to start as early as possible and should be refined over time.
This verification system has to designed and realized similar to the prediction system. Possible designs have to be developed and evaluated. The decision between different options has to be based on the ubiquitous application of the systems, the accuracy of the system, the operation complexity and the costs. Also the work to embed it into the prediction system and their interaction have to be taken into account.
Finally the quality of the interception position prediction has to be metered. This task closely interferes with the verification system. A verification system based on a gantry robot with a catching device only delivers binary information about the prediction system (object caught/object not caught) but is more similar to the goal application. A verification system which outputs the deviation of the optimal interception position from the predicted interception position offers a countinous metering range on contrary. This higher quality of information can lead to a better benchmarking of the prediction algorithm. It even enables the selection of a suitable catching device in case the maximum catching range is known. Passive [[FBWW+07]] or active catching devices [NI05] might be used depending on the required quality and available budget in a final application.
As the cost for such a system influences the usage in real world to a great extend a performance scaling analysis will be the final part of the work. The quality of the interception position prediction depending on various factors like frame rate, image resolution and so on will be examined. The goal is to find optimal factor combinations while keeping the hardware costs for the system as low as possible.
The main focus of this work is to find answers to the following two questions:

1. Are there benefits of using two cameras compared to the binocular approach by Barteit [BFK08] and what magnitude are the benefits of?

2. How do different models for the flight trajectory influence the prediction accuracy?

2 Related Work

The task of catching or intercepting is very common in nature. Predators hunt their food. Similar humans perform multiple throws and catches in sports. Tennis stars like Roger Federer are said to have a especially good hand-eye coordination and even a special vision technique [7]. Nature has worked the task of catching and trajectory prediction out very well.
Artificial catching has to be divided into multiple sub–challenges. First, a suitable sensor has to deliver information about the actual trajectory of the flying object. In case of the human these are the two eyes (compare Figure 2.1). The comparable technical approach is a stereo vision system, used for the task in the work described here. In order to extract the information about the balls trajectory the images delivered by the two cameras have to be processed. The moving ball has to be recognized and the center position calculated. The state of the art of this part will be the topic of the first subsection.

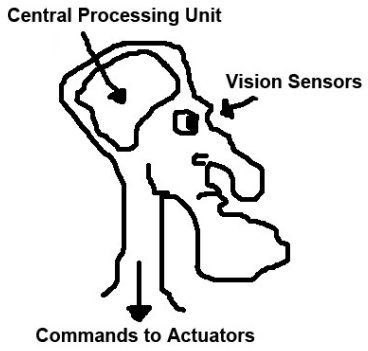

Figure 2.1: The human way. Modified from [AEN06]

The second challenge is to predict the future trajectory of the ball. In nature this very complex task is achieved by learning over many years. Basically the neural net inside the brain (figure 2.1) and it's environment learns from every single catch by using the tactile information of the catching device (hand) as a feedback for the prediction. Since many functions in humans, including this one, are by far not known completely, the human as an archetype for the solution is not suitable. Artificial ways to predict the touchdown position of a flying object will be in focus of the second subsection.
The topic of the final subsection will be the evaluation of the predicted interception position. As

mentioned above in nature this information is the key information for improving the interception positions prediction. This information is the feedback of the information loop an closes it. Available work about possible solutions will be discussed there.

Before going into the detailed subsections similar task that were already goal of research have to be noted: [AEN06], [HS91], [MHM04], [MM02b], [RA02], [NI05]. All experiments on catching described are based on a small scale of less than one meter down to some millimeters of catching range [NI05] with active [NI05], [RA02] and autonomous catching robots [RA02]. Different strategies as "gaining angel of gaze" [MHM04], [MM02b] or strategies miming the human way to catch [KSOI99], [MM02a] are used to achieve the task of catching. This work will be referenced throughout the whole chapter and the used methods and their alternatives will be described.

2.1 Vision Systems and Object Detection

Vision systems are widely used nowadays. Starting from simple low resolution photos, image acquisition has improved in quality and moved away from separated images to series of images - videos. The information obtained has developed from apparitional memories to real measured data. Improvements in optics and video processing have enabled this. An interesting example is the "'Hawk Eye"' used in Grans Slam Tennis matches in order to support the chair umpire and the linesmen [4]. Methods to extract information from videos are reviewed in the following sections. The task of obtaining the position from the video data is divided into sub-problems and alternatives dealing with them are reviewed.

2.1.1 Moving Object Detection

Basically, detecting moving objects (segmentation) in video data acquired with fixed cameras can be done by two different principles. One possibility is to consider the difference between two adjacent frames (adjacent frame difference - AFD) above a certain threshold [GW07], [TKBM99]. As the actual difference between to adjacent frames equals all the changes between the two images, objects that disappeared as well as objects that appeared newly in the field of view will be detected. One major drawback of this method lies in the sensitivity towards the outdated information which is also called ghosting and foreground aperture (Figure 2.2). On the other

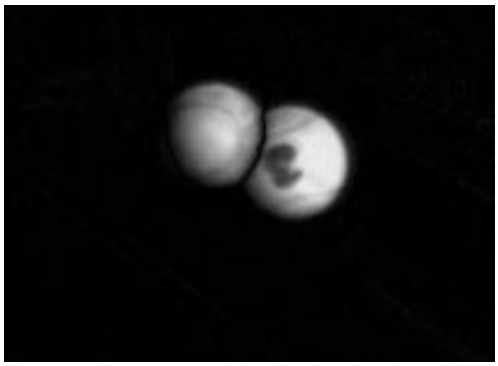

Figure 2.2: AFD-segmentation-example: Ghosting in case of a moving tennis ball; outdated ball on the left, actual ball on the right.

hand this method is not sensitive to slow changes of the background and doesn't need a lot of calculation power. In order to overcome the drawbacks of simple adjacent frame differencing different enhancements using more information about the background have been made [MMN06]. By using more information, the disadvantage of ghosting has been minimized but on the other hand the advantage of simple calculation hast been lost as well.
The second principle of detecting motion is to primary build up the background model from a number for frames and to compare the actual acquired frame with the background model. Different ways to build up the background model have been developed and reviewed [Pic04], [BJE+08]. The approaches available range from simple approaches with limited memory requirements and maximized speed to more sophisticated approaches developed in order to achieve the highest possible accuracy under any circumstances [Pic04] and have their advantages and disadvantages. Overall no perfect system is said to exists [TKBM99]. A huge advantage compared to adjacent frame differencing is the lapse of ghosting and foreground aperture (Figure 2.3).

The most simple version of this method is to compare (subtract) the actual frame to a back-

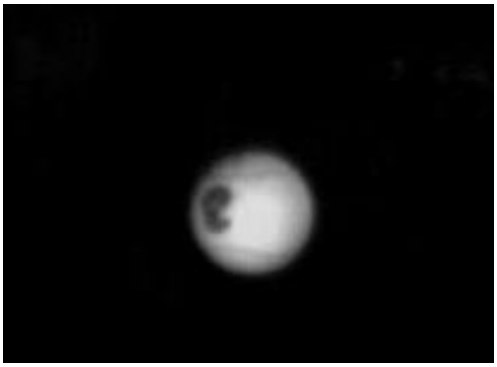

Figure 2.3: BM-segmentation-example: No ghosting in case of a moving tennis ball.

ground frame [BJE+08] which is not containing any objects to be detected. After this operation the individual pixels can be classified as background and foreground via an threshold operation. The calculation effort is minimal but the downside is that the threshold level has to be set manual. A compromise between a low threshold which implies high sensitivity also regarding to noise and low noise liability with a higher threshold has to be made.
One very common approach to build up such a background model with it's parameters automatically is the Running Gaussian Average [BJE+08], [Pic04]. A probability density function for each pixel of the frame considering the last n pixel's values is obtained cumulative. In each following frame the pixels which lie outside a certain interval around the probability density function are classified as belonging to foreground or a moving objects. By changing the adaption rate the background models sensitivity for background changes or noise in the picture can be altered. In case of a higher adaption rate also small changes which should be recognized as foreground will be suppressed. Drawbacks are lying in the necessary calculation power and storage requirements. Many more sophisticated background models are available [Pic04], [BJE+08], [TKBM99]. Their application in this domain is impossible as the calculation demands are to high. The knowledge about the shape of the thrown tennis ball and it's symmetric properties enable the use of fast and simple object detection algorithm.

2.1.2 Edge Detection

The relevant information which has to be extracted out of each image is the position of the tennis balls center. Acquiring this information out of the detected motion demands considering the shape of the object and possible sources of error. In case of a perfect recognition of the tennis ball in the frame the center of the ball lies on the extension of the ray between the cameras focal point and the balance point of the recognized shape in the image plane [SHB07, 12], [AEN06]. Calculating the balance point from the motion image is very simple: it is equal to the mean of all the pixels belonging to the ball. This simplicity causes some problems. In case other pixels than the ones belonging to the tennis ball are classified as foreground an error is introduced. This error can be minimized by optimizing the motion detection algorithm but it can not be excluded totally.
In order to minimize the impact of this error source a more complex processing algorithm of the motion image can be used. It is based on edge detection and hough transformation (Section 2.1.3). The task of edge detection is to locate changes in the intensity function [SHB07]. Changes in intensity are related to surface normal discontinuities, depth discontinuities, highlights, surface texture and shadows or illumination discontinuities. They also appear in the first and second derivate of the intensity function along a specific line of the image [GW07, 703]. Special filters called edge detectors make use of this property. These filters can be divided into three categories: Operators approximating derivates of the image using differences, operators based on the zero-crossing of the image function second derivative and operators which attempt to match an image function to a parametric model of edges [SHB07]. Common basic edge detectors of the first category are Roberts operator pair, Laplace operator, Prewitt operator pair and Sobel operator pair. Related example filter masks are shown below (Table 2.1 and 2.2).

1	0
0	-1

0	1
-1	0

Table 2.1: Roberts operator pairs

0	1	0
1	-4	1
0	1	0

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

1	2	1
0	0	0
-1	-2	-1

-1	0	-1
-2	0	2
-1	0	1

Table 2.2: $3x3$ Laplace (single), $3x3$ Prewitt and $3x3$ Sobel operator pairs

Above mentioned filter masks, excluding the Laplace operator, are grouped pairs. Each of the two masks is sensitive to intensity changes in diagonal direction orthogonal of the sensitivity direction of the other mask. This architecture enables it not only to detect edges but by merging the information of both masks also the direction of the edge can be calculated. While the Roberts operator has a high sensitivity to noise due to its small size, the Laplace operator responds to some edges twice.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Table 2.3: $5x5$ discrete approximation of LOG operator (Mexican hat)

Operators of the second group which make use of the second derivates zero-crossing are the Marr-Hildreth edge detector, the Laplacian of Gaussian (LOG) operator and Canny edge detectors [SHB07]. The LOG is based on a 2D Gaussian distribution and is calculated by using the Laplacian operator on the 2D Gaussian distribution. A $5x5$ discrete approximation of the convolution mask is shown in Table 2.3. Due to its shape this edge detector is also referred as Mexican hat.
The Canny edge detection algorithm differs from all prior mentioned operators. Edge detection based on Canny algorithm is not only based on a simple convolution with a specific operator. Instead it consists of 4 steps (modified from [SHB07] and [GW07]):

1. Convolve an image with a Gaussian filter.
2. Compute the gradient magnitude and angle images.
3. Apply non-maxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.

It is obvious that the Canny algorithm does not only deal with the local information (3x3, 5x5 pixel environment or similar+) like the prior mentioned operators. This edge detection algorithm uses the information of the whole image in order to detect the (if possible) closed loops of edges [SHB07]. This attribute separates the Canny algorithm from the other mentioned as the task of edge linking is usually very tough. The excellent detection of the Canny algorithm is essential in situations of high complexity. Figure 2.4 compares LOG (dark grey) and Canny (light grey) edge detection in two simple cases. The left image shows the detected edge in case the ball is far away from the camera at low light and the right image shows the detected edge when the ball is close to the camera and at high light.

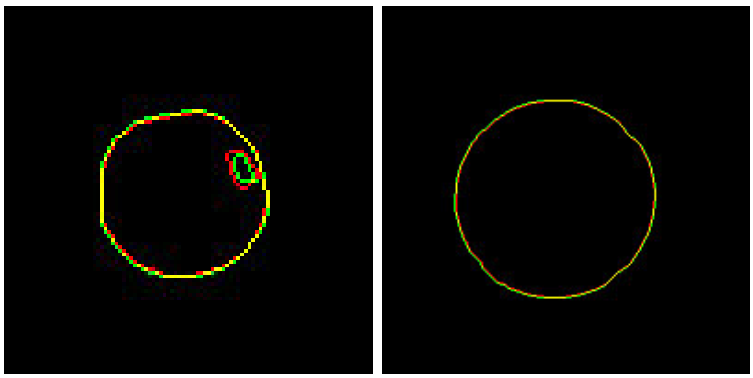

Figure 2.4: Comparison of LOG (dark gray line) and Canny (light gray line) edge detection applied to two acquired images of a tennis ball

2.1.3 Hough Transformation

As mentioned in chapter 2.1.2 the second step for the extraction of the tennis balls center in the image is to process the edge map with Hough transformation [HV62]. Hough transformation is used to detect geometrical shapes via their parameters. A coordinate system is introduced into the image in order to classify the shapes. By example all points on a line satisfy the equation $y = a * x + b$ which can be rewritten as $b = -x * a + y$ where each pair of related points (x_i, y_i) of

one line has the same parameters a and b. One problem arising by using the parameters a and b is that a approaches infinity as the line approaches vertical direction. In order to cope with this the representation of the line is changed to $x * cos(\theta) + y * sin(\theta) = \rho$ with the parameters θ and ρ where θ represents the direction of the line and ρ the distance from the origin. These parameters are again the same for all points of a single line. In case the line's equation is not given and has to be found the property of similar parameters ρ and θ for points belonging to a line is used. A parameter space is introduced and for each detected point of an edge map the possible combinations of θ and ρ are noted. Each of these combinations is used as a vote for this line parameter. After processing all the pixels of the edge map the parameter space consists of votes for all possible lines based on the edge map. All the pixels lying on the same line voted for the parameter combination representing this line. A local maximum in the parameter space for the lines property manifests this. Searching the local maxima of the parameter space leads to the most probable lines in the image.
The same procedure is also possible for geometric shapes of higher complexity. As the complexity increases also the parameters necessary to represent the object increase. Circles by example are represented by three parameters, their radius and the two coordinates of their center. Elipsoides are represented by four parameters, their center coordinates, the alignment angle and their principal axes length. As the number of parameters increase the parameter space increases and the complexity of the Hough transformation grows exponential. Due to this ways to decrease the number of parameters are searched.
In case of circle detection a known radius leads to a reduction to a two parameter problem for example. Other ways to reduce the complexity is the usage of edge detectors which output the direction of the edge (compare 2.1.2). As the line of a circle is orthogonal to the radial direction in every point the information about the direction of the edge in the related pixel can be used to plot the line on which the center of the circle must be. This simplifies the problem of circle detection again to a problem of two parameters. However, due to errors in edge direction detection (quantization errors) not even the parameter lines of the Hough transformation of a perfect circled edge map cross in one point. In case of erroneously detected edges this behavior is even boosted. The influence of the size-limited edge detection operators is minimized with the approach to determine the direction of the radial line in one point post and separated from the edge detection [SPV05]. The proposed methodology uses the final (loop-closed and filtered) edge

Figure 2.5: Example window along the edge line ($7x7$ *pixel*)

map of an image to detect the circles in the image. The errors due to wrong detected edge points and errors caused by the small size of the edge detector and the related quantization error are minimized by using a separate window around each pixel of the detected edge (compare Figure 2.5 for an example of such a window). For each quadratic window of uneven side length the containing edge is examined. If the edge segment equates a straight line, the the pixel is not used for determination of the circle as the line segment is not a valid segment of a circle. In case the segment equates an arc the two extreme pixels are used to determine the direction of the radial line which is then registered in the parameter space similar to the methods described

above. This modification of the Hough transformation leads to better and more accurate results. Still not all the radial lines intersect in the same point. Figure 2.6 shows the parameter space.

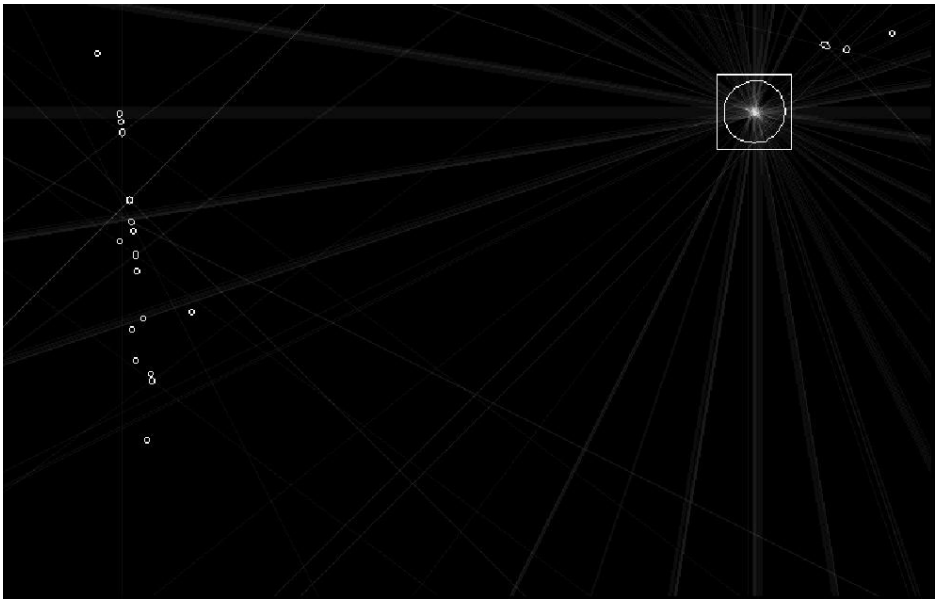

Figure 2.6: Parameter space

Also the detected edge, which is basis for the Hough transformation, is plotted. Figure 2.7 shows the regions within the square in Figure 2.6 in detail. The circle around the knoll in the middle is the related detected edge in the square in Figure 2.6.

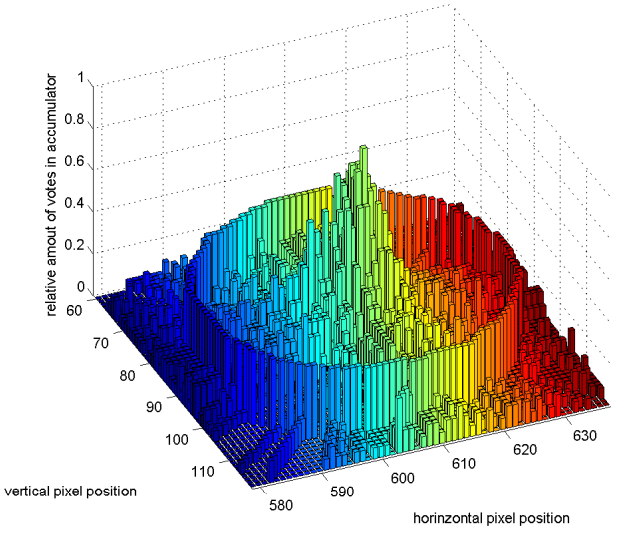

Figure 2.7: Detailed view of the parameter space

2.2 Trajectory Prediction

Predicting the future has always been a challenge for mankind. Predicting the future trajectory of a flying object is a very well researched field. One main driving force for this is the military application in form of ballistics. The prediction is always based on a known set of parameters of a model on which the trajectory calculation is based. Different related models take different forces that influence the flight into account. As the number and complexity of considered forces increases, also the calculation complexity of the trajectory increases. The first part of this section is focused on the forces that influence the flight [5] and introduces common models. The second subsection will deal with ways to find the ideal trajectory for a given set of positions.

2.2.1 Models for Trajectory Calculation

In order to calculate or estimate the behavior of a given object the objects behavior is observed first. Then a model for the behavior is introduced and the accuracy of the model is validated by comparing experimental results with calculation of the model. If the model does not describe the behavior well enough the model is refined. Basically all physical laws are built up on this and in case unexplainable behavior is found new models are searched.
This is also valid for the calculation of trajectories. The main forces that are used to model the flight of a small projectile are shown in Table 2.4 [5].

Mass Forces	**Aerodynamic Forces**
Gravity	Drag
Coriolis Force	Lift
Centrifugal Force	Magnus
	Pitch Damping
	Transversal Magnus

Table 2.4: Forces influencing an objects flight

The most simple model is a model only taking the gravity into account. Calculation based on this model can be done in each spatial direction separately. The equations in case of an 3D system and alignment of the z-axis with the direction of the force of gravity are

$$v_x(t) = v_{x0} \tag{2.1}$$
$$v_y(t) = v_{y0} \tag{2.2}$$
$$v_z(t) = v_{z0} + g * t \tag{2.3}$$

where v_{i0} equates the initial velocity in direction i and g equates the acceleration by gravity. The resulting position as a function of time in each direction ($x(t)$, $y(t)$, $z(t)$) are

$$x(t) = x_0 + v_{x0} * t \tag{2.4}$$
$$y(t) = y_0 + v_{y0} * t \tag{2.5}$$
$$z(t) = z_0 + v_{z0} * t + g * t^2 \tag{2.6}$$

This model is suitable for very slow movements. When dealing with higher object velocities the aerodynamic drag force **F** has a significant influence. The model for the drag is a quadratic one

and the equation describing the drag force on a moving object is

$$\mathbf{F} = -\frac{\mathbf{v}}{|\mathbf{v}|} * \frac{\rho * c_W * A}{2} * |\mathbf{v}|^2 \tag{2.7}$$

which is depending on the velocity $\mathbf{v}$, the air density ρ, the air drag coefficient c_W and the cross section surface of the object A. This force is nonlinear dependent on the velocity and therefor it is not possible to consider the spatial coordinates separately. The flight trajectory is described by

$$\dot{\mathbf{v}} = -\frac{\rho * c_W * A}{2 * m} * |\mathbf{v}| * \mathbf{v} \tag{2.8}$$

$$\mathbf{g} = [0, g, 0]' \tag{2.9}$$

$$\mathbf{a} = \dot{\mathbf{v}} + \mathbf{g} \tag{2.10}$$

$$\mathbf{v} = \mathbf{v}_0 + \int_{\tau=0}^{t} \mathbf{a} * dt \tag{2.11}$$

$$\mathbf{p} = \mathbf{p}_0 + \int_{\tau=0}^{t} \mathbf{v} * dt \tag{2.12}$$

$$\mathbf{p} = \mathbf{p}_0 + \int_{\tau=0}^{t} \left[\mathbf{v}_0 + \int_{\tau=0}^{t} \left(\mathbf{g} - \frac{\rho * c_W * A}{2 * m} * |\mathbf{v}| * \mathbf{v} \right) * dt \right] * dt \tag{2.13}$$

where $\mathbf{p}$ is the position in 3D space, $\mathbf{g}$ is the gravity constant g put into a vector where the alignment of g is considered. 0 as index expresses the initial value of this variable. Due to the nonlinear properties the velocity can only be computed iterative as no closed solution exists. In order to enable this computation Equation 2.11 is approximated with

$$\mathbf{v} = \mathbf{v}_0 + \sum_{n=0}^{k} \mathbf{a}(n * t) * \Delta t \tag{2.14}$$

$$\mathbf{a} = \mathbf{g} - \frac{\rho * c_W * A}{2 * m} * |\mathbf{v}| * \mathbf{v} \tag{2.15}$$

and Equation 2.13 is approximated by

$$\mathbf{p} = \mathbf{p}_0 + \sum_{n=0}^{k} \mathbf{v}(n * t) * \Delta t \tag{2.16}$$

Where Δt is the time interval between two calculations. The smaller the time step Δt is the smaller the errors the better the model resembles the real flight. As $\mathbf{a}$, $\mathbf{v}$ and $\mathbf{p}$ have to be calculated for each instant it is more effective to calculate all three variables in one step for each instant recursive. This results in the following equations for the calculating the models trajectory

$$\mathbf{a}_t = -\frac{\mathbf{v}_t}{|\mathbf{v}_t|} * k * |\mathbf{v}_t|^2 \tag{2.17}$$

$$k = \frac{\rho * c_W * A}{2 * m} \tag{2.18}$$

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t * dt \tag{2.19}$$

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{v}_t * dt \tag{2.20}$$

With the air density ρ, object air drag coefficient c_W, cross section surface A, object mass m and object velocity $\mathbf{v}$. Nonlinearity of this model and the property that the flight can not be

described for each spatial coordinate separately complicates the application of this model in case the initial parameters of the flight ($\mathbf{v}_0$ and c_W) are not known. Estimating these parameters from a set of measured flight positions of a thrown object is very hard as the initial parameters have to be approximated and the positions can only be calculated afterwards. After comparison of these calculated positions with the measured ones the applicability of these set of initial parameters can be evaluated.
Based on the model described above it is also possible to separate the movement along the spacial directions [BFK08] which introduces small errors but simplifies trajectory calculation. Therefore the movement along each direction is described by the following equations in case gravity is aligned with the y-direction:

$$\dot{v}_x = -k * \frac{v_x}{|v_x|} * v_x^2 \tag{2.21}$$

$$\dot{v}_y = -k * \frac{v_y}{|v_y|} * v_y^2 + g \tag{2.22}$$

$$\dot{v}_z = -k * \frac{v_z}{|v_z|} * v_z^2 \tag{2.23}$$

Using this separated model for the flight introduces an error as the influence of the air drag is nonlinear. This error is in scope in Section 4.3 The magnitude of the error depends on the proportion between the velocity components. In case the movement is aligned with one spatial direction no error is introduced. Due to gravity this circumstance is only valid for a movement in direction of gravity. As this is not the case in general this error has to be considered.

2.2.2 Fitting the Trajectory into the Measured Data

Compared to the military aspect of trajectory (and impact position) forecast mentioned in the introduction of this subsection the problem of finding the impact position in this particular case differs by the fact that the initial launching (velocity, position) parameters are not known. Information about the flight of the thrown object is only available throughout the flight. This fact demands that the prediction is based on the measured positions over time or that the impact position prediction is based on initial parameters which still are derived from the position of the object over time.
The number of positions acquired through out the flight is by far greater than the order of the polynomial model used or the number of parameters used to describe the flight trajectory. Due to this it is generally not the case that the fitted trajectory corresponds with the determined positions exactly. A trajectory minimizing the related error function has to be found. One common error function [LS86] is shown in Equation 2.24 where $E(p)$ is the error function which is dependand on the found polynomial ($p(x_i)$) and related data (f_i).

$$E(p) = \sum_{i=0}^{N} [p(x_i) - f_i]^2 \tag{2.24}$$

This approach was named method of least squares by Lagrange and is widely used for fitting-tasks. Algorithms to solve this minimization problem are available for linear as well as nonlinear problems. Linear and nonlinear in this regard means that the coefficients of the equation chosen to model the data is linear (or nonlinear) in the coefficients. While the linear least square approach is used for fitting the polynomial model presented in Section 4.1 the nonlinear leas square approach is used for fitting the spatial separated model presented in Section 4.3.

2.3 Prediction Accuracy Determination

Taking a look at the target application of the trajectory prediction the goal is to establish a fast and reliable transportation system (compare Chapter 1). The natural quality measure for such a system is the catching or drop rate. Factors affecting this parameter are the throwing device, thrown object, catching device and prediction algorithm. As the prediction algorithm is one of the influencing factors the catching rate can be used to evaluate it's applicability. In order to establish this a catching device [FBWW+07] and positioning equipment similar to a gantry robot [BFK08] is necessary. Figure 2.8 shows these two components used by Barteit. Based on this

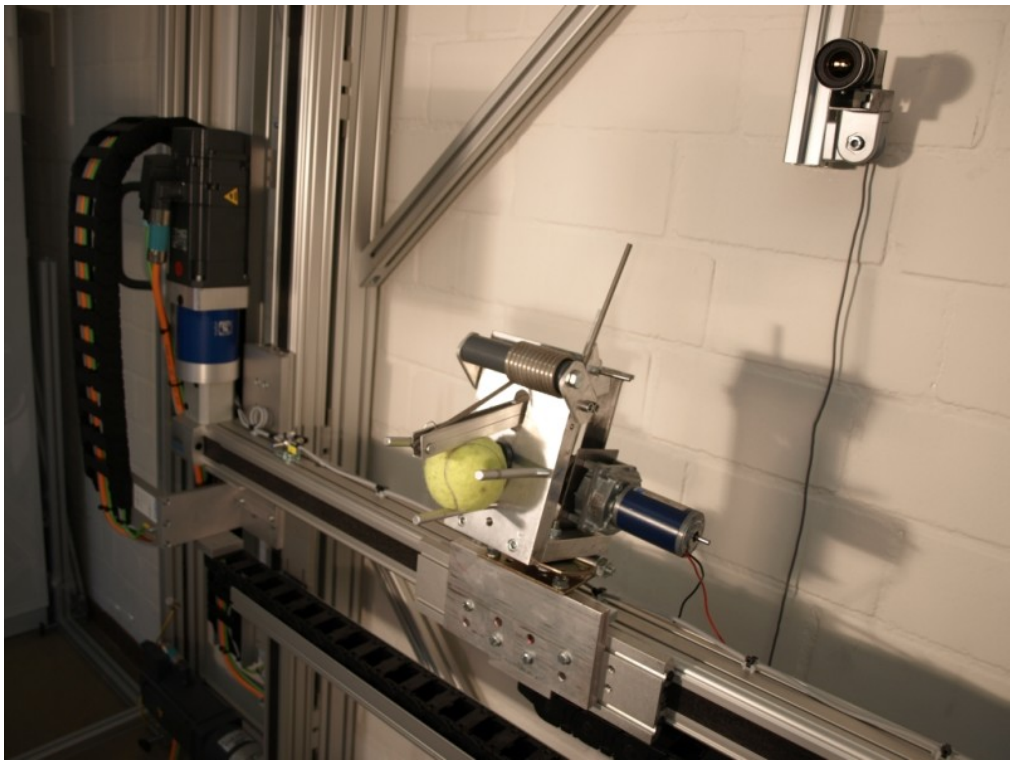

Figure 2.8: Catching device and positioning equipment (gantry robot); Source: Barteit [DBP09]

equipment the binary information caught/not caught can be used to calculate the catching rate. The remaining influence factors are the throwing device, thrown object and catching device. As only the catching device is not required to throw an object excluding it's influence on the evaluation results in a higher dependency on the prediction algorithm. Also Barteit (in [DBP09]) states that metering the prediction algorithm based on the catching rate is not well suited to outline the precision of the prediction as an increased size of the catching device easily improves the catching rate. He also presents a way to use a ensemble of light barriers and a video camera (Figure 2.9) to determine the position of a thrown object within a pre-defined plane and calls it a non-invasive position detection. Downsides of this method are that the behavior of the light barriers can not be modeled simply and that the lag of the optical system introduces a further error. Another downside is that a tennis ball passing through the detection plane between two light barriers is detected later as the ball needs to move further through the frame in order to be detected. The final accuracy of this system is estimated by 1 mm. The system used as position verification system in this work is similar to the one used by Barteit but does not allow the thrown

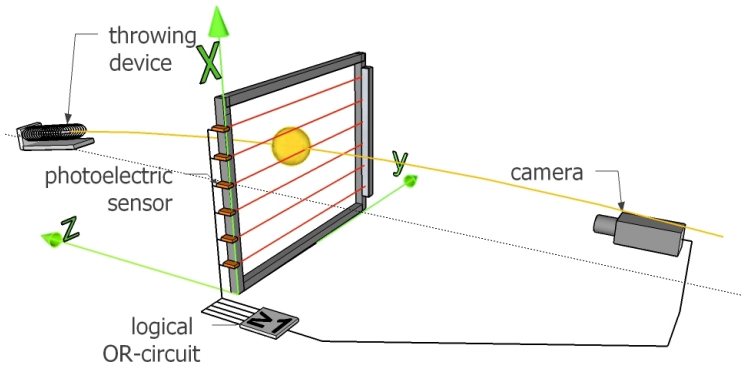

Figure 2.9: Camera used to detect trajectories interception with a plane; Source: Barteit [DBP09]

object to pass through the measuring plane thus it is an invasive position detection. The system used is described in detail in Section 5.1.

3 Setup for Video-Based Feature Extraction

This chapter deals with the setup of the equipment used to predict the interception position. The environment for the prediction consists of a throwing device and a position verification system. These two systems build the base for the research and experiments regarding the prediction. The sensor system used for acquiring the necessary data to predict the future flight of the tennis ball consists of a stereo vision system based on two separate cameras. This stereo vision system is described in detail in the following Subsection 3.1. Using image data from cameras to measure objects positions or sizes requires calibration of the vision system. Out of this calibration data the positions of the two cameras relating to each other is extracted and the properties of the cameras (lenses and image sensor) are determined. Section 3.2 deals with this calibration in detail and as the calibration of the verification system described in Section 5.1 is done simultaneously based on the same data also Section 5.1.2 deals with this topic.
Feature extraction of the image data (Section 3.3) as well as stereo triangulation (Section 3.4) close this chapter.
Prior to dealing with the further details of the setup the external parameters have to be clarified. The distance between the throwing device and the position detection plane is set to equal 3 m in order to allow comparison of the prediction results with prior-done, cooperative work [BFK08] based on a monocular approach. The velocity of the ball as a second parameter has to be approximately 10 m/s. Variation of this velocity is desired in order to check the prediction algorithms usability in a broader range. A hand operated throwing device which is shown in Figure 3.1 fulfills these requirements and is used for the work described in the next chapters.

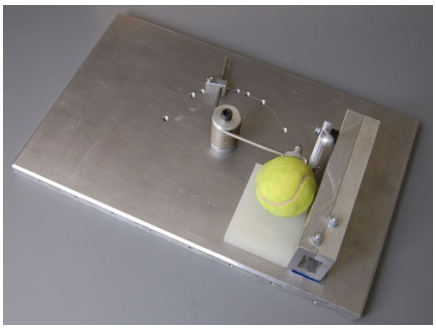

Figure 3.1: Simple throwing device

Basic informations about the flight parameters of the thrown ball and the throwing device permit to start with the description of the vision system in detail in the following section.

3.1 The Stereo Vision System

Vision systems offer a cost-effective way to acquire a huge amount of versatile of data. Extracting the information of interest on the other hand is a tough task. Prior to this extraction the camera has to be positioned appropriately.
Different requirements on the main attributes (frame rate, resolution, interface, price) are discussed. Also the selection of a suitable lens, containing a discussion of advantages and disadvantages of different focal lengths, will be topic of subsection 3.1.2.
The basic coordinate system also called world coordinate system is introduced in Figure 3.2. In

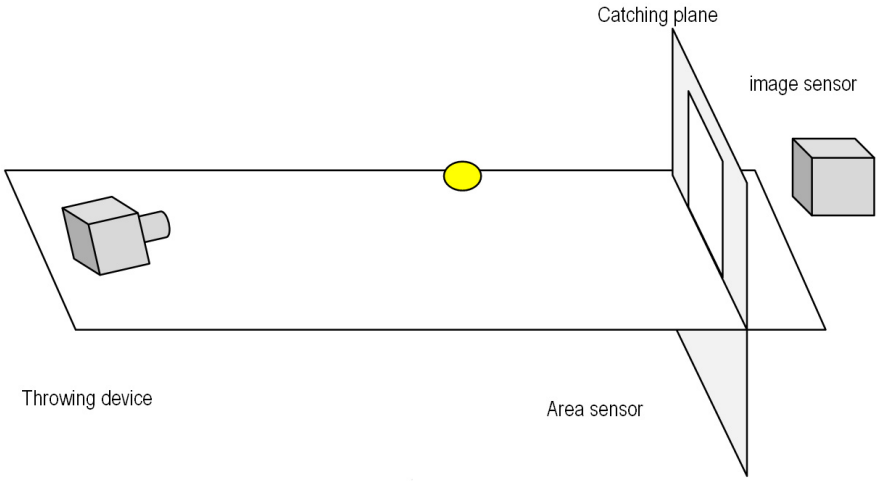

Figure 3.2: World coordinate system

order to align the x- and y-axis of the verification system with the x- and y-axis of the world coordinate system and respect the orthogonal property the the z-axis of the world coordinate system has to point away from the throwing device.

3.1.1 Camera Position Discussion

Basically stereo vision systems can be classified in divergent, parallel and convergent vision systems. This classification takes the direction of the optical axes in account. A main difference is lying in the overlapping visual field which is shown in Figure 3.3. Judging from Figure 3.3 a

Figure 3.3: Divergent, parallel and convergent visual fields

divergent system can be excluded from further discussion as a overlapping of the visual field is necessary for the tracking of the ball as the information of the right and the left picture have to be merged into a 3D position information. Parallel systems have the advantage that keystone

distortions which occur when using divergent or convergent vision systems do not occur [11]. Keystoning is related to the twist between the two optical sensors of the vision system. It is specially a disadvantage when stereo vision systems are used to create distortion free 3D images. In the context of this work where the center of an spheric object has to be obtained from the image data this effect can be neglected. When discussing the lateral distance between the two cameras the later application of stereo triangulation (Chapter 3.4) has to be taken into account. The basic concept is that two light rays from an object feature to the respective cameras are intersected (Figure 3.4). With known camera positions the position of the object feature can be determined.

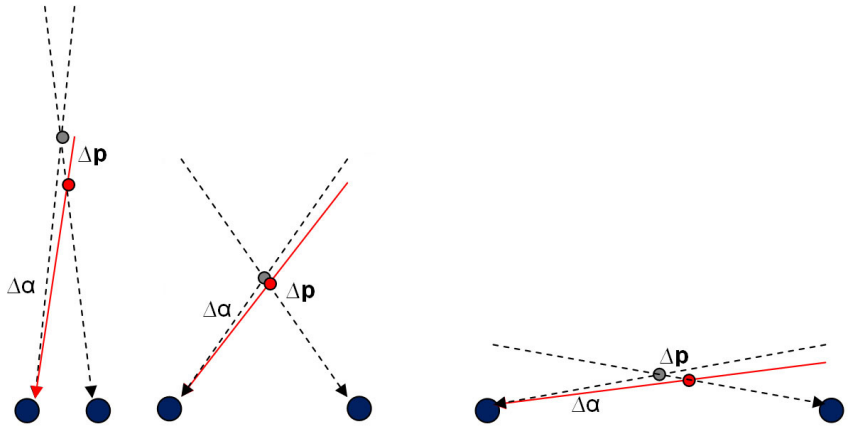

Figure 3.4: Resulting measurement error for three different lateral camera distances

The angle between the two light rays has an influence on the impact of small measurement or digitalization errors. Figure 3.4 illustrates three different camera distances and the impact of a small miss measurement of one ray's direction. The best error rejection is accomplished when the rays intersect at an angle of 90 degrees [SSL01]. Moving from a two-dimensional model in Figure 3.4 to a three-dimensional model, which resembles the target application, it is obvious that the movement of the ball along the more than 3 m long trajectory makes it impossible for a fixed camera setup to fulfill this condition. It is still possible to incorporate this condition in a different way.

By projecting the ball position onto the xy-plane, which is equal to the catching plane, the prior mentioned condition can be fulfilled again. This means that the cameras x- and y-cooridnate are related in a way that the cameras focal points and the expected position projection form a perpendicular triangle. As the projection of the balls position onto the catching plane moves and the accuracy in the final flight phase is more significant than in the early flight phase the cameras are related to the expected impact position of the ball, which is determined by averaging over multiple throws.

This conclusion fixes the position of the cameras in two dimensions (x-position and y-position are related) but still raises the question about the position in the z-direction. When taking Figure 3.3 into account a position behind the catching plane (positive z-coordinate) leads to a visual field which contains the ball during the complete flight, maximizes the visual field and as a result also the time of observation. Keeping the application in mind a catching device has to be positioned in the catching plane in order to catch the thrown object. If this movement is in the area of the visual field of the camera it might influence the object detection and disturb the interception position detection. Positioning the cameras in the caching plane is realistic for the goal application as the area behind the catching plane (seen from the objects origin) will most probably be used

to process the object.
The final mounting positions of the cameras are shown in Figure 3.5. As the cameras are mounted approximately 370 mm above the average impact position on the plane the distance between both cameras is approximately 750 mm in order to obtain the orthogonal ray intersection rule described above. The exact positions and relation of the two cameras obtained during the camera calibration and used for stereo triangulation is presented in Table 3.8 for full resolution and Table 3.10 for halve resolution.

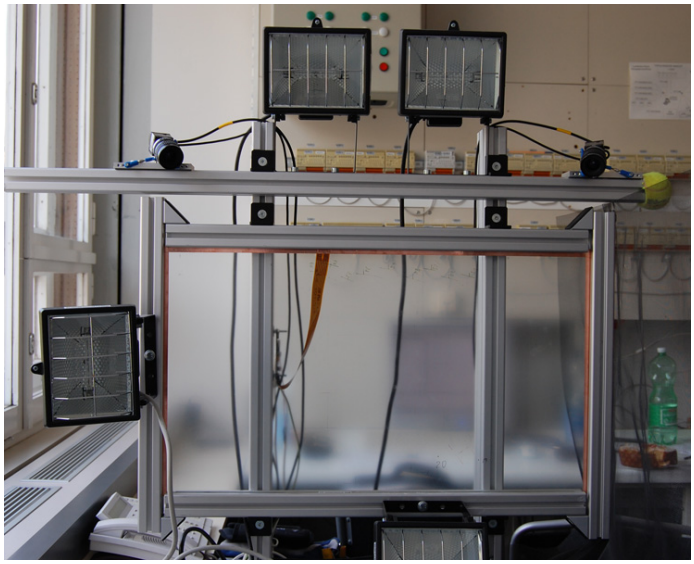

Figure 3.5: Camera positions

3.1.2 Camera and Optics Selection

A broad range of industrial cameras (Figure 3.6) is available commercially. Models differ by

Figure 3.6: Sample industrial camera

their main properties, namely color space, frame rate and triggering, resolution, interface, optical interface and price. These properties and their influence on the attainment of the task set up in Section 1.2 are discussed below.

One key advantage of a color camera for using it to obtain the position of a thrown tennis ball is that the color of the tennis ball significantly differs from usual backgrounds. The intense yellow allows better recognition of the ball than in a black/white image. The downside on the other hand lies in the triple amount of data which has to be transfered from the camera to the host system. Interface bandwidth constraints limit the product of frame rate, resolution and color information.
Cameras with frame rates ranging from 20 to more than 1000 frames per second are available. Using higher frame rates increases the information density (in this context the number of frames per throw) but the bandwidth requirements as well. The theoretical bandwidth limit of cameras connected to the USB Bus is 40 MB/s which is the net bandwidth of the USB 2.0 specified 480 Mb/s.
Another attribute influencing the bandwidth requirements of the camera interface is the image resolution. Industrial cameras with resolutions of up to 2560×1920 pixels are available commercially. Multiple techniques are used to reduce the native resolution of a camera. One common way is by consideration of an area of interest only which reduces the field of view by the same magnitude as the resolution. The number of pixels per meter at a given distance from the camera is not influenced at all, only a subsection of the native sensor pixels is read out. Another common way to reduce the resolution is called binning. The combination of information from multiple pixels is used to capture an image at lower resolution. The field of view is not altered in this case but the number of pixels per meter at a given distance is reduced by the same factor as the resolution. The third common way for resolution reduction is subsampling. Some pixels are skipped during sensor read-out and the remaining pixels information is sent to the host system. Similar to binning the field of view is not altered but the resolution is changed. Resolution reduction enables to study the impact of the resolution on the prediction result.
For synchronous stereo image acquisition a common trigger mode is necessary. Using an external hardware trigger allows synchronous acquisition as well as small step frame rate scaling analysis (Section 5.2.2).
Selecting a suitable image resolution also has to be done based on the trade-off between interface bandwidth, price and information density on the other hand. In relation with the interface bandwidth also the higher price of higher resolution and/or higher frame rate cameras has to be mentioned. Also the selection of the lens has to be considered here. The combination of the lens with the sensor has to ensure that the size of the tennis ball on the throwing device as well as in the final part of the flight is a reasonable number of pixels.
Industrial standard interfaces are USB, Firewire, Ethernet, Cameralink and many more. These interfaces differ by their maximum bandwidth, price, usability for streaming applications like video transmission. USB is by far the most common interface but has downsides regarding streaming applications as well as the maximum bandwidth. Firewire is more video-oriented but the market penetration regarding industrial cameras is very low. This results in a limited amount of choices of Firewire cameras. Ethernet interface is a common interface for communication applications and also used for industrial video transmission. Advantages in regards of the bandwidth are countered by disadvantages in cost and the fact that similar priced Ethernet versions of USB cameras use the same internals and have similar bandwidth restrictions as USB cameras.
Regarding the optical interface two types dominate the industrial market. These are C and CS interfaces. CS-mount offers small size and lower wight but also lower light sensitivity. The range of available C-mount lenses outnumbers the number of CS-mount offers. For low focal lengths (lower than 6 mm) only the smaller CS-mount objectives are available.
Based on the arguments presented above the μEye UI-1220SE-M-GL cameras was chosen to build up the stereo vision system. The main properties of the camera are shown in Table 3.1. After

Color space	monochrome
Interface	USB 2.0
Resolution	752×480 pixel
Resolution reduction	binning
Sensor size	1/3 ”
Maximum frame rate	87 fps
Focal length	≈ 6 mm
Trigger sources	Software, Hardware
Lens interface	CS-mount
Price (EUR)	≈ 440

Table 3.1: Properties of the selected μEye UI-1220SE-M-GL

selecting the cameras for the stereo vision setup the lenses can be selected. The main parameter of a lens is the focal length which influences the relation of object size, object distance and image size. Other parameters are the distortion, supported sensor size, maximum aperture and many more. Based on the image size of the used camera (Table 3.1) the field of view of lenses with different focal lengths are shown in Table 3.2. These data is based on a working distance of 3 m

Focal length (mm)	angle of aperture (°)	min. observable z-distance (mm)	Ball size (pixels on TD)
$4,2$	$64,8$	193	138
6	$44,1$	448	282
$6,5$	$40,5$	507	330
8	$32,7$	670	501
$8,5$	$30,7$	720	566

Table 3.2: Impact of different focal lengths on image size and field of view

and a camera position 370 mm above and with 370 mm lateral displacement from average impact position in case the ball impacts on this average position. The left boundary of the right cameras field of view is aligned with the left border of a 500 mm wide window including the throwing device. This is valid for the right boundary of the right camera with the right border respective. Increasing distortion with decreasing focal length also has to be considered. A 6 mm Pentax lens was chosen and is used for both cameras throughout the presented work. This lens offers a large field of view (up to ≈ 450 mm from the impact position detection plane) and a suitable number of pixels of the tennis balls image at it's largest distance from the camera (in 3 m distance at the throwing device). Lens specific properties are shown in Table 3.3.

Name	H612A(KP)
Category	1/2”
Focal Length	6 mm
Mount	C
Weight	125 g
Price	EUR 159

Table 3.3: Pentax 6 mm focal length lens specification

3.1.3 Software-Interface, Triggering and Illumination

The reference software "uEye Demo" provided by IDS allows to access all parameters of the camera. Starting with the specification of the USB Bus and the pixel clock as a main attribute. This pixel clock equals the bandwidth the cameras uses to send image information. The limiting factor is the USB Master the camera is connected with. In case of the used PC workstation (detailed specification in Table 3.4) which is based on the AMD Phenom X4 processor and AMD

Mainboard	ASUS M3A78-T
Chipset	AMD 790GX
CPU	AMD Phenom X4 9750
RAM	4 GB DDR2 1066
HDD	3x Western Digital Caviar Blue 80GB RAID0
Graphics	ATI HD 3300 (primary)
	ATI X1650 (secondary)
Operating system	Windows XP Professional SP 3
uEye Software	uEye 3.32

Table 3.4: Specification of the used PC workstation

790GX Chipset the pixel clock could be set up to 39 MHz which is 1 MHz short of the maximum which also means that 39 MB/s of the theoretical maximal 40 MB/s are used. This could only be achieved after each of the two cameras was connected to one of the two separate USB-root hubs. Also automatic CPU throttling (AMD Cool'n'Quiet) had to be disabled on the board in order to maximize the available bandwidth on the USB-bus. A screenshot of the software is shown in Figure 3.7. The screenshot shows a live image acquisition and the camera interface

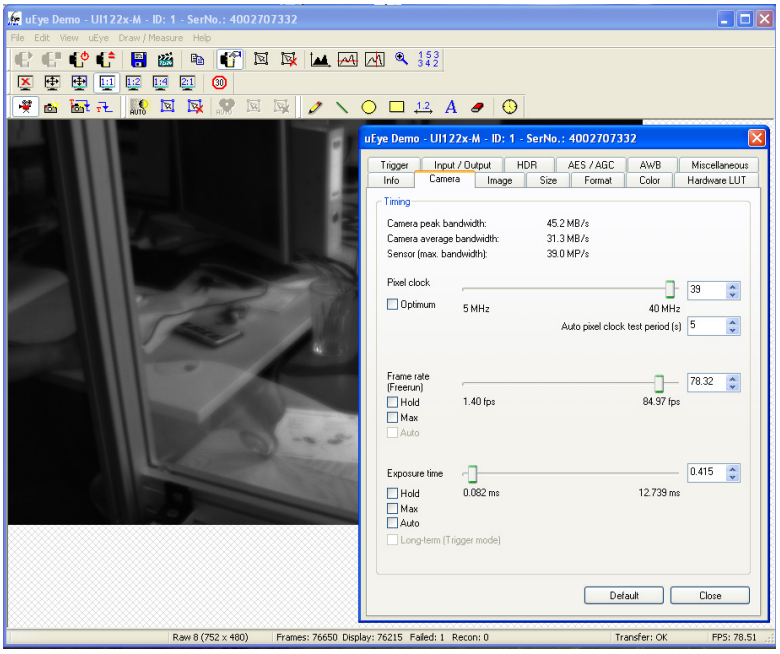

Figure 3.7: Sample screenshot of IDS "uEye Demo" software

properties. Other main parameters which were set up in the software are shown in Table 3.5. Triggering of both cameras has to be synchronous. The only possibility to do so is to trigger both

Exposure time (ms)	0.415
Master gain	1.0
Triggering	Hardware (rising edge)
Pixel format	Direct raw data (Y8) / Hardware

Table 3.5: Camera setup - video parameters

cameras by hardware as shown in Table 3.5. Due to problematic hardware interfacing addition external hardware is used to trigger both cameras. This hardware consists of a ATMEL STK-500 starter-kit and a AT90LS4433 microcontroller to execute the triggering. The software running on the microcontroller allows to select frame rates of 50, 60, 70, 80, 100, 120, 140 and 160 fps. Two different versions are used to ensure a high accuracy over a range of 50 to 80 fps and 100 to 160 fps respective. By pressing a button an individual impulse is output to the cameras and the reference background image is acquired (compare Section 3.3.1). Pressing the button for a second time starts the continuous impulse output at the preset frame rate / frequency. The starter-kit including the cabling using specific trigger connectors [3] provided by IDS is shown in Figure 3.8. For both cameras one instance of this dialog is started an the individual files created. After

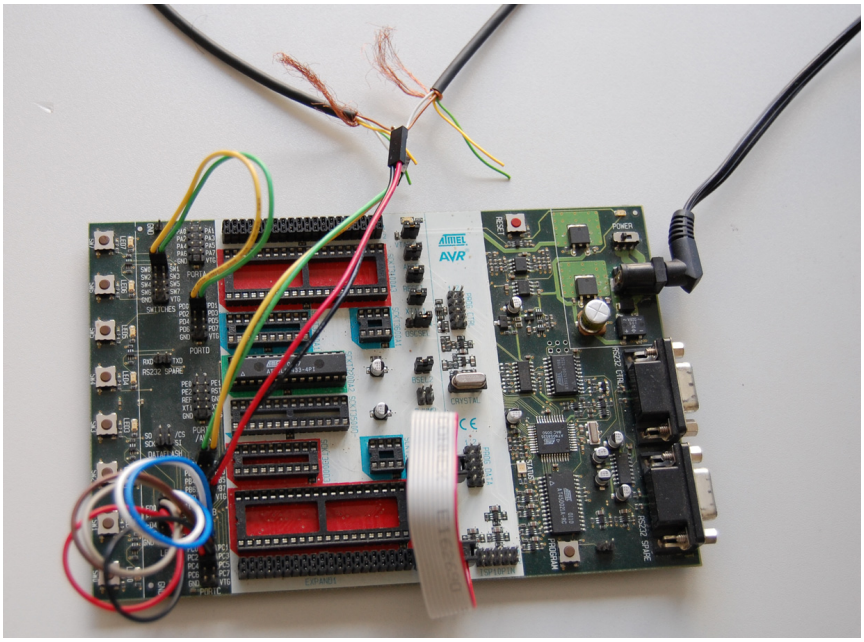

Figure 3.8: Starterkit and trigger wires

putting both instances in the recording mode first the reference frame is recorded and afterwards the video acquisition is started. If the USB bus is saturated and frames are dropped the number of dropped frames is displayed in the dialog. Only video data without dropped frames was used for the evaluation of the prediction algorithm presented in Chapter 4.
By using addition light (4x 500 W floodlights, compare Figure 3.5) usage of the short exposure time of 415 μs (Table 3.5) is possible. Motion blur effects are invisible and negligible. Sample images captured with the right and left camera are shown in Figure 3.9. Based on these images the feature extraction described in Section 3.3 is done and the center of the ball in the image is detected. Stereo triangulation (Section 3.4) leads to the position of the ball in space which is the basis for the prediction algorithms (Section 4) to work on. But prior to the further processing

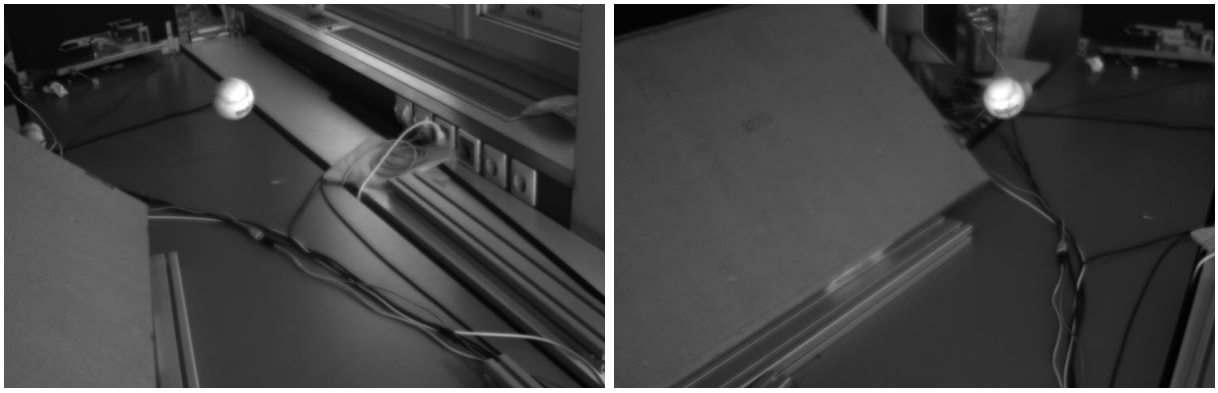

Figure 3.9: Right and left sample image

of the information the stereo vision setup has to be calibrated in order to obtain the parameters necessary to use it for spacial measurements.

3.2 Camera Calibration

Using cameras to meter object's sizes or object's positions requires knowledge of the parameters of the cameras. Parameters of the camera model are called intrinsic parameters [2]. The internal parameters of the model used in the Camera Calibration Toolbox for Matlab [2] are:

- Focal length
- Principal point
- Skew coefficient
- Distortions

while the extrinsic parameters are:

- Rotations
- Translations

The intrinsic parameters describe the relations between the objects in the visual field of the camera and the resulting image. Extrinsic parameters describe the relation between separate coordinate systems with reference to the camera internal coordinate system (aligned with the plane of the image sensor) by using a rotation and a translation operation. Extrinsic parameters are also used to describe the relation between the two cameras of the stereo vision setup.
In order to determine these parameters the Camera Calibration Toolbox for Matlab [2] is used. This toolbox, similar to other toolboxes available, calculates these parameters based on a number of images taken from a reference calibration sheet (shown in Figure 3.10). Looking like a chessboard, the calibration sheet allows to extract the corners of the pattern in the image automatically. Based on the four extreme corners in combination with the known dimensions of the pattern (compare Figure 3.12) and the extracted corners (compare Figure 3.13) in the image

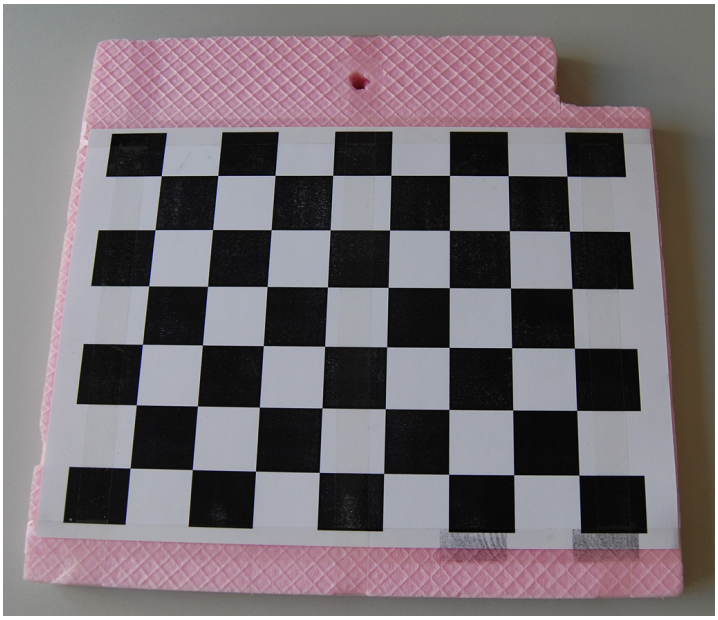

Figure 3.10: Calibration sheet

the relation of the calibration sheet to the camera is calculated. The first calibration step is initialization which is followed by nonlinear optimization. The nonlinear optimization minimizes the total reprojection error (in the least squares sense) over all calibration parameters [2]. There are 9 intrinsic parameters (2 for focal length, 2 for principal point, 5 for distortions, skew is not calculated) and 6 extrinsic for each calibration sheets. In order to minimize the error of this calculation a larger number of images of the calibration sheet in different positions has to be used. This first calibration step which is done for each camera separately yields in the intrinsic parameters of the camera and the extrinsic parameters of each of the 67 calibration sheets. Gradient descent optimization is used to find the optimal set of 411 parameters.
Merging two calibration sets based on images of the respective calibration sheets as seconds step of the calibration results in optimized intrinsic parameters of both cameras (as the additional information of the second camera is used to minimize errors) and the extrinsic parameters of the two cameras related to each other. Also this information is obtained by gradient descent optimization and is the key to use a stereo vision setup to measure dimensions or positions of objects.
Following the calibration of the stereo vision system is described in detail based on the full resolution. For the scaling analysis presented in Section 5.2.2 a second calibration for the reduced resolution of 286 x 240 pixel is necessary. The result of this second calibration are presented in Section 3.2.2.

3.2.1 Calibration at Full Resolution (572 x 480)

The quality of the calibration is one important factor on the measurement results acquired by a stereo vision setup. If more calibration sheets are used the parameters of the vision setup can be calculated more exact. But not only the number of calibration images is the key to a high quality result. The positions and rotations of the calibration sheets in the visual field of the cameras have to be as variable as possible in order to extract the parameters on a large set of data.

In order to speed up the acquisition of the calibration images two calibration sheets were used per picture. Each picture was then used twice for the calibration, once based on the right calibration sheet, the other time on the left sheet. As the images of the calibration sheet of the left and right camera have to be related both images were taken at the same instant via the hardware trigger of the cameras. Figure 3.11 shows the related right and left image of two calibration sheets. Overall,

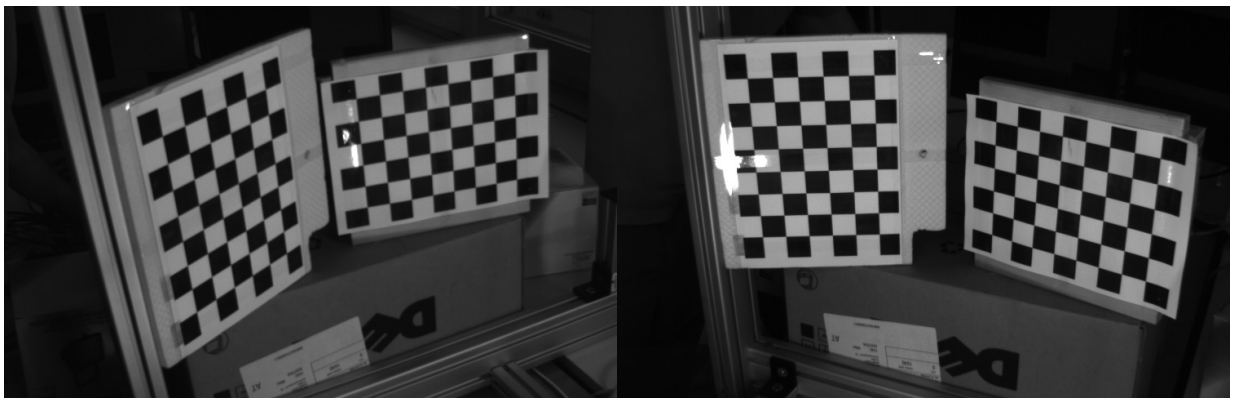

Figure 3.11: Set of related calibration images

images of 67 calibration sheets in different positions were taken for the camera calibration. 56 of them were similar to the images shown in Figure 3.11 while 11 images were especially used to calibrate the verification system as well. This extended calibration is topic of Chapter 5.1 and especially Section 5.1.2. These 11 images can be considered like the remaining 56 images at this stage.
The corner extraction function of the calibration toolbox demands the user to manually select the four extreme corners of the calibration pattern. Within a window of selectable size the toolbox searches for the corner. Inputing the the number of squares along each direction results in lattice drawn on the calibration sheet. This lattice is shown in Figure 3.12 and purely based on the four

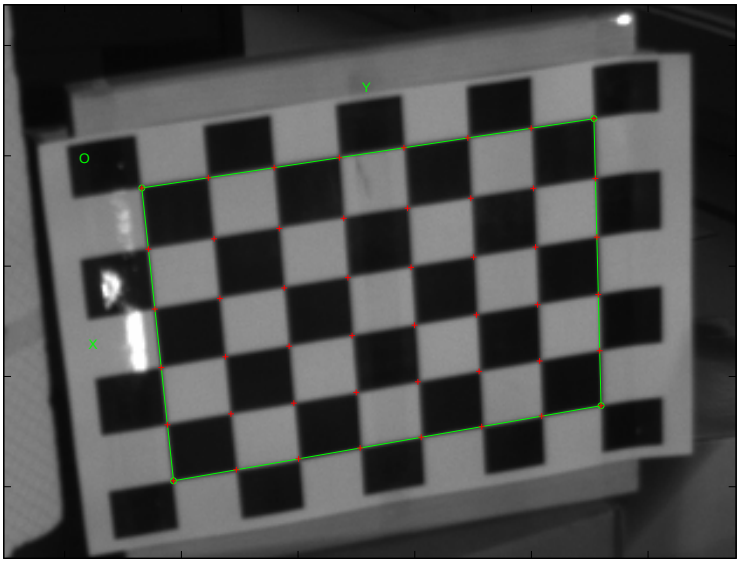

Figure 3.12: Lattice of corner extraction

extreme corners and the number of squares along each direction. As a second step each local

window around the crosses of the lattice (in Figure 3.12) are processed and the real corner of the calibration pattern is found. These real corners are shown in Figure 3.13.

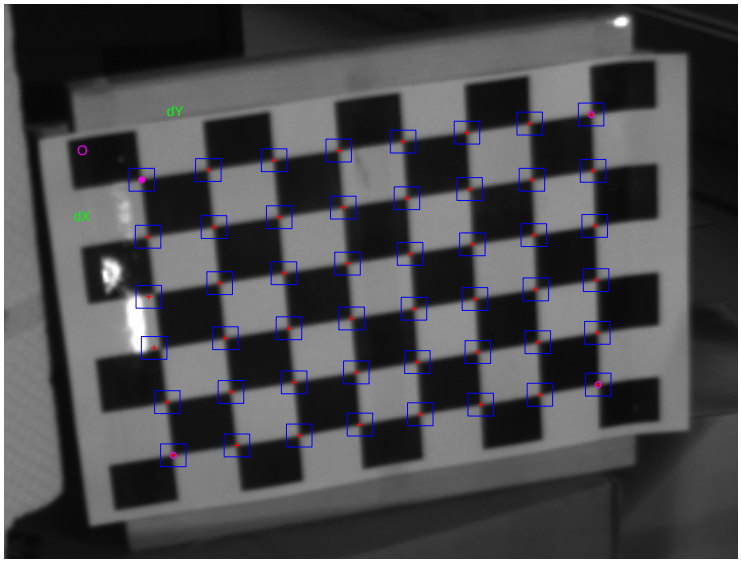

Figure 3.13: Extracted corners

The displacement between the real corners and the corners of the lattice the position of the calibration sheet in relation to the camera is calculated. The result of this calculation for the image presented in Figure 3.13 and Figure 3.12 is shown in Figure 3.14. Repeating these steps for all the

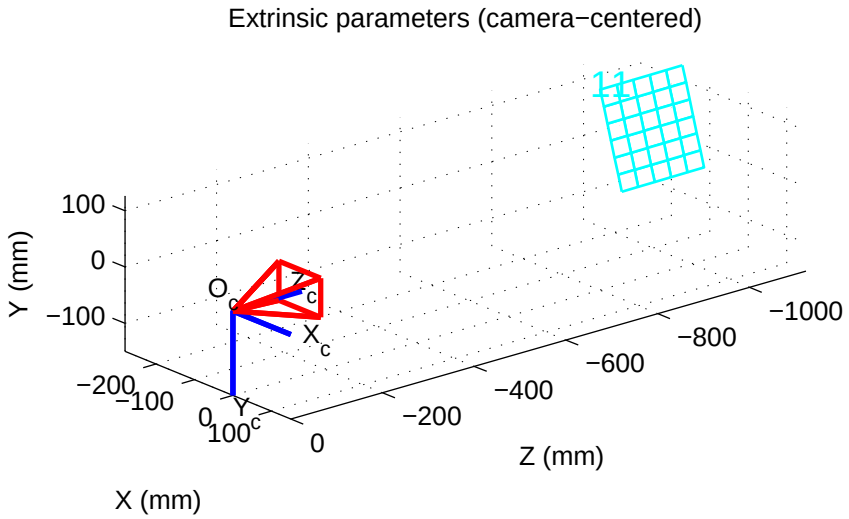

Figure 3.14: Relation of the calibration sheet to the camera

images of the calibration pattern results in a dataset containing all the positions of the patterns related to the camera.
The intrinsic parameters of the left and right (processed exactly like the left camera) camera extracted are shown in Table 3.6. The numerical errors presented are approximately three times the standard deviations (for reference). The separate stereo calibration function of the Camera

	Left camera	Right camera
Focal Length (Numerical errors)	[1036.44268 1032.95088] (± [5.23336 5.43448])	[1022.05361 1019.83780] (± [4.61866 4.75193])
Principal point (Numerical errors)	[412.10509 254.37995] (± [10.16412 9.78845])	[392.06675 245.59170] (± [9.19424 8.40561])
Skew (Numerical errors)	[0.00000] (± [0.00000])	[0.00000] (± [0.00000])
Radial distortion (Numerical errors)	[−0.20737 0.26752 0.00000] (± [0.01675 0.11394 0.00000]	[−0.21793 0.19177 0.00000] (± [0.01402 0.07958 0.00000]
Tangential distortion (Numerical errors)	[0.00126 0.00193] (± [0.00177 0.00193]	[0.00050 0.00220] (± [0.00141 0.00151])

Table 3.6: Intrinsic parameters of left and right camera at full resolution

Calibration Toolbox for Matlab allows to merge the two calibration sets into a common calibration for both cameras. The positions and rotations of the 59 camera calibration patterns are used to calculate the translation and rotation of the two cameras relating each other. Also the calibration of each individual camera is refined as the calibration pattern position information acquired by the second camera is used as well. Figure 3.15 shows the merged calibration of both cameras. Table 3.7 presents the refined intrinsic parameters after the stereo calibration. The main result

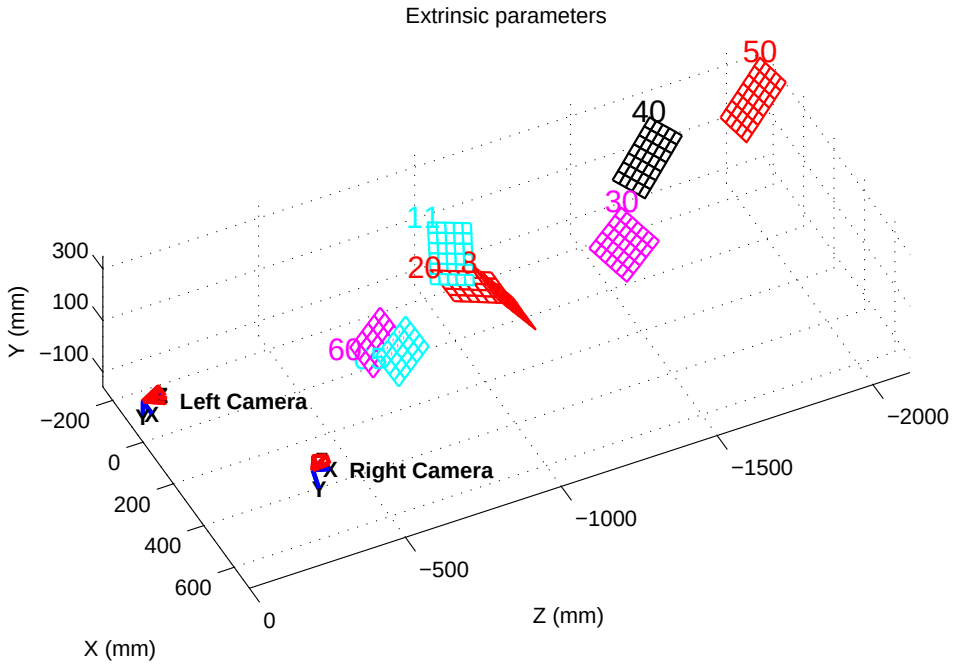

Figure 3.15: Extrinsics of the stereo vision system

of the stereo calibration process are the external parameters of the stereo vision system. These parameters (rotation vector, and translation vector of the right camera with respect to the left camera) are shown in Table 3.8. Calculating the absolute distance between both cameras based on the translation vector presented in Table 3.8 results in 718.1 mm. Since it is not possible to measure the distance between the two focal points of the camera, only an approximate measure-

	Left camera	Right camera
Focal Length (Numerical errors)	[1025.28304 1025.33307] (± [1.07529 1.16229])	[1031.19881 1029.85177] (± [1.08098 1.09165])
Principal point (Numerical errors)	[381.88962 263.95085] (± [4.75023 3.09774])	[391.69785 250.41588] (± [4.60140 2.46275])
Skew (Numerical errors)	[0.00000] (± [0.00000])	[0.00000] (± [0.00000])
Radial distortion (Numerical errors)	[−0.17337 0.02097 0.00000] (± [0.00992 0.06372 0.00000]	[−0.19688 0.10876 0.00000] (± [0.00863 0.05218 0.00000]
Tangential distortion (Numerical errors)	[0.00073 −0.00032] (± [0.00059 0.00080]	[0.00091 0.00059] (± [0.00041 0.00072])

Table 3.7: Refined intrinsic parameters of left and right camera after stereo calibration at full resolution

Rotation vector (Numerical error)	[0.00561 0.68419 0.44573] (± [0.00417 0.00522 0.00116])
Translation vector (Numerical error)	[−658.22332 −132.96479 254.48412] (± [1.07866 0.57657 2.55960]

Table 3.8: Extrinsic parameters (position of right camera wrt left camera) of the stereo vision system at full resolution

ment of the distance between both cameras is possible. This results in a estimated distance of 720 mm which means that the result of 718.1 mm is plausible.

3.2.2 Calibration at Halve Resolution (286 x 240)

The scaling analysis presented in Section 5.2.2 compares prediction results based on the full resolution of 572 x 480 pixel with results based on a reduces resolution of 286 x 240 pixel. This resolution reduction is done on the camera on hardware level and based on binning (compare Section 3.1.2). As lower resolution cameras are available for a lower price their usability shall be examined. Another advantage of the lower resolution is the higher frame rate at which the camera is able to operate.
In order to provide realistic results for the prediction based on the lower resolution also the calibration has to be done at lower resolution. Quantization errors of the calibration are bigger if the camera resolution is lower and this is also valid for a decreased resolution. The calibration is done exactly like the calibration at full resolution. The final calibration results are presented in Table 3.9. and Table 3.10. The positions of both cameras were not changed between the calibration at full and at halve resolution but it was not done consecutively. This also means that the extrinsic parameters of both calibrations should be nearly identical which is also the case (compare Table 3.8 and Table 3.10). Comparing the results of the calibration at full resolution (Table 3.7 and Table 3.8) with the results at halve resolution shows that the decreased resolution yields in small differences. Intrinsic values for focal length and principal point should be exact the halve of the values in case of full resolution which is not the case. Still the discrepancy is minor. The difference in the extrinsic parameters is minor as well. For comparison the distance between the two cameras focal point is determined with 717.9 mm compared to 718.1 mm in case of full resolution.

	Left camera	Right camera
Focal Length (Numerical errors)	[512.45225 512.84476] (± [0.65581 0.70583])	[514.81884 514.88947] (± [0.64409 0.64786])
Principal point (Numerical errors)	[191.95978 137.66740] (± [2.85776 1.89180])	[194.51067 126.75595] (± [2.73538 1.42494])
Skew (Numerical errors)	[0.00000] (± [0.00000])	[0.00000] (± [0.00000])
Radial distortion (Numerical errors)	[−0.17389 0.01075 0.00000] (± [0.01504 0.11237 0.00000]	[−0.19670 0.10606 0.00000] (± [0.00990 0.05739 0.00000]
Tangential distortion (Numerical errors)	[0.00095 0.00069] (± [0.00076 0.00096]	[0.00058 0.00049] (± [0.00051 0.00087])

Table 3.9: Refined intrinsic parameters of left and right camera after stereo calibration at halve resolution

Rotation vector (Numerical error)	[−0.00259 0.68639 0.44953] (± [0.00485 0.00601 0.00149])
Translation vector (Numerical error)	[−658.26997 −133.46864 253.43374] (± [1.33864 0.65798 3.02970]

Table 3.10: Extrinsic parameters of the stereo vision system at halve resolution

3.3 Feature Extraction

In the context of this work, feature extraction deals with determination of the tennis balls center in the image. In order to obtain this task the moving tennis ball has to be detected in the image and the balls center calculated. The first task is topic of the following subsection. The implementation of the ball detection based on a simple background subtraction including a threshold detection of changed image information will be presented as well as the more advanced following edge detection based on the resulting image. Further processing of result of this edge detection, the edge image, which is used to determine the midpoint is topic of Subsection 3.3.2. This center calculation is based on Hough transformation (compare Section 2.1.3) and allows also precise computation of the center in case parts of the ball have not be extracted from the image.

3.3.1 Ball Extraction and Edge Detection

Different techniques of feature extraction of moving objects exist as stated in Section 2.1.1. For the presented task the usage of a background model outclasses alternate frame differencing due to the lapse of ghosting and the relatively constant background and illumination. A single captured image without the ball on the throwing device is used as most simple background model. In order to deal with long term instabilities of the background the background image is captured for each throw directly before the ball is thrown. This is also possible in the target application as a catching station has to be notified about request of an origin station before the actual throw can take place. Acquiring the background image is implemented in the triggering hardware (compare Section 3.1.3). A sample background image of the left camera is shown in Figure 3.16.
Subsequently to the capturing of the background image the images of the actual flight of the ball are acquired. An example series of image is shown in Figure 3.17. Each of the flight-images is processed in the same manner. First the background image is subtracted and the absolute value of the difference is stored. This operation is done on the gray scale image delivered from the camera and results in an image that only contains the differences to the background image. As

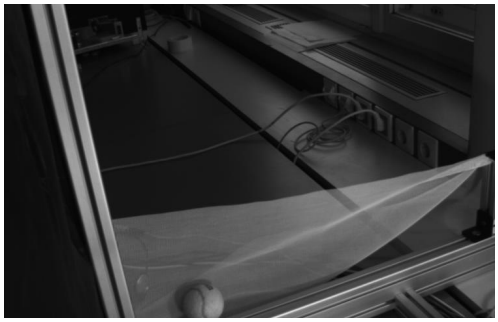

Figure 3.16: Background model: simple image captured prior the throw

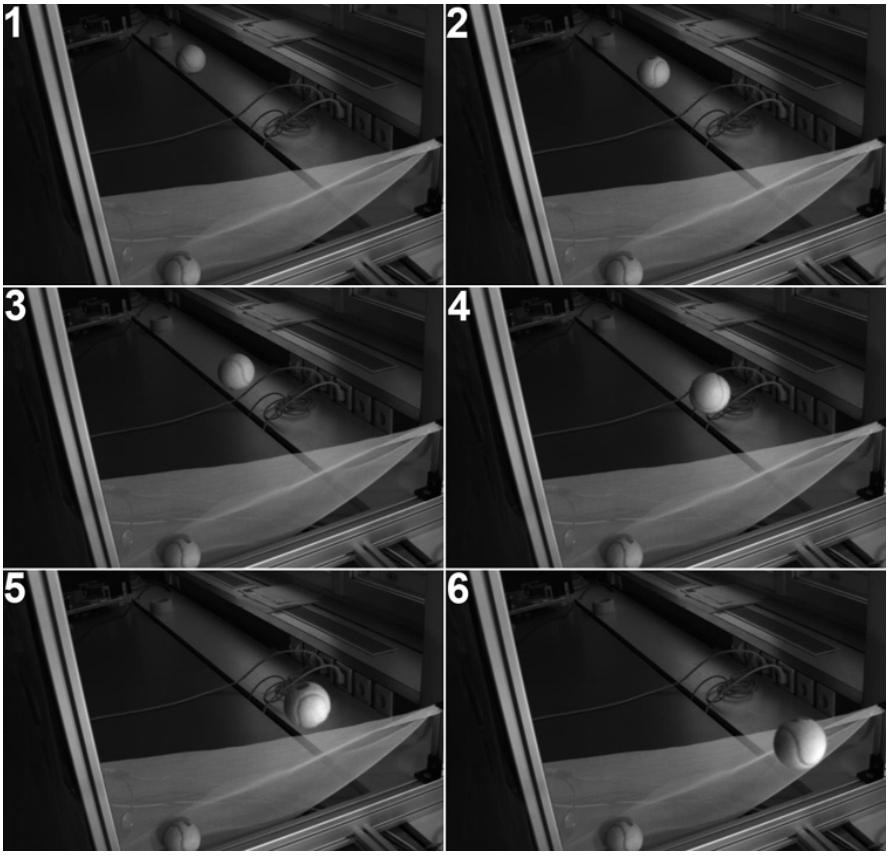

Figure 3.17: Series of images captured during the flight; note ball at the bottom of each image

the tennis ball is moving inside the visual field of the camera the altering of the optical properties result in a detection of the ball. Due to noise in the image acquisition on the image sensor and small illumination changes also the other pixel of the image, which are not related to the tennis ball, contain nonzero data. In order to reject this noise a constant threshold is used to clip each pixel either to completely black (if below the threshold) or white if its value is above the threshold. The information loss of the black/white image in comparison to the gray scale image does not affect the further processing as it only the the center of the tennis ball is of interest and the different brightness of the bails areas does not need to be considered. Sample pictures including the difference image as well as the images past the threshold operation with

different thresholds (ranging from 3 to 70 of 255) are shown in Figure 3.18. In case the difference

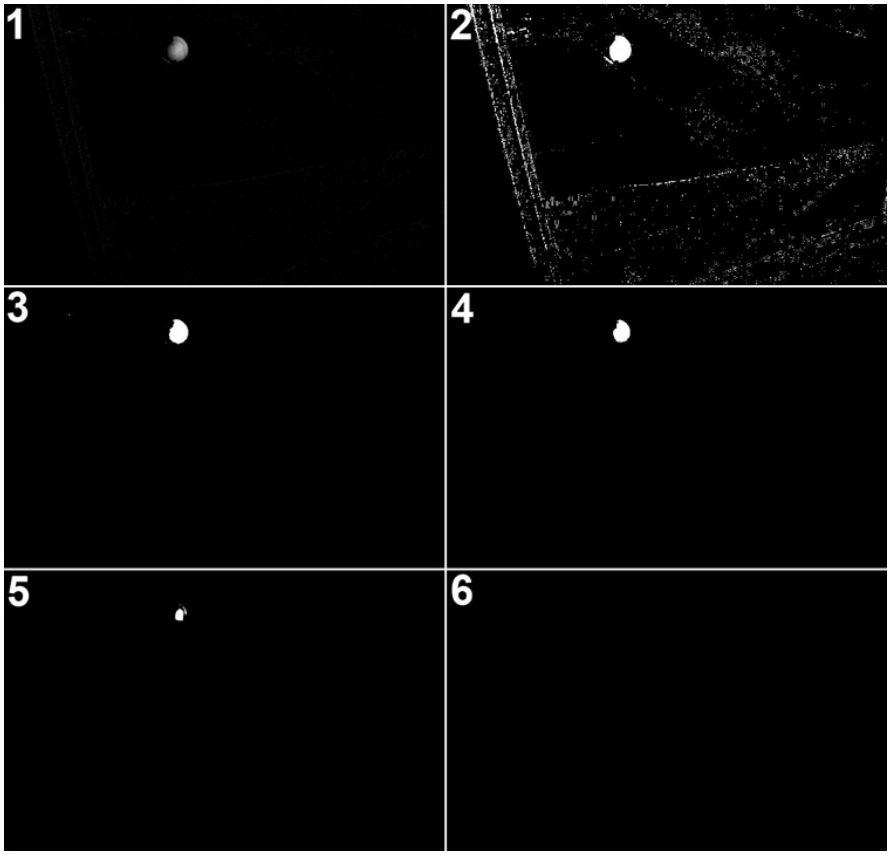

Figure 3.18: 1) 3x Boosted difference image and images after the threshold operation with a threshold of 2)3 3)18 4)30 5)50 6)70

between the background image and the ball in the image is smaller than the threshold, the pixels of the ball which are similar to the background are not detected as part of the ball. This effect mainly occurs on bright background in case the ball is close to the flood lights and the image is locally overexposed. Two types of errors result. The first is a error of the balls contour while the seconds results in holes in the ball. Sample images of both types are shown in Figure 3.19. This

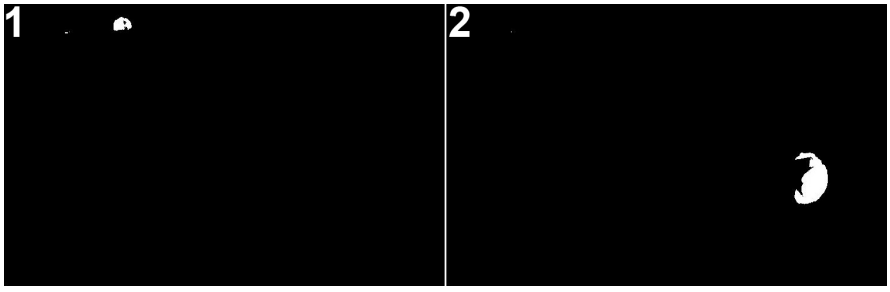

Figure 3.19: 1)Holes in case of similar foreground/background brightness 2)Shape error

error influences the detection of the balls center in case only a simple averaging of all the balls pixel position is used. Also other movements inside the visual field of the camera would result in erroneous determination of the center position. In order to reject this error sources Hough

transformation is used to calculate the center of the ball in the image (Section 3.3.2). As Hough transformation only works on edge images these edge images have to be obtained. Different operators were introduced in Section 2.1.2 and the operation used in this case is the Laplacian of Gaussian (Table 2.3) as it offers a good compromise between calculation time and quality of the resulting edge image. Using the Canny algorithm (Section 2.1.2) results in a longer computation time while the result in this simple environment is of comparable quality to using the LOG algorithm (compare Figure 2.4 in Section 2.1.2). Some sample edge images are shown in Figure 3.20. The edge detection algorithm is working on black/white images which are obtained by using a threshold of 18 (compare Figure 3.18) which is also used through out this work as it offers a good compromise between detection errors and low shape alternation. The first two images also show the edge image based on the images presented in Figure 3.19. These edge images are the

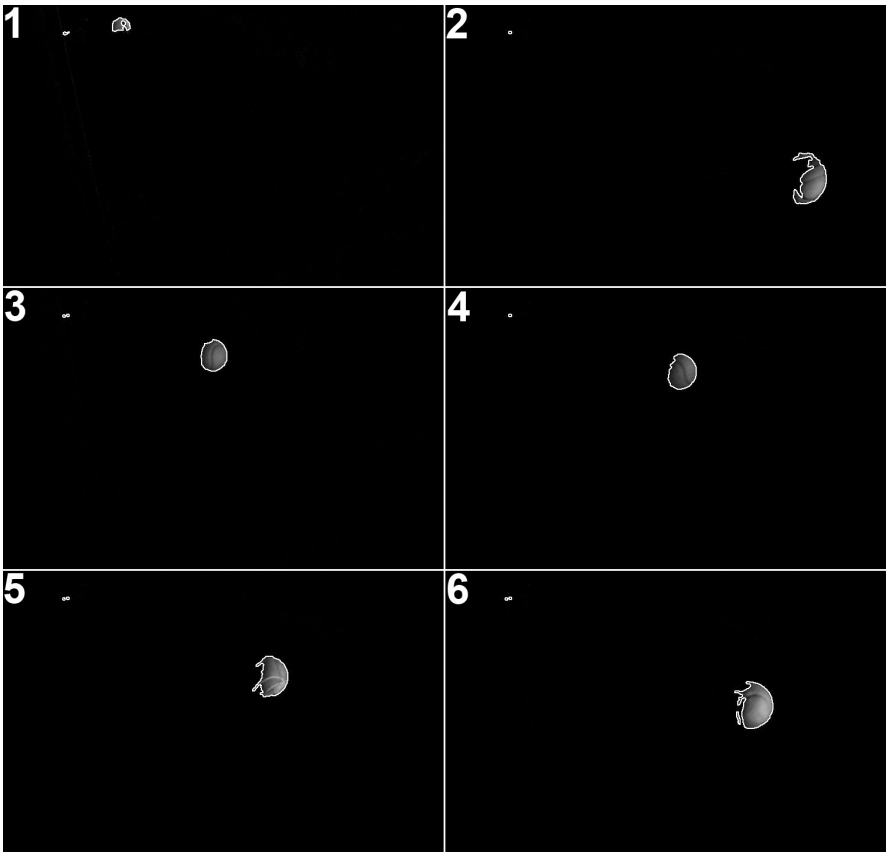

Figure 3.20: Sample edge images based on: 1-2) Figure 3.19 (1-2) 3-6) Figure 3.17 (2-5)

input to the Hough transformation which is topic in the following subsection which concludes the section of feature extraction.

3.3.2 Center Calculation

As already mentioned in the previous section Hough transformation is used to determine the midpoint of the tennis ball in the image as this center position is the input information for the stereo triangulation described in the following Section 3.4. Hough transformation is generally a method that allows information merging. In this case the balls edge information in pixels can be

transformed into information about the midpoint and the radius of the ball. While the radius is not relevant for this application, the center is of great interest, especially an accurate determination of the center even if the edge detection image is erroneous. The advanced version used (modified from [SPV05]) is build specific to detect circles or arcs and rejects straight lines.
While most of the algorithm used is similar to the presented algorithm, differences are found in the refusal of edge parts as well as in the lines of possible centers. While it is suggested to simply count the edge pixels in the window and use the window if the number of pixels equals the size of the window in pixels the implemented algorithm also tests if both extremes of the arc touch the sides of the window which suppresses circles smaller than the window.
The second modification regarding the drawing of the radial line to the accumulator is that the whole line is drawn which is in contrast to drawing the line only towards the direction of concavity as suggested [SPV05]. This is a trade-off between performance of drawing the line and the complexity to determine the direction of concavity.
Figure 3.21 shows the filtered accumulator space of some sample images. A simple $7x7$ filter of equal weights is used as filter operator. The center detection is based on the images shown in Figure 3.20. The black dot in each images is the detected position of the tennis balls center.

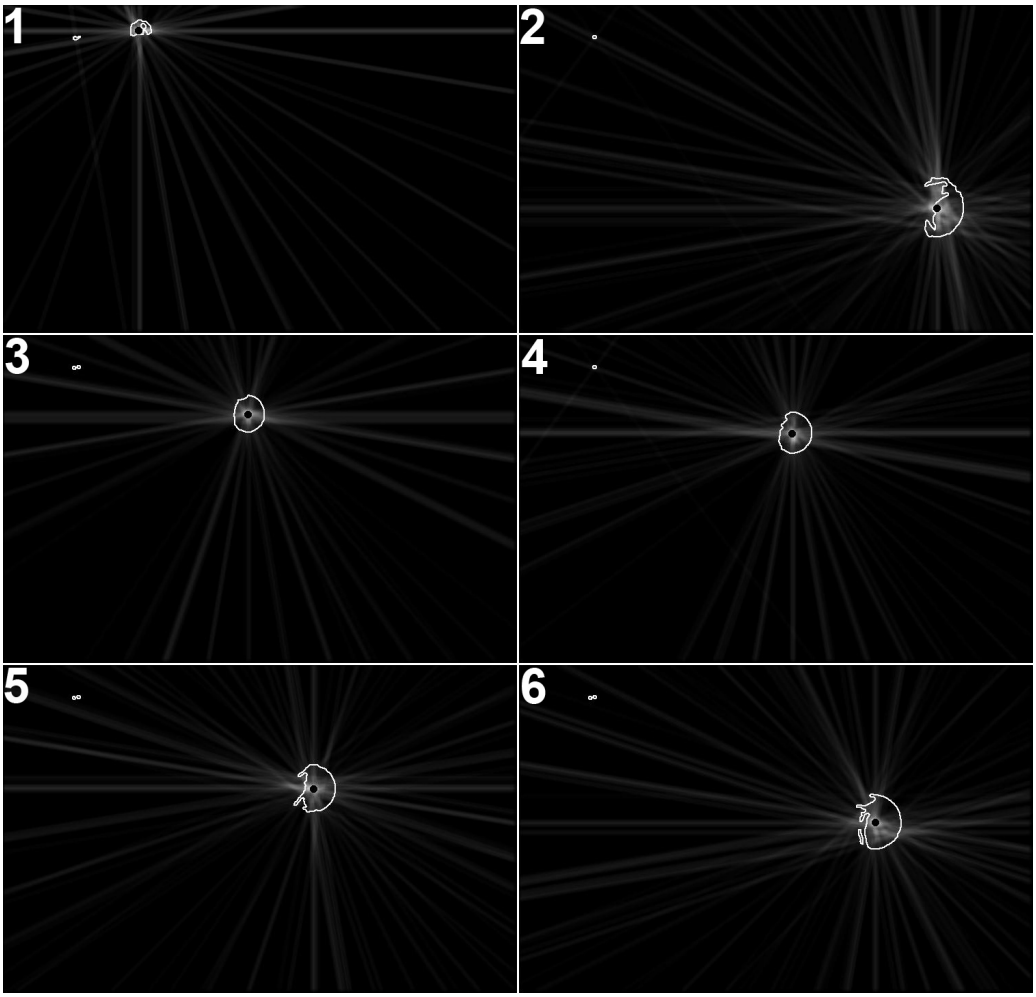

Figure 3.21: Filtered accumulator images with base edge map and calculated center (black dot). Images corresponding to Figure 3.20

For better comparison with the real image, Figure 3.22 shows the detected edge as well as the calculated center position of the tennis ball in the base images. It is easy to see that the auto-

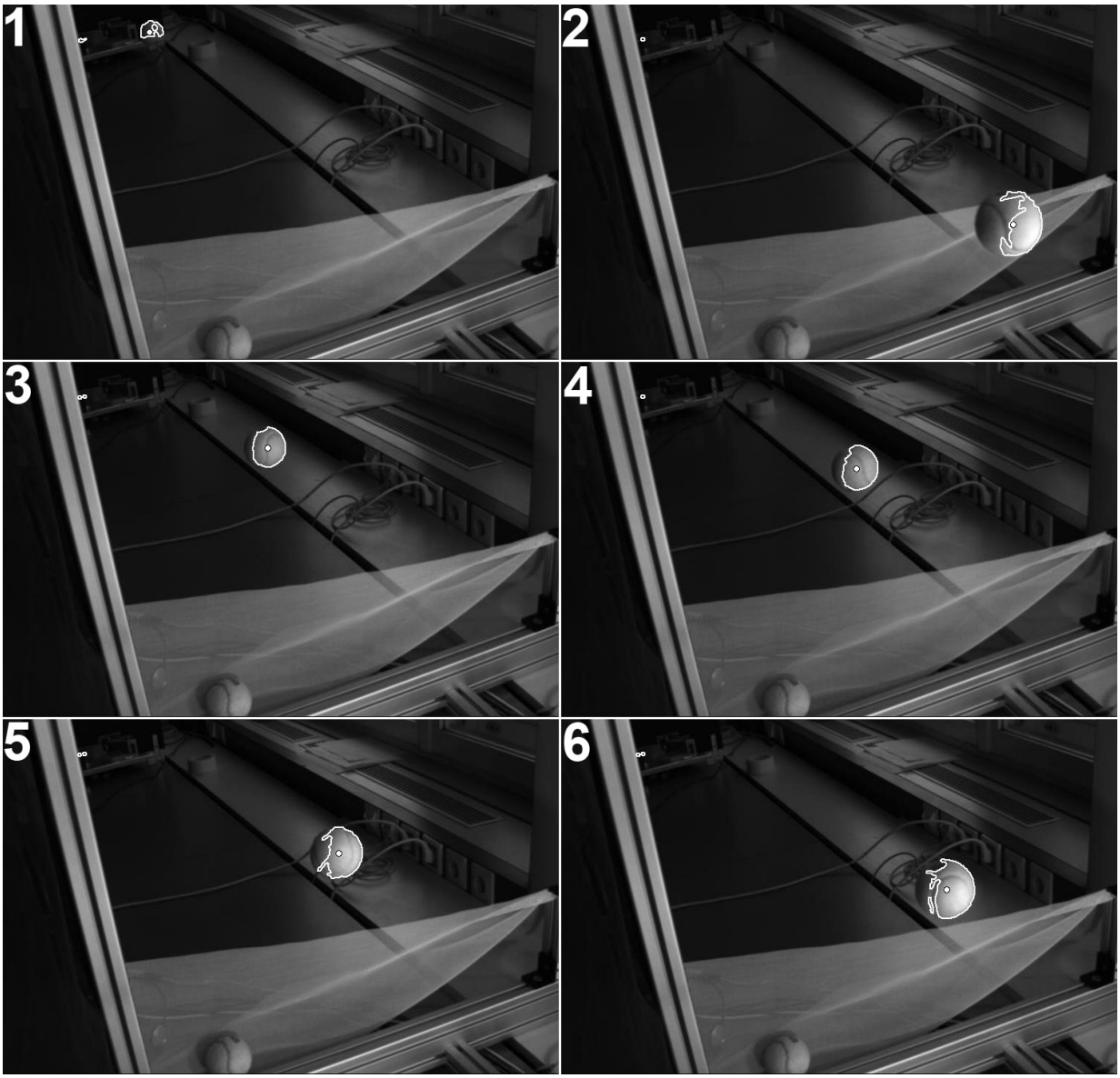

Figure 3.22: Base images with superimposed edge image and detected center (white dot). Images corresponding to Figure 3.20 and Figure 3.21

matic center calculation also works very well for erroneous edge images of the ball. Even if other (moving) objects are in the visual field of the camera the center of the ball is still measured with a high accuracy. The only downside of using Hough transformation is the increased amount of processing required. In this non real-time environment this cost is not important while it might be in the final application. As (parallel) processing power is constantly increasing this limitation is also not crucial for the target application especially as Hough transformation is a task which can easily be parallelized.

3.4 Stereo Triangulation

After acquiring the positions of the tennis ball in both cameras frames the 2D position information of both cameras has to be transformed into a 3D information by merging both 2D informations. This is done by stereo triangulation which is based on the intersection of the rays from both determined midpoints through the focal points of the related camera. As the positions of the cameras is obtained via the previous done calibration (compare Section 3.2) the intersection of the rays delivers the position of the tennis ball (compare Section 3.1.1).
Since the tennis ball is a round object, the usage of the midpoint does not introduce any errors as the projection into both cameras image planes does not distort the shape of the object and the 2D center of the image of the ball is aligned with the 3D center of the ball. Stereo triangulation is available as one feature of the Camera Calibration Toolbox for Matlab [2]. The version implemented also calculates the influence of the lens distortion and eliminates the errors based on the lens parameters obtained in the calibration (compare radial and tangential distortion in Table 3.7 and Table 3.9). Depending on the frame rate the cameras, the number of positions acquired during the flight of the ball varies from 13 to 64. A sample diagram of the positions obtained during a flight captured with 70 fps is shown in Figure 3.23. The coordinate systems used to

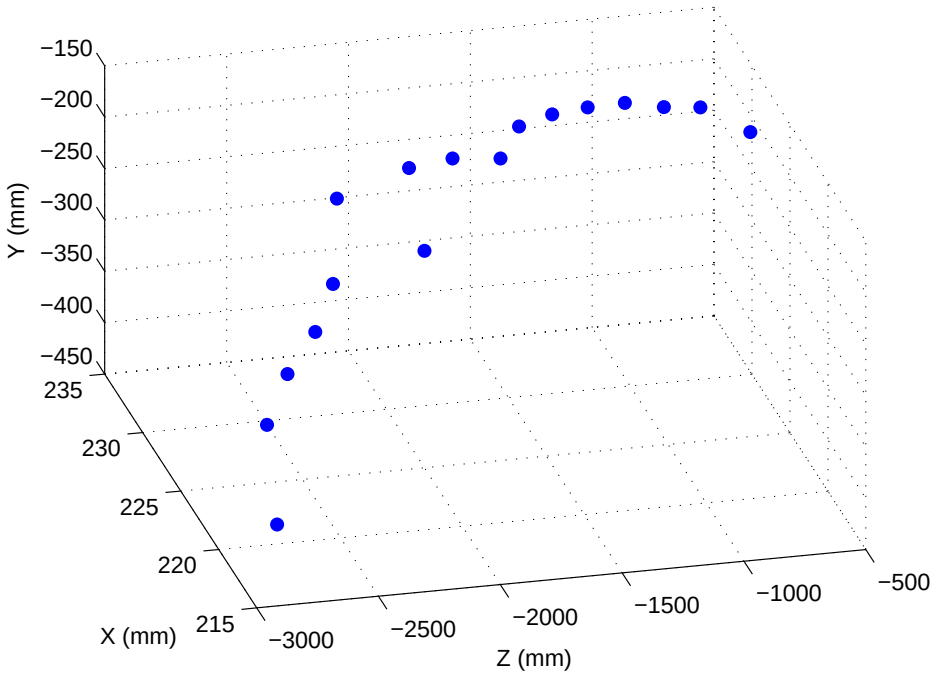

Figure 3.23: Determined positions during a flight captured with 70 fps

display the flight positions in Figure 3.23 is the world coordinate system introduced in Figure 3.2. This means that the coordinates are already transformed in to the world coordinate system by using the translation and rotation information determined in Section 3.2. A set of a complete series of throws is shown in Figure 3.24. All the throws captured at 70 fps and used for the prediction (Chapter 4 and Section 5.2) are presented. Based on this set of positions the further flight trajectory of the tennis ball is predicted. From Figure 3.24 the approximate deviation of the impact position can be estimated at $200x150$ mm which clarifies the requirement for an universal

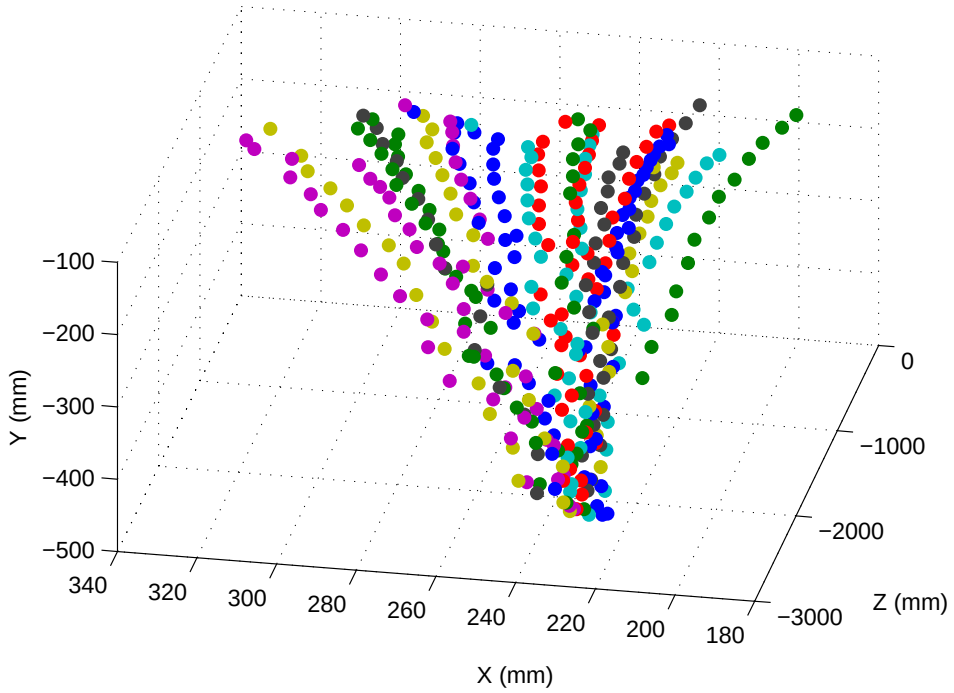

Figure 3.24: Determined positions during multiple flight captured with 70 fps

prediction algorithm. A discussion of prediction models and the resulting prediction algorithms will be topic of the next section.

4 Interception Position Prediction Algorithms

In order to catch an object a catching device (in human context the hand) has to be positioned properly. This positioning has to take place before the object has reached that certain position. Predicting the flight is necessary for this. One main advantage of a good early prediction of the flight is that the catching device can be positioned at the predicted interception position early which limits the amount of energy necessary for this task and mechanical wear as slower movement is possible.
As mentioned in Section 2.2.1, predicting the flight is based on information about the previous positions of the thrown object. The parameters of the model used for the further prediction have to be chosen suitable to the existing positions in order to obtain the prediction based on this model parameters. Since this task can be generally seen as line fitting, the first model introduced in the following Subsection 4.1 is based on polynomial line fitting using a polynomial function of second order for each of the spatial directions. On contrary both other models presented (Section 4.2 and Section 4.3) are based on the physics of the tennis balls flight. Both of them take air drag (quadratic model) and gravity into account. Due tot he nonlinear property of the air drag the flight trajectory has to be calculated iteratively based on the initial velocity $\mathbf{v}$ and the drag coefficient c_w. The fitting to the measured data is still similar to fitting functions into the measured positions as it is based on the euclidean distance between measured positions and the calculated positions of the model. Monte Carlo simulation is used in order to choose from multiple launch parameters. Initial parameters are derived from the measured positions. As the computation complexity of this model is suspected to be very high a third model is introduced. Considering the movements in each spatial direction separately enables to calculate the functions which describe the flight for each dimension. This again enables line fitting but still considers the physics of the flight. The error introduced by independent axis consideration will be discussed as well.
Section 4.4 compares the three models presented and also examines the time needed to calculate the prediction of the impact position. This time in relation to the prediction accuracy is a good measurement for a models efficiency. Prediction results obtained by these three models as well as a detailed discussion of the efficiency will be topic of the following Chapter 5.

4.1 Polynomial Line Fitting

The polynomial model is not based on on any physical background as mentioned in the introduction of this chapter. A second order model is used which is based on the Equations 4.1 for each spatial direction.

$$\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{v} * t + \mathbf{a} * t^2 \tag{4.1}$$

Fitting the model for the known positions of the ball is done by determining the best suiting constants position **p**, velocity **v** and acceleration **a**. Optimal parameter determination is done based on the linear least square approach (compare Section 2.2.2). Outliners are suppressed by Bisquare weights [8].
The reason for choosing a second order model over a model of higher order (3^{rd}, 4^{th}, ...) is that the second order model is less sensitive to measurement errors while models of higher orders are. This behavior is found based on a dataset of 20 predictions at a frame rate of 70 fps. Table 4.1 compares the prediction results based on the second order and the third order model. The deviation in Table

order	deviation	
	upper 95 % bound	max
2^{nd}	9.96 mm	15.66 mm
3^{rd}	48.39 mm	32.16 mm

Table 4.1: Comparison of prediction results based on polynomial model of second and third order

4.1 is calculated as euclidean distance between the predicted and the measured impact position of the tennis ball on the plane and the distribution used is the Rayleigh distribution. More details on the evaluation and characterization of the prediction data is presented in Section 5. This short insertion is only made to ground the choice of the second order model. Another small advantage of the second order model over the third order model is lying in the faster computation time which might be important in a real time environment.
As the general precision of the position acquisition via the stereo vision system is considered to improve as the ball moves closer to the camera the positions in the final flight phase a weighted higher in order to influence the trajectory fitting more. The weights used are shown in Table 4.2. Since accuracy of the position calculation increases over flight time also the importance (and thus

Indexes (rounded up)	Weights
$1 \cdots \frac{n}{6}$	1
$\frac{n}{6} \cdots \frac{n}{3}$	2
$\frac{n}{3} \cdots \frac{n}{2}$	4
$\frac{n}{2} \cdots \frac{2*n}{3} - 1$	5
$\frac{2*n}{3} - 1 \cdots n - 1$	8
n	1

Table 4.2: Weights used to account increased position acquisition accuracy as ball moves towards the vision system, number of positions acquired: n

the weight) of the position in increasing. The reduced weight of the last position is done because the final position is frequently based on images where parts of the ball are outside the visual field of at least one camera and thus the position calculation accuracy is reduced. The set of weights presented in Table 4.2 obtains this and was found as best suiting weights out of an small number of sets.

A graphical illustration of the polynomial model applied to a sample set of measured positions of the ball is shown in Figure 4.1, Figure 4.2 and Figure 4.3 for each of the three spatial directions. This model fits the data well and looking at the y-direction in Figure 4.2 also shows the typical

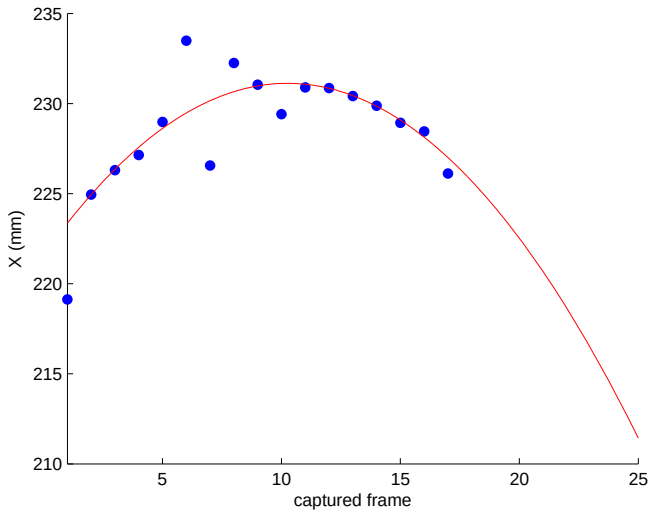

Figure 4.1: Polynomial model applied to a sample set of ball positions in x-direction (ots measured positions, line modeled flight trajectory)

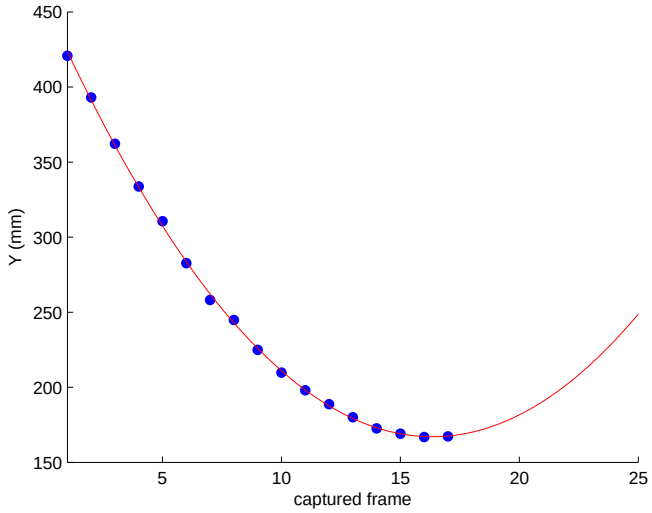

Figure 4.2: Polynomial model applied to a sample set of ball positions in y-direction (dots measured positions, line modeled flight trajectory)

parabolic shape known from a trajectory without taking the air drag into account. The parabolic shape is upside down due to the coordinate system introduced in Figure 3.2 in Section 3.1. What has to be noted is the change of the flight direction displayed in Figure 4.1. Using the polynomial model allows the ball to move with a changing sign of the movement in x-direction. This is in contrast to both physical based models as both do not consider the Magnus force (compare Section 2.2.1) which can have this effect on the movement. The change of movement direction

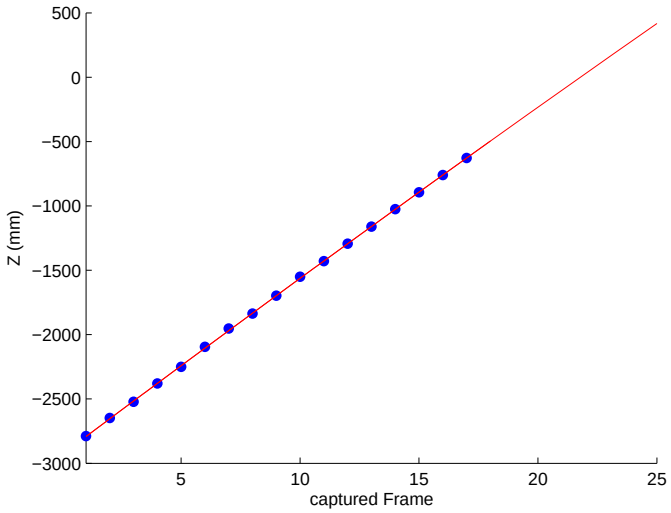

Figure 4.3: Polynomial model applied to a sample set of ball positions in z-direction(dots measured positions, line modeled flight trajectory)

is only modeled in y-direction due to gravity. Air drag can only slow movements down but not influence the trajectory in the way which is shown in Figure 4.1.

4.2 Physical Model

Contrary to the model presented in the previous chapter, this model is based on flight physics. The forces which are taken into account are gravity and air drag (compare Table 2.4 of Section 2.2.1). While gravity is a constant force on the tennis ball, air drag depends on the movement of the ball. The direction air drag is directed exactly opposed the direction of the movement and the magnitude of the force depends on the velocity of the ball quadratically. This relation is shown in Equations 4.2 to 4.4 [5]. The used symbols are: force due to air drag $\mathbf{F}_{drag}$, air drag coefficient c_W, cross section surface of the tennis ball A, air density ρ, velocity $\mathbf{v}$, acceleration due to air drag $\mathbf{a}_{drag}$, mass of tennis ball m and for simplification the air drag factor k.

$$\mathbf{F}_{drag} = -c_W * A * \frac{\rho}{2} * \mathbf{v}^2 \tag{4.2}$$

$$\mathbf{a}_{drag} = \frac{\mathbf{F}_{drag}}{m} = -c_W * A * \frac{\rho}{2 * m} * \mathbf{v}^2 = -k * \mathbf{v}^2 \tag{4.3}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.4}$$

Due to the nonlinearity of the air drag the force cannot be considered for each spatial direction independently. Furthermore, no function solving this equation can be found which yields in the fact that the flight trajectory for this model can only be calculated iterative. The calculation of the flight is based on the Equations 4.5 to 4.9 (compare Section 2.2.1). Equation 4.5 is the base for Equations 4.8 and 4.9 which are necessary as no closed solution for the Equation 4.5 exists. This means that the future velocity $\mathbf{v}(t + \Delta t)$ can only be calculated based on the actual velocity

$\mathbf{v}(t)$.

$$\mathbf{v}(t) = \mathbf{v}(0) + \mathbf{g} * t - \int_{\tau=0}^{t} k * \mathbf{v}^2 d\tau \tag{4.5}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.6}$$

$$\mathbf{g} = [0, g, 0]' \tag{4.7}$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + (\mathbf{a}_{drag}(t) + g) * \Delta t \tag{4.8}$$

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \mathbf{v}(t) * \Delta t \tag{4.9}$$

The main parameters of this calculation are similar to the ones presented in Section 4.1, namely positions $\mathbf{p}$, velocity $\mathbf{v}$ and acceleration $\mathbf{a}$. In addition this model uses the gravity vector $\mathbf{g}$ which consists of two zero elements in x- and z-direction and g in the y-direction as the world coordinate system (Figure 3.2) has the y-axis aligned wit the direction of the gravity. The time between two calculation steps is called Δt. Iterative calculation is based on a step by step approach by calculating the future parameters out of the actual ones. The calculation time step Δt has major impact on the calculation precision. The smaller the calculation time step, the better the accuracy. In order to optimize the accuracy further without decreasing Δt furthermore Equation 4.9 is replaced by

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \frac{\mathbf{v}(t) + \mathbf{v}(t + \Delta t)}{2} * \Delta t \tag{4.10}$$

which equals a parallelogram approximation compared to the lower sum used in Equation 4.9.
In order to fit this iterative calculated model into the measured data, Monte Carlo algorithm is used. This is necessary as no function describes the trajectory derived from the Equations 4.5 to 4.10. Normal fitting can not be used. In order to find a well suiting trajectory a set of initial parameters (which are the initial velocity $\mathbf{v}(0)$ and k which includes the air drag coefficient c_w, the cross section A and the atmosphere density ρ) has to be tested against each other and the best one has to be picked. Evaluation and comparison of different initial parameter sets is also based on the least square algorithm. The sum of the Euclidean distances between the measured ball positions and the predicted ones has to be minimized in order to find the optimal parameter set which includes the initial position of the tennis ball p_0, the initial velocity of the tennis ball v_0, the air drag factor k and the magnitude and direction of gravity $\mathbf{g}$. Overall this results in 10 degrees of freedom for this model. As for each set of parameters the whole flight has to be calculated iteratively this algorithm is computation intense and time consuming. Application of this model in a real-time environment is heavily dependent on computation power but this model can deal as a reference for testing derived models like the one presented in the following Subsection 4.3. In order to find a really well suiting set of initial parameters the first round of parameter selection can be followed by further round where the best parameter set of the previous round is supplemented by small bound random numbers multiple times and each of the resulting set is again tested against each other in order to find a more precise fitting set of initial parameters. In order to avoid finding a local minimum the magnitude of the random variation in combination with the number of parameter sets per calculation round is accordant. In addition the weighted number of positions considered for the calculation is also taken into account. Individual acquired positions far off the predicted trajectory are suppressed if their distance to the predicted position is more than 130 mm.
A graphical illustration of the physical model applied to the same sample set of measured positions of the ball as the ones used in Figure 4.1, Figure 4.2 and Figure 4.3 is shown in Figure 4.4, Figure

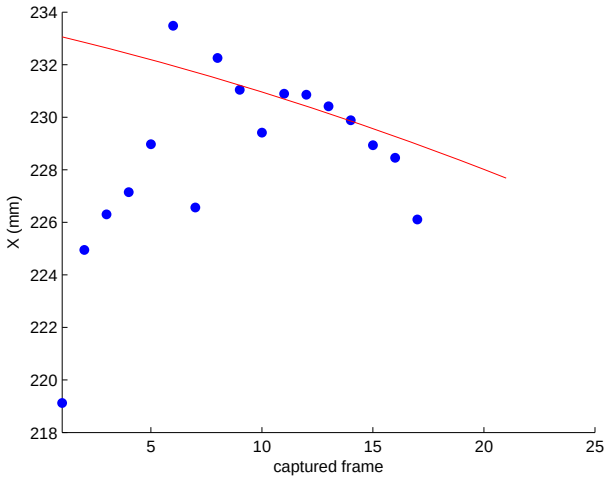

Figure 4.4: Iterative model applied to a sample set of ball positions in x-direction (red measured positions, line modeled flight trajectory)

4.5 and Figure 4.6 for each of the three spatial directions. In contrast to the polynomial model this approach does not allow the velocity in x-direction to change its sign which is shown in Figure 4.1 of the previous Subsection 4.1 while in Figure 4.4 the flight is predicted to move towards the negative x-direction through out the flight as the positions with a higher frame number are considered as more accurate as the ball is closer to the cameras. Another difference to the previous model can be seen in Figure 4.5 where the model does not fit the first measured positions exactly. Similar to the movement in x-direction this behavior occurs out of the fact that no set of parameters for the predefined trajectory suits all the positions well and suiting the positions in the later phase of the throw which are considered to be more precise and thus relevant. This prioritization is done by using unequal weights (compare Table 4.2) for the positions obtained which allow to specify the relevance of each individual position for the finally found trajectory.

The movement in y-direction of the model equates the measured ball positions very well in the

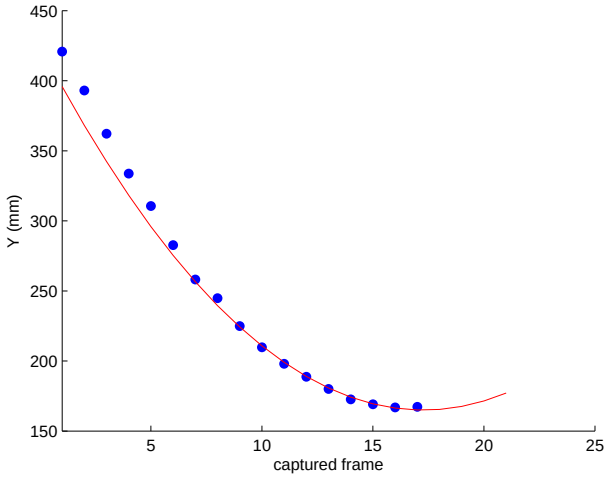

Figure 4.5: Iterative model applied to a sample set of ball positions in y-direction (dots measured positions, line modeled flight trajectory)

second halve of the flight. The discrepancy in the first halve of the flight results out of the higher weighting of the later positions as well as the flight trajectory which is predefined due to the physical characteristics considered.

Regarding the computation time of this model the selection of the final set of initial parameters

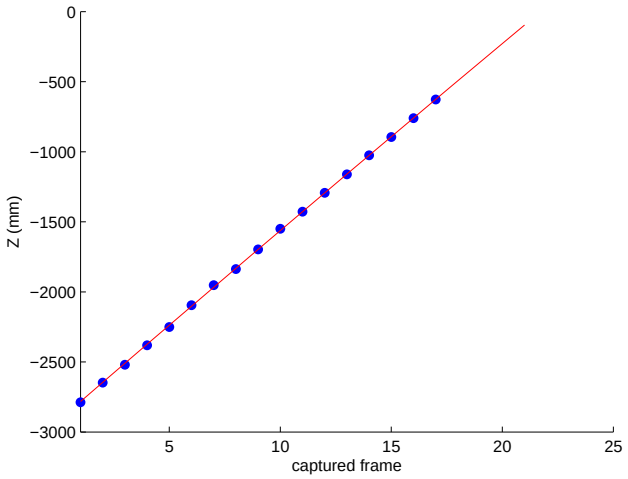

Figure 4.6: Iterative model applied to a sample set of ball positions in z-direction(dots measured positions, line modeled flight trajectory)

(20 round with 100 parameter sets each) takes 3 minutes and 24 seconds time on the work station use (compare 3.4). Due to this fact this model will only be used as a reference in the final Chapter 5. Using this model in the target application is also viable as the calculation of the 100 trajectories per simulation round can easily be done in parallel on supporting hardware. This might also be possible on general purpose graphic processing units which feature a huge number of simple execution cores.

4.3 Spatial Separated Physical Model

While the model described in the previous subsection is very accurate the big disadvantages are lying in the computation complexity in order to select a suitable set of initial parameters and also the complexity to calculate the actual flight trajectory. This drawbacks are tried to be eliminated by separating the spatial directions from each other. Due to the nonlinearity of the air drag an error is introduced. Figure 4.7 illustrates the introduced error transparently. While both velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ have the same components ($\mathbf{v}_{21}$ and $\mathbf{v}_1$) the resulting forces due to air drag $\mathbf{F}_1$ and $\mathbf{F}_2$ differ in their spatial components ($\mathbf{F}_{21}$ and $\mathbf{F}_1$) by $\mathbf{F}_3$. As illustrated the error depends on the proportion between the velocity in different directions. As the main movement of the ball occurs along the z-direction the influence of this systematic error might be a good trade-off versus the expected smaller computation demands this model offers.

Each of the directions has to be considered on its own and the equations appropriate to the model (considering air drag and gravity similar to the model presented in Subsection 4.2) have to be derived and a function solving the equation has to be fit into the measured positions. This again can be done by using optimization but in contrast to the polynomial model (Section 4.1) this the time nonlinear least square algorithm has to be used. Similar to the other two models, weights

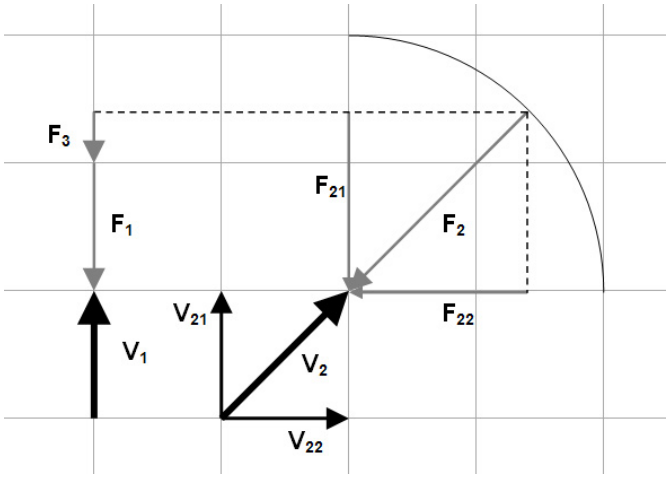

Figure 4.7: Error introduced due to consideration of each spatial direction separately

are also used here in order to give priority to the final flight phase. The weights used are the same for all models and are presented in Table 4.2.
The equations valid for the movement along the x-axis are Equation 4.11 and 4.12 as the only influence on this movement in this model is the air drag.

$$\dot{v}_x = -k * v_x^2 \tag{4.11}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.12}$$

Deriving the solution of this equation is done in Equations 4.13 to 4.19 where the differential equation is solved in general.

$$\dot{v}_x = -k * v_x^2 \tag{4.13}$$

$$\frac{\dot{v}_x}{v_x^2} = -k \tag{4.14}$$

$$\int_0^\tau \frac{\dot{v}_x}{v_x^2} = -k * \tau \tag{4.15}$$

$$\left(-\frac{1}{\dot{v}_x}\right)_0^\tau = -k * \tau \tag{4.16}$$

$$\frac{1}{v_x(0)} - \frac{1}{v_x(\tau)} = -k * \tau \tag{4.17}$$

$$\frac{1}{v_x(\tau)} = \frac{1}{v_x(0)} + k * \tau \tag{4.18}$$

$$v_x(\tau) = \frac{1}{\frac{1}{v_x(0)} + k * \tau} \tag{4.19}$$

In order to minimize the computation amount further the resulting velocity v_x is used to calculate the position of the ball according to this model in general. This is shown in Equations 4.20 to

4.22.

$$x(t) = x_0 + \int_0^t v_x(\tau)d\tau = x_0 + \int_0^t \frac{1}{\frac{1}{v_x(0)} + k * \tau} d\tau \tag{4.20}$$

$$x(t) = x_0 + \frac{1}{k} * ln\left(1 + k * t * v_{x_0}\right) \tag{4.21}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.22}$$

This is finally the function used to model the movement in x-direction. The parameters of this solution are initial position x_0, the initial velocity v_{x_0} and k (including air density, cross section area and air drag coefficient of the ball). Applying this result to the same dataset as for both previous models yields in Figure 4.8. Comparing Figure 4.8 to Figure 4.1 and Figure 4.4 shows

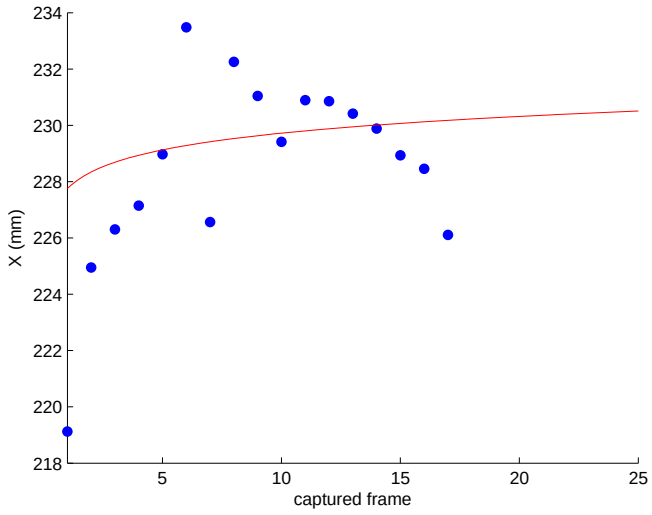

Figure 4.8: Separated model applied to a sample set of ball positions in x-direction (dots measured positions, line modeled flight trajectory)

that the separated model also does not allow the velocity in x-direction to change it's sign (similar to the iterative model in Figure 4.4). The fitted function clearly differs from the polynomial model as well as the physical model in the area where the future flight is predicted (right from the last dot in Figure 4.1, Figure 4.4 and Figure 4.8). While the fact that the direction of the velocity is similar with the iterative as well as the separated model the predicted velocities of both models in the final flight phase have opposite signs. This has a major impact on the prediction of the impact position on the interception plane.
Also for the movement in z-direction, the air drag has to be considered only. The base for deriving the function to fit into the measured positions are Equations 4.23 and 4.24 which resemble Equations 4.11 and 4.12.

$$\dot{v}_z = -k * v_z^2 \tag{4.23}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.24}$$

Following the same calculation steps as presented in Equations 4.13 to Equations 4.22 the corresponding solution for the z-position is shown in Equation 4.25.

$$z(t) = z_0 + \frac{1}{k} * ln\left(1 + k * t * v_z(0)\right) \tag{4.25}$$

Using the model on the sample position acquired results in Figure 4.9. In this direction this

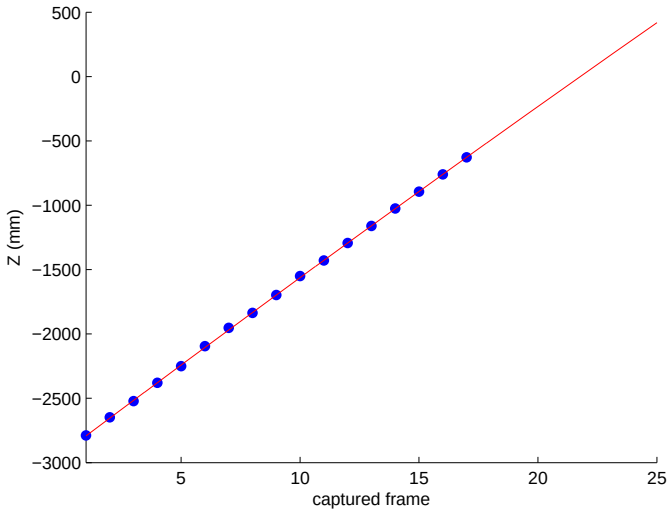

Figure 4.9: Separated model applied to a sample set of ball positions in z-direction (dots measured positions, line modeled flight trajectory)

model does also fit the measured data very well.
Considering the movement along the y-axis results in a more complex equation as air drag and gravity have to be taken into account which is shown in Equations 4.26 and 4.27 for the ascending phase of the flight.

$$\dot{v}_y = k * v_y^2 + g \tag{4.26}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.27}$$

The air drag force changes it's direction as the ball reaches the dead and as as this happens also the air drag changes it's sign. The valid equation for the descending flight phase is given in Equation 4.28 and Equation 4.29.

$$\dot{v}_y = -k * v_y^2 + g \tag{4.28}$$

$$k = c_W * A * \frac{\rho}{2 * m} \tag{4.29}$$

Obtaining the solutions for this equations is more complex as both are inhomogeneous differential equations. First focus is set on solving Equation 4.28. Equations 4.30 to 4.38 present the main

steps to obtain a solution for Equation 4.28.

$$\dot{v}_y = -k * v_y^2 + g \tag{4.30}$$

$$\frac{\dot{v}_y}{1 - \frac{k}{g} * v_y^2} = g \tag{4.31}$$

$$\int_0^{\tau} \frac{dv_y}{1 - \frac{k}{g} * v_y^2} = g * \tau \tag{4.32}$$

$$\int_0^{\sqrt{\frac{k}{g}}\tau} \frac{du}{1 - u^2} = -\sqrt{g * k} * \tau \text{ with } u^2 = \frac{k}{g} * v_y^2 \text{ and } du = \sqrt{\frac{k}{g}} * dv_y \tag{4.33}$$

$$\left(\frac{1}{2} * ln(\frac{u-1}{u+1})\right)_0^{\sqrt{\frac{k}{g}}\tau} = -\sqrt{g * k} * \tau \tag{4.34}$$

$$\frac{1}{2} * ln\left(\frac{\sqrt{\frac{k}{g}} * v_y(\tau) + 1}{\sqrt{\frac{k}{g}} * v_y(\tau) - 1}\right) - \frac{1}{2} * ln\left(\frac{\sqrt{\frac{k}{g}} * v_y(0) - 1}{\sqrt{\frac{k}{g}} * v_y(0) + 1}\right) = -\sqrt{g * k} * \tau \tag{4.35}$$

$$\frac{1}{2} * ln\left(\frac{\sqrt{\frac{k}{g}} * v_y(\tau) - 1}{\sqrt{\frac{k}{g}} * v_y(\tau) + 1}\right) = -\sqrt{g * k} * (\tau + \tau_0) \tag{4.36}$$

$$v_y(\tau) = -\sqrt{\frac{g}{k}} * \frac{1 - e^{2*\sqrt{g*k}*(\tau+\tau_0)}}{1 + e^{2*\sqrt{g*k}*(\tau+\tau_0)}} \tag{4.37}$$

$$v_y(\tau) = \sqrt{\frac{g}{k}} * tanh\left(2 * \sqrt{g * k} * (\tau + \tau_0)\right) \tag{4.38}$$

The corresponding steps for the ascending flight phase (based on Equation 4.26) are shown in Equation 4.39 to 4.46.

$$\dot{v}_y = k * v_y^2 + g \tag{4.39}$$

$$\frac{\dot{v}_y}{\frac{k}{g} * v_y^2 + 1} = g \tag{4.40}$$

$$\int_0^{\tau} \frac{dv_y}{\frac{k}{g} * v_y^2 + 1} = g * \tau \tag{4.41}$$

$$\int_0^{\sqrt{\frac{k}{g}}\tau} \frac{du}{u^2 + 1} = \sqrt{g * k} * \tau \text{ with } u^2 = \frac{k}{g} * v_y^2 \text{ and } du = \sqrt{\frac{k}{g}} * \tau \tag{4.42}$$

$$(arctan(x))_0^{\sqrt{\frac{k}{g}}\tau} = \sqrt{g * k} * \tau \tag{4.43}$$

$$arctan\left(\sqrt{\frac{k}{g}} * v_y(\tau\right) - arctan\left(\sqrt{\frac{k}{g}} * v_y(0\right) = \sqrt{g * k} * \tau \tag{4.44}$$

$$v_y(\tau) = \sqrt{\frac{g}{k}} * tan\left(\sqrt{g * k} * \tau + arctan(\sqrt{\frac{k}{g}} * v_y(0)\right) \tag{4.45}$$

$$v_y(\tau) = \sqrt{\frac{g}{k}} * tan\left(\sqrt{g * k} * (\tau + \tau_0)\right) \tag{4.46}$$

Both basic equations Equation 4.26 as well as Equation 4.28 describe the same physical relations. This is also the case with Equation 4.38 and Equation 4.46. One solution can be converted into

the other by using the complex *tan* and *tanh* function, exchanging g with $-g$ based on Equation 4.47

$$tanh(i * x) = i * tan(x) \tag{4.47}$$

Due to the real flight physics the final result of both equations will be real (not complex) again. This also enables to consider only one of both equations and use it for the whole flight phase. Also in this case the solution for the velocity v_y is used to determine the function which describes the position $y(t)$ over time. The necessary steps are shown in Equation 4.52 based on Equation 4.38.

$$v_y(\tau) = \sqrt{\frac{g}{k}} * tanh\left(2 * \sqrt{g * k} * (\tau + \tau_0)\right) \tag{4.48}$$

$$y(t) = y_0 + \int_0^t v_y(\tau)d\tau = y_0 + \int_0^t \sqrt{\frac{g}{k}} * tanh\left(2 * \sqrt{g * k} * (\tau + \tau_0)\right) d\tau \tag{4.49}$$

$$\text{using } \int tanh(u)du = \int \frac{sinh(u)}{cosh(u)}du = ln\,|cosh(u)| + C \tag{4.50}$$

$$\text{and the property that } cosh(u) > 0 \tag{4.51}$$

$$y(t) = y_0 + \frac{1}{k} * ln\left(\frac{cosh(\sqrt{g * k} * (t - t_0)}{cosh(\sqrt{g * k} * t_0)}\right) \tag{4.52}$$

This is the final solution of Equation 4.26 and Equation 4.28 which is fit into the acquired positions of the tennis ball. The complex calculation of this model has the advantage that functions for each of the three spatial directions are found. The fitting of the found functions to the measured data is fast and simple. Also the property that Equation 4.26 and Equation 4.28 result in the same solution (Equation 4.52) was expected.
Figure 4.10 shows the application of the model on the sample data set. The measured positions

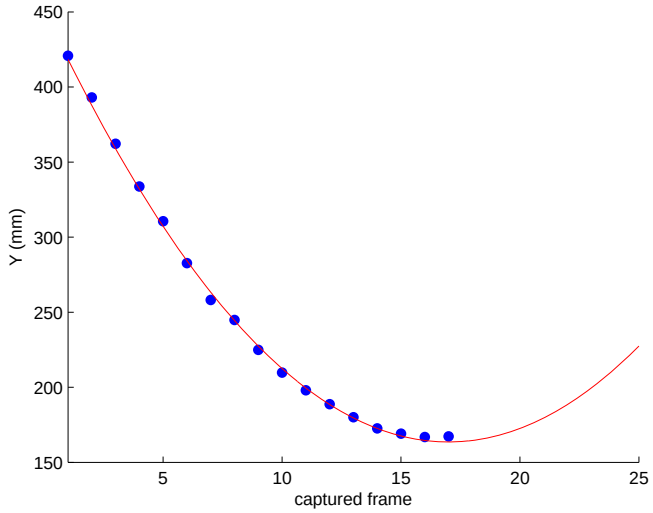

Figure 4.10: Separated model applied to a sample set of ball positions in y-direction (dots measured positions, line modeled flight trajectory)

are very well modeled with the function derived.

4.4 Discussion

The main difference of the three prediction models presented lies in matching the movement in x-direction. While there is no visible difference in both other directions (compare Figure 4.2, Figure 4.5 and Figure 4.10 or Figure 4.3, Figure 4.6 and Figure 4.9 respective) Figure 4.11 compares all three models applicability to describe the movement in x-direction. In all cases where the

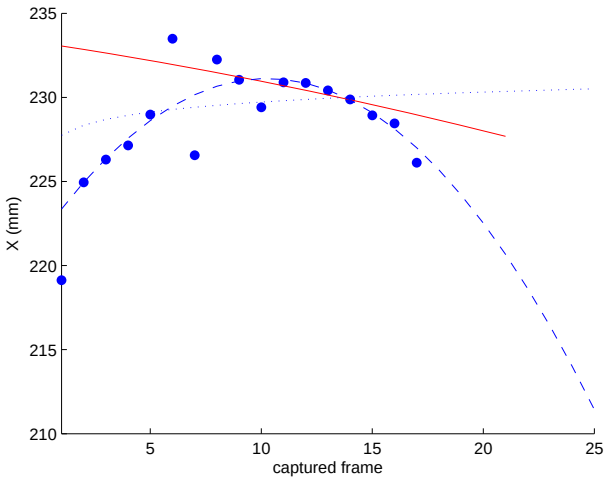

Figure 4.11: Comparison for prediction in x-direction; polynomial model: dashed line, physical model: sold line, separated model dotted line

x-component of the balls velocity changes direction only the polynomial model is able to describe the position during the whole flight accurately. Both physical based models show drawbacks here. The reason for this behavior is that no force resulting in such a movement was considered when developing those models. Only the Magnus force (compare Table 2.4) results in such a curved trajectory. This force requires spin of the flying object which was not considered to be significant. Review of the video data has confirmed this assumption. The flying ball is rotating around its vertical axis with a rotation speed of up to 1500 rotations per minute. The design of the throwing device (Figure 3.1), based on a rotational spring, is the reason for this behavior.
Another attribute differentiating the models is the time it takes to calculate the prediction based on the measured positions. For the polynomial model as well as the separated model this is the optimization time in particular while the time for calculating the prediction based on the physical model consists of the actual calculation of the trajectories as well as the time to select the optimal initial parameter set. Table 4.3 compares this attribute of all models. In order to show the calculation times sensitivity to the number of positions Table 4.3 is presenting the average calculation time for 20 predictions at 70 and 160 fps. Also the average number of position measured at each

Model	time in milliseconds	
Frame rate (fps)	70	160
Average number of positions	14.05	36.25
Polynomial model	1012	1607
Physical model	4241	4508
Separated model	4779	4635

Table 4.3: Average computation time

frame rate are presented. The polynomial model has the lowest computation time at both frame rates which is approximately only the quarter of the calculation time of both other models. It is interesting to note that the calculation time increases by 60 % for the polynomial model when going from 70 to 160 fps while the physical models computation time is only increasing by less than 7 % and the computation time for the separated model does decreases by approximately 3 %.

5 Interception Position Validation and Prediction Results

Based on the setup of the stereo vision system (Chapter 3) and the prediction algorithms presented in Chapter 4 this section presents the results of the prediction. However, before the focus lies on the validation system used to determine the real impact position. Different possible realizations with their advantages and downsides are discussed and the selection of one system established. Using a catching robot for verification would demand a real-time prediction system. In this case the predicted position has to be available at the time the tennis ball intersects with the plane of the catching robot. If the calculated interception position is only available later this information is useless. Here, a verification system which does not raise real-time requirements is used. Using a sensor which detects the impact position of the tennis ball on a plane allows to process the acquired data (tennis ball positions and the tennis balls intersection position with the plane) after the flight. This systems calibration, which expands the camera calibration, topic of Section 3.2, is in focus in Subsection 5.1.2.
Finally the prediction results of the three models presented in Chapter 4 are discussed in Subsection 5.2. Each of the algorithms is applied to multiple sets of throws and the deviation of the predicted interception position to the real impact position on the interception plane is recorded. Their applicability, advantages and disadvantages are presented.
Finally, the frame rate and resolution scaling analysis (Section 5.2.2) points out how the main parameters of the video system (frame rate and resolution) influence the prediction accuracy. This subsection will close this chapter.

5.1 Position Validation

In order to evaluate the prediction models and algorithms presented above a verification system is necessary. This verification system is used to measure the real position of the ball when passing the assumed catching range of the catching robot. Most gantry robots are similar to the one used by Barteit [BFK08] and move within one plane.
Different methods of measuring the position of the ball in space at an instant or within one plane exist. One possibility is to use a touch system while the other is to use an image sensor like a camera or a set of multiple cameras. Using a touch system is limited to objects like the tennis ball, non-flexible objects would demolish the touch system. Furthermore the mechanical wear has to be considered. Touch systems based on resistive and capacitive properties are widely available

in the size which is necessary to cover the impact area of the tennis ball.
Capacitive touch systems can not be used with objects like the tennis ball as they need finger input and are based on the concept of a changing charge due to the finger dip.
Resistive touch systems on the other hand are usable with any input object and thus they are also usable for detecting the tennis balls impact position. The mechanical wear of a resistive touch systems is a disadvantage on the other hand.
Another emerging touch system technology is Dispersive Signal Technology (DST) that is based on the propagation of waves on a glass substrate. A digital signal processor evaluates the information from four piezoelectric sensors in the corners and calculates the origin of the wave which is the point the touch system was touched. Devices based on DST are available from 32 " on and offer a resolution of 16384 by 16384 with a position detection error of less than 1 %. DST Touch systems also underly no mechanical wear as long as the glass substrate is not breaking. Pricewise the costs of a DST touch system are equivalet to the costs of a resisitve touch system of the same size with only few options for resisitve touch systems at the size of 32".
Infrared frames also have the feature that they do not underly any mechanical wear and they even can be used with non-flexible objects as the objects can be thrown through the frame and land on soft surface behind the frame. What refuses to use infrared frames is the low scan rate of commercially available models. Building an infrared frame especially for the purpose of the position detection similar to the frame used by Barteit [DBP09] has the downside of the complicated handling and the build up costs.
Using an image sensor or a set of images sensors like a video camera or a photo camera for the position determination is also possible. In case of using a photo camera the triggering is very important, in case of using a video camera this requirement does not need to be regarded. Due to the high velocity of the ball the small discrepancies of the triggering may result in big measurement errors. This is also valid for the video option - between two frames of a standard video (25 fps) the ball thrown with 10 m/s moves 0.4 m. Only using high speed cameras results in an accurate position detection. Still, processing of the video data is necessary to obtain the exact impact position and the costs of a high speed camera systems are magnitudes above the price of a touch system. A triggered photo camera system including the trigger hardware on the other hand is available for the equivalent costs of a touch system.
The previous mentioned approaches to detect the position of the ball on the plane can be grouped into invasive (touch systems) and noninvasive position detection methods. Invasive methods are characterized by the interruption of the flight due to the sensor. This has the downside of the mechanical forces on the thrown object and the sensor. While invasive methods are suitable for a thrown tennis ball and therefore selected here, the usage in future research with different objects is very limited. The following subsection will elaborate on the environment and the usage of the touch system.

5.1.1 Invasive Position Detection

The verification system for the interception position prediction is based on a 32 " DST touch system and is an invasive position detection system. Furthermore, the system is working independently of the interception position prediction. This allows to process the video data after the real interception point has already been determined and eliminates real-time requirements. Processing the same input data from the camera setup multiple times while tuning the prediction algorithm is enabled by this.
In order to put the DST touch system in position aluminum sheaths are used. These sheaths

offer a good expandability for future work. The supporting frame for the DST touch system, including also the mounting base for the cameras (compare Section 3.1) as well as for the additional floodlights, is shown in Figure 5.1. The touch systems main properties are presented in

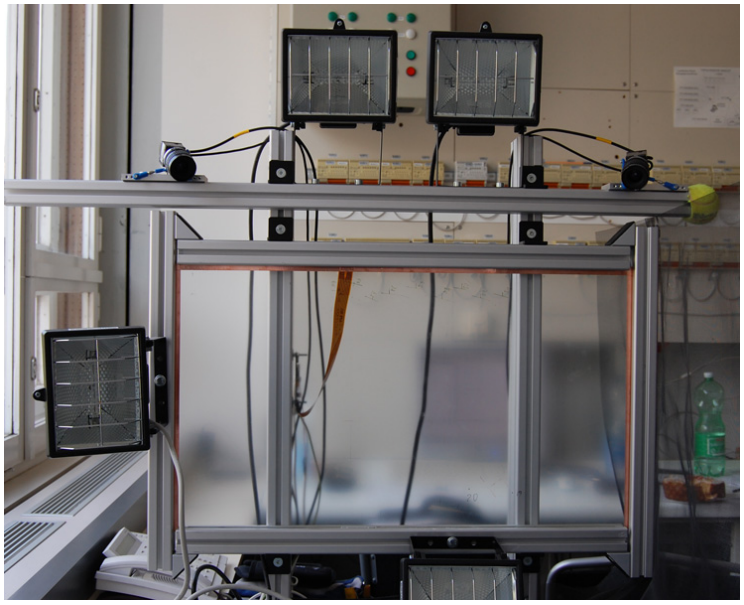

Figure 5.1: DST touch system mounting base

Table 5.1 [1]. The properties listed in Table 5.1 are provided by 3M, the manufacturer of the

Input Method	Finger and stylus input
Accuracy	Reportet coordinates are within1.0 % of true position
Active Area	727.15 × 408.05 mm
Resolution (h x v)	16384 × 16384 (maximum resolution)
Response Time	20 ms for tap input
Minimum Touch Impact	50 mN*s
Glass Thickness	2.2 mm (±0.2 mm)

Table 5.1: Properties of the DST touch system

touch System [1]. While the minimum touch impact is referenced no numbers are given for the maximum touch impact force. A advertisement video shows the DST touch system withstanding the impact of a steel ball weighting 0.45 kg dropped from 1.3 m height which equals the energy of 5.8 J. Comparing this to the energy of a tennis ball thrown with 10 m/s, which equals less than 3 J (depending on the weight which is allowed between 56.7 g and 58.5 g), allows the conclusion to use the DST touch system for the measurement of the balls impact position.
The accuracy of the DST touch system when using it to detect the impact position of the tennis ball was tested by applying chalk onto the ball previous to throwing it onto the touch system. The prepared ball leaves an imprint on the glass substrate plate which center is touched with a specially built marking tool shown in Figure 5.2. The accuracy determined through multiple throws (distance between the reported position of the ball impact and the reported position of the marking tools input) is below 6 mm. While this is not excellent it still allows the usage of the DST touch system as verification system because other methods offer the same level of accuracy with a more complicated handling or at higher costs [DBP09]. Parts of the reported error might

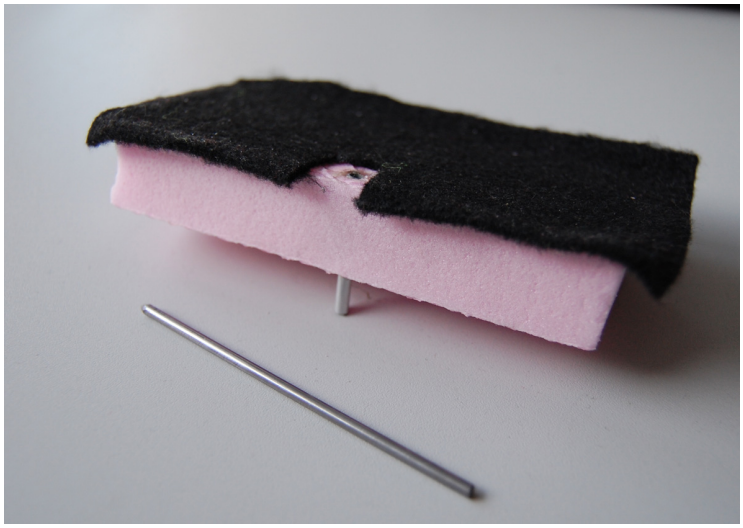

Figure 5.2: Marking tool with bolt used to touch the touch system

also be systematic as the verification of the DST touch system usability check was not done on a large number of samples.

5.1.2 Calibration

Similar to the calibration of the stereo vision setup presented in Section 3.2 also the verification system has to be calibrated. The position of the DST touch system in relation to the stereo vision system has to be determined and the DST touch system on it's own has to be calibrated. The second part of the calibration is done by using a large scale calibration sheet similar to the one shown in Figure 3.10 in Section 3.2. The four extreme corners were used to extract the relation between the real dimensions on the DST touch system and the pixels on the screen the touch kit is mapped to. The PC workstation (Table 3.4 in Section 3.1.3) offers three independent monitor interfaces. Two of them are used for Monitors while the third one uses a dummy connector [6]. The resolution on this dummy screen is set at 1600x1200 and the resulting resolution of the DST touch system is shown in Table 5.2.

To complete the calibration, the position of the touch system in relation to the vision system

Horizontal	0.428 mm/pixel
Vertical	0.321 mm/pixel

Table 5.2: DST touch system resolution

has to be determined. For this task also nine calibration sheets are used (compare Section 3.2.1). These nine calibration sheets are mounted onto a bracket and fixed on the touch system. The bracket is shown in Figure 5.3. The bracket allows to establish a relation between positions on the touch system and positions in the visual field of the stereo vision system which is necessary as the visual field does not contain the touch system at all. As the position of the calibration sheet on the bracket is known the camera positions in the "world coordinate system" (Figure 3.2) can be determined. One key attribute that made this calibration possible is the transparent calibration sheet on the bracket. It is necessary as both cameras are on the contrary side of the

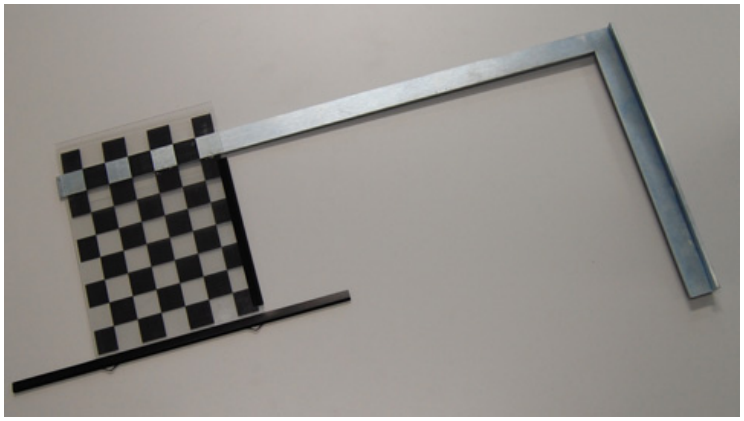

Figure 5.3: Bracket used for touch sysem calibration

calibration sheet (compare Figure 5.3). For this sheet the calibration pattern is printed onto a overhead foil which is then stiffened by two 0.3 mm thick transparent sheets. In order to provide enough contrast for the automatic corner extraction white planes were used as background for each calibration image. A sample set of calibration images is shown in Figure 5.4. As both

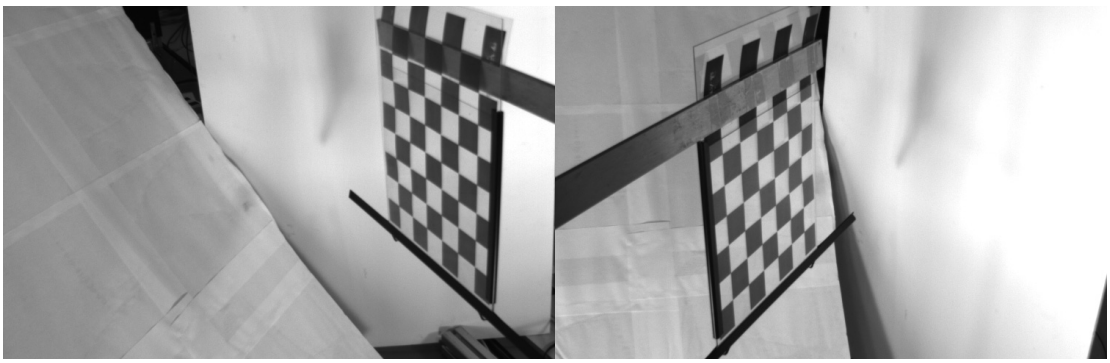

Figure 5.4: Sample set of calibration images used for the touch system calibration

cameras take the picture from the opposite side the extracted corners by the standard calibration had to be reordered in order to allow the merging of calibrations for the stereo calibration. The automatic extraction function only works on images taken from the same side as it orders the extracted corners automatically. During the stereo calibration these images are suppressed if the corner information from both images is not related. The merged information about the 9 calibration images used to calibrate the touch system is shown in Figure 5.5. Determination of the related positions in the world coordinate system uses the contact point of the bracket with the touch system. This position is marked, the bracket removed and the position extracted by a marking tool shown in Figure 5.2 of the previous Subsection 5.1.1. This tool allows to aim for the position through the pipe installed into the styrofoam. After the tool is aligned a bolt is used to touch the touch system multiple times. The foam applied on the styrofoam ensures that the mechanical attributes of the touch system are not altered. The average position of the touches is then used as the origin of the bracket. Flexible mounting of the transparent calibration sheet on the bracket allows the sheet to swing free and align vertical due to gravity. Information about the vertical direction on the touch system is extracted in a similar way - a weight on a thread is used. Based on this the positions of all corners in the "world coordinate system"

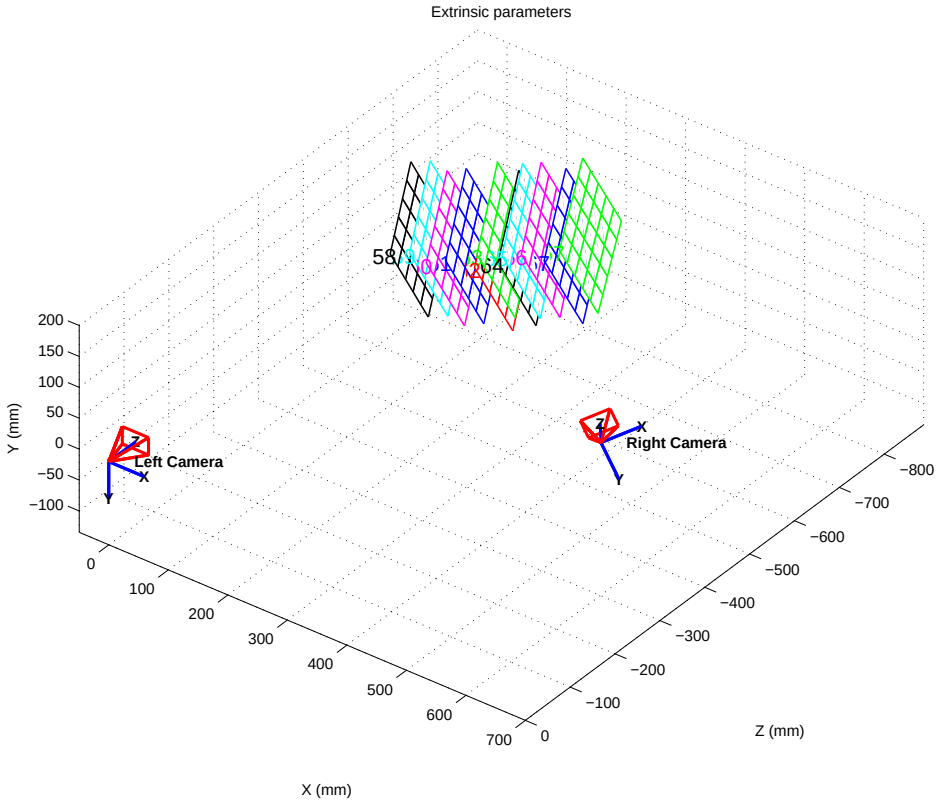

Figure 5.5: Nine calibration sheets mounted on the bracket in the camera coordinate system

are known. As the position information is also available in the camera coordinate system the rotation and translation between both coordinate systems can be calculated. This calculation has the goal to minimize the sum of all the distances between points in the "world coordinate system" and the related rotated and translated points of in the camera coordinate system. This is an optimization operation (compare Section 2.2.2). As both the rotation and the translation have to be determined the resulting operation is not linear. As no special timing requirements are valid a Monte Carlo algorithm is used to solve the optimization problem. 500 random rotations and 50 random translations are computed for 30 times with decreasing magnitude. First both sets of positions are standardized by subtracting the average of all positions of one set. This first step allows to start with selecting the best rotation, and afterwards search the best translation. The best fitting combination of the first iteration is stored and the second set is used to refine it. In each iteration the best additional rotation and translation is searched and the resulting set is stored. Table 5.3 presents the result of this calibration. A graphical representation of the

Rotation vector	[0.01082 −2.43803 1.40402]
Standardized translation vector	[0.28138 0.01515 0.051228]
Avg. camera coordinates	[203.55178 725.68757 81.97302]
Avg. world coordinates	[354.71029 136.37394 −645.00000]

Table 5.3: Rotation and translation of the touch system related to the cameras

calibration is shown in Figure 5.6. The translation vector shown in Table 5.3 is only valid if the averages are removed in each coordinate system. Summarizing both operations is possible

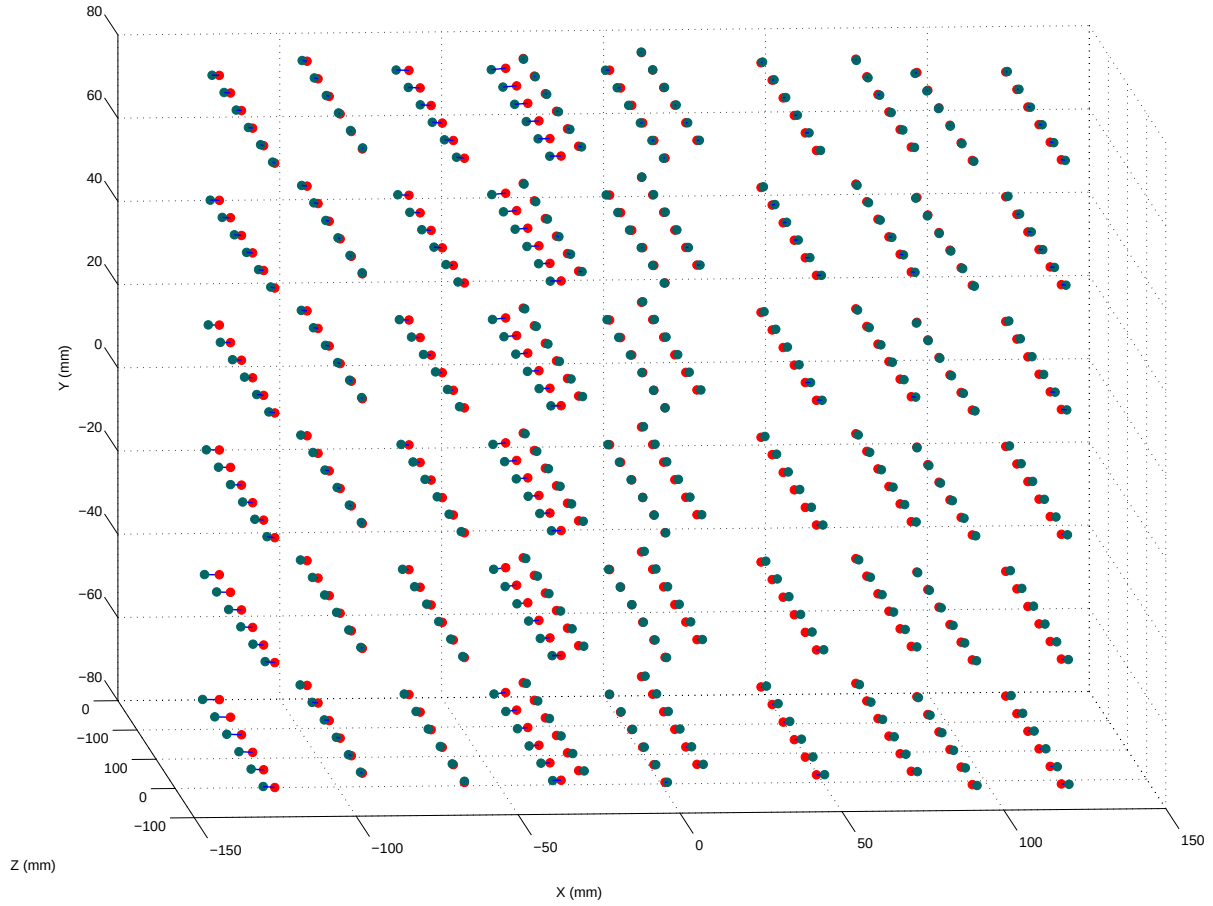

Figure 5.6: Comparison of corners in the average removed "world coordinate system" and the related rotated and translated corners in the camera coordinate system

by using the following equations results in one translation and one rotation which is similar to the rotation in Table 5.3. Symbols used are: coordinates in the world coordinate system X_{world}, coordinates in the camera coordinate system X_{camera}, respective average coordinates $\bar{X_{world}}$ and $\bar{X_{camera}}$, standardized translation vector $Trans_{std}$ and the resulting translation vector $Trans$. Table 5.4 presents the resulting rotation and translation vectors.

$$X_{world} = Rot * (X_{camera} - \bar{X_{camera}}) + bar X_{world} + Trans_{std} \tag{5.1}$$

$$X_{world} = Rot * X_{camera} + \bar{X_{world}} - Rot * \bar{X_{camera}} + Trans_{std} \tag{5.2}$$

$$X_{world} = Rot * X_{camera} + Trans + Trans_{std} \tag{5.3}$$

$$Trans = \bar{X_{world}} - Rot * \bar{X_{camera}} \tag{5.4}$$

Rotation vector	[0.01082 −2.43803 1.40402]
Translation vector	[691.6957 −199.7772 −54.6963]

Table 5.4: Rotation and translation of the touch system related to the cameras

5.2 Results

The quality measure of the prediction model and the related prediction algorithms is the deviation of the predicted impact position from the real/measured impact position. This final Section of Chapter 5 deals with the prediction results and the related deviation of the three models presented in Chapter 4 at frame rates of $50, 60, 70, 80$ fps at full resolution and $50, 60, 70, 80, 100, 120, 140, 160$ fps at halve resolution. First a detailed comparison of the prediction models at a frame rate of 60 fps and full resolution is shown in Subsection 5.2.1. The deviation of the prediction error in horizontal and vertical direction as well as the overall prediction error will be in focus of this section.
A comprehensive analysis of the prediction accuracies dependency on frame rate and resolution is presented in the second subsection (Section 5.2.2). The discussion of this dependency enables selection of a specific vision system for a required prediction accuracy.

5.2.1 Prediction Model Comparison

A overview of the prediction result is presented in Figure 5.7 for all models at 60 fps and full

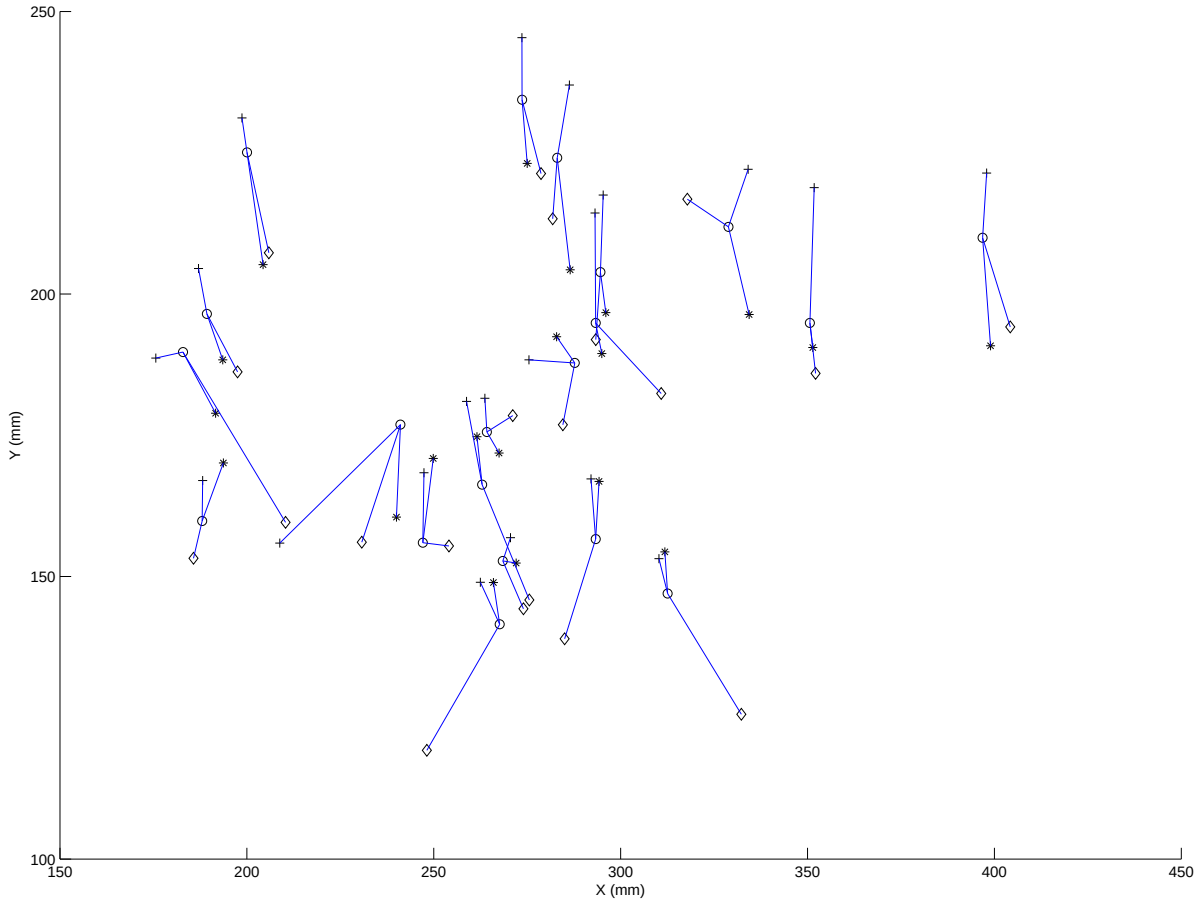

Figure 5.7: Prediction result at a frame rate of 60 fps; Detected impact positions: circle, prediction models: + polynomial, square physical, x separated

resolution. Related impact positions and prediction positions are connected by lines in order to enhance readability. The horizontal (Δx) and vertical (Δy) prediction error of the polynomial model is presented in Figure 5.8. Key properties of this prediction error is that the horizontal

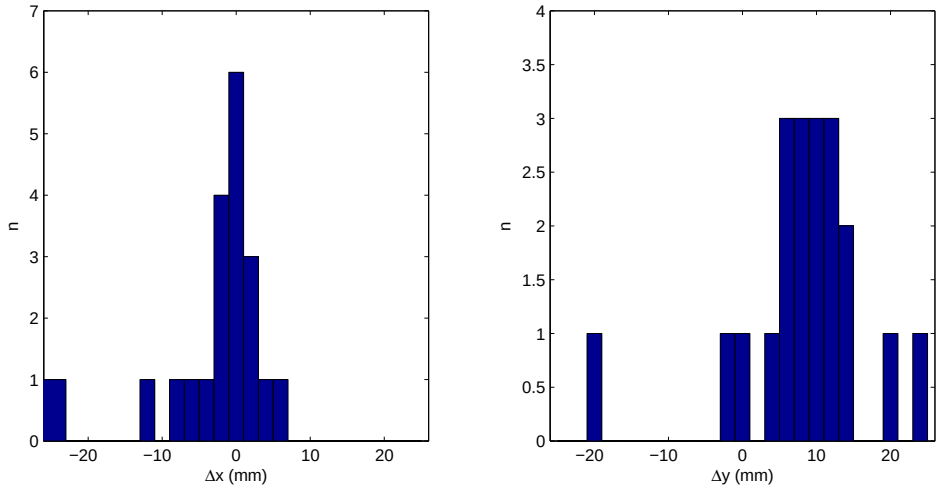

Figure 5.8: Histogram of horizontal error (Δx) and vertical error (Δy) of prediction based on the polynomial model

prediction error (Δx) has a bias close to zero (compare Table 1 in the Appendix). The bias of the vertical prediction error on the other hand is $\approx$ 10 mm. The variance in x-direction is smaller than in y-direction. This also means that the polynomial model (on average) describes the movement in x-direction better than the movement in y-direction. Figure 5.9 presents both

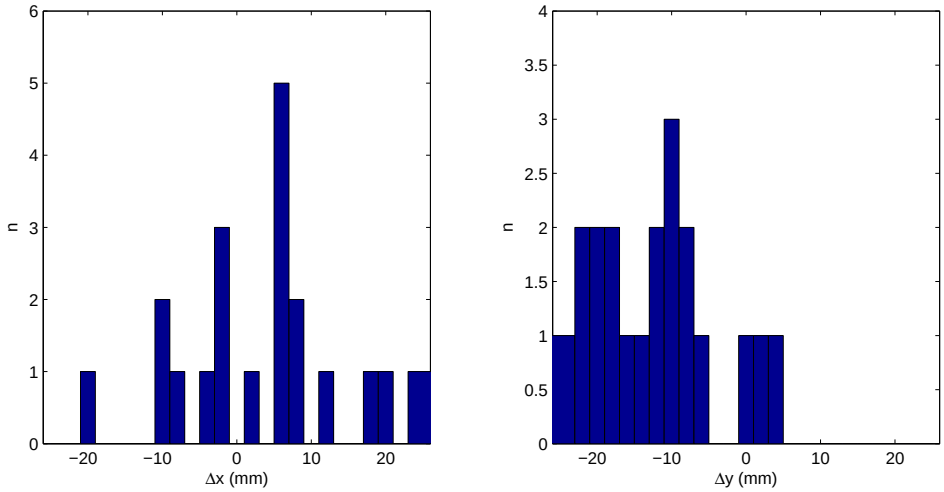

Figure 5.9: Histogram of horizontal error (Δx) and vertical error (Δy) of prediction based on the physical model

error deviation histograms for the iterative model. Both of these error have a significant bias of $\approx$ 6 mm in x-direction and ≈ -10 mm in y-direction (compare Table 2 in the Appendix). Also the variance of both errors is bigger in comparison to the errors of the polynomial model. Prediction erros based on the separated model are shown in Figure 5.10. Similar to the prediction based on

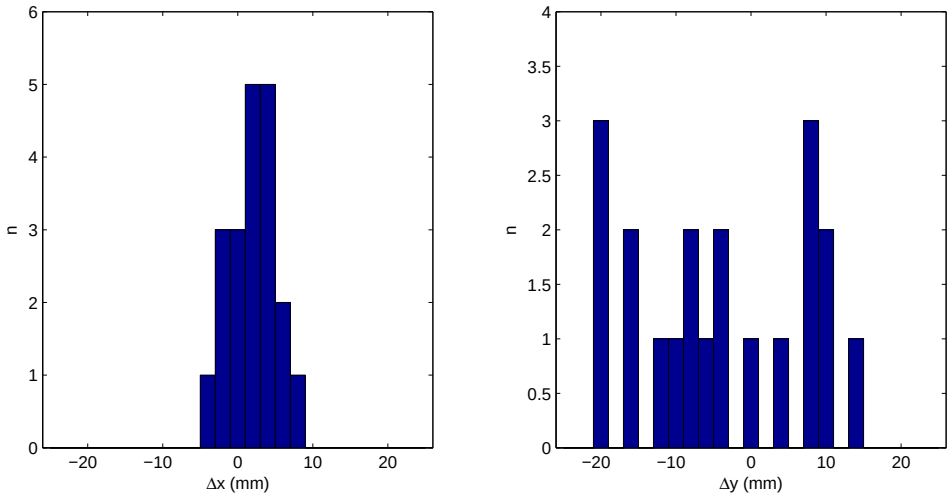

Figure 5.10: Histogram of horizontal error (Δx) and vertical error (Δy) of prediction based on the separated model

the polynomial model also the bias of the prediction error in horizontal direction is close to zero (≈ 2.7 mm; refer Table 3 in the Appendix). In addition also the bias of the vertical prediction is only approximately halve compared to both other models (≈ 5.5 mm). The horizontal variance is comparable to the polynomial model but smaller than when using the physical model. Vertical variance on the other hand is about double the size of both other models.
The spatial prediction error deviation of all models can be approximated by Gaussian distribution based on the shape on the histograms. This allows to model the overall prediction error (distance between the predicted and the measured impact position) by Rayleigh distribution and simplifies the comparison of the three models based on the overall error. The histogram of the overall error is shown in Figure 5.11. Judging from the histograms the separated model has the lowest overall

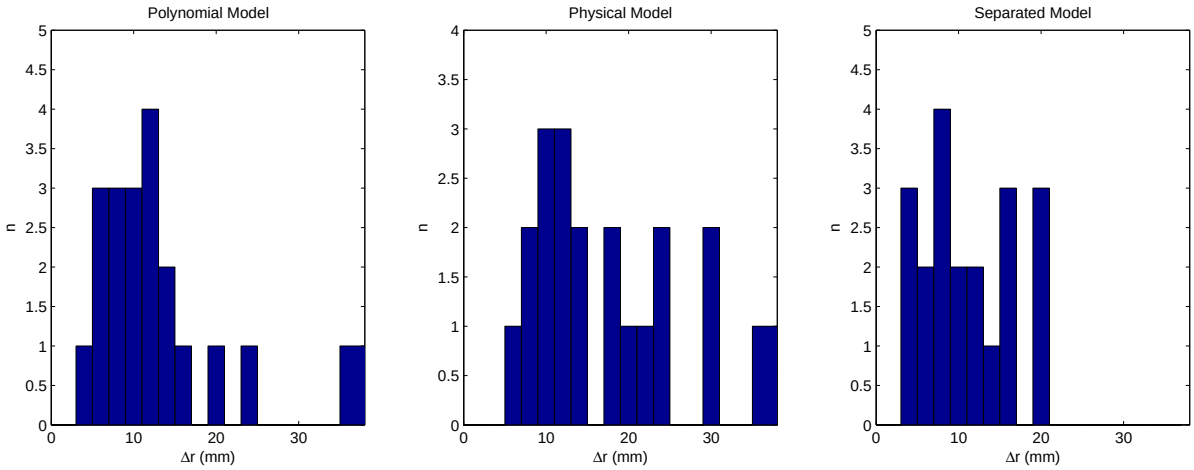

Figure 5.11: Histogram of overall error (Δr) of prediction for all prediction model

error of the three models. This is backed up by the parameters of the Rayleigh distribution used to model the distribution of the error. The three upper limits of the 99 % error band are 33.49

mm for the polynomial model, 43.79 mm for the physical model and 28.10 mm for the separated model. This is not only the case for this specific frame rate and resolution combination. Also on average over all 640 calculated predictions the separated model shows a ≈ 5 mm better prediction result when comparing the upper error band limit.

5.2.2 Frame Rate and Resolution Scaling Analysis

While the prior shown results were presented in order to give a detailed overview over the prediction results of each individual model, this section will deal with the relation between prediction accuracy and resolution as well as frame rate. The figures presented below all show the upper limit of the 99% error band of the overall prediction error r for different frame rates and resolutions. To minimize the impact of random errors only datasets based on the same input are compared. Such datasets based on the same video data are the halve resolutions low speed data $(50, 60, 70, 80)$ with respect to the high speed data $(100, 120, 140, 160)$ as well as the low speed data at full resolution with respect to the simulated halve resolution. Simulated halve resolution in this context means that video resolution is downscaled on the pc workstation based on the same binning algorithm that is used by the camera. This results in videos that show the same flight captured at the original resolution as well as the reduced resolution. Figure 5.12 presents the prediction errors resulting by the polynomial model. Comparing both resolutions of 576 x

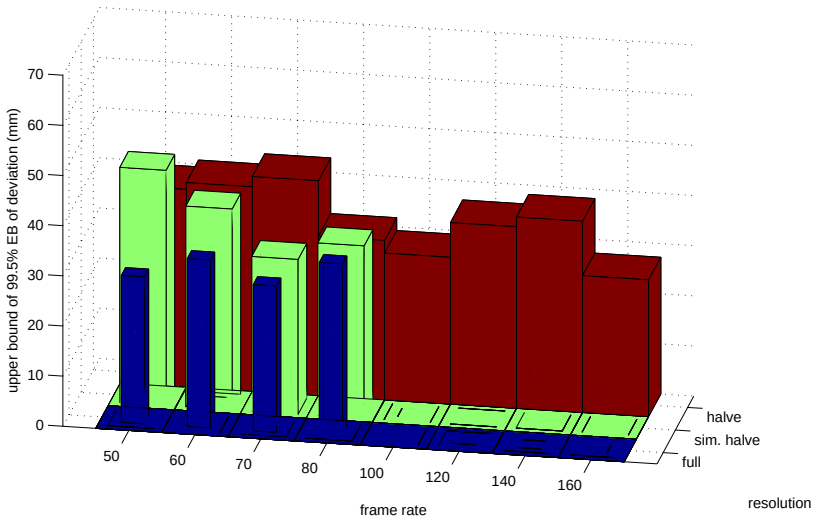

Figure 5.12: 99.5% bound of deviation of prediction based on the polynomial model of all frame rates and resolutions

480 pixel (full) and 286 x 240 pixel (simulated halve) the full resolution shows a prediction result better by ≈ 5 mm overall. This is because of the higher precision of balls position detection due to the higher resolution. It is interesting to note that doubling the frame rate at the lower resolution yields in a average improvement of ≈ 5 mm overall as well. The higher number of positions during the flight of the ball is responsible for this behavior. The numbers presented in Figure 5.12 are also show in Table 1 in the Appendix.
Figure 5.13 illustrates the prediction accuracy of the physical model over the various frame rate and resolution combinations. In contrast to Figure 5.12 the physical model shows unexpected

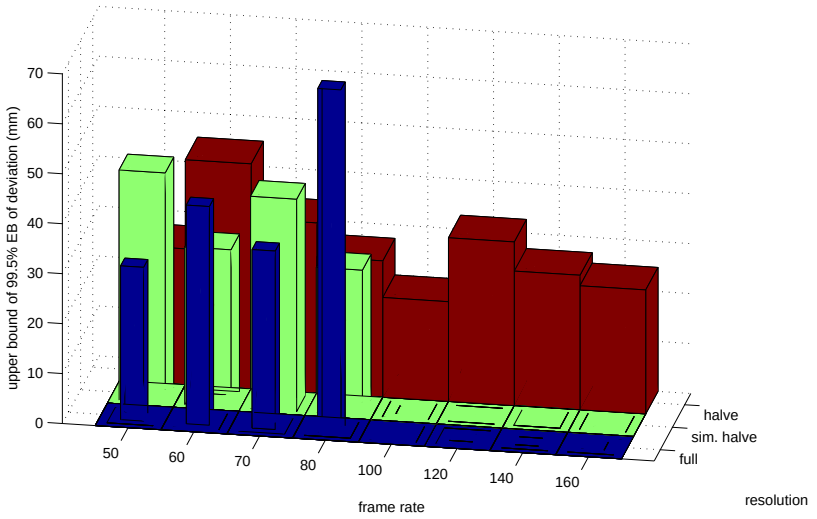

Figure 5.13: 99.5% bound of deviation of prediction based on the physical model of all frame rates and resolutions

behavior when reducing the resolution (simulated halve resolution) as the reduced resolution results in better prediction results with smaller errors. This unexpected result can be explained by the models sensitivity to errors in initialization of the parameters and only occurs in case of 60 and 80 fps at simulated halve resolution. Taking a look at the comparison of the higher frame rates at halve resolution, the decreasing prediction error with increased frame rate is illustrated. On average this error is reduced by ≈ 15 mm when doubling the frame rate.
Finally also the results dependent on resolution and frame rate of the spatial separated model are compared in Figure 5.14. Similar to Figure 5.12 increased resolution as well as frame rate

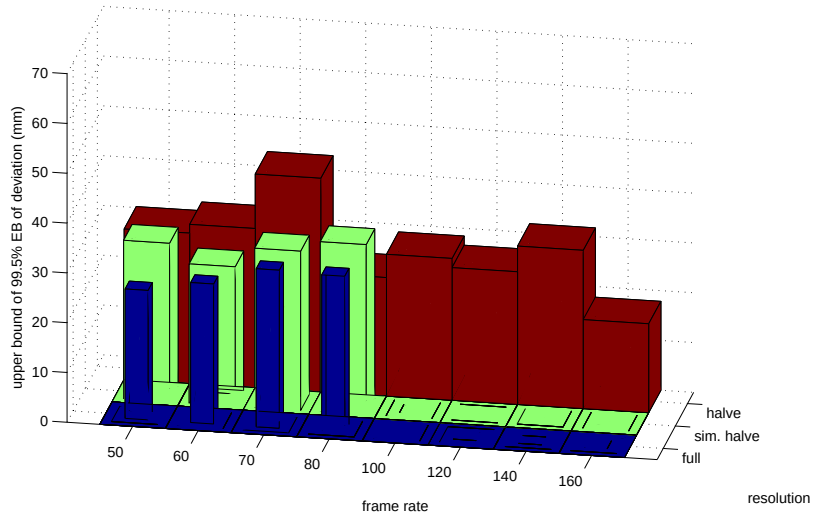

Figure 5.14: 99.5% bound of deviation of prediction based on the separated model of all frame rates and resolutions

decreases the prediction error. Also the simulated halve resolution approves this compared to

the physical model. Overall this model shows the best prediction results (compare Section 5.2.1). The gain moving from halve to full resolution is only ≈ 2 mm while the gain when doubling the frame rate is ≈ 6 mm on average.

6 Conclusion and Future Work

In the introduction transport by catching and throwing is stated as one idea to deal with the rising demand for flexible transportation systems. The presented work is an approach based on a stereo vision system and dealing with a tennis ball as thrown object. In addition to the two questions raised in the introduction the Question whether such a system can be used as reliable transportation system will be answered below. In addition ideas and options to enhance this approach are discussed in Subsection 6.2.

6.1 Conclusion

In this particular case the question whether the transport by throwing approach is a viable solution for a transportation system can clearly be answered with yes. The prediction accuracy, dependent on the model used, paired with a gripper offering a catching range between 3 and 6 cm enables the realization of a reliable transportation system over a transportation range of 3 meters. No special high speed cameras are needed to achieve this, two cost effective industrial cameras paired with a standard PC are sufficient for this accuracy. The real-time constraint is not considered at this stage but will be discussed below.
A comparison of the three prediction models presented is shown in Figure 6.1. The average

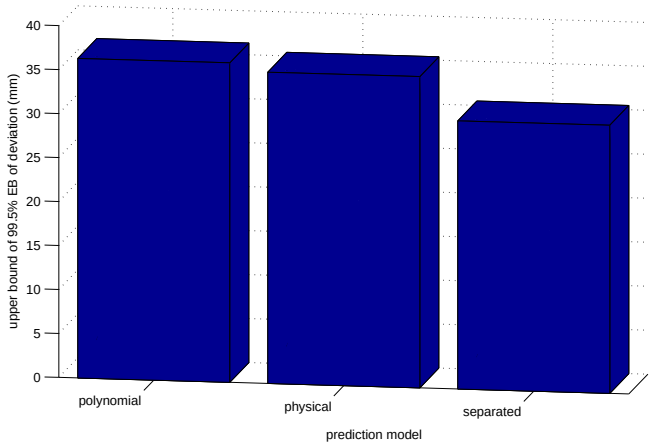

Figure 6.1: Predictions average upper 99.5% bound of error band per model

prediction accuracy (calculated as average of all frame rate and resolutions combinations upper 99.5 % error band) of each model clearly presents a prediction accuracy adavantage for the separated model. Also the physical model shows a slightly more accurate prediction result than the polynomial model. This behavior can be explained by the fact that both physics-based models are less sensitive to small measurment errors of the position acquisition system as the shape of the trjajectory is predefinded by the forces considered. Considering the fact that both physically founded models are not able to deal with the impact of spin on the trajectory this result is impressive. This is the answer to the question raised in Section 1.3. The models used to obtain the trajectory have a significant influence on the prediction accuracy. While the physics based models are not perfect, they still lead to a more accuracte prediction results despite the fact that the influence of spin is not modeled at all.
In addition to the pure contemplation of the prediction accuracy also the computation time for the prediction has to be considered in combination with the accuracy. Figure 6.2 presents this

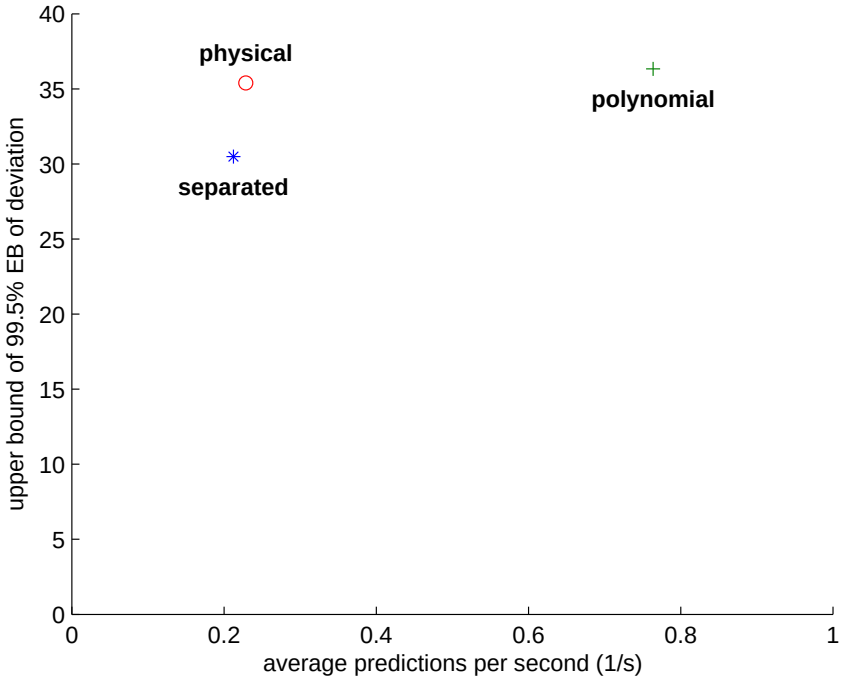

Figure 6.2: Prediction efficiency per model; data points in the right lower part of the diagram are high efficient models, data points in the left upper part less efficient ones

prediction efficiency. Comparison of two models efficiency can easily be done by connection of ones model representing point with the origin of the diagram. Points lying above or left of this line (or its extension) represent less efficient models, points below/on the right side of the line represent more efficient ones. Although only delivering the least accurate prediction result the polynomial model is the most efficient one when taking the computation time into account. The separated models calculation time is approximately four times as long as the time for the polynomial model and the better accuracy does not balance this drawback. Also the physical model shows this calculation time drawback. The degree of this drawback has to be considered with the fact in mind that none of the models calculation time allows real-time usage in the presented way. In case the same models and algorithms are implemented on an effective platform, e.g. in

hardware (FPGA), all models are applicable for real-time environments as well.
Cost effective realization of the transport by throwing approach demands knowledge of the parameters influencing the interception position prediction accuracy. For a given catching range of a gripper the required attributes of the prediction system can be derived. One factor influencing was already presented and the accuracy of different models as well as their efficiency regarding the calculation time was discussed. Also the parameters of the vision system play a role. Figure 6.3 presents the benefits when doubling either the image resolution of the camera (in each

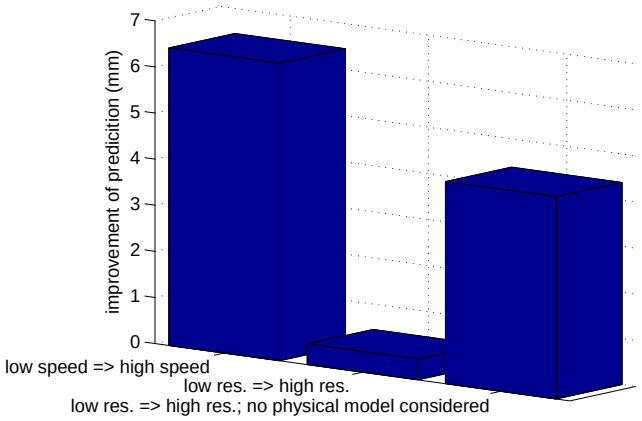

Figure 6.3: Prediction improvement depending on resolution and frame rate

direction, results in four times the pixel count for the whole image) or doubling the frame rate. Due to the behavior shown by the physical model also the improvement for both other models is shown in Figure 6.3. The prediction accuracy improvement when doubling the frame rate bests the improvement when using higher resolution (not considering the physical model) by ≈ 50 % or 2 mm. This also means that under bandwidth constraints of the camera interface a camera system offering higher speed at a lower resolution is preferred compared to a lower speed higher resolution system.
Recalling the second questions raised in Section 1.3 about the improvement when using two instead of one cameras [BFK08] the prediction accuracy accomplished by Barteit is stated that the prediction misses the impact position with more than 20 mm occurs in less than 10 % of 45 examinded throws. This prediction accuracy is comparable to the result obtained using two cameras. The upper 90 % error bound at a frame rate of 80 fps (compared to 87 fps used by Barteit) is 20.7 mm for the separated model. Overall variances of the error bound are significant. The best accuracy at one frame rate/resolution combination is 11.8 mm (160 fps, halve resolution) for example while the worst is 28.3 mm (70 fps, halve resolution). A head-to-head comparison using the same verification system can lead to a significant answer.

6.2 Future Work

One main influence of the prediction result, the weights which are used to give priority to positions acquired close to the camera in contrast to the positions acquired far off the cameras, have been left out of scope completely. A suitable set of weights which is equal for all the three prediction models was found and used throughout the work but when experimenting with different

sets of weights their strong influence on the results became obvious. Further work which includes systematic analysis of the impact of different weight-sets as well as searching for an optimal set, possible individual for each prediction model, has a big chance to improve the prediction result further.
Another decisive aspect was found in the spin which the throwing device puts on the tennis ball. The Magnus effect, which pays account to the influence of the spin, is not considered in both models based on the flight physics. Considering the spin as well might lead to a reduction of the horizontal prediction bias in the physical based models. This demands finding a solution for the problematic task of measuring the spin. Another possibility to examine the influence of the spin is to modify or exchange the throwing device in order to minimize the tennis balls rotation.
Also using different objects to throw, by example a ball without hair on the surface could lead to significant less influence of the spin on the trajectory and thus lead to information about the relevance to model the spin at all. But not only different balls can be thrown. As the target application of this transport by throwing approach is a versatile system throwing small boxes, cans or similar objects is the logical next step. Adaption of the impact position detection system as well as the object detection and flight modeling has to be done in this case as well. Another possible approach to deal with multiple shapes of objects to throw is to pack the object into a round transport container at the origin (or throwing) station, throw the object and unpack the object at the target (or catching) station again.
In coherence with the physical model further optimization of the fitting algorithm shows promise. While this model showed the largest prediction error for some combinations of frame rate and resolution the overall precision was not far off the other models. This means that in case this outliers of the prediction result can be reduced or excluded at all this model has the chance deliver more accurate prediction results. The theoretical advantage of the most accurate flight trajectory description as well as similar computation time to the separated model might manifest in prediction results as well. Another aspect in this context is that all the models are not fast enough for real time usage at the present stage on the present platform. Moving to a fully or partly hardware-based implementation is a simple solution for this problem but another interesting approach is to change the way the physical model is implemented at all. Instead of calculating a huge number of possible flight trajectories during the task of prediction a smaller set of predefined trajectories might be used and instead of picking the best and vary the initial parameters a little in order to find an even better fitting trajectory all the predefined trajectories might be translated and rotated in order to fit the positions calculated best. All the models might be used for this but the physical model is considered to have to highest potential when using this approach.
Another way to improve the performance of the prediction is to distribute the calculation over the different components of the whole system. Even inside the camera the information could be merged in a way that only the center position of the tennis ball in each captured frame is sent via the interface to the host system. Frame differencing, edge detection and hough transformation can be realized by a signal processor or (if the performance is not sufficient) by a number of signal processors. This approach is similar to the way humans use the information the eye is capturing each moment. Only the relevant information is sent to the brain, preprocessing is already done very close to the optical sensor, in this case the human eye.
In case the computation effort can be distributed over the components of the system usage of more than two cameras is reasonable as well. The effect of the reduced position calculation precision in case the ball is far off the camera can be countered by installing another pair of cameras close to the throwing device. Even more than four cameras can be used. Similar to adding more cameras also the influence of different positions of the cameras used has been left out of scope in

this work. Different positions might lead to a better prediction result. In addition also different optics and resulting field of view can be examined.
Also the chronological verification of the impact has not been discussed. In the target application the catching device has to close the gripper at the right instant in case an active gripper is used. In case of a passive gripper this requirement disappears. In this context also the development of the prediction over time has to be mentioned. If accurate interception position is already available shortly after the object has been thrown the mechanical stress and energy required to position the catching device accurately in time are minimized which lowers the cost of such a system. This development of the prediction has also not been examined in here but is a main attribute for the final application of such a transport by throwing and catching system.
Considering the work presented from a further distance a lot of possible applications are recognizable. Not only transportation for different shaped goods within production facilities can be done by throwing and catching. Automated rack filling systems in supermarkets can be based on catching and throwing. The workforce needed to refill the racks can be replaced by number of autonomous carriers with throwing devices as well as a number of catching robots on the racks. These carriers throw the goods to the catching robots that grip it and place it on the rack.
Another possible application is a sorting system for parcels. Instead of using systems based on conveyor belts sorting by throwing and catching is a viable alternative which requires less space and is more flexible. Basically such a transportation system can be used in every case where humans would rely on throwing and catching as well.

Appendix

Polynomical Line Fitting based Results

Prediction errors for each direction (x and y) are presented in histograms in Figure 4 and Figure 5. presents the histograms of the horizontal prediction error for each frame rate and resolution

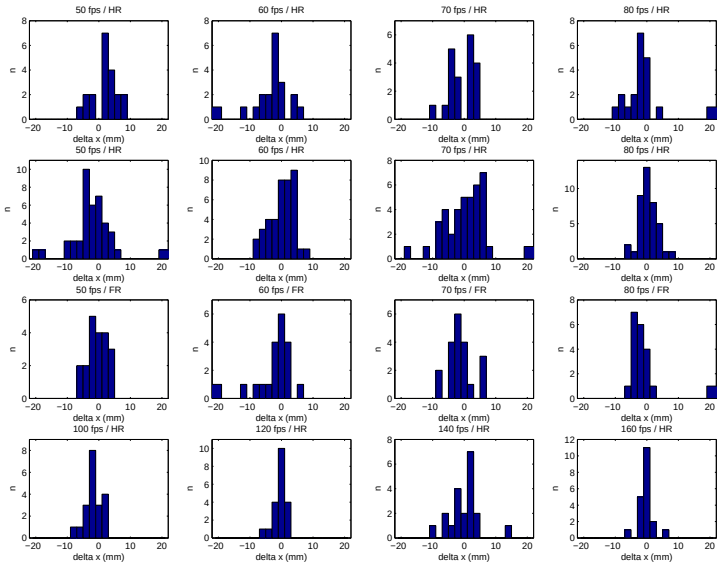

Figure 4: Histograms of horizontal deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

examined. For all of the frame rates and resolutions the bias of the error is close to zero. This can also be seen in Table 1 at the end of this subsection which summarizes all the prediction errors. For most of the predicted positions the x-coordinate of the impact position is predicted within 10 mm of the real position. The shape of the histogram also shows that the error of the prediction in this direction can be modeled by Gaussian distribution. Comparing the 2^{nd} and 4^{th} row, which are based on the same video data but the results presented in the 2^{nd} row only uses 1 out of two following frames it can be noted that the variance of the error decreases with increased frame rate. This is also valid for the 1^{st} and 3^{rd} row where the error increases as the resolution is decreased.
Histograms of the vertical deviation are shown in Figure 5.Compared to the deviation in x-

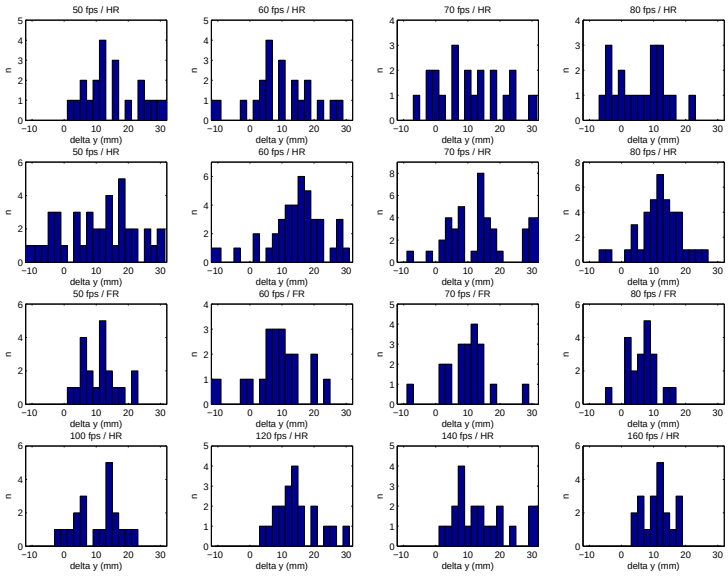

Figure 5: Histograms of vertical deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

direction the y-direction shows a bias in the error histograms for all the resolution/frame rate combinations (Note that zero is not in the middle of the x-axis!). As this error is positive the predicted impact position is below the real impact position on the catching plane in average (compare Figure 3.2). The magnitude of this error is of ≈ 11 mm overall (compare also Table 1) and using Gaussian distribution to model it is also feasible. Besides the bias the variation of the error is larger by a factor $2-3$ as well. This might be a result the larger number of forces which influence the flight in this direction. The behavior of decreased prediction accuracy with lowered resolution or decreased frame rate is also valid for the vertical prediction error.
In order to give a good overview of the prediction results Figure 6 illustrates the spreading of the overall prediction error which is the deviation of the predicted impact position from the measured impact position. Compared to the spatial error which includes negative values along the x-axis this is not the case in the histograms of the overall deviation as this is the absolute distance between the predicted and the real impact position. As a result Gaussian distribution is not applicable but Rayleigh distribution suits describing the norm of a vector whose components are independently distributed by Gaussian distribution which is the case (compare Figure 4 and Figure 5 in Appendix). As a result of the behavior shown by the components of the error also the overall error increases with decreased resolution and/or lower frame rate. This error is mainly influenced by the error of the vertical prediction as the error of the horizontal prediction is smaller by a magnitude of $2-3$.
The prediction results based on the polynomial model, topic of Section 4.1, for all frame rates and both resolutions are presented in Table 1. The used symbols in this table are: average prediction error in horizontal direction $\bar{\Delta x}$, standard deviation of average prediction error in horizontal direction $\sigma_{\Delta x}$, 99% error band bounds of the prediction error in horizontal direction, respective symbols for vertical or y-direction, average overall deviation $\bar{r}$ with the related 99% error band. The first column of the table shows the frame rate the data is based on while the second column presents the number of datasets which are used to build up the statistics. The relevant information which could also be extracted out of the histograms is manifested here in numbers. The average prediction error in x-direction (horizontal) is below 1 mm for all the 640

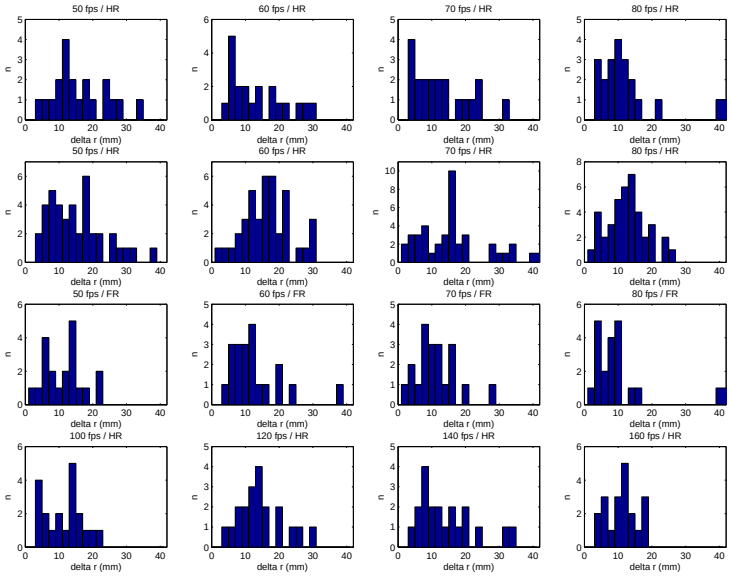

Figure 6: Histograms of overall deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

predictions while the standad deviation is ≈ 5.5 mm. In contrast to this the mean displacement of the prediction in y-direction (vertical) is $\approx$ 11 mm with a standard deviation of $\approx$ 8 mm. Comparing individual datasets is not as feasible as comparing sets which are based on the same video data. Comparable datasets are the four predictions at full resolution and the four sets of the simulated halve resolution (which are obtained by reducing the resolutions of the full resolution images with the same algortih that is used in the cameras for resolution reduction) and the bunch of lower framerate $(50, 60, 70, 80)$ fps at halve resultion in combination with the higher framerate predictions (above including 100 fps). A detailed graphical reprensentation as well as further discussion of the influence of framerate and resolution is topic in Section 5.2.2.

		hor. deviation			vert. deviation			overall deviation	
fps	n	Δx	$\sigma_{\Delta x}$	99% EB	Δy	$\sigma_{\Delta y}$	99% EB	$\bar{r}$	99% EB
Full resolution									
50	20	-0.65	2.95	[-8.25 6.96]	11.02	5.56	[-3.31 25.36]	8.99	[0.90 29.26]
60	20	-2.75	7.74	[-22.69 17.19]	8.19	8.78	[-14.43 30.81]	10.29	[1.03 33.49]
70	20	-1.25	3.84	[-11.15 8.65]	9.87	6.83	[-7.73 27.46]	8.95	[0.90 29.15]
80	20	0.51	12.49	[-31.66 32.68]	6.78	4.58	[-5.02 18.57]	10.56	[1.06 34.38]
Simulated halve resolution									
50	20	2.61	5.15	[-10.66 15.88]	16.81	10.04	[-9.05 42.67]	14.43	[1.45 46.99]
60	20	-4.18	8.47	[-26.00 17.64]	10.43	10.27	[-16.02 36.88]	12.32	[1.23 40.09]
70	20	-1.41	3.67	[-10.87 8.05]	8.50	9.62	[-16.28 33.28]	9.49	[0.95 30.90]
80	20	-0.71	12.95	[-34.06 32.64]	3.97	6.32	[-12.33 20.26]	10.58	[1.06 34.44]
Halve resolution									
50	40	-2.13	6.59	[-19.12 14.85]	10.54	11.61	[-19.37 40.46]	12.13	[1.21 39.47]
60	40	0.29	3.63	[-9.06 9.64]	14.66	9.29	[-9.26 38.59]	12.54	[1.26 40.82]
70	40	0.36	6.89	[-17.39 18.12]	13.85	10.38	[-12.89 40.60]	13.18	[1.32 42.90]
80	40	0.39	2.83	[-6.90 7.67]	11.90	6.42	[-4.65 28.45]	9.77	[0.98 31.81]
100	20	-1.78	2.73	[-8.81 5.24]	10.30	6.71	[-7.00 27.59]	8.99	[0.90 29.27]
120	20	-0.39	1.92	[-5.34 4.56]	14.21	6.45	[-2.42 30.83]	11.12	[1.11 36.20]
140	20	0.14	4.57	[-11.62 11.91]	13.67	8.17	[-7.38 34.72]	11.71	[1.17 38.13]
160	20	-0.30	2.00	[-5.46 4.87]	10.82	4.39	[-0.49 22.12]	8.38	[0.84 27.27]
overall	640	-0.67	5.78	[-15.54 14.21]	11.09	8.31	[-10.31 32.49]	11.16	[1.12 36.33]

Table 1: Prediction results based on the polynomial model, all numbers in millimeters

Physical Model based Results - 3 pages

The distribution of the error in the spatial directions is shown in Figure 7 and Figure 8. Similar to the prior section also for the physical model the histogram of the overall error is presented in Figure 9. Comparing the 2^{nd} row with the 4^{th} row it is possible to see the influence of the doubled frame rate. Also in this case the prediction error decreases with increased frame rate. On contrary the behavior shown when downscaling the image (1^{st} and 3^{rd} row) does show unexpected behavior as the prediction errors average is smaller when using the downscaled image. This behavior is due to individual throws which have a prediction error of up to 100 mm. Examination of these throws has shown that the problem lies within the initialization of the prediction parameters. Still the frequent occurrence only in case of the full resolution is not fully clarified and would otherwise have been corrected. One possible reason is that small changes in the image are captured on a higher number of pixels in full resolution which might still influence the calculated position of the tennis ball in space. This assumption could not be verified by comparison of the calculated positions as bigger prediction errors do not occur at the throws where the variation of the positions calculated between full and halve resolution is visible.
Similar to Table 1 in the prior section Table 2 shows the results of the prediction based on the physical model. Symbols in this table are the same as in Table 1. Refer to the first part of Section 6.2 for detailed description. In comparison to the polynomial model both spatial error show a bias of ≈ 5 mm. As the bias of the x-direction is positive and the bias of the y-direction is negative all the balls are predicted right above the impact position seen from the throwing device on average. The bias in horizontal direction can be explained by taking another look at Figure 4.4 and an examination of the image data which is presented in Figure 3.17. Due to the fact that the throwing device is based on a leg spring (compare Figure 3.1) the tennis ball is

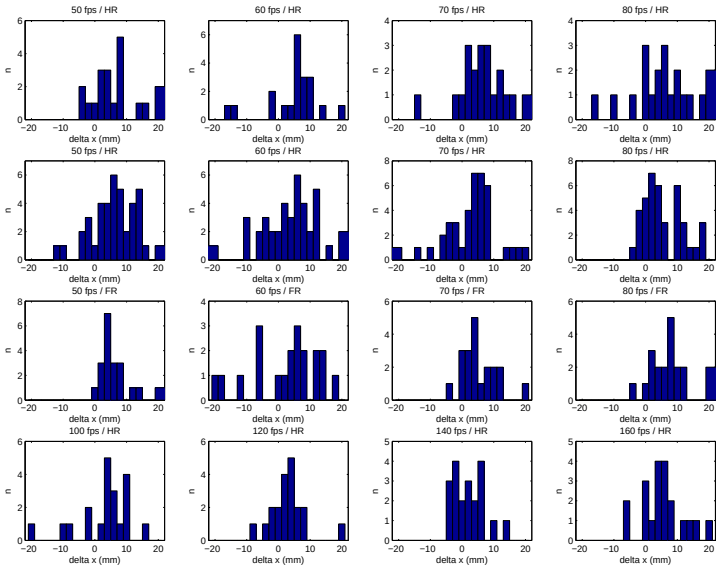

Figure 7: Histograms of horizontal deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

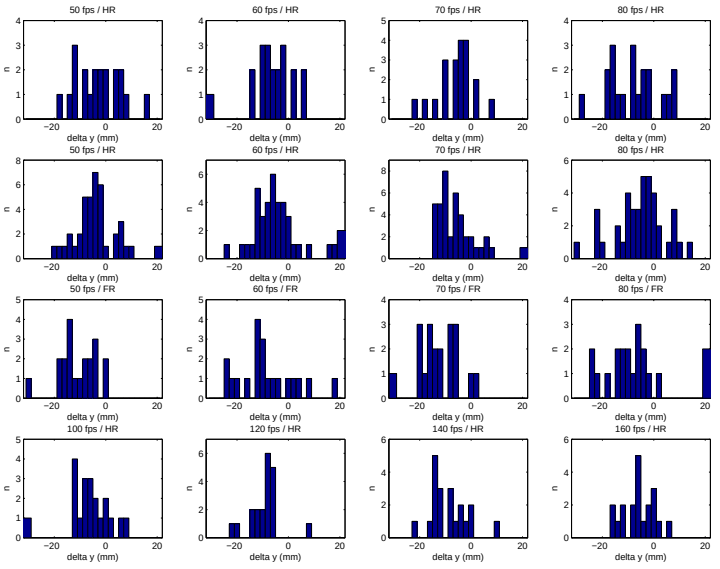

Figure 8: Histograms of vertical deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

rotating during the flight. This results in another force of the tennis ball which is not considered – the force based on the magnus effect (compare Table 2.4). The rotation direction influences mainly the balls vertical movement only in the way that the balls trajectory is flying a curve to the left looking from the throwing device towards the impact position detection plane. This is also visible in Figure 4.4 where also the the resulting prediction error can be approximated. In coherence with this behavior also the fact that the final calculated parameter $\mathbf{g}$ of this model was not aligned parallel to the y-axis of the world coordinate system (Figure 3.2) but directed also a little into the negative x-direction has to be mentioned. The reason for this lies in the fact that the model tries to fit the data as good as possible and in order to pay account to the curved

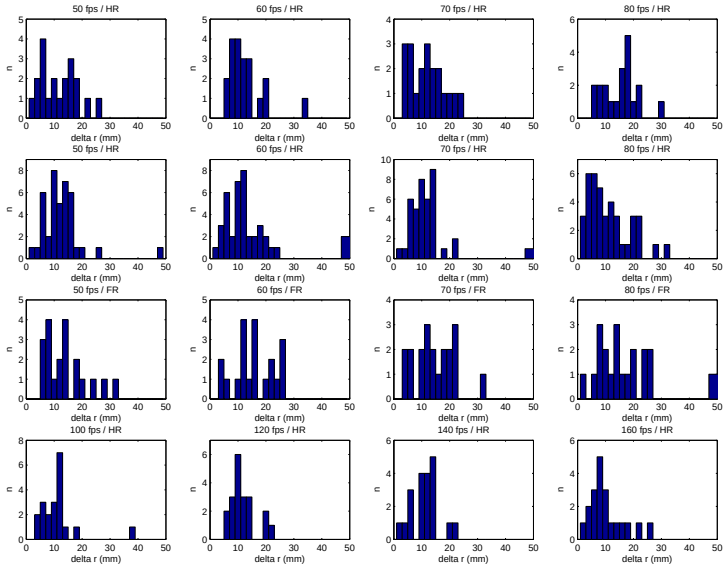

Figure 9: Histograms of overall deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

trajectory **g** was calculated in a way to model the curve as well. It is interesting to note that the vertical bias is smaller than the one of the polynomial model (compare Table 1) which can be explained by the consideration of all main forces influencing the vertical movement and the fact that the spin, which is not considered at all, does not influence this movement as the rotation axis of the ball is aligned with the y-axis.

		hor. deviation			vert. deviation			overall deviation	
fps	n	Δx	$\sigma_{\Delta x}$	99% EB	Δy	$\sigma_{\Delta y}$	99% EB	$\bar{r}$	99% EB
Full resolution									
50	20	5.34	5.96	[-10.01 20.69]	-8.07	7.04	[-26.20 10.06]	9.45	[0.95 30.76]
60	20	3.39	10.92	[-24.74 31.52]	-12.62	8.48	[-34.46 9.22]	13.45	[1.35 43.79]
70	20	6.11	7.17	[-12.37 24.59]	-10.33	6.71	[-27.62 6.95]	10.97	[1.10 35.70]
80	20	13.43	20.33	[-38.94 65.80]	-11.83	12.64	[-44.39 20.73]	21.13	[2.12 68.80]
Simulated halve resolution									
50	20	6.42	10.94	[-21.76 34.61]	-1.64	15.42	[-41.37 38.08]	14.17	[1.42 46.12]
60	20	4.35	6.76	[-13.07 21.76]	-8.59	7.04	[-26.72 9.54]	9.69	[0.97 31.55]
70	20	4.75	7.03	[-13.36 22.87]	-5.50	15.46	[-45.32 34.31]	13.06	[1.31 42.52]
80	20	6.30	7.81	[-13.81 26.41]	-5.72	5.22	[-19.16 7.72]	8.96	[0.90 29.16]
Halve resolution									
50	40	4.15	6.10	[-11.56 19.86]	-5.23	7.61	[-24.83 14.38]	8.36	[0.84 27.20]
60	40	6.04	10.03	[-19.80 31.88]	-3.81	15.22	[-43.01 35.40]	13.84	[1.39 45.06]
70	40	2.12	7.46	[-17.11 21.34]	-10.08	7.57	[-29.57 9.41]	10.47	[1.05 34.07]
80	40	5.57	6.20	[-10.40 21.53]	-4.14	7.36	[-23.11 14.83]	8.39	[0.84 27.31]
100	20	4.19	4.77	[-8.11 16.49]	-2.65	5.28	[-16.26 10.96]	6.13	[0.61 19.97]
120	20	4.45	9.00	[-18.73 27.63]	-8.21	5.87	[-23.33 6.91]	10.07	[1.01 32.77]
140	20	3.37	5.71	[-11.33 18.07]	-7.62	5.90	[-22.82 7.58]	8.27	[0.83 26.92]
160	20	4.14	5.60	[-10.29 18.57]	-2.55	7.83	[-22.72 17.62]	7.63	[0.76 24.83]
overall	640	5.16	8.09	[-15.68 25.99]	-6.39	9.23	[-30.17 17.39]	10.87	[1.09 35.40]

Table 2: Prediction results based on the physical model, all numbers in millimeters

Spatial Separated Physical Model based Results - 3 pages

Histograms of the spatial errors are again presented in Figure 10 and Figure 11 in Appendix. The distribution of the prediction errors for the spatial separated model, presented in Section 4.3, is shown in Figure 12. The spatial separated model shows similar behavior to the polynomial model as a higher frame rate (compare rows 2 and 4) results in smaller prediction errors and also increasing the resolution improves the prediction result (1^{st} and 2^{nd} row). The overall prediction error is smaller than the prediction error of the physical model and comparable to the prediction error of the polynomial model. This is also confirmed by taking a look at the statistics presented in Table 3. Symbols in this table are the same as in Table 1 and Table 2. Refer to section 6.2 for detailed description. The most interesting fact about this prediction model is that the bias of the horizontal as well as the vertical prediction is below 2 mm on average. While the standard deviation of the horizontal error is small as well (3.18 mm) the standard deviation of the vertical prediction error is 11.2 mm on average. Comparing this numbers to the respective numbers of both prior presented models the horizontal error standard deviation is the smallest of all model while the vertical is the largest. The small error of the x-direction error was not expected as the spin which is suspected for the bias of the physical models prediction is also not included in this model. The horizontal error on the other hand lies in line with the physical model while the polynomial model has a smaller standard deviation of only ≈ 8 mm on average. Even if the horizontal errors standard deviation is similar or bigger compared to the other models the overall error of this model bests the other models (polynomial ≈ 11 mm, physical ≈ 12 mm) with an average error of 9.34 mm. The overall stability of this model is significant as the three highest upper bound of a overall error band are ≈ 43 mm, ≈ 33 mm and ≈ 33 mm compared to ≈ 89 mm, ≈ 63 mm and ≈ 47 mm for the physical model and ≈ 43 mm, ≈ 41 mm and ≈ 41 mm for

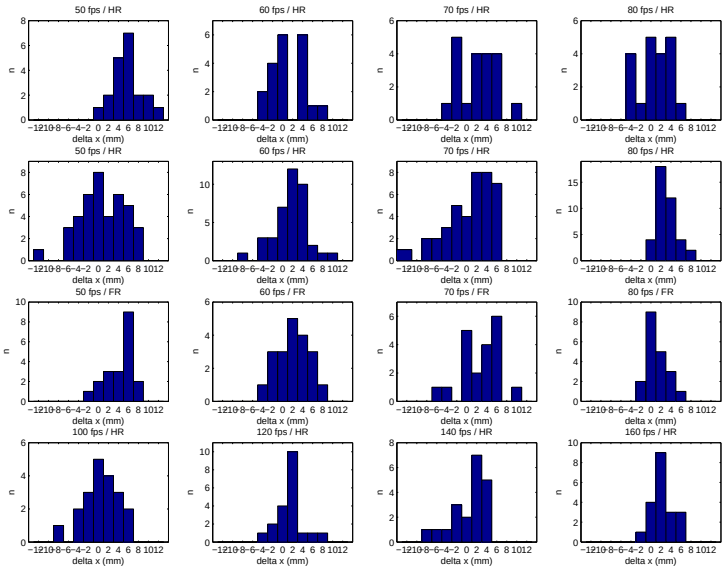

Figure 10: Histograms of horizontal deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

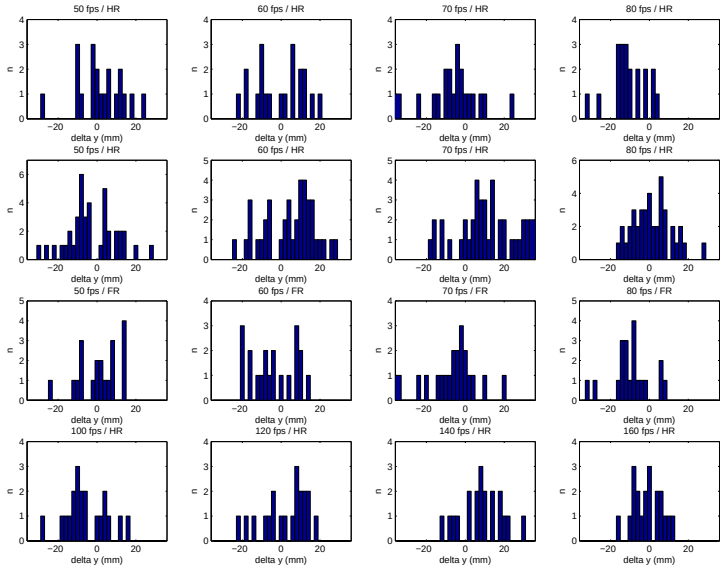

Figure 11: Histograms of vertical deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

the polynomial model.

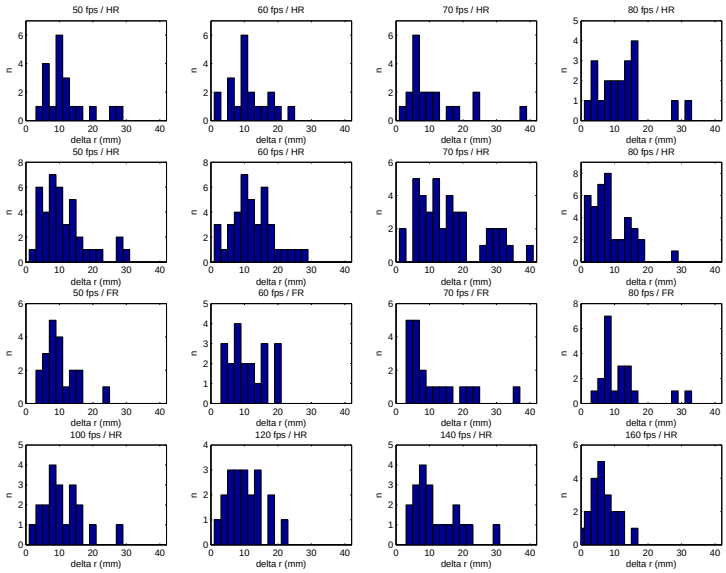

Figure 12: Histograms of overall deviation; 1^{st} row simulated halve resolution, 2^{nd} row halve resolution, 3^{rd} row full resolution and 4^{th} row halve resolution

		hor. deviation			vert. deviation			overall deviation	
fps	n	$\bar{\Delta x}$	$\sigma_{\Delta x}$	99% EB	$\bar{\Delta y}$	$\sigma_{\Delta y}$	99% EB	$\bar{r}$	99% EB
Full resolution									
50	20	4.28	2.65	[-2.53 11.10]	1.18	10.04	[-24.69 27.04]	7.99	[0.80 25.99]
60	20	2.00	3.01	[-5.74 9.75]	-3.93	10.98	[-32.21 24.35]	8.63	[0.86 28.10]
70	20	2.68	3.59	[-6.57 11.93]	-5.46	11.83	[-35.93 25.01]	9.74	[0.98 31.71]
80	20	1.13	2.07	[-4.20 6.46]	-9.71	9.24	[-33.52 14.10]	9.63	[0.96 31.33]
Simulated halve resolution									
50	20	5.85	3.35	[-2.78 14.47]	4.52	11.24	[-24.42 33.47]	9.80	[0.98 31.91]
60	20	1.11	4.05	[-9.32 11.54]	-0.47	11.41	[-29.86 28.92]	8.60	[0.86 28.00]
70	20	2.13	3.75	[-7.53 11.79]	-5.47	12.03	[-36.46 25.52]	9.83	[0.98 32.00]
80	20	0.02	3.26	[-8.39 8.43]	-11.50	8.78	[-34.12 11.11]	10.49	[1.05 34.15]
Halve resolution									
50	40	0.55	4.27	[-10.45 11.56]	-2.64	12.11	[-33.84 28.56]	9.28	[0.93 30.21]
60	40	1.80	3.26	[-6.59 10.20]	4.62	12.56	[-27.73 36.97]	9.82	[0.98 31.97]
70	40	0.50	4.72	[-11.65 12.64]	10.82	14.44	[-26.37 48.01]	13.19	[1.32 42.93]
80	40	3.07	1.92	[-1.88 8.02]	1.39	9.58	[-23.29 26.07]	7.31	[0.73 23.79]
100	20	0.56	3.17	[-7.62 8.73]	-5.79	10.44	[-32.69 21.12]	8.75	[0.88 28.47]
120	20	1.28	2.45	[-5.02 7.58]	2.25	11.01	[-26.10 30.61]	8.18	[0.82 26.63]
140	20	0.46	3.33	[-8.13 9.05]	8.34	10.48	[-18.65 35.32]	9.76	[0.98 31.78]
160	20	2.34	1.86	[-2.45 7.14]	-0.84	7.11	[-19.15 17.47]	5.48	[0.55 17.85]
overall	640	1.87	3.30	[-6.64 10.38]	-0.48	11.06	[-28.96 28.01]	9.37	[0.94 30.49]

Table 3: Prediction results based on the separated model, all numbers in millimeters

Literature

[AEN06] Ahmadi, M. ; Eghtesad, M. ; Necsulescu, D.S.: Design and Control of Three-Robot Visual Servoing Mechatronics System. In: *2006 IEEE International Conference on Mechatronics* (2006) 4, 5, 7

[BFK08] Barteit, D. ; Frank, H. ; Kupzog, F.: Accurate prediction of interception positions for catching thrown objects in production systems. In: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* (2008) 3, 13, 14, 16, 52, 67

[BJE+08] Benezeth, Y. ; Jodoin, P.M. ; Emile, B. ; Laurent, H. ; Rosenberger, C.: Review and evaluation of commonly-implemented background subtraction algorithms. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (2008), Dec., S. 1–4. – ISSN 1051–4651 6

[DBP09] D. Barteit, F. K. ; Pongratz, M.: Measuring the Intersection of a Trajectory with a Vertical Plane. In: *INDIN09* (2009) 2, 14, 15, 53, 54

[FBWW+07] Frank, H. ; Barteit, D. ; Wellerdick-Wojtasik, N. ; Frank, T. ; Novak, G. ; Mahlknecht, S.: Autonomous Mechanical Controlled Grippers for Capturing Flying Objects. In: *Industrial Informatics, 2007 5th IEEE International Conference on* (2007) 2, 3, 14

[GW07] Gonzalez, Rafael C. ; Woods, Richard E.: *Digital Image Processing*. 3rd Edition. Prentice Hall, August 2007. – ISBN 013168728X 5, 7, 8

[HS91] Hove, Barbara ; Slotine, Jean-Jacques E.: Experiments in Robotic Catching. In: *American Control Conference, 1991* (1991) 5

[HV62] Hough V, Paul C. *Method and means for recognizing complex patterns*. December 1962 8

[KSOI99] Kajikawa, S. ; Saito, M. ; Ohba, K. ; Inooka, H.: Analysis of human arm movement for catching a moving object. In: *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on* (1999) 5

[LS86] Lancaster, Peter ; Salkauskas, Kestutis: *Curve and Surface Fitting an Introduction*. 2nd Edition. Academic Press Ltd, 1986. – ISBN 0124360602 13

[MHM04] Mori, R. ; Hashimoto, K. ; Miyazaki, F.: Tracking and catching of 3D flying target based on GAG strategy. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* (2004) 5

[MM02a] Mori, R. ; Miyazaki, F.: Examination of human ball catching strategy through autonomous mobile robot. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on* (2002) 5

[MM02b] Mori, R. ; Miyazaki, F.: GAG (gaining angle of gaze) strategy for ball tracking and catching task. In: *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on* (2002) 5

[MMN06] Migliore, Davide A. ; Matteucci, Matteo ; Naccari, Matteo: A revaluation of frame difference in fast and robust motion detection. In: *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. New York, NY, USA : ACM, 2006. – ISBN 1–59593–496–0, S. 215–218 6

[NI05] Namiki, A. ; Ishikawa, M.: The Analysis of High-speed Catching with a Multifingered Robot Hand. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005) 3, 5

[Pic04] Piccardi, M.: Background subtraction techniques: a review. In: *Systems, Man and Cybernetics, 2004 IEEE International Conference on* 4 (2004), Oct., S. 3099–3104 vol.4. – ISSN 1062–922X 6

[RA02] Riley, Marcia ; Atkeson, Christopher G.: Robot Catching: Towards Engaging Human-Humanoid Interaction. In: *Auton. Robots* (2002) 5

[SHB07] Sonka, Milan ; Hlavac, Vaclav ; Boyle, Roger: *Image Processing, Analysis, and Machine Vision*. 3rd Edition. Cengage-Engineering, March 2007. – ISBN 049508252X 7, 8

[SPV05] Scaramuzza, D. ; Pagnottelli, S. ; Valigi, P.: Ball Detection and Predictive Ball Following Based on a Stereoscopic Vision System. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), April, S. 1561–1566 9, 34

[SSL01] Slabaugh, Greg ; Schafer, Ron ; Livingston, Mark: Optimal Ray Intersection For Computing 3D Points From N-View Correspondences. (2001) 18

[TKBM99] Toyama, K. ; Krumm, J. ; Brumitt, B. ; Meyers, B.: Wallflower: principles and practice of background maintenance. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* 1 (1999), S. 255–261 vol.1 5, 6

Internet References

[1] 3M, editor. *3M MicroTouch DST Touch System*, June 2009. solutions.3m.com/wps/portal/3M/en_US/3MTouchSystems/TS/Solutions/TouchScreens/DispersiveSignal/?WT.mc_id=www.dsttouch.com.

[2] J.-Y. Bouguet, editor. *Camera Calibration Toolbox for Matlab*, June 2009. www.vision.caltech.edu/bouguetj/calib_doc/.

[3] I. I. D. S. GmbH, editor. *Home - Image processing, industrial cameras, uEye, GigE, USB - uEye Industrie Kameras GigE USB - IDS Imaging Development Systems*, September 2009. http://www.ids-imaging.com/.

[4] Hawk-Eye Innovations Ltd. *Hawk-Eye Innovation*, April 2009. www.hawkeyeinnovations.co.uk/.

[5] R. Nennstiel, editor. *How do bullets fly?*, June 2009. www.nennstiel-ruprecht.de/bullfly/index.htm.

[6] S. Nielsen, editor. *The Personal Computer - VGA Dummy*, June 2009. soerennielsen.dk/mod/VGAdummy/index_en.php.

[7] M. Papas, editor. *Revolutionary Tennis Federer Vision Technique*, April 2009. www.revolutionarytennis.com/federervisiontechnique.html.

[8] I. The MathWorks, editor. *Fitting Data (Curve Fitting Toolbar)*, August 2009. http://www.mathworks.com/access/helpdesk_r13/help/toolbox/curvefit/ch_fitt5.html.

[9] Wikimedia Foundation, Inc. *Henry Ford*, April 2009. en.wikipedia.org/wiki/Henry_Ford.

[10] Wikimedia Foundation, Inc. *Oliver Evans*, April 2009. en.wikipedia.org/wiki/Oliver_Evans.

[11] R. Wilson and J. Bercovitz, editors. *RMM 3D Encyclopedia - Keystone / Convergence error*, April 2009. www.rmm3d.com/3d.encyclopedia/keystone/keystone.html.