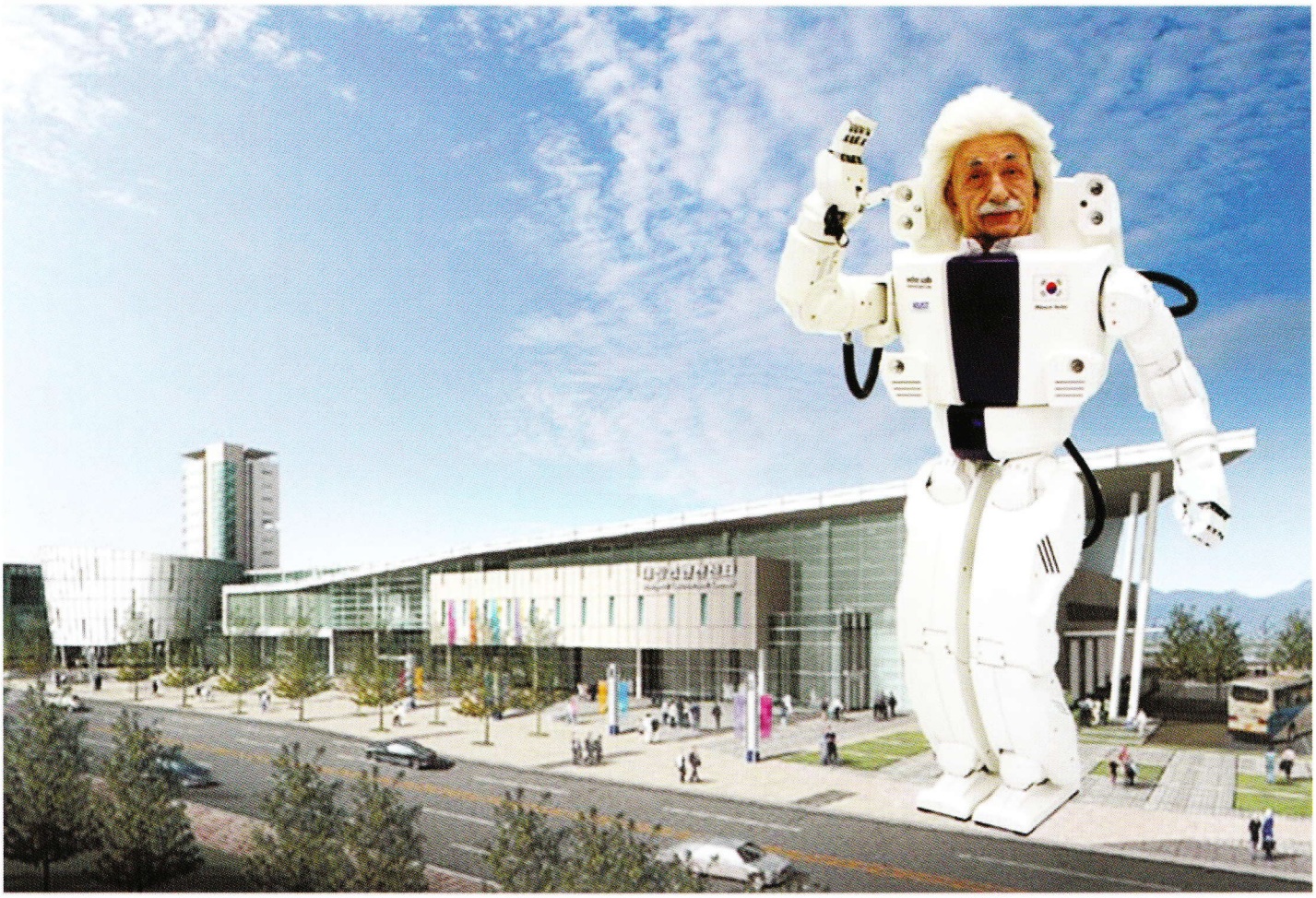
Conference Digest

# IEEE INDIN 2008

## 6th IEEE International Conference on Industrial Informatics

July 13-16, 2008
Daejeon Convention Center
Daejeon
Korea



Sponsored by

IEEE ies oneida KAIST RMIT University ICROS Institute of Control, Robotics and Systems

Liu, Xing-peng ····· TT-SCT3-2
Liu, Yabin ····· SS-IPSC2-1
Liu, YanPeng ····· TT-SCT6-3
Liu, Yao ····· SS-LIE-6
Liu, Yao ····· SS-LIE-5
Liu, Yi-Hua ····· TT-SCT2-5
Liu, Yi-Hua ····· TT-IIA2-5
Liu, Yubo ····· SS-LIE-3
Lo, Jung-Hua ····· TT-ISST2-4
Lobov, Andrei ····· TT-SOA1-5
Lobov, Andrei ····· TT-SOA2-4
Lobov, Andrei ····· TT-DENC4-2
Low, Joyce M.W. ····· TT-BNA1-1
Low, Joyce M.W. ····· TT-SCML1-6
Low, Malcolm Yoke Hean ····· TT-CCI1-3
Low, Malcolm Yoke Hean ····· TT-SCML3-1
Lu, Xu ····· TT-IIA4-4
Luis, Zarate ····· TT-EPT1-5
Luo, Zhimeng ····· TT-DENC1-2
Lv, Tao ····· SS-TCL-5
Lye, Kong-wei ····· TT-SCML1-4

## M

Ma, Bin ····· TT-SCML1-1
Ma, Bin ····· TT-SCML3-4
Ma, Tianyun ····· SS-LIE-7
Ma, Zi ····· TT-IIA8-3
Machado, Vinicius ····· TT-FA1-2
Madani, Sajjad ····· TT-BNA2-2
Mahlknecht, Stefan ····· TT-BNA2-2
Malik, Najmus Saqib ····· TT-ISST3-1
Maria, Ana ····· TT-IIA2-1
Martel, Allan ····· SS-IEC-2
Matsumoto, Shinichi ····· SS-PAC-3
Matsushiba, Yoshinao ····· SS-ITE-2
McFarlane, Duncan ····· TT-DENC4-3
Medeiros,Juliana ····· TT-FA1-2
Mendes, J. Marco ····· TT-SOA1-1
Mendes, J. Marco ····· TT-SOA1-2
Meng, Fanyi ····· SS-RF1-5
Merdan, Munir ····· TT-CCI3-5
Min, Huasong ····· TT-DENC2-6
Min, Seung Hwan ····· TT-BNA2-5
Min, Zhang ····· TT-IIA6-1
Mitsch, Stefan ····· TT-EPT1-4
Miyazawa, Kazunori ····· TT-ISST2-1
Mohamed,Shady Mohamed Korany ····· TT-SCT3-5
Monfared,Radmehr ····· TT-FA2-4
Moon, Tae Yoon ····· TT-DENC2-2
Moon, Tae-Yoon ····· TT-DENC3-3
Moon, Tae-Youn ····· TT-DENC2-1
Moon, Tae-Youn ····· TT-IIA3-5
Moon, Tae-Youn ····· TT-BNA1-5
Moon, Yongseon ····· SS-IR1-4
Morais, Antônio ····· TT-BNA2-1
Morais, Antônio ····· TT-IIA2-1
Morimoto, Sigeaki ····· TT-SCT6-2
Mous, K. ····· TT-SCML1-3
Mrazova, Iveta ····· SS-PCCT1-4
Mvungi, Nerey ····· TT-MR2-1

## N

Nakamura, M. ····· TT-IIA8-4
Nagashima, Akira ····· TT-DENC1-4
Nahavandi, Saeid ····· SS-IPSC1-1
Nahavandi, Saeid ····· TT-SCT3-5
Neto, Lauro V.B.M. ····· TT-IIA5-5
Ng, Vincent ····· TT-IIA6-4
Ng, Wee Keong ····· TT-ENIE2-2

Kim, Jeong S. ····· TT-DENC1-1
Kim, Jin Ho ····· TT-DENC2-2
Kim, Jin-Ho ····· TT-BNA1-5
Kim, Jin-Ho ····· TT-DENC2-1
Kim, Jin-Ho ····· TT-DENC3-3
Kim, Jin-Ho ····· TT-IIA3-5
Kim, Ki Hoon ····· TT-SCT3-3
Kim, Seung-Han ····· TT-BNA1-5
Kim, Seung-Han ····· TT-DENC3-3
Kim, Seung-Han ····· TT-IIA3-5
Kim, Sung Hyun ····· SS-LIE-1
Kim, Sung-Jo ····· TT-MR1-3
Kim, Taeho ····· TT-HCM3-3
Kim, Woo-Je ····· TT-SCML1-2
Kirkham, Tom ····· TT-FA2-4
Kitoko, LeBel ····· SS-DES-3
Klein, Lothar ····· SS-DES-3
Ko, Nak Yong ····· SS-IR1-4
Koo, Benjamin H.-Y. ····· SS-LIE-4
Koppensteiner, Gottfried····· TT-CCI3-5
Kuikka, Seppo ····· TT-SOA1-3
Kukacka, Marek ····· SS-PCCT1-4
Kupzog, Friederich ····· SS-ITE-4
Kupzog, Friederich ····· TT-MR1-2
Kupzog, Friedrich ····· TT-ISST3-1
Kurschl, Werner ····· TT-EPT1-4
Kvas, Gernot ····· TT-HCM1-3
Kwon, Key Ho ····· TT-DENC2-1
Kwon, Key Ho ····· TT-DENC2-2
Kwon, Key Ho ····· TT-DENC3-3
Kwon, Key-Ho ····· TT-BNA1-5
Kwon, Key-Ho ····· TT-IIA3-5
Kwon, Keyho ····· TT-HCM2-2
Kwon, Ki-Jeong ····· TT-DENC4-1
Kwon, Soon-Kwang ····· SS-IRA-4

## L

La, Mai Thanh ····· TT-FA1-4
Lai, H.F. ····· TT-SME-2
Lang, Roland ····· TT-CCI4-2
Lastra, Jose L. Martinez ····· TT-SOA1-5
Lastra, Jose L. Martinez ····· TT-SOA2-2
Lastra, Jose L. Martinez ····· TT-ENIE1-2
Lastra, Jose Luis Martinez ····· TT-SOA2-4
Lastra, Jose Luis Martinez ····· TT-DENC4-2
Le, Hieu Tue ····· TT-FA1-3
Le, Quy Ngoc ····· TT-FA1-4
Lee, Chang Hoon ····· TT-SCT3-6
Lee, Cheongjae ····· TT-HCM2-3
Lee, Choon-Young ····· TT-EPT1-2
Lee, Choon-Young ····· TT-HCM1-2
Lee, Dong-Hyun ····· SS-IR2-1
Lee, E.W. ····· TT-SCML1-3
Lee, E.W. ····· TT-ENIE2-5
Lee, Eng Wah ····· TT-SOA2-1
Lee, Fu-Shin ····· TT-DENC1-6
Lee, Fu-Shin ····· TT-IIA3-2
Lee, Gary Geunbae ····· TT-HCM2-3
Lee, Heung-ho ····· TT-SCT4-4
Lee, Heung-ho ····· TT-SCT5-1
Lee, Hoonjae ····· TT-ISST3-2
Lee, Hou-Tsan ····· TT-SCT1-2
Lee, Hyo Jik ····· SS-LIE-1
Lee, Hyo-sung ····· TT-SCT4-4
Lee, JangMyung ····· TT-MR2-3
Lee, Jiann-Der ····· TT-SCT4-2
Lee, Jongchul ····· SS-RF1-3
Lee, Joon Whoan ····· TT-IIA6-2
Lee, Joon-Woo ····· SS-IR2-3
Lee, Joon-Yong ····· SS-ISS-3
Lee, JuJang ····· TT-CCI3-3

The IEEE International Conference on
Industrial Informatics (INDIN 2008)
DCC, Daejeon, Korea July 13-16, 2008

# Accurate prediction of interception positions for catching thrown objects in production systems

Dennis Barteit and Heinz Frank
Reinhold Würth University
Department of Electrical Engineering
Daimlerstr. 35, D-74653 Künzelsau
{barteit,frank}@hs-heilbronn.de

Friederich Kupzog
Vienna University of Technology
Institute of Computer Technology
Gußhausstraße 27-29, A-1040 Wien
kupzog@ict.tuwien.ac.at

*Abstract*— **This work aims to optimize transportation processes in production by throwing objects between the working stations instead of transporting them on conveyed belts. One aspect of this new approach is the accurate prediction of interception positions of thrown objects with the catching robot. In a first step, this work analyzes the variation of the flight trajectories to specify the requirements for an appropriate catching-device. Several objects were thrown by a throwing-device and the range of their passages through a gate similar to the portal of the proposed gantry robot for object catching were measured. The second and main part of this work deals with calculating the object's position in flight and accurately predicting a point of interception with the catching robot. The proposed prediction model for interception positions bases on two inputs: the starting angles extracted from observations of a single camera and the speed of the object measured by two light barriers. The model is precise enough in respect to the size of the gripper of the catching robot.**

## I. Introduction

In contrast to other robot-catching applications, the throwing and catching of objects in industrial production systems is a well-bounded problem, which potentially can be solved by an effective and cost-saving solution [1]. This approach can reduce duration of transportation and offers a flexible design of production processes due to an easy change of working stations' sequences. As shown in Fig. 1, we propose to catch thrown objects by a gantry robot, placed above a working station. The distance between two machines and therefore the throw distance is assumed to be 3 meters. The assumed object's velocity amounts to about 10 m/s. Thus, about 300 ms flight time remain for the two tasks of predicting the object's final position when it crosses the portal of the gantry robot and to move the robot accordingly. Different aspects of this transportation approach have to be optimized in order to meet the timing restrictions. These sub-functions are: throwing of the object, detecting of the object's trajectory, tracking of the catching-device and catching of the object [1].

This paper addresses the detection of the object's trajectory. The most important advantage of the industrial production setting is that one exactly knows the launching position and time of the thrown object. Using the visual information from a single camera, an approximation of the thrown object's spatial position can be obtained. This /calculation will be described further in Section IV-C. Determining spatial trajectories of thrown objects with monocular machine vision is possible, but the question is if this approach is accurate enough for the targeted application. In human interceptive actions, usually using binocular visual information, catching performance is worse under monocular viewing but still possible (see [2]). In this paper the accuracy of predicted interception position with monocular vision is measured and was found to be sufficient for gripper we developed [3] to grab the flying object. Performance of stereo vision seems to be better (in humans) and is already used in several other machine catching applications.

According to [4] Hong and Slotine were the first to succeed with catching a flying ball using a robot and visual feedback with stereo vision in 1995 [5]. Further publications reported to manage catching a flying objects: Frese et al. in 2001 [6] as well as Namiki et al. in 2003 [7].

In [8] a humanoid robot is presented which has learned to juggle with balls. The sensor-systems to get information about flight trajectories and hence be able to track the objects is based on active stereo vision. There are also some examples for monocular vision in interceptive actions: Acosta and colleagues presented a PC-based, low-cost ping-pong robot in 2003. To get spatial information about the ping-pong ball they used 2D-positions of the ball and its shadow [9]. Mori et al. developed a strategy called Gaining Angle of Gaze (GAG) to catch a thrown object tracked by a single camera mounted on the robot manipulator [10]. In a task where the application conditions allow simplifying assumptions and therefore offer the possibility to use only one camera, some advantages of a monocular over a binocular vision system can be exploited. A simple system like a single fixed camera is likely to be more robust than two actively orientated cameras with moving parts and besides that causes fewer costs. It is also needs less computational power.

## II. Trajectory variation

To specify requirements for the gantry-robot we first studied the variation of several objects' flight trajectories. We measured the position range of passages through a gate while performing 100 throws. This range is defined as tuple $r$ which describes the maximal distance between passage positions of a test series in horizontal and vertical direction respectively:

978-1-4244-2171-8/08/$25.00 ©2008 IEEE

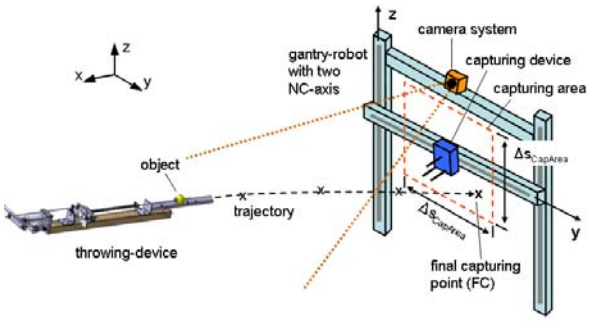



Fig. 1. Structuring of the catching task.

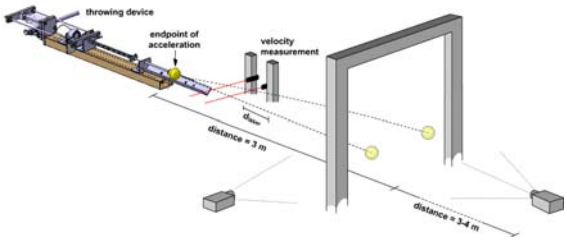



Fig. 2. Range measurement set up

$$r = (|\max(Y) - \min(Y)|, |\max(Z) - \min(Z)|) \quad (1)$$

Where Y marks horizontal positions in the robot's plane and Z marks vertical positions. If $r$ would describe an area within which the gripper can catch an object, it would be possible to center a gripper at the center of the passage positions set. The gripper would not have to be moved. However, the results of a range analysis for diverse throwing devices and several different thrown objects show ranges between 100 mm and up to 400 mm. The tolerance allowed by the gripper itself is only 25 mm. Thus, the flight of the object to be catched has to be observed and a suitable flight trajectory model is required to predict the accurate passage position, so that the gripper can be moved to this position.

### A. set up

Two throwing devices were used. Both utilize spring energy to accelerate the object to be thrown. Throwing device A operates with a torsion spring, whereas throwing device B linearly accelerates the object along a guide rail. Here, a spring is stressed along the acceleration axle. An solenoid keeps the spring's tensioned until the current supply is interrupted. Then, a ram accelerates the object.

Five objects of different shape, volume, mass and material were thrown (see table I). From these initial experiments, it was decided to focus on catching a tennis ball subsequently [1]. For the trajectory range experiment, a gate similar to the gantry robot's portal is placed 3 m away from the acceleration endpoint and defines the measurement plane. Cameras used are standard industrial USB cameras (IDS uEye UI-1220-MM-GL: 1/3" CMOS monochrome sensor, 752x480 pixel, 87 fps, Global Shutter). One camera (camera A) is aligned frontally against the flight direction in a distance of about 3-4 m to the measurement plane. Another camera (B) laterally observes the measurement plane and triggers camera A to shoot an image. It observes only a small part of the scenery and therefore can operate with a relative high temporal resolution of about 215 fps, resulting in a inter-frame time of 5 ms. Given the object's launching velocity of about 10 m/s, the object's position inside the two-dimensional measurement plane can vary over at most 50 mm during this time.

### B. results

Ranges of passage positions can be taken from Table I. In Table I $size_r$ describes the radius of a sphere and cylinder's face side respectively. $size_l$ marks the cylinder's and the cube's edge length respectively.
It is obvious that objects which are small in mass and size have the smallest ranges in passage positions. This is due to the nature of turbulences affecting the flight trajectory. Turbulences occur at high Reynold numbers and become more chaotic when Reynold number increase i.e. for instance when a larger object is thrown. Considering the tennis ball ranges in vertical and horizontal direction respectively for both throwing-devices, it can be seen that the torsion spring of throwing device A produces a wider horizontal distribution that device B. Despite the ram of throwing device B often does not hit the ball centrally, the horizontal distribution is minimal due to the the guide rail.
The main result of the experiment is that the developed gantry-robot is able to handle the observed ranges of passage positions in the given time. The robot accelerates with $25m/s^2$ in vertical and $15m/s^2$ in horizontal direction. Vertical high-speed is about $4m/s$ and horizontal $2.4m/s$ respectively. The size of the robot's working-area is $400mm \times 400mm$. Starting from the center of the area the robot covers a distance of $200mm$ within $180ms$.

## III. Spatial Measuring using Monocular Vision

While other researchers use stereoscopic vision systems for spatial information of an object [5], [6], [8], [11], [12], in our approach one camera is sufficient. In order to obtain spatial information about the thrown object, we use the flight distance which is approximated using launching velocity and flight duration. We assume that the ball position lies on a straight line $g$ that goes through the camera's focal point and the position of the ball's projection on the camera sensor (see Fig. 3). Then the ball's position can be calculated as a point of intersection of this straight line and plane $E$. $E$ is determined by a normal vector equal to the direction in which the velocity is measured and a displacement vector consisting of the launching position and the current flight distance.

For velocity measurement, two light barriers placed in 100 mm distance were used (see Fig. 2). The distance between

| throwing device | object | size (mm) | mass (g) | mean position (mm) | $range_y$ (mm) | $range_z$ (mm) |
|---|---|---|---|---|---|---|
| A | tennis ball | r=65 | 56 | (29.08 \| 30.42) | 7.26 | 30.41 |
| B | tennis ball | r=65 | 56 | (29.08 \| 30.42) | 23.59 | 18.09 |
| B | big sphere | r=59 | 80 | (22.81 \| 39.81) | 24.50 | 26.34 |
| B | small sphere | r=35 | 20 | (26.13 \| 44.01) | 8.76 | 12.60 |
| B | cube | l=28 | 16 | (30.71 \| 32.45) | 7.39 | 21.16 |
| B | cylinder | l=88, r=30 | 49 | (28.99 \| 33.34) | 14.77 | 24.49 |

TABLE I

VARIATION OF DIFFERENT OBJECTS.

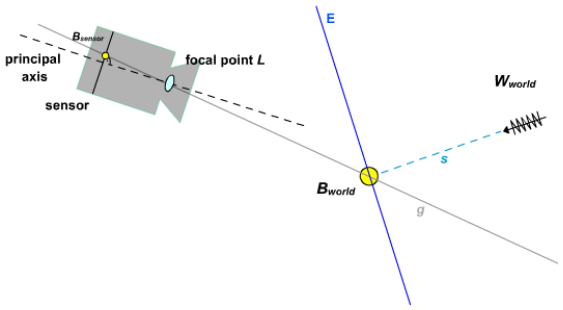



Fig. 3. Ascertainment of the ball's spatial position using the intersection of plane $E$ and line $g$.

endpoint of acceleration and first light barrier is either 100 mm or 200 mm depending on throwing device. Let $\Delta t_0$ be the time difference with that the light barriers answer to an interruption. Thus the launching velocity amounts to:

$$v(0) = 0.1/(\Delta t_0) m/s \tag{2}$$

In the following work, only tennis balls as thrown object are considered. Additional to the velocity measurement by light barriers we use one camera looking toward the flying ball (see Section II-A). An algorithm based on this simple measurement method is developed that performs object tracking and estimates the interception position.

## IV. TRACKING ALGORITHM

First the calibration of the camera in respect to world coordinate system as well as the robot's coordinates is needed. The launching velocity is measured as described above. During the flight the object has to be detected in the current camera image. Then the spatial positioning method can be applied. With the information about the object spatial position the flight's trajectory can be approximated and based on this a possible interception point is predicted.

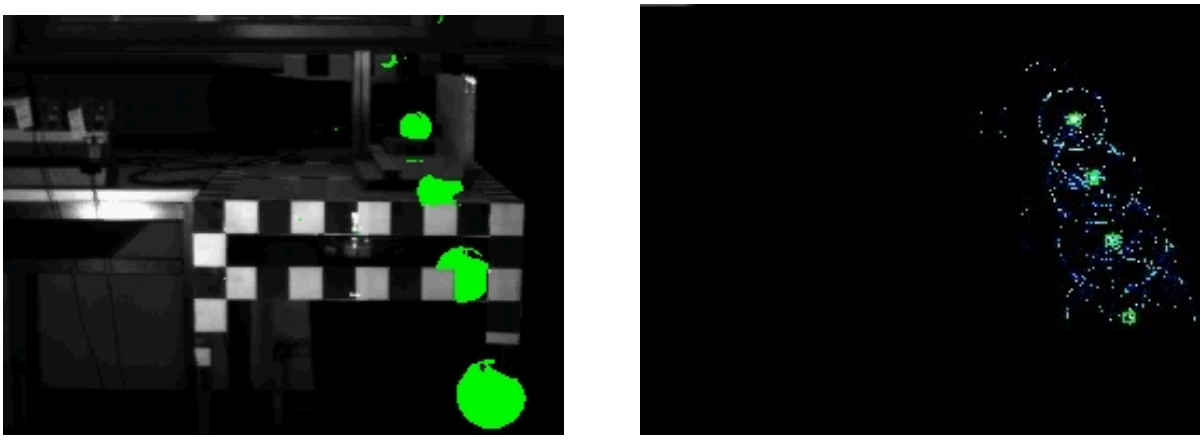
### A. Calibration

A simple 6-parameter camera model is used to simulate the translation (vector $\vec{l}$) and rotation $(\alpha, \beta, \gamma)$ of the camera's optical axis in respect to a world coordinate system. For reasons of simplicity, the robot is also described in world coordinates. Calibration is achieved by calibration methods of the openCV open-source C++ library.

### B. Object detection

*1) Motion detection:* The Hough transformation is performed on motion information which is the difference of a current image and a previously taken background image: $I_{motion} = b(I_t - B)$ with $b(\cdot)$ is a binary function using a threshold. This simple background subtraction (BS) prevents problems appearing in image differencing like ghost or the apparent problem and is sufficient for most of the throws. Main problems of BS like changing backgrounds or light fluctuations usually don't appear during the short flight.

*2) Circle detection:* Throwing a tennis ball simplifies the object detection to a circle-searching task. Circle detection based on Hough transformation is widely used, fast and robust [12], [13], [14]. Several real-time applications utilize this method for detection of spherical objects. Our implementation of the Hough transformation profits from a previous prediction of the ball's spatial position and thus a prediction of its radius in the image. Hence, only two parameters have to be found: x- and y-Position of the center without considering a third searching space dimension for the radius. Additionally, the gradient of the motion image is used. This is also an often considered information. The gradient at a particular edge-point of a circle shows toward the center. One simply has to follow this direction for a distance equal to the predicted radius of the projection of the sphere in the current image. Hence, in the accumulator it only has to be voted for one potential circle center per considered edge-point. Edge detection of the motion information is performed using a smoothed Laplace operator, followed by binarization. On the edge image the Hough transformation for circles can be applied. Further calculation is performed on a set $P_R$ of randomly chosen edge points (i.e. 20 percent of all points). At each point $p \in P_R$ a Sobel operator applied to the motion image results in a gradient vector at this position. Along this gradient direction, in a distance of the calculated radius at time $t$, it is counted as a possible circle center in the accumulator. The real radius of the tennis ball (32.5 mm) is transformed into its size in the image. This depends on the distance between ball and camera and the camera's focal length. Therefore, the ball's currently predicted position in space must be considered. The maximum of the accumulator entries, which are box filtered before, should mark the circle's center.



(a) Detected motion (green) on four different times.

(b) Accumulator with votations on four different times. Green circle marks detected middle.

Fig. 4. Example of object detection

### C. Spatial measuring

The object velocity $v_x$ is measured along the X-axis. It decreases over time by

$$\dot{v}_x = -\frac{k \cdot v_x^2}{m} \tag{3}$$

with

$$k = \frac{c_w \cdot \rho \cdot A}{2} \tag{4}$$

This quadratic differential equation is numerical approximated using the fourth-order Runge-Kutta method. The flight distance $s$ along the X-axis is then approximated similarly. In Equation 4 factor $k$ subsumes the properties of the aerodynamic resistance. Here $c_w$ is the air drag coefficient of a thrown object, $\rho = 1.294\ kg/m^3$ the air's density at 20°C surrounding temperature and $A$ the front surface size of the object.

Further, information is available about rotation ($\alpha$, $\beta$, $\gamma$) and position ($\vec{l}$) of the camera's principal axis in respect to the world's coordinate system from the camera calibration. Furthermore the object's velocity $v_x(0)$ is measured shortly after the endpoint of acceleration (i.e. in 200 mm distance). The current duration of flight $t$ is received by a high-resolution timer. The camera provides an image of the scenery and thus the object's position at $t_i$. The position is primarily given in a coordinate system of the image (i.e. $B_{image}$) and the sensor respectively (i.e. $B_{sensor}$).

First of all, the transformation of the coordinate systems has to be performed. There are three coordinate systems: image, sensor and world. The world coordinate system is determined by the calibration object. The current positions of throwing and catching devices are given in world coordinates. At first, the image coordinates of the detected circle center have to be transfered to sensor coordinates. Pixel units of the image are converted into Millimeters and the origin of image coordinate system is shifted half the sensor sizes toward the center. Hence positions of the flying object are given by distances to the principal axis on the sensor:

$$\vec{B}_{sensor} = \frac{\vec{B}_{image}}{\kappa} - \vec{C}_s \tag{5}$$

where $\kappa$ is the pixel size in mm and $\vec{C}_s$ the sensor center coordinates (in mm). Given the camera parameters ($\alpha$, $\beta$, $\gamma$) and $\vec{l}$), it is possible to determine the sensor's position in world coordinates. For that, the ball position on the sensor is shifted by vector $\vec{l}$ and subsequently rotated by the inverse of the rotations matrix $A$ (defined by $\alpha$, $\beta$ and $\gamma$):

$$\vec{B}_{CSworld} = A^{-1} \cdot (\vec{B}_{sensor} + (\vec{l} + \begin{pmatrix} 0 \\ 0 \\ f_k \end{pmatrix}) \tag{6}$$

$\vec{B}_{CSworld}$ now represents the position of the detected circle center on the sensor (which is not the ball center) in world coordinates.

After these transformations, $\vec{B}_{world}$ (i.e. the ball's spatial position) can be computed. If not noted differently, all following positions and vectors refer to world coordinates. As declared above, the point of intersection between a plane $E$ and a straight line $g$ has to be calculated. The equation of the straight line is

$$\vec{x} = \vec{o} + \lambda \cdot \vec{d} \tag{7}$$

The position vector $\vec{o}$ is given by $\vec{B}_{CSworld}$ and the direction vector $\vec{d}$ is received by

$$\vec{d} = \vec{L} - \vec{B}_{CSworld} \tag{8}$$

with $\vec{L}$ is the camera's focal point. Plane $E$ is given by its normal vector $\vec{n}$ and a scalar $c$ marking the distance to the origin:

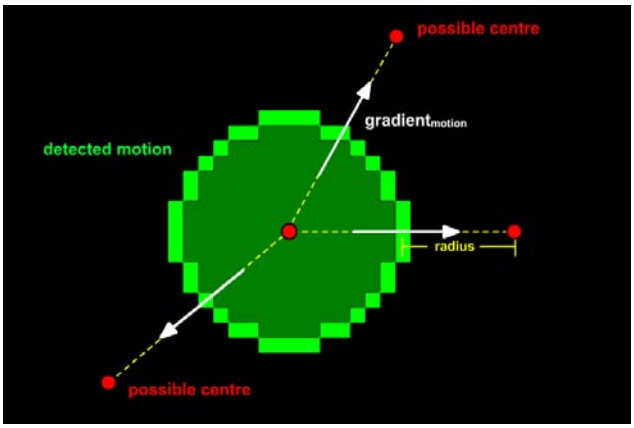



Fig. 5. Principle of voting for a circle center in the accumulator. Along the gradient at an edge point and with distance equal to the ball's transformed radius two possible circle midpoints are counted in the accumulator.

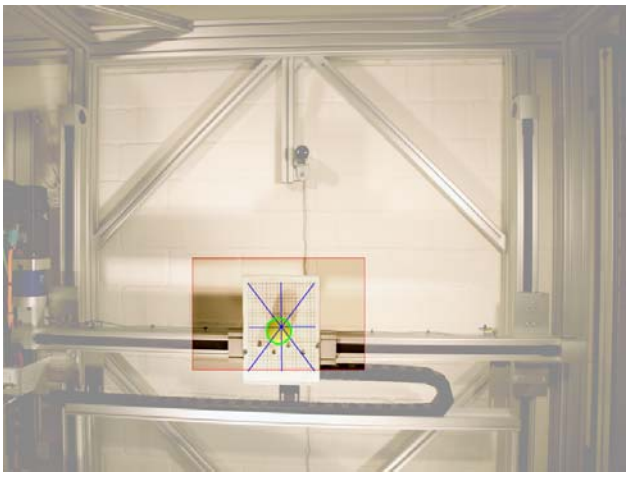


Fig. 6. Picture of the gantry-robot. Red rectangle marks the region which the camera for prediction-analysis observed. Model of plane and tennis ball is sketched blue and red respectively.

$$0 = \vec{n} - \vec{x} - c \tag{9}$$

Setting Equation 7 in Equation 9 results in:

$$0 = n_x a_x + n_y a_y + n_z a_z + \lambda d_x n_x + \lambda d_y n_y + \lambda d_z n_z - c \tag{10}$$

The velocity measurement is performed in the XY-plane and is orientated along the x-axis and so does $\vec{n}$. Therefor the z- and y- part of $\vec{n}$ equals 0 and $n_x = 1$. In Equation 10 this yields to:

$$0 = n_x a_x + \lambda d_x n_x - c \tag{11}$$

By solving to $\lambda$, we obtain

$$\lambda = \frac{-a_x + c}{d_x} \tag{12}$$

$a_x$ equals the x-value of $\vec{B}_{CSworld}$. $d_x$ is defined above as the difference of focal point $\vec{L}$ and $\vec{B}_{CSworld}$. $c$ is calculated by the flight distance $s$ and z-coordinates of launching point $\vec{W}_{world}$.

Using $\lambda$ in the line equation 7 one gets the point of intersections.

### D. Trajectory prediction

In general, an object with complex geometry behaves non-deterministically when it is thrown. A sufficient prediction of the passage position needs information about the flight. For example spatial positions of thrown objects during flight. Then launching parameters like velocity and launching direction can be calculated. As mentioned above the trajectory projected in the XY-plane is assumed to be straight. Thus the launching angle $\xi$ in this plane can be determined by X- and Y-position of ball and throwing device. For prediction the numerical calculation has to be continued until $s$ has passed the distance between launching position and gripper (at time $t_g$). Vertical intersection position can approximated using Equation 13. Vertical acceleration is given by

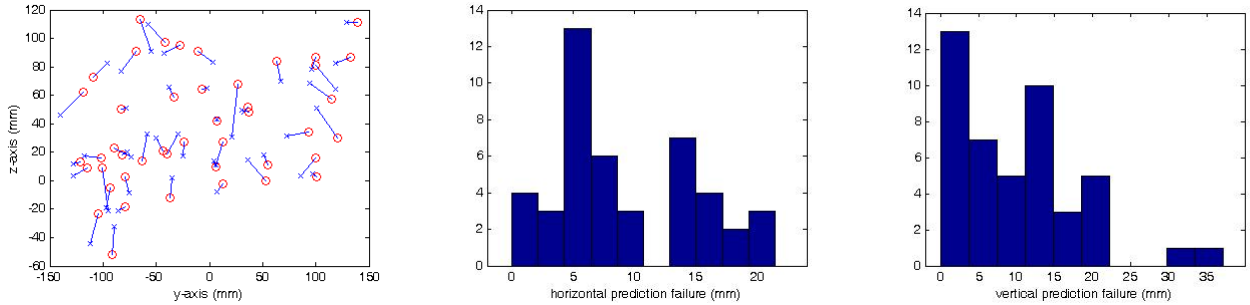
$$\dot{v}_z = r \cdot \frac{k \cdot v_z^2}{m} - g \tag{13}$$

for upward and downward motion respectively. Factor $r$ is set -1 until the reversal point of the flight and 1 afterward. $g$ is the acceleration of gravity. The initial value for vertical velocity in the prediction is the gradient between the last two measured vertical positions. Numerical computing of the vertical position is done until time $t_g$ when the ball arrived at the gripper.

## V. Results

The presented algorithm was tested in an installation consisting of a 2-DoF gantry-robot and throwing-device A, i.e. acceleration using a torsion spring. Calculations were done on a laptop PC. Three different tennis balls were thrown. Instead of a gripper, a disc was mounted in the robot arm and moved by it to the predicted interception points. A second camera observed this disc for analysis of the prediction failure. After 160 ms of flight, prediction is finished and a movement command is send to the robot.

Pictures of the interception are taken by the second camera. On these pictures, catching points were manually measured. This method obviously produces some failures due to temporal and spatial resolution of the camera as well as perspective distortion. While spatial resolution of the camera is maximal 752x480 pixel (pixel-size is 0.006 mm) with reduced area of interest there is a time span of 10 ms between two pictures of a sequence. In most cases the tennis ball remains at the plane long enough to get a picture from this position. To avoid measurement failures based on perspective distortion, camera calibration was performed. For calibration, real world points marked by robot's traverse distances were associated with the identified centers of the plane in the interception




(a) Differences of predicted passage position (red circle), i.e. where the robot is send to, and actual position of interception (blue cross)

(b) Histogram of horizontal differences between predicted and actual position

(c) Histogram of vertical differences between predicted and actual position

Fig. 7.

images. Hence, the position and orientation of the camera have been verified. Subsequently, real world positions of the tennis ball were calculated using the method described above.

Measured ranges of passage positions confirm results presented in section II-B. Fig. 7(a) shows actual passage positions through the gate of the gantry robot (blue crosses) and prediction of interception point (red circles). Horizontal range is about 260 mm and vertical is about 165 mm respectively.

In two-third of all measured throws (29 out of 45) the predicted position of interception was within a horizontal range of 25 mm (i.e. less 12.5 mm failure of predicted interception position). Less than 12.5 mm vertical failure was achieved in 28 throws. In less than 10 percent of measured throws the predicted interception position missed the ball more than 20 mm. Histograms in Fig. 7(b) and Fig. 7(c) show horizontal and vertical differences between predicted and actual positions.

## VI. Conclusion

In this paper an approach for accurate prediction of interception positions of thrown objects was presented. As this is a transportation solution in industrial production systems, throwing as well as catching is accomplished by robots. Due to the given launching position and a initial velocity measurement a monocular vision-system is sufficient to calculate spatial positions of objects during flight. The accuracy of this approach, i.e. the difference of predicted and actual interception positions, was measured. In about two-third of the measured throws this difference amounts less than 15 mm. The implementation was successfully tested within an existing catching robot-system. This is a successful first step on the route toward fully automated throwing and catching of objects as a novel approach for transportation in industrial production systems.

## Acknowledgment

The authors wish to thank the Foundation for the Promotion of the Reinhold-Würth-University of the Heilbronn-University in Künzelsau for their support of this work.

## References

[1] H. Frank, N. Wellerdick-Wojtasik, B. Hagebeuker, G. Novak and S. Mahlknecht (2006), *Throwing Objects - A bio-inspired Approach for the Transportation of Parts*, IEEE Int. Conf. on Robotics and Biomimetics, December 2006, Kumming, China, pp. 91-96.

[2] L. I. N. Mazyn and M. Lenoir and G. Montagne and G. J. P. Savelsbergh (2004): *The contribution of stereo vision to one-handed catching*; Exp. Brain Res., 157: 383-390.

[3] H. Frank, D. Barteit, N. Wellerdick-Wojtasik, T. Frank, G. Novak and S. Mahlknecht (2007), *Autonomous Mechanical Controlled Grippers for Capturing Flying Objects*, In Proceedings of the 2006 IEEE International Conference on Industrial Informatics, July 3-26, Vienna, Austria, pp. 431-436.

[4] A. Namiki and M. Ishikawa (2001): *Sensory-Motor Fusion Architecture Based on High-Speed Sensory Feedback and Its Application to Grasping and Manipulation*; Proceedings of the 32nd ISR(International Symposium in Robotics): 19-21 April 2001

[5] W. Hong and J.E. Slotine (1995): *Experiments in hand-eye coordination using active vision*; Proc. 4th Int. Symp. on Experimental Robotics (ISER'95), 1995

[6] U. Frese and B. Bauml and S. Haidacher and G. Schreiber and I. Schaefer and M. Hähnle and G. Hirzinger (2001): *Off-the-Shelf Vision for a Robotic Ball Catcher*; IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Oktober, 1623-1629.

[7] A. Namiki and M. Ishikawa (2003): *Robotic Catching Using a Direct Mapping from Visual Information to Motor Command*; IEEE Int. Conf. on Robotics and Automation, September, 2400-2405.

[8] M. Riley and C.G. Atkeson (2002), *Robot Catching: Towards Engaging Human-Humanoid Interaction*, In Autonomous Robots, Vol. 12, Issue 1, Jan. 2002, 119-128.

[9] L. Acosta, J. J. Rodrigo, J. A. Méndez, G. N. Marichal, and M. Sigut (2003): *Ping-Pong Player Prototype - A PC-Based, Low Cost Ping-Pong Robot*; IEEE Robotics & Automation Magazine, December, 44-52.

[10] R. Mori and K. Hashimoto and F. Miyazaki (2004): *Tracking and Catching of 3D Flying Target based on GAG Strategy*; Proceedings of the 2004 IEEE Int. Conf. on Robotic & Automation, April, 5189-5194.

[11] P.K. Allen, A. Timcenko, B. Yoshimi and P. Michelman (1991): *Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System*; Technical Report CUCS-034-91, Department of Computer Science, Columbia University, New York.

[12] D. Scaramuzza, S. Pagnottelli and P. Valigi (2005), *Ball Detection and Predictive Ball Following Based on a Stereoscopic Vision System*, In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, April 2005, Barcelona, Spain, pp. 1573-1578.

[13] T. Burghardt, B. Thomas and A. Calway (2003), *Circle Detection in Images*, In Image Processing Assignment - COMS 30121.

[14] A.A. Rad, K. Faez and N. Qaragozlou (2003), *Fast Circle Detection Using Gradient Pair Vectors*, In Proc. VIIth Digital Image Computing: Techinques and Applications, 10-12 Dec. 2003, Sydney, pp. 879-887.