Bull’s Eye – Never miss a shot!

Class project report - Neural Engineering, SS2015

Team members:
Jan Behrenbeck, Andreas Plieninger, Silvan Streit

Short description:
In our class project we tracked a flying dart with two DVS128 cameras, reconstructed the flight curve and predicted the point of impact in the dart board plane. This was done with curve fitting, taking into account gravity and the separability of the movement. This could theoretically enable us to move a motorized dart board to the estimated position.

Our setup:

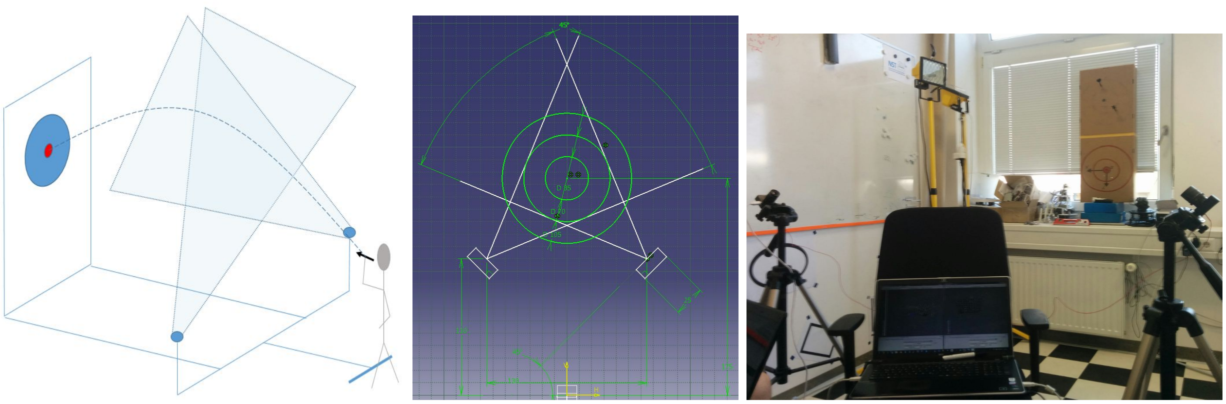

Figure: Sketch of our setup (left and center) and the final setup (right). The DVS are mounted perpendicular the motion vector of the dart, with a relative angle of 90 degrees. The coordinate system of the dartboard is defined with x and y lying in the plane of the board and z pointing towards the player. Y points upwards and x points to the right from the perspective of the player. The origin is in the center of the relaxed dartboard.

Explanation of the control loop:

Step 1: Dart tracking - To track the dart, we analyze the data that we receive from the DVS and search for for a fast moving object (many events). This was implemented by a trigger, activated by a high event rate. In order to map the points of the two cameras, further the timestamp has to be synchronized. Mapping is done on average points over multiple events, in order to reduce the error by the finite size of the dart.

Step 2: Calculation - Based on the pixel position in the two cameras we can calculate the true coordinates in real space. With all points we can calculate the flight curve and estimate the point of impact on the dartboard. Details on this are described below.

Step 3: Board actuation – The output of our analysis is an estimate of the position on the dartboard. Given this, the automatic dartboard could move to that position in order to let the dart hit the Bull’s Eye.

<u>Selecting the position of the DVS</u>

The position of the cameras is crucial to the problem, because it determines the spatial resolution for the tracking of the dart and the reaction time left for the dart board. We focused a lot on different angles and tried to analytically and practically determine a good choice. The following positions were considered.

1. Mounting two cameras on the moving dartboard

This position would allow easy positioning of the dartboard, as we do not need absolute coordinates of the dart, but much rather can try to center it in the field of view of the two cameras with a simple control loop. The problem with this setup is, that the dartboard has to move and the camera has to continue evaluating the position of the dart relative to the board continuouslyl. This was found to be tricky, as once the dartboard starts moving, the cameras perceive a lot of noise data from the moving background. Further, as the movement of the board is not uniform, trying to remove the background and detect the dart seemed a nontrivial task and therefore this position was not considered further.

2. Mounting the cameras on the frame of the dartboard

This setup removes the fast movement of the dart board itself but needs absolute coordinates for positioning. Tests of this setup showed, that one can easily detect the dart coming towards the board, as it drastically increases in size as it approaches the setup. People who threw the dart were hardly perceived, as they were far enough away to not be resolved good enough. This however leads to the problem of timing, as also the dart is only perceived once it gets close to the dart board. This means, that only the last 50cm can efficiently be tracked, which does not work with positioning time of the dartboard of 150ms. Further, fixing the cameras to the dart board frame shows, that when the dartboard accelerates, the frame vibrates strongly, leading to a large amount of noise in the camera.

3. Mounting the cameras facing towards the dartboard

The next alternative was, to mount the cameras facing towards the dartboard at a distance of about 1.8m. This has two advantages. On the one hand side, this allows for auto-calibration, as the target is visible for the cameras and so they can determine their own position relative to it. On the other hand, also the beginning of the flight curve is perceived, which allows early positioning of the dart board. Here, however, only few pixels are activated during the first few centimeters, which leads to a low spatial resolution of the trajectory. Also the target is badly resolved in the far distance and hence does not allow for very accurate auto-calibration.

4. Mounting the cameras perpendicular to the dart-motion

We mounted the cameras at a distance of 1.8m away from the dartboard perpendicular to the motion vector of the dart with a relative angle of 90 degrees (see setup above). This position was selected to be the most suitable setup for this project. Here, a high spatial resolution is achieved at the very beginning of the trajectory which allows a early prediction of the point of impact. This allows early positioning of the dartboard. On the downside however, no direct auto-calibration is possible, as the target is not in the field of view. This could theoretically however be indirectly achieved by using a reference laser point to the target (pulsed) or a test-shot with a precision tool.

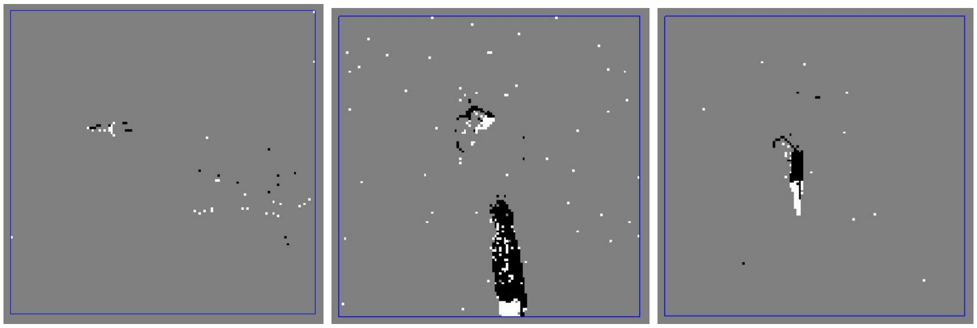

Figure: Frames of the recording of different dart positions. (left) dart from the side (center) front perspective (right) bottom perspective

Dart tracking algorithm

1. Triggering: As soon as more than 30 new events per 5 milliseconds arrive, the dart is assumed to be in the field of view. Analog for the end of the trajectory, when the dart leaves the field of view.

2. Estimate position: A mean is iteratively updated with the new events to estimate the dart position. The new mean is the old mean plus the difference between the event and the old mean. After the initial 45 events, where the difference is weighted times 0.2, the mean moves slowly and smoothly with the trajectory of the dart. The difference is then weighted with a factor one over the euclidian of the difference, resulting in close pixels to be weighted higher.

3. Matching: The points are matched by their timestamp to correlate them to the same event with an accuracy of one millisecond.

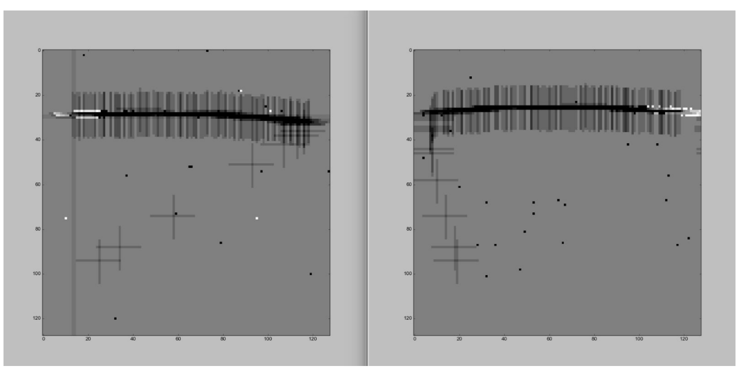

Figure: Tracked event means of the trajectory of the dart from left and right.

Conversion to real world coordinates

1. Converting points to angles

Each pixel corresponds to a certain angle in space, due to the opening angle of the cameras. We determined this by measuring the size of objects which cover the whole field of view in various distances. This was done with the blinking LED-board supplied by the chair.

The total angle was then linearly divided among the pixels, with the center of the camera having an opening angle of 0 degrees. This leads to:

angle_of_pixel = angle_per_px * (pixel - center_offset_px)

2. Matching angles to real space coordinates

Knowing the viewing angles of the pixels from the camera allows to find the intersection of the two directions of the two cameras. First in the vertical x-y-plane the angles were converted to points there, and then the z-value could be determined for each camera. As we had a system with four inputs (2 angels per camera) and three unknown parameters (x,y,z) the system is overdetermined, and we have two solutions for z: z and z_{prime}, which could be averaged or used for verification. With a properly justified setup the difference between the two was as low as 0.5cm.

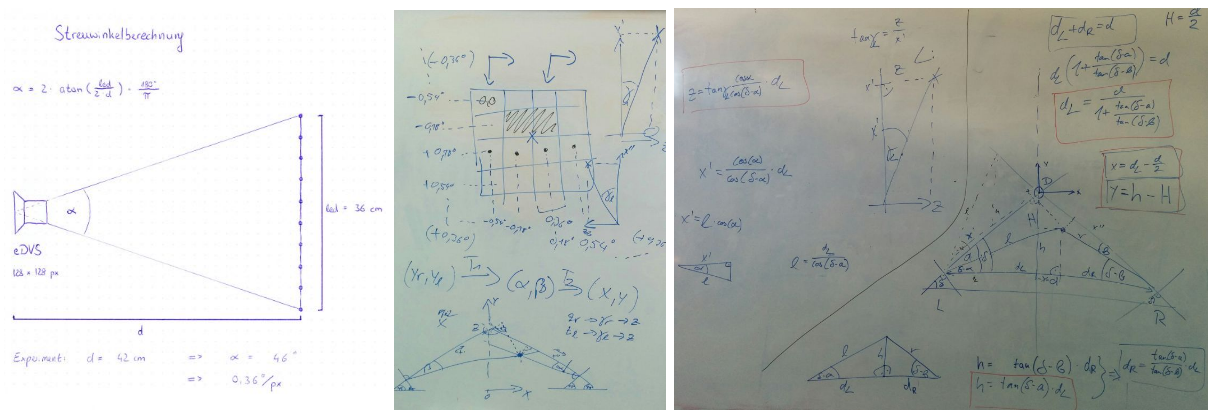

Figure: Conversion to real world coordinates. (left) determination of the opening angle of the camera (center) ideas to convert pixels to opening angle (right) map angles to space coordinates

<u>Estimation of the point of impact</u>

We developed two versions of the calculation of the point of impact.

1. Linear and quadratic fit

The real coordinate values of the fitted crosses on the dart trajectory are projected in the horizontal x-z-plane and into the vertical y-z-plane. In the horizontal plane a linear movement of the dart is expected, because no force acts on it (we neglect friction due to the short time scale and aerodynamic shape of the dart). In the vertical plane a quadratic fit has to be done, because gravity acts on the dart which adds a acceleration component pointing downwards, in negative y direction. The point of impact is then estimated by the crossing of the fits with the z=0 plane, where the dart board lies.

2. Fit with considering time

The movement of the dart can also be split into three independent movements along each axis. As there is no horizontal force (again neglecting friction) the movement along the x and z axis will be linear in time and can be fitted with $x = x_0 + v_x * t$ and $z = z_= + v_z * t$. Along the y axis a quadratic term has to be added, due to gravity: $y = y_0 + v_y * t + 0.5 * g * t^2$. As the gravity is known, the fit is only done over the first two coefficient and the last one is set to $g = -9.81 m/s^2$.

The point of impact is then calculated in two steps. First the time of impact is evaluated by setting $z = 0$ and solving for $t_a = t$. The result can then be inserted into $x(t)$ and $y(t)$ resulting in the input coordinates x_a and y_a. This technique has the advantage of using more knowledge from our measurements and leaving out the quadratic fit, which is not very accurate on our short observation time.

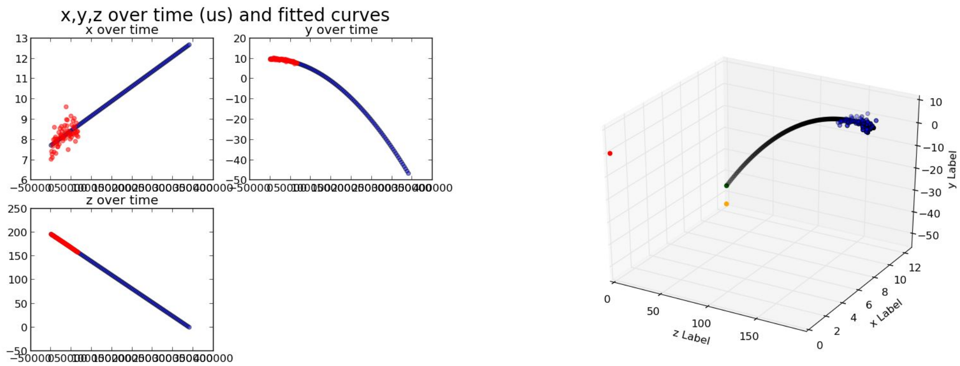

Figure: Flight path and point of impact over time (left) and in 3D-space (right).

Error Propagation - estimating the spatial resolution

Assuming that the setup is properly justified and all angles and distances are correct, we did a error propagation calculation on the effect by having the events one pixel of. Doing an error propagation calculation on the calculation of the real world coordinates leads to a maximum error of 0.5cm for the x-coordinate per pixel and -1.4cm for the y-coordinate, assuming a shift of the signal by one pixel along the y-axis. A shift by one pixel along the x-axis of the camera results in a maximum error of 1.2cm for the z-coordinate. This also seems reasonable, as z and z_prime have a fluctuation of up to 1 cm on a properly justified setup.

If we further assume that there is no error at the beginning of the detection zone and an one bit shift appeared at the end, then this would lead to impact point estimation error of -4.6cm on the y-axis and 1.5cm on the x-axis.

Conclusion and outlook

In this class project, we were able to show, that it is possible to track a dart using DVS cameras. This allows using all benefits from event based vision sensors. Here, especially the independence of the signal to a static background and the high temporal resolution was used. With physically correct fitting a good approximation could be generated for the impact point of the dart.

Unfortunately the time of this class project was not enough to further integrate the system with the existing setup of the motorized dartboard. As a next step further improvements of the code have to be done, in order to meet the timing requirements (50 ms). With proper calibration one could theoretically resolve a good estimate of the impact point with an accuracy of a few centimeters (see error propagation calculation). This was not achieved in practice in this project, due to the simple setup and inacurate calibrations. Further improvements could be, including auto-calibration of the system by detecting the relative position of the cameras to one another and maybe using a third camera to bring the two in correlation with the dartboard position. A good calibration could also be achieved by justifying the setup and fixing it by connecting it with dart board frame.