

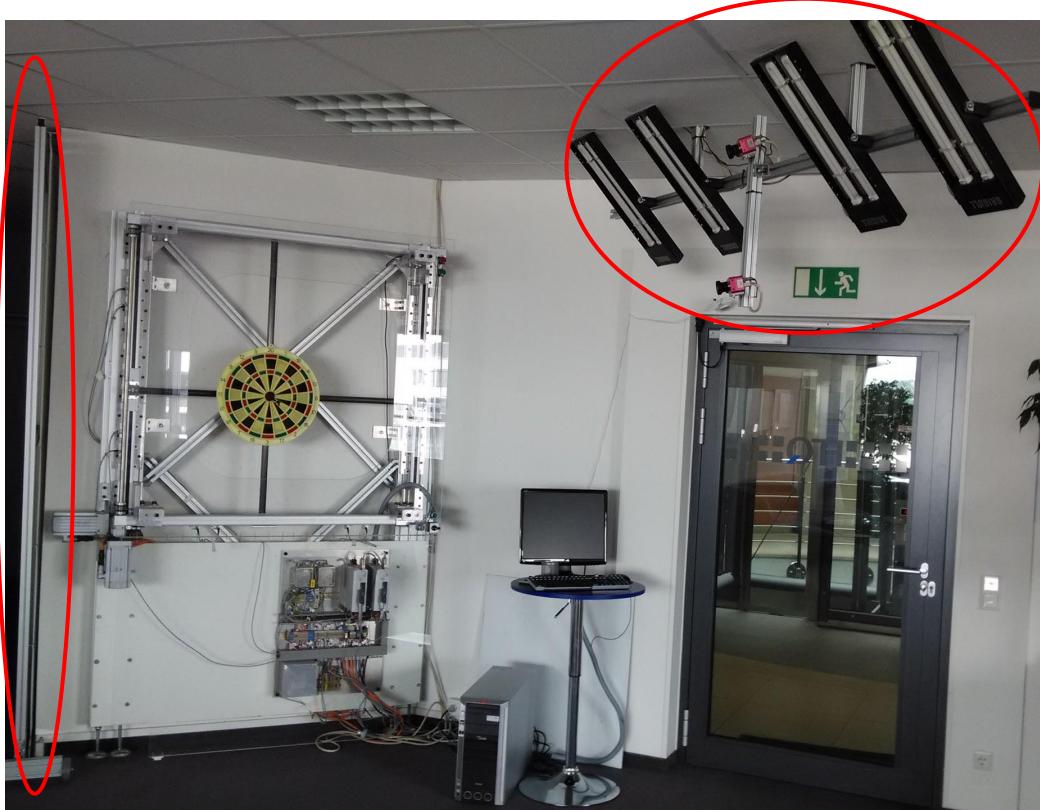
Bull's Eye

Never miss a shot!



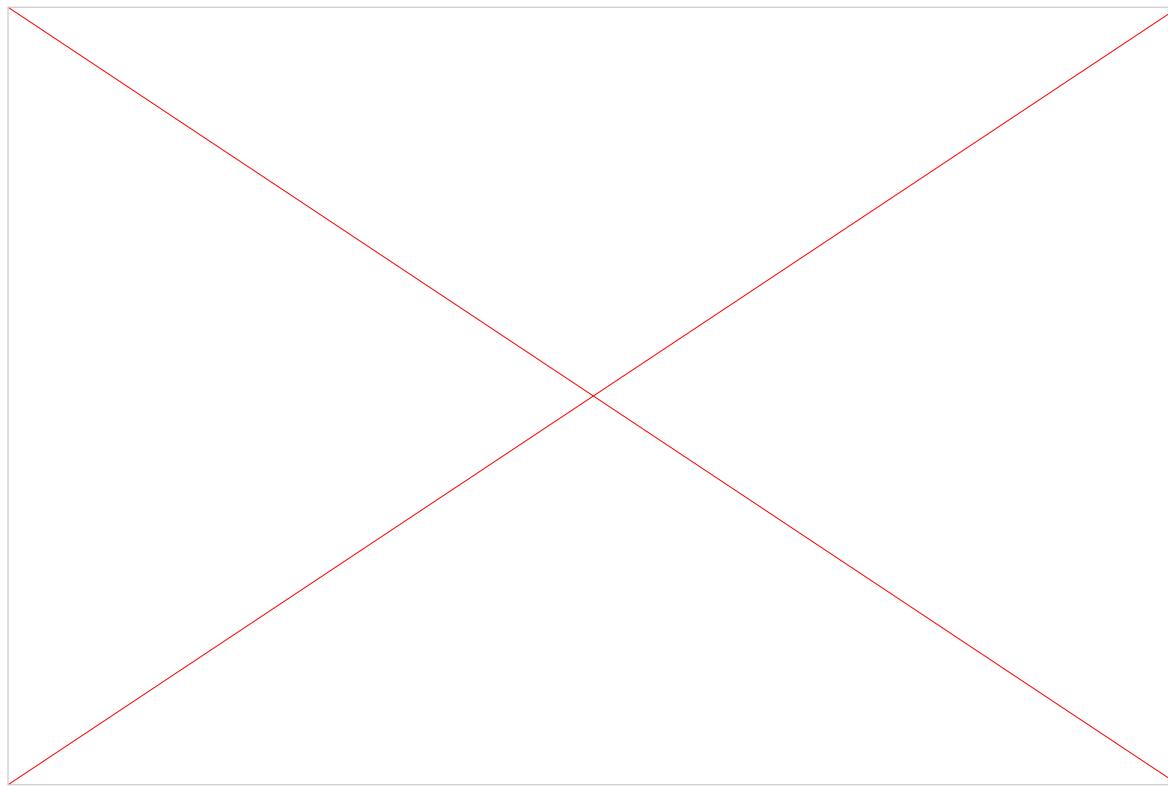
The old model: big and sensitive

White wall to remove background noise

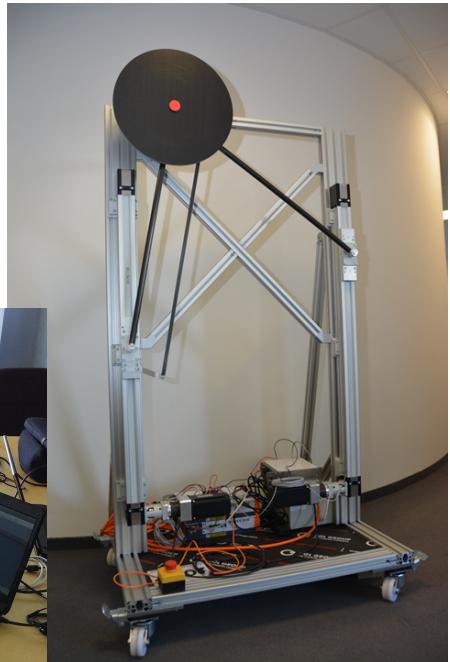
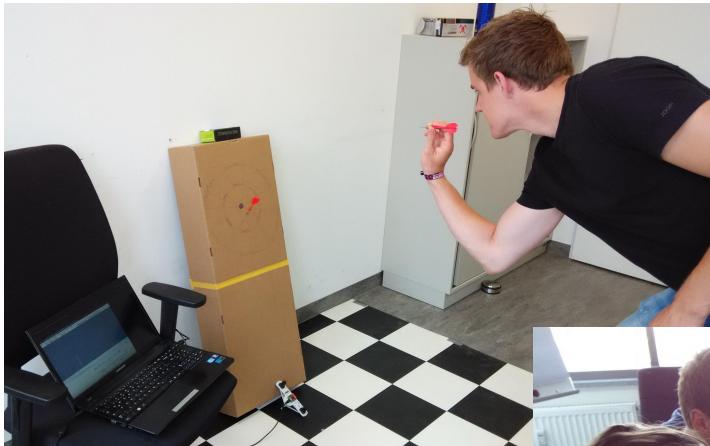


Camera system with special lightning

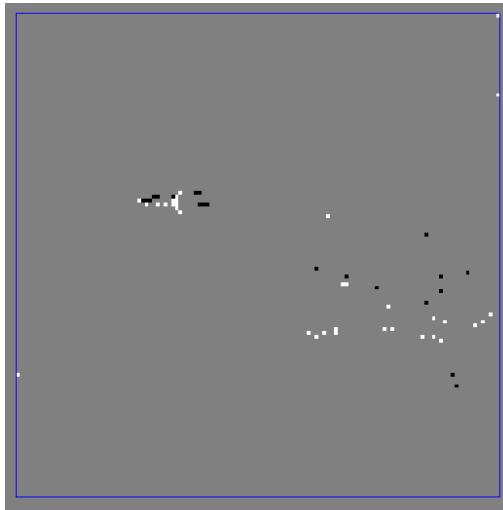
Final Demo



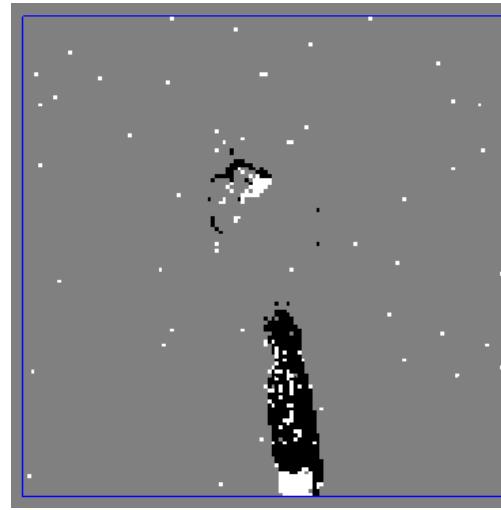
Sample recordings with two eDVS/DVS128



Rudimentary setup: test ideas



side perspective

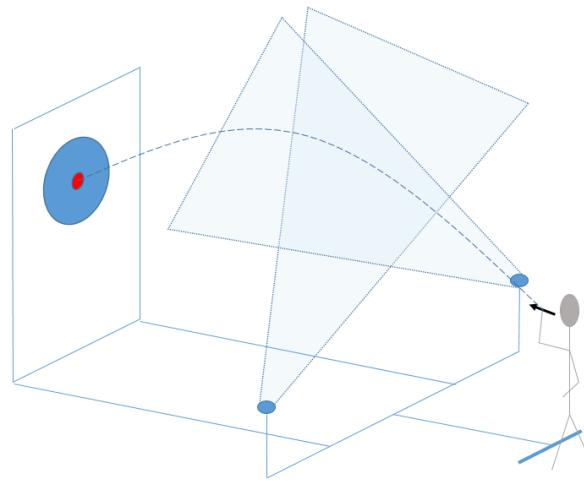
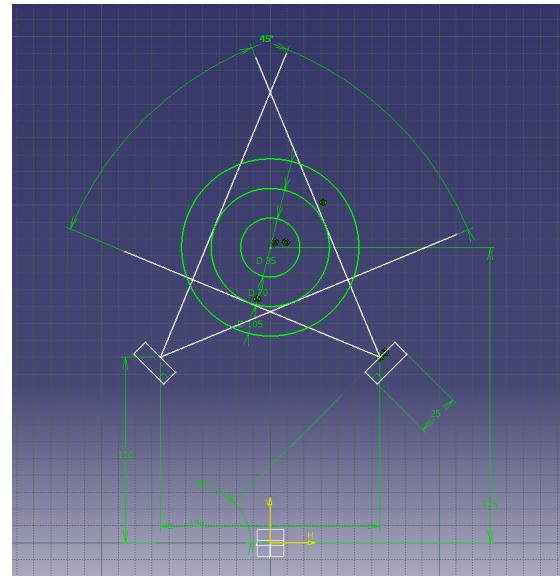


front perspective

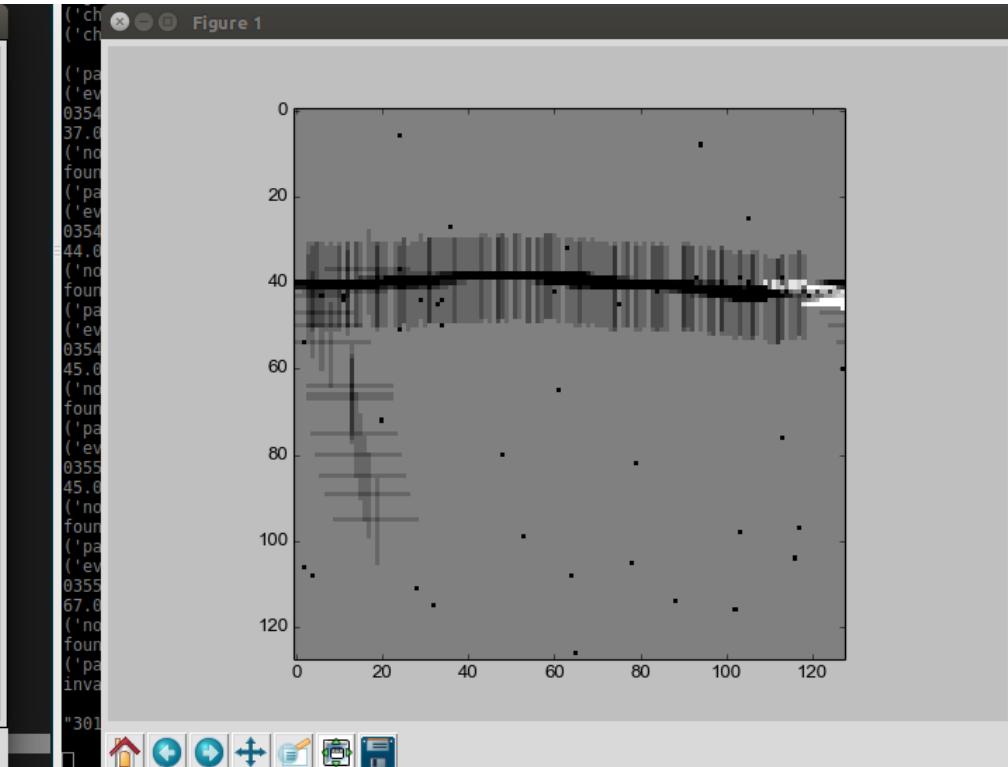
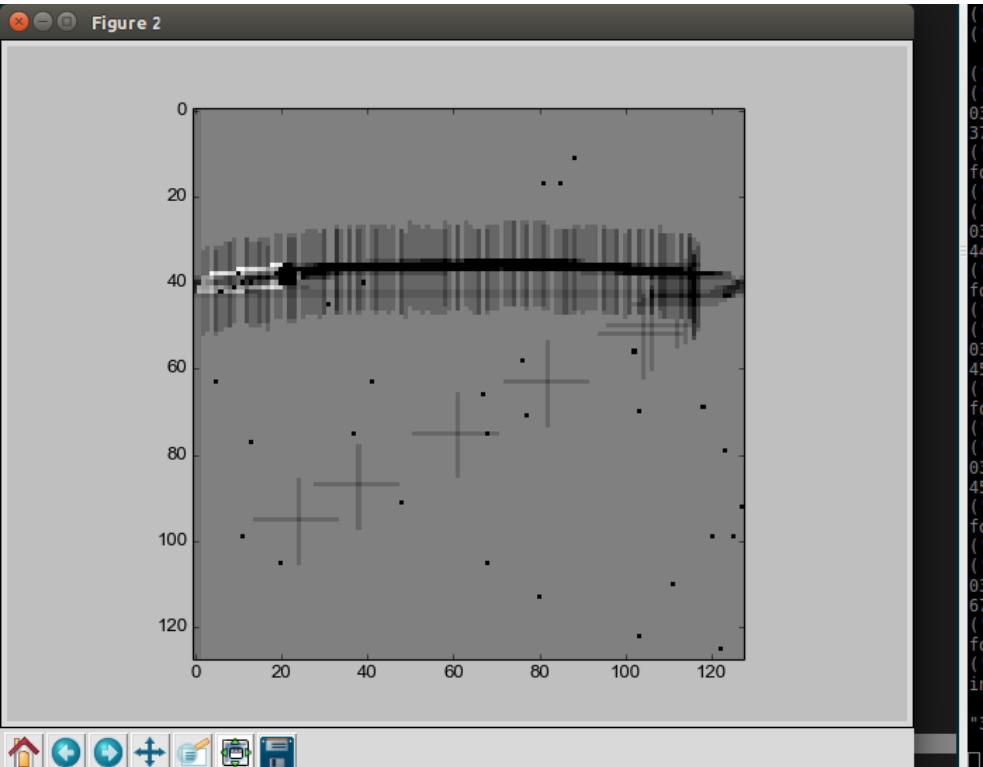


bottom perspective

Final Setup

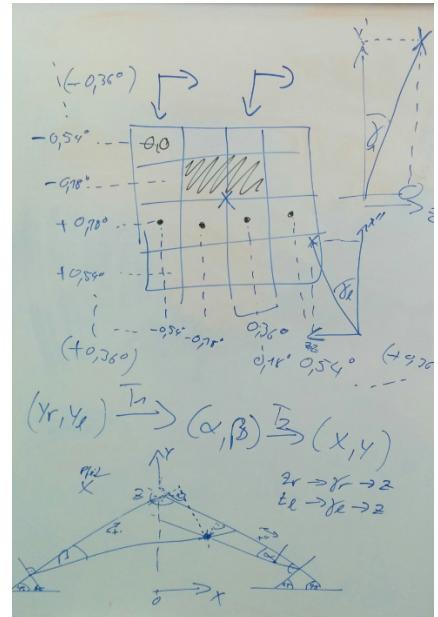


Triggering, Position Estimation & Matching



Calculate 3D Position

- use opening angle per pixel (1 px = 0,36°) and trigonometry
- accuracy: (5,14,12) mm



Handwritten mathematical derivation showing the conversion from 2D image coordinates to 3D world coordinates. It includes trigonometric formulas for calculating distances d_L and d_R , and the resulting 3D coordinates (x, y, z) .

$$z = \tan(\alpha) \frac{d_L}{\cos(\delta - \alpha)} \cdot d_L$$

$$x' = l \cdot \cos(\alpha)$$

$$l = \frac{d_L}{\tan(\delta - \alpha)}$$

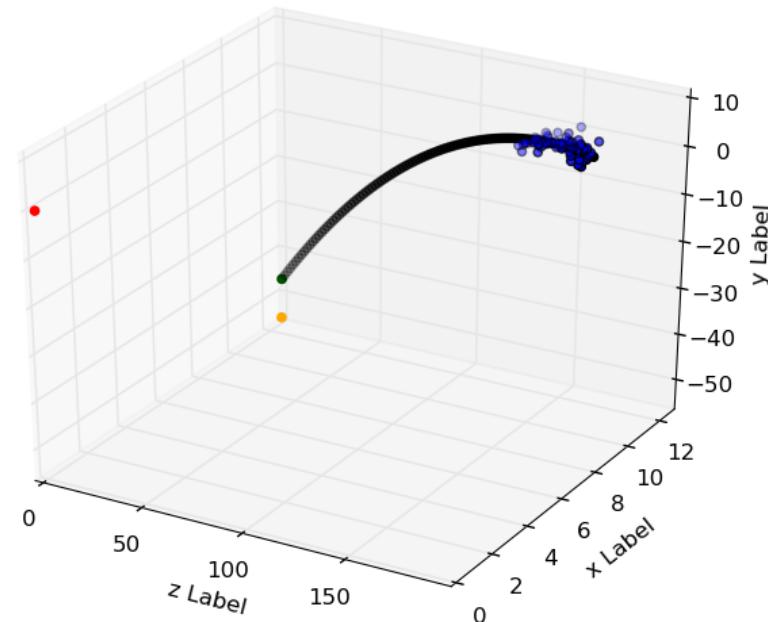
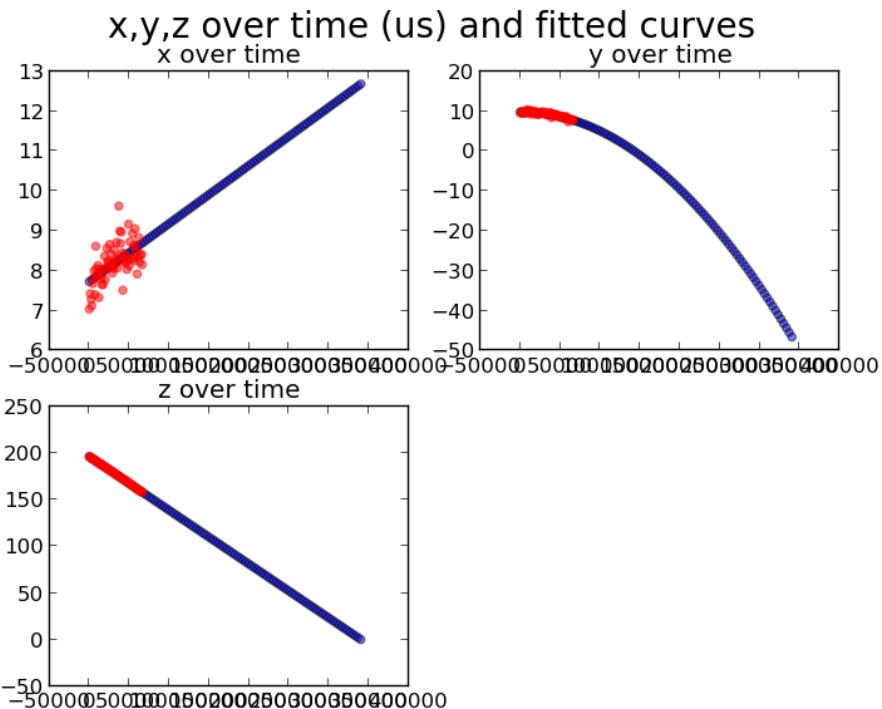
$$\begin{aligned} d_L + d_R &= d \\ d_L \left(1 + \frac{\tan(\delta - \alpha)}{\tan(\delta - \beta)} \right) &= d \\ d_L &= \frac{d}{1 + \frac{\tan(\delta - \alpha)}{\tan(\delta - \beta)}} \end{aligned}$$

$$x = d_L - \frac{d}{2}$$

$$y = h - H$$

$$\begin{aligned} h &= \tan(\delta - \beta) \cdot d_R \\ h &= \tan(\delta - \alpha) \cdot d_L \end{aligned} \Rightarrow d_R = \frac{\tan(\delta - \beta)}{\tan(\delta - \alpha)} d_L$$

Estimate impact point



Conclusion

We are able to...

1. track the dart
2. calculate its 3D position each millisecond
with an accuracy of (5,14,12) mm
3. predict the point of impact with a theoretical
accuracy of (15,46) mm

During the process we experienced many more hurdles

- camera lenses and configurations
(focus, angle, positioning, “blind” spots, lightning conditions, colour of dart → black, synchronisation)
- time for calculation
- difficult communication with the moving bullseye (CAN bus)
- calibration (→ precision)
- not enough time (!)

Outlook

Possible next steps:

- control the dartboard via our program
- calibrate the cameras by precise positioning
- build a system to attach the cameras to the dart rack