Path Tracking for Car-Like Mobile Robots Using Computer Vision

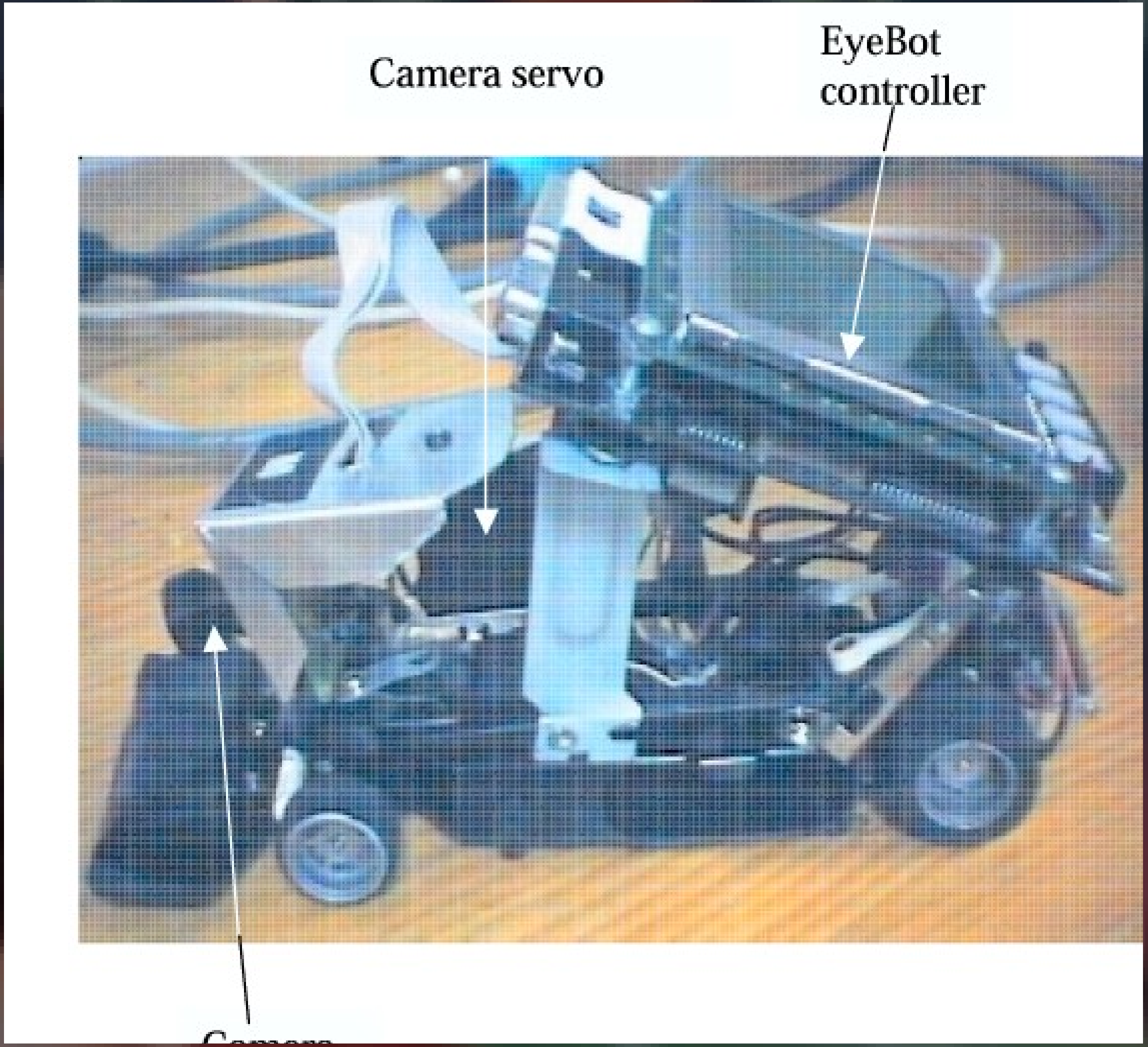
Andrew Thomson
CITR
University of Auckland
New Zealand

Jacky Baltes
Office 529 Machray Hall
Department of Computer Science
University of Manitoba
Canada
jacky@cs.umanitoba.ca

Introduction

- Following a path marked on the floor with a mobile robot
- Simple, quick algorithm
- Path is unknown
- Few assumptions about the kinematics of the robot
- Visual servoing: path marked on the floor with white tape
- Evaluation using a global vision system

Robot Hardware



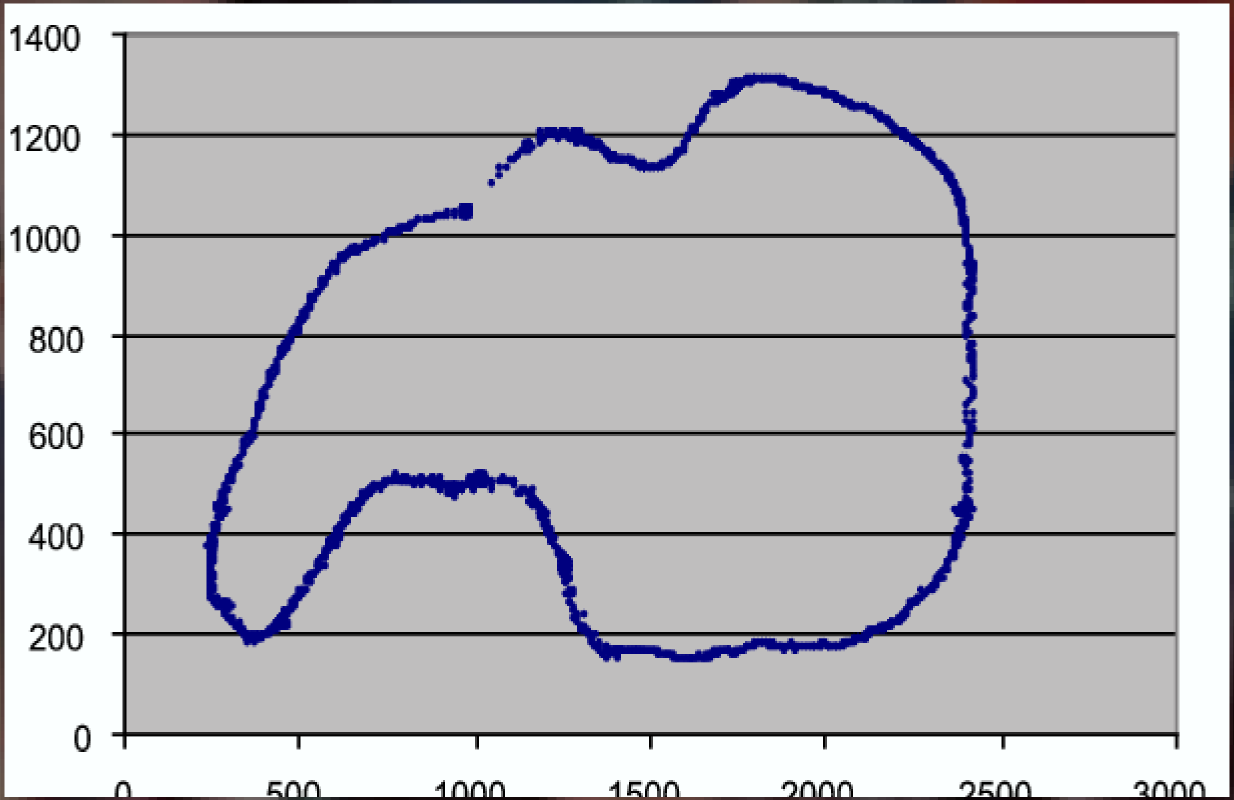
Base: Tamiya Toy Car, proportional speed and direction. No odometry.

Processing: Eyebot controller, 25MHz 68332 robotics controller, 1MB Ram

Sensors: CMOS camera 80x60 resolution in 24 bit color. 15 FPS

Global View of the Path

Path Geometry

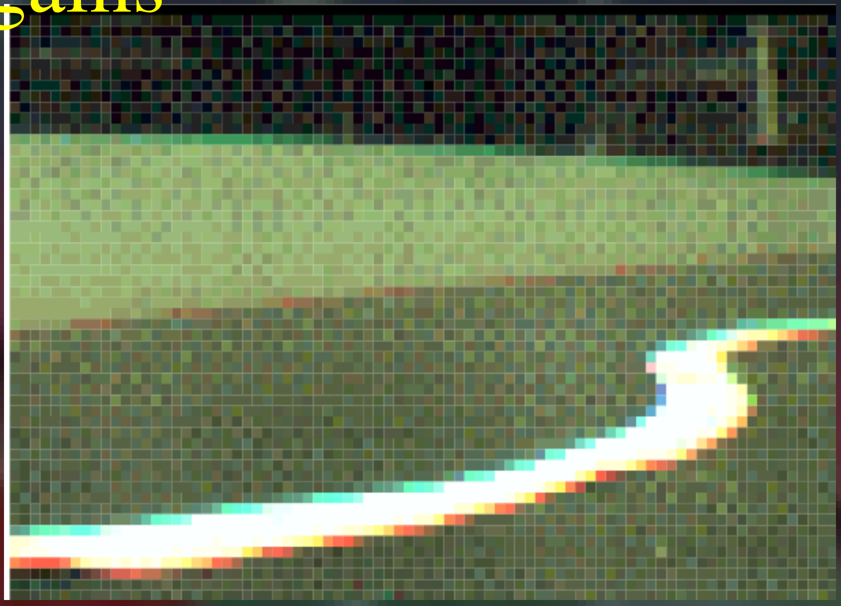


Camera Geometry

- Camera slanted towards the ground
- Robot can not see directly in front
  - Minimum distance is 8.8cm
- Lookahead distance

Camera Views

- Simple RGB threshold to extract path from image
- Sufficient in robotics application
- Green tint because the CMOS sensor is uncalibrated for R,G, and B gains



Path Tracking Control

- Most path tracking controllers for mobile robots use offset and gradient
- Balluchi uses offset, gradient and curvature
- Egerstedt uses look ahead

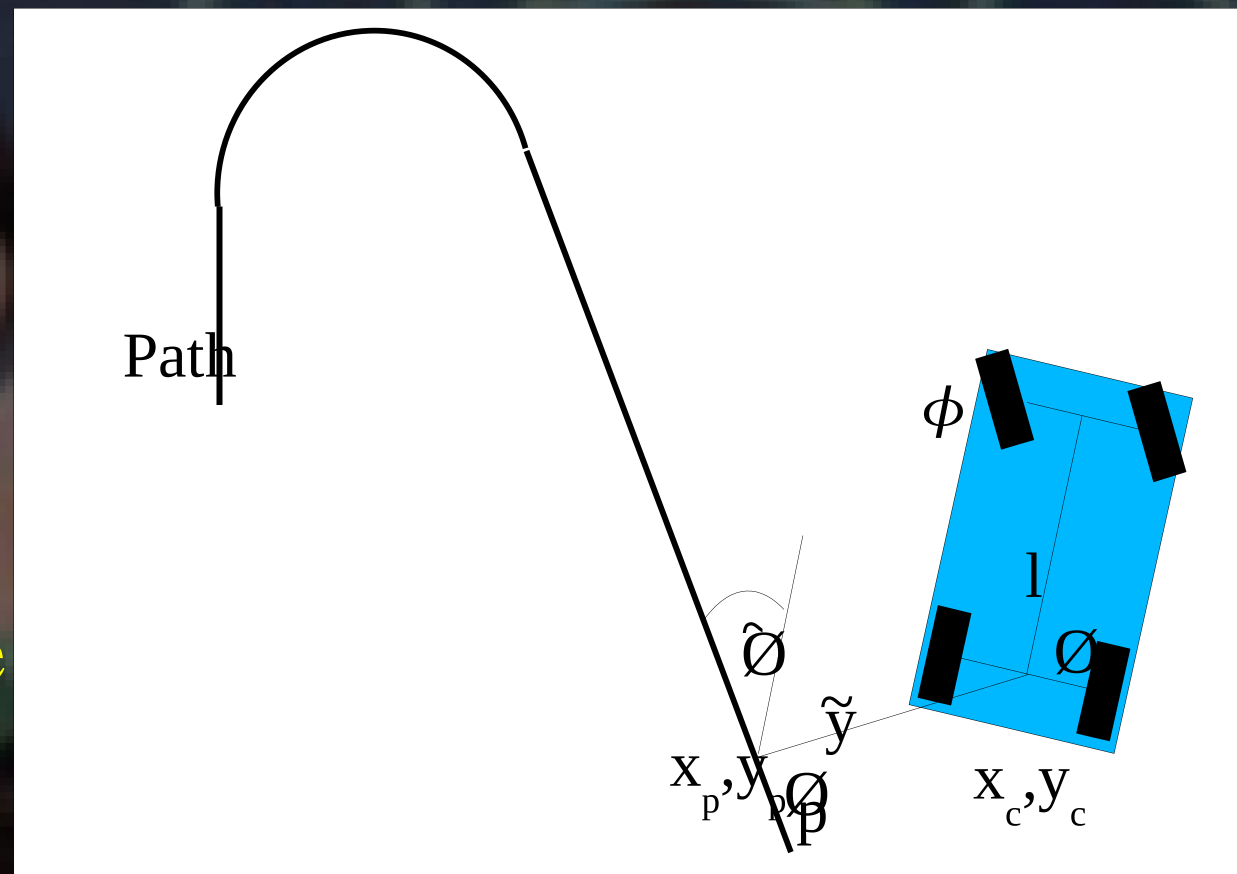


Image Processing

- Maintain 8 fps
- Process rectangle
- P1 bottom and right edge
- P2 top and left edge
- Worst case 210 Pixels
- Offset: distance to path
  - Avg. x of P1 and P2
- Gradient of P2P1

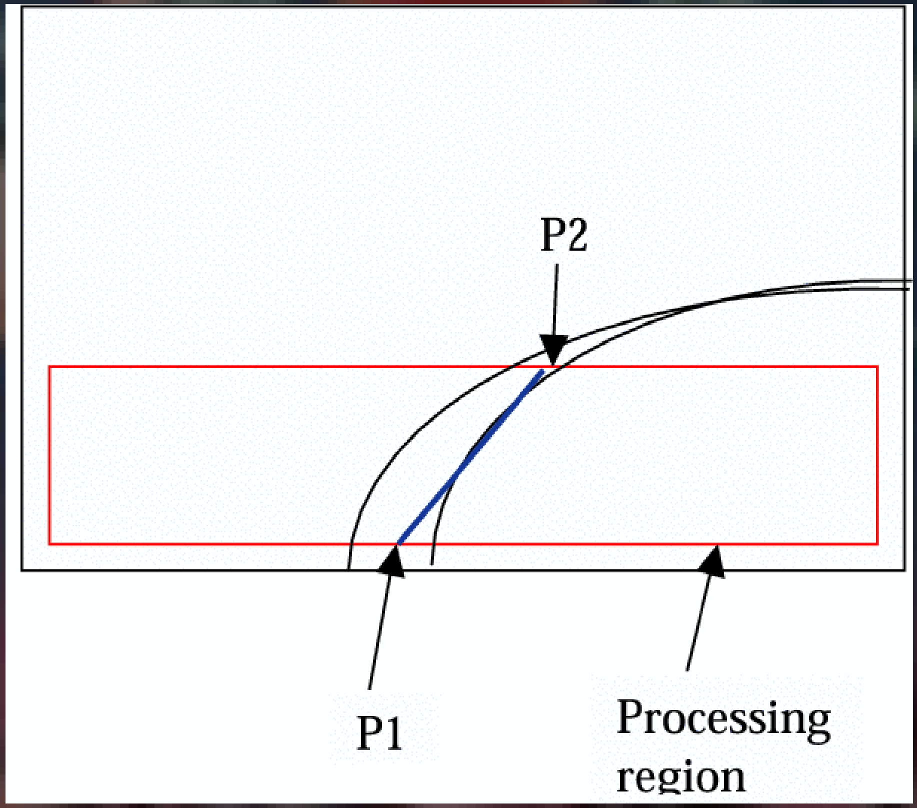
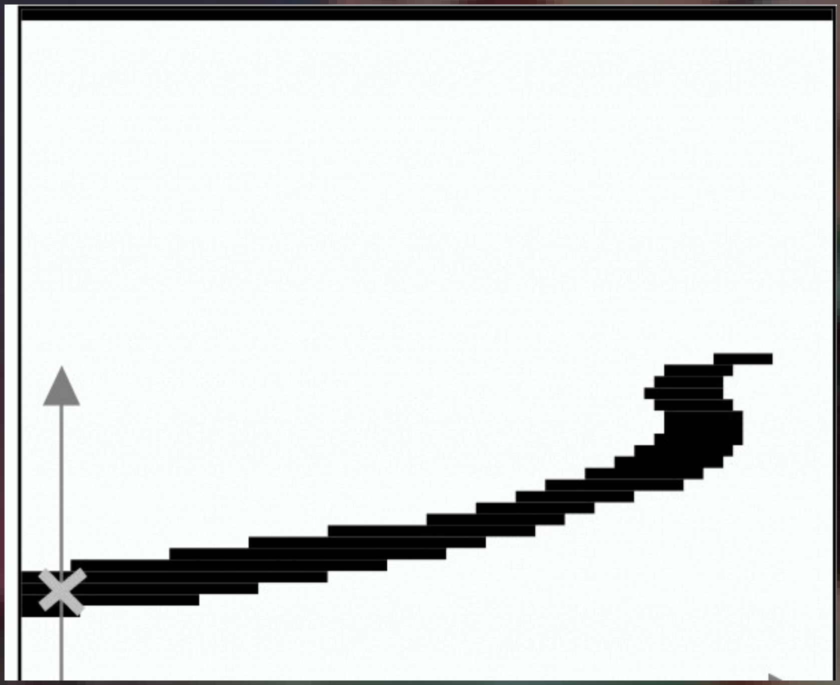
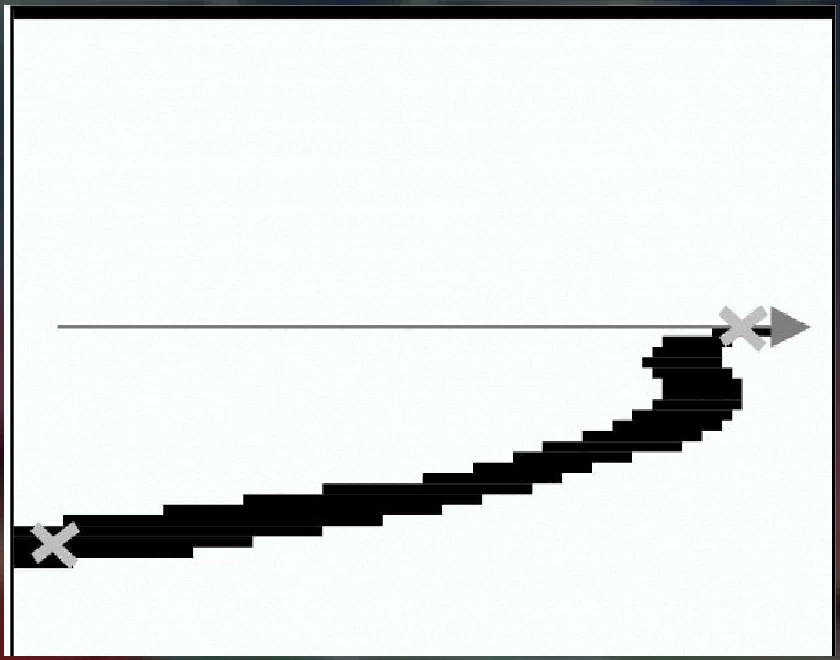


Image Processing Example

- Scan for P1: Bottom and left
- Scan for P2: Top and right





Control

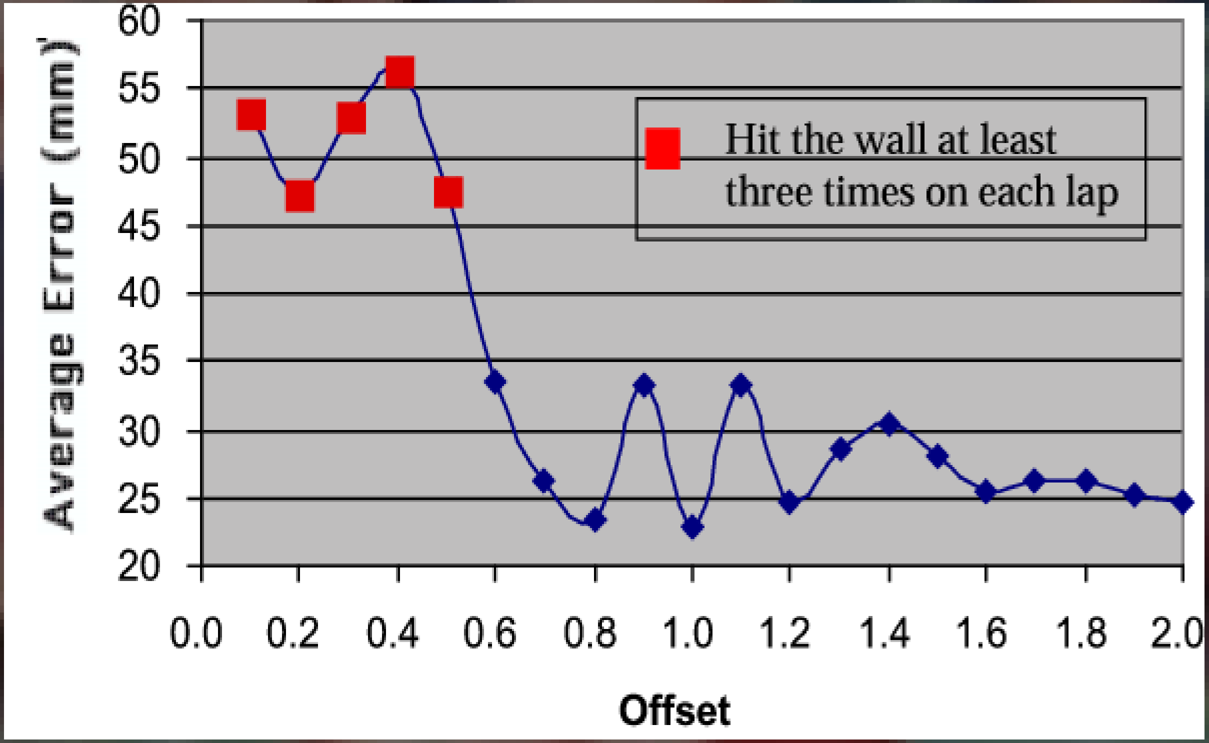
- Gradient and offset are normalized to -1 .. +1
- Weighted sum of gradient and offset is computed and a steering angle is extracted from a lookup table (9 entries)
- Steering angle determines speed (Slow, medium, fast)
- Specific to robots
  - Linear interpolation between the extreme steering angles

Evaluation

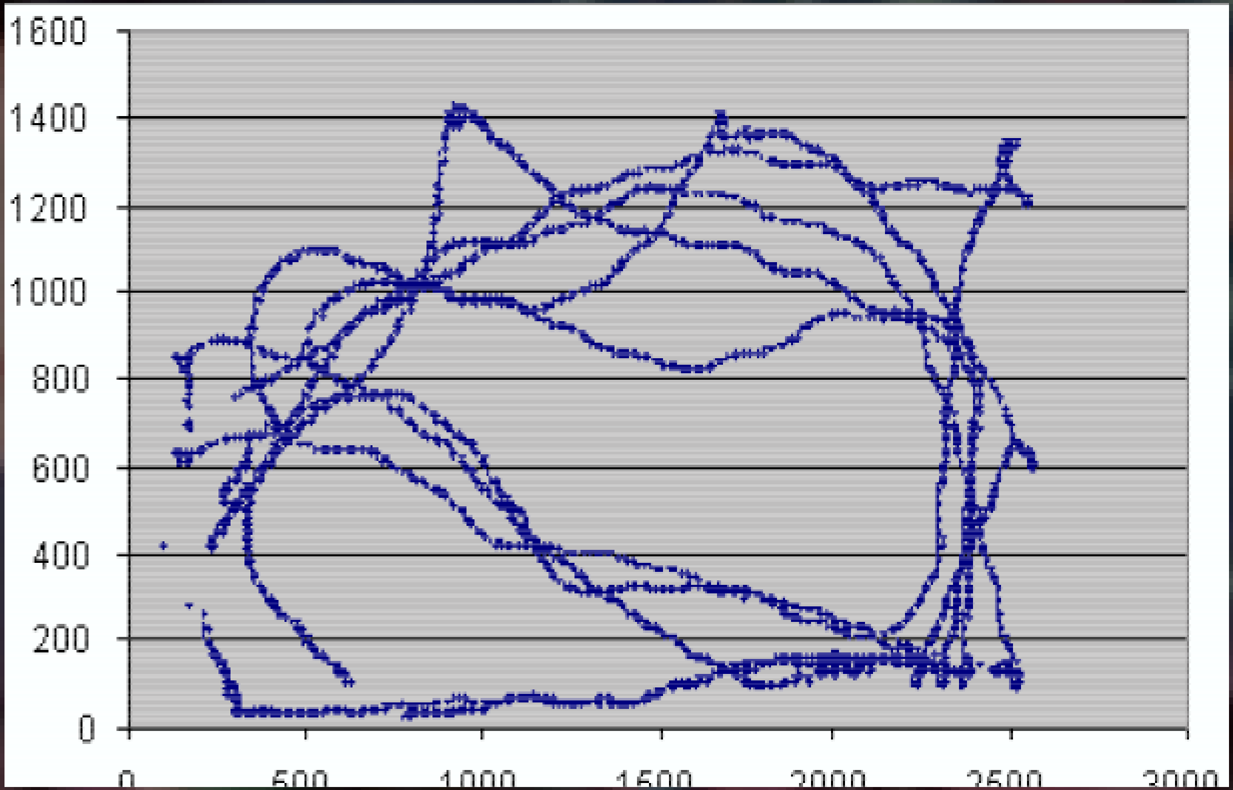
- Influence of different weightings for offset and gradient.
- What information is more important?
- Avg. error vs speed and avg. error vs. commands
- 3 Lookahead distances
  - 10 Pixels = 11.3cm
  - 20 Pixels = 13.5cm
  - 30 Pixels = 17.1cm
- 5 laps around the racetrack as quickly as possible

Average Error vs Offset Weighting 10 Pixel Lookahead

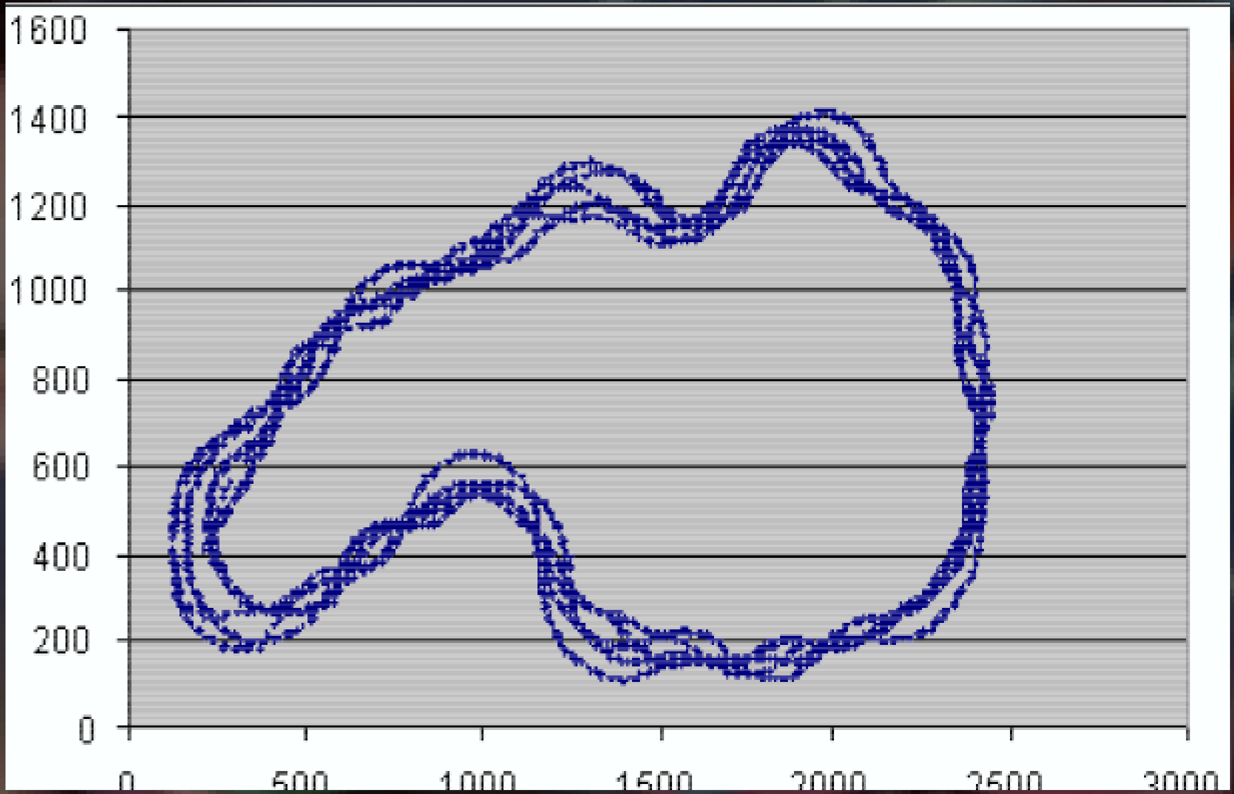
- Offset clearly needed
- Small influence of gradient



Path Offset 5%
10 Pixel Lookahead

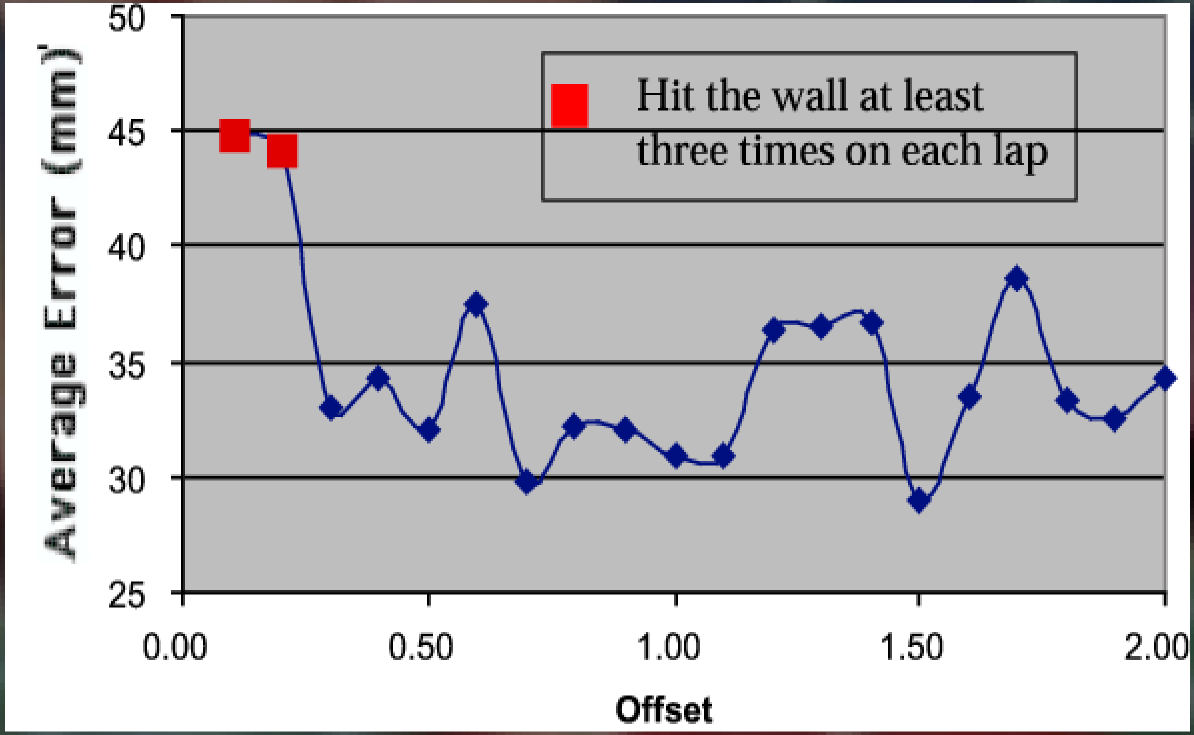


Evaluation Offset 50%
10 Pixel Lookahead



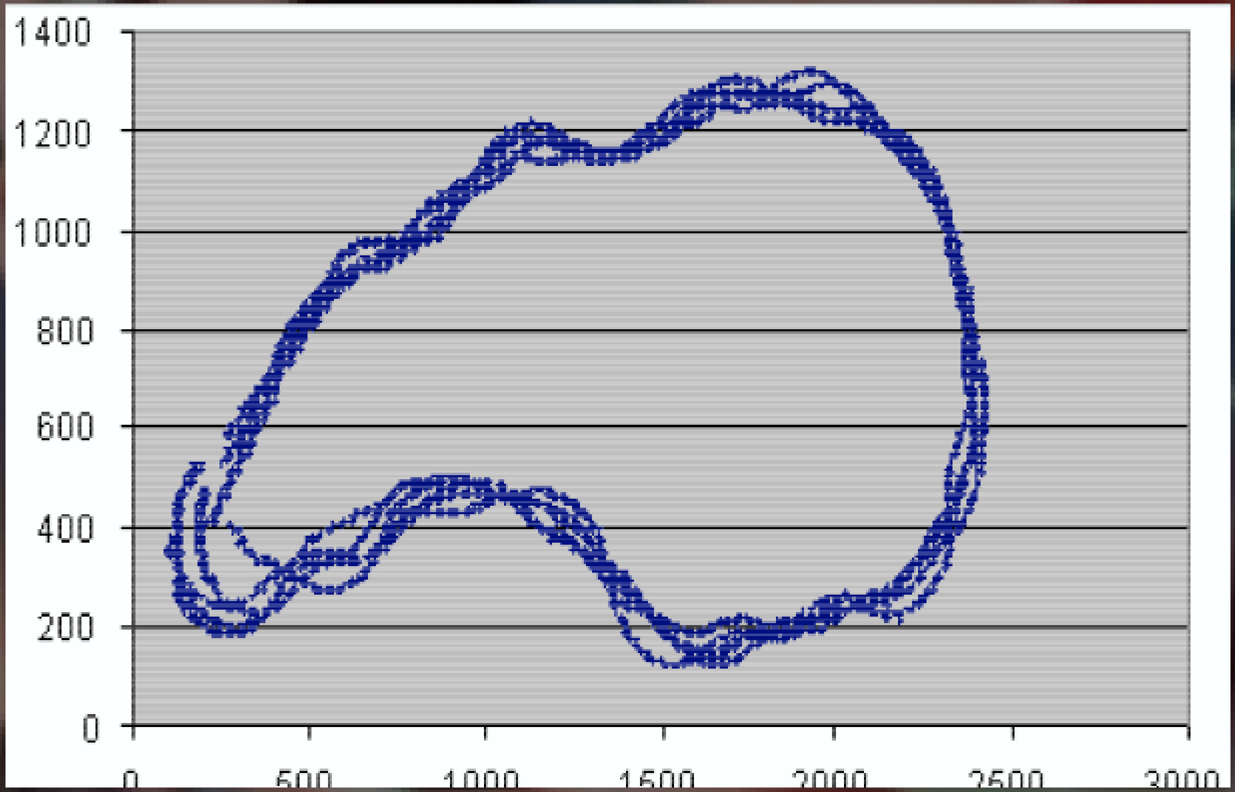
Average Error vs Offset
20 Pixel Lookahead

- Similar result to 10 pixel lookahead
- More variety in results
- Worse at 100% offset weighting



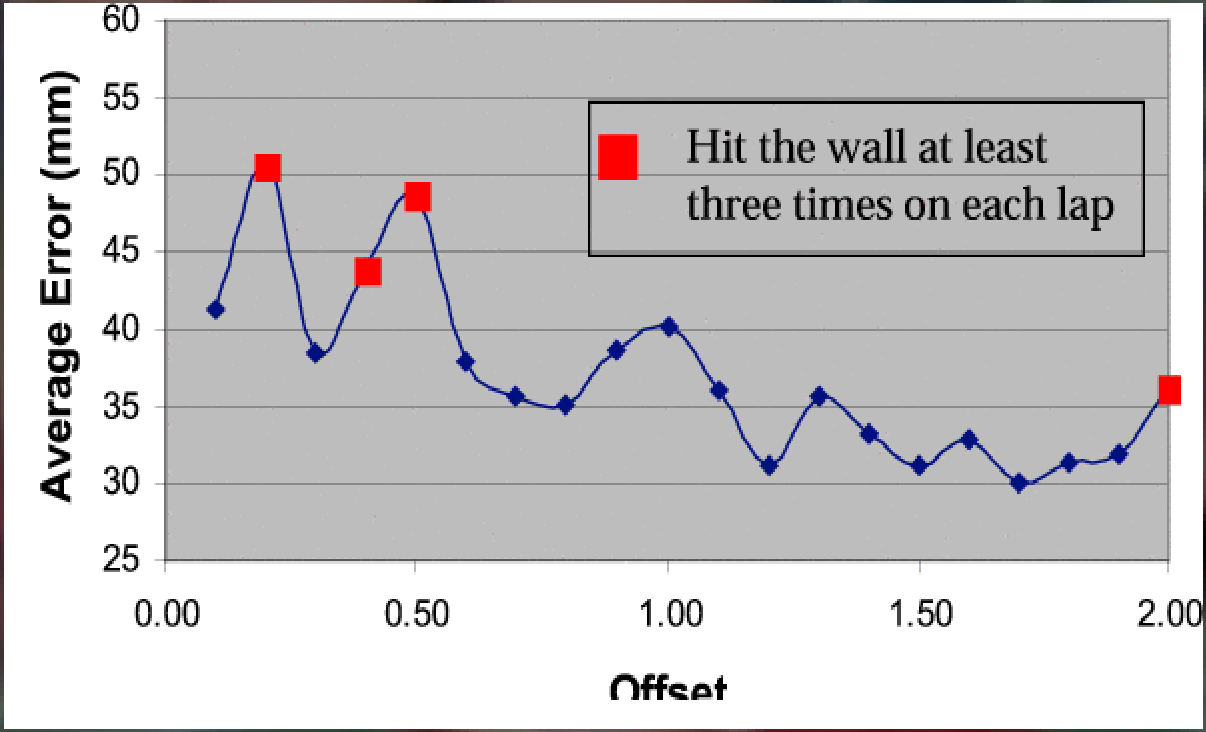
Evaluation Offset 75%
20 Pixel Lookahead

- Smoother curve than 10 pixel lookahead



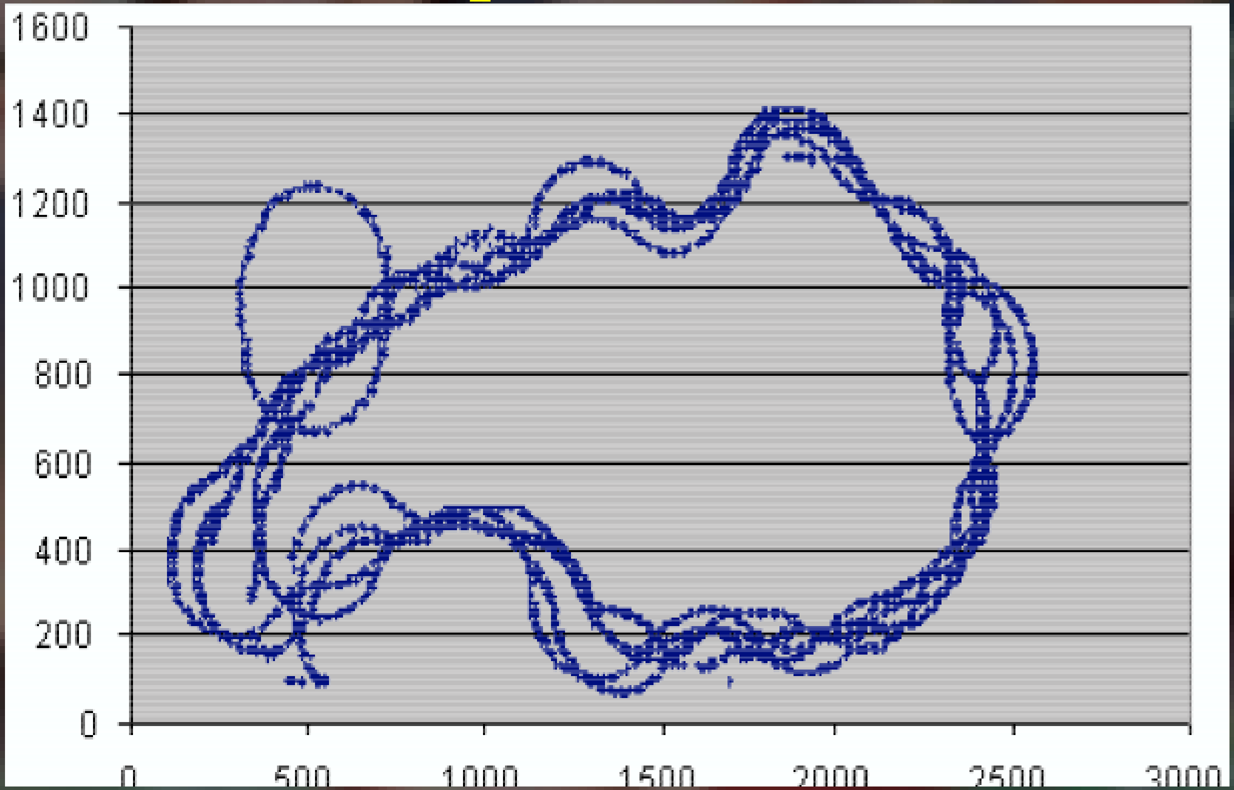
Average Error vs Offset Weighting 30 Pixel Lookahead

- Poor performance on 100% offset
- Average X coordinate of P1 and P2



Evaluation Offset 85%
30 Pixel Lookahead

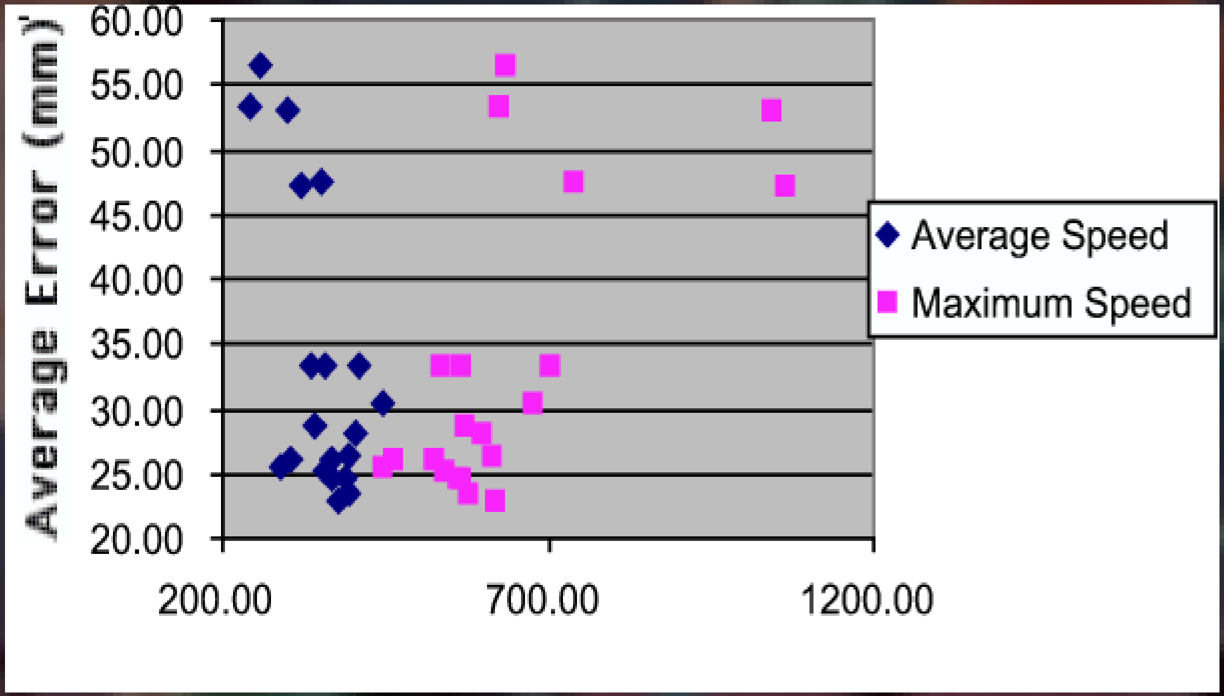
- Lost sight of track sometime
- Follow last known position



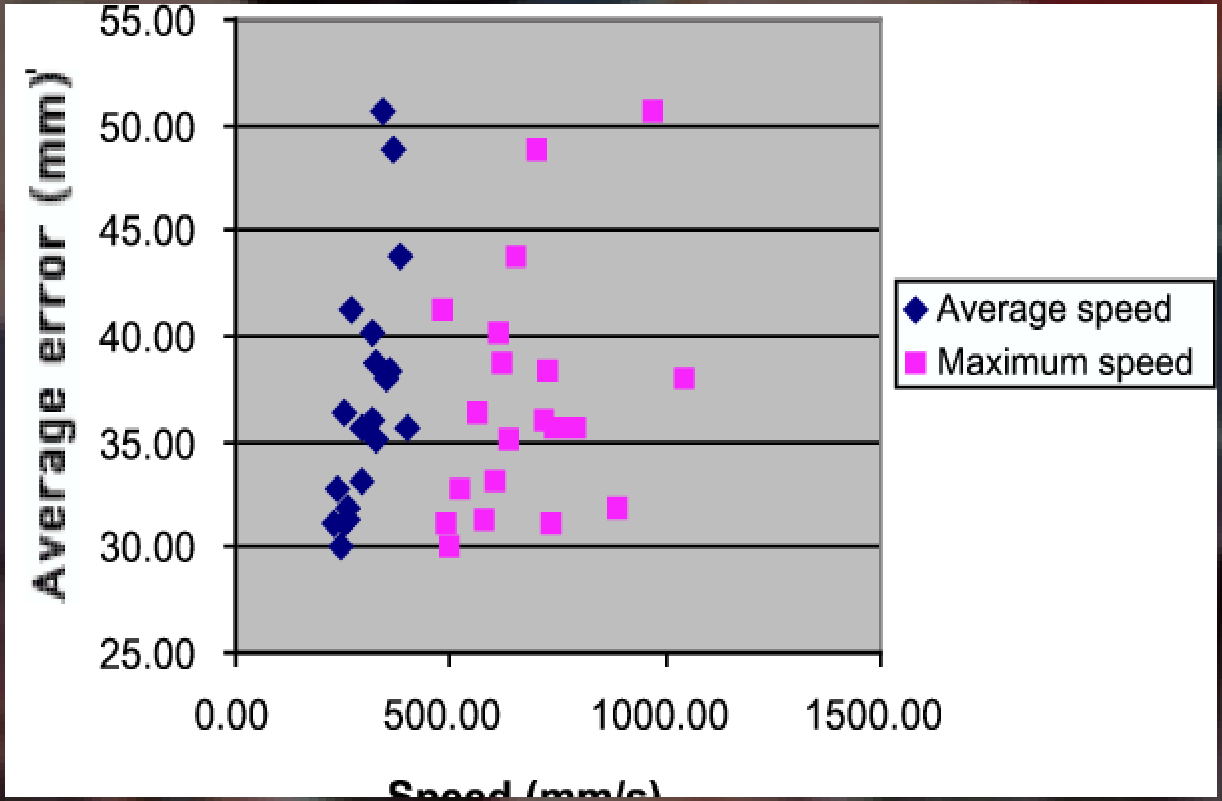
Average Error vs. Speed
Average Error vs. #Commands

- What is the influence of the error on the average and maximum speed of the robot
- No correlation
- Strong correlation between number of commands and error
  - Slow motion of the robot
  - More commands => closer to the path
  - No oversteering (Balluchi)

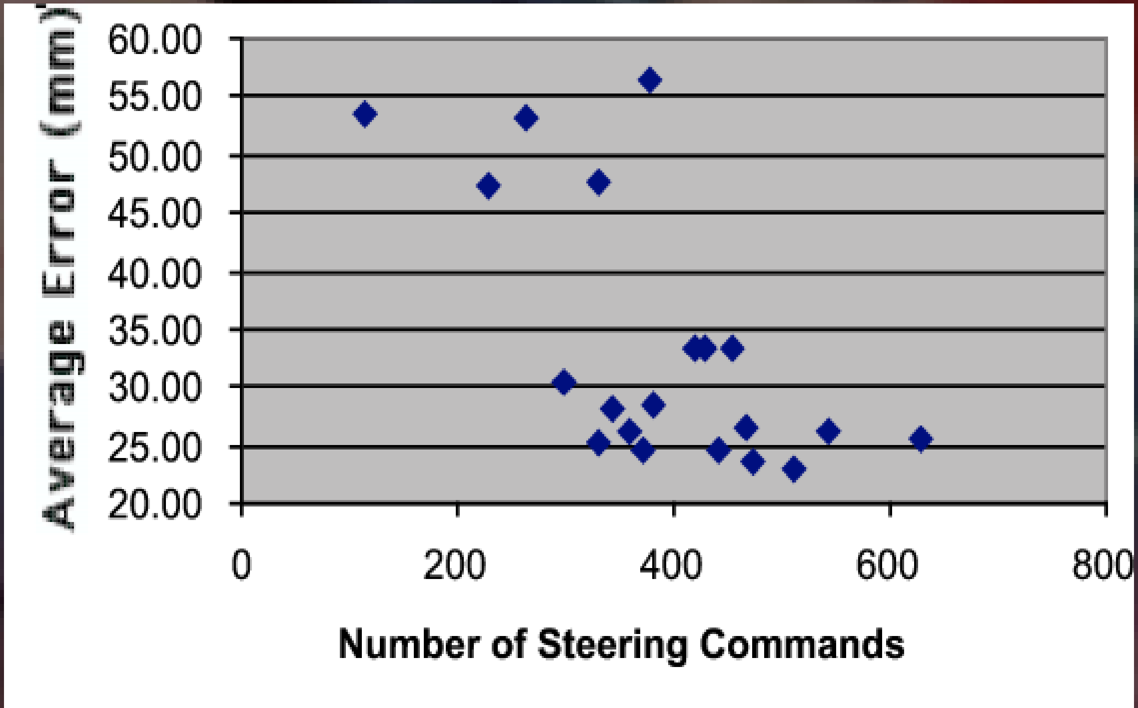
Average Error vs Speed
10 Pixel Lookahead



Average Error vs Speed 30 Pixel Lookahead



Average Error vs Control Work
10 Pixel Lookahead



Conclusion

- Simple control algorithm for visual servoing of mobile robots
- Quick approximation of offset and gradient
- Short sighted robots: offset much more important
- New controller using Fuzzy Logic
- Better extraction of the path