Fiducial Markers for Robotic Vision, Navigation and Object Recognition

Mike Thompson

November 18, 2009

Goals For Tonight

From the perspective of a hobbyist:

Demonstrate using OpenCV and other tools to decode fiducial markers for practical and robust robotic vision

Inspire others to add vision to their robots

The Problem

- At Runbot building small biped robots intended to have "The Sims" type artificial intelligence for game play and entertainment
- How does the robot interact with environment?
  - What objects are around robot for interaction?
  - Where is location of robot in environment?
- Ideally something simple and robust so we could focus on other aspects of the robot

Shape recognition?

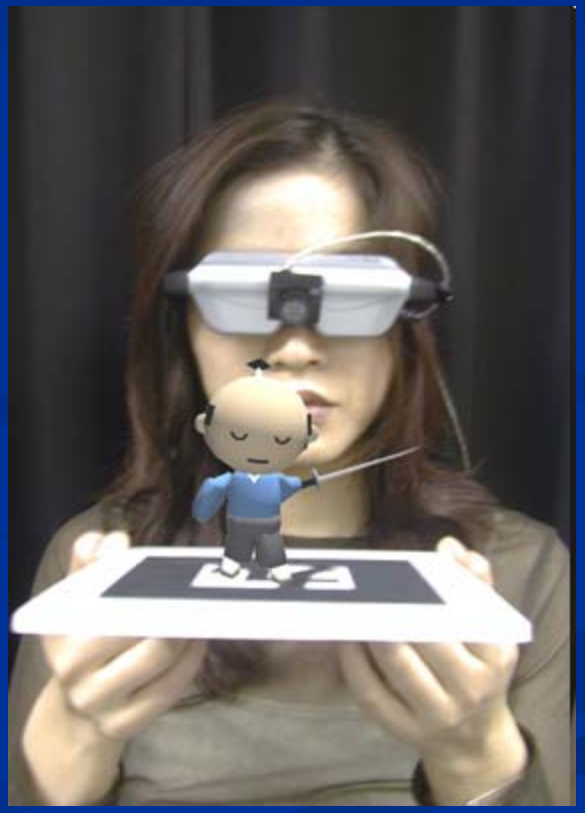
- Hard to do, particularly with low power CPUs
- Probabilistic with recognition a fuzzy probability rather than binary (black/white) data
- Needed another solution that would be quick & easy to develop, cheap to implement, worked well enough to proceed with other development

My Quick & Dirty Solution

- Borrow inspiration from Augmented Reality research – in particular how real world and virtual world camera views are merged
- Specifically, borrow fiducial marker technology
- Large amount of research and published papers within the AR community

Fiducial Marker Definition

An object in the field of view of an imaging system which is used as a point of reference or measure

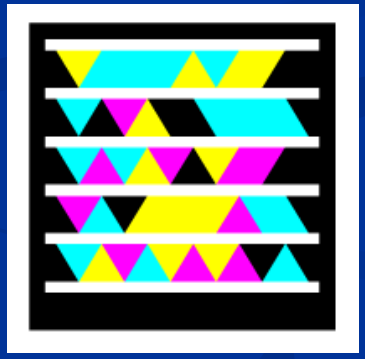


2D Barcodes

Fiducial markers share many characteristics with 2D Barcodes you may have seen.

QR Code

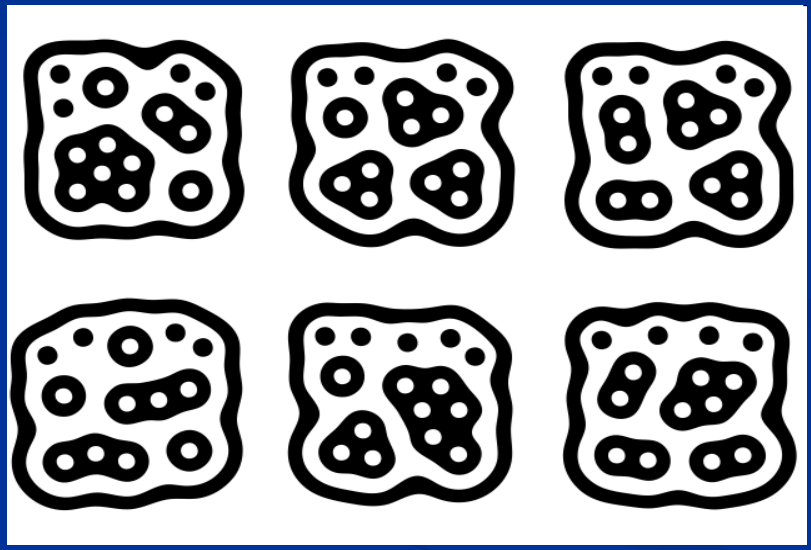
Aztec



Microsoft Tag

Fiducial Marker Example

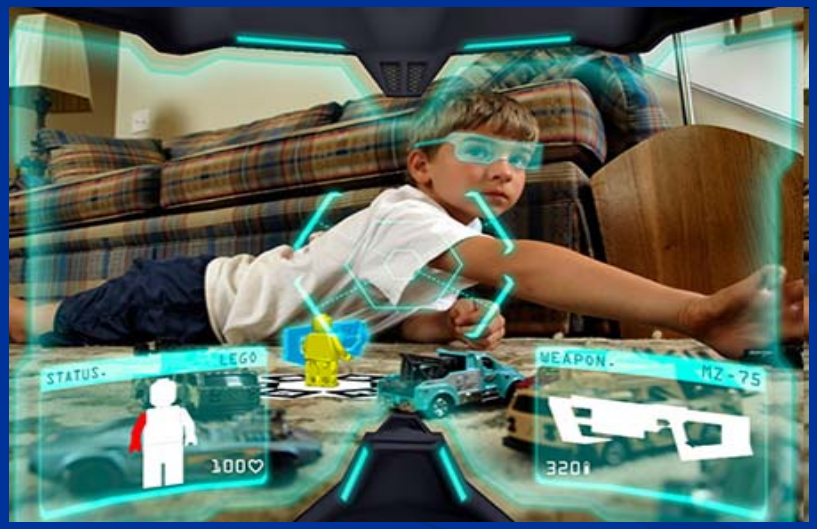
Reactable Music Instrument



Musicians controls the system by manipulating tangible objects tagged with fiducial markers

Fiducial Marker Example

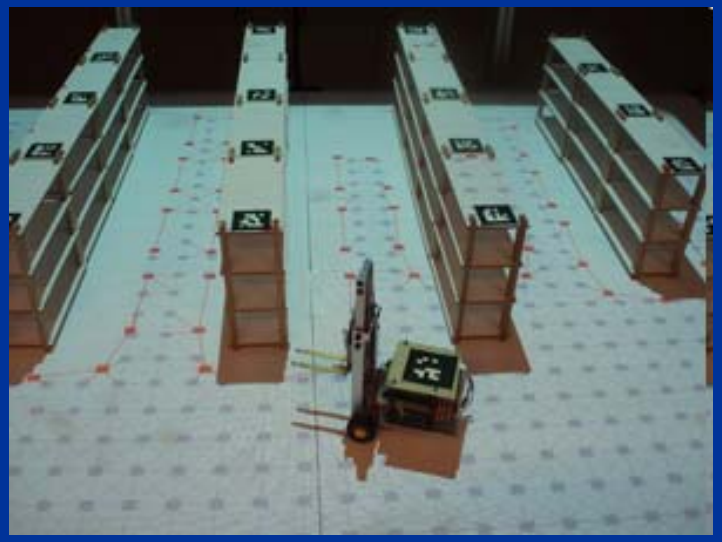
Augmented Reality Toy by Frantz Lasorne

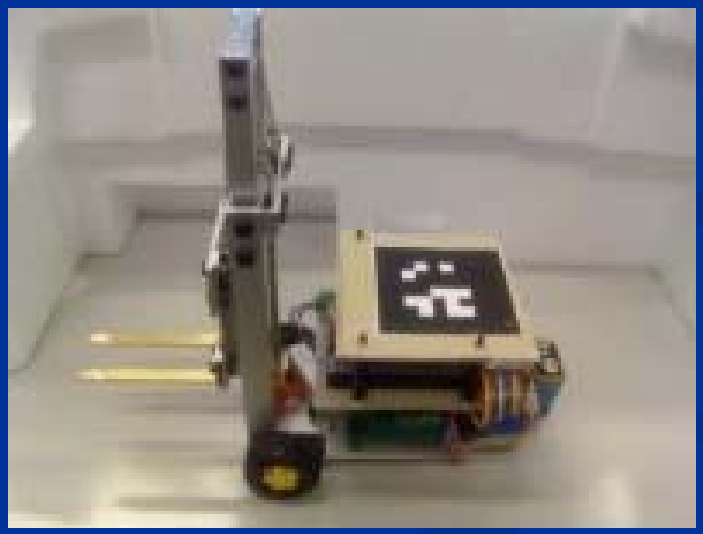


Game that mixes both virtual and physical reality

Fiducial Marker Example

Robotic Industrial Simulator for PhD Thesis
by Guillaume Zufferey





Small industrial robot tracked by ARTag markers

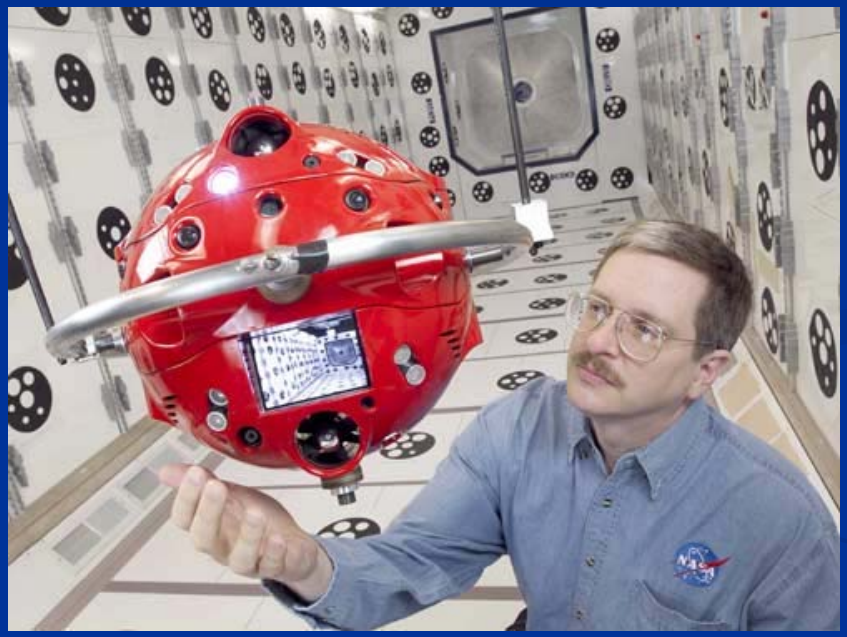
Fiducial Marker Example

Sony AIBO with Charging Station

Fiducial markers help the robot dog find and mount the station for charging

Fiducial Marker Example

NASA Personal Satellite Assistants (PSA)

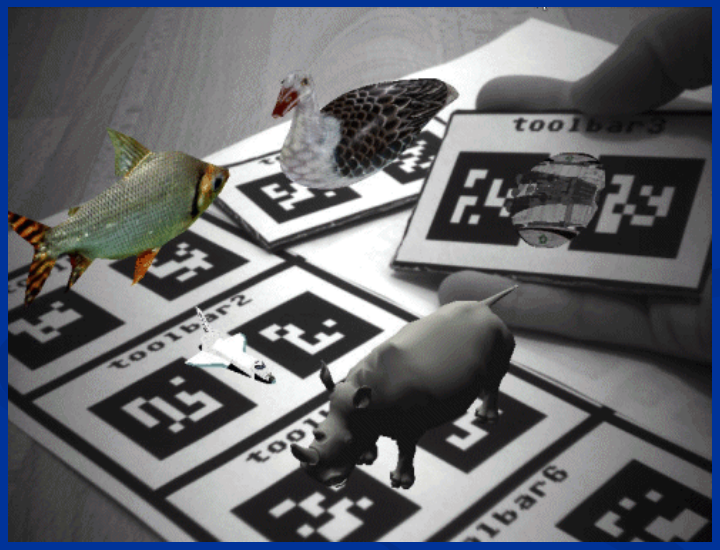


PSA in MicroGravity Test Facility with fiducial based vision system

My Inspiration

ARTag Revision 1.

A Fiducial Marker System Using Digital Techniques



Mark Fiala

November 2004

Published by National Research Council of Canada

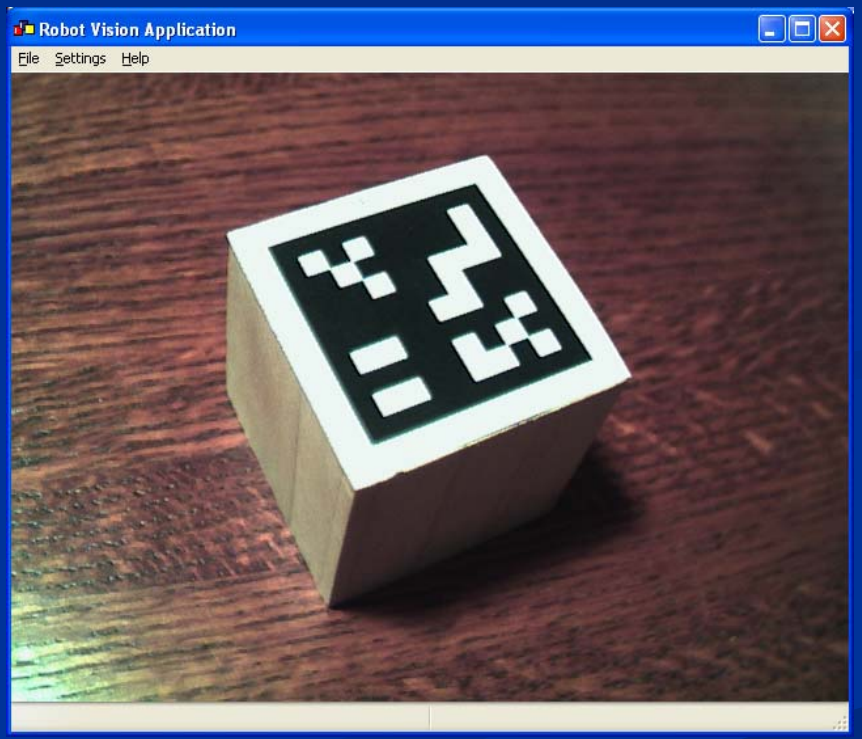
http://www.artag.net/

Fiducial Tag Recognition

- 12 easy steps
- 80% OpenCV with some added sugar
- OpenCV 1.0 APIs for all image, matrix and vector handling
- Added code for user interface, camera interface, tag decoding, cyclic redundancy checks, forward error correction, etc…

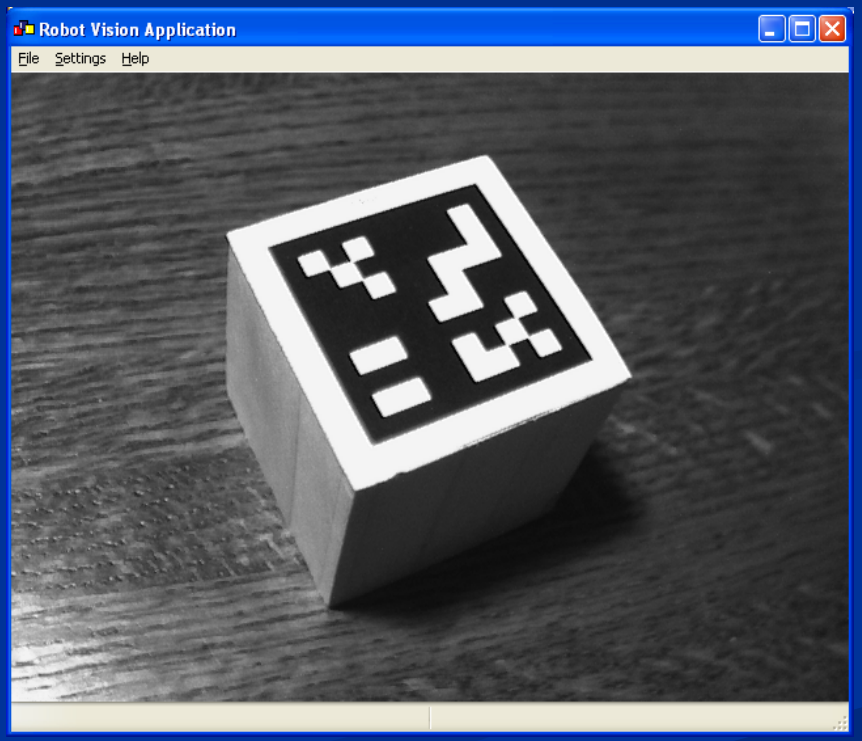
Fiducial Tag Recognition

- RGB image from the camera
- In this case using DirectX DirectShow APIs to obtain image from webcam (what a pain in the @%#$#)



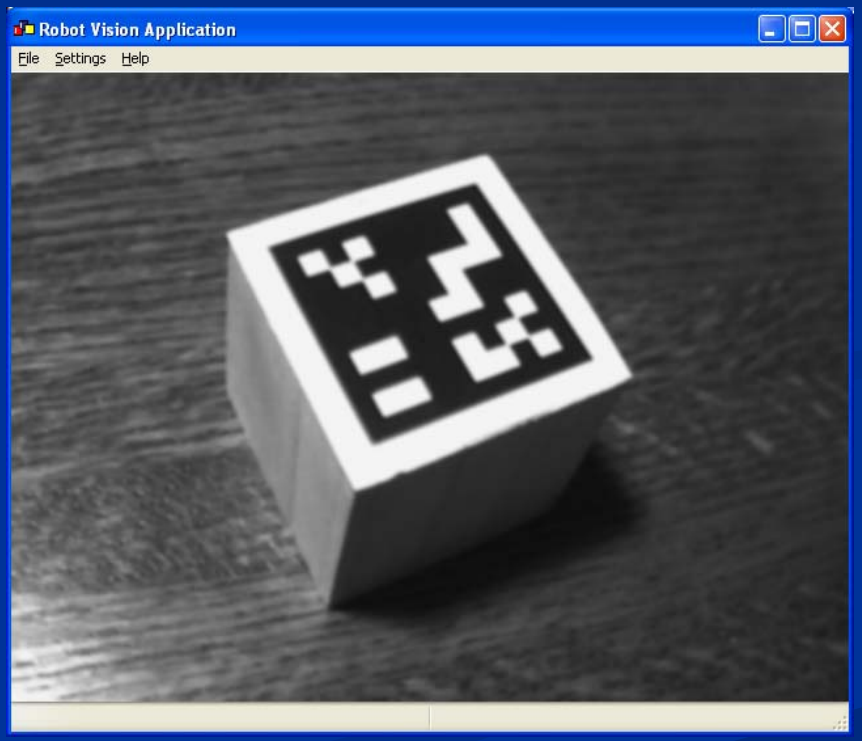
Fiducial Tag Recognition

- Convert color image to gray scale image
- Needed for edge detection
- cvCvtColor( ) with CV_RGB2GRAY



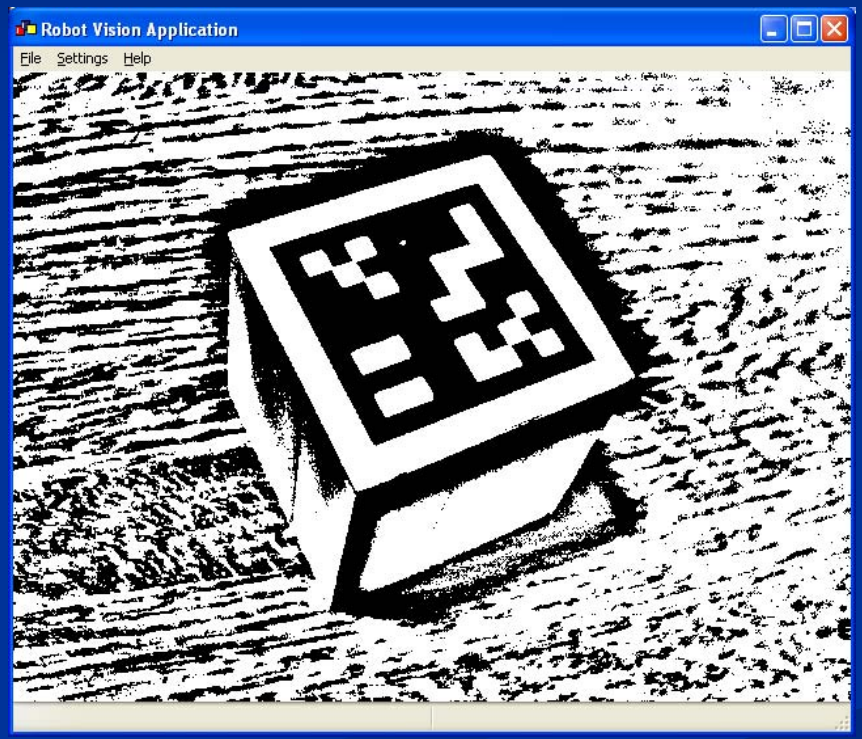
Fiducial Tag Recognition

- Apply Gaussian blur
- Optional – helps eliminate noise within some edge detection algorithms
- cvSmooth( ) with CV_GAUSSIAN



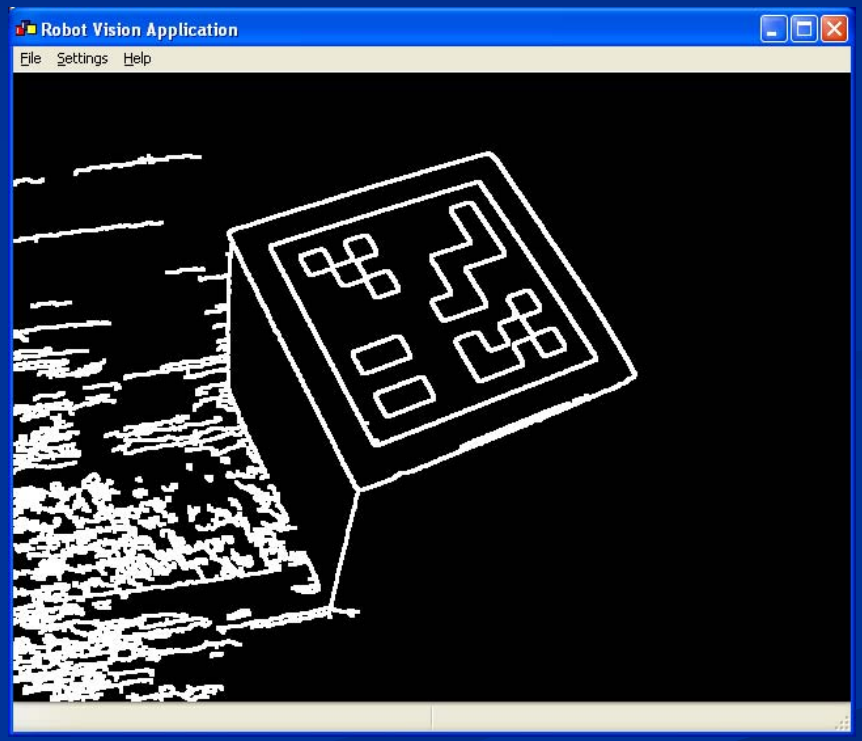
Fiducial Tag Recognition

- Adaptive threshold edge detection to create binary image with edges
- cvAdaptiveThreshold( )
- GAUSSIAN option seems to work better than MEAN option



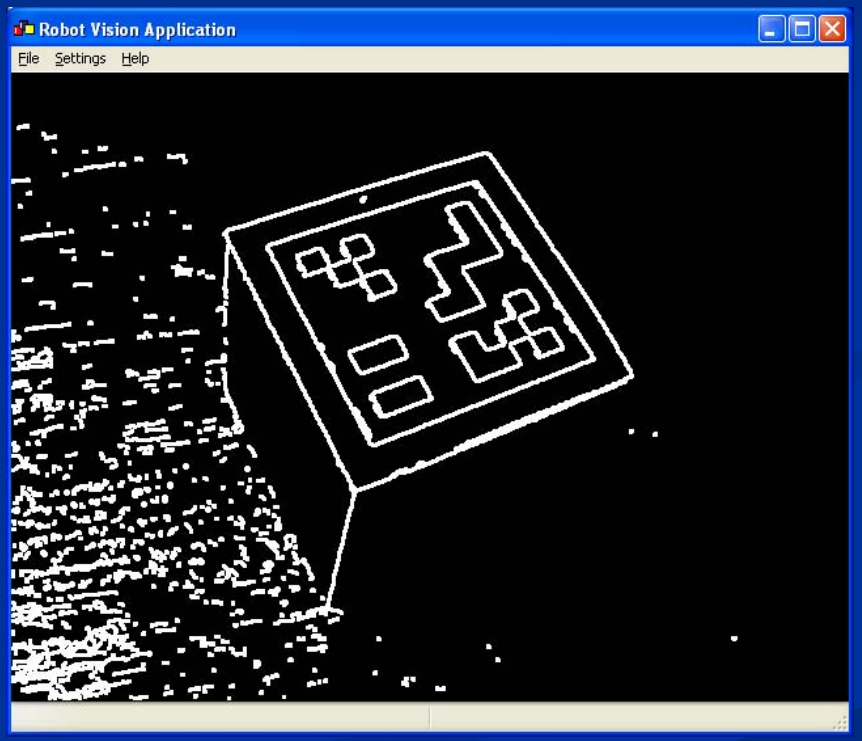
Fiducial Tag Recognition

- Alternate Canny edge detection algorithm
- cvCanny( ) with cvDilate( ) to fill any gaps



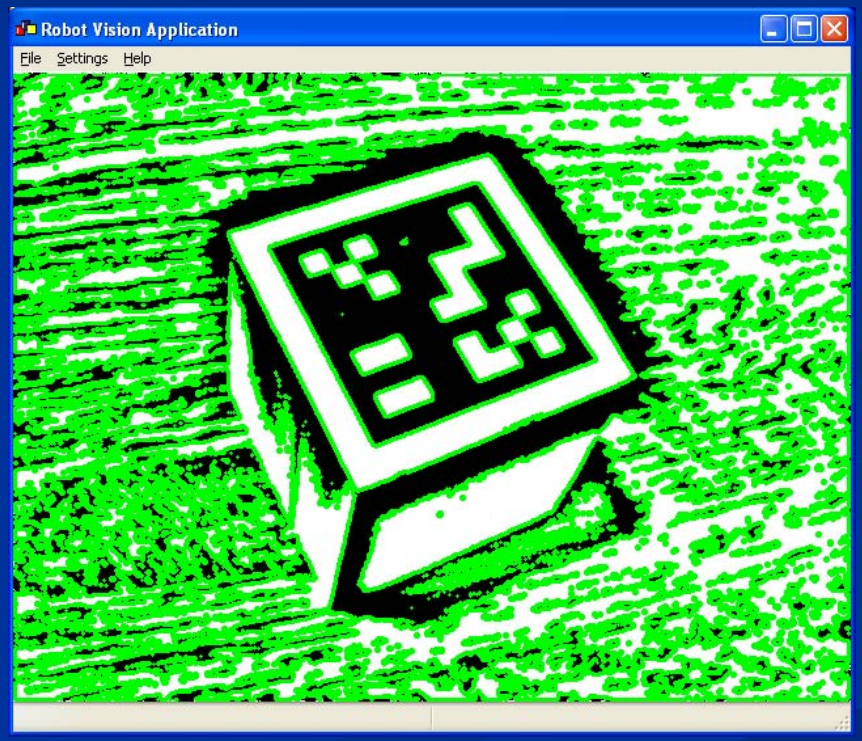
Fiducial Tag Recognition

- Alternate Susan edge detection algorithm
- cvCanny( ) with cvDilate( ) to fill any gaps



Fiducial Tag Recognition

- Convert edges to contour data structures – edge between dark and light as a sequence of line segments
- cvFindContours( )

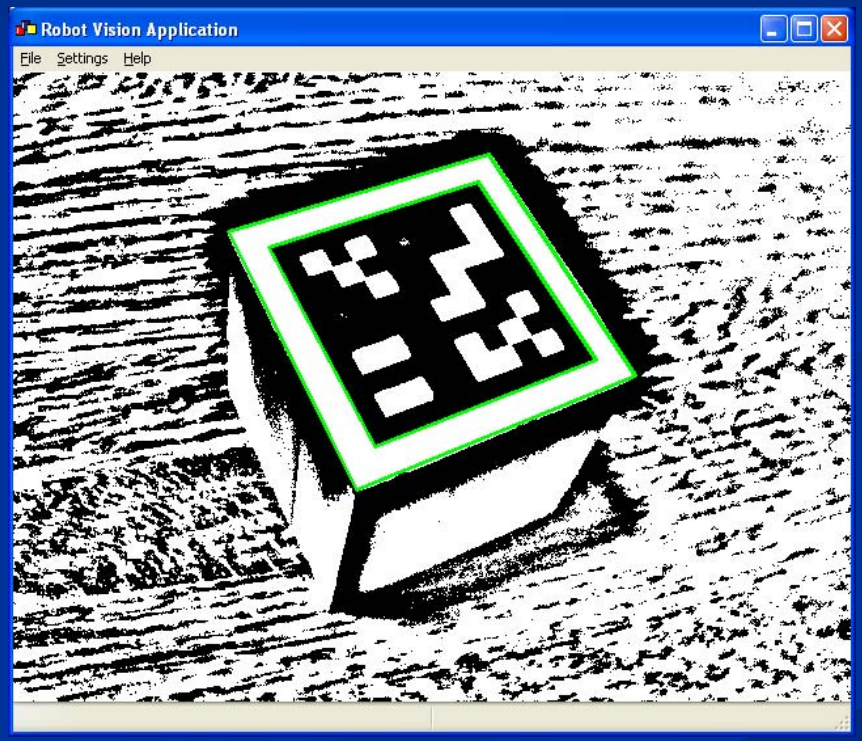


Fiducial Tag Recognition

- Simplify complex contours into a sequence of multi-sided polygon contours
- cvApproxPoly( )

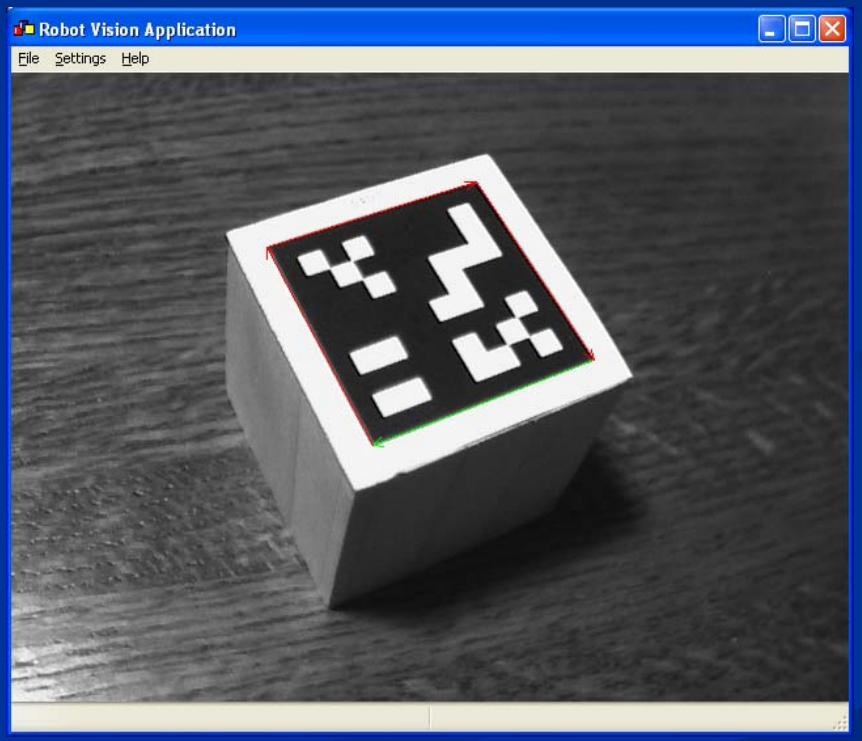
Fiducial Tag Recognition

- Identify convex polygons that have four sides and are not too small – all others ignored
- cvCheckContourConvexity( ) with cvContourArea ( )



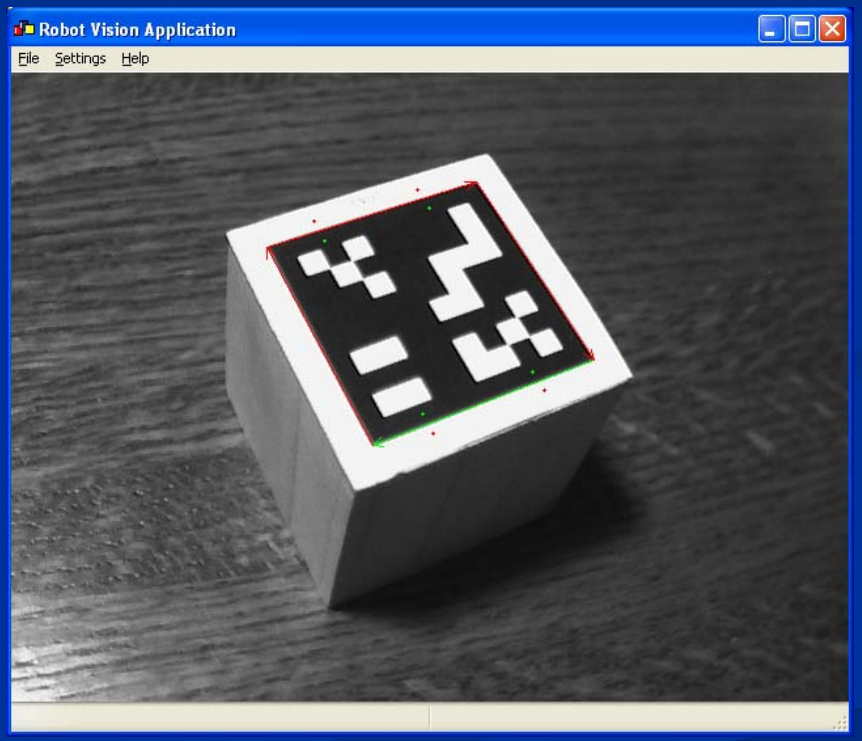
Fiducial Tag Recognition

- Apply corner detection to the corners of the remaining polygons
- Increase precision to match shape
- cvFindCornerSubPix( )



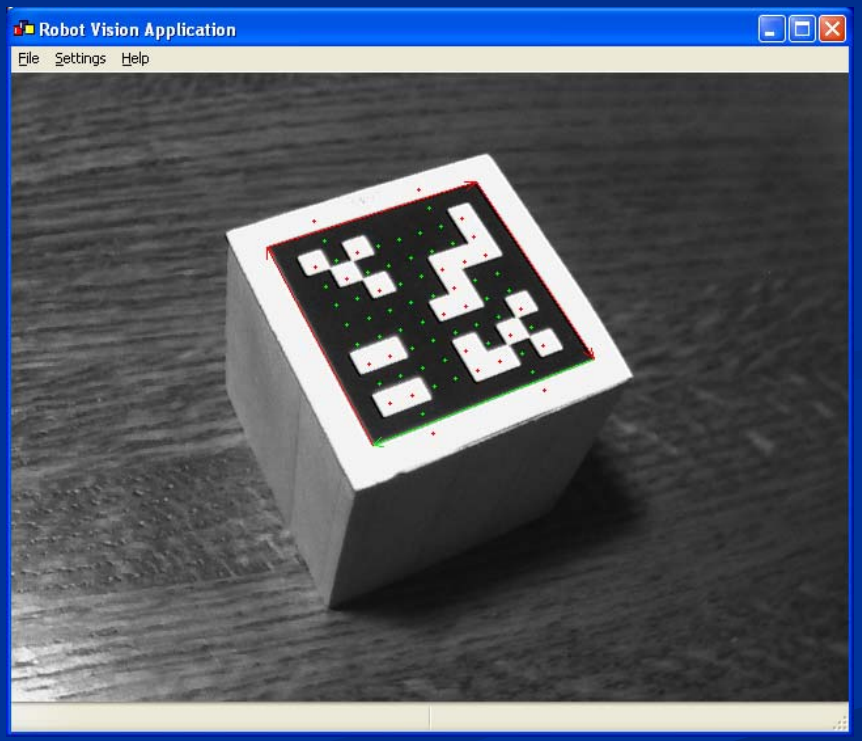
Fiducial Tag Recognition

- Normalize polygon to always flow in clockwise direction
- Obtain reference samples for white and black white areas within the polygon
- Dereference pixel data directly from gray image



Fiducial Tag Recognition

- Determine sample points within the polygon and sample the pixels values to determine 0's and 1's
- Dereference pixel data directly from gray image
- Four possible coding sequences – one for each edge

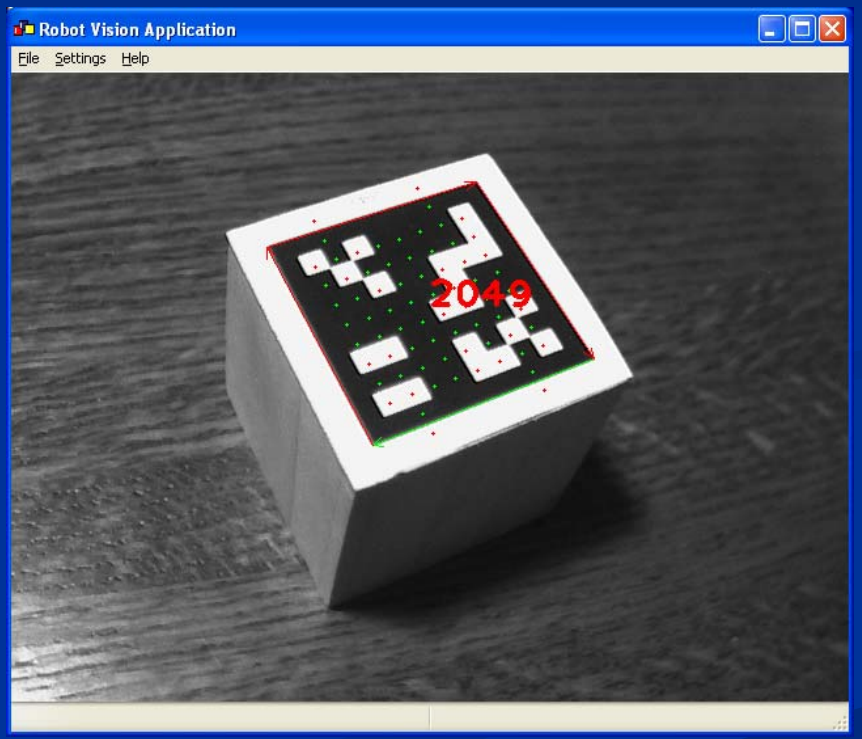


Fiducial Tag Recognition

- Bits encode the following:
  - 16 bit identifier as data payload
  - 16 bit CCITT Cyclic Redundancy Check (CRC) adds uniqueness to 16 bit data payload
  - 32 bits Reed-Solomon Forward Error Correction with 8 bit symbol size, 4 data symbols and 4 parity symbols – up to two bits recovered per tag
- Goal is to eliminate potential for false positives

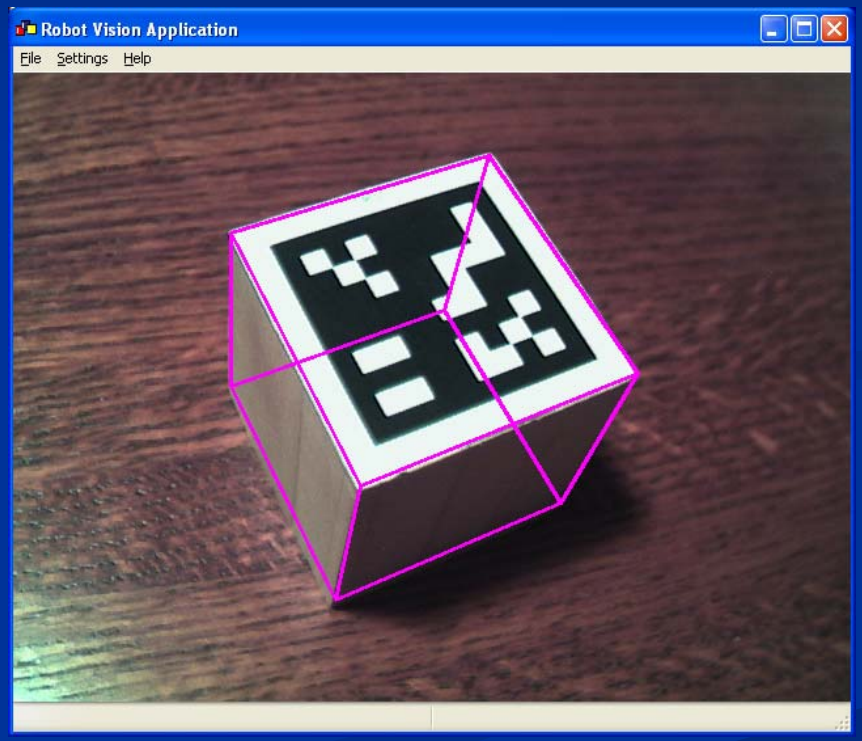
Fiducial Tag Recognition

- Did we obtain valid tag identifier?
- Categorize tag identifier to navigation, object or character tag



Fiducial Tag Recognition

- Handle tag based on tag identifier type
- In this case, it is an object tag so re-project 3D coordinates of the object onto 2D image



3D Positioning

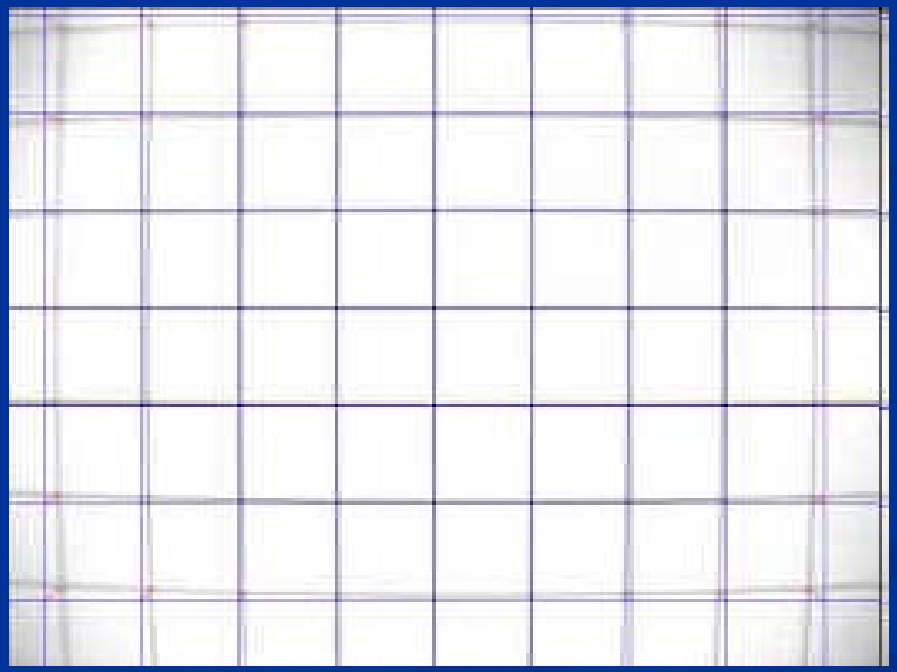
- Magic of cvFindExtrinsicCameraParams2( )
- Estimates extrinsic camera parameters using known intrinsic parameters and extrinsic parameters for each view
- Determines 3D camera position from a set of known 2D positions of 3D objects within an image and a description of the camera lens distortion
- Object position is inverse of camera position

3D Positioning

- Intrinsic camera parameters
  - Describes distortion imposed an idealized “pin hole” camera view by the lens of the camera – radial and tangential
  - Independent of scene viewed – thus “intrinsic” to camera
- Extrinsic camera parameters
  - Describes translation/rotational position of camera
  - Dependent upon scene viewed – thus “extrinsic” to camera
- Extrinsic object parameters
  - 2D positions of objects in image
  - 3D positions of objects in space

3D Positioning

- Example of intrinsic "barrel" distortion
- Radial and translational distortion can be seen

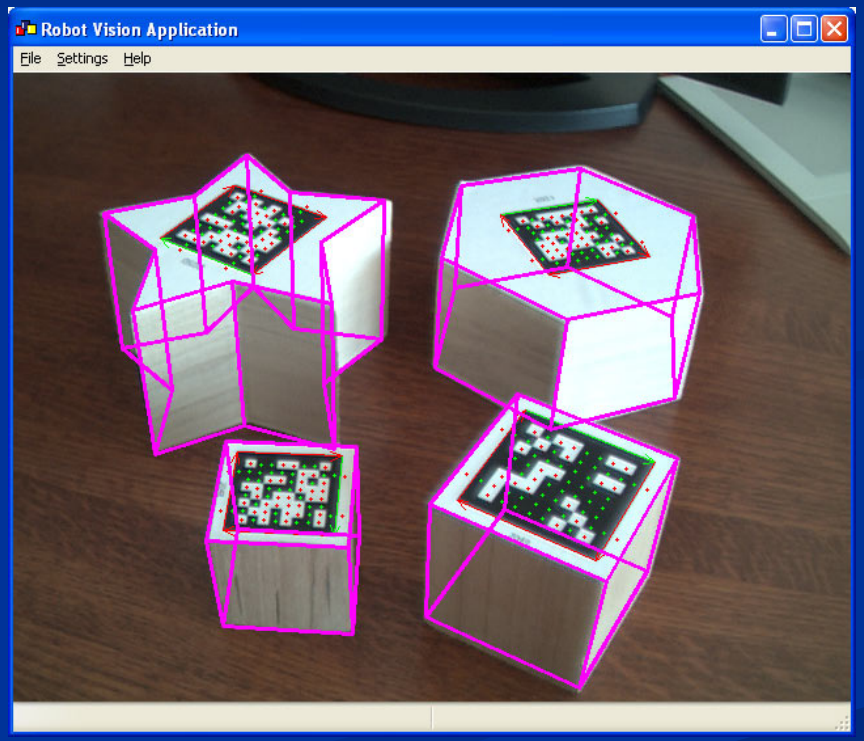


3D Positioning

- From camera intrinsic parameters and view extrinsic parameters we can determine camera extrinsic parameters
- From camera extrinsic parameters and view extrinsic parameters we can determine camera intrinsic parameters
  - Known as camera “calibration”
  - cvCalibrateCamera2( )

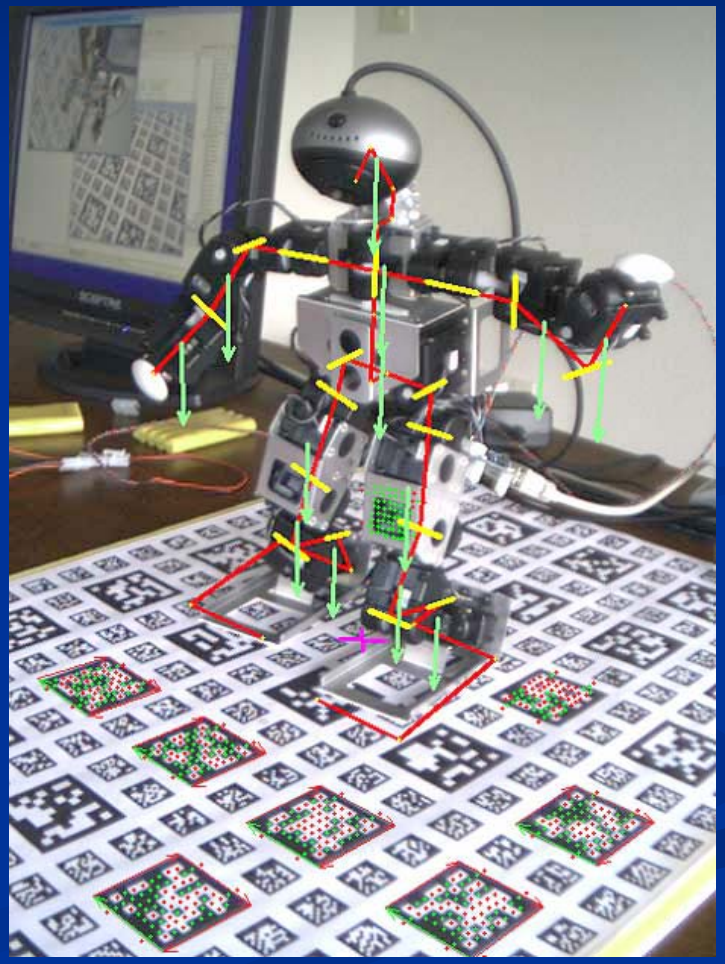
3D Shape Positioning

- Inverse of external camera intrinsic positions each tag in 3D space relative to camera
- Tag identifier used to look up 3D object points
- cvProjectPoints2( ) projects 3D points to 2D points in image



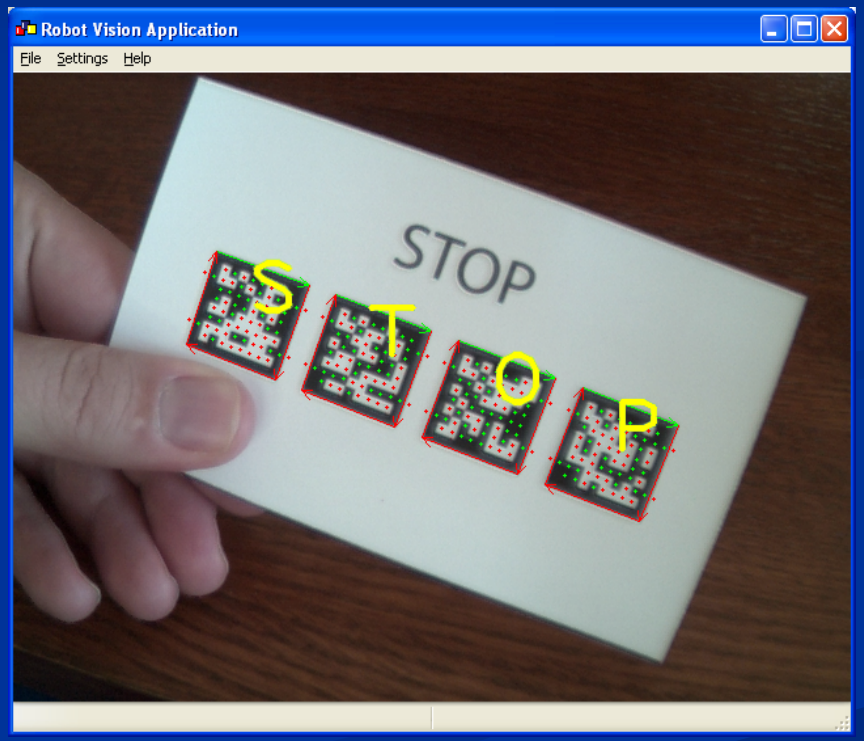
3D Robot Localization

- External camera intrinsic positions camera 3D space relative to tags
- Inverse kinematics used to position rest of robot in 3D space
- Special test rig with two cameras – robots has camera and external camera used to view robots kinematic model



Written Communication

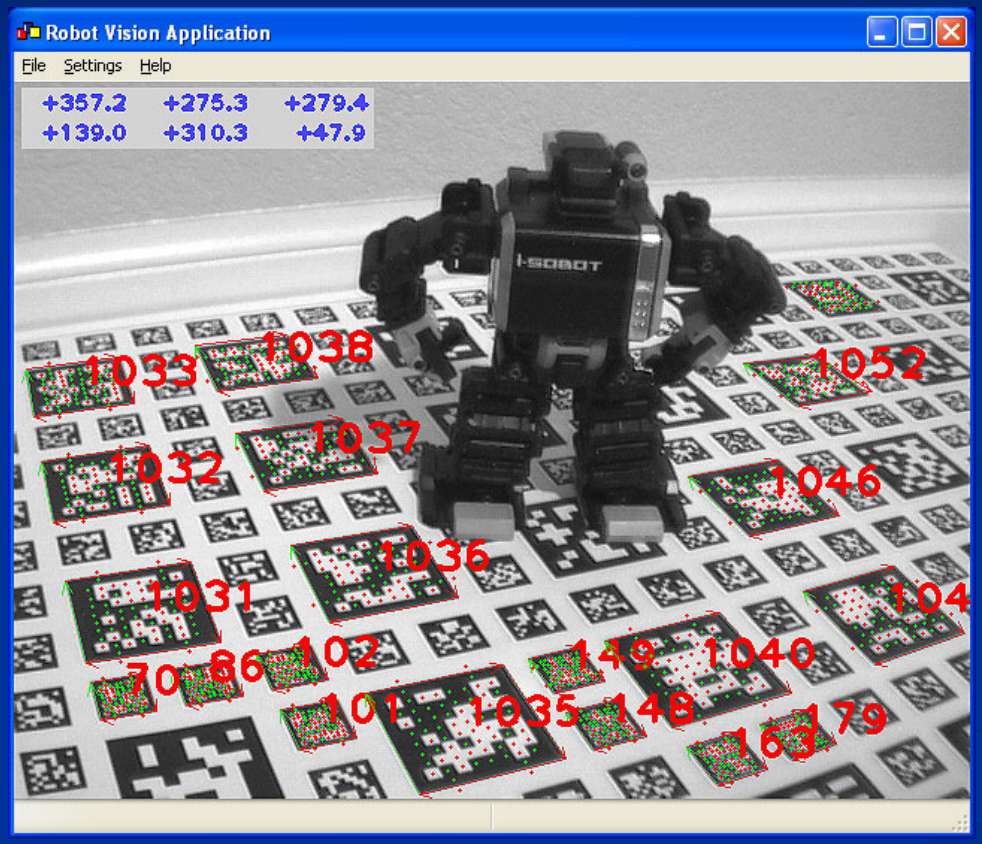
- Tags simply mapped to ASCII characters
- Multi-tag messages and signs a robot can easily understand
- 3D can enhance information such as adding directional information



Review

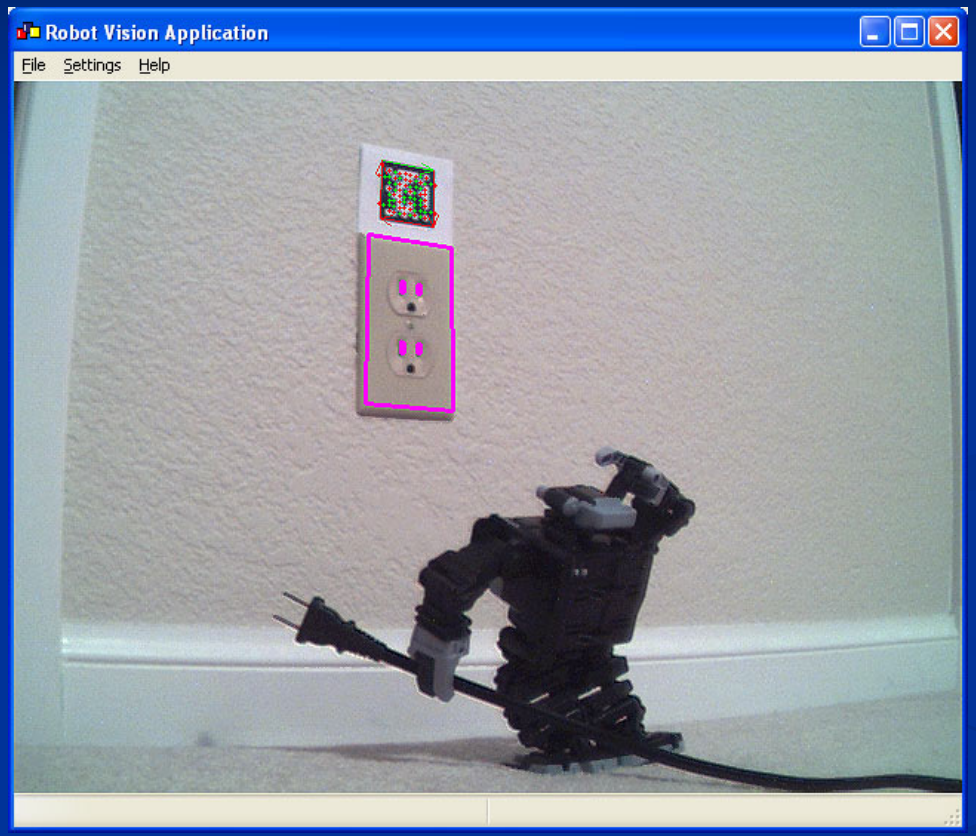
- Fiducial markers are applicable to many robotic navigation, recognition and communication problems
- Provide accurate, high-fidelity 3D information
- Relatively lean computational resources
- Relatively easy to implement
- OpenCV gives you 80% of what you need
- Only a single camera needed

Robot 3D Localization



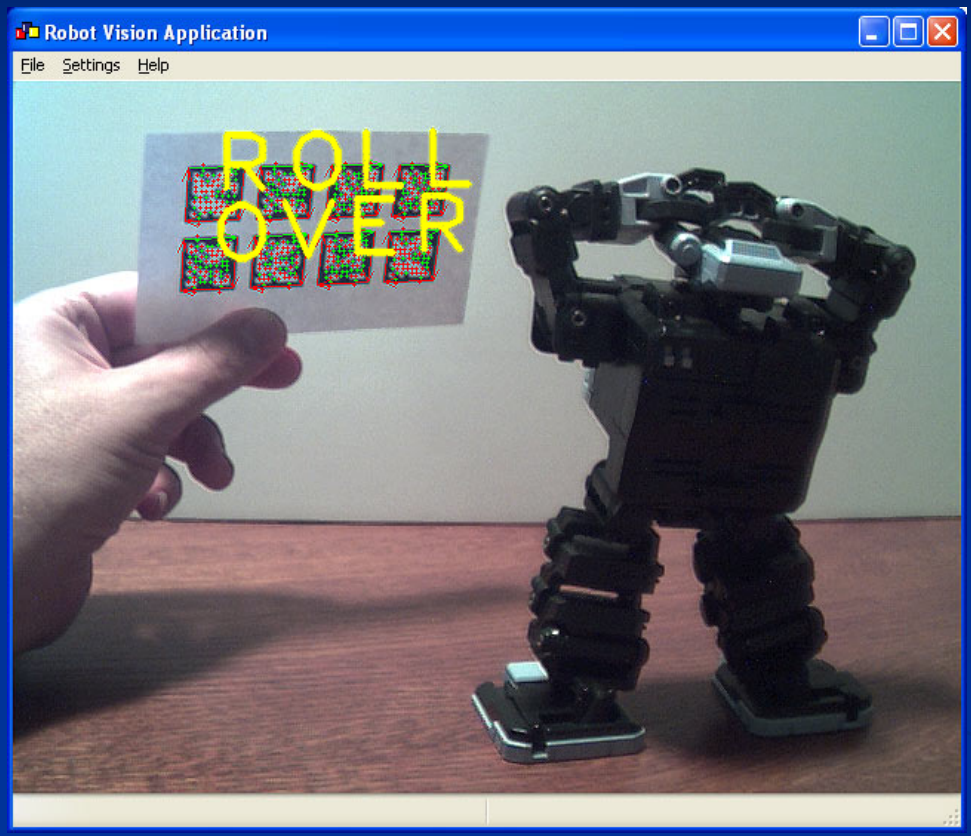
Where is the robot in the environment?

Objects In 3D Space



What objects are around the robot and how can the robot manipulate them?

Communication



New way for you and the environment
to communicate with the robot

Demo

Future Enhancements

- More organic looking tags to better blend into different environments
- Alternate non-geometric encoding schemes

Cantag

Reactable

Microsoft

Future Development

- More organic looking and aesthetically pleasing tags to better blend into different environments
- Utilize non-geometric encoding schemes

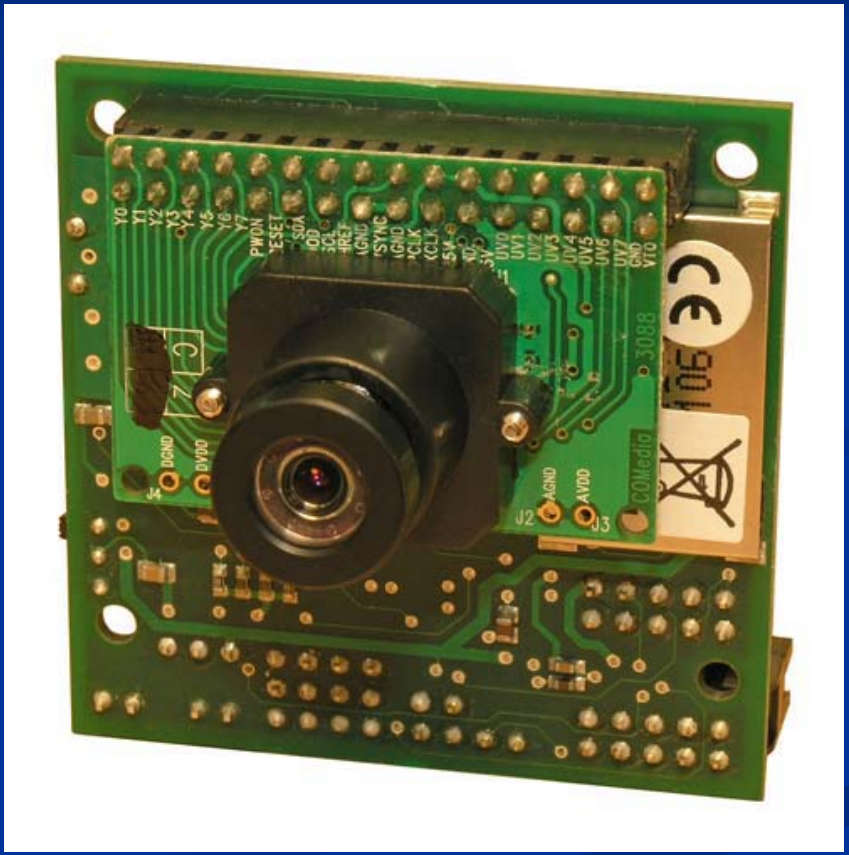
Cantag

Reactable

Microsoft

Future Development

- Integrate into standalone hardware unit similar to CMUCam or AVRCam
- Allow for easy integration with hobby level and educational robotics
- May not really be needed now that hardware like BeagleBoard exist



Resources

- 3 years ago I had to roll my own solution, but now many other libraries are out that potentially save a lot of work
- **ARToolKit**
  http://www.hitl.washington.edu/artoolkit/
- **Studierstube Tracker**
  http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php
- **Cantag Machine Vision Framework**
  http://www.cl.cam.ac.uk/~acr31/cantag/
- **reacTIVision Computer Vision Framework**
  http://reactivision.sourceforge.net/

Thank you for watching and listening