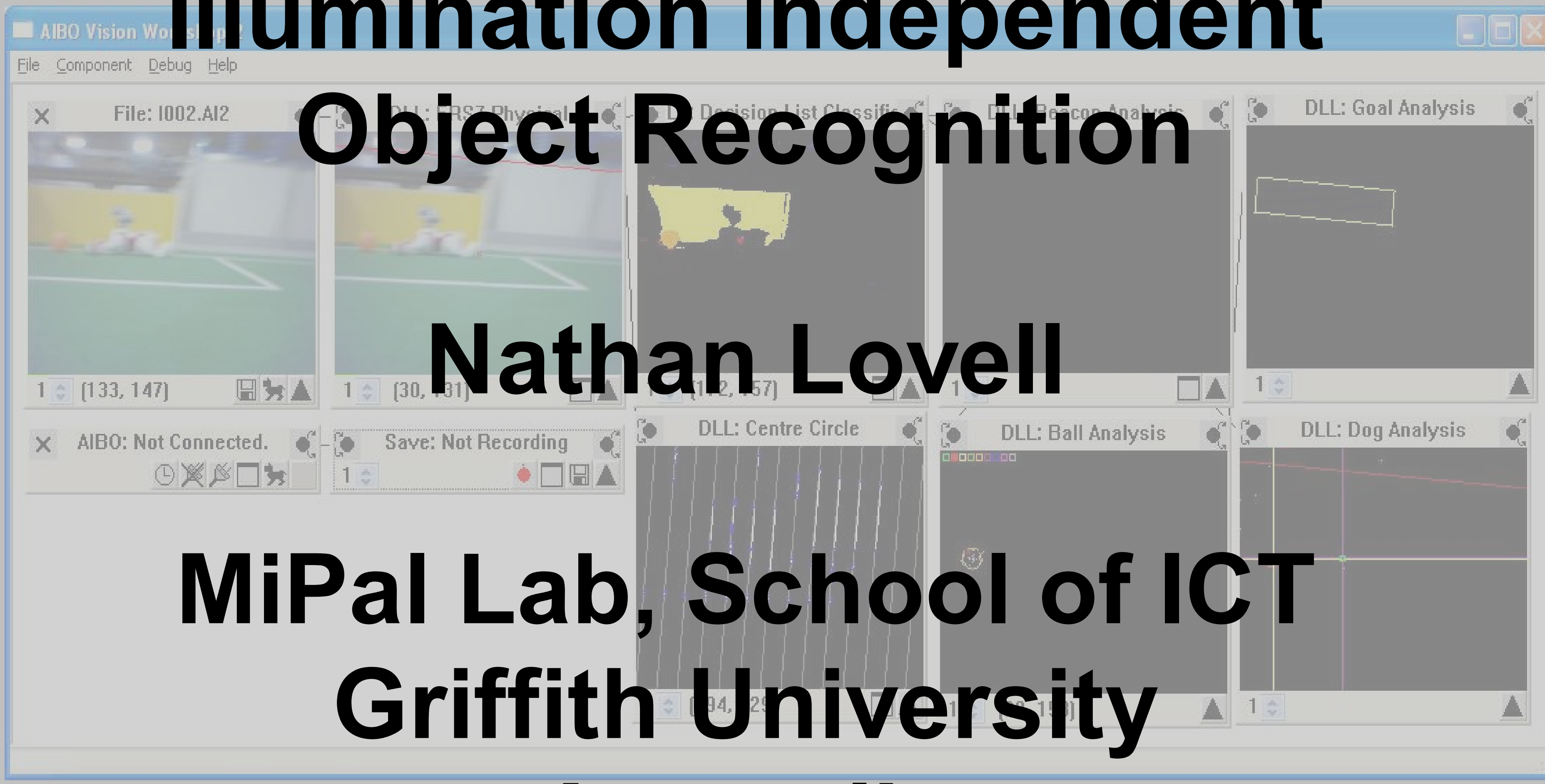
Illumination Independent Object Recognition



Nathan Lovell

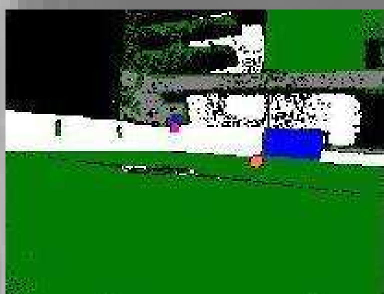
MiPal Lab, School of ICT
Griffith University
Australia

Object Recognition Systems

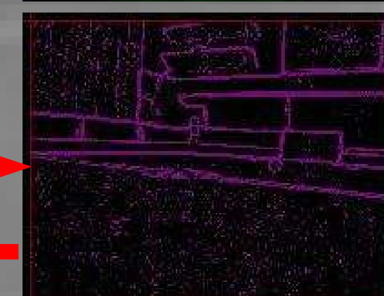
Old-style Object Recognition

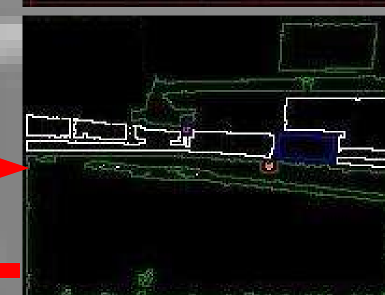
Our New Object Recognition

Classification



Sparse Classification





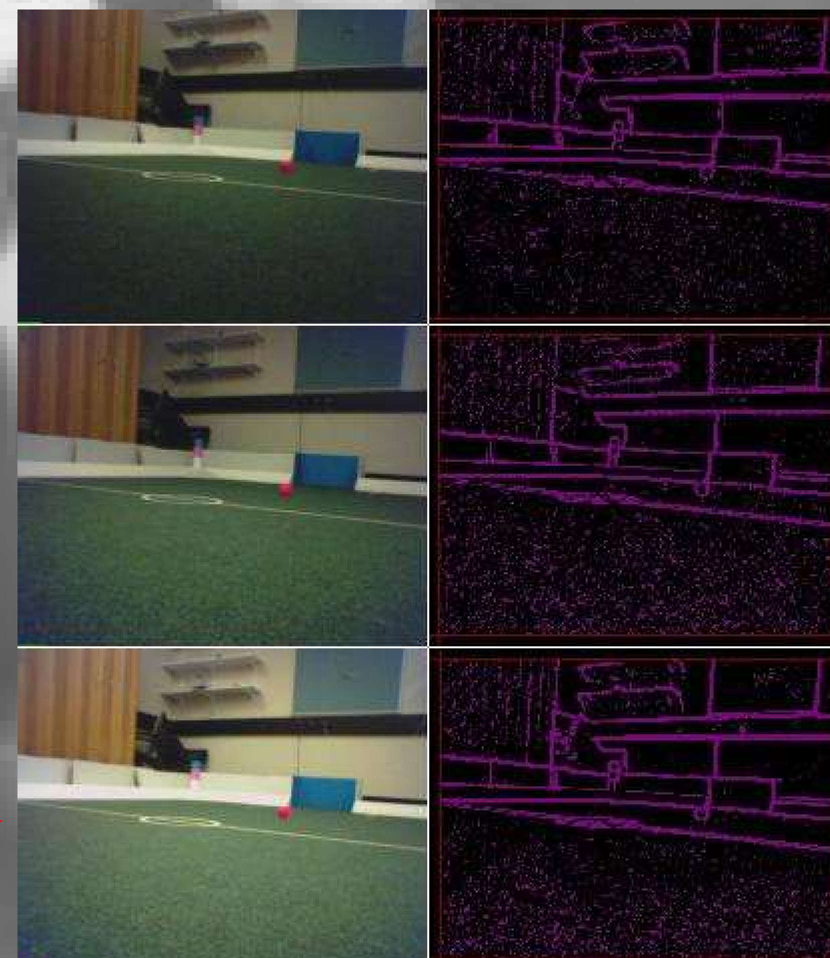
Sparse classification combined with edge detection gives the same results

Blobs are formed by grouping classified pixels, and objects recognised from blobs

Why Edge Detection?

- Classifiers are usually over-fitted to one particular lighting condition
- Edge detection is fairly robust to changing illumination conditions
- A fast edge detector is required (see paper)

250-430 LUX
450-700 LUX
700-900 LUX

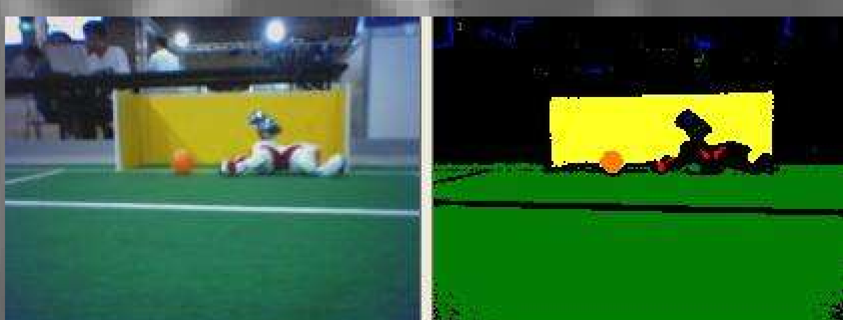


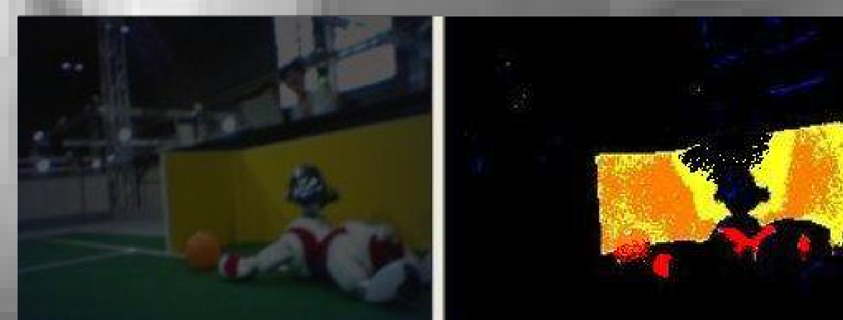
Classification

- Given:
  - A set of images *I*
  - A function *P(n, i)* that gives the *(y,u,v)* tuple of pixel *n* in image $i \in I$
  - A function *R(y,u,v)* that gives the *real* class, C_{cls}, of a *(y,u,v)* tuple (done by humans)
  - A function *C(y,u,v)* that gives the colour class, C_{cls}, of a *(y,u,v)* tuple
- Standard classification task:
  - Optimise *C* so that it will assign class C_{cls} to each tuple in $\bigcup_{l=1}^{|I|}\bigcup_{p=1}^{|i|} P(n,i)$ where $R(P(n,i)) = C_{cls}$

Classification

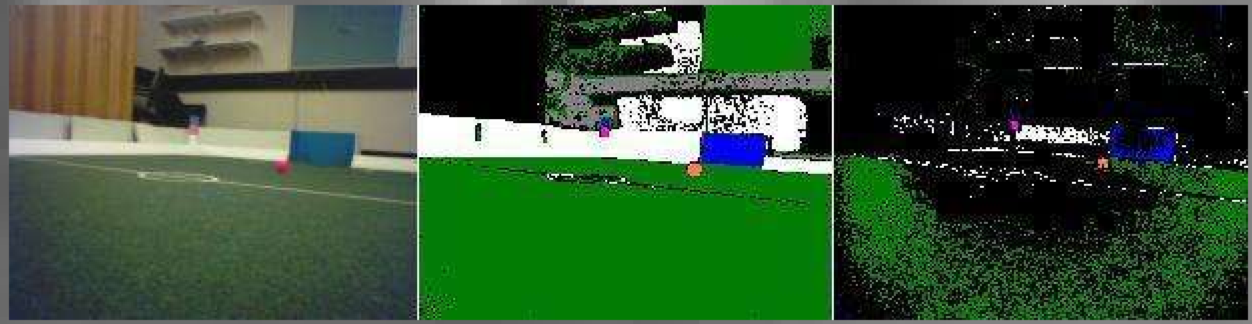
- The problem: the function, *C*, depends on the lighting conditions
- E.g.: In low light the orange ball looks more “red” and the yellow goal looks more “orange”
  - Running a classifier trained in good light, in low light conditions, will often result in the ball being classified as red and the goal as orange





Sparse Classification

- Sparse classification task:
  - Optimise C so that it will assign class *C* to each value in $\bigcap_{l=1}^{|I|}\bigcap_{p=1}^{|i|} P(n,i)$ where $R(P(n,i)) = C_{cls}$
- Take sample images in a wide variety of lighting conditions and learn the intersection (as opposed to union) of all (e.g.) orange pixels
- *We only want to assign a tuple to a class if it belongs to that class in a wide variety of different lighting conditions*

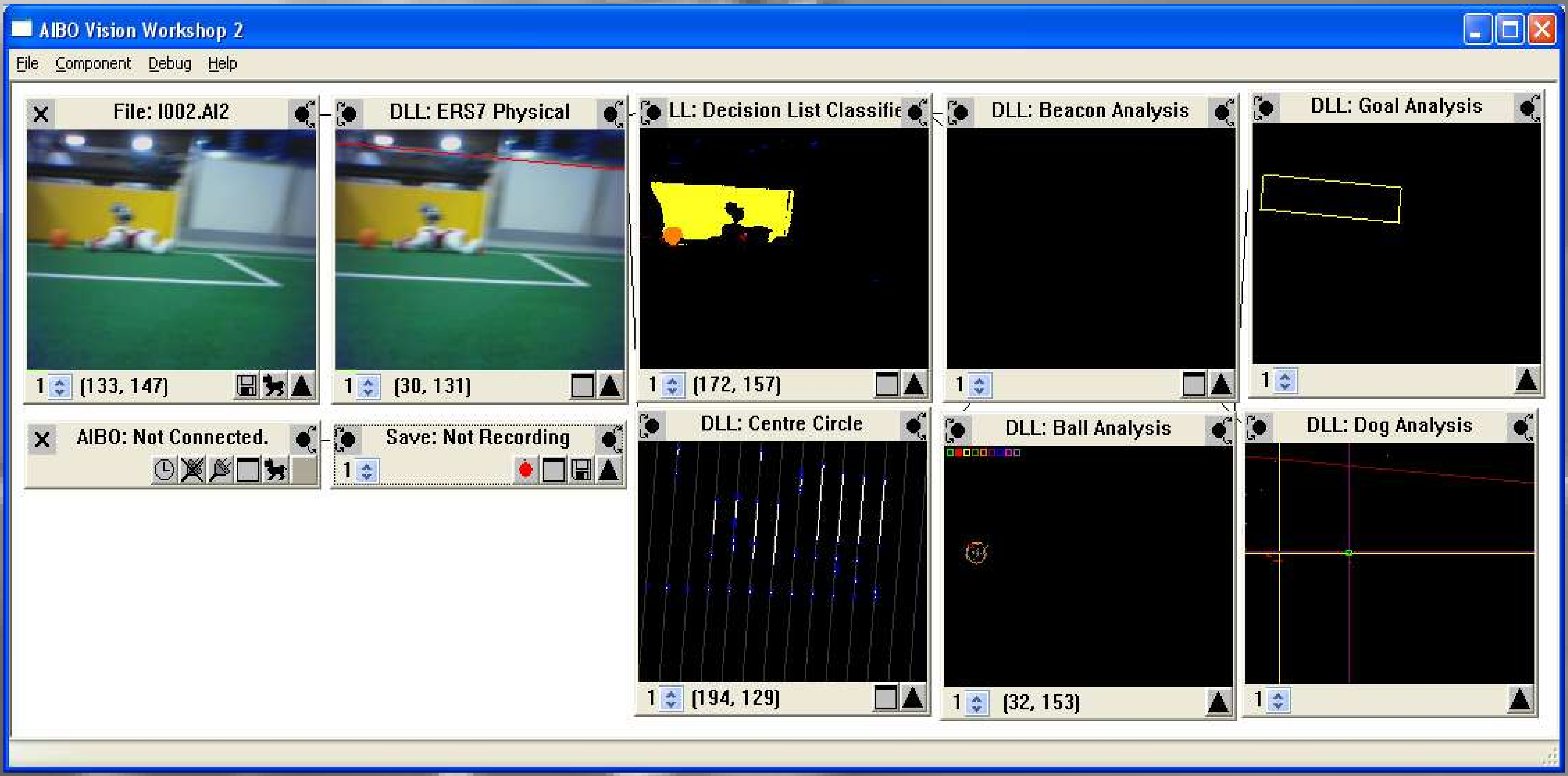


Advantages

- Colour calibration takes less than 10 minutes and can be performed by hand
  - Usually the calibration file will not need to change
- Our entire classifier calibration is very small (<32K)
  - This years lighting challenge colour calibration:

```
1 60 255 76 106 159 195
4 0 255 0 101 137 255
3 0 255 160 255 0 100
6 0 255 105 130 180 255
5 0 255 121 255 166 255
0 0 255 0 255 0 255
```

AVW2 Demo

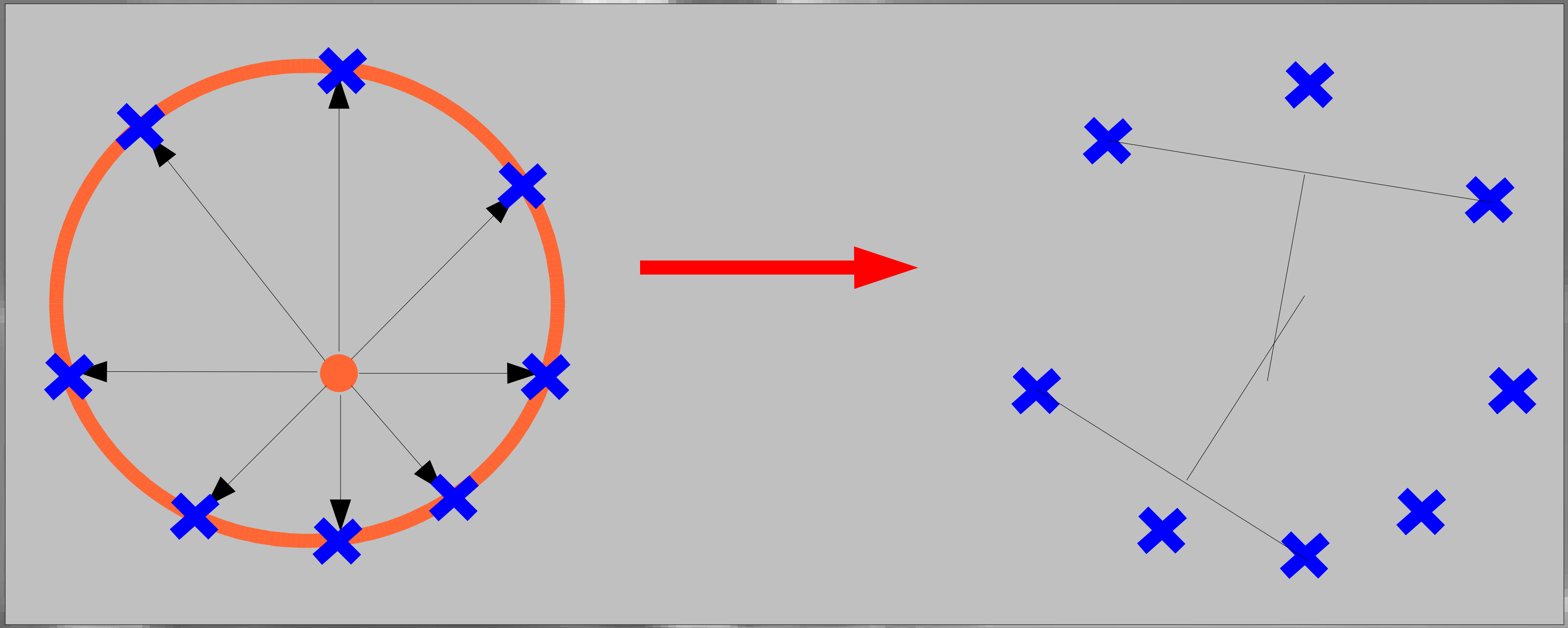

AIBO Vision Workshop 2
File Component Debug Help
File: I002.AI2
DLL: ERS7 Physical
LL: Decision List Classifie
DLL: Beacon Analysis
DLL: Goal Analysis
(133, 147)
(30, 131)
(172, 157)
AIBO: Not Connected.
Save: Not Recording
DLL: Centre Circle
DLL: Ball Analysis
DLL: Dog Analysis
(194, 129)
(32, 153)

Object Recognition

- After sparse classification, most pixels will remain unclassified
- Cluster the classified pixels
  - We only need to remember one representative pixel in each cluster so this is very fast
- Depending on the level of detail required, the next step varies

Simple Object Recognition

- For simple shapes (such as circles, squares, rectangles etc)



- Requires no separate edge-detection step
  - Edge detection is performed along each individual projection from the seed

Griffith UNIVERSITY

Complex Objects

- Our method is easily extended when the object has a complex shape
  - Object appears different depending on the viewing angle (e.g. AIBO's uniform)
- Edges detection is performed as a pre-processing step
- A border following algorithm finds the outline of each image segment

Performance

	Average Time (ms) on AIBO	
	Simple Objects	Complex Objects
Classification	0.91	0.91
Pixel Clustering	2.77	2.77
Edge Detection	N/A	14.86
Border Following	N/A	5.21
Object Recognition	2.12	7.91
Total time	5.82	31.69

There are approximately 33ms available per frame on an AIBO ERS-7 (30fps)

Our new vision system is actually 34% faster than the old one (avg 8.9ms per frame)

Video