



# Computer Vision for Microscopes

**Microsystem & Machine Vision Laboratory**

16th Dec 2005



MMVL <http://vision.eng.shu.ac.uk/mmvl/> (vision/www)

MRC [http://www.shu.ac.uk/research/meri/modelling\\_rc.html](http://www.shu.ac.uk/research/meri/modelling_rc.html)

MERI <http://www.shu.ac.uk/research/meri/>

SHU <http://www.shu.ac.uk/>

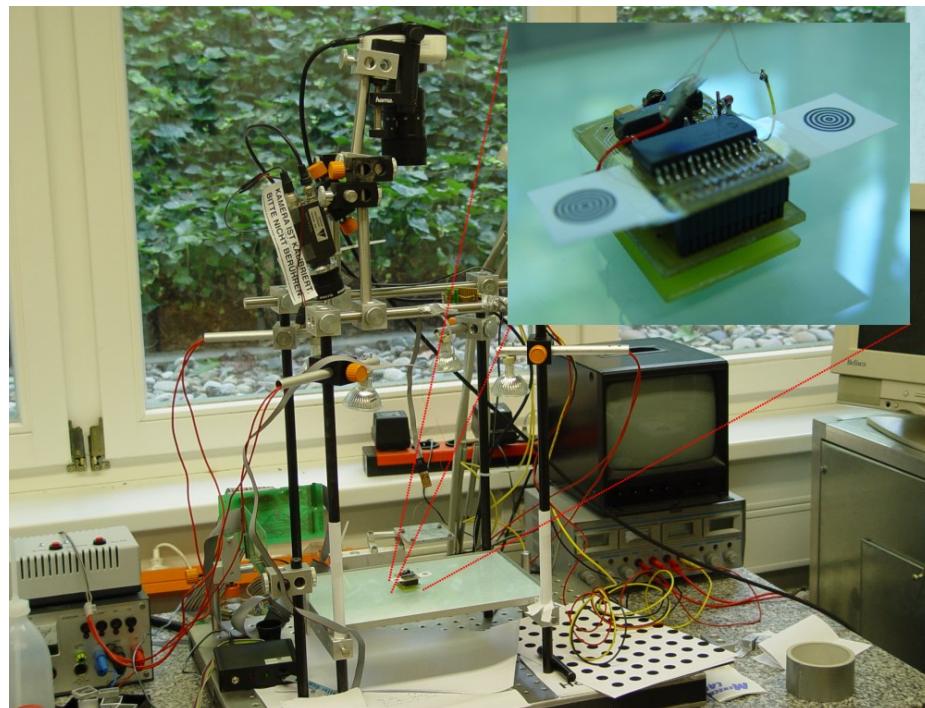




## MiCRoN

European Union IST project

Uppsala, Lausanne, St. Ingbert, Athens, Pisa, Barcelona, Karlsruhe



<http://wwwipr.ira.uka.de/~micron/>

<http://www.cordis.lu/ist/>

### project goals

- Manipulate  $\mu\text{m}$ -sized objects
- Closed-loop control of robots
- 3D object recognition and tracking

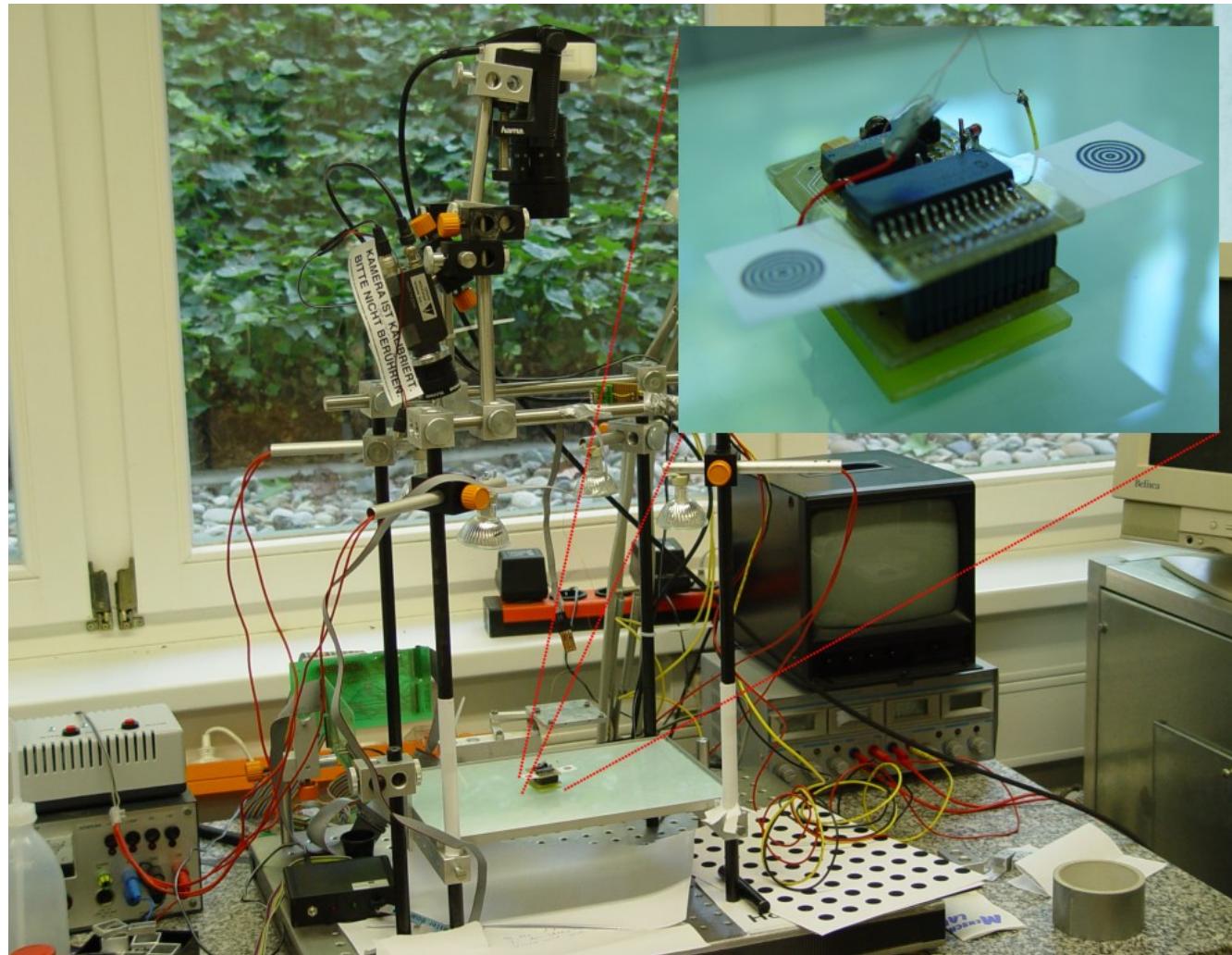
### image acquisition

- micro-camera
- microscope





## MiCRoN





## MiCRoN hardware (i)

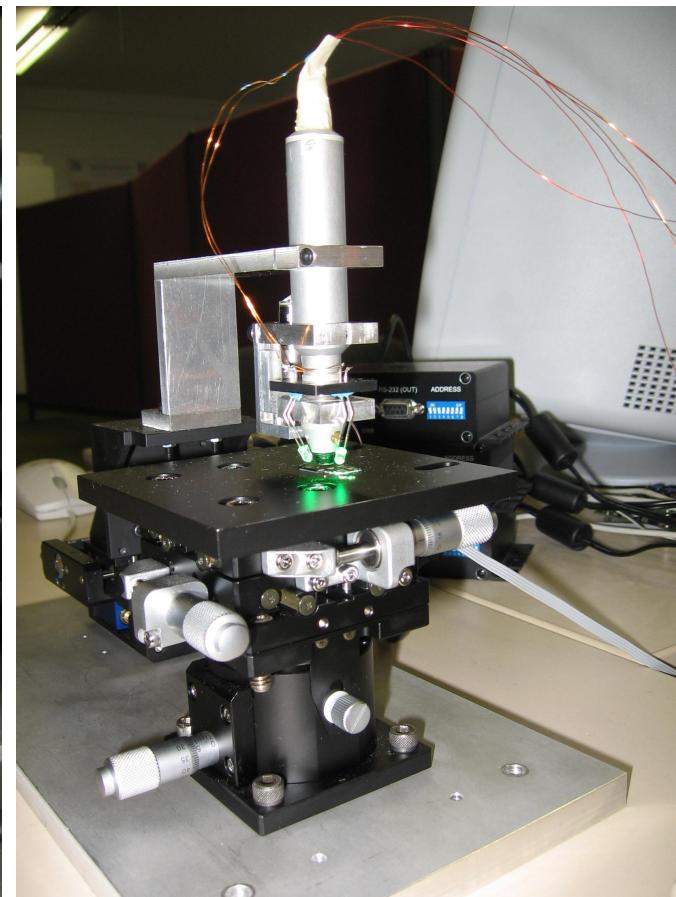
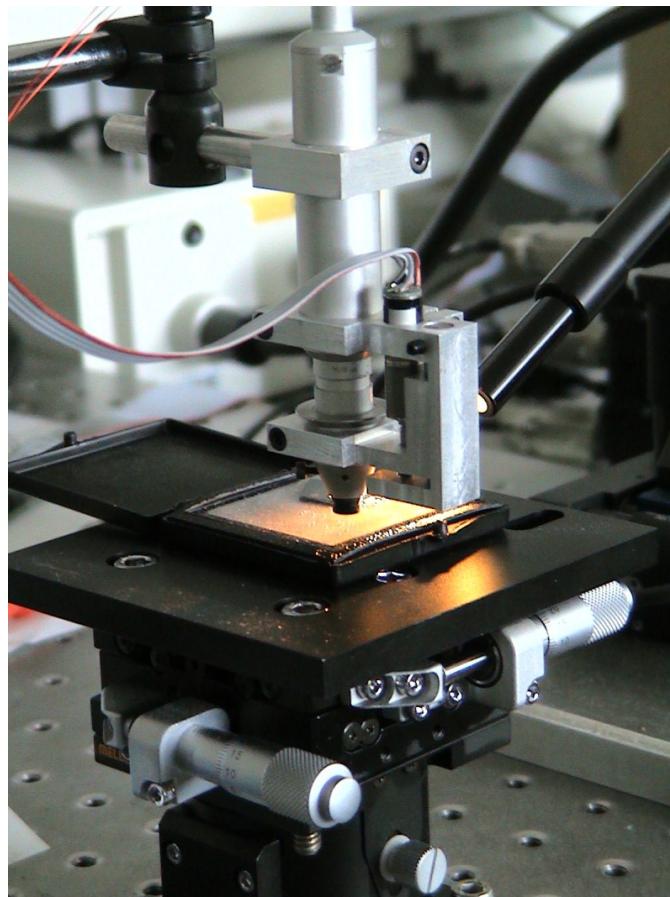
### camera





## test hardware (i)

### test environment





## MiCRoN hardware (ii)

first pictures



[video](#)



Sheffield Hallam University



MATERIALS AND ENGINEERING  
RESEARCH INSTITUTE

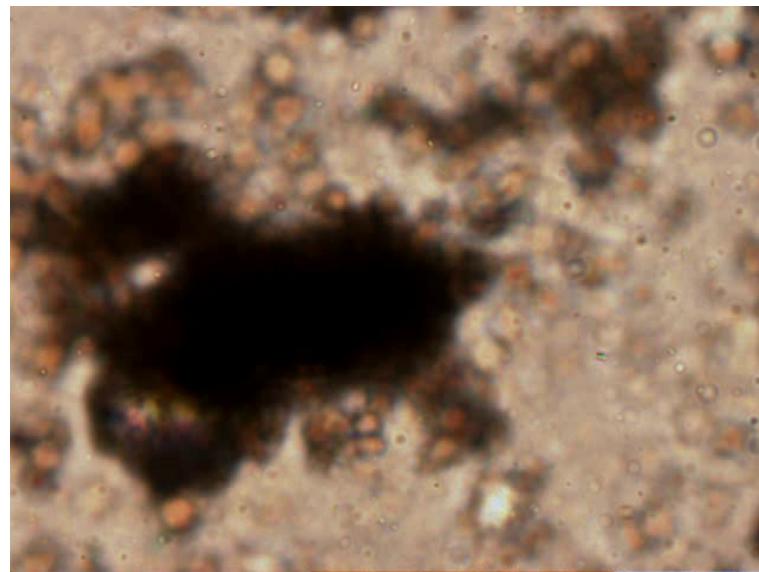


Microsystem & Machine Vision  
Laboratory



## test hardware (ii)

### microscope environment



camera sensor



actors



## software

## Mimas computer vision toolkit

# open source computer vision library



deliver solution  $\Leftarrow$  restrict vision domain  
 $\Leftarrow$  develop reusable toolset

- Sensor data: V4L, gstreamer-interface
  - Segmentation: LSI-filters, morphology, disparity-estimator, DFT, image-processing
  - Feature-extraction: edges, corners, ...
  - Feature-matching: optic flow, SVD-correspondence, correlation, champfer matching, PGH, fast POL
  - pose-estimation: particle filter, hough transform

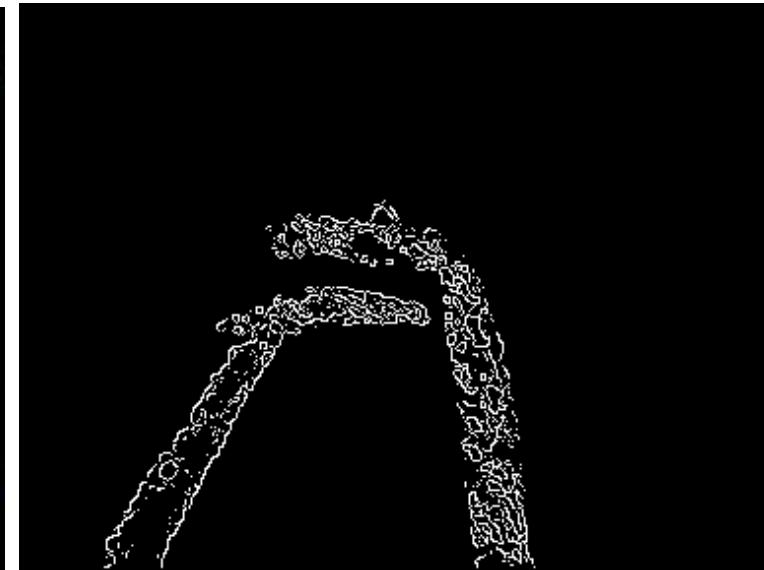
<http://vision.eng.shu.ac.uk/mediawiki/index.php/Mimas>



### Canny-like edge detector



video



edges





## geometric hashing

1988, Lamdan & Wolfson

### Geometric Hashing : A General and Efficient Model-Based Recognition Scheme

*Yechezkel Lamdan and Haim J.Wolfson*

Robotics Research Laboratory  
Courant Inst. of Math., NYU  
715 Broadway, 12'th floor,  
New York, N.Y. 10003.

**Abstract:** A general method for model-based object recognition in occluded scenes is presented. It is based on *geometric hashing*. The method stands out for its efficiency. We describe the general framework of the method and illustrate its applications for various recognition problems both in 3-D and 2-D. Special attention is given to the recognition of 3-D objects in occluded scenes from 2-D gray scale images. New experimental results are included for this important case.

#### 1. Introduction.

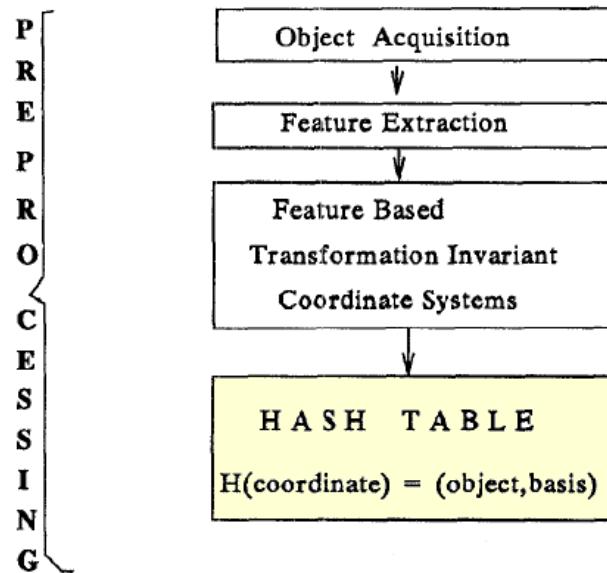
We present a unified approach to the *representation* and *matching* problems which applies to object recognition under various geometric transformations both in 2-D and 3-D. The objects are represented as sets of geometric features, such as points or lines, and their geometric relations are encoded using minimal sets of such features under the allowed object transformations. This is achieved by standard methods of *Analytic Geometry* invoking *coordinate frames* based on a minimal number of features, and representing other features by their coordinates in the appropriate frame. Our



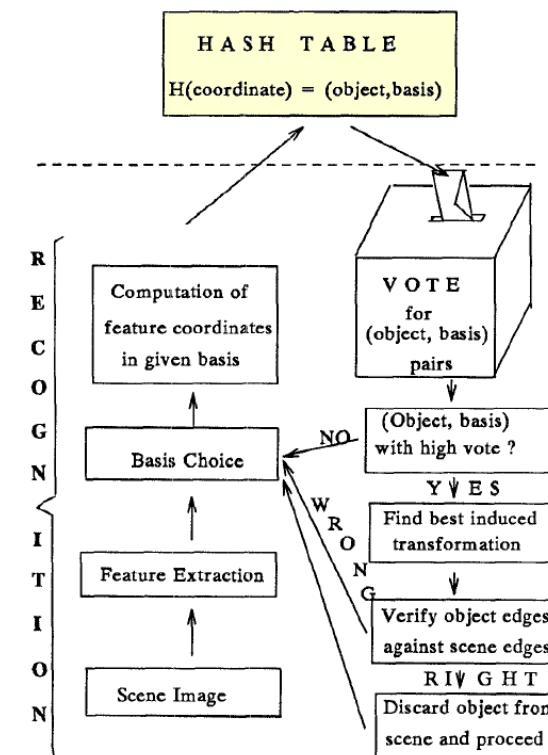


## geometric hashing

### preprocessing



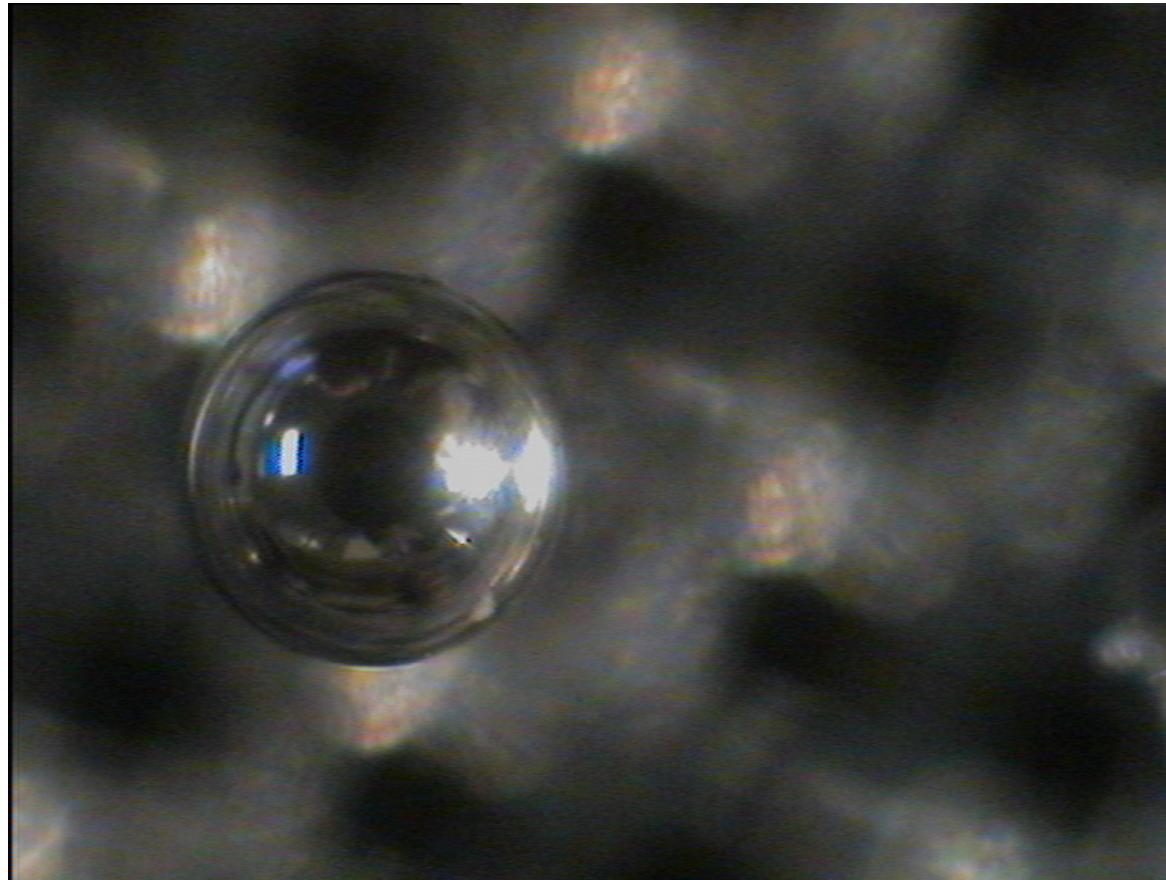
### recognition





## geometric hashing

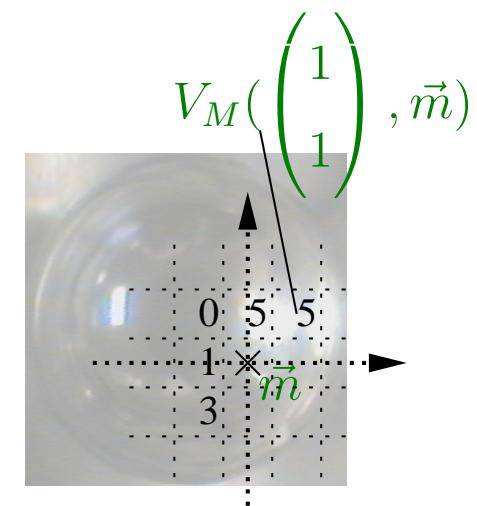
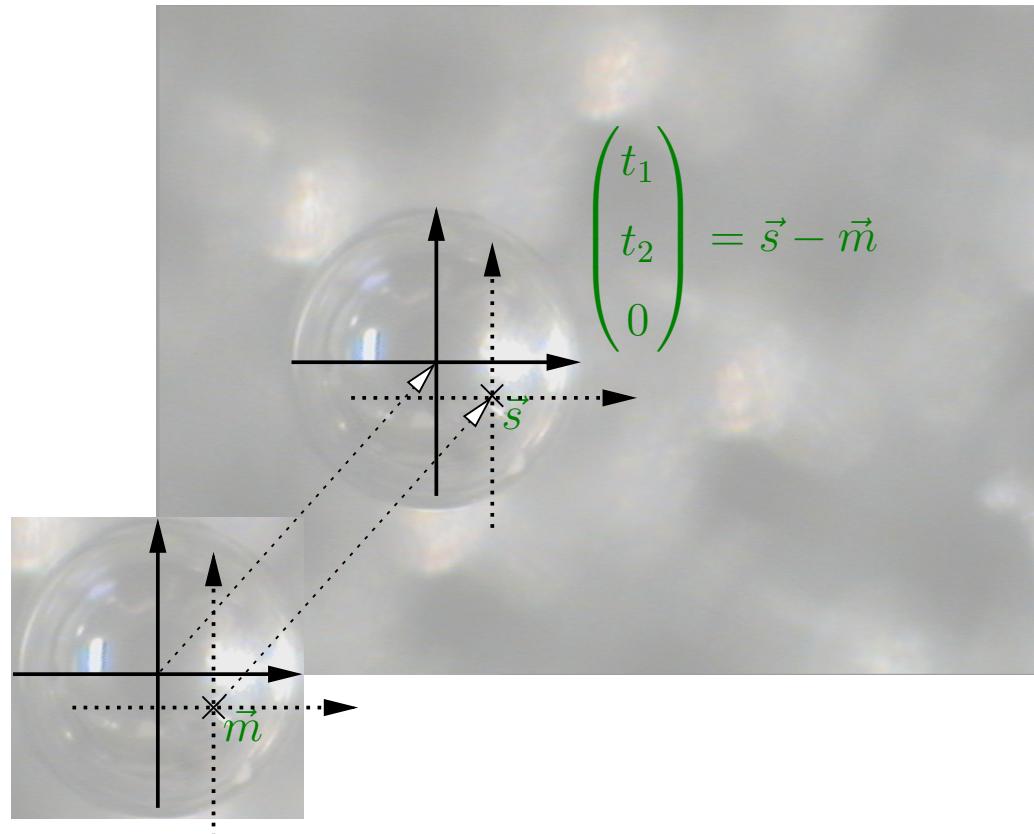
2-D / 2 DOF





## geometric hashing

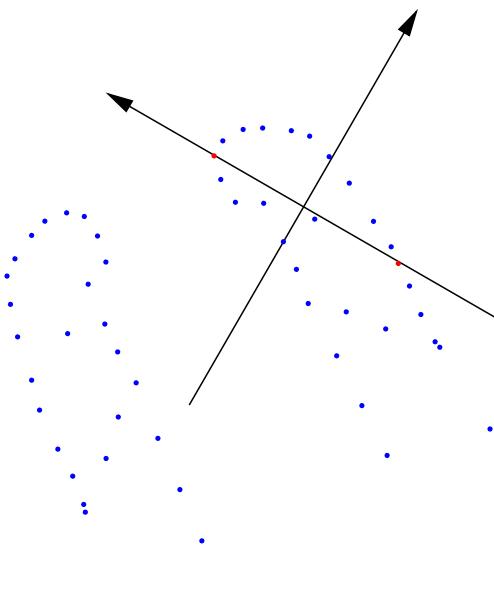
2-D / 2 DOF



## geometric hashing

2-D / 3 DOF

scene features



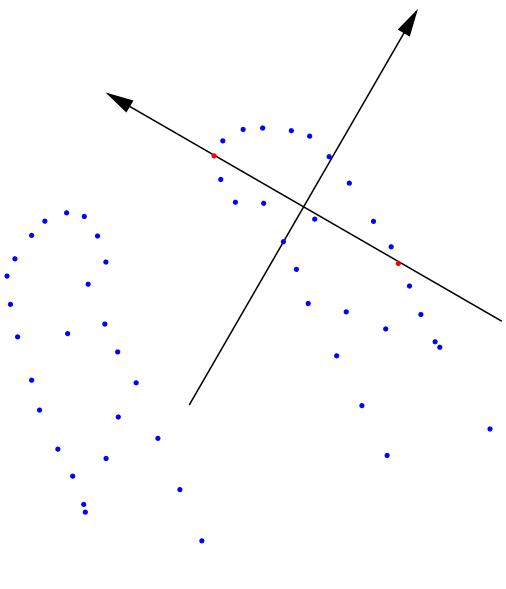
Sheffield Hallam University



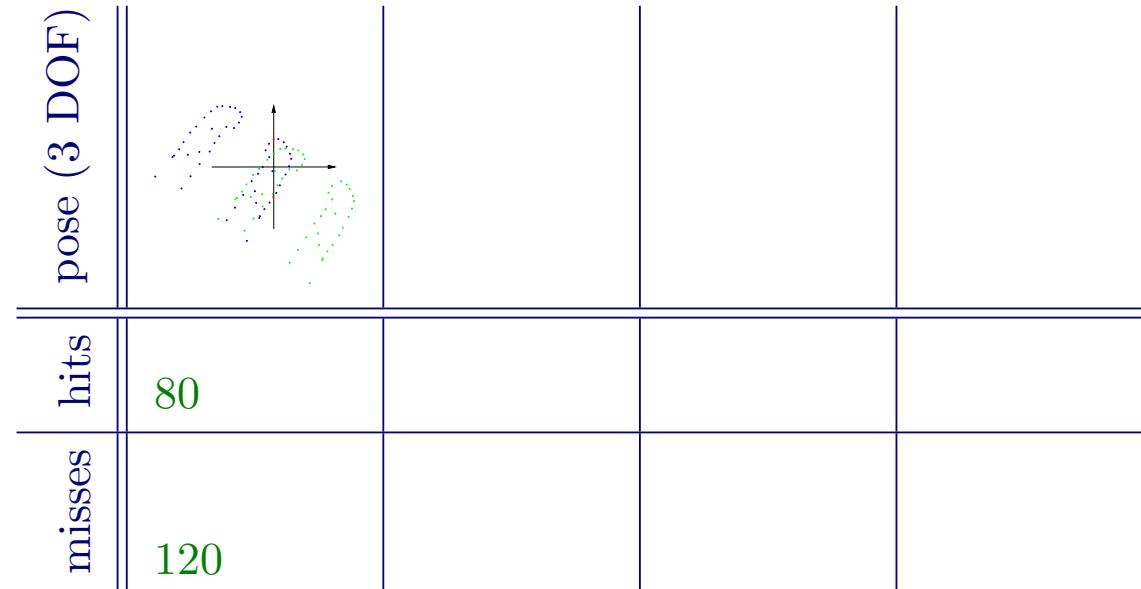
## geometric hashing

2-D / 3 DOF

scene features



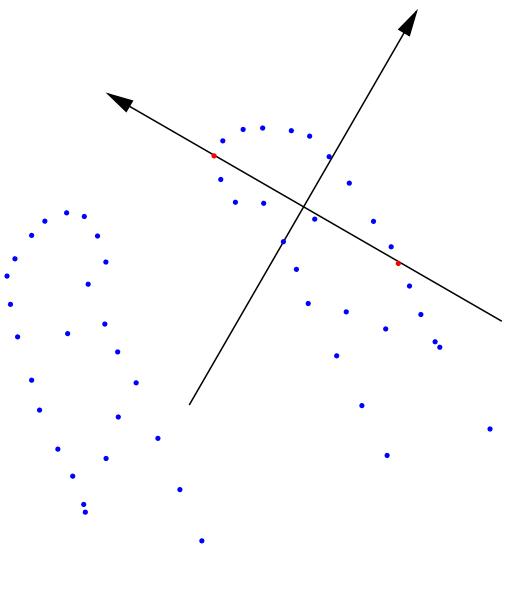
hash table



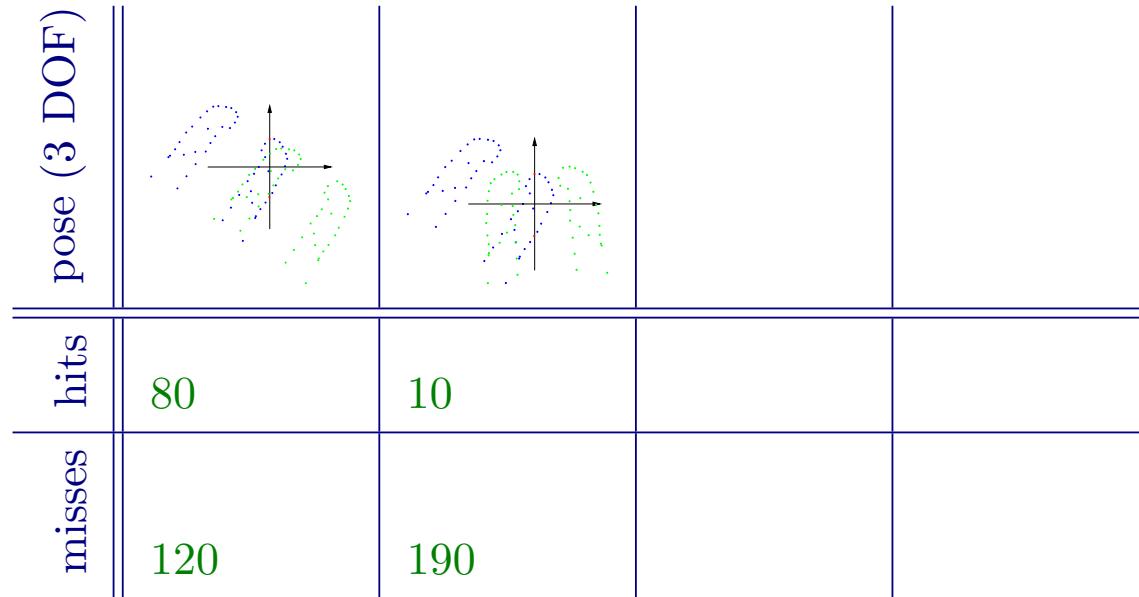
## geometric hashing

2-D / 3 DOF

scene features



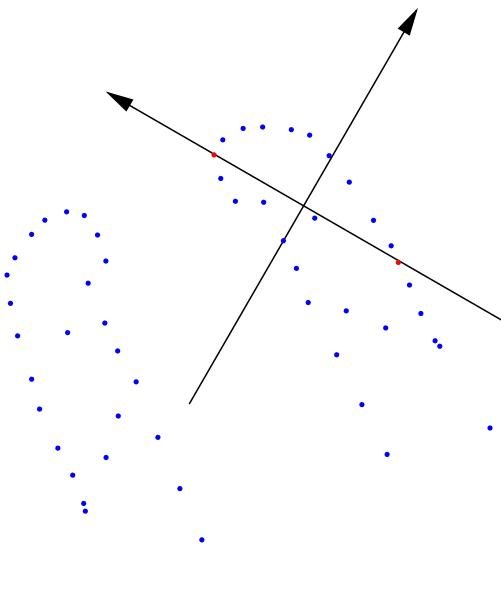
hash table



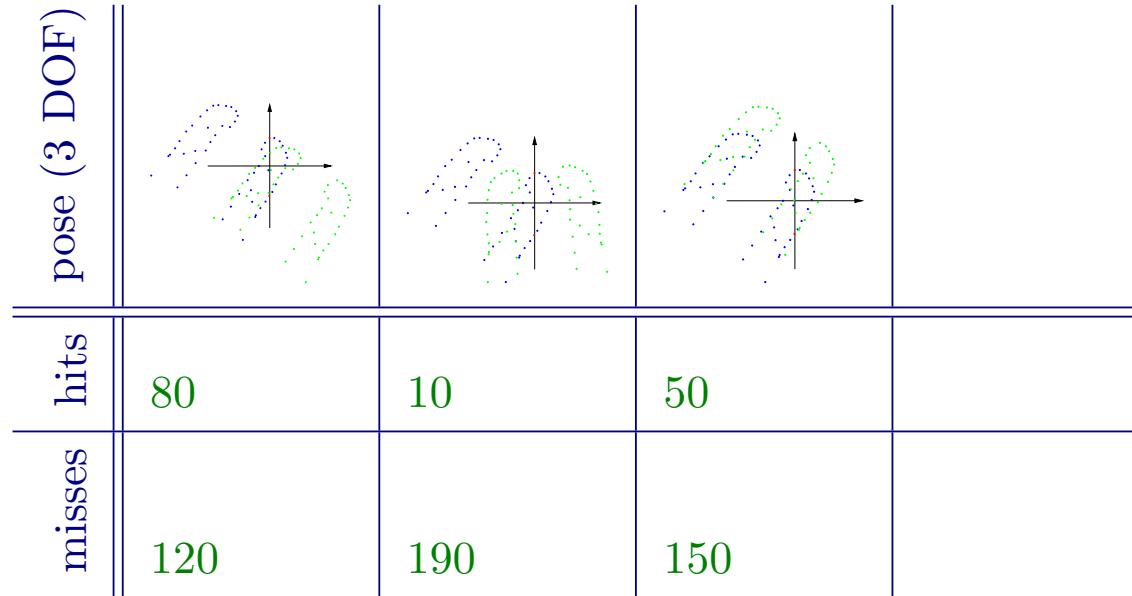
## geometric hashing

2-D / 3 DOF

scene features



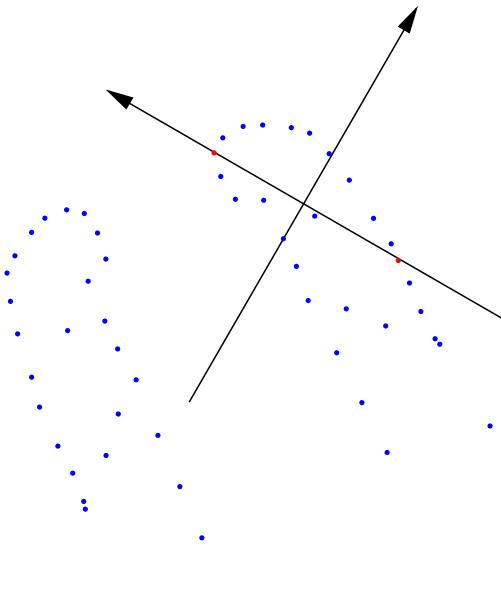
hash table



## geometric hashing

2-D / 3 DOF

scene features



hash table

		pose (3 DOF)			
		hits	10	50	200*
misses	hits	80	10	50	200*
	misses	120	190	150	0*

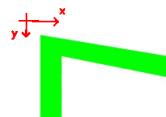




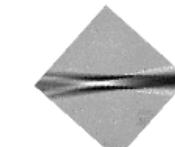
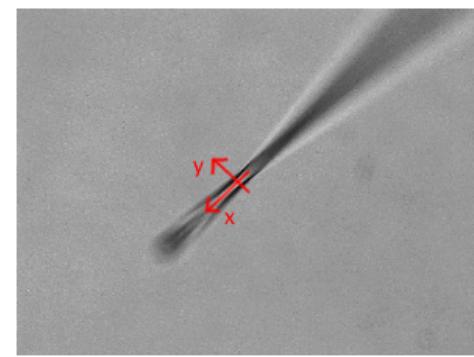
## geometric hashing

2-D / 3 DOF

Mimas 2005 (C) MMVL, Sheffield Hallam University  
2 object(s), timestamp = 22.00 s, 23th frame  
polygon at (-225.05 um, -116.84 um, 0.00 um), angle is 1.7 deg.  
triangle at ( 59.49 um, 189.31 um, 0.00 um), angle is -50.0 deg.



test on [artificial sequence](#)



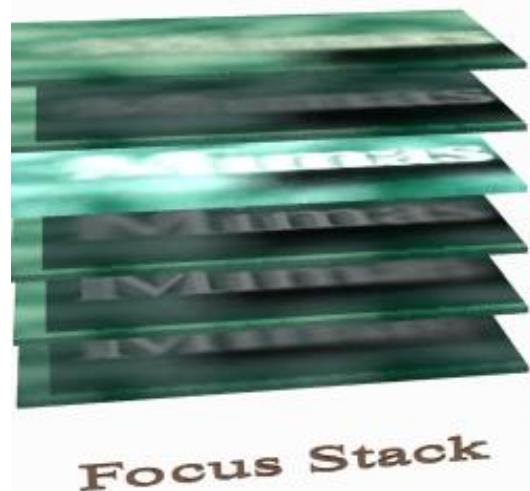
test on [pipette](#)





## geometric hashing

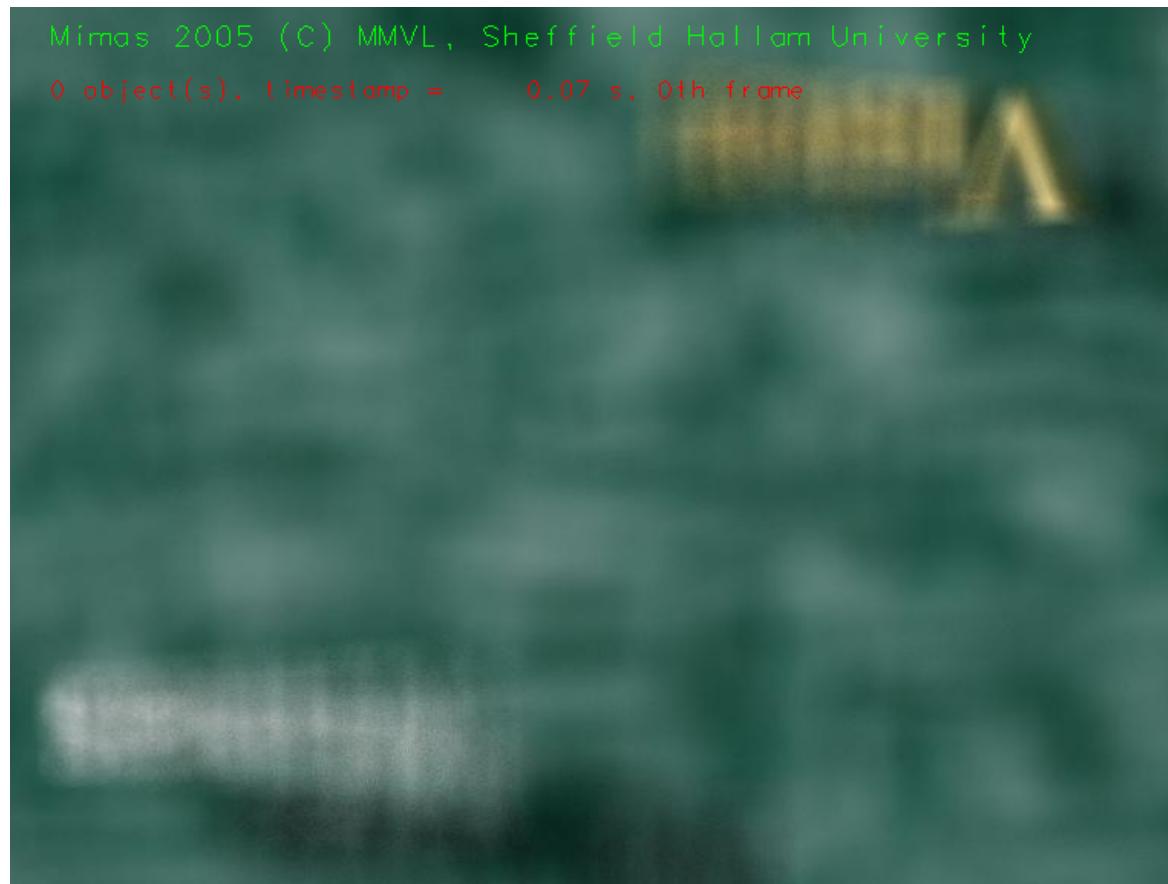
focus stacks as model database





## geometric hashing

### 3-D/4 DOF



test on povray sequence (time lapse)





## bounded hough transform

2001/2004, Greenspan, Shang & Jasiobedzki

### Efficient Tracking with the Bounded Hough Transform

Michael Greenspan<sup>1,2,4</sup> Limin Shang<sup>1</sup> Piotr Jasiobedzki<sup>3</sup>

<sup>1</sup>Dept. of Electrical & Computer Engineering, <sup>2</sup>School of Computing, Queen's University, Canada

<sup>3</sup>MDRobotics, 9445 Airport Rd., Brampton, Ontario, Canada

<sup>4</sup>corresponding author: michael.greenspan@ece.queensu.ca

#### Abstract

*The Bounded Hough Transform is introduced to track objects in a sequence of sparse range images. The method is based upon a variation of the General Hough Transform that exploits the coherence across image frames that results from the relationship between known bounds on the object's velocity and the sensor frame rate. It is extremely efficient, running in  $O(N)$  for  $N$  range data points, and effectively trades off localization precision for runtime efficiency.*

*The method has been implemented and tested on a variety of objects, including freeform surfaces, using both simulated and real data from Lidar and stereovision sensors.*

ing variants of the Iterative Closest Point Algorithm (ICP) [1]. This is primarily because range data is more expensive to collect, and so the images tend to be sparse, which makes it difficult to extract meaningful features. Examples of ICP-based tracking are [2, 3] and recently [4], which simultaneously reconstructs while tracking.

The Hough Transform is a well known and effective method of feature extraction and pose determination that has been explored thoroughly in the literature [5]. Many variations of the Hough Transform have been proposed [6], some of which are specifically tailored to tracking. The Velocity Hough Transform (VHT) [7] included a specific ve-





## bounded hough transform

2001/2004, Greenspan, Shang & Jasiobedzki

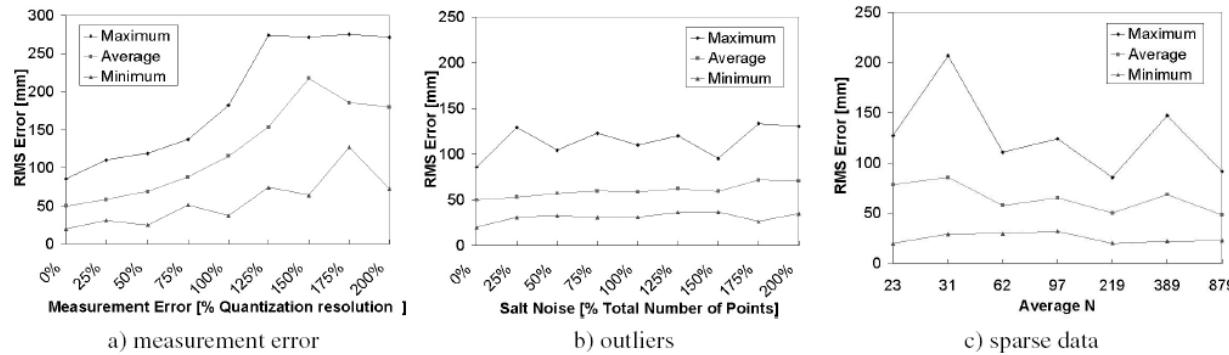


Figure 6: Effects of Data Degradation



Figure 7: Robot Mounted Satellite Model  
of a Radarsat satellite was mounted on a 6 dof articulated

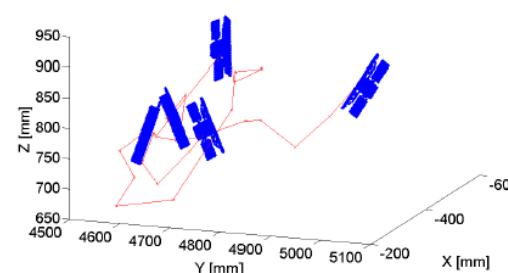


Figure 8: Satellite Trajectory, Lidar Data

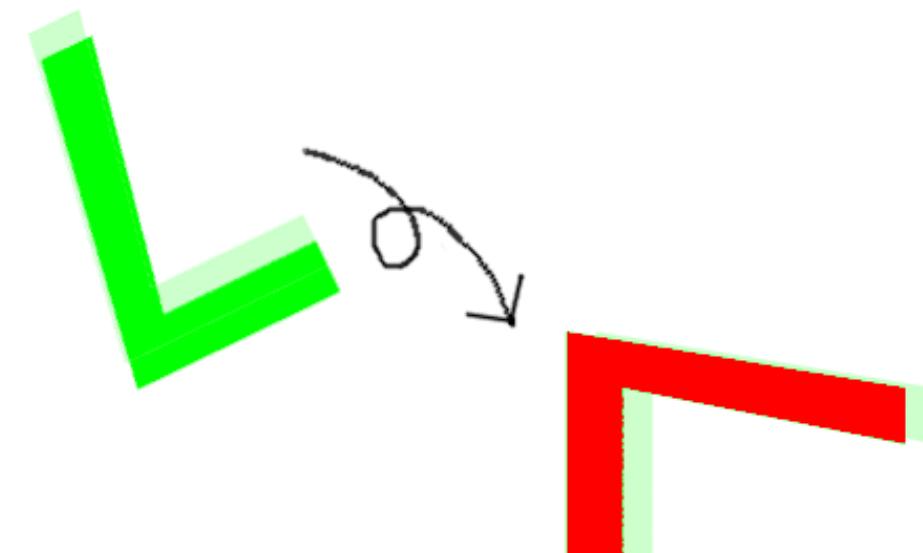
majority of which fell on the surface of the robot and the background and were therefore outliers. To demonstrate the effectiveness of the method at tracking in sparse as well as





## bounded hough transform

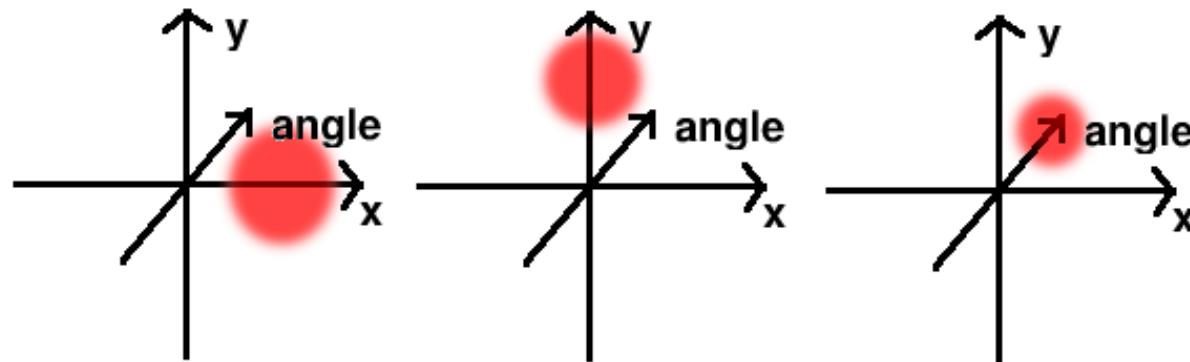
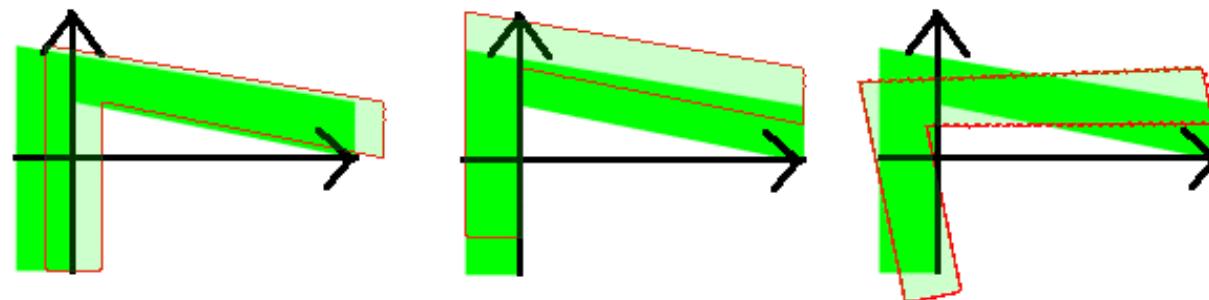
### inverse transformation





## bounded hough transform

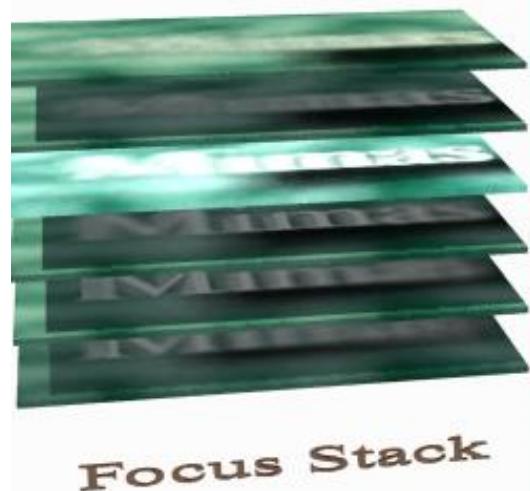
houghspace for relative shifts





## bounded hough transform

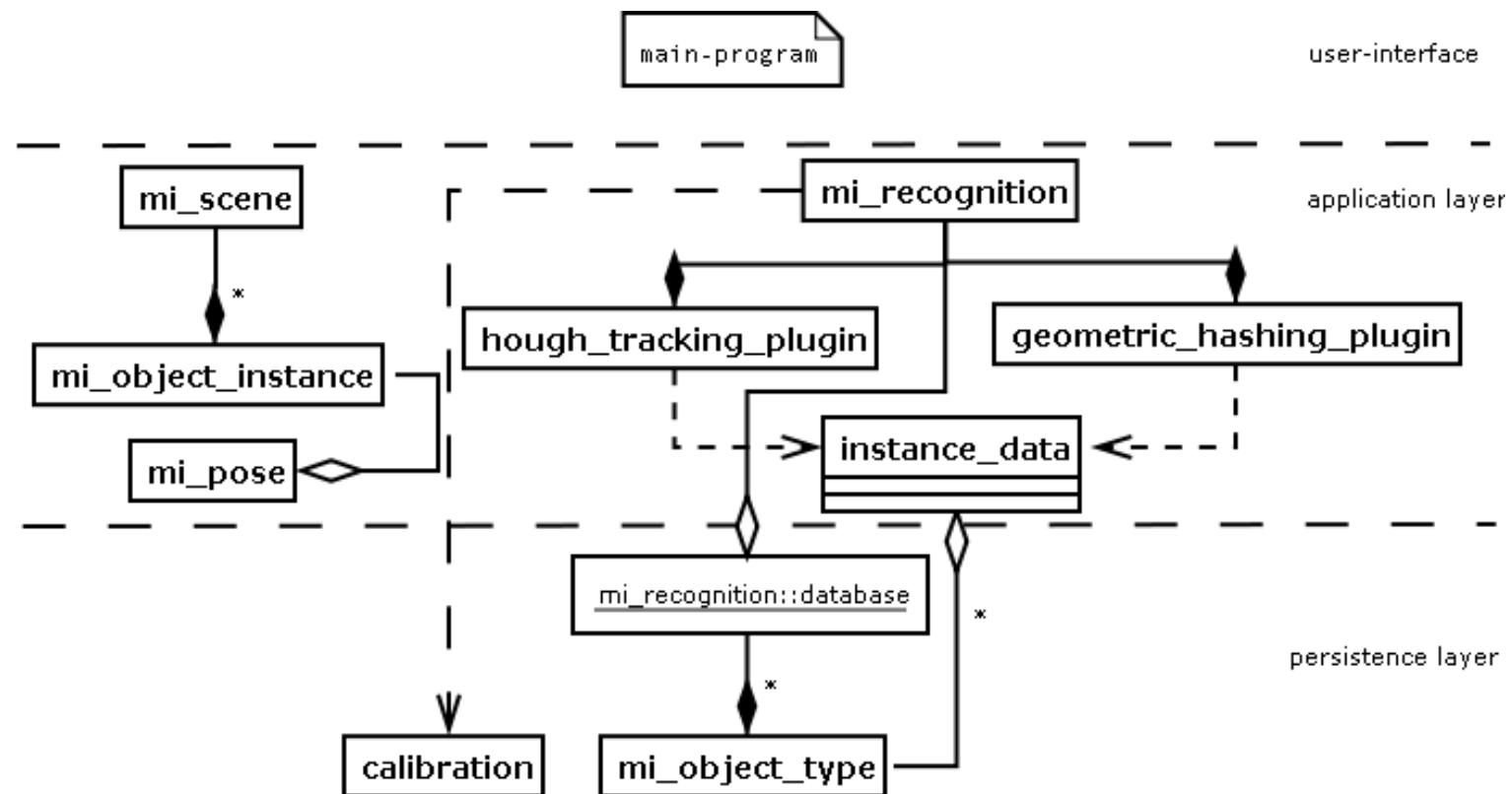
focus stacks for 4 DOF





## software architecture

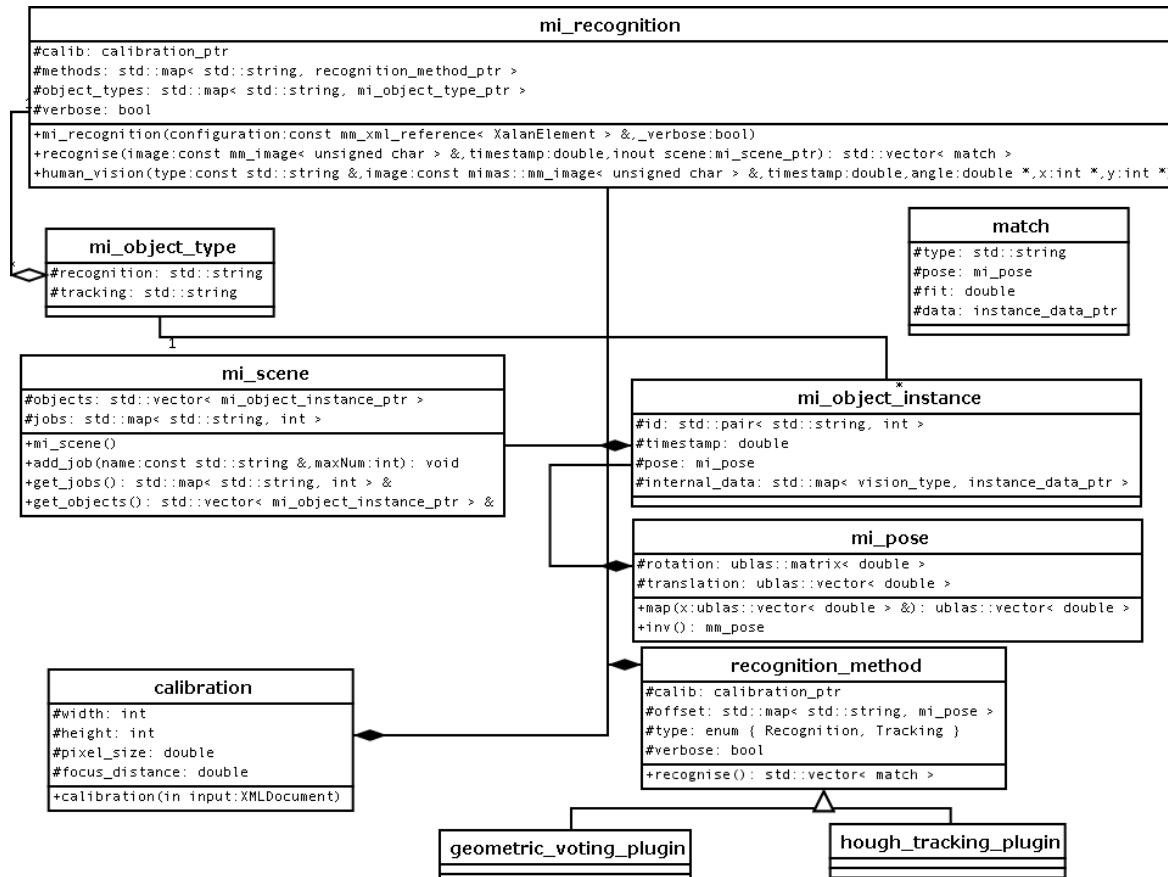
### application layers





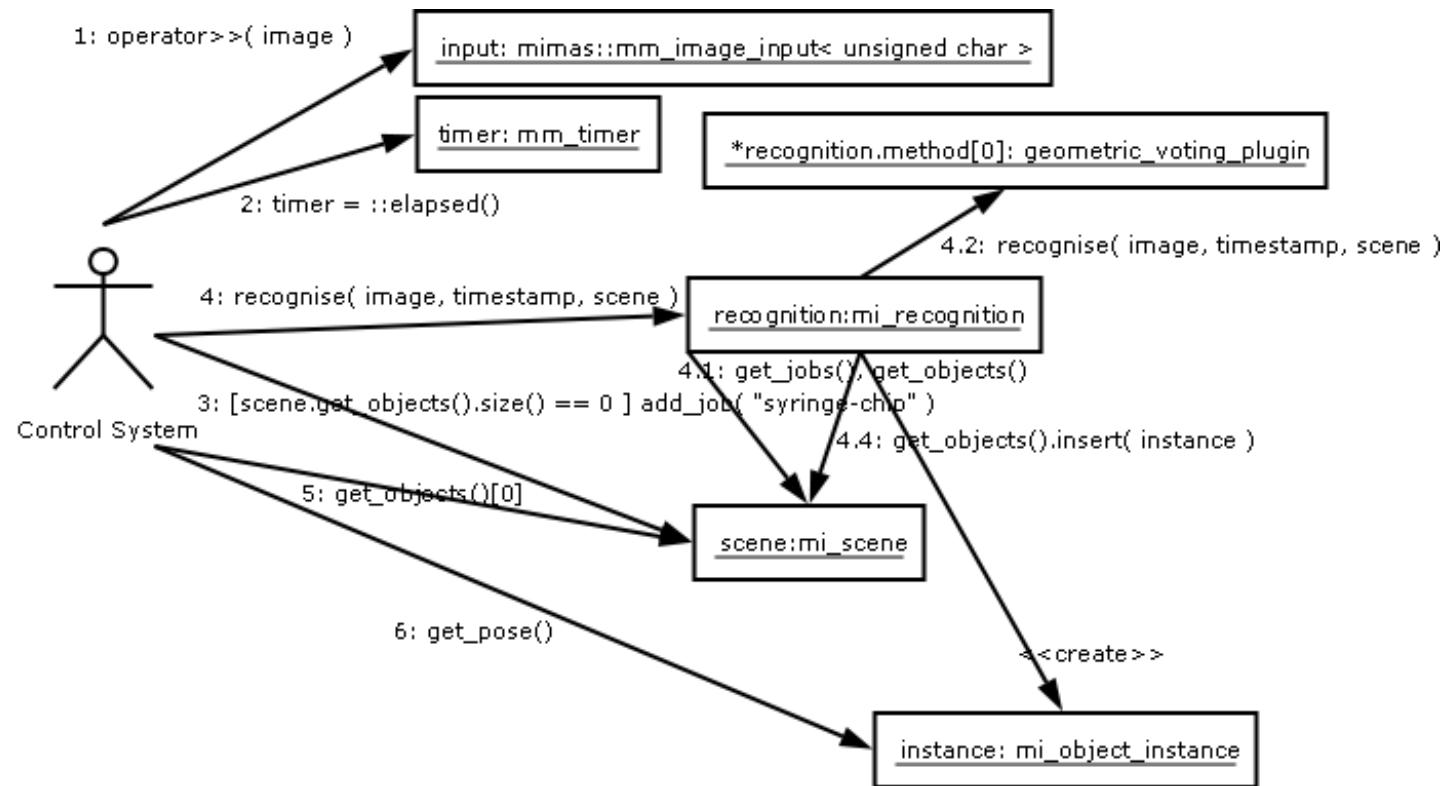
## software architecture

### UML static structure



## software architecture

### UML collaboration diagram





## command-line tool

```
jan@wedesoft:~/Documents/micron-vision/micron-vision - Befehlsfenster 2 - Konsole: [x]
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
jan@wedesoft:~/Documents/micron-vision/micron-vision> ./micron --help
Usage: ./micron [FILE]...
  -c, --config=xml-file-name      Specify alternate configuration file
                                  (default: "configuration.xml")
  -s, --schema=xsd-file-name    Specify alternate location of schema file
                                  (default: "configuration.xsd")
  -g, --gstreamer                Select gstreamer input for reading images
  --shm=integer                  shared memory key for gstreamer input
                                  (default: 333)
  --sem=integer                  semaphore key for gstreamer input (default:
                                  333)
  -v, --v4l                      Select video4linux for reading images
  --brightness=brightness        Brightness for video4linux-input [0..65535]
                                  (default: 32768)
  --contrast=contrast            Contrast for video4linux-input [0..65535]
                                  (default: 32768)
  --video1394                   Select firewire digital camera for reading
  --shutter=shutter              images
  --gain=gain                    Shutter for video1394-input (default: -1)
  --balance=balance              Gain for video1394-input (default: -1)
  --device=device-name           White balance for video1394-input (default:
                                  -1)
  --channel=channel-number       Specify video4linux-/video1394-device
  --fps=integer                  Specify video4linux-channel/video1394-node
                                  (default: 0)
  --framedrop                   use number of image divided by given
  --savepics                     frame-rate for timestamp (default: use time
                                  elapsed)
  -d, --framedrop                Drop frames, if algorithm is slower than
  --wait                         given frame-rate
  --verbose                       Save pictures for debugging-purposes
  --disablex11                   (debug*.ppm)
  --enablex11                     Pause after initialisation
  -w, --wait                      Get verbose output from vision-methods
  --usage                         Disable graphical output
  -x, --disablex11                Enable graphical output (default)

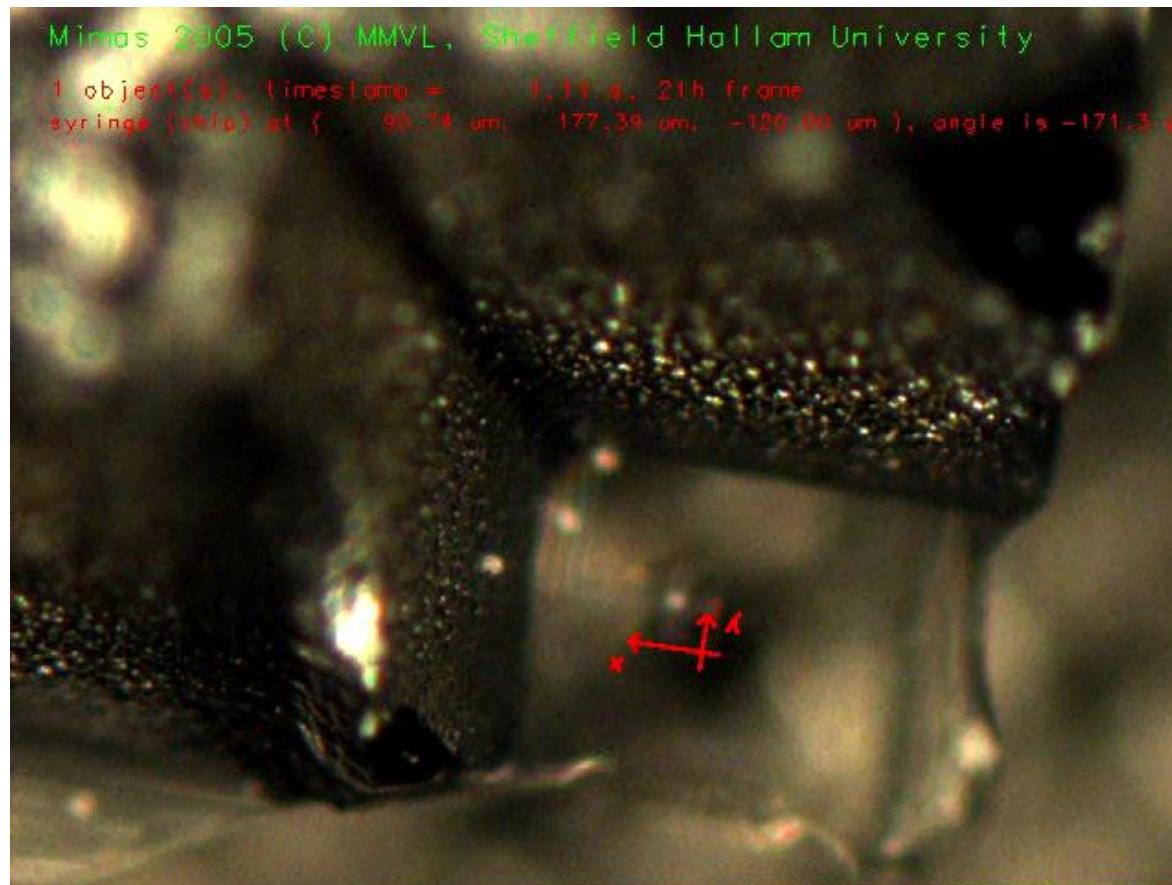
Help options:
  -?, --help                      Show this help message
  --usage                         Display brief usage message
jan@wedesoft:~/Documents/micron-vision/micron-vision> [x]
```





tests

## syringe-chip



[video](#)





tests

gripper



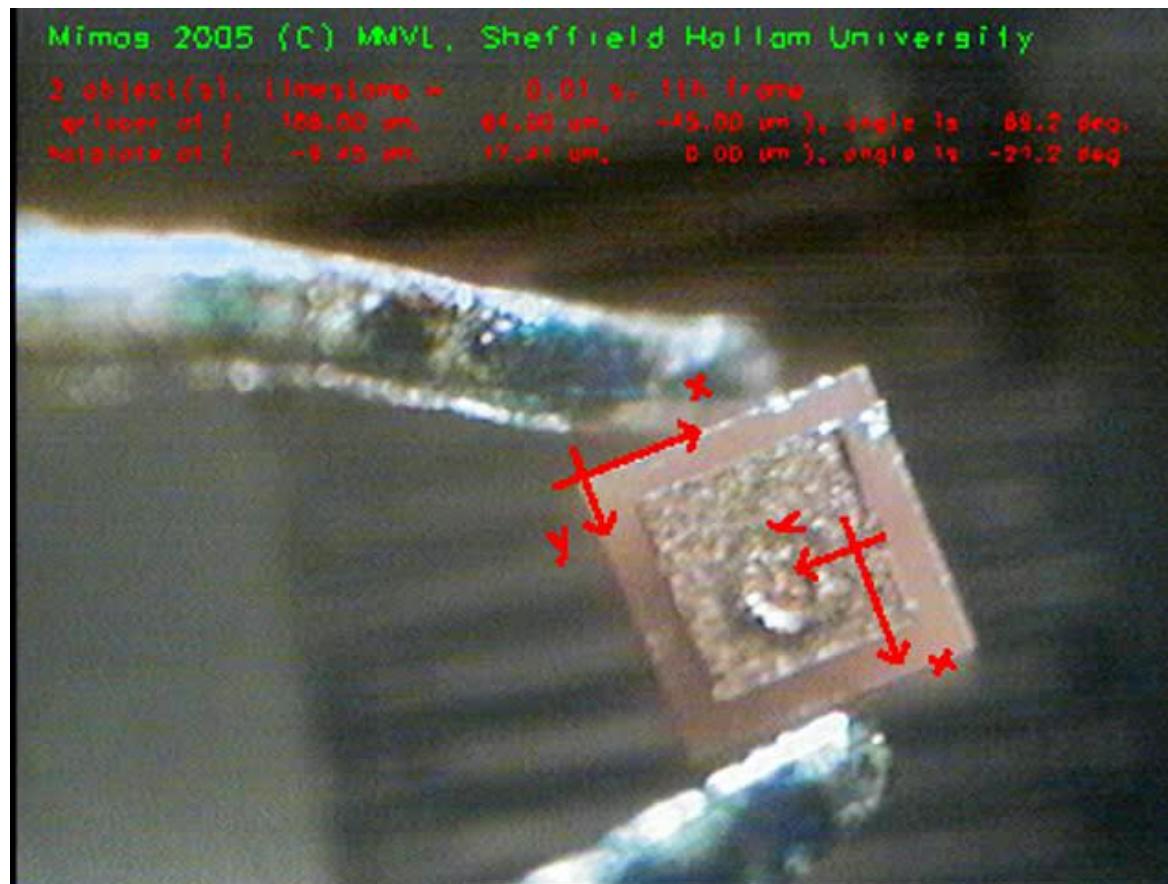
video





tests

gripper2



[video](#)





tests

## full automation



video

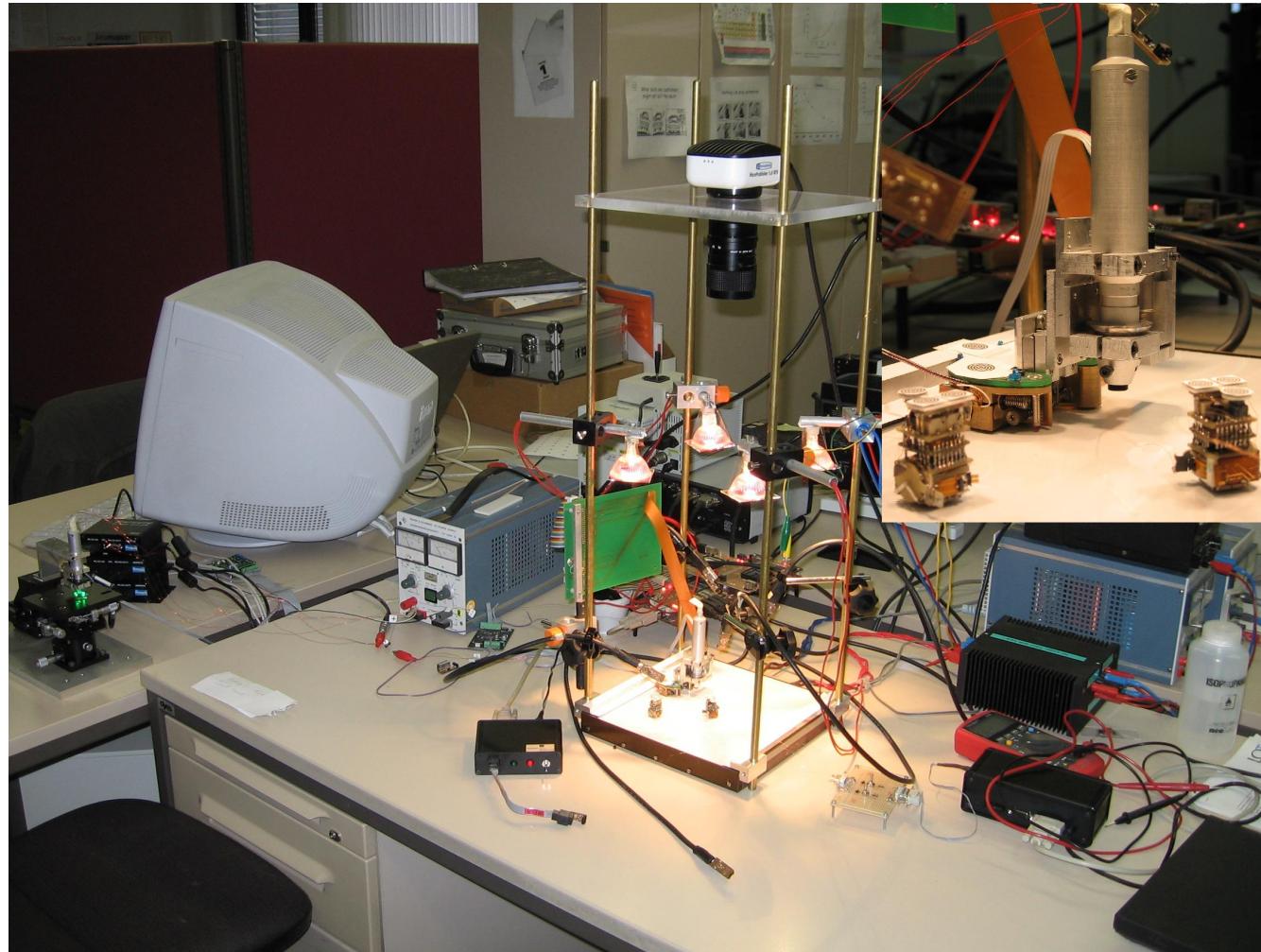


video



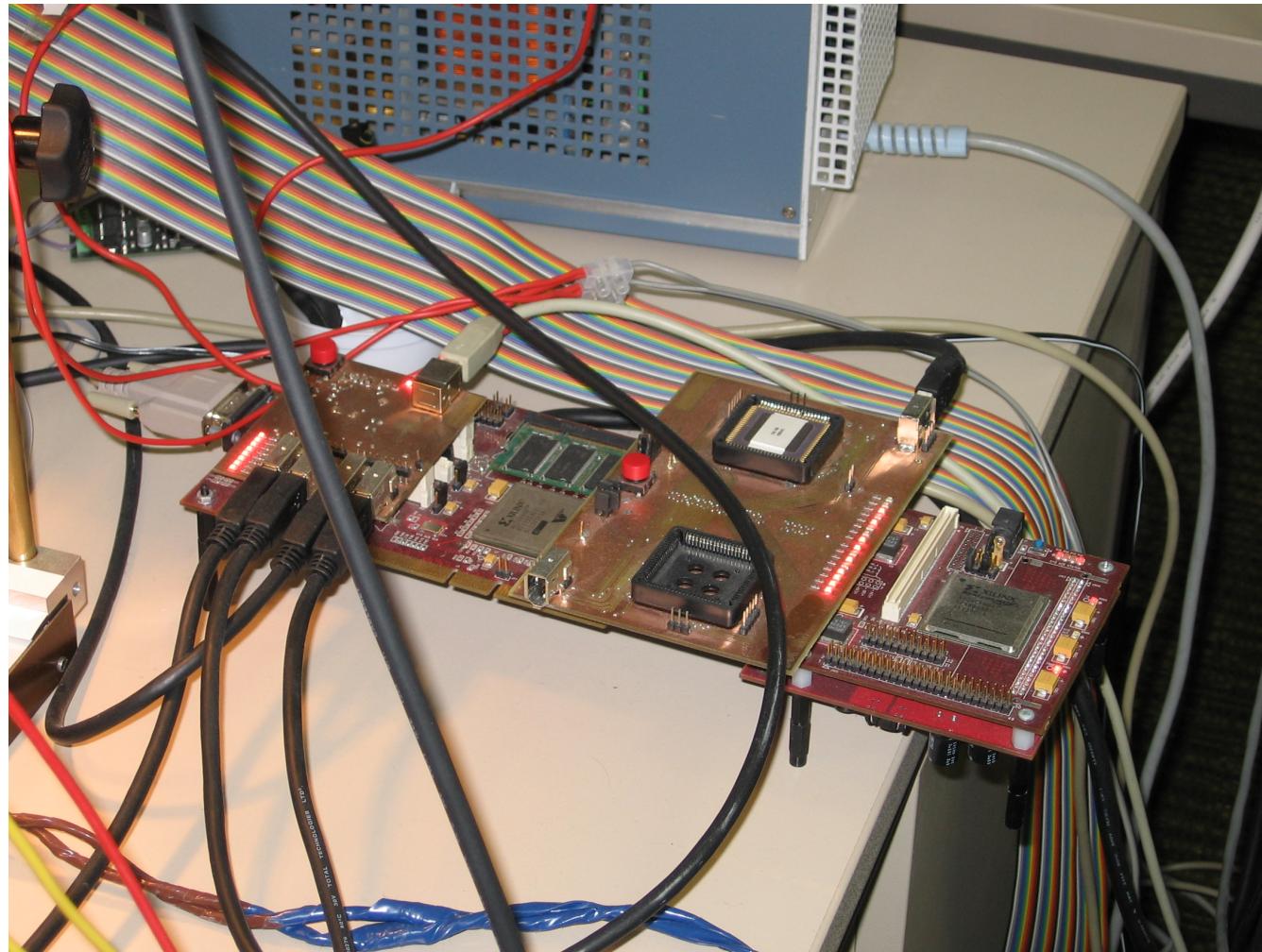


### MiCRoN setup (i)



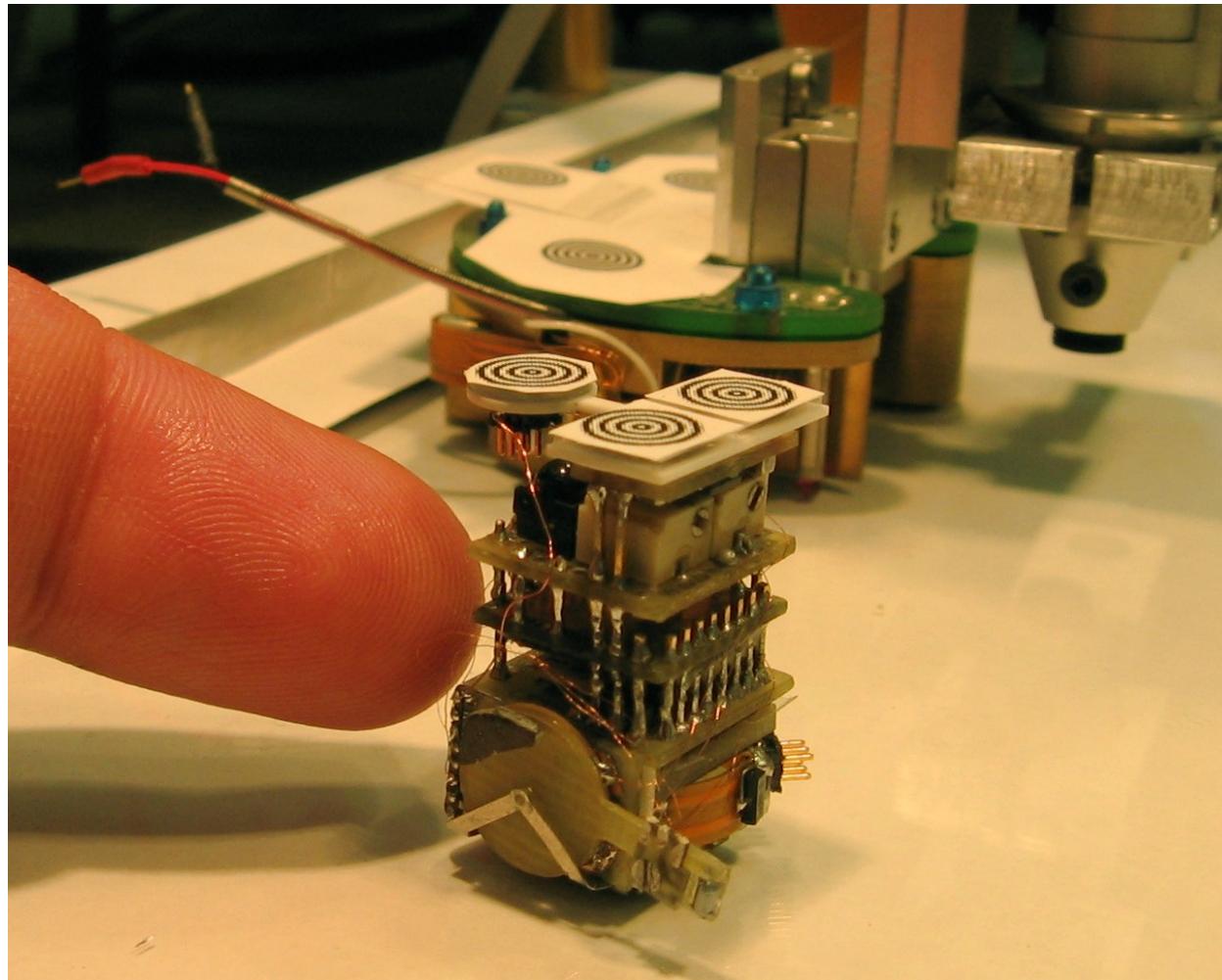


## MiCRoN setup (ii)



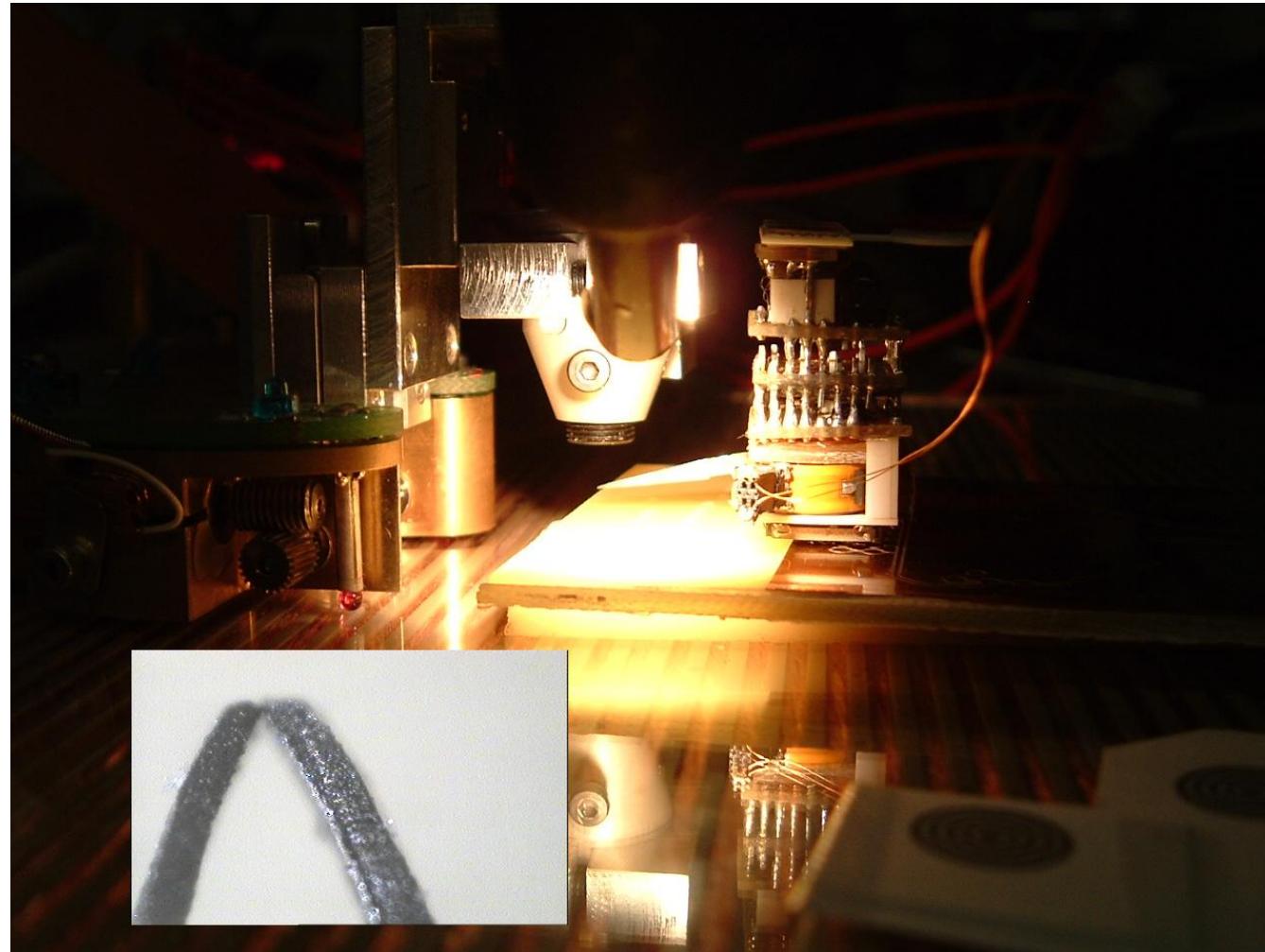


### MiCROn setup (iii)





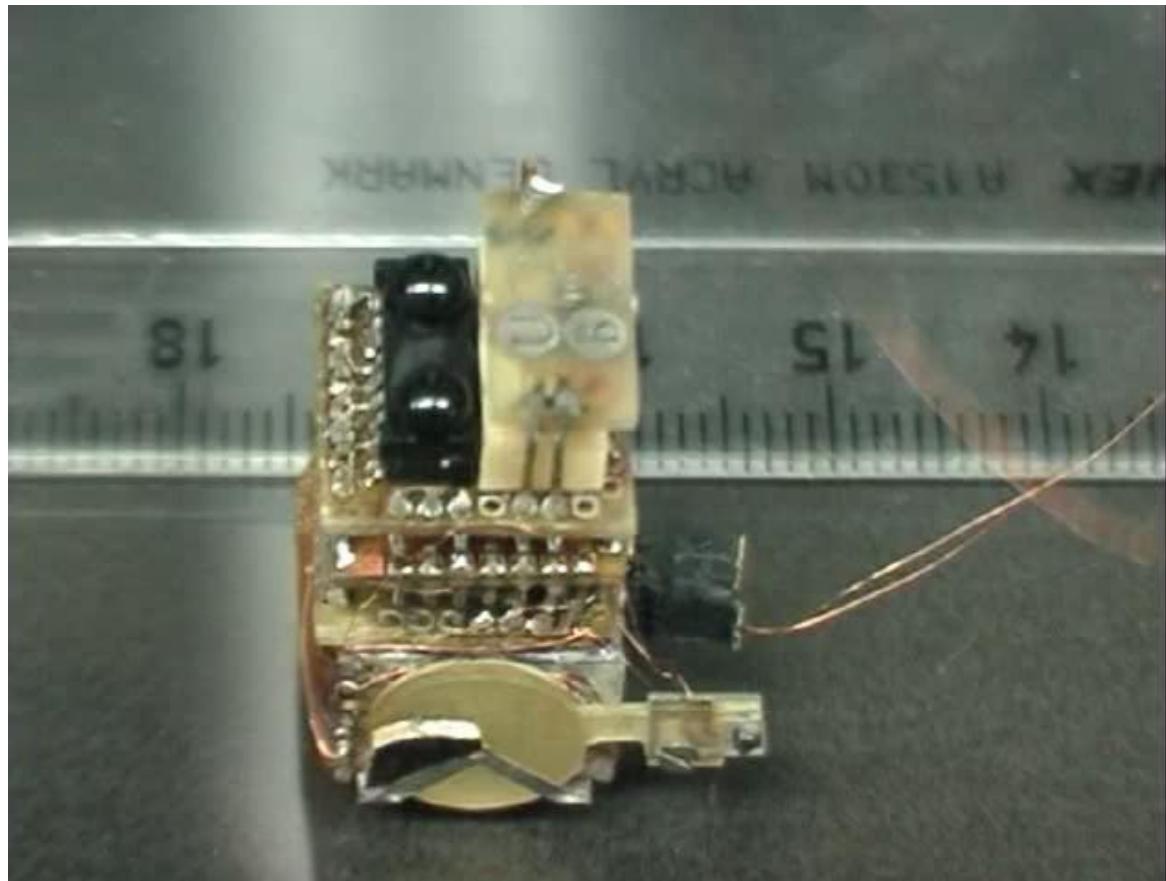
### MiCROn setup (iv)





MiCRoN

assembly mockup



[video](#)



Sheffield Hallam University





## MiCRoN

### assembly mockup



[video](#)



Sheffield Hallam University





## demonstration



macro-scale demonstration



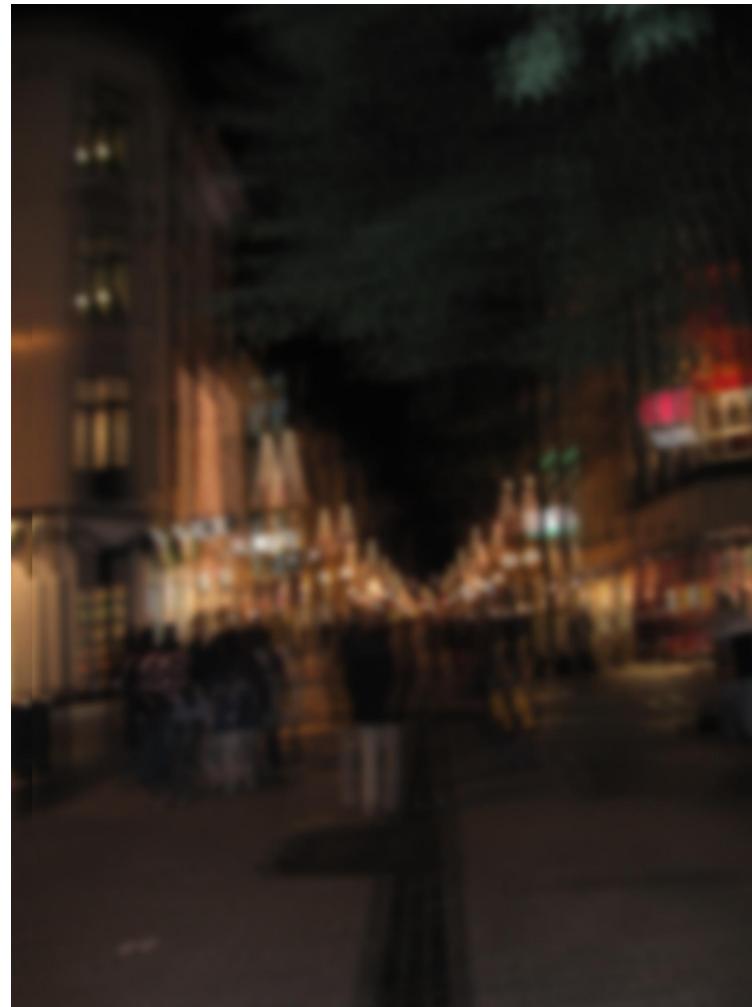


## acknowledgements

- Jon Travis: technical suggestions, administrative support
- Fabio Caparrelli: camera drive/electronics/driver software, electronics setup, PI driver software, management
- Balasundram Amavasai: proposal, computer vision suggestions, mimas long-term strategy
- Arul Selvan: students supervision, telecentric optics, shift estimation
- Manuel Boissenin: realtime tracking of multiple micro-objects, parallelisation
- Jan Wedekind: software architecture and integration, realtime recognition of multiple micro-objects













## Happy Christmas!

<http://vision.eng.shu.ac.uk/mediawiki/>

article discussion edit history protect delete move unwatch

# Main Page

This is the MediaWiki of the [Microsystem & Machine Vision Laboratory](#).  
Have a look at the MMVL's open-source [mimas](#) computer vision toolkit and the MMVL's efforts in [medical image processing](#).  
There is also information about the [Miniman](#), [Micron](#) and [I-Swarm](#) European funded projects and the [Nanorobotics](#) EPSRC funded project.  
If you need assistance, please visit the [help section](#).

[edit]

## See Also

- [Mimas Mimas](#)
- Depth from Focus
- Microscope Control and Sugar Pushing
- Camera Surveillance
- Iso Surface Extraction
- Jennic ZigBee Device

[edit]

## External Links

- Official MMVL homepage [↗](#)
- Autonomous mobile robots [↗](#)
- Medical image processing [↗](#)
- High Speed imaging at IPOT 2005 Birmingham (12.8 MByte Divx3 video) [↗](#)

This page was last modified 14:57, 16 Nov 2005. This page has been accessed 303 times.  
Content is available under [GNU Free Documentation License 1.2](#). [About MMVLWiki](#) [Disclaimers](#)

Upcoming: MiCROn public final report

