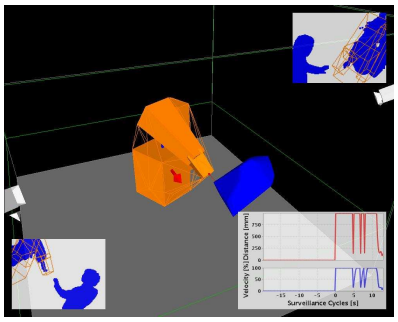
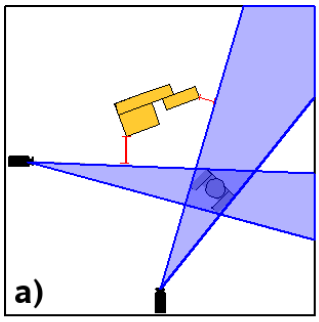
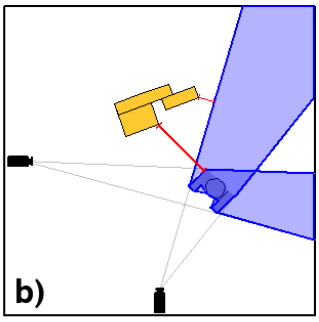
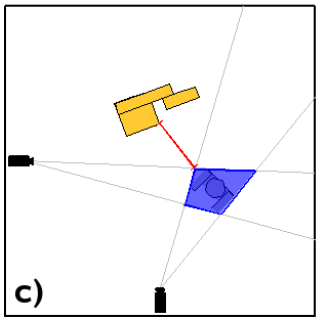
Surveillance of Robots using Multiple Colour or Depth Cameras with Distributed Processing

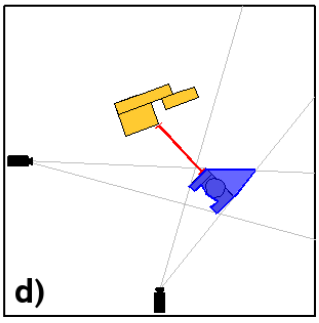
Fischer Markus and Dominik Henrich











Surveillance of Industrial Robots

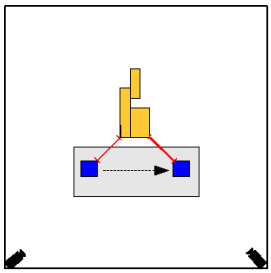
A robot without surveillance of its workspace is unaware of unexpected changes of its environment and can not react properly. The robot workspace must be surveilled using multiple camera images to detect unknown objects in it.

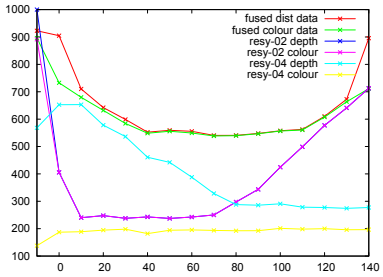
The robot velocity (blue line) is controlled using the minimum distance (red line) between the robot (orange polyhedron) and any detected unknown object (blue polyhedron) in the workspace.

Minimum Distance Approximation Quality

For multiple colour images (a, c) or depth images (b, d), the minimum distance can be determined as the multi-camera image-based distance (a, b) or the fusion-based distance (c, d) between the robot and all detected unknown objects.

While the distances determined by depth images are always more accurate than by colour images, the fusion-based distance is always more accurate than the multi-camera image-based distance, too.





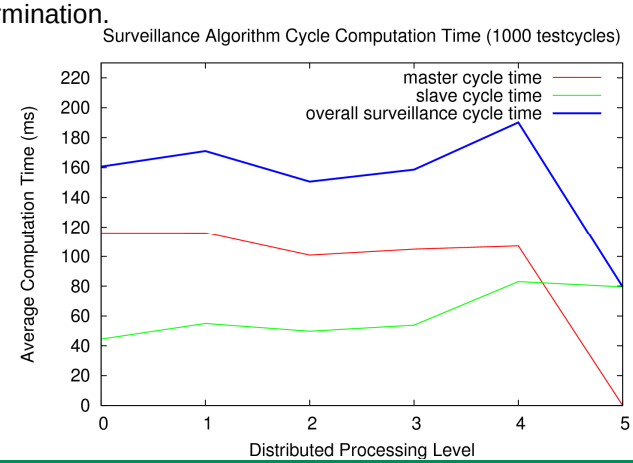
Distributed Processing

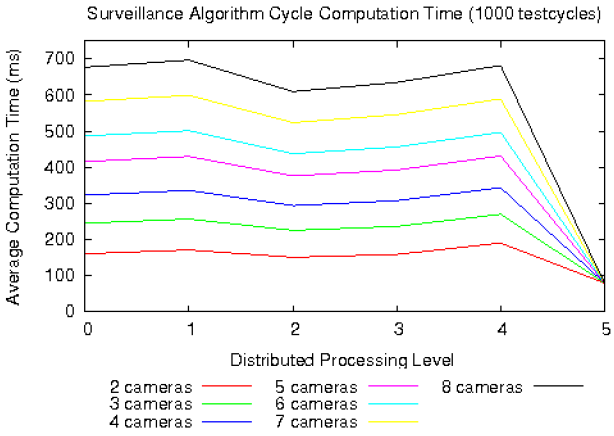
The surveillance algorithm cycle consists of three major computational steps, which are divided into sub-steps. Most sub-steps can be processed on both, consecutively on the master or parallel on the camera slaves. The distributed processing level controls the number of sub-steps processed on master and slaves and the data transferred.

The most significant impact on the complete cycle time for a set number of cameras is given by the choice of the distance determination method, since for the multi-camera image-based distance determination only the processing time of the camera slaves is important.

For an increasing number of cameras, the overall cycle time remains almost constant for the multi-camera image-based distance determination.

Distributed Processing Level		Substeps on Slaves	Substeps on Master	Submitted Data
0	Raw Image	0	9	N*M Float
1	Segmentation Image	1	8	N*M (Float+Bool)
2	Object Pixel	2	7	O*P (2*Byte+Float)
3	Object Points	3	6	O*P (3*Float)
4	Object Vertices	4	5	O*P (3*Float)
5	Minimum Distance	7	1	Float





Prof. Dr. Dominik Henrich, Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme), Universität Bayreuth, http://www.ai3.uni-bayreuth.de