

A System to Navigate a Robot into a Ship Structure

Markus Vincze¹, Minu Ayromlou¹, Carlos Beltran³, Antonios Gasteratos³,
Simon Hoffgaard², Ole Madsen², Wolfgang Ponweiser¹, and Michael Zillich¹

¹Institute of Flexible Automation, Vienna University of Technology
Gusshausstr. 27-29/361, 1040 Vienna, Austria, e-mail: vm@infa.tuwien.ac.at

²Aalborg University, Department of Production, 9220 Aalborg DK

³Lab for Integrated Advanced Robotics of the University of Genova, 16145 Genova, IT

Abstract. A prototype system has been built to navigate a walking robot into a ship structure. The robot is equipped with a stereo head for monocular and stereo vision. From the CAD-model of the ship good viewpoints are selected such that the head can look at locations with sufficient features. The edge features for the views are extracted automatically. The pose of the robot is estimated from the features detected by two vision approaches. One approach searches in the full image for junctions and uses the stereo information to extract 3D information. The other method is monocular and tracks 2D edge features. To achieve robust tracking of the features a model-based tracking approach is enhanced with a method of Edge Projected Integration of Cues (EPIC). EPIC uses object knowledge to select the correct features in real-time. The two vision systems are synchronised by sending the images over a fibre channel network. The pose estimation uses both the 2D and 3D features and locates the robot within a few centimetres over the range of ship cells of several metres. Gyros are used to stabilise the head while the robot moves. The system has been developed within the RobVision project and the results of the final demonstration are given.

1 Project Overview

Robot navigation is a common problem in mobile robotics. In most cases, the problem is considered a 2D problem. The sensor data is projected to the ground plane and then used for path planning and robot control. The task of navigating a climbing robot into a ship structure requires 3D navigation, since the robot shall be also able to climb walls.

The objective of the RobVision project is to develop a vision system that finds and measures the location of 3D structures with respect to a CAD-model for the purpose of robot navigation. The project focused on using the vision system for guiding a robotic vehicle to enable it to navigate and position itself in order to deliver work packages for inspection, welding and other tasks for the structure/body of a large vessel during production. The main motivation for this project is the demand of the end user Odense Shipyard, DK, who are ultimately looking for a robotic operator that can replace human workers to perform the task outlined within a shipyard

environment. Of particular interest is to execute the final welding task at the dock, where conditions for the human worker are exhausting and dangerous.

The integration of a CAD-model to visual measurement and direct feedback of measurement results is the key aspect. In terms of system requirements the major concern is reliability. Other needs are fast operation, sufficient accuracy, autonomous behaviour, automated operation and a simple to use interface to the customer.

Reliability is tackled by developing a method of robust visual finding and tracking by integrating redundant low level image cues and high level object knowledge. Image cues and object knowledge are exploited and integrated both at a local and global level. For the extraction of basic visual cues independent and complimentary modules have been designed.

The paper presents the system layout (Section 2), the main components for feature extraction (Section 3), control of the head and 3D feature measurement (Section 4) and tracking 2D features and pose estimation (Section 5). Section 6 presents part of the demonstrations, which saw the walking robot enter and walk through the vessel structure and gives example results of tracking and pose estimation.

1.1 Related Work

The work of this project is related to navigating mobile robots in indoor environments or grasping parts with a robot. Autonomous robots hold a CAD-map of the building and use landmarks, such as walls or pillars, for navigation (e.g. [15,14,7]). The robot assumes a rough position and matches the landmarks of its map to those detected by the vision system. The main problems are a changing background and high computational demands. For example, a space application where background is dark and the object consists of parts of different surface characteristics, requires dedicated hardware to run at frame rate [23]. Probably the most successful system that uses vision to control a mechanism is the automatic car and air-vehicle approach using dynamic vision [9]. It integrates the dynamic aspects of a continuously operating system and image data to update the model description of the world.

Another series of technique that can be used for 3D navigation relate to object recognition. Object recognition matches image features to features in a data base of multiple objects [11,24]. The match reports object hypotheses, which are subsequently verified to report the most likely object. As a by-product of this process, most approaches report an estimate of the object pose. Impressive results have been shown using edge features (e.g., [11,24,3,6]). However, object recognition suffers from two common problems. (1) Matching requires extensive search and cannot be scaled to operate in real-time for 3D objects of reasonable complexity [4]. Newest results on using indexing [3,6] still require several seconds in simple cases and minutes in more complex images. Therefore most approaches are not used for navigation. An exception is a noticeable work that realises fast indexing by exploiting image and stereo lines, though the authors concede the "reliability-bottleneck" introduced by using one type of feature [7]. And (2), the recognition rates are high under the assumption of good feature extraction. Invariant (to perspective distortion [24] or to illumination [1]) features enable robust recognition, however this requires a solution to the equally difficult problem of robust feature segmentation.

Regarding reliable feature extraction, cue integration has been found in a few approaches as feasible approach (see upcoming book from Dagstuhl Seminar 421 Oct.

2000 in Springer Lecture Notes in Computer Science). An approach studied most closely is voting in cue integration. Voting is a model-free approach and requires a common classification space. Plurality voting gives best results when using four simple blob trackers [PiCh98]. In [16] the authors show that weighted consensus voting of five cues for view-based tracking performs better than a fuzzy fusion method and the single cues. The approach in [2] uses voting to integrate four cues to find planar surfaces but requires a good initial start segmentation to give good results.

2 System Layout

The goal of the project is to define a path and to navigate the robot through this path. Figure 1 shows the system components that have been built to fulfil this task. To summarise, the system works as follows.

From the shipyard's CAD database a 3D model is generated. The model seen in the top left corner of the Figure is a representative example of the typical geometries that can be expected. The model is also similar to the final mock-up that has been built. An operator will then interact with this model to specify intermediary points (IPs), which roughly describe the path along which the robot should move. Whilst each IP describes a pose, additional attributes attached to these IPs describe specific properties applicable to either the robot or the environment. For example, one of these attributes provides the robot with information about what type of walking gait it should adopt (for example, high gait to step over a truss, or narrow gait to step out the hole of the mock-up). For each IP the CAD system searches for good views to look for the vision system. These views present expected views that are seen by the vision system at each respective IP. From these views the CAD to Vision (C2V, developed by AAU) system extracts all significant features and sends them to the two vision components PRONTO and V4R (Vision for Robotics). The goal is to automatically select features that are expected to be robust and can be used by the vision system to reliably calculate pose.

Using the view direction, PRONTO (developed by DIST) controls the head to look at the specified directions. Figure 2 (right) shows the head mounted on the robot. PRONTO also searches for junction features and measures the 3D pose of these features.

The feature list is also used by Vision for Robotics (V4R, developed by INFA) to find 2D line, junction and arc features. To assure that the two vision systems are synchronised, the images from the head are submitted via messages using the fibre channel network from PRONTO to V4R. Due the optical network this is done in a few milliseconds. Furthermore, the two vision systems mutually report about features found to increase the reliability of finding and tracking features.

Finally V4R estimates at each tracking cycle the pose of the head and the pose of the robot with respect to the coordinate system of the ship. Knowing its present pose from the message of V4R and the path described by the IP, the robot can calculate its trajectory and can travers to the next IP and finally the final target pose.

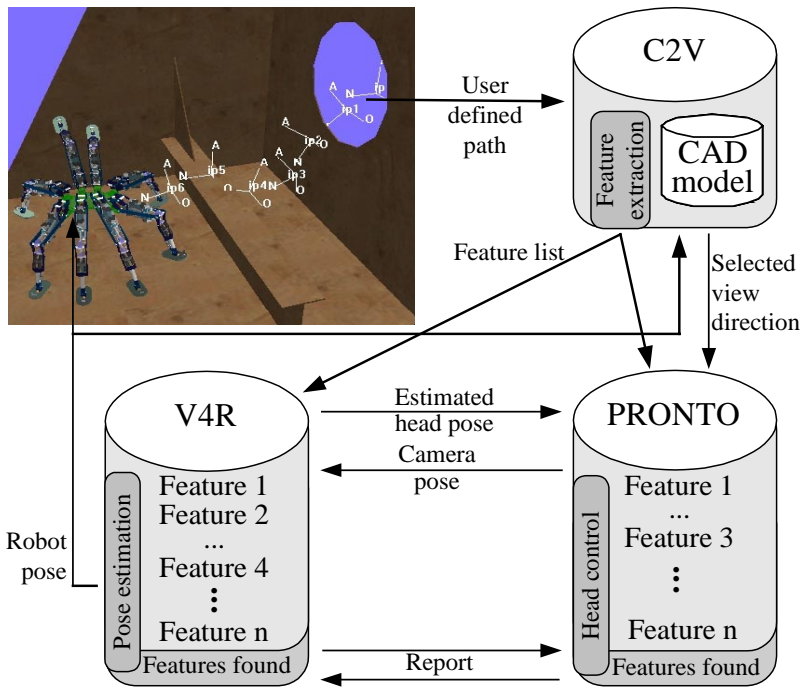


Fig. 1. Principal approach of the RobVision project indicating the main functions of the system components C2V (CAD to Vision, view generation and feature extraction), PRONTO (head control and 3D feature finding) and V4R (Vision for Robotics, 2D feature tracking, pose estimation). In the top left picture, the white co-ordinate systems are the intermediate target poses defined by the user. The trajectory between these intermediate target poses is calculated automatically by V4R. The robot uses this information and the robot pose message of the V4R system to stay on the trajectory.

The robot Robug IV has been developed by Portech, Portsmouth, UK. The robot can carry a weight of 50 kg and is actuated pneumatically. The body of the robot is designed to carry a robot arm for the welding or inspection task. Figure 2 (left) shows the robot at the entrance hole of the mock-up. The right part of the Figure shows the stereo head, called Eurohead, developed by DIST and here mounted on the robot.

2.1 Simulating and Integrating Communication

The stable communication between several components is of utmost importance for systems integration. A communication tool has been developed that provides the *capability to simulate the components*, test each component individually *and to conduct hardware tests in the loop*.

The core of the communication tool is a supervisor process running on a computer with a Windows operating system. The components need to be specified only once. For each component it is then possible to run a simulation or the actual system, which can both run on the same or another computer. The communication protocol utilises

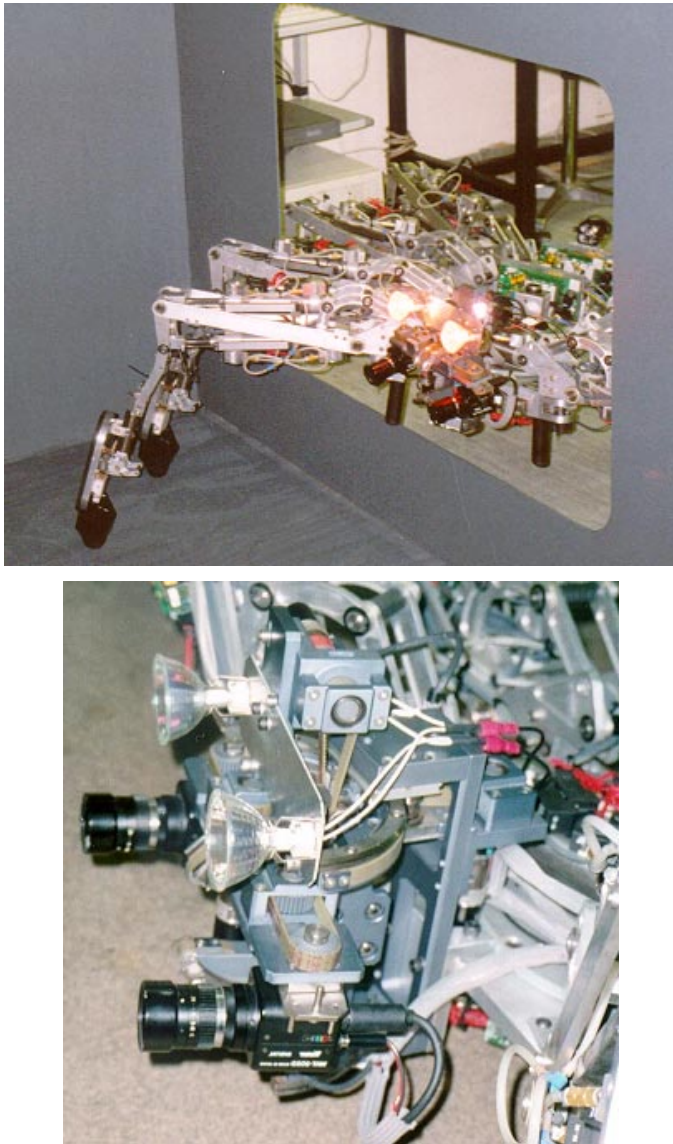


Fig. 2. The Robug IV robot entering the mock-up (left) and a close-up of the Eurohead mounted on the robot front (right).

the TCP/IP standard. Therefore, the components can run on any operating system. The *system components are detected automatically*, therefore no reconfiguration is needed when switching from the simulated component to the actual component.

This communication tool has been successfully applied within RobVision to test the communication of individual components and to enable rapid integration of the components.

2.2 Summary of System Specifications

The project presented a demonstrator to enable autonomous robots in ship building. The approach is flexible, because it can operate in any environment that has distinct features and that is modelled or that can be modelled.

The path of the robot can be planned off-line using the ship model and the *PathPlanner* tool to easily specify the path (for more details on the tools contact the authors). The tool shows the robot and the ship section graphically in one display as the user enters or modifies the position descriptors in a dialog. Each robot position is defined by a pose, a tolerance, and a gait. The tolerance specifies the required accuracy that the robot controller needs to achieve before changing focus to the next position in the path.

The accuracy of the pose estimation of the demonstrator has been 5 to 35 mm for three test series at distances to the features of two to three meters. The pose is continuously estimated along the path in the ship section. The accuracy does not depend on the path length, but on the distance to features at the final location. Therefore this pose accuracy can be reached also after long paths.

Robustness of feature tracking is attempted with appropriate visual processes (Hough transform and a method for Edge Projected Integration of Cues) and the integration of several 2D and 3D line and junction features. The egomotion of the robot is compensated using gyros.

The update cycle time of the pose estimation is presently 120 milliseconds, however, the visual tracking system is capable of lowering this cycle time to 40 milliseconds (if not executing several other tools for system development). The rate of this feedback allows a robot speed of 25 cm/s.

The following sections will now give more detail on the subsystems.

3 Feature Extraction: CAD to Vision (C2V)

The basic idea of C2V is to select good view points, to send new view points along the path, and to provide model and feature information to the vision systems, which then find features in the images and determine pose from the features found. Sending and supervising the present view point operates at a rate of about 1s.

The C2V component *automatically extracts distinct features from the object model* to enable visual navigation. It also automatically evaluates the quality of view points to look at for the vision system. The CAD subsystem concentrates on determining the reference features, which are defined as a coherent set of an intermediate point (IP), the corresponding robust features that should be seen in a view and a robot gait. The features are geometric references determined from the CAD model of the work piece, i.e. surface boundaries and intersections, represented as lines, circles, half circles, junctions (intersections of edges), regions, etc. By robust we mean features, which will not be confused by the vision subsystem when viewed from the specific intermediate point or which are too small to be significant. C2V consists of 3 major systems:

- C2VoffLine (RobCad): C2VoffLine is the part of the AAU system that simulates the motion of the robot and cameras from the path created by the Odense shipyard. During this simulation of the movements the features are collected, which the

camera will to see during the execution of the task (Figure 3 gives an example desk top view). To simplify the system and make it available to a large number of customers, a simplified version has been developed: C2VoffLine (Windows NT), described next.

- C2VoffLine (Windows NT): The kernel (C2Vkernel) used in the RobCad version is implemented in an NT-based version of C2VoffLine. The NT version does not have all the functionality and the same degree of automation, which is found in the RobCad solution, but this version is fully capable of creating features for demos and can be used by any user.
- C2VonLine: C2VonLine is the specially designed communication software of C2V that communicates with the entire RobVision network at the demanded rate. The software is used to send Models and Features generated by C2VoffLine depending on the present pose of the robot.

The loop between the vision subsystem and the CAD subsystem will then be as follows:

1. After a small robot motion, the knowledge of the former pose and the direction of the movement the vision subsystem predicts the robot pose, which is also passed on to the CAD subsystem.
2. The CAD subsystem provides 3D geometric features detected in the CAD model from these poses and the present view of the camera(s) if the view changed considerably due to the robot movement.
3. If the features or images are recognised by the vision subsystem, a the view direction is kept and the robot moves towards its target. Go to 2 and continue the loop from here.
4. If the features or images are not recognised by the vision subsystem, the cameras move to another direction of view. This view point is suggested by the CAD subsystem evaluating areas that are rich with features to render the task of the vision system more easy.
5. Go to 3 and continue the loop from here.

The model just described is very generic, which enable the use to track any object that is represented with a CAD model. Details on the method to extract features can be found in [Del00].

4 Head Control and 3D Feature Measurement: PRONTO

The PRONTO system fulfils several tasks, which in brief are:

- Head control and calibration.
- Inertial sensor data acquisition, distribution to other systems and use for head stabilization.
- Image acquisition and delivery to other systems and image processing.
- Communications and synchronization with V4R vision system.

PRONTO, therefore, performs a large number of complex and interrelated task that need to be synchronized. For these reasons the software is implemented in C++ as a *multithread object oriented distributed application* with the aid of the distributed programming technology DCOM (Distributed Component Object Model) to create a

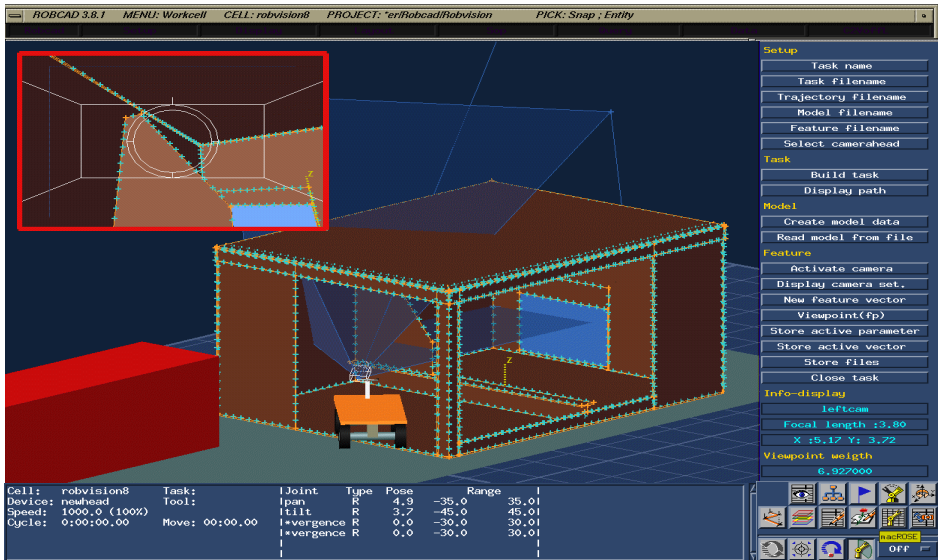


Fig. 3. Example of mock-up in the C2Voffline system. For part of the demonstration a cart was used since the robot was not always available. The top left is the view as seen from the left camera of the stereo head. From this view features are extracted automatically.

software architecture that tries to simplify the complexity of this subsystem. From the hardware point of view PRONTO consists of a two processors computer running Windows NT and, of course the Eurohead (the head is shown in Figure 2 above)

The accuracy of the Eurohead has been evaluated in detail in [10]. To find and measure the 3D junctions a Hough technique is used to extract the lines on the image planes of a stereo pair using the features from C2V. The extracted lines and the junctions are related to the CAD model using a weighted least mean squares method. Then a closed loop method follows, such that by simultaneously moving the three degrees of freedom of the head the junction is fixated at the principal point of the image in both images. When this is the case the two cameras are verging on the certain junction and the direct kinematics of the head are applied in order to determine the 3D position of the junction relative to the head.

5 2D Feature Tracking and Pose Estimation: Vision for Robotics (V4R)

The task of the vision system is to extract features from the cues of images and relate them to the features provided by C2V. C2V provides geometric features such as line, arc, junctions, region, and attributes connected to these features. The attributes of a line can be, for example, welded or not welded, chamfered, or rounded. Regions can have attributes such as intensity, texture, or colour.

The V4R software package is a well tested prototype, which is available to interested researchers from the authors. V4R provides a tool for tracking using images from video (mpeg), live camera, or image sequences. V4R contains two major components, which can be exploited separately: (1) *framework for tracking of features* and (2) *pose estimation* using a model and the feature found in the image.

The tracking tool within V4R is capable of following line, junction and ellipse features at field rate. The tracking method is edge based and uses a scheme of Edge Projected Integration of Cues (EPIC) to obtain robustness against varying background and continuous illumination changes [20]. The goal of tracking is to be able to follow fast motions. Therefore fast cycle rate and a windowing approach have been adopted as formally derived in by considering the dynamics of the robot - vision system [17]. The entire vision system of INFA is designed in C++ and presents a generic structure for any model-based vision method [20]. Its development pays tribute to the developments of the XVision system by Greg Hager [12].

The pose estimation tool of V4R uses the object model and the features found in the image to determine an optimal pose. The following features are utilised for pose estimation: line, 2D point (junction), 3D point, and surface normal. Outliers are detected and a least squares procedure over all remaining features gives a best pose fit.

Figure 4a shows the windows projected into the image. Yellow lines present found features, red line features not found. Figure 4b gives the pose found in green as a re-projection into the image.

5.1 Performance Evaluation of 2D Feature Finding Methods

To track the edges (lines, ellipses) a method for cue integration (Edge Projected Integration of Cues, EPIC) is used, which has been proposed in [20]. EPIC uses cues (intensity, colour, texture, ...) to distinguish object from background edgels. However, the initialisation of the feature cannot use this information and is therefore the most critical event.

For first finding the edge feature several methods can be used. This section introduces a comparison of methods to find the edges. The next section then outlines how topological knowledge from the model can be used to discriminate candidate edges.

During the project the following methods have been evaluated when the model is first projected into the image. The two references are tracking, where previous information can be taken into account. This is simpler than finding the feature for the first time. The second reference is only using the edge information from a classical edge extractor. For all methods the warped image approach of [12] has been used.

1. Only-edge: Edge finding only using an edge filter as in [12] and least squares fit to find line.
2. Tracking: Using information from a simulated previous finding.
3. LMedS: uses zero crossings to find edgels and a Least Median Square regression to fit the line.
4. EPIC-centre: Integrating the Using the centre location as most likely source to indicate edgels.

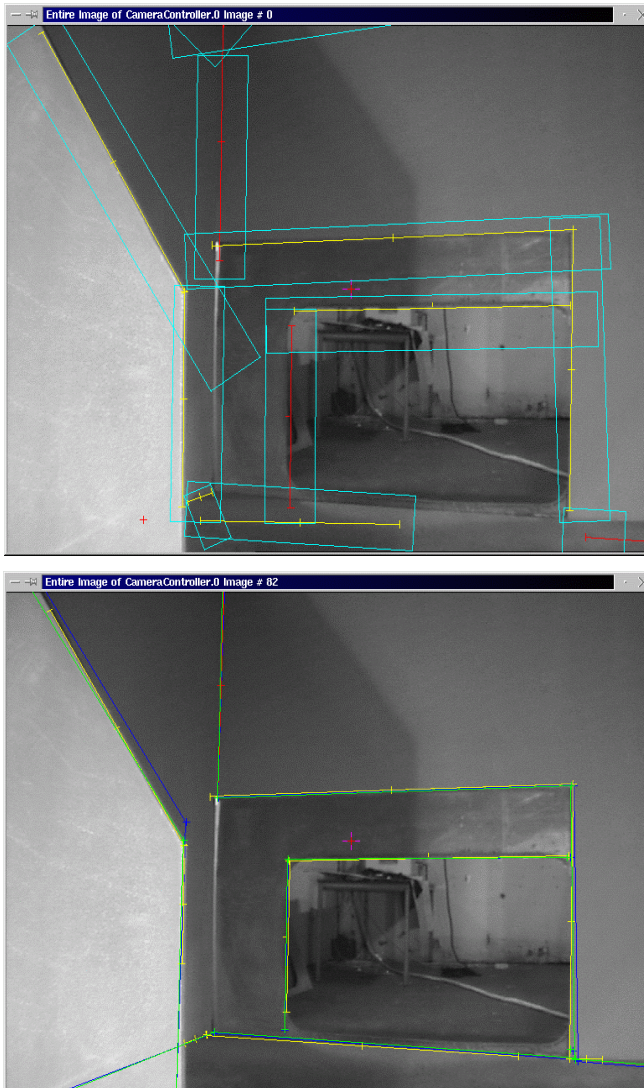


Fig. 4. Top: Tracking windows to search for features. Bottom: Blue is the projection into the image, yellow features found, red are features not found (due to poor contrast in this case), and green is the re-projection of the estimated pose into the image.

The methods have been evaluated on real images of typical ship sections. Figure 5 shows a typical example. The detection of features has been tested by moving the images in the horizontal and vertical axis. This renders the initialisation inaccurate by a known range covering plus and minus 30 pixels to the correct centre position.

The four algorithms have been tested. The results are summarised in Figure 6. The detection rate gives the percentage of lines found of all lines initialised. Each data point corresponds to about 120 lines in two dozen images. The next figure then shows

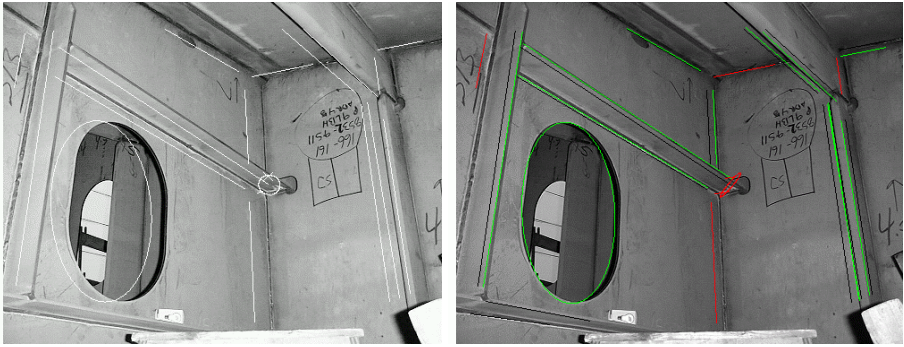


Fig. 5. Initialisation (left) and detection results (right) when horizontally displacing the image of the ship section in ten pixel steps. The EPIC-centre algorithm is used in this example.

the percentage of the detected lines that has been detected correctly. It shows that the Only-edge method detects many lines but also many incorrect lines. The LMedS improves the performance, but adding the centre information is better even if the displacement becomes larger.

A significant problem is the false detection of feature due to ambiguities in the image. The introduction of probabilistic likelihood values to evaluate the quality of a line did not proof successful. Lines have been found with many edgels along the line. Therefore other measures has to be taken. The basic approach is to integrate the topological knowledge from the CAD model.

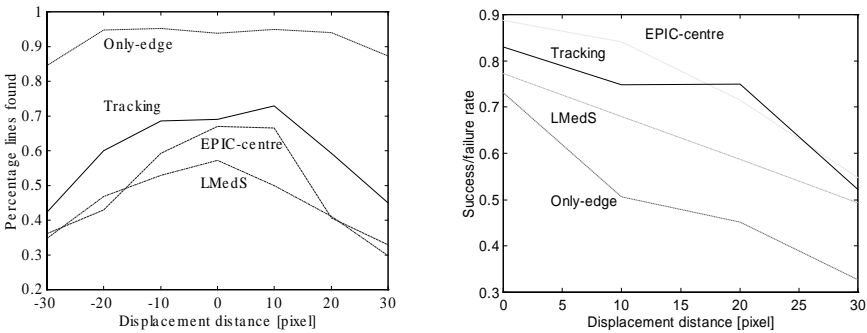


Fig. 6. The detection rate (left) and the success/failure rate of the "Only-edge", the "LMedS", the "EPIC-centre" and the "Tracking" algorithms over the distance relative to correct localisation. The S/F rate is given by the relation of correctly versus falsely found features over the features found.

5.2 Integrating Object Topology

Double lines cause severe problems due to ambiguity. Displacements of junctions on both image planes due to either wrong estimation of the robot position or bad construction also causes problems.

On the other hand, exactly this information can be exploited to discriminate features: parallel lines and lines intersecting at a junctions must show some specific properties. The use of this topological information is referred to as validation. It helps to disambiguate local ambiguities.

Figure 7 shows a tracking example and also an example of using parallelity information. The bottom front of the images shows the T-truss with three parallel line. By defining a “parallel line“ relation between two lines, the lines can be ordered locally. By applying the method two times (to the top pair of lines and the bottom pair of lines, the centre line of the three lines is an element of both pairs) the lines are discriminated most of time. Therefore the ends of lines in a tracking window need to be found. This seems to be the most feasible extension and should be considered for implementation.

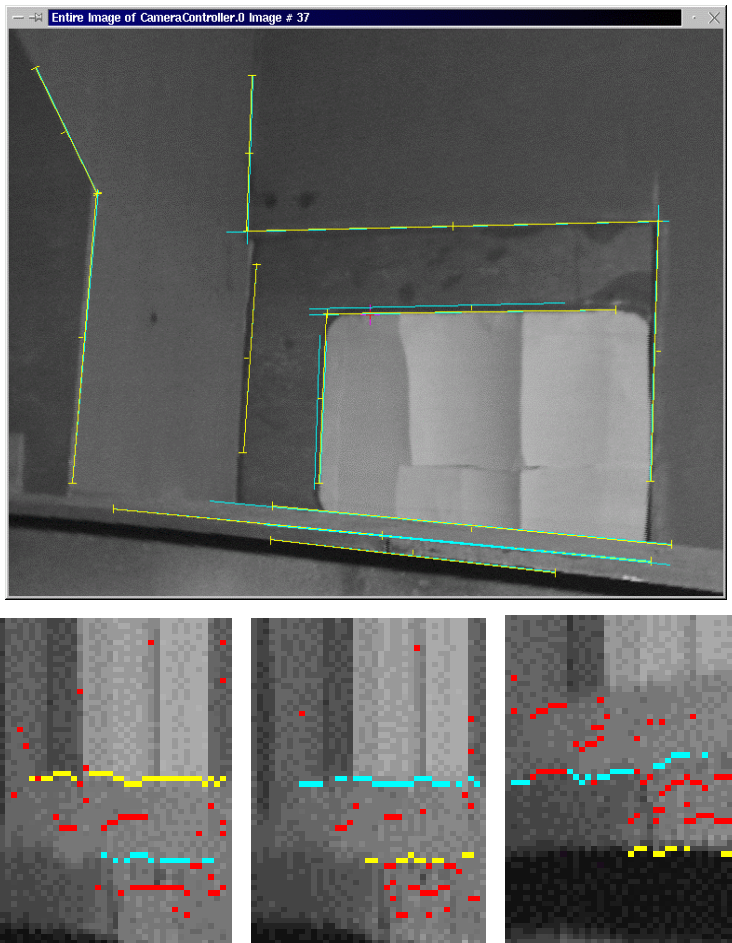


Fig. 7. Discriminating between close parallel lines. The small windows in the bottom line show the warped tracking images of the three lines from top to bottom in the above image. Yellow indicates the selected edge, cyan a candidate, red are all edges found.

A second method is to use the information of a junction. When intersecting potential candidate lines at one junction, only the two lines that form the junction will lead up to the junction point. This can be validated locally in the image without requiring expensive computations.

Using these simple validation methods it is possible to track complex scenes as given here with a standard Pentium PC 500 MHz at frame rate. In the project only 120 ms have been used to run several other tools to improve the system performance.

6 Demonstration Results

The final demonstration of the RobVision project brought together all of the tools to demonstrate the integrated system at the demonstration site at Odense Shipyard in Odense, DK. Figure 8 gives an example of the tracking performance from the robot. The robot moves towards its goal and the vision system tracks the features to estimate the robot pose continuously.

The motion of the robot is depicted in Figure 9. The jerky motion of the robot can be clearly seen. While tracking is fast and shows good robustness, the maximum allowed object motion is still restricted. The use of inertial sensors proved effective to compensate fast angular motions. This aspect will be investigated more closely by integrating a full 6D inertial sensor suit with the optical tracking methods.

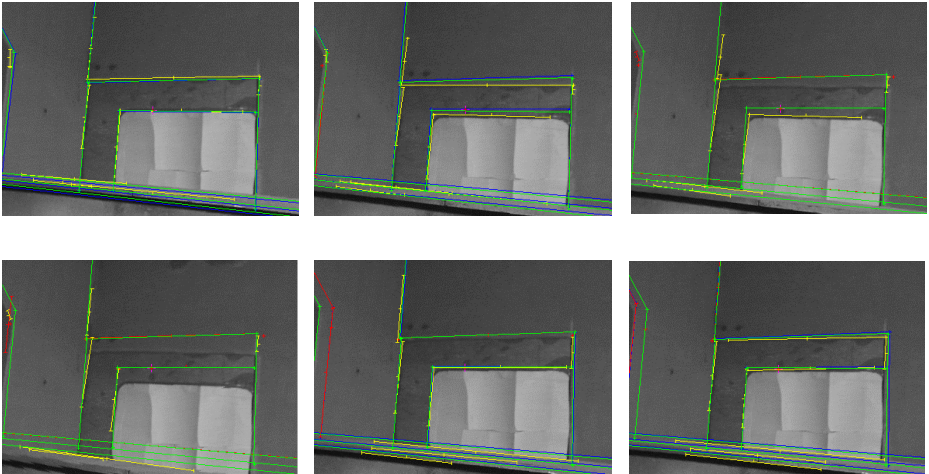


Fig. 8. 6 images from a sequence of more than 300 images of tracking from the moving robot. The colours are used as in Figure 4. Note that a short jerk of the robot body due to its pneumatic actuators is compensated after two cycles and tacking regained.

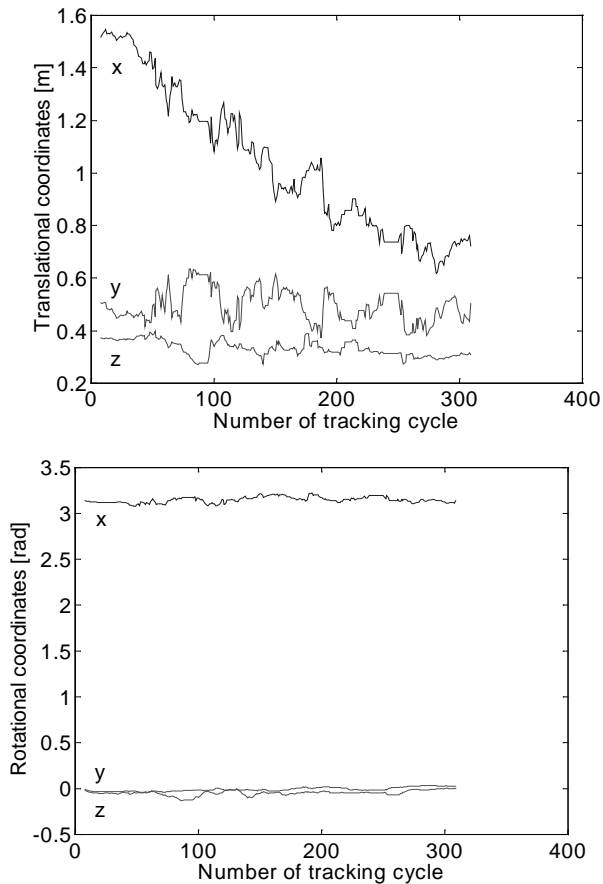


Fig. 9. The result of pose estimation during a tracking sequence from the walking robot.

7 Summary of Results

The objective of the RobVision project is to navigate a walking robot into ship sections using visual feedback. The system consists of the components walking robot, CAD-system, and two redundant vision systems that provide the feedback to steer the robot. The system demonstrated the following.

- A modular walker is able to operate with 6 or 8 legs without any hardware or software changes and can carry a work package of 50 kg into the cell.
- The CAD-model can select, evaluate and deliver significant features that the cameras on the robot should see along the path of the robot.
- The redundancy of two vision systems can be used to improve the reliability to find and track features.

- Features such as lines, 2D and 3D points and junctions, ellipses, or surface normals can be integrated with one algorithm to determine the pose of the object.
- Image processing is executed in 120 ms (and can be improved to obtain frame rate, 40 ms), which allows fast real-time operation.
- The approach is ready to be applied to locate and measure any object that has been described with a standard CAD-system.

This technique opens up other potential applications. The model-based approach enables to measure any modelled object and to feed back the measurement data directly into the CAD-system. The impact of this capability is manifest. The industrial partners in the ship building and the construction industry can supervise the production quality and consequently reduce production time.

Acknowledgements. This work has been mainly supported by the RobVision project Esprit 28867 and is partly supported by the Austrian Science Foundation (FWF) under grant P13167-MAT and Project GRD1-1999-10693 FlexPaint.

References

- [1] Alferez, R., Wang, Y.F.: Geometric and Illumination Invariants for Object Recognition, IEEE Transactions on PAMI 21(6), 1999, pp. 505-536.
- [2] C. G. Bräutigam: A Model-Free Voting Approach to Cue Integration, Dissertation, KTH Stockholm, 1998.
- [3] J.S. Beis, D.G.Lowe, Indexing without Invariants in 3D Object Recognition, Pattern Analysis and Machine Intelligence, Vol.21, No.10, 1999, pp. 1000 - 1015.
- [4] J.R. Beveridge, E.M. Riseman, How Easy is Matching 2D Line Models Using Local Search?, Pattern Analysis and Machine Intelligence, Vol.19, No.19, 1997, pp.564 - 579.
- [5] H.I. Christensen, D. Kragic: Robust Vision for Manipulation and Navigation, in: Markus Vincze (ed.): Proc. Robust Vision for Industrial Applications 1999, 23rd Workshop ÖAGM/AAPR.
- [Del00] Deliverable of the RobVision project, also available from robvision.inf.tuwien.ac.at, 2000.
- [6] Dickinson, S.J., Wilkes, D., Tsotsos, J.K.: A Computational Model of View Degeneracy, IEEE Transactions on PAMI, vol.21, no.8, 1999, 673-689.
- [7] C. Eberst, M. Barth, et.al., Robust Vision-based Object Recognition Integrating Highly Redundant Cues for Indexing and Verification, IEEE ICRA 2000, pp. 3757-3764.
- [8] Förstner, W. and Gülch, E., (1987): A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features; ISPRS Intercommission Workshop, Interlaken.
- [9] Fuerst, S., Dickmanns, E.D.: A Vision Based Navigation System for Autonomous Aircraft; Robotics and Autonomous Systems 28, pp.173-184, 1999.
- [10] A. Gasteratos and G. Sandini, On the Accuracy of the Eurohead, LIRA – TR 2/00, July 2000.
- [11] Grimson, W.E.L.: Object Recognition by Computer, MIT Press, 1990.
- [12] Hager, G. and Toyama, K. (1998): The XVision-System: A Portable Substrate for Real-Time Vision Applications, Computer Vision and Image Understanding, 69 (1), 23-37.
- [13] Garding, J. and Lindeberg, T. (1994): Direct Estimation of Local Surface Shape in a Fixating Binocular Vision System, in: J. O. Eklundh, Lecture Notes in Comp. Science, 800, 365-376.

- [14] D.S. Kim, R. Nevatia, Recognition and localization of generic objects for indoor navigation using functionality, *Image and Vision Computing* 16(11), Special Issue SI, 1998 Aug 1, 729-743.
- [15] Kosaka, A., Nakazawa, G.: Vision-Based Motion Tracking of Rigid Objects Using Prediction of Uncertainties; *ICRA*, pp.2637-2644, 1995.
- [16] D. Kragic, H.I. Christensen, Cue Integration for Manipulation, in [21], pp. 1 - 16.
- [17] P. Krautgartner, M. Vincze: Optimal Image Processing Architecture for Active Vision Systems; *Int. Conf. on Vision Systems, Gran Canaria, S. 331-347, January 13-15, 1999.*
- [18] P. Pirjanian, H.I. Christensen, J.A. Fayman, Application of Voting to fusion of Purposive Modules: An Experimental Investigation, *Robotics & Autonomous Sys.*, Vol.23, 1998, pp. 253-266.
- [19] Smith, S. M. and Brady, J. M. (1997): SUSAN - a new approach to low level image processing, *Int. Journal of Computer Vision*, 23, (1), 45-78.
- [20] M. Vincze, M. Ayromlou, W. Kubinger: An Integrating Framework for Robust Real-Time 3D Object Tracking; *Int. Conf. on Vision Systems, Gran Canaria, S. 135-150, January 13-15, 1999.*
- [21] M. Vincze, G.D. Hager, Eds., *Robust Vision for Vision-Based Control of Motion*, IEEE Press, 2000.
- [22] M. Vincze: Robust Tracking of Ellipses at Frame Rate; *Pattern Recognition* 34(2), 487-498, 2001.
- [23] Wunsch, P., Hirzinger, G.: Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion; *ICRA*, pp.2868-2873, 1997.
- [24] A. Zisserman, D. Forsyth, et.al., 3D object recognition using invariance, *Artificial Intelligence*, Vol.78, 1995, pp. 239 - 288.