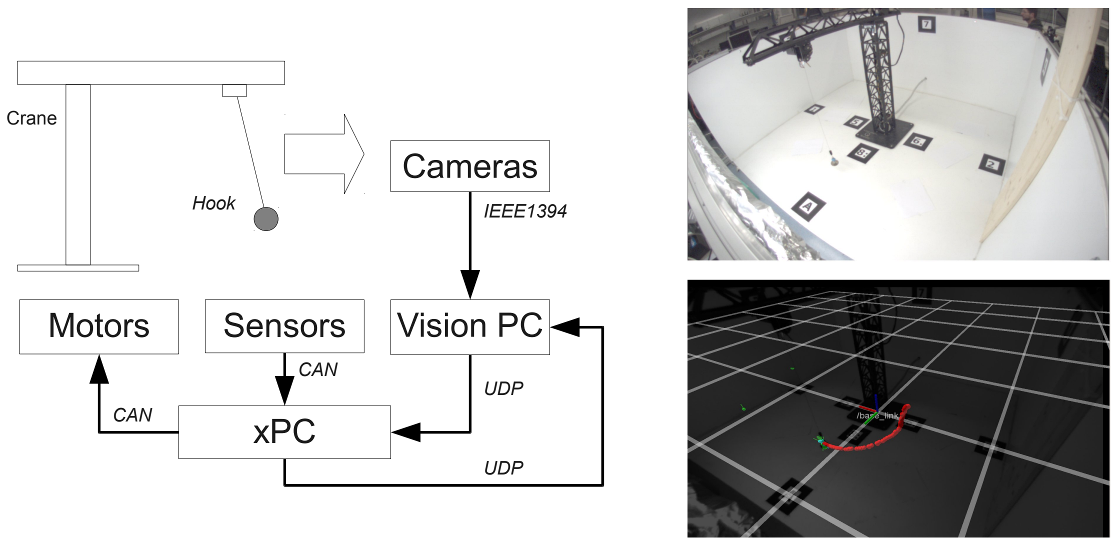
A Visual Servo Control Overhead Crane

Markus Bader, Markus Vincze
Automation and Control Institute, ACIN,
Gusshausstrasse 27-29 / E376, 1040 Vienna, Austria
markus.bader@tuwien.ac.at, markus.vincze@tuwien.ac.at

1 Introduction

Steering an overhead crane with 5 degrees-of-freedom[1] (DOF) to move loads from A to B is a complex task, which is normally done by especially trained personal. This paper extends work done on the crane's real time control [1] [2] and [3] towards a visual servo control system [4] [5] to realize a semi autonomous crane. Semi autonomous means in this context that the system is aware of its environment and able to pick and place arbitrary objects without collision. The key to realize such a system lies in the accurate visual detection and the estimation of the crane's hook trajectory.
An accurate estimate and prediction of the hook's trajectory is essential to overcome time delays of computer vision applications. These delays are caused due to different transport protocols and the time needed to process the image data. In our show case the crane's real time control runs with a cycle time of 2ms, but the vision system is triggered by the cameras frame rate (33ms) and has a constant latency due to the data transport. Figure 1(a) shows the involved components and the protocols used to transfer data between them.
This paper presents a Monte Carlo localization (MCL) filter [6] to compensate false positives on the visual detection and to estimate the state of the crane's physical model including the free swinging conical pendulum [7]. Figure 1(b) top shows a distorted camera image and the image below shows the same view undistorted with the projected coordinate system, tracked path and the particles of the MCL.



(a) Block diagram of the system　(b) Camera images with overlay

Figure 1: Crane with 5 DOF

[1]Crane with 5 DOF: The rotation around the rope was not counted

2 Related Work and Set-Up

Different projects have dealt with the control of an overhead crane. The most relevant work is [8] and [9] presenting a cable array robot and simulator equipped with cameras but, unlike to our system, all DOFs where actuated.
The presented set-up uses cameras to determinate the hook's position and obstacles in the environment. Visual markers[2] [8] also called *fiducials* are placed in the environment to determine the camera extrinsic parameters automatically and to detect obstacles which are currently simplified to boxes tagged with fiducials. The fast low-level control was implemented in Matlab for an xPC which is able to control the system with an update rate of 2ms. The vision system uses ROS [10] and ROS messages to connect the different in- and outputs of the vision algorithm and camera driver. A special designed ROS node manages the communicates with the xPC over simple UDP packages. The crane is therefore able to steer around obstacles by using an A* algorithm [11] running on the xPC. But the system relies on prior knowledge of the markers identity and needs to undergo an initialization phase to ground the relative measurements of the system. The presented detection and tracking algorithm will close the control loop and overcome measurement errors caused during the initialization phase by calibrating the system automatically.

3 Hook detection and filter

The hook detection quality can be increased in two ways: by increasing the visual detection quality and by increasing the underlying motion model to track the current state of the system. Object structure based computer vision algorithm like [12] or [13] can be used to increase the detection quality but they will also lead to a higher computational cost. The idea is to use a simple color based blob detection algorithm [14] and to increase the pose estimation by incorporating the physical model in the filter used to track the systems state. Tracking the system state using a particle filter will additionally overcome typical false positives caused by the color based detection algorithm. Figure 2 shows particles around the hook pointing towards the estimated trajectory. We can also see that the majority of particles are pointing to the current moving direction of the hook.

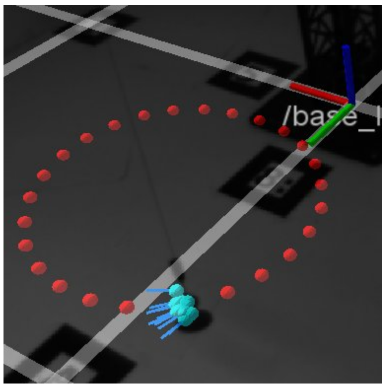


Figure 2: Particles

The motion model used is based on the mathematics of a pendulum [7] and incorporates the control commands generated by the low-level controller. A gaussian isotropic error is assumed and modelled on the visual detection. This error function and the underlying ROS framework allows to extend the system to multiple cameras to increase the accuracy and to fill blind spots. The developed camera pose estimation node can also cope with moving cameras as long as enough fiducials are in the cameras field of view.
In the future we will extend the work towards a better visual detection and towards dealing with arbitrary obstacles within the environment. This will lead us to our final goal: to place loads on moving objects and to initialize the system from an arbitrary state.

[2] ARToolKit: `http://www.hitl.washington.edu/artoolkit/`

References

[1] K. Graichen, M. Egretzberger, and A. Kugi, "Suboptimal model predictive control of a laboratory crane," *8th IFAC Symposium on Nonlinear Control Systems,*, pp. 397 – 402, 2010.

[2] K. Graichen and A. Kugi, "Stabilitt von inkrementellen modellprdiktiven Regelungsverfahren ohne Endbeschrnkungen," *Talk: GMA-Fachausschuss 1.40 "Theoretische Verfahren der Regelungstechnik", Anif/Salzburg; 09-20-2009 - 09-23-2009; in: "Tagungsband GMA-Fachausschuss 1.40 "Theoretische Verfahren der Regelungstechnik""*, pp. 219 – 230, 2009.

[3] M. Egretzberger, K. Graichen, and A. Kugi, "Flatnessbased mpc and global path planning towards cognitionsupported pickandplace tasks of tower cranes," 2010.

[4] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *Robotics and Automation, IEEE Transactions on*, vol. 12, pp. 651 –670, Oct. 1996.

[5] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82 –90, 2006.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[7] D. C. Giancoli, *Physics for Scientists and Engineers with Modern Physics and MasteringPhysics(R) (4th Edition).* Prentice Hall, 4 ed., 9 2008.

[8] K. Usher, G. J. Winstanley, P. I. Corke, D. Stauffacher, and R. Carnie, "Air vehicle simulator: an application for a cable array robot," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2241 – 2246, 2005.

[9] K. Usher, G. Winstanley, and P. Corke., "A cable-array robot for air vehicle simulation," in *Proceedings of the Australian Conference on Robotics and Automation, Canberra, Australia*, December 2004. ISBN: 0958758360.

[10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[11] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100 –107, 1968.

[12] C. Kimme, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Commun. ACM*, vol. 18, no. 2, pp. 120–122, 1975.

[13] M. Bader, "Feature-based real-time stereo vision on a dual core dsp with an object detection algorithm," Master's thesis, Vienna University of Technology, 2007.

[14] J. Bruce, J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)* (T. Balch, ed.), vol. 3, pp. 2061–2066 vol.3, 2000.