Automatic Robot Supervision within a Lunar Crater Environment

Alexander Dettmann, Stefan Haase, Frank Kirchner
DFKI GmbH – Robotics Innovation Center, Robert-Hooke-Str. 5, 28359 Bremen, Germany

Abstract

The DFKI RIC built an artificial lunar crater scenery to create an experiment environment for exploration robots. This test facility called *Space Testbed* is equipped with a wide range of supervision tools. The presented paper describes a concept to control these tools automatically in order to support the operators. For this purpose, a motion tracking system is used to focus the Pan-Tilt-Zoom cameras on the robot. Furthermore, a gantry crane is controlled by a tracking algorithm to keep a vertical position above the robot to allow a constant top view.

1 Introduction

Robotic missions are identified as important precursors for human missions on Mars and Moon [1]. The aims of such missions are exploration of landing sites, building up infrastructure, and installing In-Situ Resource Utilization (ISRU) units [2]. In this context, water is an important resource which can be found as ice inside lunar craters at polar regions[1]. This is because they are permanently shadowed due to the steep slopes and the flat angle of incidence of sunlight. One robot mission could be to climb into the interior of such a crater and gather that crucial resource. During that mission the robots have to cope with different adversities such as overcoming steep slopes and dealing with very bright light conditions outside the crater as well as with complete darkness inside. To test those abilities, the *Space Testbed* (STB) was built in the context of the LUNARES project [3].
The STB is an artificial lunar crater environment which provides slopes between 30° and 45° (see **Figure 1**). Its surface consists of hard rocks with gray basalt chips as regolith substitute including stones and mini craters. In addition, a lighting system is installed that is able to create very bright spots and complete darkness in order to simulate the difficult light conditions at the lunar polar regions. The STB is equipped with a wide range of supervision tools to acquire experiment data for evaluations of the tested systems and to enable the operators to control the robots. The presented paper describes an approach to automatically control the tools mentioned below. This supports the operators to concentrate on the mission progress and on the robot control. The automatic supervision system consists of:

- a Motion Tracking System (MTS),
- two Pan-Tilt-Zoom cameras (PTZ),
- and a gantry crane with an observation camera and a tracking camera attached.

[1] proven by NASA LCROSS mission http://lcross.arc.nasa.gov/

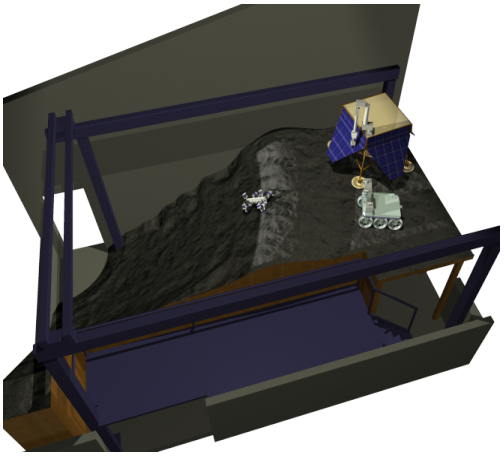


Figure 1: CAD-Model of STB

The gantry crane is included in the STB to maintain a constant top view of the tested robot. In addition, the gantry crane is used to simulate lunar-like gravity force by vertically lifting up the robot using deflection rollers and a constant weight. In either role, the gantry crane has to be positioned as vertical as possible above the robot and thus trace each robot movement. As well as the gantry crane, the PTZ cameras also have the goal to constantly keep the robot in focus. But object recognition via image processing has to cope with several difficulties due to the lighting conditions in the STB (see **Figure 2**). The robot's color is very similar to the ground, and its contour is complex and variable. Therefore color- and contour-based approaches are not promising. Furthermore, additional light sources cannot be used to improve the scenery because it would influence the simulated lunar conditions and degrade the demonstration. So concepts had to be developed for the automatic supervision of the PTZ cameras (section 2) and for the automatic position control of the robot (section 3).

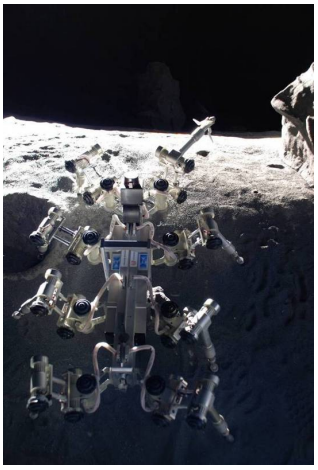


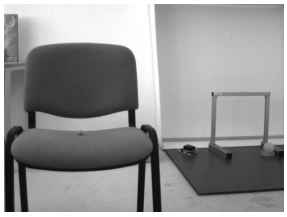
Figure 2: Top view of the tested robot in the STB

2 Automatic Supervision Using Pan-Tilt-Zoom Cameras Controlled by a Motion Tracking System

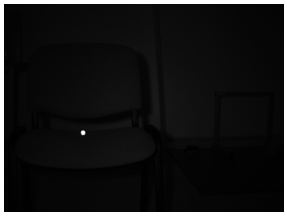
The STB is equipped with a Qualisys motion tracking system consisting of ProReflex MCU1000[2]. This system allows the position determination of reflective markers which are attached to the tested robotic system. Thus, the movement of the robot inside the crater can be tracked and analyzed. Furthermore, AXIS 215 PTZ[3] cameras are installed whose orientation can be set by pan- and tilt-angles. The tracking system works robustly and accurately to the millimeter and therefore is used to direct the AXIS cameras to the robot. In this way, a robust camera alignment can be realized which can hardly be influenced by occlusion, additional movements, light sources, and blurred camera images, which could occur because of the dusty environment within the STB and the autofocus functionality of the AXIS cameras.
The realization of the camera alignment using the motion tracking system consists of three working steps: First, an algorithm is implemented to detect the reflective markers within the camera image reliably (section 2.1). This allows the collection of passpoints which consist of a 3D point - representing the marker position in the Qualisys coordinate system[4](WCS) - and its corresponding pixel in the camera image. Second, the collected passpoints are used to calculate position and orientation of the camera in the WCS using the optimization algorithm CMA-ES [4] (section 2.2). The known camera pose allows the transformation of 3D marker points into the camera coordinate system (CCS). Third, a mapping function based on spherical coordinates is created to align the camera to the transferred 3D points (section 2.3).

2.1 Detection Algorithm

The collection of passpoints requires a reliable detection of the spherical reflective markers used in the STB, because even a single wrong passpoint in a set of passpoints could lead to a wrong calculation of the camera position and orientation. To simplify the detection, the AXIS 215 PTZ cameras are equipped with an infrared emitter. Additionally, the infrared cut filters of the cameras are switched off, the exposure time is reduced, and the movement and zoom of the PTZ cameras are disabled during the marker detection. This reduces the complexity of the marker detection (see **Figure 3**).



(a) Marker in unmodified scene



(b) Marker in scene with activated infrared emitter and reduced exposure time

Figure 3: Scene improvement

During the marker detection, a run-length encoding algorithm binarizes the incoming images and builds horizontal intervals of pixels over a certain gray value, whereas the intervals have to provide a minimal length l_min. The minimal length of the intervals is used to separate the shiny marker from the dark surroundings and to delete white scattered disturbances. Next, a union-find algorithm with pass compress merges connected intervals to regions [5]. Since there is the eventuality of several unwanted regions in the image caused by unintended reflections, the region characteristics have to be calculated. The two axes of inertia and the size of the region are good characteristics to filter out the expected circular image of the spherical marker. The inertial axes describe an ellipse consisting of a short radius, a long radius, and the angle of the larger inertial extension. The roundness r is computed by dividing the large by the small radius. If the roundness of a region lies below r_lim, it will be added to the collection of valid markers. If no appropriate regions can be found, the algorithm takes less brighter pixels into account until the lowest threshold t_lim is reached. A passpoint will not be created if too many valid regions are found.
The optimal values for r_lim and t_lim in the STB are determined experimentally. Therefor markers with diameters of 19 mm and 40 mm are placed at a distance of 2 m and 6 m in the STB, and the roundness of the detected region is calculated. The highest roundness can be expected if the pixel-brightness threshold is set to the darkest pixels

[2]http://qualisys46.kaigan.se/archive/product_information_pdf/AN_ProReflex.pdf
[3]http://www.axis.com/files/datasheet/ds_215ptz_34462_en_0902_lo.pdf
[4]World Coordinate System

of the marker. It has shown that this minimal brightness lies around 200[5] (see **Figure 4**). Below this threshold the roundness value r increases because pixels are used which do not belong to the marker. Furthermore, better results can be achieved by using the larger marker, especially at long distances. On the basis of this experiment, r_lim was set to 1.21 and t_lim to 203.

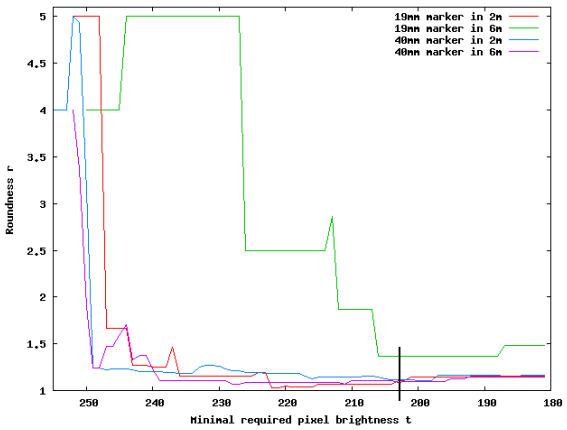


Figure 4: Roundness of the region depending on pixel brightness

The implemented detection algorithm proved to be very reliable. Nevertheless, for every added passpoint an image is stored in which the detected region is marked by drawing its position, orientation, and the radii of the ellipse. This allows later validation of the detected marker and sorting out of wrongly added passpoints. The popular Hough Circle Transform does not work well in this case because situations could occur in which the marker covers an insufficient number of pixels due to the long distance to the camera [6].

2.2 Camera Position

The calculation of the camera positions consists of three further working steps. First, the camera calibration function is composed which transfers 3D points from the WCS into pixel coordinates of the camera image. Second, a cost function is created which rates the current extrinsic parameters on the basis of the collected passpoints. Third, the optimal camera position is calculated using CMA-ES. The function is adapted to the right-handed coordinate system shown in **Figure 7**.

Camera Calibration Function The camera calibration function is used to transfer the 3D points of the passpoints from the WCS into the CCS and further into the camera image. The camera calibration function contains the intrinsic and extrinsic parameters. The intrinsic matrix describes the internal camera parameters like focal length and CCD-chip resolution. The 3x4 extrinsic matrix T describes the position and orientation of the camera in the WCS. In this work, the intrinsic parameters are taken from the data sheet to reduce the dimension during the optimization (see **Table 1**). Therefore, only three rotational and three translational parameters of the extrinsic matrix have to be considered.

Intrinsic Parameter	Parameter Value
Focal Distance f	3,8 to 46mm
Chip-Dimensions	3,6 x 2,7mm
Resolution	704 x 576 Pixel
Chip-Center c_y,c_z	352 x 288 Pixel
Pixelwidth d_y	3,6mm / 704 Pixel
Pixelheight d_z	2,7mm / 576 Pixel
Focalwidth f_y	f/d_y
Focalheight f_z	f/d_z

Table 1: Intrinsic parameters of the AXIS 215 PTZ

The extrinsic matrix T is equivalent to the transformation matrix T_{WCS}^{CCS} which transfers a point from the CCS to the WCS (1).

$$T = \begin{bmatrix} & & tx \\ & R & ty \\ & & tz \end{bmatrix}, \text{with } R = R_x R_y R_z \qquad (1)$$

During optimization, the points have to be transferred from WCS into CCS which can be achieved by the inverted matrix T^{-1} (T_{CCS}^{WCS}). In (2) (X, Y, Z) are the coordinates of a 3D point in the WCS and (x, y, z) describes a 3D point in the CCS.

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = T^{-1} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad (2)$$

After the transformation, the intrinsic parameters are used to calculate the equivalent pixel (u, v) in the camera image (3), where c_y, c_z describes the center of the CCD-chip and f_y, f_z the focal length in pixel-related units.

$$u = f_y \cdot \frac{y}{x} + c_y \qquad v = f_z \cdot \frac{z}{x} + c_z \qquad (3)$$

Cost Function The cost function uses the camera calibration function to transfer the 3D point of each passpoint into the camera image and compares the calculated image coordinates with the desired ones. Thus a set of extrinsic parameters is rated by the average deviation over all available passpoints in pixel-related units. This rating is represented by the fitness value which has to be minimized via optimization. To speed up the optimization, the algorithm will add an additional penalty, if after the transformation the x-coordinate is negative meaning the 3D point is situated behind the camera.

[5] Corresponding to value 80% of HSV colorspace.

Optimization CMA-ES is a Covariance Matrix Adaptation using an evolution strategy (ES) to optimize non-linear and non-convex problems. The algorithm creates generations of individuals in our case representing the six extrinsic parameters which are rated by the above described cost function. In each generation, the weakest individuals are replaced using a maximum likelihood method and evolution paths, which prevent premature convergence and improve the convergence to an optimum. Because of the time advantage, methods like the Downhill Simplex [7] are generally preferred for such a small number of parameters. Nevertheless, CMA-ES is used because of its greater robustness against local minima and because of the greater comfort, handling, and the comprehensive data logging of the available implementations. Experiments have shown that a pose estimation on the basis of 100 passpoints with Downhill Simplex takes 0.04 seconds. Although this is twice as fast as CMA-ES the gain in time for the presented usage is negligible.
In the context of the optimization experiments, a single camera was focused to the approximate center of the WCS and several lists with 20 to 100 passpoints were gathered. Afterwards the extrinsic camera parameters were optimized and compared with the estimated position and orientation of the camera. While a distance measuring equipment could be used to approximate the camera position, the orientation had to be estimated roughly regarding the pan- and tilt-angle during the passpoint-collection. **Table 2** displays the estimated, initial, and optimized extrinsic parameters.

	Position(tx,ty,tz)	**Orientation(rx,ry,rz)**
Estimated	3550 1090 2190	180 30 165
Initial	2000 500 3000	180 0 180
Optimized	3413 1093 2163	197 27 159

Table 2: Estimated, initial, and optimized extrinsic parameters.

The fitness function value for the different passpoint lists lies between 6.79 and 7.84 pixel, and accordingly the calculated parameters correspond to the optimized parameters in Table 2. **Figure 5** shows an example for a parameter optimization using a list of 100 passpoints, and **Figure 6** shows the corresponding development of the fitness function.
The relative high fitness function results from the fact that the camera data is requested before the qualisys data during the passpoint collection. Because the marker is continuously moved, the computation time of the detection algorithm leads to a small offset. Furthermore, the optimization algorithm needs initial parameters describing the camera position and orientation roughly to prevent running into local minima. If these rough initial parameters are supplied, the algorithm will calculates the exact camera position reliably.

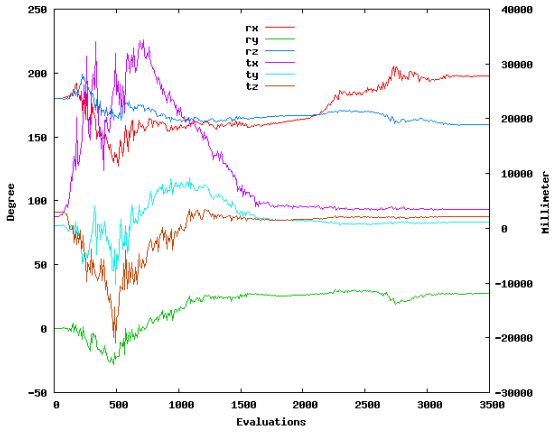


Figure 5: Optimization of the extrinsic parameters

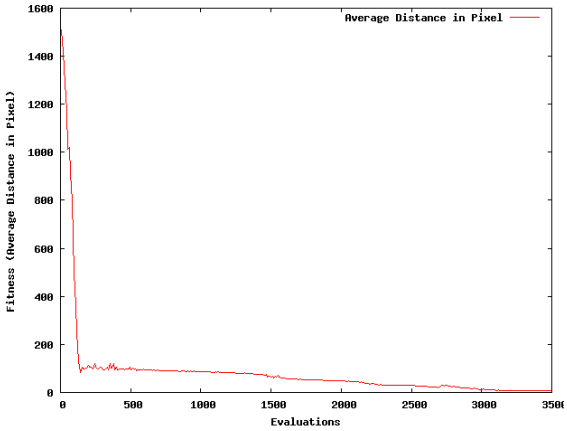


Figure 6: Fitness optimization from a distance of 1509.54 to 6.79 pixel (4446 evaluations in 0.08 seconds)

2.3 Camera Alignment

The internal configuration menu of the AXIS 215 PTZ cameras allows the definition whether the camera is used in a standing or hanging orientation. This affects the image orientation and inverts the rotation direction concerning the pan angles. For this reason, the right-handed coordinate system shown in Figure 7 can be used independently from the configured camera orientation. The used mapping function allows the cameras to focus on each point within the CCS. Furthermore, the tilt angle is restricted to $\pm 90°$. If it laid below $-90°$ or above $90°$ respectively, every new focusing of the camera would result in a full $180°$ pan rotation.
In this work, modified spherical coordinates were used to focus the camera. Spherical coordinates describe 3D-points by defining the length of the vector r, the inclination / tilt angle ϑ ($0°$ to $180°$) and the azimuth / pan angle φ ($0°$ to $360°$). To meet the requirements of the cameras, the tilt angle is restricted to the interval $[-90, 90]$ and the pan angle is restricted to $[-179, 180]$ like shown in (4).

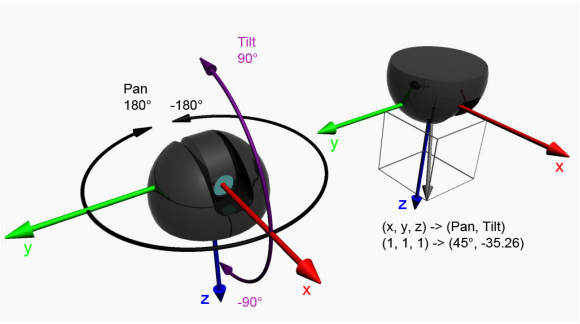


Figure 7: Coordinate system of the AXIS cameras in standing and hanging orientation, direction of rotation, and an example mapping on point (1,1,1)

$$\vartheta = -\arctan\frac{z}{\sqrt{x^2+y^2}} \qquad -90° \leq \vartheta \leq 90°$$

$$\varphi = \begin{cases} \arccos\frac{x}{\sqrt{x^2+y^2}} & y \geq 0 \\ -\arccos\frac{x}{\sqrt{x^2+y^2}} & y < 0 \end{cases} \qquad -180° < \varphi \leq 180°$$

$$\text{with } x^2+y^2 \neq 0 \tag{4}$$

The following code extraction represents the implementation of the formula considering the special cases in which x, y and z are zero and which realizes the camera focusing on a point in the CCS.

```
if(x==0 && y==0) {
    if(z==0)
        return; // do nothing
    rotTilt = (z<0) ? 90 : -90;
} else {
    float factor = sqrt(x*x + y*y);
    rotTilt = RAD2DEG(-atan(z/factor));
    rotPan = RAD2DEG((y<0) ?
        -acos(x/factor) :
         acos(x/factor));
}
```

Together with the calculated camera pose this mapping function allows a fast, calculation cost-effective and accurate alignment on each marker-equipped robotic system within the STB.

3 Automatic Robot Tracing with a Gantry Crane

To achieve automatic position control of the gantry crane, the actual robot position has to be determined (section 3.1) and new desired positions for the gantry crane have to be calculated (section 3.2). The experimental results are presented in section 3.3.

3.1 Robot Position Detection

The gantry crane is equipped with two cameras which allow a vertical view from the topside to the ground. An observation camera is used for data acquisition and monitoring tasks, while a black-and-white tracking camera is used for determining the robot position. Because of the mentioned difficult lighting conditions, the infrared spectrum is used. The utilized tracking camera is sensitive to infrared light which is transmitted by an additional infrared emitter mounted next to. To obtain just the specific image information, a daylight suppression filter is installed inside the objective of the tracking camera. A spherical reflective marker is attached to the robot indicating the coordinates of the robot's center of gravity. It weighs less than one gram and has a diameter of 19 mm. Hence the reflector is not influencing the behavior of the tested robot. One marker is sufficient since the robot's orientation is not a matter of interest. The resulting scene improvement reduces the complexity of object detection to the detection of white circles and makes vision algorithms independent of lighting conditions in the STB.

The center position of the marker has to be kept on a constant reference position in the image in order to automatically trace the robot. This coordinate is determined by the detection algorithm from section 2.1. In case of several marker-like regions in the image, the developed algorithm prefers the region whose center of gravity is closest to the last known coordinates. Occlusion of the marker is not expected.

The developed detection algorithm determines the relative position offset in pixel from the marker to the desired reference position. Since the gantry crane expects absolute world coordinates for control input, a transformation from image to world coordinates is necessary. This transformation needs the intrinsic camera parameters determined by a camera calibration. Thus, the distortion can be eliminated and the pinhole camera model can be used (see **Figure 8**). To transform the x and y coordinates of the center of the detected marker into world coordinates (x_W, y_W), the image distance b and the distance z between marker and objective have to be known. The distance z can be measured by two methods which do not require additional equipment. Either the size of the marker in the camera image or the absolute position of the marker in the STB can be used. The first method does not work because of the small amount of marker pixels in the image. Thus, small pixel changes could result in large distance changes and would cause inaccuracy. The presented solution uses the second method. Therefor a look-up table was created with a prior laser scan of the STB which maps a suitable distance z to each coordinate in the STB (x_{STB}, y_{STB}). The image distance b can be calculated with the knowledge of z and the focal length f, which was determined by the camera calibration (5).

$$\frac{1}{b} = \frac{1}{f} - \frac{1}{z} \tag{5}$$

With the known values for z and b, the transformation in world coordinates can be accomplished by (6).

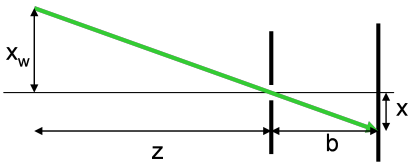


Figure 8: Model of pinhole camera

$$x_W = \frac{z}{b} \cdot x \qquad y_W = \frac{z}{b} \cdot y \tag{6}$$

The distances x and y are built out of the differences between the center of the marker region (x_c, y_c) and the desired reference position in the image (x_{ref}, y_{ref}) multiplied by pixel width d_x and height d_y. The error in world coordinates x_{err} and y_{err} is calculated by (7).

$$\begin{aligned} x_{err} &= \frac{z}{b} \cdot (x_c - x_{ref}) \cdot d_x \\ y_{err} &= \frac{z}{b} \cdot (y_c - y_{ref}) \cdot d_y \end{aligned} \tag{7}$$

The absolute position of the robot's center of gravity in the STB is calculated by adding the actual error (x_{err}, y_{err}) to the actual position of the gantry crane (x_{crane}, y_{crane}) (8).

$$x_{STB} = x_{err} + x_{crane} \qquad y_{STB} = y_{err} + y_{crane} \tag{8}$$

3.2 Desired Position Calculation

Basic Algorithm If the actual position is directly used as the new desired position of the gantry crane, an error would remain, which is proportional to the speed of the robot. This happens because of the time the gantry crane needs to reach the desired position, while the traced object is still moving. For this reason the speed has to be taken into account, which is calculated by (9) using the actual and last known position (x_{STB-1}, y_{STB-1}) as well as the time between measurements Δt. This time is defined by the control frequency of the gantry crane, which is limited to 4 Hz.

$$v_x = \frac{x_{STB} - x_{STB-1}}{\triangle t} \qquad v_y = \frac{y_{STB} - y_{STB-1}}{\triangle t} \tag{9}$$

In a first step a constant speed between two measurements is assumed. So the basic equation (10) calculates the expected point for the next measurement as the new desired position for the gantry crane (x_{set}, y_{set}).

$$x_{set} = x_{STB} + v_x \cdot \triangle t \qquad y_{set} = y_{STB} + v_y \cdot \triangle t \tag{10}$$

This basic algorithm works in principle, but striking errors can occur while tracing. They are caused by numerous uncertainties. On the one hand, (8) shows that the calculation is based on measurements which can be blurred by noise. On the other hand, the gantry crane is not made for real-time applications. Even though the timestamp of an image is very precise, the delay and exact time of measurement of the gantry crane is roughly known. Therefore, the calculated position of the robot in the STB could perform little jumps proportional to the velocity. In order to cope with the uncertainties and to realize a smooth tracing, a particle filter is used similar to the one presented in [8].

Algorithm with Particle-Filtered Position A particle filter consists of a large number of samples or particles. A particle i represents the estimated state of the marker, in particular its position and velocity (x_{STB_i}, y_{STB_i}). For each particle i a weight w_i is calculated, the closer to the actual measurement the higher it gets. This is done by the perceptual model. Afterwards, the resampling step randomly draws new particles based on the weight of the old particles. This means that states around particles with a higher weight are more likely to be drawn. This leads to a filter effect. The old particles get replaced by new particles of the same initial weight. In the last step, each particle is moved accordingly to the motion model which predicts the expected state at the next measurement (x_{bel_i}, y_{bel_i}). The motion model uses the basic equation (10) to move each particle and adds Gaussian noise to the velocity in order to simulate measurement noise (11). After the movement of the particles, the calculation of the new desired position of the gantry crane is possible, which is the average of all believed particle positions (x_{bel_i}, y_{bel_i}).

$$\begin{aligned} x_{bel_i} &= x_{STB_i} + (v_{x_i} + v_n) \cdot \triangle t \\ y_{bel_i} &= y_{STB_i} + (v_{y_i} + v_n) \cdot \triangle t \end{aligned} \tag{11}$$

In order to separate the good from the poor beliefs, a weight has to be calculated for each particle. Two different perceptual models were investigated. First, a Gaussian weighting was chosen. In this approach, taller differences between particle positions and measured position (x_{diff_i}, y_{diff_i}) are considered less. The result is a very smooth movement, but it slows down the filter. So a linear weighting was implemented where the maximum Euclidean distance $diff_{max}$ has to be known. The linear perceptual model (12) has an improved dynamic behavior. The resampling step filters out the poor beliefs and creates new normalized particles (x_{STB_i}, y_{STB_i}). Afterwards, the motion model is applied again.

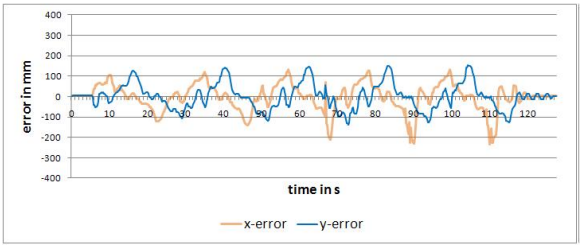
$$w_i = diff_{max} - \sqrt{x_{diff_i}^2 + y_{diff_i}^2} \tag{12}$$

Algorithm with Particle-Filtered Velocity The calculated velocity of the marker is the derivative of the noisy position measurement. For this reason the velocity is even more noisy which has influence on the marker prediction.

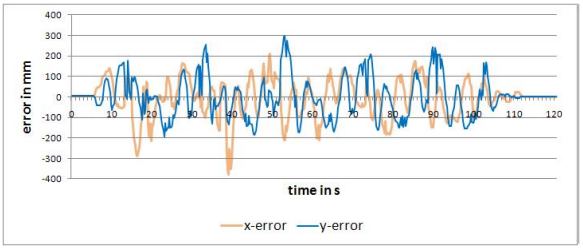
So the developed algorithm with particle-filtered velocity uses the implemented particle filter in the following way to smoothen the velocity. The idea is that particles with higher weight must have had a good motion model. In particular, the velocity which differs between each particle due to the Gaussian noise has to be a good prediction. It follows that the arithmetic, weighted average results in a filtered velocity. This velocity is used in the basic equation (10) to calculate the desired position for the gantry crane.

3.3 Experimental Results

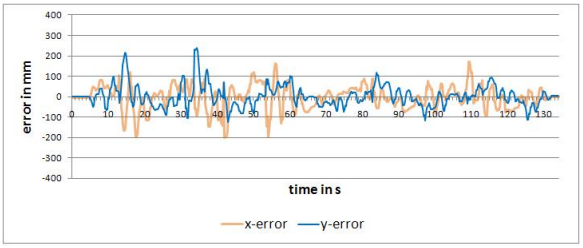
First of all, the basic tracing algorithm was evaluated. For this purpose, a model railway was integrated in the STB. In each experiment, the electric locomotive started, ran five rounds with a speed of ca. 29 cm/s, and then stopped again at the start position. In this way, repeatable results could be obtained, and parameters of the algorithm (exposure, binarize threshold, velocity and acceleration limits of the gantry crane, number of particles, etc.) could be set. Although the minimum speed of the electric locomotive is much faster than the expected velocity of the robots under test, the different algorithms were tested to see advantages and disadvantages. One representative graph per algorithm can be seen in **Figure 9**.



(a) Basic algorithm



(b) Algorithm with filtered position



(c) Algorithm with filtered velocity

Figure 9: Errors while tracing the quickly moving electric locomotive

algorithm	x-error	y-error
basic	68.0 mm	62.5 mm
position filtered	94.0 mm	97.1 mm
velocity filtered	59.4 mm	53.2 mm

Table 3: Standard deviation errors while tracing the electric locomotive

The basic algorithm has problems following the electric locomotive because the maximum control frequency of 4 Hz is too slow for tracing fast objects. In addition, it can be observed that the gantry crane has a reaction time of approx. 0.5 s which causes the huge errors at higher speeds (see **Table 3**). This delay also effects the steady-state error. An oscillation around a fixed point is possible. The developed position filter eliminates this problem. However, the main disadvantage of this method is the worse dynamic behavior because particles with a greater distance are weighted less causing a decreased reaction on faster movements. For this reason errors increase when following the electric locomotive. The velocity filtering algorithm creates the best results. The standard deviation error is decreased while having no residual steady-state error. This is achieved through more reliable velocity information (see **Figure 10**).

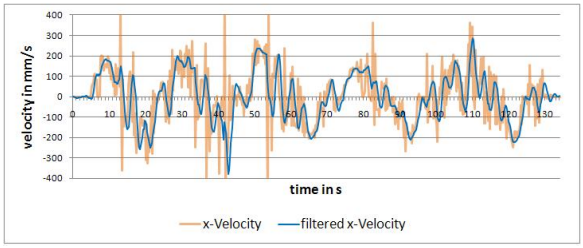


Figure 10: Representative filtered velocity in x-direction

The tested robot moves slower but in a more oscillating way than the electric locomotive which creates different results (see **Table 4**). The basic algorithm performs badly because the control frequency and the delay of commands build up the crane movement around the desired positions. Contrarily, the position filter causes just minor errors (see **Figure 11**). The slow position control damps the oscillating behavior of the robot and is also fast enough to follow the robot's main movement. Only in cases when the robot slides a little faster downhill in y-direction, the position control cannot cope with the speed immediately, so small errors occur. The velocity filtering algorithm is better than the basic one due to its smoother motion. But the filter has little problems avoiding oscillating behavior. Its dynamic advantages do not play a major role here.

algorithm	x-error	y-error
basic	46.3 mm	50.9 mm
position filtered	9.41 mm	13.1 mm
velocity filtered	28.0 mm	30.8 mm

Table 4: Standard deviation errors while tracing the robot

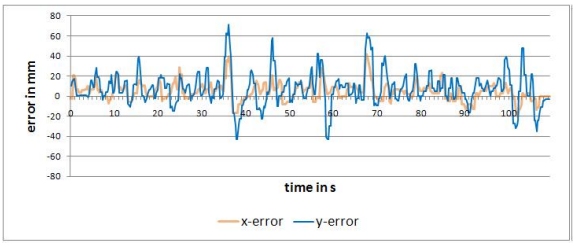


Figure 11: Errors while tracing the slowly moving robot with activated position filter

4 Conclusions and Outlook

The automatic alignment of the AXIS 215 PTZ cameras using the Qualisys tracking system could be realized successfully. This method allows a robust, accurate and fast supervision of the tested robots within the STB. However, a rough camera position and orientation has to be supplied by the user to prevent local minimum during the optimization. Further work will focus on the automatic estimation of appropriate start parameters. Furthermore, the gathering of passpoints will be optimized by reducing the time offset between the data-request of the cameras and the tracking system. The more precise passpoint lists will support the combined optimization of the extrinsic parameters and the focal length. The consideration of the focal length will allow a camera-zoom independent calibration and therewith increases the volume in which a calibration could be performed. Finally, experiments with active markers will be conducted. By using self-luminous markers the infrared emitters can be removed from the AXIS cameras.
The automatic position control of the gantry crane allows a constant top view of the robot during experiments. This makes the observation of each leg movement possible and helps to analyze the robot's locomotion. The mechanical part of the gravity compensation unit is not built yet. This is why the performance of the tracing concerning the gravity compensation could not be tested. Theoretically, a worst-case error of 10 cm would reduce the desired vertical force by less than 1% and thereof create an additional horizontal force of ca. 5%. The gravity compensation concept itself will probably cause major errors.
In general, an automatic robot supervision is realized which supports the operators controlling the whole scenario. In addition, post data analysis is made possible via continuous video material from different points of view including a constant bird's eye view of the robot under test. That feature combined with the recorded trajectory helps to discover malfunctions and to improve walking behaviors.

5 Acknowledgements

The presented work is part of the LUNARES-project. The project is funded by the German Space Agency (DLR, Grant Number: 50RA0706) and the Investment Association Bremen (BIG, Grant Number: INNO1036A). This work is a foundation for ongoing work in the project INVERITAS, funded by the German Space Agency (DLR, Grant Number: 50RA0910) with the financial aid of the Federal Ministry of Economics and Technology (BMWi).
The authors would like to thank all members of the LUNARES team. At DFKI Robotics Innovation Center: Sebastian Bartsch, Timo Birnschein, Florian Cordes, Jens Hilljegerdes, Steffen Planthaber and Thomas Roehr. At EADS Astrium: Ingo Ahrns, Stéphane Estable and Bernd Langpapp. At OHB System: David Koebel and Marc Scheper.

References

[1] T. Huntsberger, G. Rodriguez, and P. S. Schenker. Robotics challenges for robotic and human mars exploration. In *Proceedings of ROBOTICS 2000*, pages 84–90, 2000.

[2] G. Sanders, K. Romig, W. Larson, D. Rapp, K. Sacksteder, D. Linne, P. Curreri, M. Duke, B. Blair, L. Gertsch, et al. Results from the nasa capability roadmap team for in-situ resource utilization (isru). In *International Lunar Conference, Toronto, Canada*, 2005.

[3] F. Cordes, S. Planthaber, I. Ahrns, T. Birnschein, S. Bartsch, and F. Kirchner. Cooperating reconfigurable robots for autonomous planetary sample return missions. In *ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots (ReMAR-2009)*, London, United Kingdom, June 22-24 2009.

[4] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[5] C. Fiorio and J. Gustedt. Two linear time union-find strategies for image processing. *Theoretical Computer Science*, 154(2):165–181, 1996.

[6] E.R. Davies. *Machine vision: theory, algorithms, practicalities*. Academics Press, 1990.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, September 2007.

[8] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.