Vision-based control of the Manus using SIFT

Citation for published version (APA):

Sijs, J., & Liefhebber, F. (2007). Vision-based control of the Manus using SIFT. In *Proceedings of the 10th IEEE International Conference on Rehabilitation Robotics (ICORR '07), June 13-15 2012, Noordwijk, The Netherlands* (pp. 854-861). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ICORR.2007.4428524>

DOI:

[10.1109/ICORR.2007.4428524](https://doi.org/10.1109/ICORR.2007.4428524)

Document status and date:

Published: 01/01/2007

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Vision-based control of the Manus using SIFT

Freek Liefhebber, Joris Sijs

Abstract—The rehabilitation robot Manus is an assistive device for severely motor handicapped users. The executing of all day living tasks with the Manus, can be very complex and a vision-based controller can simplify this. The lack of existing vision-based controlled systems, is the poor reliability of the computer vision in unstructured environments. In this paper, a computer vision solution is presented, which can estimate real-time the pose of an object and co-operate with a vision-based controller. The computer vision is robust to illumination changes, a varying scale and rotation and is robust to occlusion. The computer vision is mainly based on the SIFT-algorithm and the usage of a 3D-model of an object. Important steps to create this 3D-model are discussed. The detection and recognition of the required SIFT-keypoints, has become real-time, by exchanging redundancy against calculation time. With a position-based visual servoing controller, the Manus can be positioned with respect to an object.

I. INTRODUCTION

The rehabilitation robot Manus [15] is an assistive device for severely motor handicapped users. With it, users can perform daily tasks that would otherwise be impossible or would require the help of other persons. Manus has eight degrees of freedom (DOFs); three for positioning the gripper, three for rotating the gripper about three axes, one for opening the gripper and one external lift.

Because of its 8 DOFs, Manus is a complex device, which is sometimes cumbersome to control for persons with very limited residual functionality. To perform an ADL-task (All Day Living), such as pouring a glass of water, the user has to perform a high number of actions and complex motions. This will result in a high physical and cognitive load [17] on the robot users. The result of these problems is that users get fatigued and stop using the robot.

In the past, there has been done much research [19],[20] in how to improve the functionality of rehabilitation robots. The improvement of the functionality is reached by: 1. advanced user-interfacing and 2. innovative robot capabilities. With advanced user-interfacing functionality, such as an eye-mouse and EMG-interfacing, a disabled user is better able to use the rehabilitation robot. While innovative robot capabilities, such as visual-servoing and redundant robot joint control, simplifies the execution of the

ADL-tasks. In this paper, we will only focus on vision-based control to simply the control of a rehabilitation robot, because this technology is very versatile and can be used for many tasks.

In [16],[17],[18], we have presented our previous work on assistive technology to support users of the Manus. The aim wasn't to create a fully autonomous robot, where the user only gives high-level commands, because it is impossible to fully automatize the large variety of ADL-tasks. Therefore we have applied the collaborative control paradigm, where sensor-based control can be exploited to assist an user during the execution of a task. The sensor-based controller is only responsible for a small part of the entire tasks and it co-operates with the user to fulfil the entire task. Central in our system was an image-based visual servoing (IBVS) controller, which can control the orientation of the Manus with respect to an object, while simultaneously the position is controlled by another sensor-based controller or user. This IBVS-controller receives information from the computer vision, which can track objects after being pinpointed by the user. This system has been evaluated by actual Manus-users and on basis of this evaluation have we chosen for a new concept of the vision-based control of the Manus.

In this paper, we will first discuss the results of our evaluation. The recommendations from this evaluation, has resulted in a re-design of the computer vision, which described in section III. In this section various important components of the computer vision will be presented, such as the SIFT-algorithm, pose-estimation and the creation of a 3D-model of an object. In section IV, we will describe our vision-based controller.

II. EVALUATION

The collaborative control paradigm [16] releases some of the requirements on additive sensors and controllers, to assist the user. This paradigm demands a higher degree of the user's involvement during the executing of the task, in order to enable sensor-based control, which eventually simplifies the executing of tasks. In our system, a vision-controller is the main sensor-based controller. On basis of an eye-in-hand camera and computer vision, the vision-based controller can position the Manus with respect to an object. The basic principle of the design of the computer vision, was that it didn't require any prior knowledge about the object. By simply pinpointing the object, the computer vision was able to track the object and measure elementary image properties (features), such as centre-of-mass and surface. On basis of these elementary features, the vision-based controller can

Manuscript received February 11, 2007.

Freek Liefhebber is with TNO Science & Industry, PO Box 155, 2600 AD, Delft, The Netherlands (phone: +31 15 2692267; fax: +31 15 2692111; email: freek.liefhebber@tno.nl)

Joris Sijs is with TNO Science & Industry, PO Box 155, 2600 AD, Delft, The Netherlands (phone: +31 15 2692217; fax: +31 15 2692111; email: joris.sijs@tno.nl)

position the Manus.

The functionality of this vision-based-controller and the interaction with the user has been evaluated. This evaluation is done with experienced Manus-users, in order to get unbiased results. With disabled users, the evaluation of one new component will depend heavily on the overall required physical and mental load to use this component. In other words, the evaluation of the vision-based controller will also depend on the rest of the system, such as the user-interface, input-device and the possibility to configure the system to the residual capabilities of the user. That is why, we have put much effort in other components, next to the vision-based-controller.

New options and control-modes of the system have to be incorporated a new menu-structure, which will be larger than the existing menu-structure. It was a challenge to keep the extended menu-structure intuitive and to keep the required number of user-actions to go from a submenu to any other submenu, low. The main-solution is a menu-structure, where you can go directly from a submenu to four other submenu's [17]. The user can switch to another submenu by giving a fast deflection in one of four directions of a joystick. Some users aren't able to give a fast deflection in all four directions and therefore these users use a extra button to switch from submenu. When an user presses the button, a small clock will start and depending on the moment the user releases the button, we will switch to a certain submenu.

The menu-structure is visualized (fig. 1) in a graphical user-interface (GUI) on a 8 inch lcd-monitor, which is attached to the wheelchair. Next to the menu-structure, the camera view has to be displayed in the GUI. We use a camera with a large field-of-view and therefore we should display the camera view as large as possible. To display the camera images, we use the whole display and we visualize the menu-structure by a translucent overlay (fig. 1).

The evaluation tasks were various ADL-tasks, such pouring a glass and grasping objects. To get realistic results, we have done 50% of the evaluations in the home environment of the user. On basis of these results, we can

draw the following conclusions and recommendations with respect to the implemented vision-based controller. The assistance of a vision-based controller can simply the execution of ADL-tasks and reduce the physical load, which is experienced by users. Especially the execution of ADL-tasks, where a lot rotation is required or the user has poor visibility, is simplified.

The vision-based controller was integrated in the system, according to the collaborative control paradigm and therefore it requires a large involvement of the user. The vision-based-controller did only control the orientation of the robot and the user has only to concern about the position of the robot. The user can collaborate with vision-based-controller to perform a task, but this collaboration was confusing to some users. More autonomy of the vision-based-controller by controlling more DOFs can reduce the required large involvement of the user and it will reduce the mental and physical load.

The user can pinpoint an object in the camera view, which the computer vision will track. The computer vision doesn't have any prior knowledge about the object, except one pixel of the object at the start of the track. Afterwards the object is segmented in the HSV-space by adaptive thresholds, in order to be robust to changing illumination. This approach is sufficient to objects, which are uniform of color and different with respect to the background, but it isn't very robust with all kinds of objects in complex environments. Another limitation of this approach, is that the computer vision can't recover after failure, because it has too little prior knowledge to recognize the object and continue with tracking.

To select an object, the user has to pinpoint a pixel of the object, by moving a cursor with the joystick. This pinpointing can require many actions of the user and it can be very time consuming, especially with small objects. Because of these actions and time, the overall benefit of the vision-based controller is reduced. The object selection should be simplified and this can be achieved by object recognition, which enables easier object selection methods.

III. COMPUTER VISION

On basis of the evaluation, we have concluded that the existing computer vision wasn't reliable enough to operate in an unstructured environment. The poor reliability is mainly caused by the lack of prior knowledge and the usage of the assumption, that an object is uniform of color. Therefore we have looked for a new approach of the computer vision, which is able to co-operate with a vision-based controller. We want to use a classical visual-servoing controller, which consist of a closed-loop controller and the eye-in-hand camera configuration. This control strategy will put some requirements on the computer vision. The computer vision should be:

1. Scale- and rotation-invariant. The camera will be positioned (6D) with respect to an object and

Fig. 1. Graphical user-interface of the collaborative control system. The image of the camera is shown as background and the menu-structure is visualised by a translucent overlay.

therefore it should be robust to a varying scale and rotation.

2. Illumination-invariant. The illumination can continuously change and especially during complex motions of the camera. The camera can be directly illuminated by a lamp or the sun and the camera will be partly over-illuminated. The auto-shutter of the camera will directly change the shutter time and the appearance (intensity) of the object will change in the image.
3. able to recognize an object. This required to enable more advanced object selection methods and the computer vision can recover after a short failure.
4. Viewpoint-invariant. Due to the perspective mapping of a camera, the size and appearance of the object in the image will change during motion.
5. Occlusion-invariant. The object can be partly occluded or during motion of the camera, the object can be occluded.
6. able to cope with 3D-objects. In general image-processing algorithms use the assumption, that objects are planar. In our case, most objects aren't planar and we should be able to cope with 3D-objects.
7. able to "measure" enough information to position the camera with respect to an object. The computer vision should be able to detect and track features, which contain information about the position of the object and this information can be used by an image-based or position-based visual-servoing controller.
8. fast enough for closed-loop control. The absolute accuracy of the Manus is low, due to backlash, but the relative accuracy is high. We want to position the Manus with a high accuracy and therefore we want to use feedback control to exploit the high relative accuracy. A disadvantage of feedback control, is the requirement of a small sample time, to achieve fast positioning and avoid instability.

In the following subsections, our solution and approach to these problems are described. We start with the basis of our computer vision: SIFT. Afterwards shall we shortly describe how we estimate the pose of an object. A key component of our system is the 3D-model of an object and the important steps to create such models is presented. Finally some concepts to speed up the computer vision are discussed.

A. Scale-Invariant-Feature-Transform

Our computer vision should be robust to illumination-changes, varying scale and etc. An appropriate approach to fulfill this is with the usage of keypoints (very characteristic points in an image). Recently various new approaches of keypoint-detection [2],[3],[4] are developed. The most popular algorithm is SIFT [2]: Scale Invariant Feature Transform. The SIFT-algorithm is able to detect keypoints in

Fig. 2. Some properties of the SIFT-algorithm. The blue lines represent the matches between the two images and only a small percentage (2 or 5%) is shown. Top-left: scale-invariant; Top-right: rotation-invariant; Bottom-left: Reasonably viewpoint-invariant; Bottom-right: Illumination-invariant

an image and create from each keypoint a descriptor. On basis of this descriptor, a keypoint can be recognized in an other image or in a database, which consist of SIFT-descriptors. The SIFT-algorithm is probably, the most robust computer vision algorithms to detect and recognize keypoints and it is: 1. Illumination-invariant, 2. Scale-invariant, 3. Rotation-invariant, and 4. reasonably viewpoint-invariant. Because of these properties (fig. 2), the SIFT-algorithm is suitable to be used as the basis of our computer vision module, which supports a vision-based controller. A disadvantage of the usage of keypoints, is that objects should have a textured surface.

The SIFT-algorithm is very robust, but at the cost of the computational complexity. The SIFT-algorithm is very computational expensive algorithm, what complicates the integration with a closed-loop controller. In a closed-loop vision-based system, we want to minimize the delay (calculation-time) of the computer vision, because of stability issues and minimization of the response- and settling-time. In general, a low camera-resolution is used, in order to get a small delay of the computer-vision. This solution has two disadvantages:

1. A low camera resolution will result in a low accuracy of the localization (image-coordinates) of the keypoints. This low accuracy will eventually result in a low accuracy of the positioning of the Manus.
2. The capability of the SIFT-algorithm to recognize objects in low resolution image is small. We want to be able to recognize, a large variety of objects, at various distances and therefore the camera resolution should be as high as possible.

Because of these two reasons, we want to use the maximum resolution of the camera: 640x480 pixels. To

achieve “real-time” detection and recognition of the SIFT-keypoints, we have used the implementation of “Evolution Robotics” [5], which is very fast compared to other known implementations. Unfortunately this implementation is still not fast enough and therefore we applied some tricks, which will be discussed in section II D.

B. Pose-estimation

The computer vision should provide enough information to the vision-based controller to position the Manus. An IBVS-controller [6][7] only requires the location of the keypoints and an estimate of the depth, to position a camera with respect to an object. Because of various reasons (section IV), we want to use a PBVS-controller instead of an IBVS-controller. A disadvantage of PBVS, is that it requires an estimate of the pose (3D position and 3D orientation) of the object.

Various pose-estimation algorithms [10],[11][12] are able to estimate the pose of objects, on basis of the location of keypoints in an image. We have chosen for the algorithm of Auroja, because of its simplicity, good convergence-rate and this approach is generic. This algorithm can iteratively optimize the 6D-pose on basis of image-points and the 3D-position of those points with respect to the object frame. The optimization is an iterative Newton-Raphson optimization, which converges from an initial guess (or the previously estimated pose) to the actual pose. When the difference between the initial guessed pose and the actual pose is not too large, the algorithm will always converge. The accuracy of this algorithm will increase, in case more keypoints are used, but unfortunately the convergence-rate will reduce.

Fig. 3. Estimation of the pose (and Kalman-filtered pose), when an object is coming closer to the camera. The computer vision is able to estimate the pose at distance of 1 meter and the accuracy increases, when the distance becomes smaller.

Also the possibility of the usage of outliers will increase with an increasing number of points. To become robust to this phenomenon, we have incorporated the pose-estimation algorithm in a RANSAC (Random Sample Consensus) algorithm [21]. The RANSAC algorithm executes the pose-estimation algorithm k times and each time with a different set of n keypoints with the corresponding 3D-positions. Each time, the RANSAC algorithm validates the correctness of the estimated pose for all points. After the k executions, it uses the pose with the best score on the validation.

C. 3D Model Generation

The pose-estimation algorithm requires next too the image-coordinates of the keypoints also the corresponding 3D-position (with respect to the object-frame). In other words, we need a point-based 3D-model of an object from which each point can be recognized via the SIFT-algorithm. The generation of such 3D-model contains the following three steps:

1. Capturing images from various.
2. Calculation of the 6D pose of the camera with respect to the object frame.
3. Calculate 3D-position of a keypoint via triangulation.

1) Capturing images

We want to create a model of an object, as complete as possible. This purely on basis of images, which are made from various viewpoints. To capture these image, we can do the following:

1. A camera moves around the stationary object.
2. The object is placed on a turntable, which rotates in front of a stationary camera.

The first method seems to be very attractive, because the camera is already attached to the Manus and we don't require any extra tools. But unfortunately the workspace of the Manus is too small, to rotate around the object with a radius of 30 cm. This distance of 30 cm is assumed to be the average working distance to the object. Therefore we use a turntable in combination with the stationary camera. In order to calculate the 3D-position of a keypoint, the same keypoint should be recognized in multiple images and therefore the images should overlap each other. In our case, we take an image after a rotation of approximately 20° .

2) Calculate camera position

To calculate the 3D-position of a point in an image, the 6D-pose of the camera with respect to a certain “fixed” frame should be known. Desirably this fixed frame is equal to the object frame. There exist two approaches to calculate this 6D-pose purely on basis of images:

1. Estimate the camera-displacement via the decomposition and estimation of the fundamental matrix [1]. The relative camera-displacement between two images can be directly estimated, but to get the absolute pose the

Fig. 4. Four example images (of in total 16 images) from an object. The object is placed on a turntable, from the table is covered with a picture. This picture is our reference object, which use to estimate the position of the camera. Each time the turntable is slightly rotated.

previous relative camera-displacements should be cumulatively added to each other. This will result in an inaccurate pose.

2. We estimate the pose (camera to object) of a known reference object, which is visible in the image and it is fixed to the object frame. By estimating the pose of the reference object, we indirectly estimate the pose of our object.

The second method is in favor if the visibility of the reference object doesn't introduce practical problems with our setup. There don't exist any constraints about the reference object, except the visibility and the fixation to the object frame. Therefore we have chosen for the ideal reference object: the table of the turntable. We have covered the table with a picture, which the SIFT-algorithm can recognize. Because the picture is a 2D-surface and we have the image-coordinates of the keypoints (via the SIFT-algorithm), we directly know the 3D-position of the keypoints. These 3D-positions and image-coordinates are used to estimate the pose (see previous section) of the reference object with respect to the camera. In case the

Fig. 5. Estimated poses of the camera with respect to the objectframe. These poses are estimated on basis of the reference object. The blue arrows are the optical axis of the camera or the Z-axis.

object is placed in the center of the table, the origins of the object frame and reference object frame are equal to each other and then we directly know the required camera pose.

3) Triangulation

In this step, the 3D-position of the SIFT-points will be estimated and it forms the basis of the 3D-model. Estimating the 3D-position of a image-point, which is seen / recognized in multiple images, is a well-known problem. Various approach exist, but in [1] a linear approach is presented, which can estimate the 3D-position on basis of multiple ($n > 1$) images. In our case, we check for all keypoints (in all images), in which other camera's this keypoint is also seen and then we apply triangulation.

This results in an accurate estimation of the 3D-position, of the keypoints. Because next too the object (of interest) also other objects may be visible. These other objects can easily be discarded by eliminating 3D-positions below the turntable and outside the expected dimension of an object, which the Manus can manipulate. Unfortunately still some outliers will be part of the 3D-model. The pose-estimation algorithm is robust to these outliers and it isn't a problem. But because of cosmetic reasons, we have removed almost all outliers via clustering. We have applied hierarchical single linkage clustering, to cluster the points, which belong to the object and thereby removing the outliers. With this clustering technique, a point is added to a cluster, if the

Fig. 6. Two views from a constructed 3D-model of an object (see fig. 4). The texture of the object is recognizable in the 3D-model.

Euclidian distance to one of all points in that cluster is smaller than a certain threshold. This clustering technique works very good, because the object is very dense and the regions, where the outliers are, are very sparse. A result of the 3D-model generation is shown in fig. 6.

D. Accelerate computer vision

The computer vision can recognize an object via the SIFT-algorithm and with the recognized keypoints and a known 3D-model, the pose is estimated. The calculation time (< 9 ms) of the pose is almost negligible compared to the calculation time of the SIFT-algorithm. And therefore we mainly concentrate on the acceleration of the detection- and recognition of the required keypoints, via the SIFT-algorithm. The SIFT-algorithm is a computational expensive algorithm and it is too slow (with 640x480 resolution images) to be directly used in a closed-loop system. The calculation time mainly depends on the image size (number of pixels) and the number of detected keypoints. We use the fast implementation of "Evolution Robotics", where we unfortunately don't have control over some essential parameters of the SIFT-algorithm. Therefore we have to apply two concepts to speed up our computer vision, to enable closed-loop control.

The first concept is a well-known approach to speed up a part of the computer vision: region-of-interest (ROI). It isn't necessary to process the whole image, if we are only interested in small part of the image and we exactly know which part this is. In general, only the part of the image is used, where the object was the previous time (plus an extra margin). But in case of (temporary) occlusions, the possibility exist that each time a smaller part is used and eventually the object will be lost. A more robust approach is to project the points of the 3D-model, by using the estimated pose, on the image-plane. This projection will result in a region in the image, where the object can be expected. This region is used as region-of-interest by our SIFT-algorithm.

The first concept works very good, if the object is far away and it results in a small region-of-interest. But in case the object is very close by, the region-of-interest will cover the whole image and still the detection- and recognition of the required keypoints will be very slow. In this case, the SIFT-algorithm will find hundreds of keypoints, while even less than ten keypoints are required to perform reliable object recognition and accurately estimate the pose. The information in the recognized keypoints is redundant and we want to exchange this redundancy with a reduce in the calculation time. This goal can be achieved by resizing (shrinking) the image. By shrinking the whole image, the SIFT-algorithm will find less keypoints and the required calculation time will be much smaller. The reduce in the calculation can be attributed to the reduce in the number of pixels and thereby a reduce in the number of keypoints. We have exploited this concept by the implementation of a

Fig. 7. Estimated poses of the camera with respect to the objectframe. These poses are estimated on basis of the reference object. The blue arrows are the optical axis of the camera or the Z-axis.

"sample rate controller", which controls the number of recognized keypoints by changing the resize factor. This sample rate controller reduces slowly the size of the image, if too much ($n > 30$) keypoints are recognized and increases the size if too less keypoints are recognized. This controller will continuously change the image size, in order to get enough information from the SIFT-algorithm, but as fast as possible.

On basis of these two concepts, the computer vision (image grabbing, SIFT-point detection and recognition and pose-estimation) operates at a sample rate of 15 Hz on a Pentium VI 3 GHz single core PC.

IV. VISION-BASED CONTROL

In this section, we will describe the vision-based controller, which positions the Manus on basis of the computer vision. Figure 8 shows a diagram of the entire system in a closed-loop. The miniature camera is attached to the gripper. We have chosen for this classical visual-servoing approach (closed-loop controller and eye-in-hand camera configuration), because:

1. it avoids the low absolute accuracy and optimally exploits the high relative accuracy of the Manus.
2. with this camera configuration, the manipulator will not cause occlusion of the object in the image. Also the vision-based controller can operate outside the visibility range of the user (on a high closet shelf or on the ground), so the workspace of the camera is much larger compare to a fixed camera at the base.

The important components of the computer vision are described in the previous section and it is used to estimate the real pose $\vec{p}_{real}$. The robot controller controls the Manus and it is responsible for the inverse kinematics calculation. The robot controller, is able control the velocity of the tool frame $\vec{v}_{tool}$ and it receives the setpoints from the vision-based controller. The input to the vision-based controller is the difference between the desired pose $\vec{p}_{des}$ and the real pose $\vec{p}_{real}$. The vision-based controller will control the actual pose $\vec{p}_{real}$ towards the desired pose $\vec{p}_{des}$ in order to position the Manus with respect to an object. The purpose of the vision-based controller, is that, at the desired pose, the

Fig. 8. Schematic diagram of the entire system.

user can easily grasp or manipulate the object.

We have chosen for a Position-Based Visual Servoing [6] (PBVS) controller instead of an Image-Based Visual Servoing [7][8] (IBVS) - or Hybrid Visual Servoing [9] (HVS) controller. The main-disadvantage of IBVS and HVS are:

1. these control-strategies cause an unnecessary long trajectory of the robot. These control-strategies choose this long trajectory, because it make sure that the object is always in the center of the image and therefore always visible. This is of course a nice property, but with this application, where (disabled) people are nearby, only logical and bounded trajectories are aloud.
2. these control-strategies are less flexible with a varying desired pose. The desired pose should always be learnt by the classical “teach by showing” approach, which captures an image from the object at the desired pose and this image is required, by IBVS or HVS, to position the camera to this desired pose.

The input of the PBVS-controller is the difference between the desired pose and the estimated pose and it transforms this error to the equivalent-axis representation [14]. In this representation, the orientation is represented by a rotation θ around a vector $[k_x \ k_y \ k_z]^T$. The error which the PBVS-controller wants to minimize is: $e = [t_x \ t_y \ t_z \ \theta k_x \ \theta k_y \ \theta k_z]^T$, where t is the translation. We use a PBVS control-law with a proportional gain P and a differential gain D : $v_{cam} = (P + Ds)e$, where v_{cam} is the camera-velocity and s is the Laplace-operator. On basis of this error-vector and this simple control-law, the camera will move in a “straight line” from the start-position towards the desired position. In general, PBVS-controller only have a proportional gain, but we use also a differential gain to reduce the settling-time of a positioning-task and improve the stability of the closed-loop system.

The origin of the camera isn’t equal to the origin of the

Fig. 9. The estimated pose during the positioning by the PBVS-controller. A small overshoot and some coupling between the DOFs is noticeable. Next too the estimated pose, the Kalman-filtered pose is shown.

tool frame and therefore the camera-velocity v_{cam} will not be equal to the tool-velocity v_{tool} . The tool-velocity is equal to: $v_{tool} = W v_{cam}$, where W is equal to $W = \begin{bmatrix} R & [t]_{\times} \\ 0_{3 \times 3} & R \end{bmatrix}$.

This vision-based controller is, in combination with the computer vision, able to accurately position the Manus with respect to an object. The closed-loop controller results in a fast and stable response (fig. 9).

V. CONCLUSIONS

This paper has presented the results of the evaluation, by Manus users, of the “old” vision-based system on the Manus. It concluded that assistance of vision-based controller on the Manus, can simplify the executing of ADL-tasks. To improve the functionality, the computer vision should be more robust, enable object recognition and the vision-based controller should be able to control more degrees-of-freedom.

These recommendations has resulted in completely new computer vision, which is mainly based on the robust SIFT-algorithm. The SIFT-algorithm detects and recognizes keypoints, which is used to recognize objects and estimate the pose of the object. The pose-estimation algorithm uses a 3D-model of an object, which can easily be created. The computer vision is very reliable and robust to illumination-changes, scale- and rotation-variation and occlusion.

To speed up the computer vision, a region-of-interest is used and the concept of exchanging data redundancy against a smaller calculation time is exploited.

A PBVS-controller is used to position the Manus with respect to the object.

REFERENCES

- [1] Multiple View Geometry in Computer Vision (Second Edition), R. Hartley en A. Zisserman, Cambridge University Press, Cambridge, 2003
- [2] Distinctive Image Features from Scale-Invariant Keypoints, D.G. Lowe, <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [3] SURF: Speeded Up Robust Features, H. Bay, T. Tuytelaars and L. van Gool, ECCV2007, <http://eccv2006.tugraz.at/files/demos/Bay.pdf>
- [4] A Performance Evaluation of Local Descriptors, K. Mikolajczyk and C. Schmid, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 27, No 10, October 2005
- [5] Evolution Robotics, <http://www.evolution.com/>
- [6] Survey of vision-based robot control, Ezio Malis, 05-01-2007, www.sop.inria.fr/icare/personnel/malis/papers/malis_CCIV-02.pdf
- [7] A new approach to Visual Servoing in Robotics, B. Espiau, F. Chaumette en P Rives, IEEE TRANSACTIONS on Robotics and Automation, Vol. 8, No. 3, June 1992
- [8] Chaumette, F., A first step towards visual servoing using image moments, Proceedings of the 2002 IEEE/ RSJ Int. Conf. on Intelligent Robots and Systems, October 2002
- [9] Chaumette, F. and Malis, E., 2½ D visual servoing: a possible solution to improve image-based and position-based visual servoings, Proceedings of the 2000 IEEE Int. Conf. on Robotics and Automation, April 2000
- [10] Model-Based Object Pose in 25 Lines of Code, D.F. Dementhon en L.S. Davis, 05-01-2007, http://www.cfar.umd.edu/~daniel/daniel_papersfordownload/Pose25Lines.ps.gz
- [11] Numerical Methods for Model-Based Pose Recovery, R.L. Carceroni en C.M. Brown, 05-01-2007, historical.ncstrl.org/litesite-data/rochester_cs/97.tr659.Numerical_methods_for_model-based_pose_recovery.ps.gz
- [12] A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm, 05-01-2007, <http://www.cs.rochester.edu/u/www/u/carceron/pubs/matlab/index.html>
- [13] Intelligent Systems – Fusion, Tracking and Control, G.W. Ng, Research Studies Press Ltd., Hertfordshire (England), 2003
- [14] Introduction to Robotics, J.J. Craig, 2^{de} druk, 1989, Addison-Wesley Publishing Company inc.
- [15] Verburg G, Kwee H, Wisanaka A, Cheetham A, van Woerden J. MANUS: The evolution of an assistive technology. Technology and Disability, 5(2):217-228, 1996.
- [16] Collaborative Control of the Manus Manipulator, B. Driessen, F. Liefhebber, T. ten Kate and K. van Woerden, ICORR 2005, Chicago
- [17] A Framework of Interface Improvements for Designing New User Interfaces for the Manus Robot Arm, H. Tijsma, F. Liefhebber and J. Herder, ICORR 2005, Chicago
- [18] Collaborative Visual-servoing of the Manus Manipulator, F. Liefhebber, B. Driessen and A.H.C. Thean, CWUAAT 2006, Cambridge
- [19] Integration of a Rehabilitation Robotic System (KARES II) with Human-Friendly Man-Machine Interaction Units, Z. Bien and et. al., Autonomous Robots 16, 165-191, 2004, Kluwer Academic Publishers
- [20] Rehabilitation Robot in Intelligent Home Environment – Software Architecture and Implementation of a Distributed System, O. Prenzel, J. Feuse and A. Graser, ICORR 2005, Chicago
- [21] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.