Discrepancy Detection of Robot Arm Motion and its Motion Estimation for Safety in Rehabilitation Robotics

U. Lange and A. Gräser

Institute of Automation, University of Bremen
Emails: lange@, ag@iat.uni-bremen.de

Keywords: *Robotics, Safety, Time-of-Flight-Camera, Image Processing, Simulation*

Abstract

In rehabilitation robotics impaired humans are in the workspace of robot arms. Due to the lack of any ability to avoid the robot arm, if it moves unexpectedly, there is a specific higher need for safety. Safe motion planning for the robot depends on models of the robot arm and the environment. If the models are incorrect, the planning and subsequently the movement are also incorrect. The approach to detect incorrect models or erroneous motion is the comparison of the view of a *„time-of-flight"* camera with the depth image of the 3D robot model. The difference of these two images privides a direct benchmark for correctness – currently with a sampling rate of about 10 Hz.

Before a comparison can be done, the measurement errors of the camera have to be known. Preliminary error measurements of the *„time-of-flight"* camera show systematic errors depending on the distance and the material of the object in focus. These errors were analyzed and estimated by a 5^{th} order equation. First result images of the discrepancy detection are presented. Having detected the robot arm correctly, the rest of the image can then be used for the detection of known and unknown objects in the environment model.

One limitation of this approach is the use of a single camera, which can have occluding objects in front of the robot arm. Another limitation is the use of projection matrices given by the permanent camera calibration, which can be erroneous and result in higher discrepancy.

When the robot arm moves, it can get out of the *„time-of-flight"* camera image. Using a *„pan-tilt-head"*, tracking of the robot will be possible in the next development step to have the arm always in the field of view. This can also be used to look at a predicted position given from the motion planning.

1. Introduction

Whenever humans and robots share a common space, humans might get hurt. In service robotics and in its specialization rehabilitation robotics, robots have to share space with humans in order to serve or to help rehabilitate the human. In the case of the rehabilitation robot FRIEND (Functional Robot arm with user-frIENdly interface for Disabled people) [1, 2], developed at the Institute of Automation (IAT), a quadriplegic human is constantly in the workspace of a robot and cannot escape a malfunctioning robot. Detecting joint errors is necessary because the path planning and collision avoidance depends on a correct model.

The FRIEND system consists of a wheelchair with a robot mounted is designed to restore the user's capability to grasp objects for activities of daily living or in the working life (see figure 1.1). Providing 90 minutes of autonomy to the user from service personal is the goal of the FRIEND project.

Figure 1.1: The FRIEND system.

To assure safety for the user in FRIEND's wheelchair, the FRIEND::Safety framework is being developed. This framework consists of three components:

- Observe,
- Classify Risk and
- Intervene.

Sensors observe the system and send safety relevant facts to a database; the risk analyzer component classifies the current risk to the user by taking the previously gathered safety relevant facts into consideration. Finally, the intervention can be initialized by software-stop and/or by a power interruption actuator, which is set out to return the system back to a safe state. The proposed idea in this article is part of the 'Observe' component of the FRIEND::Safety framework.

The following chapters describe related work (2) and preliminary investigations (3) for the proposed fast discrepancy detection algorithm (4). Chapter 5 discusses first experimental results before the article closes with a conclusion and future work (6).

2. Related Work

In the dissertations of Prenzel [1] and Ojdanic [3], the FRIEND system is described in detail. Prenzel created a process model for semi-autonomous service robots, like FRIEND. He proposed offline verification of the task planning for safe and logically correct execution of tasks, which the user wants to execute. These logical checks will be used by the FRIEND::Safety framework for intervention rules, as correct behavior was shown reliably.

Ojdanic wrote about path planning and collision avoidance of robot arms, especially for service robots, like FRIEND. His path planning does not search in joint space (seven dimensions in the case of FRIEND) but in Euclidean space (three dimensions), which results in faster computation and also more predictable, human-like movement. Higher acceptance by human users is expected, when the robot moves invariably. The 3D-model, used for

FRIEND's path planning is displayed in figure 2.1. This model is of great importance for the proposed algorithm in chapter 4.

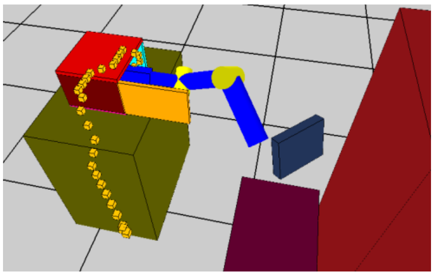

Figure 2.1: FRIEND's 3D-model for motion planning of a robot. This example shows a planned path (dotted) out of a microwave oven to a desired free position.

A Time-of-Flight-Camera (ToF) can capture a scene into a depth image, which can also be transformed into an equivalent 3D-surface. The ToF-camera Swissranger 3000 by MESA Imaging is available in the FRIEND system.

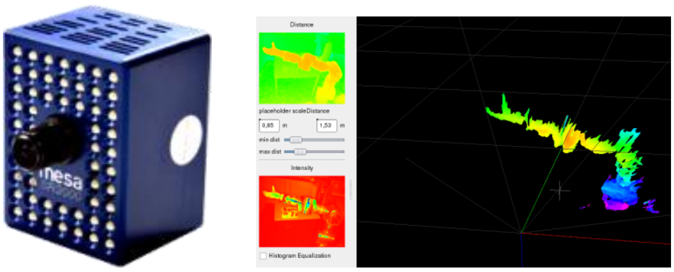

Figure 2.2: Left - Swissranger 3000 ToF-camera by MESA Imaging [4]. Right – Robot, captured by Swissranger 3000, visualized in 2D and 3D.

With SafetyEye [5] by Pilz, a safety solution exists on the market for industrial robots. This product consists of a stereo camera sensor, which can detect humans in the workspace of a robot and reduces its speed down to zero, depending on the distance between robot and human. Unfortunately, SafetyEye is not appropriate for the FRIEND system, as the safe minimum distance between robot and human should always be one meter. In the FRIEND system this is permanently the case.

Project SIMERO [6] by University of Bayreuth concentrates on the fusion of multiple ToF-cameras. The fused data results in real three-dimensional information which is also used to detect humans in the workspace of robots. The project focuses on human robot cooperation, which results in wanted soft collisions. The disadvantage of using multiple ToF-cameras is the higher need of computation power. Therefore, this project also has to handle with distributed computing, which is followed by increased complexity of the overall system.

3. Preliminary Investigations

Before it makes sense to use the fast discrepancy algorithm (see next chapter), two preliminary investigations have to be done, to assure a correct malfunction detection behavior. On the one hand, this is an investigation of the accuracy of the ToF-camera in comparison to manually measured distance between the camera and an object. On the other hand, the generation of depth images from a 3D model of the world, which should look exactly like the real world, captured by a ToF-camera. Additionally, the virtual view of the generated depth image has to have the same image size and field of view as the real ToF-camera.

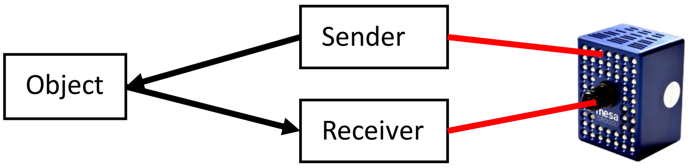

Figure 3.1: The time-of-flight measuring principle. The infrared diodes emit modulated light and the CCD sensor receives this light, knowing the modulation frequency. The phase shift between emitted and received modulated light is in linear relation to the distance between camera and object.

ToF-cameras are a relatively new sensor type. In comparison to laser scanners, they are faster but also more inaccurate and both sensors are active measurement methods and therefore they are robust with respect to environmental light changes. Comparing ToF-cameras with Stereo-cameras at first the lower resolution of ToF-cameras stands out negatively. ToF-cameras return evenly distributed depth values for unicolored objects, whereas algorithms for stereo-cameras can only roughly estimate the depth of the middle of the object. In addition, ToF-cameras return their depth images at a constantly high frame rate (25Hz at a reasonable exposure time for indoor environments) and need no transformation calculation, while stereo-cameras need intensive computation power. Further limitations of ToF-cameras are visualized in figure 3.2.

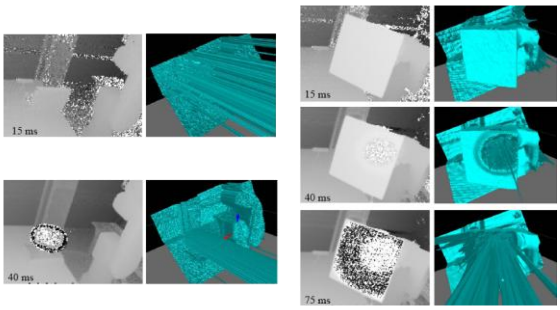

Figure 3.2: Limitations of ToF-cameras: Top left: Dark noise due to too few reflected modulated light. Bottom left: Interferences with other ToF-camera results in noise. Right images: Different exposure times for the same scene; near objects tend to over saturate the sensor. Images by D. Henrich, University of Bayreuth

3.1 Investigation of the depth accuracy of the ToF-camera

To compare a robot (as in figure 3.3) and its virtual model, the relation between intensity of the depth image and distance to an object has to be measured.

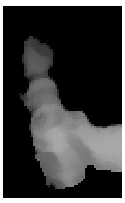

Figure 3.3: Depth image of a robot, captured by a Swissranger 3000 ToF-camera.

To investigate the depth accuracy of the ToF-camera, a planar object was captured from different distances by the camera. Ground truth is the manually measured distance and the camera has to look at the object perpendicularly. The reason why the object should be planar is that only one manually measured value is valid for the whole captured scene. By departing from the object, the object has to be relatively big in order to fit in the field of view of the camera. Hence, a region of interest should be defined.

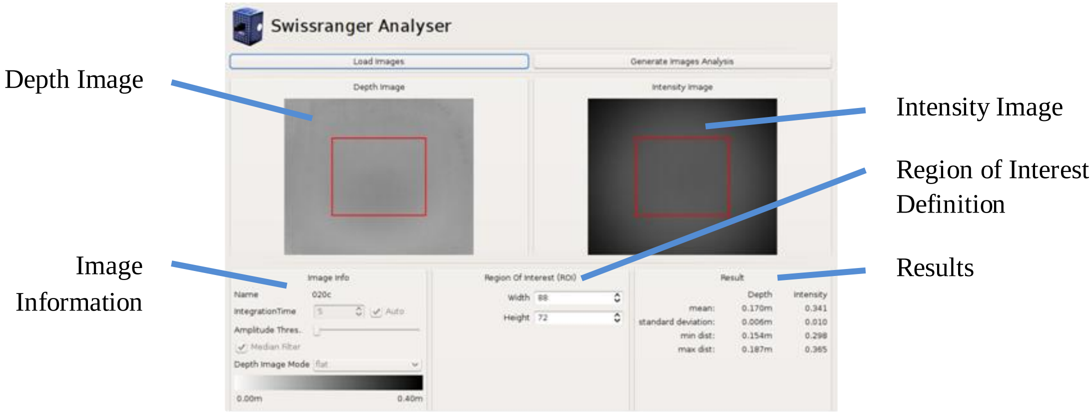

Figure 3.4: Swissranger Analyser – Graphical user interface to find out mean error and standard deviation of a set of ToF-camera images.

For every distance step, a set of images is taken to have a bigger data set in comparison to one image. With this data set the mean error and standard deviation can be calculated with a graphical user interface, seen in figure 3.4. With this user interface, it is also possible to set up a region of interest to reduce the field of view, so that the distance test can also be done from far distances and with small objects.

The depth test was conducted on two available planar objects with significantly different infrared reflectivity. A white wall with high reflectivity and a green door with weak reflectivity were chosen. The measured curves behaved very similarly, which can be seen in figure 3.5, left image. Knowing this fact, it makes sense to calculate the mean of the two curves (see figure 3.5, left, blue line). This mean curve then was approximated by a 5th degree polynomial (figure 3.5, left, blue, dotted line). This polynomial is now used in a low-level software layer (FRIEND::HardwareServer) and gives out the calibrated depth instead of the uncalibrated depth. The calibrated depth accuracy can be seen in figure 3.5, right, where the error is now reduced to 1 – 4 cm instead of before 5 – 15 cm. With this calibrated depth data, the following discrepancy detection will be more accurate.

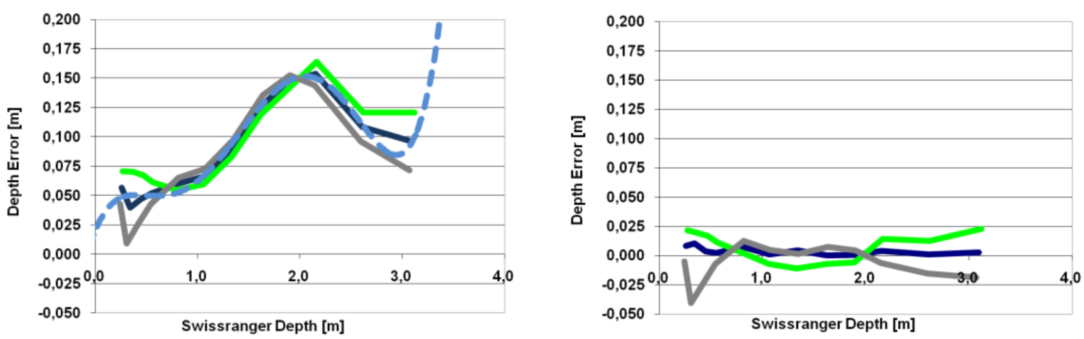

Figure 3.5: Depth accuracy without (left) and with (right) subtraction of a polynomial calculated from the data of the uncalibrated camera. Gray: White wall, green: green door, dark blue: mean of gray and green, dotted blue: 5th order polynomial of the dark blue mean curve.

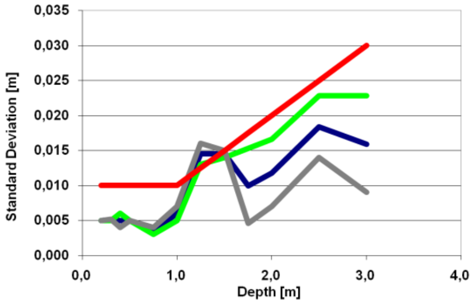

Figure 3.6: Standard deviation relative to distance. Gray: white wall, green: green door, red: manufacturers data, dark blue: mean of white wall and green door.

Additionally, the graphical user interface gave the result for the standard deviation of all pixels in the regions of interest and of all images in a set (see figure 3.6). In comparison to the manufacturer's information about the standard deviation, the depth test gave almost everywhere lower standard deviations. The reason for that can be the usage of the best-illuminated pixels in the middle of the images due to the region of interest definition.

3.2 Investigation of OpenGL fog for the depth image generation

OpenGL [7] is an open graphics acceleration language, which is already used in the FRIEND system to visualize the internal model (see figure 3.7, left). Because of this, the generation of the depth image has to be done with OpenGL. Generally, the drawing order is very important for 3D visualization software. All objects have to be ordered by their distance to the virtual camera so that nearer objects are drawn on top of farther object. Due to this reason, OpenGL strongly supports depth images. On the one hand over a so called depth buffer and on the other hand over virtual fog where objects slowly change their color to the fog's color as farther they are to the virtual camera.

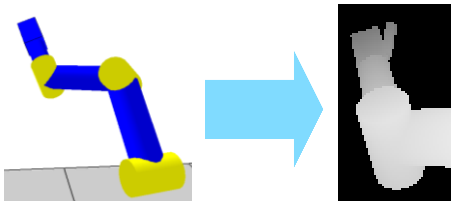

Figure 3.7: Depth image generation from 3D Model of a robot.

It has to be checked if the generated depth images behave identical to the depth image which is captured by a ToF-camera. This means that the intensity has to change linearly to the depth at a certain pixel.

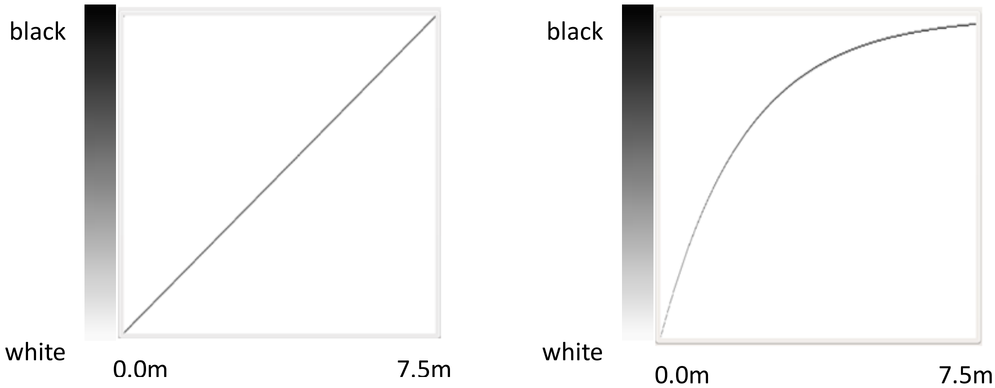

Figure 3.8: X-axis is the distance of the object to the virtual camera and Y-axis is the intensity of the object. Left image – linear fog behavior, right image – exponential fog behavior.

The procedure to test the linearity of the OpenGL fog was done with a planar object with white color, which is captured by the virtual camera. If black OpenGL fog is activated and the object is moved away from the camera, the object's color changes from white over gray to black. The analysis images in figure 3.8 were created with a white background and black points for each distance (centimeter wise) and the respective intensity of the object at that distance. It was also checked whether the whole object has the same color, which is the case. This check assures that the fog behavior is identical for the whole image (and not only for the center pixel) like the ToF-camera.

In figure 3.8, left image, the linearity is given for the full range of 0 to 7.5m, when the fog's exponential attribute is turned off. On the right image the fog's exponential attribute is activated. Like this, the generated image will be unequal to the real world scene captured by the ToF-camera. Subsequently, the exponential attribute has to be deactivated in the process of depth image generation to have a perfectly linear behavior, which was specified before.

4. Fast Discrepancy Detection

For the fast discrepancy detection algorithm the following is needed:

- Robot model,
- current calibration between robot and ToF-camera and
- the current joint angles of the robot.

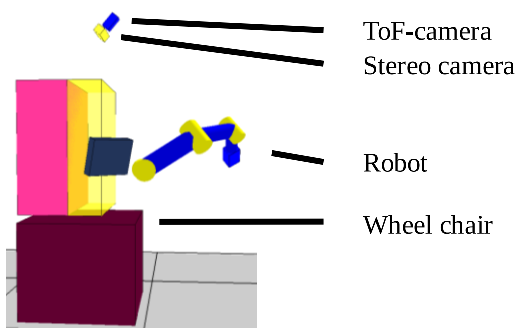

Figure 4.1: Robot model and static environment model.

The robot model has to be built according to the manufacturer's description. In the FRIEND system case, the very accurate CAD-model was transformed to basic geometric objects such as cylinders and cuboids (see figure 4.1). Here the distances between the joints, the angles to each other and the rough size of the describing basic geometric objects are important. Previous works on the FRIEND system provided the complete model [3]. Using the current joint angles the virtual robot can be transformed so that it looks like the real robot.

The current transformation between robot and ToF-camera depends in the FRIEND system on several translations. A detailed description about the translation between a stereo camera and the robot can be found in [8]. One problem handled by the calibration concept is suspension between the stereo camera and the robot, which is compensated by a mono camera. This camera is mounted a static attachment with the stereo camera and can capture a marker, placed on the robot base. As the ToF-camera has only a fixed transition to the stereo camera, the transformation data of the stereo camera can be used to easily calculate the transformation between ToF-camera and robot, handling the suspension problem, providing a highly accurate calibration. In addition, the stereo camera and the ToF-camera are mounted on a pan-tilt-unit, which can also change the translation between robot and cameras. This is also already handled by the calibration concept and can easily be used for the ToF-camera real-time calibration.

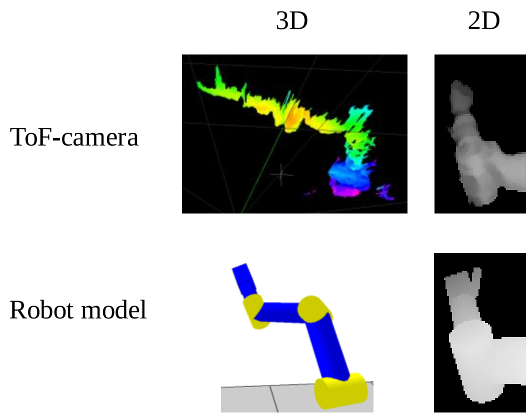

Figure 4.2: Representation of a ToF-camera image and the robot model in 2D and 3D.

The ToF-camera depth image provides the distance between the camera and an object for every pixel (in the case of the Swissranger 3000: 176x144 pixels). With the known field of view, stated by the manufacturer, a ray from the camera to the pixel results in a 3D-coordinate. When the 3D-coordinates are calculated for all pixels, a 3D-surface can be drawn, with small faces between each 4-point neighborhood. Because this 3D-surface and the 2D depth image can be converted bi-directionally without loss of information, these images are also called 2.5D. The 3D-surface is different to real 3D only because it is the view from only one camera. Nonetheless, the 3D-model of the robot is real 3D, because it can be viewed from infinite views and gives always a complete image.

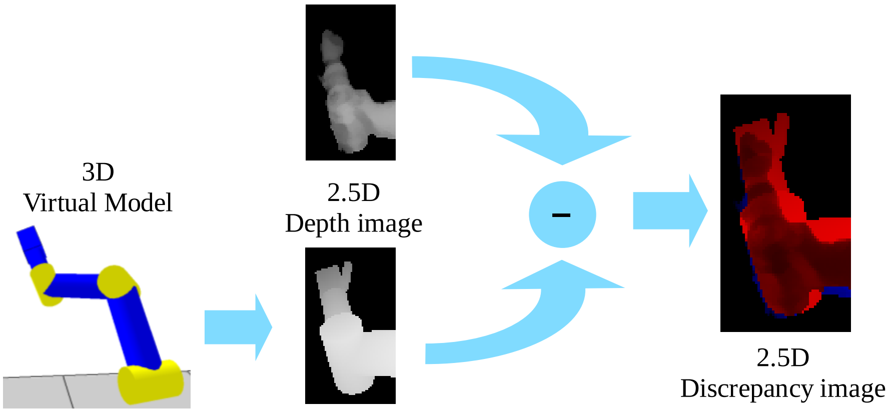

Figure 4.3: Discrepancy detection algorithm. Conversion of the robot model to a depth image and subtraction of the two depth images result in the discrepancy image.

To have a fast calculation of the discrepancy is to convert the 3D-model of the robot to 2.5D (see figure 4.3, bottom left), seen from the same virtual view as the real camera, provided through the above-mentioned calibration. A simple subtraction of these two depth images gives the discrepancy image. Instead, the comparison could also be done in 3D-space but this would result in a much higher computation time or low 3D-resolution and poor prediction of discrepancy. Another positive effect of the reduction to 2.5D is that normal image processing algorithms become possible and could enhance the result without implementing new 3D-algorithms.

In the right image in figure 4.3 the colors were changed from gray scale to blue and red to be able to display also negative values.

$$I(u,v)_{Dis} = I(u,v)_{Gen} - I(u,v)_{ToF}$$

Formula 4.1: Intensity of discrepancy image.

Formula 4.1 shows the intensity value of each pixel in the discrepancy image calculated from the subtraction of the generated depth image and the depth image captured by the ToF-camera. Because both depth images can have values between 0 and 255 the result is between minus 255 and 255. To be able to visualize negative values, the color space was switched from gray to RGB. Negative values become red and positive values become blue. No discrepancy is visualized by black color.

5. Experimental Results

The setup for the experiments was the FRIEND system with the ToF-camera mounted on top of a stereo-camera over which the transformation between robot and ToF-camera is known. This transformation has been acquired from the available stereo-camera calibration in addition to the manually measured translation and without rotation because the cameras are mounted on a common stand with which they look parallel.

Surveillance with about 10Hz can be achieved depending mostly on the exposure time of the ToF-camera. As the image resolution of the Swissranger 3000 ToF-camera is relatively low with 176x144 pixels, all image operations are fast, even on medium-class notebooks. The time to generate a depth image from the robot model strongly depends on the visible objects in the scene. In the FRIEND system, the robot model is very simple and consists only of basic geometric objects like cylinders and cuboids, which can be described by very few 3D-points.

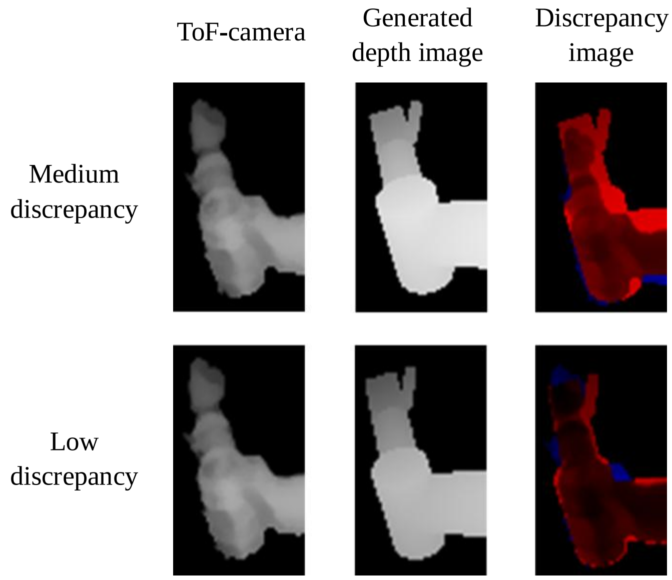

Figure 5.1: Visualization of different discrepancies between the real world scenes captured by a ToF-camera and generated depth images.

Summation of pixel intensities of the discrepancy image gives a direct benchmark for discrepancy. In figure 5.1, right column, two examples are provided: medium and low discrepancy. Depending on a threshold, the situation can be classified as dangerous or not. The threshold has to be between 0 and the logical maximum error count:

- Minimum error: 0
- Maximum error: <max intensity> * <image width> * <image height>

The theoretical maximum is the inversion of the real world scene, captured by the ToF-camera, which is very unlikely. Therefore the threshold should be probably relatively near to the minimal possible error.

When the current discrepancy is higher than the specified threshold, the situation is declared as dangerous. In this case an intervention has to be executed i.e. a signal is sent to a component, which stops the power supply of the robot. This process will establish a safe situation as the robot cannot harm the human in the FRIEND system, the environment or even the robot itself.

6. Conclusion and Future Work

A fast comparison method between a robot and its model was proposed. When this method is applied in a safety relevant task, it can be estimated, that the robot soon can harm itself, the environment or most importantly a human. In the FRIEND system it is extremely important to ensure the normal behavior of the robot because the user in his wheelchair cannot escape the malfunctioning robot.

It was presented how the fast discrepancy detection works and how it had been ensured that both input images for the discrepancy image calculation show specified linear behavior.

In future work the error source should be investigated. Therefore the search for this source can either be done in 2D or in 3D. In 2D, normal image processing algorithms can be used while in 3D certain joint angles of the robot could be visualized by transforming only the relevant information of the discrepancy image from 2D to 3D. Simulated joint errors can be added easily to the actual joint angles so that the model is incorrect. A fast decision of whether one joint angle or the calibration between the robot and the ToF-camera has an error can be implemented, because normally the discrepancy image should be black and a sudden change in the whole image will indicate a calibration problem.

Moreover, considering also the intensity image of the camera and not only the depth image can enhance the depth accuracy of the ToF-camera.

As the robot moves in its workspace, it might move out of the field of view of the ToF-camera. Due to this circumstance, the robot has to be followed by the camera with the help of a pan-tilt-head on which the camera is mounted.

Another possible future area of research is the prediction of movement of the robot by using the knowledge of path planning. Path planning is always used before the robot actually moves. With this knowledge the generated depth image can be generated before the robot actually moves and so the calculation of the discrepancy image is even faster.

References

[1] O. Prenzel: "Process Model for the Development of Semi-Autonomous Service Robots"; *PhD Dissertation, University of Bremen*; 2009; Shaker Verlag, Aachen, ISBN 978-3-8322-8424-4

[2] C. Martens, O. Prenzel, and A. Gräser: "The Rehabilitation Robots FRIEND-I & II: Daily Life Independency through Semi-Autonomous Task-Execution"; *Rehabilitation Robotics*; 2007; I-Tech Education Publishing, Vienna, Austria, ISBN 978-3-902613-01-1

[3] D. Ojdanic: "Using Cartesian Space for Manipulator Motion Planning - Application in Service Robotics"; *PhD Dissertation, University of Bremen,* 2009

[4] MESA Imaging Website: http://www.mesa-imaging.ch

[5] SafetyEye Website: http://www.pilz.com/products/sensors/camera/f/safetyeye/index.jsp

[6] M. Fischer and D. Henrich: „Surveillance of Robots using Multiple Colour of Depth Cameras with Distributed Processing“; 2009; Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), Aug 30 – Sep 02, 2009, Como/Italy

[7] OpenGL Website: http://www.opengl.org

[8] T. Heyer, S.M. Grigorescu, and A. Gräser: “Camera Calibration for Reliable Object Manipulation in Care-Providing Robot FRIEND”; *ISR/Robotik, Munich, to be published, June*; 2010