



CENTER FOR  
MACHINE PERCEPTION



CZECH TECHNICAL  
UNIVERSITY IN PRAGUE

TECHNICAL REPORT

# Visual compass for the NIFTi robot

Tomáš Nouza

nouzato1@fel.cvut.cz

June 27, 2013

Available at

[https://cw.felk.cvut.cz/doku.php/misc/projects/nifti/sw/start/visual\\_compass](https://cw.felk.cvut.cz/doku.php/misc/projects/nifti/sw/start/visual_compass)

**Supervisor: Ing. Michal Reinštein, Ph.D.**

Center for Machine Perception, Department of Cybernetics  
Faculty of Electrical Engineering, Czech Technical University  
Technická 2, 166 27 Prague 6, Czech Republic  
fax +420 2 2435 7385, phone +420 2 2435 7637, www: <http://cmp.felk.cvut.cz>

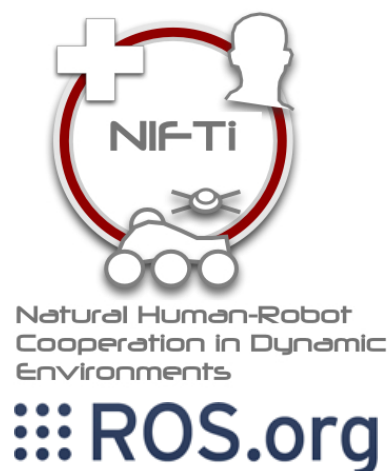
# Visual compass for the NIFTi robot

Tomáš Nouza

June 27, 2013

## Abstract

A visual compass for the NIFTi [1] robot running ROS (Robot Operating System) [2] was developed as a semester project during the course A3M33IRO (in Czech called “Inteligentní robotika”). This compass serves as a computationally undemanding alternative for the visual odometry and provides information about the robot rotation in the horizontal plane (z-axis). The system was successfully evaluated in the indoor environment.



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	NIFTi robot . . . . .	3
<b>2</b>	<b>Algorithm</b>	<b>4</b>
2.1	DFT speed-up . . . . .	5
<b>3</b>	<b>Usage</b>	<b>5</b>
3.1	Parameters . . . . .	5
<b>4</b>	<b>Evaluation</b>	<b>6</b>
4.1	synthetic data . . . . .	6
4.2	360° turn . . . . .	7
4.3	Forward movement . . . . .	7
4.4	Free race . . . . .	8
<b>5</b>	<b>Conclusion</b>	<b>8</b>

# 1 Introduction

The navigation of the mobile robot is a necessary part for any of its autonomous functions. The robot rotation can be measured from the odometry but namely on the belt platforms (which is the case of the NIFTi robot) this is very inaccurate. Second widely used method is integration of measured acceleration from the gyroscope but this process is susceptible to the noise data. The third option is to use the visual information from the camera and calculate the rotation between two following frames. Even this method has many fail cases e.g. a men walking round the robot. The correct way to obtain a reliable information about the real robot rotation is a data fusion of all above mentioned methods.

The NIFTi robot is equipped with the Point Grey Lady-bug 3 omnicaamera from which spherical projection of the robot surrounding can be obtained. This projection characteristic is that every horizontal line covers the  $360^\circ$  view and every vertical line has points that shares the same angle deviation from the center of image. Using this knowledge there is no need to complicatedly reconstruct the scene around the robot but simply calculate the image horizontal shift.

## 1.1 NIFTi robot

The NIFTi robot is being developed in the NIFTi project. It involves 6 different European universities, two fire departments and a robot manufacturer. The robot should be suited for outdoor conditions. One of its purposes is to take part in Urban Search and Rescue (USAR) missions, where it should map the situation and help locate victims of a catastrophe. The robot is shown in the Fig. 1.



Figure 1: The NIFTi robot.

## 2 Algorithm

Almost half of the camera image is occupied by the robot body which is static and does not change when the robot turns. The upper section of the image consists of the scene over the robot which is more sensitive to the robot's forward movement than the rest of the scene. With respect to the above mentioned restrictions, only a region of interest (ROI) from the Fig. 2 is selected for the further computing.



Figure 2: The region of interest for the visual compass.

The ROI is then separated to individual horizontal lines which are correlated with the corresponding lines from the previous image. The shift with a maximum correlation coefficient is then selected as the rotation shift and the angle is calculated using:

$$angle = \frac{shift \cdot 360}{image\ width} \quad (1)$$

where the *shift* and *image width* are in pixels and *angle* in degrees.

The resulting rotation angle is calculated as a weighted average where the correlation coefficients are used as the weights.

## 2.1 DFT speed-up

The correlation can be from a definition calculated using:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2)$$

where the  $\bar{x}$  and  $\bar{y}$  are mean values. Disadvantage of this approach is the quadratic computational complexity. The better way is to use a discrete Fourier transformation and use the following formula:

$$\mathbf{r}_{xy} = \text{idft}(\text{dft}(\mathbf{x}) \cdot \text{dft}(\mathbf{y}^*)) \quad (3)$$

Main advantage of this formula is that  $\mathbf{r}_{xy}$  is a vector containing correlation coefficients for all the image shifts. Finding a maximum in this vector is a problem with linear computational complexity. Disadvantage of this approach is that the output value is not normalized in the range  $< -1, 1 >$  as the definition approach.

## 3 Usage

The algorithm is distributed as a ROS package. For running it is sufficient to `rosmake` the package and `roslaunch` the relevant *launchfile* depending on the field of usage. There are two *launchfiles* prepared:

- `visual_compass.launch` for the standard usage (e.g. on the robot)
- `visual_compass_BAG.launch` allows to specify the *bagfile* to be played and also republishes the compressed camera images to the raw format

### 3.1 Parameters

All the configuration parameters are set in the *launchfile*. As the only input to the node is the visual information given by the omnicaamera, there must be set the corresponding *topic* where the camera node is published. For this purpose there is the parameter `compass_image`. If not set, the default value is `/viz/pano/image`.

The output from the *node* is the standard `nav_msgs/Odometry` message which produces the quaternion orientation and the angular twist of the robot. The position and the linear twist are zeros hence the visual compass does not

measure it. If not set by the parameter `compass_output_topic`, this message is published on the topic `/compass_pose`.

For the debug purposes there is also a possibility to print the current measured angle to the terminal using the `ROS_INFO` command. To enable this feature there is the parameter `compass_textout` which has the *true/false* values.

## 4 Evaluation

The algorithm was tested in following experiments:

- synthetic data
- $360^\circ$  turn
- forward movement
- free race

### 4.1 synthetic data

The synthetic data were generated at the begging phase of the development by the simple shifting the panorama image by the desired angle for the testing. The example of these shifted images is in the Fig. 3. Even the algorithm



Figure 3: Panorama images turned by  $-90^\circ$ .

works 100% correct on these data it is not saying anything about the practical usage because these images are shifted including the robot body and have no noise.

## 4.2 360° turn

To test algorithm in the real situation, the robot was turned three times on the one spot with close surrounding, where the operator set the 5 m distant door frame to the center of the camera as a reference. After three turns, the door frame was set to the same position and measured angle was  $1091.11^\circ$  which leads to an error of 1%. The fixation to the door frame is shown in the Fig. 4.



Figure 4: Position of the robot before (left) and after (right) three rotations.

The same test was also performed in the exterior where the robot was turned three times near the wall very slowly and the robot orientation was fixed according to the straight joints of concrete panels on which the robot was moving. The measured angle was  $1077.84^\circ$  which leads to an error of 0.2%. The outside experiment was also made in the middle of the school yard where the closest obstacle was about 2 m distant. The robot rotated five times at full speed and the measured angle was  $1799.14^\circ$  which leads to the error lower than the operator was able to measure.

## 4.3 Forward movement

In the next experiment the robot operator followed the paving on the floor to drive the robot in the forward direction in the narrow passage. After driving the 6 m distance, the measured angle was  $-6.59^\circ$ . The narrow passage is shown in the Fig. 5.

The same experiment was repeated in the outdoor condition where the robot were following the straight joints of concrete panels. On the 7 m long race the measured angle was  $-4.93^\circ$ , on the 25 m the angle was  $17.72^\circ$ . Both experiments ended next to the wall.



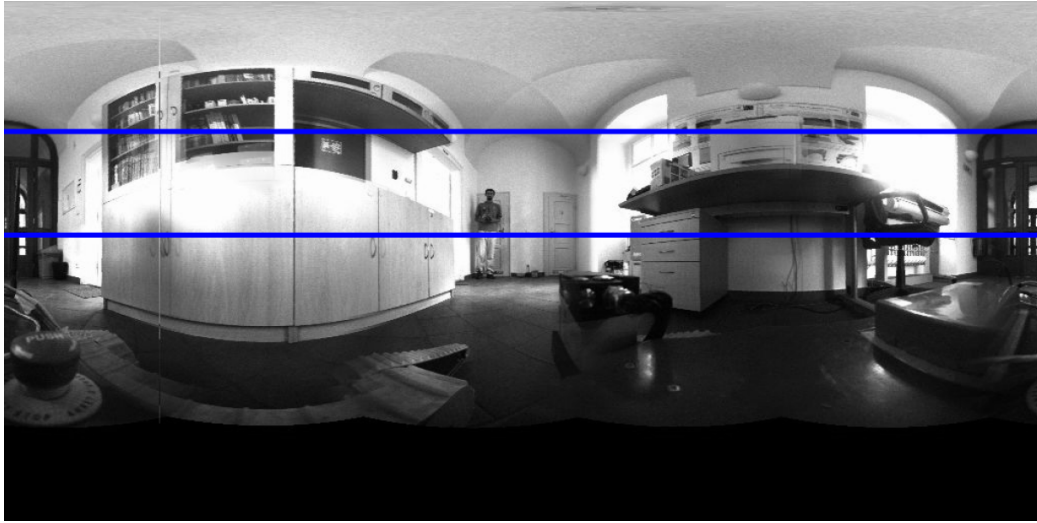


Figure 5: The narrow passage where the robot followed the paving as a reference of direct movement.

#### 4.4 Free race

To test the long term stability, the robot was driven through the narrow passage two times and made a  $180^\circ$  left turn at each end. Starting and ending orientation was fixed to be parallel with a wall where the robot distance from the wall was about centimeter. The resulting measured angle was  $330.7^\circ$  which means  $30.7^\circ$  error on the approximately 16 m distance.

The similar trajectory was also driven in the outdoor environment without the narrow passage and the resulting error was  $0.58^\circ$  which is below the precision that the robot operator was able to set.

In the last experiment the robot were following the square trajectory of 20 m length and the resulting error was  $8.3^\circ$ .

## 5 Conclusion

The algorithm for visual compass measuring the robot yaw from an omni-camera was created as a ROS package and was successfully tested on the NIFTi robot. Although the resulting accuracy is not sufficient for the long term operation based only on this sensor, the accuracy of the measurement where the dominant movement component is the rotation is good enough to improve accuracy of other methods.

The dominant angle errors are caused by the close obstacles which are in

the panorama view moving faster than the remote environment. In the future work this could be eliminated by separating the panorama image to the sectors and calculating the image shift in each sector separately. The resulting angle should be the median of these sector angles which could eliminate the influence of the close obstacles.

## References

- [1] “Natural human-robot cooperation in dynamic environments.” <http://www.nifti.eu>. Accessed: 26/04/2013.
- [2] “Robot Operating System.” <http://www.ros.org/wiki/ROS>. Accessed: 26/04/2013.