

The Autonomous City Explorer Project: Towards Navigation by Interaction and Visual Perception

Quirin Mühlbauer, Stefan Sosnowski, Tingting Xu, Tianguang Zhang, Kolja Kühnlenz and Martin Buss

Institute of Automatic Control Engineering

Technische Universität München

D-80290 Munich, Germany

{qm, sosnowski, xu, zhang, koku, mb}@tum.de

Abstract—The Autonomous City Explorer (ACE) project has the goal, to develop a robot which is capable of finding its way to a given destination in an unknown urban environment. An exemplary mission is to find the way from our institute to the marienplatz, a public place in the center of munich, without any prior knowledge or gps information. Inspired by the behavior of humans in unknown environments, ACE must find its way by asking pedestrians. The distance of the route is about four kilometers and includes heavily traveled roads and crowded public places. In order to navigate safely in an unknown urban environment, some challenges arise for the vision system. Robust human detection, tracking and the estimation of human body poses is essential for natural interaction with pedestrians. Furthermore, the robot needs to be able to detect sidewalk and crossroads. A visual odometry system is used to support the conventional navigation. This paper describes both, an architecture of the vision system used for ACE and the algorithms used to deal with the described challenges.

I. INTRODUCTION

Most of the current robots have been used for specific industrial tasks, such as working at an assembly line. Other robots have been developed to assist humans in household activities [1] [2] or guide humans through museums [3] [4] [5]. Autonomous cars have been developed for outdoor navigation on roads. One big goal of research in robotics is to bring robots into the real world, so they have to be able to interact and act safely in both, indoor and outdoor crowded human environments. In order to address these challenges a robot capable of autonomously exploring highly populated urban environments is created in the Autonomous City Explorer (ACE) project. The main mission for ACE is to find its way to the marienplatz, a public place in the center of munich. The robot must be able to perform vision guided dialogue-based navigation in an unknown urban outdoor environment without any prior map knowledge or GPS information. Considering the highly populated environments, we constrain our vision system into human detection and tracking as well as body pose estimation for the human-robot interaction, and sidewalk detection as well as visual odometry for the autonomous navigation.

An example for natural human-robot-interaction is shown in figure 1, where ACE is interacting with a pedestrian at the marienplatz. The pedestrian is pointing into the direction ACE has to move. Consequently, the robot must be able



Fig. 1. The ACE robot interacting with a pedestrian at the marienplatz, a crowded public place in the city center of Munich.

to find a human, drive towards them and initialize the interaction. Most proposed human detection models are based on feature extraction and classification [6]. Most of them are robust but not real-time capable or highly dependent on high resolution, which is not suitable for highly dynamic outdoor environment. Some skin color based strategies are also proposed [8] and [9] which can process in real-time but not robustly enough.

As a speech based dialog system, the most natural way of interaction, is impossible with the background noise at heavily frequented public places or traffic noise, the human-robot-interaction is performed by a touch screen. To enhance the natural interaction, ACE has the ability to speak and to recognize human body poses. Some algorithms for the estimation of human body poses using camera systems are also proposed [10] [11] [12], which serve as foundation of our work.

Another important ability for a robot, which has to drive on a sidewalk, is to detect the sidewalk robustly. ACE is not allowed to cross junctions or to enter the road, so crossroads have to be detected robustly. For an effective and safe navigation in outdoor environments, visual odometry also plays an essential roll. Optical flow based algorithms [13][14][15][16][17] are widely used. For a more accurate ego motion estimation, high-speed sensors and processing are still lacking.

The remainder of this paper is organized as follows: The

technical details about ACE, particularly the multi-focal camera platform, is introduced in Section II. Section III describes the architecture of the vision system. The proposed approaches to the several challenges of the application and corresponding results are described in Section IV. Section V gives an outlook to future works.

II. TECHNICAL DETAILS

Significant effort has been devoted to develop a robotic system that meets the requirements and challenges outlined in the previous section. In this section, the hardware components and the software architecture are explained. The ACE robot in its newest revision consists of a mobile platform with two differential drive wheels, four castor wheels, as well as an upper body with a camera head, a communication system and two PCs. The upper body was completely redesigned to achieve better stability due to a lower center of mass.

A. Multifocal Camera Head

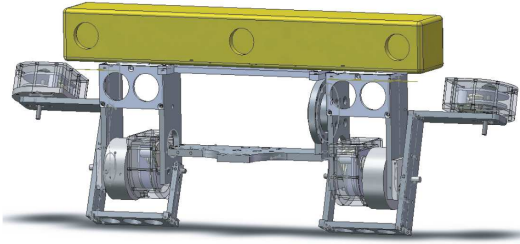


Fig. 2. New revision of the high-performance active camera platform [18]

With the new revision of the ACE setup, a new multi-focal high-performance vision system is also introduced. The design is based on the multi-focal vision system, which has been developed for the humanoid robot *LOLA* [18]. It comprises several vision sensors with independent motion control which strongly differ in fields of view and measurement accuracy. High-speed gaze shift capabilities and novel intelligent multi-focal gaze coordination concepts provide fast and optimal situational attention changes of the individual sensors. Thereby, large and complex dynamically changing environments are perceived flexibly and efficiently.

This multi-focal vision system generalizes the foveated vision concept by introducing independent motion control of several vision sensors, thus adding more flexibility in sensor resources allocation [18]. This feature is particularly beneficial in robot navigation and scene observation providing higher robot localization accuracy and tracking performance than conventional systems.

The vision system consists of a wide-angle stereo-camera mounted on a central pan/tilt-platform, see Fig. 2. As an upgrade from the previous vision system, the main camera is now a 3-sensor, multi-baseline Bumblebee XB3 by Point Grey Research, with enhanced flexibility and accuracy because of the switchable baseline. Additionally, two telephoto cameras are gimbal-mounted on the central platform with 2 DoF each. Aperture angles of approximately 85° (wide)

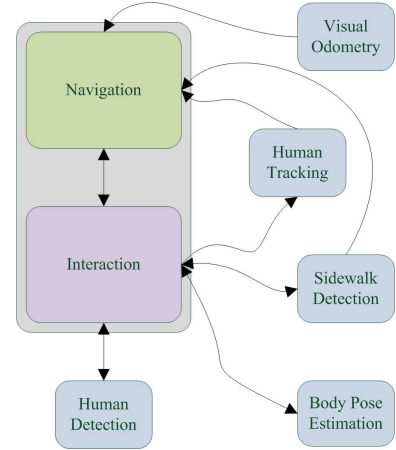


Fig. 3. System architecture of the vision system in combination with the navigation and interaction of ACE

and 20° (telephoto) and focal-lengths of 2 mm and 25 mm, respectively, are provided. The central platform is driven by DC drives with harmonic drive gears, the gimbal-mounted cameras by brushless DC direct drives providing high torques and accelerations at small dimensions and weights. Top open-loop speeds and accelerations measured are $8400^\circ/s$ and $100000^\circ/s^2$. An embedded RISC processor (MPC555, Motorola) controls the camera motions on joint-levels. The position feedback for the control loop is provided by incremental magnetical encoders (512 counts per motor-revolution) on the dc-motor side and processed in the RISC processor. For the brushless-motor side, position is measured by light-weight and small optical absolute encoders, which were developed specifically for this camera head. The position is encoded in a 16bit gray code on the encoder disc, processed directly in the respective sensor and can be requested via I2C. The system is encapsulated and accepts camera pose commands from a higher-level decision and planning unit via a CAN-based interface. The system body is made of aluminum alloy. Overall dimensions are (37x30x5)cm and the weight is 2.2 kg.

III. ARCHITECTURE

Figure 3 shows the architecture of the vision system and the connections to the *navigation* and *interaction* modules of ACE. More information about this two modules can be found in [7]. The vision system can switch between several states and is controlled by the *interaction*. The *visual odometry* is running continuously to support the conventional odometry and localization of the navigation module. When the robot wants to interact, the module *human detection* is activated and the vision system will search for humans and send a signal to the *interaction*, when a human is detected. Now, ACE will ask the pedestrian for a way and the user will point in a certain direction. The module *body pose estimation* will be activated and will send the estimated body pose to the *interaction*. Now, the robot will turn its head in the given direction and will make a picture to specify the direction and

the robot will drive in this direction. When ACE is following a sidewalk, the vision system needs to activate the module *sidewalk detection* to follow sidewalks safely and to detect crossings. The detection of crossroads is performed by both, detection of traffic signs and traffic lights and by analysis of the sidewalk's shape. In order to provide a natural way of interaction, semantic information is passed to the *interaction* module, e.g. ACE will be able to process user information like "follow the road to the next crossing, then turn right until you reach the next but one crossing". When ACE needs to cross a road, the module *human tracking* is activated and the robot will follow a certain person which will guide ACE across the street.

IV. SYSTEM DESCRIPTION

This section describes the subsystems and the used algorithms. Furthermore, some results of the modules will be presented.

A. Robust Human Detection and Tracking

For a successful human-robot interaction, pedestrians should be detected at first. Considering the common characteristics of pedestrians, for instance, the skin color, motion and the upright posture in the street, we propose a top-down biased, task-relevant saliency map model, which predicts the positions of potential pedestrians in an input image.

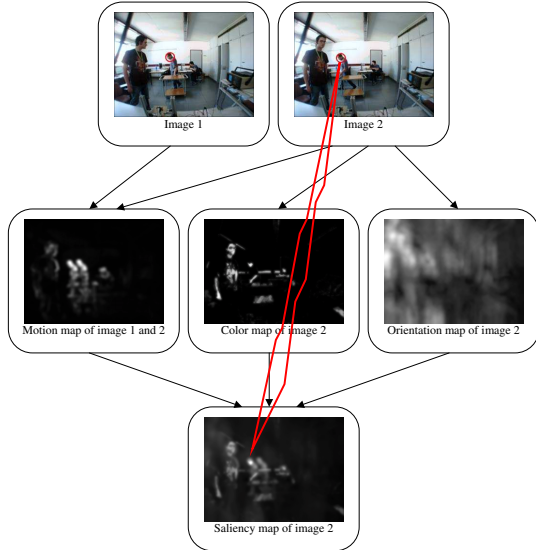


Fig. 4. Feature maps and saliency map

Fig. 4 illustrates the human detection model. From the incoming image sequence two consecutive images are being taken into account to compute the saliency map. This saliency map is derived from feature maps for color, orientation, and motion. For color and orientation, feature maps are only computed for the second image, while the motion feature map is computed using the difference between color values of both images. Each feature map is going through a normalization process before the weighted sum is computed to get the saliency map. Color and motion feature maps were

weighted almost equally strong, about three times stronger than the orientation map.

When implemented on the robot, the most salient spot in the saliency map will then become the center of attention for the camera head. As a final result the robot will always look directly at the pedestrians, which also shows its current interest in interaction with one of the pedestrians.

1) Human detection:

Color map: To achieve robustness, the color feature map is the weighted sum of four feature maps in different color spaces. For each color space the model is given rules by which a certain pixel is either determined to be skin color (pixel set to 255, white) or not (pixel set to 0, black). Color spaces used by this model are RGB, normalized RGB, HSV and YCrCb.

Motion map: The input data for the motion feature map is the absolute value of the difference between the gray scale values for each pixel in two consecutive images. The result is one gray scale image showing intensity changes from one image to the next. Note that areas in this image showing great intensity are not necessarily the areas where the greatest motion can be seen.

One problem in computing motion map is fast motion caused by shaking of the camera and thus causing little offsets between images in the sequence. To reduce this effect in theory, the gray scale motion image as described above is not only computed once, but several times while the two input images are shifted towards each other by one pixel in one dimension at a time. The resulting images are compared to each other and the one with the least overall motion detected serves as motion feature map. Since camera movement shows effect all over the image, motion to be detected in the scene will still show effect after this stabilizing process, while camera shaking is compensated (see Fig. 5). In terms of a mathematical formula, this is a simple optimization problem:

$$\min_{k,l} \sum_{i=1}^N (I_1(x_i, y_i) - I_2(x_i - k, y_i - l)) \quad (1)$$

where $k \in (-k_{max}, k_{max})$ and $l \in (-l_{max}, l_{max})$. N is the number of pixels in the image, where each pixel can be addressed by a pair (x_i, y_i) . k and l represent the offset in x- and y-direction between the two gray scale images. In order to use this method more efficiently regarding computational cost, N is set to a smaller value representing several smaller areas in the image, while k_{max} and l_{max} are chosen within reasonable limits.

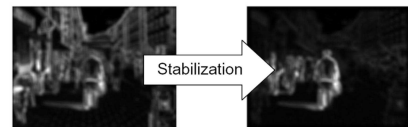


Fig. 5. Illustrating the effect of the stabilization algorithm

Orientation map: Considering that most pedestrians stand uprightly, we also integrate an orientation map into the final saliency map. We use Gabor filters introduced in [20]. In order to apply the filter at different scales, the input image is resized accordingly to save computation time. Modeling the task to detect humans in the scene, Gabor filters at an angle of 90° and scales of 20, 40, and 80 pixels were used. The latter two scales were weighted double.

2) *Human Tracking:* The camera platform is controlled to locate the most salient position of the saliency map in the center of the images. Using template matching we acquire the respective salient positions in the left and the right images from the stereo camera, compute the 3D position using stereo triangulation, and control the camera platform.

To lower the computational cost, we constrain a search/interest area for human detection as follows: If the salient position was very close to the principle point, it would be reasonable to only have a look at a smaller area in the center of the image for the next time step. On the opposite, if the search area has already been reduced and the most salient point was close to the image boarder it would make sense to enlarge the size of the interest area to be processed again. The size of the interest area varies between 250x220 to 640x480 (see Fig. 6). This method also helps rudimentarily to keep the system focused on one target / human being and not switch back and forth between several points of interest.

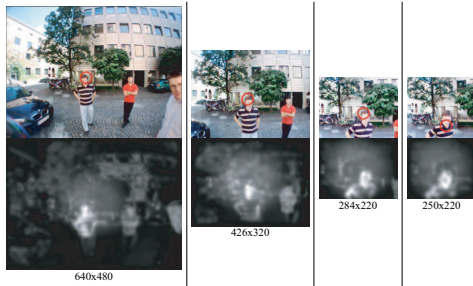


Fig. 6. Interest areas during the image processing loop

3) *Experimental Evaluation:* Table I shows the result of our human detection model and the interest area used during an 11 minute test run. Pedestrians were usually located in a range up to 8 meters away from the camera. The results in the outdoor environment are very pleasing.

Since the interest area for human detection varies, the computational cost for each step also varies. Working on the hardware described previously, the maximum computation time for one time step using 640x480 is 0.7s, while the minimum computation time using 250x200 is 0.2s.

B. Gesture Recognition

A robot working in close collaboration with humans must be able to interact safely. To increase its acceptance, the robot should have the ability to recognize the gestures, actions and the intention of its companions, so it is essential to estimate the human body pose in real time. The algorithm presented in the following is able to leave out body parts and

	Indoor environment		Outdoor environment	
	abs.	percent.	abs.	percent.
Humans detected	287	51.7 %	498	92.6 %
Humans not detected	268	48,3 %	40	7.4 %
No humans in the scene	319		53	

Interest areas used for image processing				
640 x 480	303	34.7 %	135	22.8 %
564 x 480	23	2.6 %	6	1.0 %
426 x 330	25	2.9 %	8	1.4 %
426 x 320	115	13.2 %	54	9.1 %
376 x 330	49	5.6 %	23	3.9 %
376 x 320	13	1.5 %	5	0.8 %
284 x 220	79	9.0 %	39	6.6 %
250 x 220	267	30.5 %	321	54.3 %
Total number of images	874		591	

TABLE I
RESULTS OF A 11 MINUTES TEST RUN

is therefore able to deal with occluded body parts. A more detailed description can be found in [22].

In a first step, possible humans need to be detected, e.g. by using a skin color filter. A disparity map containing depth information is computed using a stereo matching algorithm. It leads to a three-dimensional representation of the scene. Starting with the detected skin parts, our algorithm segments this point cloud into smaller clusters. The possible matches are then verified, and the body pose is estimated using a kinematic human model with 28 degrees of freedom. As our algorithm is capable of dealing with arbitrary three-dimensional representations, it can easily be adapted to use a three-dimensional laser range finder instead of a stereo camera system. The reduced (the hand has not been considered) human model has 15 links and 28 degrees of freedom. Each link provides one to three degrees of freedom, and is rotated around the axis of the coordinate system of the link it is connected to. As the hands and the feet are too small to be detected robustly by the stereo matching and they are not relevant to estimate the pointing direction, they have not been taken into account. Hence, this model is reduced by 10 degrees of freedom. To validate the human model, the link lengths and the angles between the links are considered. For each link, a minimal and a maximal value for each of the parameters have been estimated. As they are coupled, the left and right shoulder are treated specially.

1) *Segmentation:* The three-dimensional representation of the scene acquired by the stereo vision system contains a large colored point cloud. With the skin detector, we know what parts of the point cloud may belong to a human body. Consequently the remaining parts of the human body have to be found. The results of the skin color detection and the corresponding color point cloud obtained by stereo vision can be found in figure 7. Starting with the detected skin parts, the segmentation algorithm searches for locally adjacent structures and will create one cluster for each detected skin part. As the skin detector may deliver false positives and



Fig. 7. Image with detected skin parts (left). This image has been taken with a stereo camera with a high aperture and has already been rectified. The right side shows the three-dimensional reconstruction of the scene using a stereo matching algorithm.

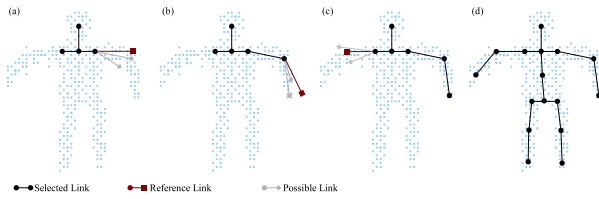


Fig. 8. Illustration of the selection of the best link in 4 steps.

two or more skin parts of the same human are detected, the clusters have to be validated.

2) *Algorithm for the Body Pose Estimation:* After one cluster has been computed and validated for each human in a scene, they can be used to extract the body poses. The algorithm will be executed once for each cluster, so the accurate number of humans can be found. As the algorithm should be able to deal with all different types of colors and clothes, color provides little useful information and is consequently not used. Thus, the segmentation and estimation of the body pose is performed by using only the position of the points. Starting with the head, the algorithm will try to fit the attached links iteratively. If a link is not part of the image or is occluded by an object in front of it, no valid fit can be made and the link and all links attached to it will not be included in the resulting human model. The algorithm is illustrated in figure 8, while figure 9 shows the results from different scenes. The first row shows the undistorted images as seen from the camera, with the detected skin parts. In the next two rows the estimated human body poses are shown from two different points of view.

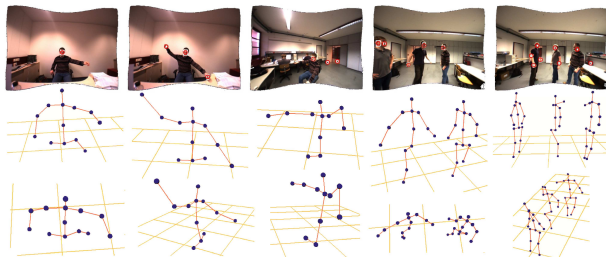


Fig. 9. Results of the human body pose estimation.

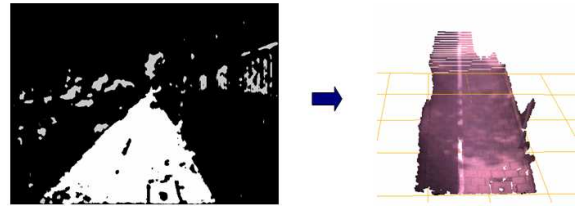


Fig. 10. Illustration of the sidewalk detection algorithm.

C. Sidewalk and Traffic Sign Detection

Safety is a main issue for a robot navigating through an urban environment. As ACE must drive together with pedestrians on the sidewalk, the robot is not allowed to travel on streets or to cross streets autonomously. When ACE needs to cross a street, it will follow a certain person. Hence, the robot needs the ability to robustly detect sidewalks and crossroads.

1) *Sidewalk Detection:* To detect the sidewalk, an algorithm based of the algorithm presented in [23] was used. The proposed method uses one camera and is real time capable at a high resolution. The algorithm uses the assumption, that the area in front of the robot is free. This assumption can be cross checked with a laser scanner. If the area is free, the texture in the area is compared to the texture of the rest of the image. All areas in the image with a similar texture are assumed to be free. To deal with different terrain types, an extended version of the algorithm was equipped with a memory. Textures which are known to be free are saved and this history of textures is used to detect free areas in the image. In order to increase the robustness for sudden changes of the terrains texture, the oldest textures are weighted with a low weighting factor and the new images with a higher one.

The results of the sidewalk detection are saved in a 2.5-D map, which is not recreated in every computation step. The last 10 results of the sidewalk detection are merged into one map. As the algorithm runs with 15-20 Hz, the movement can be neglected. Furthermore, the shape of the sidewalk was used to detect crossings and to estimate, if the robot is driving on the left or on the right side of the street. A simple classifier was trained with measurements to perform this detection. Combined with the detection of traffic signs, which will be explained in the next section, a robust detection of crossroads and a safe following of sidewalks can be ensured.



Fig. 11. Images of the traffic signs, that can be found at every junction in the city center of Munich, the operational area of ACE.

2) *Traffic Sign Detection*: For the detection of crossroads, the assumption that every crossroad in the area where ACE is driving is equipped with traffic signs or traffic lights can be made. The algorithm developed for ACE searches for both, for traffic lights and the traffic signs shown in figure 11. OpenCV's rapid object detection was used to detect the traffic lights and signs. To achieve good results, over 10000 images have been made to train the haar like features [21]. One classifier has been trained for each traffic sign or light. The achieved results are quite good, the hit rate lies within a range of 79 to 88% and the number of false positives within 2 to 6%, depending on the classifier. When the robot is approaching a crossroad, the traffic signs will be seen in more than one image. The false positives will only occur in one single image, so by tracking the detected traffic signs, the false positives can be detected and discarded. If the algorithm misses a traffic sign in one single image within a sequence of images, it will be able to detect the sign in the rest of the sequence. Consequently, most of the not detected signs and false positives can be handled.

The computation time of the algorithm is about 100 ms for each classifier, so the total computation time is 1.2 seconds. As the robot drives with less than 0.5 meters per second, the computational time is adequate for a robust detection of traffic signs. As the used camera has a large field of view, the computational time can be decreased by selecting a region of interest.

D. Topological Image Processing

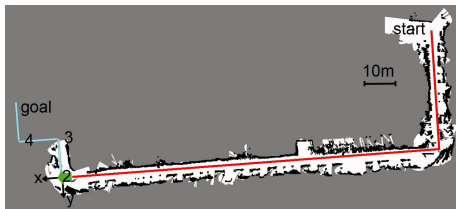


Fig. 12. Topological route graph with 3 topological and 3 metric nodes.

With the gestures, the human users provide the robot with information how to get from crossing to crossing. As described above, the robot has the ability to recognize junctions. Based on this information the robot creates a topological route graph as a representation of the path that lies ahead. While it follows this path the robot updates the graph to a metrical route graph with the data from the real

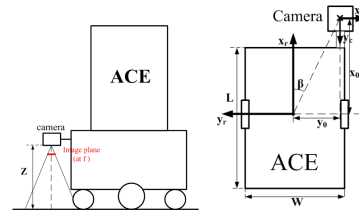


Fig. 13. Cutaway and planform of the

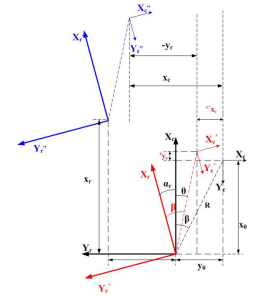


Fig. 14. Geometry relationship with variables definition

environment. A more detailed description of this process can be found in [7]. Figure 12 shows such a route graph with 6 nodes. When the robot has gained topological route knowledge (node *start*, 1 and 2, it can start following the route towards the goal (nodes 3, 4 and *goal*). Starting at a crossing it follows the street in the given direction until it reaches the given goal.

E. Visual Odometry

The goal of ACE is to navigate in an unpredictable and unstructured urban environment. For achieving the aim, accurate pose estimation is one of the preconditions. By now, ACE only has the information from the angleencoders on the wheels. If there are sands, cobblestone on the ground, the wheels will slip, which causes an inaccurate localization. Therefore, we want to use the visual information to support the localization. In this subsection a visual odometry system is presented to estimate the current position and orientation of ACE platform. The existing algorithms of optical flow computation are analyzed, compared and an improved sum-of-absolute difference (SAD) algorithm with high-speed performance is selected to estimate the camera ego-motion. The kinematics model describing the motion of ACE robot is set up and implemented. Finally the whole odometry system was evaluated within appropriate scenarios.

1) *Modeling*: Because ACE will explore the outdoor urban environments, e.g. the city centre of Munich, and communicate frequently with the humans, so the camera for visual odometry may not gaze directly forward. For avoiding the disturbance of moving crowd, the camera is mounted in the front of ACE and the optical axis is perpendicular to the ground (see Fig.13). The relative position between the camera and the robot does not change in the whole process. Any actuated motion of the robot will result in a movement of the camera relative to its original position. Because the displacement between camera and ground in z-direction is much smaller than the distance between camera and ground z , we can approximately assume that the ground is a flat plane and the ACE-platform displaces without any roll and pitch angle. Based on this assumption only 3 variables must be considered: the movements in x and y directions and the orientation around the z axis. We divide the movement of robot in two parts (see Fig. 14). Firstly, it rotates with an angle of without any translation. Then, the robot has

movement of (T_x, T_y) . After that, the three variables of camera relative to its original position can be denoted as:

$$\alpha_r = -\alpha_c \quad (2)$$

$$x_c = -y_0 + R \sin(\beta + \alpha_c) - y_r \quad (3)$$

$$y_c = x_0 - R \cos(\beta + \alpha_c) - x_r \quad (4)$$

where $R = \sqrt{x_0^2 + y_0^2}$ and $\beta = \arctan(\frac{y_0}{x_0})$.

2) *Vision based motion estimation*: Our long-term objective is to fuse the visual information at 200 Hz and the information provided by the angle-encoders at 30 Hz to achieve a high accuracy visual odometry. Currently, we focus on the visual information. The vision processing is as follows: the input images will be undistorted at first. Then, using SAD the optical flow is computed. The relationship between optical flow and the camera ego-motion is indicated by image Jacobian. To reduce the noise and optimize the results of the redundant equations, a Kalman filter is applied.

Compared with other optical flow computation algorithms, SAD performs more efficiently and less system resources are required. The size of our images is 640x480 pixels and the central 400x400 pixels are chosen as interest area. A searching window of 20x20 is defined so there are totally 400 windows in this interest area. SAD algorithm is used in every window with a block size of 8x8 pixels. The block with the least SAD values will be taken as the matched block. After SAD matching 400 sets of optical flow values have been acquired and a further elaboration is fulfilled as follows: The searching windows on the boundary of the interest area are abandoned and the remaining 18x18 windows can be separated into 36 groups. Each group consists of 3x3 windows as show in Fig. 15. In every group we set a threshold to eliminate some windows whose optical flow values seem not to be ideal enough. The average optical flow values of remaining windows in every group should be determined and could be seen as a valid optical flow value of this group. Every group can be considered as a single point and we just calculate the optical flow values of 36 feature points with a better accuracy. After calculating optical

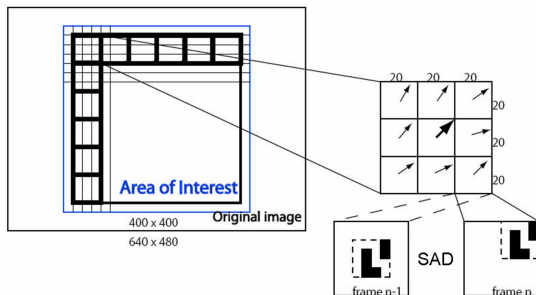


Fig. 15. Elaborated SAD algorithm

flow values with an elaborated SAD algorithm, we apply Kalman filter to determine the redundant equations based on image Jacobian matrix. The basic thought of Kalman filter is to predict the state vector x_k according to the measurement vector z . This vector comprises the 36 sets

of points velocities acquired from optical flow values of 36 feature points. The measurement matrix is a simplified image Jacobian matrix J .

The basic process of Kalman filter in our experiment is as follows:

$$\begin{aligned} x_k &= x_{k-1} + w_k \\ z_k &= Jx_k + v_k \end{aligned} \quad (5)$$

Random variables w_{k-1} and v_k represent the process noise and measurement noise respectively. The estimation process can be divided into two parts: predict part and correct part. At the beginning, the camera velocity vector, which is also the state vector in Kalman filter, is initialized with null vector, after the predict part, prior camera velocity estimation and prior error covariance estimation are transferred to the correct part. In correct part the posterior camera velocity estimation are computed by incorporating current point velocity vector, which is also the measurement vector. A posterior error covariance is also calculated in correct part and together with posterior camera velocity estimate transferred as initialization of the next step. In every step the posterior camera velocity estimation is the result of the redundant equations.

3) *Experimental results*: Utilizing a 1394b PCI-express adapter, the dragonfly® express camera (Point Grey Research Inc.), fitted with a normal wide-angle lens, is connected to our vision processing computer with an AMD Phenom 9500 @2.2GHz Quad-Core processor and 4 GB memory.

In the ACE platform there is an encoder which can estimate the current position of ACE. We read the data from the encoder at a frequency of 4-5Hz and consider them as ground truth. The camera mounted on ACE works at a frequency of 200Hz. Our experiment data is obtained when ACE is moving in the environment of stone sidewalk. The experiment is divided into two parts. In the first part, ACE ran about 6-7m in a straight line, which is taken as pure translation. The second part is pure rotation test. ACE only rotated at the starting point and passes about 450 grads. Fig. 16 left shows the results of estimating the robot displacements in pure translation. The red curve indicates the displacement in x-direction measured by encoder, and the blue curve indicates the displacement in x-direction estimated by visual odometry. The right part of Fig. 16 shows the angular result in pure rotation. The red curve indicates the ground truth from encoder, and the black curve indicates the estimation result from visual odometry.

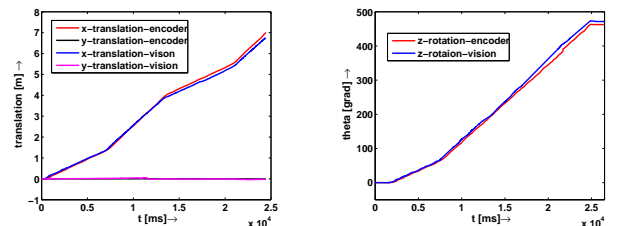


Fig. 16. Position estimation in pure translation (l) and in pure rotation (r)

V. SUMMARY AND FUTURE WORKS

A vision system meeting all the requirements of ACE, a robot which is able to navigate in an unknown urban environment was presented. Both, hardware setup and software architecture have been described. The vision system uses a novel multi focal camera head and is composed of several modules. Some of the modules serve interaction purpose, like the robust detection of humans to initialize an interaction or the estimation of human body poses. Other modules support the navigation of the robot, like the robust human tracking for crossing roads. Furthermore, the Visual odometry is used to enhance ordinary localization and the detection of sidewalks and crossroads plays an important role to increase the safety of the robot. The modules, the connections between the modules as well as the interplay with the navigation and interaction systems of ACE have been presented. Several experiments have supplied promising results. ACE had the mission to find its way from the stachus to the marientplatz, two public places in the city center of Munich which are connected with a crowded pedestrian zone. The distance between the two places is around 700 meters. As ACE had no GPS or map knowledge, it had to ask pedestrians for the way and rely on the given information. ACE was able to fulfill the mission and has reached the marientplatz after 90 minutes.

Most of the modules of the vision system have already been implemented, but the experiments have shown the need for a module which can compute a confidence for the body pose estimation. Due to bad light condition when the camera is oriented towards the sun, some of the estimated body poses are invalid. Those poses have to be detected and discarded. Needless to say, it is crucial for the interaction module to be informed about the confidence of the estimated pose. As some pedestrians evaluated the interaction by pointing in directions as rather technical and to achieve a better and more fluent interaction with pedestrians, the interaction has to be enhanced. For this, the computational speed of the body pose algorithm has to be increased and the connection to the interaction module has to be tuned. Furthermore, the performance and robustness of the human tracking have to be increased. To achieve a more fluent and natural behavior of the robot, the connection to the navigation module has to be enhanced as well. Another upcoming challenge is the integration of the semantic navigation.

VI. ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also www.cotesys.org.

REFERENCES

- [1] F. Schubert, T. Spexard, M. Hanheide and S. Wachsmuth. *Active Vision-based Localization For Robots In A Home-Tour Scenario*. Proceedings of the 5th International Conference on Computer Vision Systems, 2007.
- [2] A. M. Arsenio. *Map Building from Human Computer Interaction*. IEEE CVPR Workshop on Real-Time Vision for Human Computer Interaction, 2004.
- [3] Nourbakhsh, I.R.; Kunz, C.; Willeke, T. *The mobot museum robot installations: a five year experiment*, Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Volume 4, Issue , 27-31 Oct. 2003 Page(s): 3636 - 3641.
- [4] Schraft, R.D.; Graf, B.; Traub, A.; John, D.; *A Mobile Robot Platform for Assistance and Entertainment*. In Industrial Robot Journal, Vol. 28, 2001, pp. 83-94.
- [5] M. Shiomi, T. Kanda, H. Ishiguro and N. Hagita. *Interactive humanoid robots for a science museum*, Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. pp. 305-312, 2006.
- [6] B. Leibe, A. Leonardis and B. Schiele. *Robust object detection with interleaved categorization and segmentation*. IJCV Special Issue on Learning for Vision and Vision for Learning, Sept. 2005. revised version Nov. 2006.
- [7] K. Klaasing, G. Lidoris, A. Bauer, F. Rohrmüller, D. Wollherr, M. Buss. *The Autonomous City Explorer Project: Towards Semantic Navigation in Urban Environments*. Submitted to CoTeSys Workshop-2008.
- [8] S. K. Singh, D. S. Chauhan, M. Vasta, and R. Singh. *A robust skin color based face detection algorithm*. Tamkang Journal of Science and Engineering, 6(4):227-234, 2003.
- [9] V. Vezhnevets, V. Sazonov and A. Andreeva. *A Survey on Pixel-based Skin Color Detection Techniques*. In Proceedings Graphicon-2003.
- [10] S. Pellegrini and L. Iocchi. *Human Posture Tracking and Classification through Stereo Vision and 3D Model Matching*. In Journal on Video and Image Processing-2007.
- [11] S. Knoop, S. Vacek, R. Dillmann. *Sensor fusion for 3D human body tracking with an articulated 3D body model*. In Proceedings of the IEEE International
- [12] H. Yang and S. Lee. *Reconstructing 3D Human Body Pose from Stereo Image Sequences Using Hierarchical Human Body Model Learning*. In Proceedings of the 18th International Conference on Pattern Recognition-2006.
- [13] Campbell, J.; Sukthankar, R.; Nourbakhsh, I. and Pahwa, A. *A robust visual odometry and precipice detection system using consumer-grade monocular vision*, Proceedings of the 2005 IEEE International Conference on robotics and automation, pp.3421 - 3427, ISBN: 0-7803-8915-8 , Barcelona, April 2005.
- [14] Wang, H.; Yuan, K.; Zou, W. and Zhou, Q. *Visual odometry based on locally planar ground assumption*, Proceeding of the 2005 IEEE international conference on information acquisition, Hong Kong and Macau, China, June 2005
- [15] Nister, D., Naroditsky, O., Bergen, J. *Visual odometry*, Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, Vol.1, pp. 652-659, ISBN: 0-7695-2158-4, Washington DC, July 2004
- [16] Fernandez, D. and Price, A. *Visual odometry for an outdoor mobile robot*, Proceeding of the 2004 IEEE conference on robotics and mechatronics, Singapore, December 2004.
- [17] Dornhege, C. and Kleiner, A. *Visual odometry for tracked vehicles*, Proceeding of the IEEE international workshop on safety, security and rescue robotics, Gaithersburg, USA, 2006.
- [18] K. Kühnlenz, M. Bachmayer and M. Buss, *A Multi-Focal High-Performance Vision System*. In the Proceedings of the International Conference of Robotics and Automation (ICRA), pp. 150-155, Orlando, USA, May 2006.
- [19] K. Kühnlenz, *Aspects of Multi-Focal Vision*. PhD Thesis, Institute of Automatic Control Engineering, Technische Universität München, Munich, Germany, 2006.
- [20] N. Courty and E. Marchand. *Visual perception based on salient features*. In Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robotics and Systems, 2003.
- [21] R. Lienhart and J. Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection*. In Proceedings of the International Conference on Image Processing-2002.
- [22] Q. Mühlbauer, K. Kühnlenz and M. Buss. *A Model-based Algorithm to Estimate Body Poses using Stereo Vision*. In Proceedings of the 17th International Symposium on Robot and Human Interactive Communication-2008.
- [23] U. Iwan and N. Illah. *Appearance-Based Obstacle Detection with Monocular Color Vision*. In Proceedings of the AAAI National Conference on Artificial Intelligence-2000.