Heat Mapping for Improved Victim Detection

Ruwen Hahn, Dagmar Lang, Marcel Häselich, Dietrich Paulus
Active Vision Group, AGAS Robotics
University of Koblenz-Landau
Universitätsstr. 1, 56070 Koblenz, Germany
{rhahn,dagmarlang,mhaeselich,paulus}@uni-koblenz.de

Abstract — Disasters, such as earthquakes or tsunamis, result in destroyed buildings and other dangerous scenarios for human rescuers. Remotely controlled robots are often used to aid humans. Since those robots require steady communication, they are limited to a certain range and might get stuck in case the connection is interrupted. To augment remote controlled robots, autonomous robots are needed. These robots are able to navigate in devastated areas to detect victims while creating a map of the environment at the same time. Victim detection based on thermal sensors is the most widely used approach. In this paper we present a novel approach based on low-cost thermal sensors, using the global heat distribution.Therefore we developed a 2D heat map, which is created by the combination of thermal and laser information during continuous autonomous exploration. The map is build from the history of all sensor readings over time resulting in a heat distribution.

The main contribution of this paper is the introduction of a 2D heat map accumulating thermal sensor readings over time for improved victim detection.

Keywords: *mapping, victim detection, victim verification*

I. INTRODUCTION

Victim detection and verification on raw thermal images is used in conjunction with image processing algorithms. For victim search only special points of view are considered, resulting in blind spots, which in turn may lead to missed victims. According to this problems, we present a mapping solution to reduce or eliminate blind spots and to improve mission run-time.

To evaluate the abilities of different robots the National Institute of Standards and Technology (NIST) develops methods and benchmarks. In context of the RoboCupRescue league robots are tested in and compete against each other in different scenarios. The goal of the league is to advance manipulation, autonomous robot operation and mobility in difficult terrain.

In context of the search and rescue scenario of RoboCupRescue different victim detection and verification algorithms are developed. Most of them rely on connected component search on thermal images (e.g. [1]). Additionally different high-level algorithm are developed for this scenario:

Bahadori et al. [2] investigate different computer vision methods and their adaptability for RoboCupRescue. A sensor fusion approach of camera, microphone, pyroelectric sensor and infrared camera for motion detection, sound and heat detection is presented by Noubarksh et al. [3]. Kleiner et al. [4] present an approach of fusing thermal and camera images with Markov random fields and the classification of faces, heat, color and motion in the context of RoboCupRescue. Additionally, Meyer et al. [1] develop a semantic world model based on an Extended Kalman Filter. For victim detection a combination of visual cameras and thermal images is used.

This paper is organized as follows: Section II describes the search and rescue scenario of the RoboCupRescue competition. Our victim detection is depicted in Section IV and the new heat mapping algorithm is presented in detail in Section V. Section VI contains the results of our experiments followed by the conclusion in Section VII.

II. ROBOCUPRESCUE SCENARIO

The RoboCupRescue scenario provides a versatile benchmark for search and rescue robots in a competitive environment that simulates a disaster area. Several tasks define the challenges for the robot, such as the detection and localization of simulated victims, the performance of map generation, manipulation capabilities and the mobility of each robot.

A RoboCupRescue arena consists of multiple color-coded arenas, each with a different set of constraints to facilitate the benchmarks. In this paper we will focus only on the yellow arena, as it is the part for the autonomous robots. For a full discussion of all arena types refer to [5].

The yellow arena consists of corridors that are 1.2 m wide, possibly a few open areas and $15°$ ramps with 0.6 m and 1.2 m depth for a maximum elevation of about 0.38 m. Simulated victims are placed throughout the arena on heights between 0 m and 1.2 m above the ground, mostly in holes inside the walls or in another challenging position. A simulated victim consists of a heating pad for thermal identification, a digital audio player to simulate communication with the victim and a doll to represent it. Additionally, eye charts and hazmat labels are spread in the area to evaluate image quality and object recognition capabilities of each robot.

Each competing robot is challenged in multiple missions of up to 20 minutes to find as many victims as possible with as many sensor readings as possible while it constantly needs to map the environment. The position of each found victim has to be added to this map, as it is the map that indicates a human rescuer where to find the victims in a real disaster.

III. ROBBIE

Robbie is a research robot developed at the University of Koblenz-Landau (see Fig. 1). It is based on a Pioneer 3-AT platform with a custom top-mounted sensor framework.

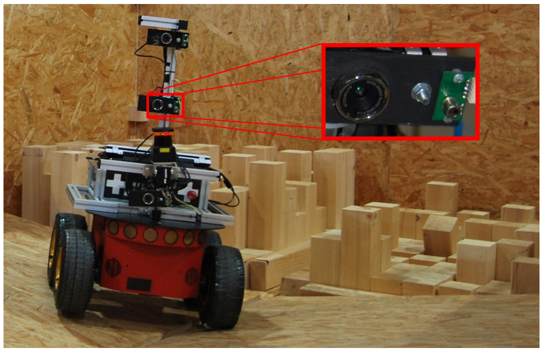

Fig. 1. Sensor setup of the rescue robot *Robbie*. One of the low-cost heat sensor and a camera on the panning device are zoomed in for better visualization (red rectangle).

Attached to the sensor framework are one IDS uEye camera and three panning Philips SPC1330NC web-cams. For distance measuring we use a Hokuyo UTM-30LX laser range finder (LRF) and for heat detection three TPA81 thermopile arrays are mounted next to the panning cameras. Additional sensors include odometry readings on both axis, sonar sensors on either front and back of the Pioneer platform and an Xsens MTi inertial measurement unit. Both the Hokuyo LRF and the thermopile arrays are mounted on servomotors to allow for larger viewing angles on both sensor types. In the case of the LRF additional servomotors adjust the orientation of the LRF to horizontal aligned readings for 2D mapping [6]. Refer to [7] for further details.

IV. Previous Victim Detection

A single TPA81 thermopile array provides a row of 8 pixels of thermal data with an effective field of view of $41°$ by $6°$. Due to the relatively large field of view for each individual pixel the effective range of the thermopile arrays is limited to less than 2 m. Up to now we use the thermopile arrays to create a thermal image by panning the thermopiles at a fixed speed and linearly interpolating the recorded readings to create a thermal image, i.e. the thermopile arrays are panned over a $120°$ angle, each creating a partial thermal image of the area in front of the robot. The partial images are then stitched together into the resulting thermal image, which may therefore include duplicated information where the fields of view of the thermopile arrays overlap vertically for distant heat sources. On this image a connected component analysis is performed to find groups of warm pixels to accommodate for erroneous readings and non-victim heat sources. For each detected connected component the distance to the center of the image is then computed to find the victim's relative angle to the current robot pose. Using this angle we select a likely position of the victim on our 2D map.

For this algorithm to work the robot has to stand still for the duration of the thermal scan, otherwise the accuracy of the computed angles might suffer or the thermal images may have an uneven heat distribution due to differing distances to the heat sources.

From the robot's current position and the estimated victim position an observation position is computed that should allow the victim to be visible on at least one of the cameras. After the robot has successfully navigated to the observation position a second thermal scan is initiated to verify the victims position in front of the robot. In case of a successful second thermal scan the operator is notified and the autonomous operation is suspended.

V. 2D Heat Mapping

In this section we motivate and describe a major reinterpretation of the thermal data provided by the TPA81 thermopile arrays as a map of heat distribution in a rescue environment. Our approach is based upon established techniques for mapping i.e. laser range data in an occupancy grid map.

Like the laser range data, heat can be mapped while the robot is moving and thereby reduce or in some cases even eliminate navigation pauses. As a result the time in which the robot stands still and stops exploring is significantly reduced. As a secondary goal, the created map should provide an easy way for humans to identify heat sources and their proportions in relation to other mapped objects.

Our robotics software framework provides navigation and exploration based on occupancy grid maps and mapping using laser range finder data. For easier integration with the autonomous system, the heat map, too, is represented as a grid map. This enables us to achieve a consistent world view between existing maps within the system and the heat map, which allows for improved victim detection and verification. In the following subsections we give a high-level overview of the laser mapping, and navigation and exploration before introducing heat mapping and discussing victim detection and verification in detail.

A. Laser Mapping

We use a particle filter to solve the simultaneous localization and mapping problem (SLAM problem) (see [8], [9]). Each particle represents a hypothesis for the pose of the robot in 2D-space: $(x, y, \Theta)^T$. The map is shared among all particles and represented as an occupancy grid map.

The algorithm of the particle filter includes the following steps: *resample*, *drift*, *measure*, and *normalize*. The result is the most likely pose of the robot when the laser scan was taken. This pose is then used for the *map update*.

Resample: Depending on the weight of the particles (generated by the last measurement step), the probability distribution is adjusted. The resampling step is only done if the robot moved a certain distance (at least 20 mm) or turned (at least $5°$) since the last mapping step.

Drift: During the drift step, the particles are moved depending on the odometry data. It also models noise due to sensor inaccuracies and wheel slippage.

Measure: During the measurement step, new weights for each particle are calculated using the current laser scan. To weight a particular particle, the endpoints of the laser beams are calculated, using the robot pose stored in this particle.

We use a subsampling of the laser data: Only laser beam endpoints, that are at least 10 cm away from the last considered endpoint are used for the weighting function. This speeds up the measurement function significantly.

Normalize: The assigned weight of a particle is a sum of occupancy probability values. During the normalization step, the weight of each particle is divided by the sum of all particle weights.

Map update: The average pose of the top 5% particles with the highest weight is assumed to be the location of the robot. Using this pose, the current laser range scan is added to the global occupancy grid map by constructing a local map and "stamping" it into the global map, incrementing the counts for *seen* and *occupied* cells.

B. Navigation and Exploration

By combining Yamauchi's frontier based exploration [10] with Zelinsky's path transform (see [11], [12]) an elegant solution for the exploration problem can be achievedby the exploration transform [13]. Here the path transform is extended in a way that not the cost of a path to a certain target cell is calculated, but the cost of a path that goes to a close frontier (the boundary between known and unknown areas). The path is not necessarily the shortest and the frontier not necessarily the closest, since the cost is determined by the travel distance *and* the safety of the path. The overall formula of the exploration transform [13] is given in (1); it yields for a given cell c the frontier cell that is close and save to reach:

$$\Psi(c) = \min_{c_{\mathrm{g}} \in F} \left(\min_{C \in \chi_c^{c_{\mathrm{g}}}} \left(l(C) + \alpha \sum_{c_i \in C} c_{\mathrm{danger}}(c_i) \right) \right) \quad (1)$$

with F the set of all frontier cells, $\chi_c^{c_{\mathrm{g}}}$ the set of all paths from c to c_{g}, $l(C)$ the length of the path C, $c_{\mathrm{danger}}(c_i)$ the cost function for the "discomfort" of entering cell c_i, and α a weighting factor ≥ 0. The exploration transform has the favorable property that by construction no local minima can occur. Therefore, from each cell a path to a close frontier can directly be extracted. Compared to the path transform, the exploration transform performs a search over all possible frontier cells.

C. Heat Mapping

To provide a meaningful and easy to understand map for human observers and processing programs we represent a heat map as a 2D grid of heat pixels $G(x, y) = (T_\Sigma, w)$ with T_Σ as the sum of weighted heat readings and w the weight factor at the position (x, y) in the grid G.

The grid is then populated with readings taken from all 3 thermopile arrays with a frequency of about 50 Hz. To increase the observed area, the thermopile arrays are continuously panning at a speed of about $100\,\frac{\mathrm{deg}}{\mathrm{s}}$ between $-60°$ and $60°$ where a panning angle of $0°$ is directly in front of the robot.

As described in Section IV, our previous victim detection focused on finding warm areas in a thermal image while essentially ignoring "cold" pixels. To model a similar behaviour for heat maps, T_r is selected as the warmest of all vertically aligned thermopile array pixels.

Each reading of all three thermopile arrays triggers a heat map update, which involves updating all pixels in the field of view of the selected thermopile array with the current temperature reading T_r weighted by the distance to the (averaged) room temperature T_e for heat map pixel $G(x, y)$

$$\begin{aligned} \Delta T &= T_r - T_e & (2) \\ T'_\Sigma &= T_\Sigma + T_r \Delta T & (3) \\ w' &= w + \Delta T & (4) \\ G'(x, y) &= (T'_\Sigma, w') & (5) \end{aligned}$$

For the RoboCupRescue League a few assumptions about heat distribution and the general environment are made:

- Heat sources can not de sensed through walls.
- Victims are always "inside" or at least near walls.
- Victims may be placed anywhere between ground level and a maximum height of 1.2 m.

During the heat map update heat is stopped from being "projected" through walls by clipping the field of views to the laser map, so that pixels behind walls and other obstacles are not updated with probably unrelated heat readings.

Since laser map updates typically take about 300 ms only readings taken before the most recent laser map update are committed to the heat map to ensure accurate view port clipping at all times. Readings taken between laser map updates are visualized without clipping but using these preliminary estimates for further uses such as victim detection is discouraged since false positives imply penalties in the RoboCupRescue League.

The height of walls is quite often a problem for robots competing in the league since spectators and other heat sources outside of the arena may be detected as victims. In a real scenario this would not be a problem since autonomous robots should report potential victims on all possible heights and not be artificially limited to 1.2 m.

To deal with this limitation with the sensors at our disposal we elevate the height-offset of our LRF readings by a certain amount, depending on the current roll and tilt angles, and project the resulting world points onto the image planes of our thermopile arrays. Only pixels that are below the lowest projected world point are then taken into account for further processing.

D. Victim Detection

Victim detection on heat maps can be split into offline detection and online detection, where the former consists of finding warm spots in a complete map, i.e. warm objects are identifiable as a continuous cluster of warm pixels on the map with the warmest pixels in the center of the warmest region of the object and probably a gradient of less warm pixels near them. The hot spots of these clusters are then detected as potential victims or, depending on map quality and other sensor data, verified as real victims.

As the name suggests, offline detection cannot be used in time-critical environments like RoboCupRescue or while detecting victims in a disaster area since it requires the pixels in warm areas to have as many updates (from differing angles) as possible for accurate heat distribution. While it is possible to have the robot navigate to get a good coverage of these warm areas it introduces a significant overhead of time spent per warm area, not to mention the hazards to both robot and victim of repeatedly navigating possibly dangerous terrain.

Therefore we implemented various schemes to detect victims on data that would be gathered on a normal exploration mission as our robot has proven to map almost any arena with good coverage while operating in this mode.

We further classify those schemes as either forward or backward victim detection, depending on the latency between first observation of a potential victim position and the decision to mark a position as a potential victim position for final victim verification. An algorithm is considered a backward victim detection, if the latency until the final decision is high enough that a robot would have to backtrack to a previous pose for optimal observation of the selected victim position. The following definitions each provide a set of potential victim positions V_p.

1) Greedy victim detection: Our first strategy on online victim detection is to mark all high temperature wall pixels as potential victim coordinates for a threshold temperature T:

$$V_p = \left\{ (x,y) | G(x,y) = (T_\Sigma, w), \frac{T_\Sigma}{w} \geq T \right\} \quad (6)$$

This requires registration of positions with actual victims and the computation of a final victim location to be done as part of the victim verification. Also, this definition does not provide a stable set of victims in case a heat pixel is cooled down in subsequent map updates.

Therefore we suggest to only use this detection to provide a visualization of high temperature wall pixels for humans. Other possible uses, outside of victim detection, would be to track changing heat distribution over a period of time, e.g. to track human movement.

2) Forward victim detection: Another approach that focuses on low latency while providing only one coordinate for a potential victim is to mark the warmest wall pixel within the current scan area that was at least updated twice while still retaining a temperature above a certain threshold. To ensure that only one coordinate is generated per victim we discard all future positions that are within the general area of previously selected coordinates, described by minimal distance along the 2D map axes Δd.

For our selected pixel (x, y) and the set of previously selected pixels V_p:

$$\begin{aligned} \delta(x', y') &= |x - x'| > d \wedge |y - y'| > d \quad (7) \\ V'_p &= V_p \cup \{(x,y) | \forall (x', y') \in V_p : \delta(x', y')\} \quad (8) \end{aligned}$$

Temperature Range	Color Gradient
0° C - 20° C	Dark Blue - Dark Green
20° C - 30° C	Dark Green - Yellow
30° C - 38° C	Yellow - Orange
38° C - 45° C	Orange - Red

TABLE I

COLOR OF HEAT MAP PIXEL IN RELATION TO TEMPERATURE.

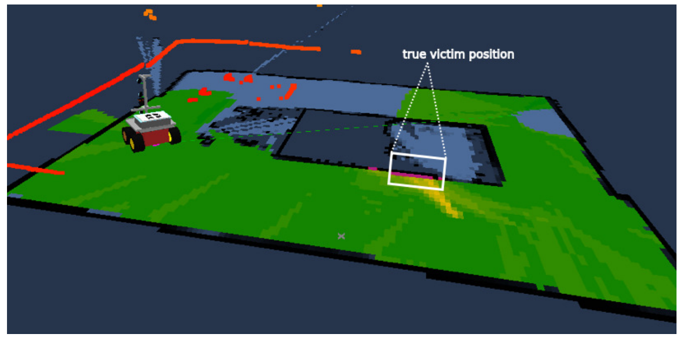

Fig. 2. Evaluation of the greedy victim detection in a NIST arena. Walls and other obstacles are displayed as black pixels, dark and light blue areas are unknown and drivable, respectively. The ground texture is overlayed with the heat map where green areas signify temperatures around room temperature and yellow-red spots show higher temperatures (see Table I). Detected victim pixels are the purple colored wall pixels above the yellow spot to the right.

3) Backward victim detection: Finally we introduce a highly accurate victim detection that iteratively selects the best victim position over a range of heat map updates. The iteration is started as soon as a wall pixel with more than one update and a temperature above a certain threshold is detected. On subsequent updates the highest temperature wall pixel is selected and the distance along the 2D-axes is computed if the pixel is warmer than the previously selected pixel and within a short distance (usually less than 30 cm) of the previous pixel the new pixel selected as best candidate. If the same pixel is selected at least three times or the newer candidate pixels are less warm than the selected pixel (for example when the robot's field of view moves away from the victim) the pixel is added to V_p.

As with the forward detection algorithm, once a potential victim position is selected future victim positions within its general area are discarded.

We choose to extend the minimal latency approach to iteratively select better victim positions on consecutive scans as long as the wall pixels hit by those scans where within 15 cm of the last estimated position. We observed that the victim position improved considerably in most cases, but unfortunately by the time the victim position became stable enough for final verification the robot had moved away from the position where the most likely victim pixels was detected and thereby required to navigate back to the observation position. Fig. 5 shows the resulting victim positions and corresponding robot positions

While the backward victim detection approach tends to provide the most accurate victim positions the added cost of moving back to the observation position clearly does not

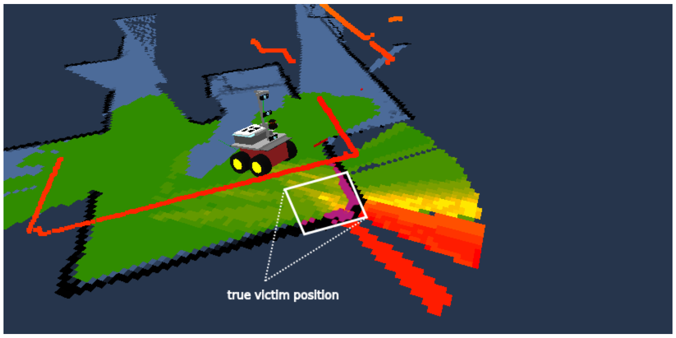

Fig. 3. In this picture the victim was placed right in the corner in front of the robot at the lower end of a pair of half length ramps with their apex running from the left to right in parallel to the longer bottom wall segment. A large number of victim pixels can be seen due to the victim's placement.

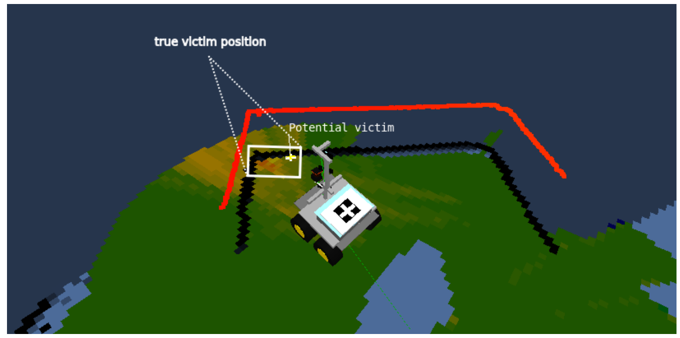

Fig. 4. This image shows the same scenario as Fig. 3 albeit this time with the forward victim detection. The result of the detection is labeled with "Potential victim".

satisfy the goal of minimizing required movement. Also, the extra latency that is introduced by waiting for nearby updates should be avoidable.

We therefore decided to use the less accurate forward victim detection and combine the initial estimated victim position with the thermal image based verification while the robot stops moving. Thereby we gain the additional accuracy that the thermal images have always provided while still retaining a significant speed boost for areas that do not contain heat sources. In tests we observed that *Robbie* did not move backwards on a path already taken while using this hybrid approach, unlike the higher latency solutions that required navigating to a previous location for optimal victim visibility. We therefore call this approach a forward victim detection while the higher latency solutions can be classified as backward victim detection.

E. Victim Verification

Victims are verified using the approach of capturing a thermal image while the robot is not moving (Section IV) to allow last-minute corrections in case of accuracy losses for forward detection algorithm or inaccurate observation positions due to navigation inaccuracies while backtracking.

Additionally, we gain the ability to detect multiple victims that are on differing heights of the same map pixel and a fallback solution to provide judges with a more familiar representation of thermal sensor data.

VI. Experiments and Results

In this section we provide details on the evaluation of the differing detection algorithms in relation to each other. We also compare the new developed algorithms to the previous victim detection described in Section IV. The latter was accomplished by recreating a scenario from sensor recordings of the RoboCup GermanOpen 2011 in Magdeburg.

A. Greedy victim detection

Multiple test runs with *Robbie* were made to evaluate the general feasibility of victim detection using heat maps. For the first few tests we used a simple arena that consisted of a single loop (see Fig. 2). Walls, or rather all obstacles within the scanning plane of the LRF are marked as black pixels. Around the obstacle pixels are darker and lighter blue areas denoting unknown terrain and obstacle free terrain as determined by our SLAM algorithm. To the left is a (simplified) representation of *Robbie* with the current LRF measurements displayed in orange and red.

In this picture the heat map overlay for our laser map is enabled, whereby most of the visible free ground is colored according to observed temperature (see Table I). Wall pixels that were detected as potential victim positions are colored purple. The bottom corridor of the arena consisted of 5 full length ramps with their apex to the bottom of the image.

In this scenario the detected victim position is very localized for this kind of algorithm, however in the next arena we will see a scenario that provides rather too many victim pixels.

In Fig. 3 the victim was placed at the lower end of a pair of half length ramps with their apex running from the left to the right in parallel to the longer bottom wall segment. The most interesting feature of this image is the high concentration of temporary high temperature color which was the result of almost touch contact with the victim heating pad.

The next feature is the high density of purple victim pixels that are arrayed around the victim's true position. Clearly this is an unnecessary big sample size, but it demonstrates the feasibility of the general approach since all marked pixels are still within 0.1 m of the victim.

B. Forward victim detection

We retried the same scenario that created problems for the greedy victim detection with the forward detection algorithm as can be seen in Fig. 4. Since we are now creating only small numbers of potential victim positions we changed the visualization of these positions to a standard point of interest visualization within our software, i.e. a white label ("Potential victim") that is anchored to the ground with a white cross.

It is obvious that the chosen victim position is lacking in accuracy, however we still need to verify the victim, which provides the last few centimeters of accuracy in this scenario.

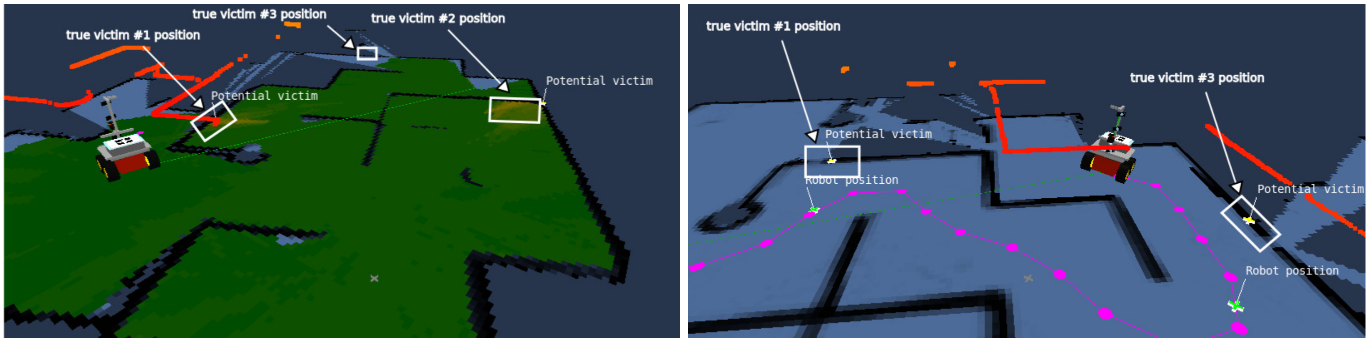

Fig. 5. Recreation of two scenarios from the RoboCup GermanOpen 2011. In the left picture victim #1 is placed at about 0.3 m height within a corridor of pairs of 0.6 m ramps in parallel to the wall the victim is attached to. Victim #2 is placed on the ground. Both victims are detected on the right position. The right picture shows the same arena from a different perspective. The purple connected dots show the path the robot has taken to reach its current position. The points of interest labeled with "Robot position" at the time the potential victims were detected.

C. Backward victim detection

In Fig. 5 we recreated two scenarios from the RoboCup GermanOpen 2011. Especially victim #1 was undetectable using our old victim detection due to bad choice of scanning position and general navigation problems. The whole center area consists of pairs of 0.6 m ramps with their apex parallel to the wall the victim #1 is attached to at about 0.3 m height.

The right picture shows *Robbie*'s path while detecting the victims and its positions at the time the victims were detected instead of the heat map overlay. To balance the success with the victim on the left now the victim on the right was not detected sometimes due to the hole next to the victim that is at the exact height of our LRF measurements. In certain positions laser points from behind the wall hiding the victim are used to determine the thermopile readings that need to be discarded which results in cutting off a large region from the upper part of that wall.

Another aspect that all pictures have in common is that the heat map coverage of the laser map is very good, even though no modifications were made to the navigation.

VII. Conclusion

In this paper we presented heat mapping as a way to improve victim detection in a RoboCupRescue scenario with a low cost thermal imaging solution. We will use heat mapping for the RoboCup World Championship in Istanbul as we expect the arena to be even more difficult than the GermanOpen 2011 arena in Magdeburg was.

With the experiments we showed that for most victim positions heat mapping was on par or better than our previous approach. We demonstrated that an reduction of mission time by continuously mapping with a thermal sensor is possible. Or approach is limited to find victims located in front of wall or within the arena bounds. In one case, as visualized in Fig. 5, we sometimes miss victims due to this fact. Another problem of the increased complexity and frequency for handling the thermal data is that *Robbie*'s notebook was working at full capacity even before our new implementation.

We look forward to test our algorithms with higher specification thermal sensors and other complementary hardware in the future to provide even more accurate maps of any variety. Furthermore the restriction to victims in front of walls needs to be eliminated, e.g. by using a thermal sensor with a larger visual range or by interpreting the data in 3D. Another extension could be the detection and following of dynamic heat sources like moving humans.

Acknowledgment

This work was partially supported by Bundeswehr Technical Center for Engineer and General Field Equipment (WTD 51), Koblenz, Germany.

References

[1] J. Meyer et al., *A Semantic World Model for Urban Search and Rescue Based on Heterogeneous Sensors*, Proceedings of the RoboCup Symposium, 2010

[2] S. Bahadori and L. Iocchi, *Human Body Detection in the RoboCup Rescue Scenario*, Proceedings of the RoboCup Symposium, 2003

[3] I. Nourbakhsh et al., *Human-Robot Teaming for Search and Rescue*, Pervasive Computing: Mobile and Ubiquitous Systems, pp. 72-78, 2005

[4] A. Kleiner and R. Kümmerle, *Genetic MRF Model Optimization for Real-Time Victim Detection in Search and Rescue*, Proceedings of the International Conference on Intelligent Robots and Systems, pp. 3025-3030, 2007

[5] RoboCup Wiki, Robot League, http://wiki.robocup.org/wiki/Robot_League

[6] J. Pellenz, *Rescue robot sensor design: An active sensing approach*, Proceedings of the International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster, pp. 33-37, 2007

[7] D. Lang et al., *RoboCupRescue 2011 - Robot League Team resko@UniKoblenz (Germany)*, Technical Report, University of Koblenz-Landau, 2011

[8] M. Isard and A. Blake, *Condensation - conditional density propagation for visual tracking*, International Journal for Computer Vision, Vol. 29 (1), pp. 5-28, 1998

[9] J. Pellenz, *Mapping and Map Scoring at the RoboCupRescue Competition*, Quantitative Performance Evaluation of Navigation Solutions for Mobile Robots, Robotics: Science and Systems, 2008

[10] B. Yamauchi, *A frontier-based approach for autonomous exploration*, Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation, pp. 146 ff., 1997

[11] A. Zelinsky, *Robot Navigation With Learning*, Australian Computer Journal, Vol. 20(2), pp. 85-93, 1988

[12] A. Zelinsky, *Environment Exploration and Path Planning Algorithms for a Mobile Robot using Sonar*, PhD Thesis, Wollongong University, Australia, 1991

[13] S. Wirth and J. Pellenz, *Exploration Transform: A Stable Exploring Algorithm for Robots in Rescue Environments*, Workshop on Safety, Security, and Rescue Robotics, pp. 1-5 , 2007