Chapter 4: Human Victim Detection and Stereo-based Terrain Traversability Analysis for Behavior-Based Robot Navigation

Geert De Cubber, Daniela Doroftei

4.1 Introduction

Crisis management teams (e.g. fire and rescue services, anti-terrorist units ...) are often confronted with dramatic situations where critical decisions have to be made within hard time constraints. In these circumstances, a complete overview of the crisis site is necessary to take correct decisions. However, obtaining such a complete overview of a complex site is not possible in real-life situations when the crisis management teams are confronted with large and complex unknown incident sites. In these situations, the crisis management teams typically concentrate their effort on a primary incident location (e.g. a building on fire, a crashed airplane ...) and only after some time (depending on the manpower and the severity of the incident), they turn their attention towards the larger surroundings, e.g. searching for victims scattered around the incident site. A mobile robotic agent could aid in these circumstances, gaining valuable time by monitoring the area around the primary incident site while the crisis management teams perform their work. However, as the human crisis management teams are in general already overloaded with work and information in any medium or large scale crisis situation, it is essential that such a robotic agent - to be useful - does not require extensive human control (hence it should be semi-autonomous) and it should only report critical information back to the crisis management control center.

In the framework of the View-Finder project, such an outdoor mobile robotic platform was developed. This semi-autonomous agent, shown on Figure 4.1, is equipped with a differential GPS system for accurate geo-registered positioning, and a stereo vision system. The design requirements for such a robotic crisis management system give rise to three main problems which need to be solved for the successful development and deployment of such a mobile robot:

1. How can the robot automatically detect human victims, even in difficult outdoor illumination conditions?
2. How can the robot, which needs to navigate autonomously in a totally unstructured environment, auto-determine the suitability of the surrounding terrain for traversal?
3. How can all robot capabilities be combined in a comprehensive and modular framework, such that the robot can handle a high-level task (searching for human victims) with minimal input from human operators, by navigating in a complex, dynamic and environment, while avoiding potentially hazardous obstacles?

In response to the first question, we present an approach to achieve robust victim detection in difficult outdoor conditions, by going out from the Viola-Jones algorithm for Haar-features based template recognition and adapting it to recognize victims. Victims are assumed to be human body shapes lying on the ground. The algorithm tries to classify visual camera input images into human body shapes and background items. This approach is further explained in section 4.3.

The second problem which is stated above is that of the traversability estimation. This is a challenging problem, as the traversability is a complex function of both the terrain characteristics, such as slopes, vegetation, rocks, etc and the robot mobility characteristics, i.e. locomotion method, wheel properties, etc. In section 4.4 of this chapter, we present an approach where a classification of the terrain in the classes “traversable” and “obstacle” is performed using only stereo vision as input data.

The third question which was raised above relates to the robot control mechanism and the control architecture. For this application, the behavior based control paradigm was chosen as a control mechanism due to its flexible nature, allowing the design of complex robot behavior through the integration of multiple relatively simple sensor-actuator relations. Through this control architecture, the robot is able to search for human victims on an incident site, while navigating semi-autonomously, using stereo vision as a main source of sensor information. The behavior based control mechanism is further detailed in section 4.5 of this chapter, while section 4.2 gives an overview of the global robot control architecture.

4.2 Robot control architecture

The control architecture describes the strategy to combine the three main capabilities of an intelligent mobile agent: sensing, reasoning and actuation. These three capabilities have to be integrated in a coherent framework in order for the mobile agent to perform a certain task adequately. To combine the advantages of purely reactive and planner-based approaches, a behavior-based controller was implemented for autonomous navigation.

Figure 4.2 illustrates the general robot control architecture, set up as a test bed for the algorithms discussed in this chapter. The RobuDem robot used in this setup features 2 on-board processing stations, one for low-level motor control (Syndex Robot Controller), and another one for all the high-level functions. A remote robot control PC is used to control the robot and to visualize the robot measurements (color images, victim data) from a safe distance. All data transfer between modules occurs via TCP and UDP-based connections, relying on the CORBA (Slama, 1999) and CoRoBa (Colon, 2006) protocols. The wireless link from the on-board high-level PC to the remote robot control PC is set up via a wi-fi connection.

A behavior-based navigational architecture is used for semi-autonomous intelligent robot control. Behavior-based techniques have gained a widely popularity in the robotics community (Jones, 2004), due to the flexible and modular nature of behavior-based controllers, facilitating the design process. Following the behavior based formalism, a complex control task is subdivided into a number of more simple modules, called behaviors, which each describe one aspect of the sensing, reasoning and actuation robot control chain. Each behavior outputs an objective function, $o_1(x), \ldots, o_n(x)$. These objective functions are multi-dimensional normalized functions of the output parameters, where $x = x_1, \ldots, x_n \in \mathbb{R}^n$ is an *n*-dimensional decision variable vector. The degree of attainment of a particular alternative *x*, with respect to the k^{th} objective is given by $o_k(x)$.

Recall that the RobuDem robot is equipped with two main sensing abilities: a stereo vision system and a GPS system. The information from the stereo vision system is used threefold:

1. The color images are sent over the wireless link, such that the human operator receives at all time a visual cue of the environment. This is absolutely necessary when the robot is operating under tele-operation mode.
2. The (left) color image is sent to a victim detection module. This module incorporates a vision-based human victim detector, presented in section 4.3. The victim detection module will report any detected human victims back to the human operator at the remote control station.

3. The calculated stereo disparity image is sent to a terrain traversability estimation module. This module incorporates a stereo-based terrain traversability analysis algorithm, as presented in section 4.4. From the obstacle map, a behavior is constructed to steer the robot away from obstacles.

The GPS system delivers accurate robot positioning information, which is sent to the operator at the remote control station. At the same time, this data is sent to a path planning module. From the robot control station, the human operator is able to compile a list of waypoints for the robot. The path planning module compares this list of waypoints with the robot position and calculates a trajectory to steer the robot to the goal positions in the list. The first point on this trajectory list is sent to a *GoToGoal* behavior module, which aims to steer the robot to this point, as such executing the trajectory defined by the path planner. The specific design and integration of all different behaviors is discussed in section 1.5 of this chapter.

4.3 Human victim detection

4.3.1 Problem statement and state of the art of human victim detection

Automated human victim detection is a very difficult task, especially in complex, unstructured environments. In order to detect a human victim, one or more physical parameters of the victim need to be perceived by a sensor. These physical parameters can be (Burion, 2004): voice, temperature, scent, motion, skin color, face or body shape. In recent years, a number of research teams have developed human victim detection algorithms based on the detection of these physical parameters. In this section we give an overview of these activities and approaches used for human victim detection.

Voice

Kleiner et al. (Kleiner, 2007) perform audio-based victim detection by positioning two microphones with known distance. Given an audio source left, right or between both microphones, the time difference, i.e. phase shift, is measured between both signals. This is carried out by the Crosspower Spectrum Phase (CSP) approach, which allows to calculate the phase shift of both signals based on the Fourier transformation. Using this approach, the bearing of the sound source can be successfully determined, if there is not too much background noise.

Temperature (body heat)

Infrared cameras give a complete picture of the environment heat which is very useful in human detection. Although infrared cameras are very expensive, they seem the best solution to make the discrimination between human and non human presence and, as such, seem to be essential to a robust and efficient solution for human finding. Infrared cameras are used by Kleiner and Kummerle in (Kleiner, 2007), Nourbakhsh et al. in (Nourbakhsh, 2005) and Birk et al. in (Birk, 2006). A new development in this field is the hyperspectral IR imaging approach of Trierscheid et al. (Trierscheid, 2008). Hyperspectral IR images contain a contiguous spectrum in the bandwidth of IR in a spatial scanline in the scene and provide the technique to combine spectroscopy and imaging, which makes it very suited for human victim detection.

Pyroelectric sensors are another type of heat-based human detectors. These sensors are designed specifically for human detection. This sensor is made of a crystalline material that generates a surface electric charge when exposed to heat in the form of infrared radiation. It is calibrated to be sensitive to human heat wavelength (8 - 14 µm). These sensors are very sensitive, cheap and robust. They are composed of two infrared sensors, so they detect humans only if the human or the sensor is moving.

Pyroelectric sensors have been used by Pissokas and Malcolm in (Pissokas, 2002) and Nourbakhsh et al. in (Nourbakhsh, 2005)

Scent

CO_2 sensors allow to detect the carbon dioxide emission, and even the breathing cycle of a victim. It is thus possible to determine if he is still alive. These sensors have been used by a number of participants of the RoboCupRescue, but the disadvantage is that the response time of a CO_2 sensor is very slow and that the sensor has to be very close to the victim to have useful data because it is very directional and depends much on the air conditions like humidity, temperature, wind, and dust. This makes it difficult to use it in a disaster area.

Motion

Motion can be detected by a variety of sensors (sonar, laser, visual & IR camera …) and can serve as an indication that somebody alive is present. However, motion analysis alone can never determine whether the cause of this motion field is a human being. Therefore, it is only used in combination with other characteristics (Kleiner, 2007), (Nourbakhsh, 2005).

Skin Color

Skin color is a popular parameter in the computer vision community to detect humans. Visser et al. use skin color in (Visser, 2007) for human victim detection. They construct a 3D color histogram in which discrete probability distributions are learned. Given skin and non-skin histograms based on training sets, the probability that a given color value belongs to the skin and non-skin classes can then be learned. The problem with these approaches is twofold: 1) in unstructured outdoor environments, there is no a priori data on the colors present in the environment (which could lead to a large number of false positives), and 2) the field of view of typical outdoor robot cameras is quite large, which means that a person's face only consists of a limited number of pixels (which would reduce the detection rate).

Face & Body Shape Detection

Another popular approach in computer vision to detect persons is to perform face detection. Other detectors are specifically trained at detecting the upper body. Together, these detectors provide powerful cues for reasoning about a person's presence. The problem with these methods is that detecting victims lying on the ground using standard camera images is very different from standard person detection. These standard person detection algorithms, relying on face or upper body detection, assume that the person's face is clearly visible in the camera image and that the person is standing straight up, such that the upper body can be easily detected. Victims, however, do not tend to stand up and do not tend to look straight into the camera. Therefore, special techniques have to be applied, as proposed for example in (Bahadori, 2003). The approach presented in this article aims to classify the body shape of lying human victims and thus falls into this category of victim detectors.

Combined approaches

With such a multitude of detection approaches, each having their advantages and disadvantages, it is evident that the integration of multiple cues can provide better results. Therefore, several teams have investigated hybrid approaches, mixing for example motion, sound and heat in (Nourbakhsh, 2005) or motion sound and faces. Others (Kleiner, 2007) have focused specifically on determining the best way to integrate the information from all cues, leading to MRF-based approaches.

4.3.2 The Viola-Jones based victim detector

Here, we present an approach to achieve robust victim detection in difficult outdoor conditions. The basis for this work is a learning-based object detection method, proposed by Viola and Jones(Viola, 2001). Viola and Jones originally applied this technique in the domain of face detection (Viola, 2004). Their system yields face detection performance comparable to the best previous systems and is considered the fastest and most accurate pattern recognition method for faces in monocular grey-level images.

The method operates on so-called *integral images*: each image element contains the sum of all pixels values to its upper left allowing for constant-time summation of arbitrary rectangular areas.

During training, weak classifiers are selected with AdaBoost, each of them a pixel sum comparison between rectangular areas.

The object detection system classifies the images using simple features. The features are able to encode ad-hoc domain knowledge and the features-based system operates much faster than a pixel-based one. The Haar-wavelets are single-wavelength square waves, which are composed by adjacent rectangles. The algorithm does not use true Haar-wavelets, but better suited rectangle combinations. This is why they are called Haar-features instead of Haar-wavelets. The Haar-features detection procedure works by subtracting the average dark-region pixel value from the average light-region pixel value. If the difference is above a threshold, which is set during the training, the feature is present.

Hundreds of these classifiers are arranged in a multi-stage cascade. Lazy successive cascade evaluation and the constant-time property allow the detector to run fast enough to achieve an overall low latency.

In a first attempt at victim detection, we used the standard Viola-Jones detector for face and upper body detection. The first tests were executed on indoor and good quality images. These tests were very successful, 90% of the faces and 80% of the upper bodies were detected. All together the hit rate reached the 95% while the false alarm rate stayed under 25%. However, the target hardware, the

RobuDem, is going to operate in outdoor environment where the background is various and the illumination is unpredictable. So, outdoor experiments were strongly suggested. Although the results were better than expected, the false alarm was increased dramatically while the hit rate was decreased to 70% for the upper body and to 30% for the face detection.

The conclusion from these tests is that in outdoor environment the face detection based person detection is not viable. Usually it only consumes the computation time without giving any results or any correct results. If the detection was more detailed, the system became too slow with minor success. If the detection was tuned to be faster, the hit rate decreased below 10%.

The upper body detection is more robust, as it adapts itself to different illumination conditions much better. However, it gives much more false alarms.

Our first idea was to fuse the face and upper body detector for having a more robust system. Unfortunately, the face detector does not really improve the performance of the upper body detector. Also, these detectors work only for standing, sitting or crouching person detection. In the case of the person lying on the ground, all of the detectors based on the existing Haar-cascade classifiers fail. Therefore, we decided to fuse the upper body detector with a new detector which has to be trained for victims.

First, we adapted the Viola-Jones technique, by training the algorithm with bodies, lying on the ground. To deal with the huge number of degrees of freedom of the human body and the camera viewpoint, the configuration space for human victims was reduced to victims lying face down and more or less horizontally in front of the camera. This case has been chosen because in real disasters this pose has the highest probability. The people try to protect their head and their ventral body parts which are the most vulnerable. Another reason is that in this position, the possible positions of the limbs form a relatively small pool comparing to the other cases. Also the orientation of the body must be considered because the legs have a different shape than the upper body and the head. To handle this, the sample images were taken with the both body orientations (left-to-right and right-to left). To enlarge the data-set, the images were then later flipped horizontally and re-used during the Haar-training.

Figure 4.3 shows some example of the sample images. They were taken in an outdoor environment, featuring human victims in several orientations and under varying illumination. These images were taken with an on-board stereo camera system. In total, 800 positive scenes were recorded, in each case the color rectified image of the left and right camera were recorded.
Furthermore 500 pairs of negative images were recorded outside and 100 pairs inside. These images contain no humans but the variety of the background is high in order to make the AdaBoost learning method set up good thresholds.

4.3.3 Human victim detection results

Theoretically, the cascaded Haar-classifier for victim detection has a 100% detection rate and less than 10^{-6} % false alarm rate. Of course, this is only true in the case of the positive and negative sample images. With new test images – which were taken in similar illumination conditions as the sample images but mainly in different positions –, the correct detection rate was approximately 65%.
Figure 4.4 shows the result of the victim detection algorithm. The red rectangles are the hits of the detector for the victims whose head is at the left, the yellow ones for those whose head is at the right. In the first image of Figure 4.4, the victim was correctly found besides of a lot of false positives. These false alarms are eliminated by merging the adjacent rectangles of correct posture.

In the case of Figure 4.5, it is more difficult to decide whether a correct classification is performed. In the first picture a smaller and bigger rectangle cover the victim. The smaller rectangle is a true positive, but the bigger rectangle is a false alarm which may have some typical features of a victim. As it is showed in the second picture, these rectangles are considered neighbors and they were merged together. The merging is done by computing an average rectangle; this is why the marked area is bigger than the actual victim.

The processing time for running the victim detector is between 60 and 80 milliseconds, which means 13-16 frames per second. This is a very good result, as it allows near real-time reactions in the robot control scheme and it also allows integrating the results of multiple detection runs over time by means of a tracking scheme, to enhance the detection rate and reduce the false positive rate. As such, the victim detection quality can be further improved, making this methodology a valuable aid for human crisis managers, as enables a robot to scan a designated area for human survivors semi-automatically.

4.4 Stereo-based terrain traversability estimation

4.4.1 Problem statement and state of the art of terrain traversability estimation

Terrain traversability analysis is a research topic which has been in the focus of the mobile robotics community in the past decade, inspired by the development of autonomous planetary rovers and, more recently, the DARPA Grand Challenge. However, already in 1994, Langer et al. (Langer, 1994) computed elevation statistics of the terrain (height difference and slope) and classified terrain cells as traversable or untraversable by comparing these elevation statistics with threshold values. Most of the terrain traversability analysis algorithms employ such a cell-based traversability map, which can be thought of as a 2.5D occupancy grid. The problem with Langer's method was that the traversability was only expressed in binary forms and soon other researchers (Singh, 2000), (Gennery, 1999) presented solutions to lift this limitation. In (Seraji, 2003), Seraji proposed a fuzzy-logic traversability measure, called the Traversability index, which represents the degree of ease with which the regional terrain could be navigated. This degree was calculated on the basis of the terrain roughness, the slope and the discontinuity, as measured by a stereo vision system.

Schäfer et al. presented in (Schäfer, 2005) a similar stereo-discontinuities based approach without explicit calculation of a traversability map. Other researchers (Shneier, 2006), (Kim, 2006), (Happold, 2006) have embedded the stereo-based terrain traversability analysis in an on-line learning approach. The results of these methods depend greatly on the quality of the training set.

In ((Ulrich, 2000), Ulrich and Nourbakhsch presented a solution for appearance-based obstacle detection using a single color camera. Their approach makes the assumption that the ground is flat and that the region in front of the robot is ground. In (Kim, 2007), Kim et al. present another single-camera traversability estimation method based upon self-supervised learning of superpixel regions.

Besides monocular and stereo vision, laser range finders are a useful sensor for terrain traversability estimation. In (Andersen, 2006), Andersen et al. present a method for terrain classification using single 2D scans. The Stanford Racing Team (Thrun, 2006) utilized a Traversability Map based on data from six laser scanners registered with pose from an unscented Kalman Filter to classify grids as undrivable, drivable, or unknown. A Markov model was used to probabilistically test for the presence of an obstacle leading to an improved Traversability Map.

4.4.2 V-disparity based terrain traversability estimation

Detecting obstacles from stereo vision images may seem simple, as the stereo vision system can provide rich depth information. However, from the depth image, it is not evident to distinguish the traversable from the non-traversable terrain, especially in outdoor conditions, where the terrain roughness and the robot mobility parameters must be taken into account. Our approach is based on the construction and subsequent processing of the *v-disparity* image (Labayrade, 2002), which provides a robust representation of the geometric content of road scenes. The v-disparity image is constructed by calculating a horizontal histogram of the disparity stereo image.

Consider 2 stereo frames, as shown in **Error! Reference source not found.**4.6 and 4.7, and the computed disparity image I_D, as shown in **Error! Reference source not found.**4.8. Then, the v-disparity image I_V can be constructed by accumulating the points with the same disparity that occur on a horizontal line in the image. **Error! Reference source not found.**4.9 displays the v-disparity image I_V for the given input images.

The classification of the terrain in traversable and non-traversable areas goes out from the assumption that the majority of the image pixels are related to traversable terrain of the ground plane. The projection of this ground plane in the v-disparity image is a straight line, from the top left to the bottom right of the v-disparity image. Any deviations from this projection of the ground plane are likely obstacles or other non-traversable terrain items.

As such, the processing of the v-disparity image comes down to estimating the equation of the line segment in the v-disparity image, corresponding to the ground plane. This is done by performing a Hough transform on the v-disparity image and searching for the longest line segment. Then, one must choose a single parameter which accounts for the maximum terrain roughness. As this parameter depends only on the robot characteristics, it only needs to be set once. This parameter sets the maximum offset in v-disparity space to be considered part of the ground plane. Any outliers are regarded as obstacles, which enables to compile an obstacle image I_O.

4.4.3 Terrain traversability estimation results

Figure 4.10 shows the V-Disparity image after Hough transform. The red line indicates the largest line segment, corresponding to the ground plane. The two pink lines indicate the region in v-disparity space where pixels are considered part of a traversable region. Terrain corresponding to pixels in v-disparity space in between the two pink lines is considered traversable, otherwise it is considered as an obstacle. The result of this operation can be judged from the right image of **Error! Reference source not found.** 4.11, showing the obstacle image. This is a version of the color input image, where false color data corresponding to the disparity is superposed for pixels classified as belonging to non-traversable terrain.

It may be noted that the lower part of the legs of the person standing in front of the robot were not detected as obstacles. This is due to the choice of the threshold parameter for the ground plane, discussed above. After tests in multiple environments, we used a threshold parameter of 50, which offers a good compromise between a good detection rate and low false positive detection rate.

4.5 Behavior-based control

4.5.1 Problem statement and state of the art of behavior-based control

Following the robot control architecture depicted by Figure 4.2, the Robudem robot employs a behavior based control architecture, which allows it to navigate to a number of goal positions while avoiding obstacles and detecting human victims. As can be noted from Figure 4.2, a number of behaviors are foreseen for this application. In order for the robot to accomplish its task, 2 main questions need to be solved:

1. How can we design the individual behaviors, such that the robot is capable of avoiding obstacles and of navigating semi-autonomously?
2. How can these individual behaviors be combined in an optimal way, leading to a rational and coherent global robot behavior?

Indeed, the performance of the behavior-based controller depends on the implementation of the individual behaviors as well as on the method chosen to solve the behavior fusion or action selection problem. However, as in behavior-based control, the idea is to split a complex task in multiple simple tasks or behaviors, the design of the individual behaviors in general does not pose major problems. However, the behavior fusion or action selection problem does require deeper investigation.

In general, the action selection problem can be formulated as a multiple objective decision making problem, as proposed by Pirjanian in (Pirjanian, 1999). Mathematically, a multiobjective decision problem can be represented as finding the solution to:

$$\arg\max_{\mathbf{x}} \; o_1(\mathbf{x}), \ldots, o_n(\mathbf{x}) \qquad [4.1]$$

A common method for solving a multiple objective decision making problem is the weighting method. This method is based on scalar vector optimization and is formulated in the following way:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in X} \sum_{i=1}^{n} w_i o_i(\mathbf{x}) \qquad [4.2]$$

where w_i is a set of weights, normalized such that:

$$\sum_{i=1}^{n} w_i = 1 \qquad [4.3]$$

An important question is how to vary the weights w_i to generate the whole or a representative subset of the solutions. Most approaches described in the literature handle the problem in the same manner. Each weight w_i is considered to have a reasonable number of discrete values between 0 and 1. Then the equation is solved for each combination of values. The obvious limitation of this approach is the large number of computations to be performed, as the computational complexity grows exponentially with the number of objective functions.

The interval criterion weights method (Steuer, 1974) reduces the number of computations by incorporating some subjective knowledge in the process of generating optimal solutions. Basically, rather than letting each w_i assume values in the interval [0; 1], a narrower interval is used that reflects the importance of each objective function. Apart from reducing the computational burden, the interval criterion weights method, produces the `best' part of X*. Here 'best' reflects the decision maker's preferences which are expressed as weight values. A further advantage of the interval criterion weights method is due to the fact that in practice a decision maker seems to be more comfortable defining the range within which a weight value should be, rather than estimating the single weight values.

Goal programming methods (Nakayama, 2006) define a class of techniques where a human decision maker gives his/her preferences in terms of weights, priorities, goals, and ideals. The concept of best alternative is then defined in terms of how much the actual achievement of each objective deviates from the desired goals or ideals. Further, the concept of best compromise alternative is defined to have the minimum combined deviation from the desired goals or ideals. Goal programming methods thus choose an alternative having the minimum combined deviation from the goal $\hat{o}_1, \ldots, \hat{o}_n$, given the weights or priorities of the objective functions. According to the goal programming theorem, the solution to the multi-objective decision making problem can be expressed as:

$$\arg\min_{\mathbf{x} \in X} \sum_{i=1}^{n} w_i \left| o_i(\mathbf{x}) - o_i^* \right|^p \qquad [4.4]$$

where $1 \leq p \leq \infty$, $\mathbf{o}^*$ is the ideal point, w_j is the weight or priority given to the j^{th} objective.

In section 4.5.3 of this chapter, we will present a novel way of combining the inputs of multiple behaviors, but first we'll investigate the design of each individual behavior for the chosen application.

4.5.2 Behavior Design

As mentioned before, a behavior is a function which relates sensor measurements to actions in the form of an objective function. In the case of robot control, the objective function of each behavior can be regarded as two-dimensional normalized function of robot steering velocity v and direction α. For this setup, three behaviors are defined which relate the abstract sensor information into robot actions.

These three behaviors are:

1. Obey Joystick Commands

If desired, the human operator can control the robot by means of a joystick. The joystick commands are directly related to the robot steering angle and direction, so the transformation of the joystick control command into an objective function can be performed straightforward by calculating a two-dimensional Gaussian from the joystick input $v_{Joystick}, \alpha_{Joystick}$:

$$o_{Joystick}(v,\alpha) = \frac{1}{\sqrt{2\pi^2 \sigma^4}} \exp\left(-\left(\frac{v - v_{Joystick}}{2\sigma^2}^2 + \frac{\alpha - \alpha_{Joystick}}{2\sigma^2}^2\right)\right) \qquad [4.5]$$

2. Obstacle Avoidance Using Stereo

To drive the robot away from obstacles detected by the terrain traversability analysis algorithm, the obstacle image is analyzed. The depth values of pixels corresponding to obstacles are accumulated per vertical line in the image and the resulting function is inverted and normalized. This allows to deduce a function *f* of the viewing angle α as shown on Figure 4.12. This function can be regarded as a one-dimensional objective function for obstacle avoidance from stereo input, considering only the viewing / steering angle. This one dimensional objective function can then be extended for velocity as well, using the following formulation:

$$o_{stereo}(v,\alpha) = \frac{f(\alpha)}{1 + |vf(\alpha)/c|} \qquad [4.6]$$

3. Go To Goals

The goal seeking behavior is assigned two tasks. First, it points the robot to the goal position and it varies the velocity respective to the distance to the goal. This means the development of the objective function can be split up as:

$$o_{GoToGoal}(v,\alpha) = o^{\alpha}_{GoToGoal}(\alpha).o^{v}_{GoToGoal}(v) \qquad [4.7]$$

To calculate these objective functions, the (Euclidian) distance to the goal d_{Goal} and heading to this goal θ are calculated from the current robot position given by the GPS system and the current waypoint given by the global path planner. The goal seeking behavior aims to minimize the difference between the robot heading α and the goal heading θ, which can be formulated as:

$$o_{GoToGoal}^{\alpha} = \frac{1}{1+\left(\frac{\alpha-\theta}{\beta}\right)^2} \quad [4.8]$$

with β the window size which is considered. $o_{GoToGoal}^{v}(v)$ is set up such that the velocity is always high, with the exception that when the robot approaches a goal position, the speed should be reduced. This is expressed as:

$$o_{GoToGoal}^{v}(v) = \begin{cases} \left(\frac{v}{v_{\max}}\right)^2 & \text{if} \quad d_{Goal} > d_{Threshold} \\ \frac{1}{1+\left(\frac{v}{v_{\max}}\right)^2} & \text{if} \quad d_{Goal} < d_{Threshold} \end{cases} \quad [4.9]$$

4.5.3 Behavior Fusion

In (Doroftei, 2007), we proposed a method to choose the weights based upon a reliability measure associated to each behavior. The principle behind the calculation of the activity levels is that the output of a behavior should be stable over time in order to trust it. Therefore, the degree of relevance or activity is calculated by observing the history of the output of each behavior. This history-analysis is performed by comparing the current output $\varpi_{\varpi_{j,k}}^{b_i}$ to a running average of previous outputs, which leads to a standard deviation, which is then normalized. For a behavior b_i with outputs ϖ_j these standard deviations $\sigma_{\varpi_j}^{b_i}$ are:

$$\sigma_{\varpi_j}^{b_i} = c_{\varpi_j} \sum_{k=i-h}^{i} \left(\varpi_{\varpi_{j,k}}^{b_i} - \frac{\sum_{l=1}^{N} \varpi_{\varpi_{j,l}}^{b_i}}{N} \right)^2 \qquad [4.10]$$

The bigger this standard deviation, the more unstable the output values of the behavior are, so the less they can be trusted. The same approach is followed all behaviors. This leads to an estimate for the activity levels or weights for each behavior:

$$w_{b_i} = \sum_{j=1}^{number\ of\ outputs} 1 - \sigma_{\varpi_j}^{b_i} \qquad [4.11]$$

For stability reasons, the activity level is initialized at a certain value (in general 0.5) and this estimate is then iteratively improved.

The approaches towards solving the multiple objective decision making problem for action selection which we have reviewed up to this point all have their advantages and disadvantages. Solving the multiple objective decision making problem using reliability analysis has the big advantage of incorporating direct information from the system under control into the control process. On the other hand, this architecture does not offer a human decision maker the ability to interact with the decision process. As autonomous agents more and more have to interact with humans on the field, exchanging knowledge and learning from each other, this is a serious shortcoming. Common techniques for solving the multiple objective decision making problem while taking into account a decision maker's preferences take into account these issues by offering a human operator the ability to input some objectives or ideal points. These approaches, however, suffer from the disadvantage that no reliability data from the sensing and other processes is taken into account while performing action selection. One could thus argue that while reliability analysis-based approaches are too robot centric, these second set of approaches is too human-centric. Here, we propose an approach to integrate the advantages of both theorems. This can be achieved by minimizing the goal programming and reliability analysis constraints in an integrated way, following:

$$\arg\min_{\substack{\mathbf{x}\in X \\ w_i\in\mathbf{w}}}\left[\lambda\left(\sum_{i=1}^{n} w_i\left|o_i(\mathbf{x})-o_i^*\right|^p\right)+\ 1-\lambda\ \sum_{i=1}^{n}\left|w_i-\sum_{j=1}^{number\ of\ outputs}\ 1-\sigma_{\varpi_j}^{b_i}\ \right|^p\right] \tag{4.12}$$

with λ a parameter describing the relative influence of both constraints. This parameter indirectly influences the control behavior of the robot. Large values of λ will lead to a human-centered control strategy, whereas lower values will lead to a robot-centered control strategy. The value of λ would therefore depend on the expertise or the availability of human experts interacting with the robot.

It is obvious that this method increases the numerical complexity of finding a solution, but this does not necessarily leads to increased processing time, as the search interval can be further reduced by incorporating constraints from both data sources.

4.6 Results and conclusions

In this chapter, we have discussed three main aspects of the development of a crisis management robot.

First, we have presented an approach for the automatic detection of human victims. This method is based upon the Viola-Jones (face) detector, which was adapted, such that human victims lying on the ground can be detected. The results of this approach are very encouraging, although further research is required to increase the detection rate and reduce the number of false positives. This will be done by integrating the human victim detector in a tracking scheme.

Next, we have presented a stereo-based terrain traversability estimation algorithm. This method is based upon the analysis of the stereo disparity, more specifically the vertical histogram of the disparity image, called the v-disparity. This approach makes it possible to robustly classify the terrain of outdoors scenes in traversable and non-traversable regions quickly and reliably, using only a single parameter to describe the robot mobility characteristics.

Ultimately, we have shown how all different robot capabilities can be integrated into a robot control architecture. A behavior-based control scheme is employed due to its flexible design. In this context, a novel approach for solving the behavior fusion problem was presented. This method combines the advantages of the traditional weighting methods and the more recent reliability analysis based method.

As a final result, the Robudem robot has become a semi-autonomous agent: it can handle a high-level task (searching for human victims) with minimal input from human operators, by navigating in a complex, dynamic and environment, while avoiding potentially hazardous obstacles. If required, a remote human operator can still take control of the robot via the joystick, but in normal operation mode, the robot navigates autonomously to a list of waypoints, while avoiding obstacles (thanks to the stereo-based terrain traversability estimation) and while searching for human victims.

4.7 References

Andersen J C, Blas M R, Ravn O, Andersen N A and Blanke M (2006), 'Traversable terrain classification for outdoor autonomous robots using single 2D laser scans'. *Journal of Integrated Computer-Aided Engineering*, **13**(3), 223-232.

Bahadori S and Iocchi L (2003), 'Human Body Detection in the RoboCup Rescue Scenario', *First International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster*.

Birk A, Markov S, Delchev I and Pathak K (2006), 'Autonomous Rescue Operations on the IUB Rugbot', *International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, IEEE Press.

Burion S (2004), *Human Detection for Robotic Urban Search and Rescue*, Diploma Work, EPFL.

Colon E, Sahli H and Baudoin Y (2006), 'CoRoBa, a multi mobile robot control and simulation framework', *International Journal of Advanced Robotic Systems*, **3**(1), 73-78.

Doroftei D, Colon E, De Cubber G (2007), 'A behaviour-based control and software architecture for the visually guided Robudem outdoor mobile robot', *ISMCR2007*, Warsaw, Poland.

Gennery D B (1999), 'Traversability analysis and path planning for a planetary rover', *Autonomous Robots*, 1999, **6**(2), 131-146.

Happold M, Ollis M and Johnson N (2006), 'Enhancing supervised terrain classification with predictive unsupervised learning', *Robotics: Science and Systems*.

Jones J L (2004), *Robot Programming: A practical guide to Behavior-Based Robotics*.

Kim D, Oh S and Rehg J M (2007), 'Traversability Classification for UGV Navigation: A Comparison of Patch and Superpixel Representations' *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 07)*, San Diego, CA.

Kim D, Sun J, Oh S M, Rehg J M and Bobick A (2006), 'Traversability Classification Using Unsupervised On-Line Visual Learning for Outdoor Robot Navigation', *IEEE Intl. Conf. on Robotics and Automation (ICRA 06)*, Orlando, FL.

Kleiner A and Kuemmerle R (2007), 'Genetic MRF model optimization for real-time victim detection in Search and Rescue'. *International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, California.

Labayrade R, Aubert D, Tarel J P (2002), 'Real Time Obstacle Detection on Non Flat Road Geometry through V-Disparity Representation', *IEEE Intelligent Vehicles Symposium*, Versailles, France.

Langer D (1994), 'A Behavior-Based system for off-road navigation', *IEEE Transactions on Robotics and Automation*, **10**(6) 776-783.

Nakayama H, Yun Y (2006), 'Generating Support Vector Machines Using Multi-Objective Optimization and Goal Programming'. *Multi-Objective Machine Learning*, 173-198.

Nourbakhsh I R, Sycara K, Koes M, Yong M, Lewis M and Burion S (2005), 'Human-Robot Teaming for Search and Rescue', *IEEE Pervasive Computing*, **4**(1), 72-78.

Pirjanian P (1999), *Behavior coordination mechanisms - state-of-the-art*, Institute of Robotics and Intelligent Systems, School of Engineering, Univ. of South California.

Pissokas J and Malcolm C (2002), 'Experiments with Sensors for Urban Search and Rescue Robots', *International Symposium on Robotics and Automation*.

Schäfer H, Proetzsch M, Berns K (2005), 'Stereo-Vision-Based Obstacle Avoidance in Rough Outdoor Terrain', *International Symposium on Motor Control and Robotics*.

Seraji H (2003), 'New traversability indices and traversability grid for integrated sensor/map-based navigation', *Journal of Robotic Systems*, **20**(3) 121-134.

Shneier M O, Shackleford W P, Hong T H, Chang T Y (2006), 'Performance Evaluation of a Terrain Traversability LearningAlgorithm in The DARPA LAGR Program', *PerMIS, Performance Metrics for Intelligent Systems (PerMIS) Workshop,* Gaithersburg, MD, USA.

Singh S et al (2000), 'Recent Progress in local and global traversability for planetary rovers', *IEEE International Conference on Robotics and Automation*.

Slama D, Garbis J and Russell P (1999), *Enterprise CORBA*, Prentice Hall.

Steuer R E (1974), 'Interval Criterion Weights Programming: A Portfolio Selection Example, Gradient Cone Modification, and Computational Experience', *Tenth Southeastern Institute of Management Sciences Meeting*, Clemson University Press, 246-255.

Thrun S et al (2006), 'Stanley: The robot that won the darpa grand challenge: Research articles', *Journal of Robotic Systems*, **23**(9) 661-692.

Trierscheid M, Pellenz J, Paulus D and Balthasar D (2008), 'Hyperspectral Imaging for Victim Detection with Rescue Robots', *IEEE Int. Workshop on Safety, Security and Rescue Robotics*.

Ulrich and Nourbakhsh I (2000), 'Appearance-Based Obstacle Detection with Monocular Color Vision', *Proceedings of AAAI*.

Viola P and Jones M (2001), 'Robust Real-time Object Detection', *Intl. Workshop on Statistical and Computational Theories of Vision*.

Viola P and Jones M (2004), 'Robust Real-Time Face Detection', *International Journal of Computer Vision,* **57**(2), 137-154.

Visser A, Slamet B, Schmits T, González Jaime L A and Ethembabaoglu A (2007), 'Design decisions of the UvA Rescue 2007 Team on the Challenges of the Virtual Robot competition', *Fourth International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster*, Atlanta, GA.