In *International Conference on Intelligent Autonomous Systems* (IAS 10),
Ottawa, Canada, August 30-September 1 2010.

Safe Navigation on a Mobile Robot using Local and Temporal Visual Cues

XIANG LI [a,1], MOHAN SRIDHARAN [a]
[a] *Department of Computer Science, Texas Tech University, USA*
{*xiang.li,mohan.sridharan*}*@ttu.edu*

Abstract. An open challenge to the widespread deployment of mobile robots in the real-world is the ability to operate autonomously in dynamic environments. Such autonomous operation requires full utilization of the relevant sensory inputs to adapt to environmental changes. Despite being a rich source of information, vision is however, still under-utilized in robot domains because of the sensitivity to environmental changes and the computational complexity of visual input processing algorithms. This paper enables a mobile robot to better utilize the visual input to navigate safely in dynamic environments—it describes a novel algorithm that: (a) uses local image gradient cues to characterize target objects reliably and efficiently; and (b) uses temporal correspondence of visual cues for robust localization and tracking of environmental obstacles. Furthermore, the information extracted from these visual cues is merged effectively with information obtained from other visual cues and range sensors, using autonomously learned error models of the different information processing schemes. All algorithms are fully implemented and tested on a humanoid robot in dynamic indoor environments.

Keywords. Machine Learning and Adaptivity, Perception, Humanoid Robotics, Robust Sensor Fusion.

1. Introduction

The ready availability of high-fidelity sensors [3] and the development of sophisticated algorithms has resulted in the deployment of mobile robots in several applications [2,5,14,23]. However, one key challenge to the widespread deployment of mobile robots is the ability to operate autonomously by adapting to environmental changes. Each sensor mounted on a mobile robot has a limited field of view, and the sensory inputs can be processed using algorithms with different levels of uncertainty. Images from a camera, for instance, are a richer source of information than range finders. However, the visual input is more noisy and the visual information processing algorithms are computationally expensive. As a result, the high-level decisions in many robot applications are still based on non-visual sensory inputs [23]. In addition, it is not feasible for a robot to process all the sensory inputs and still respond in real-time to dynamic changes such as the movement of objects or a change in the illumination. At the same time, relevant sensory inputs need to be utilized in order to respond autonomously to such changes. A body of impressive work exists on vision-based learning—see [20] for a survey, and planning of visual processing [21]. However, very few methods are computationally efficient, au-

[1]Corresponding Author: Xiang Li, xiang.li@ttu.edu

tonomous and able to model the uncertainty of robot domains. Similarly, though sensor fusion has been studied in different fields [1,13], many robot domains use manually specified heuristics for merging sensory inputs [23].

The above-mentioned challenges are offset by the presence of a moderate amount of structure in many robot domains, which can be used to operate autonomously. Our prior work enabled a mobile robot to learn error models of different sensory input processing algorithms, and to use these models to effectively merge the extracted information [19]. However, color was used as the primary visual feature to characterize objects of interest. This paper enables a mobile robot to better utilize visual input by: (a) using a combination of an efficient gradient feature detector (MSER [8]) and a reliable feature descriptor (SIFT [7]) to characterize target objects; and (b) using temporal visual cues to robustly localize and track the desired objects. The local, temporal and color-based visual cues are merged with range information, using the existing autonomous information fusion scheme. All algorithms are implemented on humanoid robots (Naos [11]) for safe navigation in dynamic environments.

The remainder of the paper is organized as follows. Section 2 briefly reviews some related work. Section 3 presents the proposed method based on local and temporal visual cues, in addition to the overall information fusion strategy. The experimental setup and results are described in Section 4, followed by the conclusions in Section 5.

2. Related Work

This section describes related work on the use of local and temporal visual cues, in addition to instances of integrated robots systems deployed in real-world applications.

Vision research has resulted in the development of several methods that use local visual cues for tasks such as object recognition [4] and robot localization [17]. These approaches are based on local gradient descriptors designed to be invariant to scale, orientation, affine transformations and illumination [7,8,9]. A recent comparison of these techniques [25] has shown that SIFT [7] provides the most reliable performance, while MSER [8] provides the highest efficiency. Other recent methods such as FERN [12] have used simpler features for efficient operation, but require extensive training.

Mobile robots are increasingly being deployed in real-world applications such as disaster rescue and human-robot interaction [2,5,14,23]. This deployment is a result of sophisticated algorithms, for instance for the control and balance of humanoid robots [15,16]. Specific algorithms have also been designed to enable mobile robots to use image gradients [17], stereo input [6] and other visual cues [20]. The visual information is typically fused with range information, based on sensor fusion research in fields such as networks and multiagent systems [1,13]. However, high-level decisions in many robot applications are still based on non-visual sensors because the visual input is sensitive to environmental changes, and visual input processing algorithms are computationally expensive. However, prior work has shown that a robot can use visual input to autonomously adapt to environmental changes [20], and that incorporating temporal cues results in robust performance in tasks such as navigation [10]. This paper therefore uses temporal and local image cues to robustly characterize and localize the desired target objects, in order to navigate safely in dynamic indoor environments.

3. PROPOSED APPROACH

This section describes the test scenario and robot platform, the proposed approach to characterize and localize target objects, and the information fusion scheme.

(a) Overall Test Scenario. (b) Robot Soccer Domain. (c) Robots.

Figure 1. (a) The overall human-robot collaboration scenario; (b) The robot soccer framework; and (c) Robot platforms: wheeled and humanoid.

3.1. Test Scenario and Robot Platform

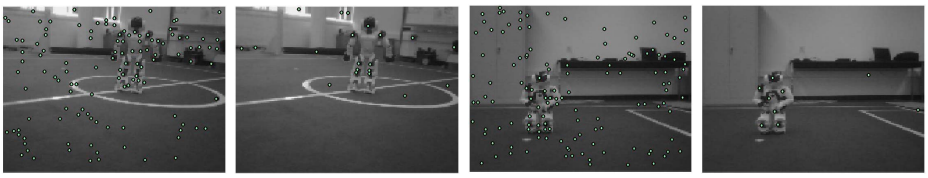
The work described in this paper is part of a project that enables mobile robots to monitor the elderly in indoor scenarios—Figure 1(a). This scenario requires processing management (i.e. tailoring sensing and processing to the task) and bootstrap learning (i.e. learning and adaptation using sensory inputs). This paper presents an instance of bootstrap learning using local and temporal visual cues.

Though the scenario uses both robot platforms shown in Figure 1(c), the experiments reported in this paper use the *Nao* [11], a 58cm tall humanoid robot with 23 degrees of freedom—see Figure 1(b). The primary sensors are two color cameras, though only one camera can be used at a time. Each camera has a 58^o diagonal field of view and provides images at a resolution of 640×480, 320×240 or 160×120. There are two ultrasound sensors in the chest, each with a 60^o field of view, other sensors (e.g., accelerometers, bump sensors) and Wi-Fi to communicate with other robots or an off-board PC. All processing is typically performed in real-time (30Hz) on board the robot, using a 500MHz CPU. One popular application domain for the Naos is the Standard Platform League of RoboCup [18], a research initiative with the goal of creating a team of humanoid robots that can beat the champion human soccer team by the year 2050. Scenarios were set up in indoor offices and the robot soccer framework to simulate the challenges related to autonomous vision, navigation and coordination, which need to be addressed in the scenario of Figure 1(a). The task is to navigate safely in the presence of stationary and moving obstacles. The robot uses different visual cues and range information to robustly localize the obstacles, i.e., to compute their relative distance and bearing. This paper uses moving robots as the representative "obstacles", but the techniques are applicable to other environmental objects, robot platforms and application domains.

3.2. Image Gradient-based Obstacle Detection

Features based on local image gradients are being used extensively in computer vision research to characterize and hence recognize the desired objects [7,8,9]. These approaches are aimed at being robust to one or more factors such as scale, orientation, affine transformations, illumination and viewpoint. All such methods have two components: a *detector* that uses second-order gradients to extract *keypoints*, i.e., image regions that are consistent across variations in the viewing conditions; and a *descriptor* that identifies the appearance of each such extracted region compactly. Objects of interest are represented by a database of *feature descriptors* generated from a set of images, and features extracted from test images are compared with this database for classification.

Experimental comparison of the existing detectors and descriptors [25] has shown that the MSER (Maximally Stable Extremal Regions) detector efficiently identifies a small set of unique regions [8], while the 128-dimensional SIFT descriptor provides the most reliable performance [7]. Figures 2(a)–2(d) show images with keypoints detected



(a) SIFT example 1. (b) MSER example 1. (c) SIFT example 2. (d) MSER example 2.

Figure 2. (a),(c) Keypoints detected with DoG+SIFT; (b),(d) Keypoints detected with MSER. MSER finds more distinctive keypoints.

using MSER and the default SIFT technique that uses a Difference of Gaussian (DoG) detector—MSER finds a smaller set of more distinctive image regions. This paper combines these two approaches to characterize objects reliably and efficiently.

The DoG detector for SIFT is implemented in scale-space and has four parameters: (x, y) denote the location of the detected region, σ represents the scale-space for the detector, and θ is the orientation. MSER finds elliptical covariant regions on level sets of the image and has five parameters: (x, y) denote the location of the detected region, (a, b) represent the region as an ellipse, and c represents the ellipse's orientation. The DoG defines the scale space of an image as the function:

$$L(x, y; \sigma) = G(x, y; \sigma) \otimes I(x, y) \tag{1}$$

which is the convolution of a variable-scale Gaussian $G(x; y; \sigma)$ with the image $I(x; y)$. The parameter σ defines the range of the mask and hence that of the detector. The MSER representation can be transformed to that its DoG equivalent in two ways:

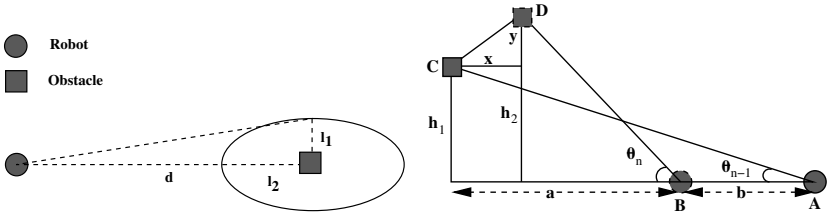
$$\sigma = K \cdot \sqrt{(a^2 + b^2)} \quad \text{or} \quad \max(a, b) \tag{2}$$

These options are compared in Section 4.2. However, the orientation c in MSER cannot be used for DoG whose θ is computed from the orientation histogram in the Gaussian smoothed image. The new orientation in scale-space is hence computed after estimating σ. In this paper, a small set of unique MSER features are hence extracted from image regions with the target object. Next, the equivalent DoG representations are obtained and the SIFT descriptors are computed to build the *training* database of object features. A similar database is built for the environment. A test image region with sufficient number of features similar to the database of obstacle features can be labeled as an obstacle location. The distance and bearing of the obstacle relative to the robot are computed using coordinate transforms that consider the robot's joint angles and compare the physical size of the obstacle with the size of the image region in pixels—see [22]. One key difference is that the training database is learned, as described in Section 4.1.

3.3. Information Fusion

As with other robot platforms, the Nao has multiple information processing schemes: (a) *Ultrasound (US)*: each ultrasound sensor computes object distance up to 150cm—the bearing information is limited to direction (left and/or right) in a 60^o cone; (b) *Vision-Color (VC)*: color segmented image regions are used to detect objects; and (c) *MSER-SIFT (VM)*: local image gradients are used to detect objects.

The moving obstacles (i.e., other robots) have red or blue parts arranged in a pattern on the shoulder, head, chest and arms. VC localizes the obstacles by detecting patterns of a suitable color, and computing relative distance and bearing using the coordinate transforms used for VM. However, VC only works from viewpoints where the uniform



(a) Model used in the measurement update. (b) Temporal distance estimation scheme.

Figure 3. (a) Gaussian model used in the measurement update of the Kalman filter; and (b) Pictorial representation of the temporal scheme for distance estimation.

patterns are visible, and it is reliable only up to a distance of $\approx$ 2m as against $\approx$ 4m with VM. In addition, VC and VM compute distance by comparing the known object size with the detected size in image pixels: noise in segmentation or feature detection can hence introduce errors. In terms of computation, US and VC are inexpensive, while VM is expensive. Typically, heuristic constraints are imposed on when and how the information from each of these schemes should be used. Instead, this paper builds on prior work [19] to enable a mobile robot to model the errors in the processing schemes, and use the models to merge the available information. The key differences are: (a) the use of local and temporal visual cues in addition to the color and range input; and (b) the robust localization and tracking of moving obstacles in complex scenarios.

The information fusion algorithm associates each obstacle estimate with a Kalman Filter [24]. The existing estimates are first adjusted to account for the robot's (and obstacle's) motion since the previous update, and stale estimates are removed. Each processing scheme then identifies obstacles and computes the distances and bearings. Next, similar measurements are grouped using the learned error models—if the difference in measurements from two schemes is more than the corresponding expected errors, they are not grouped. A single estimate is then computed for each group:

$$d^j = \sum_i w^j_{d,i} d^j_i, \quad \theta^j = \sum_i w^j_{\theta,i} \theta^j_i \tag{3}$$

where the distance and bearing to the jth obstacle in the current frame (d^j, θ^j) are the weighted average of the individual measurements ($i \in \{US, VC, VM\}$). The weights associated with the values from the ith source ($w_{d,i}$ and $w_{\theta,i}$) are based on the predicted measurement errors. The individual and merged estimates from the current frame are then merged with existing Kalman filter estimates from prior frames or tracked as new estimates. This grouping and merging scheme performs much better than directly using the measured values in the Kalman filters and manually tuning the error models. Figure 3(a) shows the Kalman filter representation of a new estimate: the errors (i.e. axes widths l_1, l_2 of the Gaussian) are based on the measurements (d, α), robot velocity and obstacle velocity. Section 4.1 describes the learning of the error models.

3.4. Bootstrap Learning with Temporal Cues

MSER-SIFT uses a smaller set of unique features (compared to SIFT) to represent the target object. Noise in the feature extraction can hence cause errors in the measured size of the obstacle region in the image, leading to errors in the measured distance to the obstacle. The error in bearing is not as significant due to the limited field of view of

the robot. Segmentation errors can cause similar errors in VC. Temporal visual cues are hence used to achieve reliable localization of the obstacles.

Consider the situation in Figure 3(b): at time t_{n-1} the robot at position A detects an obstacle at bearing θ_{n-1} at point C (distance estimate is noisy). At time t_n, the robot has moved to point B and it detects an obstacle at point D at relative bearing θ_n. If the associated Kalman filters consider the two measurements to be close enough to represent the same obstacle, the *corrected* distance h_2 from the obstacle to the robot's direction of motion can be computed:

$$h_1 = (a+b) \cdot tan(\theta_{n-1}); \quad h_2 = (a-x) \cdot tan(\theta_n); \quad h_1 = h_2 - y \tag{4}$$

$$\implies h_2 = tan(\theta_n) \cdot \frac{tan(\theta_{n-1}) \cdot b + tan(\theta_{n-1}) \cdot x + y}{tan(\theta_n) - tan(\theta_{n-1})}$$

where b is the displacement of the robot between time t_{n-1} and t_n; and x, y represent the obstacle's local motion. The velocity of the obstacle is assumed to be a constant until it is estimated by the robot. As the measurements are corrected, the robot can measure obstacle velocity more accurately, thereby accurately localizing the obstacle, which in turn enables the robot to measure the distance and track the obstacle robustly. This ability to *bootstrap* is a key advantage of the temporal scheme that is used with VM and VC in the information fusion algorithm (Section 3.3).

4. Experiment Setup and Results

This section describes the learning of the error models of the processing schemes and the training database of MSER-SIFT features, followed by the experimental results.

4.1. Bootstrap Learning of Models

The weighted merging of the individual measurements in Section 3.3 uses models that predict the measurement errors of the processing schemes. In order to learn these predictive models, obstacles are placed on the field at positions that are known to the robot, and the robot moves through a sequence of poses (position+orientation) that it can reach accurately. The robot segments input images using a *color map* that maps pixels to numerical color labels. Contiguous regions of the same color are grouped into regions that are used to detect objects. Measured distances and bearings to these objects are used for global localization. At each pose, the robot then compares the actual distance and bearing to the known obstacle locations against the measured values. The error values are used to estimate the parameters of a polynomial function approximator (degree, coefficients) that is then used to predict the error as a function of the measured distance and bearing. The weights in Equation 3 are inversely proportional to this expected error.

At each pose, the robot also projects the known positions of the obstacles within the field of view of the camera, to the image. The MSER-SIFT features extracted from the corresponding image regions are used to train a database of features that represent the obstacles. A similar database is created for the background and other obstacle categories (e.g., humans). The robot also estimates other properties such as the actual size of the obstacle regions that can be detected in the image. In addition, the color map used for segmentation is learned autonomously [20]. The map of the world, though currently provided manually, can be learned by the robot.

Actual \ Observed	Obs	$\overline{Obs}$
Obs	92.0	8.0
$\overline{Obs}$	12.7	87.3

Table 1. Accuracy (%) with K = 1.3.

Actual \ Observed	Obs	$\overline{Obs}$
Obs	93.0	7.0
$\overline{Obs}$	19.3	80.7

Table 2. Accuracy (%) using max().

4.2. Automatic Parameter Tuning

The modeling of objects using MSER-SIFT gradient features involves the certain parameters, which are tuned automatically during the bootstrap learning described above. Feature descriptors extracted from a set of 30 images each with and without obstacles are used to generate the training database. These images contain obstacles at different scales and orientations. A *validation* set is obtained by extracting feature descriptors from a different set of 50 images each with and without obstacles. Feature vectors that are similar are eliminated by computing the ratio of distances between closest and second-closest neighbor of each feature vector [7].

As mentioned in Equation 2 in Section 3.2, there are two ways to compute the σ to transform the MSER representation to an equivalent DoG representation. Tables 1, 2 summarize the best results obtained with the two options, in the standard "confusion matrix" format—for e.g., $Obs|Obs$ represents the true positives (obstacles classified as obstacles). Based on these results, all experiments use the first option with $K = 1.3$.

A nearest neighbor approach is used to detect obstacles in test images. Each extracted feature (descriptor) in the test image is assigned a class label (obstacle, background) by computing the *most similar* feature vector in the trained database of obstacle and background features—the euclidean distance is used as the similarity measure. If the number of test image features that match the trained database of obstacle features is above a threshold, the corresponding image region is considered to be the location of an obstacle. The best value of this threshold is estimated by computing the classification accuracy over the validation set, for different values of the threshold. Figure 4 shows a pictorial representation of the classification accuracy as a function of the number of matched features—the best performance occurs at a threshold of 5, and Figure 5 shows the corresponding classification accuracy.

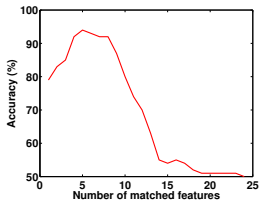


Figure 4. Accuracy vs. no. of matched features.

Actual \ Observed	Obs	$\overline{Obs}$
Obs	85.0	15.0
$\overline{Obs}$	13.0	87.0

Figure 5. Accuracy (%) when no. of matched features $= 5$.

4.3. Experimental Results

The following hypotheses were evaluated: (H1) MSER-SIFT gradient cues enable reliable and efficient recognition of the target object; (b) temporal visual cues increase reliability of obstacle localization; and (c) the information fusion scheme enables reliable tracking of the (moving) obstacles. The experimental setup consisted of stationary (e.g., desks) and moving (e.g., other robots) obstacles in dynamic scenarios such as indoor of-

Method	Testing Time (msec)	Training Time
MSER-SIFT	121.4 ± 35.3	86.5 ± 13.5 msec
SIFT	413.2 ± 72.1	153.7 ± 16.7 msec
FERN	40.7 ± 14.8	26.5 ± 0.3 sec

Table 3. MSER-SIFT takes longer than FERN during testing but is significantly faster during training.

Method	$Obs \vert Obs(\%)$	$\overline{Obs} \vert \overline{Obs}(\%)$
MSER-SIFT	86.8 ± 6.14	87.2 ± 1.75
SIFT	64.6 ± 3.03	81.5 ± 4.7
FERN	85.6 ± 4.20	67.8 ± 4.47

Table 4. Accuracy of the different techniques. MSER-SIFT provides the best performance.

fices and the robot soccer field. Once the obstacles are localized, navigation is based on artificial potential fields, as described in our earlier work [22].

In order to evaluate H1, the MSER-SIFT approach was compared against the default SIFT approach and the method called FERN [12], on a test set of 300 images, with 150 images each with and without obstacles. The evaluation measures used were the accuracy of classification (i.e., reliability) and the running time (i.e., efficiency). The training databases were set up for all three approaches in a similar fashion. From the test set, 200 images were chosen at random for evaluation, and the process was repeated 10 times to obtain the results in Tables 3, 4.

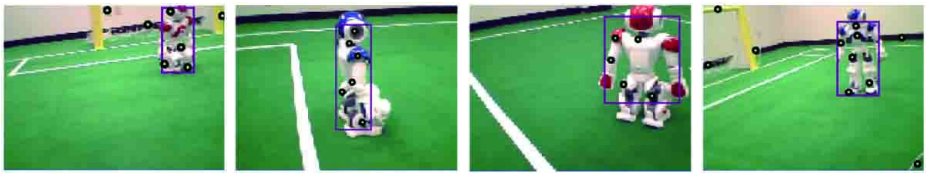
Table 3 shows that MSER-SIFT is significantly faster than SIFT, because it detects a smaller set of more unique features. FERN is the fastest during testing because it uses simpler features. However, unlike MSER-SIFT or SIFT, FERN takes several seconds per image to learn the training database, making incremental revisions infeasible. Table 4 compares the methods in terms of their classification accuracy. MSER-SIFT recognizes obstacles and rejects non-obstacles better than the other two methods. Based on Tables 3, 4, MSER-SIFT is used to characterize the obstacles in all subsequent experiments.

In order to evaluate H2, obstacles were placed at different distances from the robot, with the minimum distance being greater than the maximum range of the ultrasound sensors (1.5m). Then, as the obstacle moved randomly, the robot used temporal cues to correct the distance measurements, as described in Section 3.4. When only MSER-SIFT features were used to compute the distances, the average measurement error over 50 trials was 87.4 ± 45.1cm, making it infeasible to use VM-based distance measurements. However, when temporal cues were included, the error is much smaller: 21.7 ± 18.5cm, and VM-based distances can be used in the information fusion scheme.

Finally, Table 5 evaluates H3— it summarizes the distance error, bearing error and classification accuracy of the processing schemes (US, VC, VM, US+VC+VM). Obstacles placed at different locations performed specific displacements, and the robot walked through fixed poses, using color-coded objects and a learned map of the world for global localization. The distance and bearing errors were computed over 15 trials for 20 different obstacle positions where the obstacles were detected correctly. The detection accuracy was computed over 400 images captured during this testing process—the ground truth was provided manually. The individual processing schemes have different drawbacks (e.g. VM provides accurate bearings but noisy distance measurements). However, a combination of these schemes fully exploits the relevant information, resulting in

Scheme	Error		Accuracy(%)
	Distance (cm)	Bearing (deg)	
Ultrasound (US)	6.5 ± 3.6	$--$	70
Vision-Color (VC)	17.5 ± 8.7	8.5 ± 4.0	81.5
MSER-SIFT (VM)	87.4 ± 45.1	1.8 ± 1.5	86.1
$US + VC + VM$	9.0 ± 4.9	4.8 ± 4.1	92.4

Table 5. The distance and bearing errors, and the detection accuracy of the processing schemes.



(a) Image 1. (b) Image 2. (c) Image 3. (d) Image 4.

Figure 6. (a)–(d) Obstacle localization with MSER-SIFT; detected obstacles are enveloped in pink rectangles.

low measurement errors and high classification accuracy. In order to meet the computational constraints of the test platform, VM is run only once every second. However, the same algorithm has been evaluated on a wheeled robot platform with additional computational resources, resulting in a much faster (and smoother) performance while providing similar reliability. Finally, Figures 6(a)–6(d) show images with rectangular boxes depicting the obstacles detected using all available cues. Additional images and videos can be viewed on the authors' web-sites.

5. Conclusions And Future Work

A key challenge to the widespread deployment of mobile robots is the ability to fully exploit the relevant sensory inputs to operate autonomously in dynamic environments. This proposed algorithm enables a mobile robot to better exploit the available visual information to navigate safely in the presence of mobile obstacles in dynamic indoor environments. The robot characterizes the desired objects using local image gradient cues and temporal visual cues, and effectively merges the corresponding information with other visual cues and range information.

One direction of future work is to apply the proposed method to the overall test scenario of Figure 1(a), using different robot platforms—see Figure 1(c). In such a dynamic scenario, the robots will have to incrementally revise the trained database in response to changes in the obstacle configurations and environmental factors. In addition, other sensory inputs and information processing schemes can be included, and the sensing and information processing can be tailored to the task at hand [21]. Furthermore, the approach can be extended to a team of robots that share information in order to collaborate robustly towards a common objective.

Though this paper focuses on the task of safe indoor navigation, it provides the tools that can be used in many other domains. In addition, the results show that a mobile robot can operate autonomously by bootstrapping off of the learned models of environmental objects and visual features. The long-term goal is to enable mobile robots to autonomously learn environmental models, effectively merge information obtained from different sources, and operate robustly in real-world application domains.

References

[1] R. Brooks and S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Application with Software*. Prentice Hall, 1998.

[2] J. Casper and R. R. Murphy. Human-robot interactions during urban search and rescue at the wtc. In *Transactions on SMC*, 2003.

[3] Videre Design. Videre Design Robot and Sensors, 2010. `http://www.videredesign.com/index.php?id=21`.

[4] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous Object Recognition and Segmentation by Image Exploration. In *European Conference on Computer Vision*, 2004.

[5] Michael A. Goodrich and Alan C. Schultz. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[6] J.-S. Gutmann, M. Fukuchi, and M. Fujita. Real-time path planning for humanoid robot navigation. In *International Joint Conference on Artificial Intelligence*, 2005.

[7] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[8] J. Matas, O. Chum, M.Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, 2002.

[9] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[10] Aniket Murarka, Mohan Sridharan, and Benjamin Kuipers. Detecting Obstacles and Drop-offs using Stereo and Motion Cues for Safe Local Motion. In *International Conference on Intelligent Robots and Systems*, 2008.

[11] Nao. The Aldebaran Nao Robots, 2008. `http://www.aldebaran-robotics.com/`.

[12] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[13] Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[14] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun. Towards Robotic Assistants in Nursing Homes: Challenges and Results. In *RAS Special Issue on Socially Interactive Robots*, 2003.

[15] J. Pratt and B. Krupp. Design of a Bipedal Walking Robot. In *SPIE*, 2008.

[16] J. Rebula, F. Canas, J. Pratt, and A. Goswami. Learning Capture Points for Humanoid Push Recovery. In *International Conference on Humanoid Robots*, 2007.

[17] S. Se, D. Lowe, and J. Little. Global Localization using Distinctive Visual Features. In *International Conference on Intelligent Robots and Systems*, 2002.

[18] SPL. The Robosoccer Standard Platform League, 2008. `http://www.tzi.de/spl/`.

[19] Mohan Sridharan and Xiang Li. Autonomous Information Fusion for Robust Obstacle Localization on a Humanoid Robot. In *International Conference on Humanoid Robots*, 2009.

[20] Mohan Sridharan and Peter Stone. Color Learning and Illumination Invariance on Mobile Robots: A Survey. *Robotics and Autonomous Systems*, 75(1):1–38, 2009.

[21] Mohan Sridharan, Jeremy Wyatt, and Richard Dearden. Planning to See: A Hierarchical Aprroach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence*, 174:704–725, 2010.

[22] P. Stone, K. Dresner, P. Fidelman, N. K. Jong, N. Kohl, G. Kuhlmann, E. Lin, M. Sridharan, and D. Stronger. UT Austin Villa 2004: Coming of Age, AI TR 04-313. Technical report, Department of Computer Sciences, UT-Austin, October 2004.

[23] S. Thrun. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[24] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, USA, 2005.

[25] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.