Vision-based Safe Local Motion on a Humanoid Robot

Xiang Li
Texas Tech University, USA
xiang.li@ttu.edu

Shiqi Zhang
Texas Tech University, USA
s.zhang@ttu.edu

Mohan Sridharan
Texas Tech University, USA
mohan.sridharan@ttu.edu

Abstract— **Humanoid soccer robots are increasingly becoming more autonomous as sophisticated approaches are being developed for challenges in vision, motion and team coordination. The stated goal of the RoboCup initiative is to beat the human soccer champion team by 2050 [1]. In order to achieve this goal, it is essential to enable the robot to fully utilize the information extracted from the available sensors. In the standard platform league of RoboCup [2], one major challenge is the ability to detect and avoid the mobile obstacles i.e. the other robots on the field. This paper presents an image gradient-based scheme to efficiently and reliably characterize the obstacles in the environment. In addition, information extracted from color images and range sensors is incorporated to build a robust obstacle model. Furthermore, a potential field-based method is used to navigate safely in the presence of obstacles. All algorithms are implemented and tested on the Aldebaran Nao [3] robot platform.**

Keywords: Visual learning, Safe navigation, Humanoid robots.

I. Introduction

The ready availability of high-fidelity sensors at moderate costs [4] has resulted in the deployment of mobile robots in real-world applications such as disaster rescue, medicine and navigation [5], [6], [7], [8]. Each sensor mounted on a mobile robot may however provide information about different regions of the robot's environment, in different formats and with different levels of uncertainty. A color camera, for instance, is a high-bandwidth source of information compared to a laser range finder. The visual input is however more noisy and the corresponding information processing algorithms are computationally expensive. As a result, many mobile robot applications base their decision-making predominantly on other sensory inputs (e.g. range information, GPS etc) [6], [8]. Such approaches that do not fully utilize the available information are likely to be at a disadvantage in dynamic environments.

There has been considerable interest in recent years on the development of humanoid robots, particularly in the robot soccer community [2] and the human-robot interaction scenarios [9]. Humanoid platforms can be deployed in practical settings where they can learn from and interact with humans. The research on humanoid robots has resulted in sophisticated methods for challenges such as motion control and dynamic balancing [10], [11]. Within the humanoid soccer setting, researchers have focused on challenges in vision, motion and team coordination [12], [13]. However, the ability to effectively use the available information is still a challenge on robots deployed in dynamic environments.

In the standard platform league of RoboCup [2], teams of three humanoid robots play a competitive game of soccer on an indoor soccer field. The dynamics of the game make it important for the robot to detect and avoid the moving obstacles i.e. other robots on the field. Unreliable communication makes it difficult to reliably avoid the teammates, making obstacle detection and safe navigation all the more challenging. Most current methods detect obstacles using the range information or color-coded image regions [12]. Such methods do not fully exploit the available information, especially the information encoded in camera images, and as a result do not result in robust performance.

Computer vision research has produced methods that use scale, orientation and affine invariant image gradient features to characterize objects in images [14]. However, these sophisticated approaches are either computationally expensive or do not provide the high reliability required for mobile robot application domains. Considerable research has also been done in the field of safe coordination and navigation on individual robots and mobile robot teams, using information from a variety of sensors [17], [18]. In parallel, sensor fusion has been studied in fields such as networks and multiagent systems [20], [21]. However, applying the existing strategies to mobile robot domains requires heuristic constraints and manual supervision [8].

This paper draws from the existing work in the related fields to make the following contributions:

- we incorporate a combination of an efficient gradient feature detector (MSER [22]) and a reliable feature descriptor (SIFT [14]) in order to characterize the target objects reliably and efficiently;
- we enable the robot to use learned error models for the processing schemes, and our existing information fusion scheme [23], to build robust obstacle models;
- we incorporate a potential field-based obstacle avoidance scheme, which uses the obstacle models to navigate safely in the presence of obstacles;

All algorithms are tested on a humanoid robot platform (Aldebaran Naos [3]) in the robot soccer framework.

The remainder of the paper is organized as follows. Section II presents the proposed approach, including the image gradient feature-based obstacle characterization, the information fusion scheme, and the proposed navigation scheme. The experimental setup and results are described in Section III, followed by a brief review of related work (Section IV) and the conclusions (Section V).

Proceedings of the 4th Workshop on Humanoid Soccer Robots
A workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009),
Paris(France), 2009, December 7-10
ISBN 978-88-95872-03-2
pp. 1-8

II. PROPOSED APPROACH

This section first describes the test platform and the challenge task addressed in this paper. Next, we describe the proposed approach for building robust obstacle models that are used for safe local navigation.

A. Test Platform and Challenge Task

The Aldebaran Nao humanoid robot platform [3] is used as the test platform in our experiments. The 58cm tall robot has 23 degrees of freedom; five in each arm and leg, two in the head, and one at the pelvis. The primary sensors are the monocular color cameras in the forehead and nose, though only one camera can be used at a time, i.e. stereo capabilities do not exist. Each camera has a 58^o diagonal field of view and a maximum resolution of 640×480; 320×240 or 160×120 images can be used for faster processing. In this paper, the 160×120 resolution images are used for experimental analysis. There are two ultrasound sensors in the chest, one each on the left and the right with a 60^o field of view. The robot also has accelerometers, bump sensors, microphones, loudspeakers, LEDs, and Wi-Fi to communicate with other robots or an off-board PC. However, all processing for vision, locomotion, localization and team coordination is to be performed in real-time (30Hz) on board the robot, using the x86 AMD GEODE 500MHz CPU that runs embedded Linux.

One application domain for the Nao is RoboCup, an international research initiative with the stated goal of creating, by the year 2050, a team of humanoid robots that can beat the champion human team in a game of soccer on an outdoor soccer field. The Standard Platform League [2] of RoboCup has teams of Naos (three per team) playing a competitive game of soccer on a $6m \times 4m$ indoor soccer field. Figure 1 shows some images of the domain.

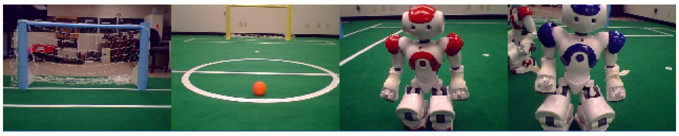

Fig. 1: The Nao [3] robot and the robot soccer field.

The robot soccer framework presents many of the challenges that need to be addressed for deploying a humanoid robot in the real-world (e.g. vision, motion, localization and team coordination), while providing a moderate amount of structure that makes the domain tractable to solutions. One significant challenge in the domain is the safe navigation in the presence of obstacles. On the robot soccer field, the other robots (opponents and teammates) are the "obstacles". Collision with other robots can cause physical damage and provide the opponents with a major advantage since the rules of the game penalize robots that collide with each other. Teammates are considered obstacles despite the Wi-Fi capability since the communication is delayed and unreliable.

B. Image Gradient-based Obstacle Detection

In recent computer vision literature, features based on local image gradients have been used extensively to characterize and hence recognize objects of interest [14], [15], [22]. These approaches are aimed at being robust to one or more factors such as scale, orientation, affine transformations, illumination and viewpoint. Typically there are two components in these approaches: a *detector* that uses second-order image gradients to extract small image regions (called *keypoints*) that are consistent across variations in the factors of interest, and a *descriptor* that associates each extracted region with a signature that identifies its appearance compactly. Objects of interest can be represented by a database of such *feature descriptors* extracted from a set of images.

Recent experimental comparison of the existing detectors and descriptors [24] has shown that the MSER (Maximally Stable Extremal Regions) detector [22] provides the most efficient performance by identifying a small set of unique regions to characterize the target objects. In addition, the SIFT descriptor [14] uses a 128-dimension feature vector to represent each of these distinctive regions, and provides the most reliable object recognition performance. The default detector for SIFT is the DoG (Difference of Gaussian) operator that is implemented in scale-space, while MSER finds elliptical covariant regions on level sets of the image. Figure 2 shows images with the keypoints detected using the MSER approach and the default DoG+SIFT technique. The images show that MSER finds fewer i.e. more distinctive image regions. In this work, we therefore use a combination of these two approaches to represent the target objects i.e. the obstacles on the field.

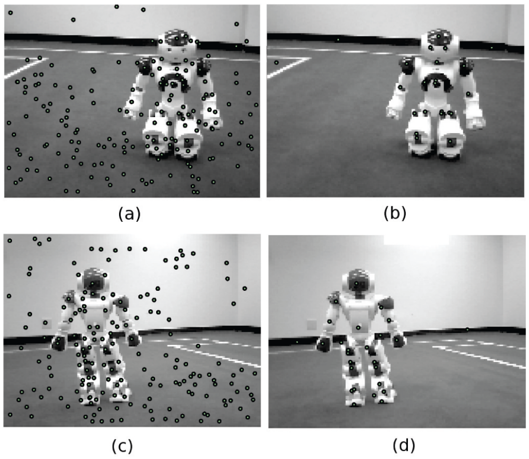

Fig. 2: Keypoints detected with DoG+SIFT: (a),(c); and MSER: (b),(d)—MSER finds more distinctive keypoints.

A DoG detector represents each detected region using four parameters: (x, y, σ, θ); (x, y) denote the location of the distinctive image region, σ represents the scale-space, and θ is the orientation. The MSER detector uses five parameters: (x, y, a, b, c); (x, y) denote the location, (a, b) are the axes of the ellipse representing the distinctive region, and c represents the ellipse's orientation. The scale space for a DoG operator is defined as:

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

It is the convolution of a variable-scale Gaussian $G(x, y; \sigma)$, with an input image $I(x, y)$. The parameter σ defines the range of the mask and hence described the range of the detector. There are two ways to transform the MSER representation to an equivalent DoG detector:

$$\sigma = \begin{cases} K \cdot \sqrt{(}a^2 + b^2) & option1 \\ \max(a, b) & option2 \end{cases} \tag{2}$$

Section III-B experimentally compares these options. However, the orientation of MSER cannot be used for DoG where θ is computed from the orientation histogram in the Gaussian smoothed image (Equation 1). The equivalent orientation in scale-space is hence computed after first computing σ as described above.

The obstacle detection proceeds as follows. In an initial training phase, the obstacle regions in input images are extracted and characterized using a small set of unique MSER features. The equivalent DoG representation of these regions is obtained and the corresponding SIFT descriptors are found to build the *training* database of the obstacles. A similar database can be built for the background i.e. environment. For any test image, an image region with a sufficient number of features similar to the features in the training database of obstacles can be labeled as the location of an obstacle. Rectangular bounding boxes are constructed around the image regions corresponding to the detected obstacles. The size of each bounding box (i.e. height and width in pixels) and its offset with respect to the image center are computed. Given the known size of the robot, this information in the image space can be used in geometric i.e. projective transformations to compute the distance and bearing of each detected obstacle relative to the robot.

C. The Overall Algorithm

The Nao robot has multiple sensors and hence multiple information processing schemes. This paper considers the following processing schemes:

1. *Ultrasound (US)*: Each ultrasound sensor computes object distance up to a maximum of 150cm. The bearing information is limited to object presence to the left and/or right.
2. *Vision-Color (VC)*: Since many objects in the domain (e.g. robots, goals) are color-coded, color segmented regions in the input images can be used to detect objects.
3. *MSER-SIFT (VM)*: Described in the previous section.

In the robot soccer scenario, each robot has a uniform consisting of regions of a specific color (red or blue) arranged in a specific pattern—see the color patches on the head, shoulder and chest of the robots in Figure 1). VC can detect obstacles by color-segmenting the image and detecting specific patterns of image regions of suitable color. Distance and bearing are computed using the same transforms used for the image gradient-based detection (i.e. VM). However, VC only works up to a distance of ≈ 2m (as against ≈ 4m with VM), and from specific viewpoints where the uniform patterns are visible. In addition, both vision-based schemes compute distance by analytically comparing the known object size and the detected size in image pixels. Noise in color segmentation or feature detection can hence introduce errors in distance computation. In terms of computational complexity, US and VC are inexpensive operations, while VM is computationally expensive to execute on the robot. Typically, heuristic constraints would be imposed on when (and how) the information from each of these sources should be used. Instead, we incorporate an instance of our existing work that uses learned error models of the individual processing schemes to robustly merge the available information [23], as summarized in Algorithm 1. The individual steps of the algorithm are described below.

Algorithm 1 Multisensor Information Merging

```
Require: : Learned models that predict the error in distance
    and bearing measurements from each information source.
Require: : Learned MSER-SIFT model of the obstacles and
    background.
  repeat
    $UpdateExistingEstimates()$
    $\{d_{us}, dir\} = CurrentObstacles_{us}()$
    $\{d_c, \theta_c\} = CurrentObstacles_{vc}()$
    $\{d_m, \theta_m\} = CurrentObstacles_{vm}()$
    $ResolveCurrentEstimates()$
    $MergeWithExistingEstimates()$
  until end of the game
```

In order to maintain obstacle estimates across several frames, each obstacle estimate is associated with a Kalman Filter [25]. The first step in Algorithm 1 is the *time-update* step of Kalman filters ($UpdateExistingEstimates()$, line 2) that adjusts the existing estimates to account for the robot's motion since the previous update. This step also removes estimates that have not been updated for some time. Each processing scheme is then used to compute the distances and bearings of the obstacles in the current image ($CurrentObstacles()$, lines 3-5). As mentioned earlier, the ultrasound sensor can only provide directional bearing (left, right or both), and the vision-based schemes provide noisy distance measurements. The next step ($ResolveCurrentEstimates()$, line 6) groups similar distance and bearing measurements provided by the processing schemes in the current frame. This grouping is based on the expected errors in the measured values (see Section III-A). For instance, if the difference between the bearings computed using VC and VM is more than the expected error in the individual measurements, these values are not grouped together. A single estimate is then obtained for the values within each group:

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

where the distance and bearing to the jth obstacle in the current frame (d^j, θ^j) are the weighted average of the values from the individual schemes ($i \in \{US, VC, VM\}$). The weights associated with the values obtained from the ith source ($w_{d,i}$ and $w_{\theta,i}$) are based on the learned models that predict the error in distance and bearing measurements. The individual and merged estimates from the current frame are matched with estimates from prior frames that are being tracked using Kalman filters. This matching is accomplished using the same grouping procedure used in line 6 above. The next step is the "measurement-update" step of Kalman filters that merges the matched estimates ($MergeWithExistingEstimates()$, line 7). It could be argued that the measured values and the predicted errors (from each processing scheme) can be input directly to the Kalman filters. However, the current measurements still need to be matched with the existing estimates. Furthermore, it would require appropriate manual tuning of the Kalman filter's noise models.

D. Robot Navigation

Once estimates have been obtained for the obstacles in the environment, the robot needs a scheme for safe local motion. This objective is achieved by incorporating a variant of the potential field method that has been used in other robot applications (e.g. [19]). Typically, such approaches overlay a potential field over the environment. The functional form of the potential field assigns values to the environmental regions covered by the field. The gradient of the potential field can hence be followed to reach a local maximum or minimum of the function. In a multirobot setting, the potential field can be posed as a sum of linear components, with each component representing the heuristic information about an individual player's behavior.

Since the goal is to achieve safe local motion in the presence of moving obstacles, we establish local potential fields centered at each obstacle estimate i.e. the estimated locations of obstacles relative to the robot. We model each potential field as a 2D Gaussian whose axes correspond to the expected error in the corresponding obstacle estimate. The potential field hence assigns a repulsive force whose value is maximum at the center of the field and decreases rapidly as we move away from the estimated location of the obstacle.

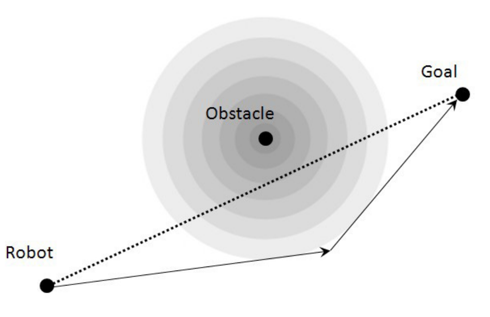

Fig. 3: Obstacle avoidance using Potential Fields.

Consider, for example, the situation shown in Figure 3. Each annular ring in Figure 3 represents a region in space with the same approximate value of the repulsive force—the darker regions close to the obstacle location represent a larger repulsive force, while the brighter regions represents locations where the repulsive force is smaller. Such a representation introduces hysteresis and prevents sudden transitions in robot motion. The robot has to move from its initial location to a target location (the "Goal" in Figure 3). In the absence of the obstacle, the robot would pursue a straight line path to the target location. However, given the presence of an obstacle along the intended path, the direction of motion of the robot is a vector sum of the intended direction and the direction of the repulsive force exerted by the obstacle. The relative strength (i.e. magnitude) of the vectors is based on the relative importance of avoiding the target and achieving the target location—these values can be tuned to provide aggressive or defensive robot behavior. When the robot or the obstacle moves (a common occurrence on the robot soccer field), the obstacle estimates are updated to account for the relative motion (based on the Kalman filter update in Algorithm 1). The updated obstacle estimates will lead to a corresponding change in the location and magnitude of the potential fields.

With multiple obstacles on the field, the robot's motion is affected by the potential fields corresponding to all the obstacles. In the current implementation, there is a fixed strong bias towards reaching the target location as soon as possible. However, this objective can also be achieved by assigning an attractive field to the robot's target location. Similar constraints can also be imposed to ensure that the robot does not walk off the soccer field. The net effect of the proposed approach is that the robot is able to move to its target location while avoiding obstacles along its path.

III. Experiment Setup and Results

This section first describes the approach to learn the required object models and error models for the information fusion scheme. Next, we briefly describe the parameter tuning performed in order to characterize obstacles based on image gradient features. We then analyze the ability of the proposed algorithms to provide safe local navigation.

A. Model Learning

In order to use Algorithm 1 to effectively estimate the relative position of the obstacles, we need models that predict the measurement errors of the individual processing schemes. Measurements with a larger expected error will have proportionately lower weights in Equation 3.

Since the robots are the obstacles in these experiments, they are placed at different known positions on the field. The robot moves through a sequence of poses (position+orientation) that it can reach with high accuracy—for instance the points on the center line of the field. For localization, the robot uses the standard procedure of color segmenting the image, detecting objects of specific colors, and feeding the measured distances and angles to a particle filtering algorithm [25], [26]. When the robot reaches each pose in the sequence, it compares the measured distance and bearing values to the obstacles against the true values. The robot uses the known positions of the obstacles to compute the "ground-truth" i.e. the true measurements. The difference between the measured and ground-truth values provides the error values. The error values are then used to train a polynomial function approximator that models the measurement error as a function of the measured distance (or bearing).

In addition to the error models, the robot needs gradient feature-based models that characterize the obstacles. While collecting the data for the error models, the robot can project the known positions of the obstacles within the field of view of the camera to the image. The MSER-SIFT features extracted from the appropriate image regions provide the training database of features that represents the obstacles (i.e. robots), and a similar database is created for the background.

A key requirement of the proposed learning scheme is the ability to localize accurately to the poses along the intended sequence. This requirement is satisfied by allowing the robot to walk slowly and make finer adjustments when it gets closer to each desired pose. Though the robot walks as fast as possible during games in order to meet the time constraints on reaching the desired pose, there are no such constraints in the initial learning phase.

B. Parameter Tuning

In order to get the best performance from the MSER-SIFT technique, certain parameters have to be tuned. In order to tune the parameters, a training set is constructed based on features extracted from 30 images each of the obstacles and the background. This set includes images of obstacles at different scales and orientations. Some of the image regions in the training set were labeled manually, while regions in images collected during the model learning phase were labeled automatically. A validation set is constructed using features extracted from a separate set of 50 images each with and without obstacles.

As mentioned in Section II-B, the transformation from the MSER representation to an equivalent DoG representation requires the computation of σ, which can be done in two ways (Equation 2). For option 1, object models learned from the training set are used to compute the classification accuracy over the validation set for various values of K. Table I reports the best classification results, which are obtained for $K = 1.3$. Table II shows the best classification performance obtained with option 2, over the same validation set—there is no parameter tuning involved in option 2. The tables are "confusion matrices" that show the true positives (obstacles classified as obstacles–$Obs|Obs$), true negatives, false positives and false negatives. The *best* classification result corresponds to the case where the robot detects obstacles accurately and does not detect any false positives. Based on this criterion and the experimental results, we conclude that option 1 (with $K = 1.3$) provides better overall performance. Hence it is used in all subsequent experiments for transforming the MSER representation into the equivalent DoG representation.

Observed Actual	*Obs*	*NObs*
Obs	92.0	8.0
NObs	12.7	87.3

TABLE I: Classification (%) with K = 1.3 in Equation 2.

Observed Actual	*Obs*	*NObs*
Obs	93.0	7.0
NObs	19.3	80.7

TABLE II: Classification (%) using max() in Equation 2.

When recognizing obstacles in the images, a nearest neighbor approach is used—features extracted in the images are compared against the training database of features. If the number of image features that match the features in the training database of obstacle features is more than a threshold, the corresponding image region is recognized as the location of an obstacle. In order to tune this threshold, the features in the training database are once again used to compute the classification accuracy over the validation set for various values of the threshold. Table III summarizes the best classification result, which is obtained for a threshold value of 5.

Observed Actual	*Obs*	*NObs*
Obs	85.0	15.0
NObs	13.0	87.0

TABLE III: Accuracy (%) when number of matched features = 5.

Figure 4 shows a pictorial representation of the (true positive) classification accuracy over the validation set as a function of the number of matched features. Similar graphs were generated for other categories in the confusion matrix, and based on these experiments, the value of this threshold was set as 5 for all subsequent experimentation.

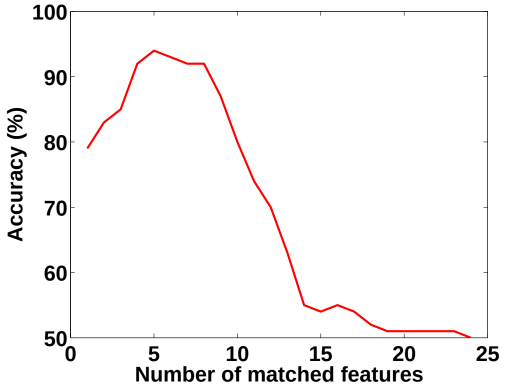

Fig. 4: Classification Accuracy vs. the number of matched features.

C. Experiment Results

Our proposed approach was designed to achieve the following: (a) to compute robust estimates of the obstacles by effectively merging the image gradient-based representation with other processing schemes based on visual and range information; and (b) to use the computed estimates in order to navigate safely in the presence of obstacles. This section describes the experiments conducted to evaluate the ability of the proposed algorithms to achieve these goals.

For evaluating the MSER-SIFT approach, we created a separate test database from a set of 300 images, which consist of 150 images with obstacles and 150 images without obstacles. Since local invariant features have been used extensively in computer vision [24], the proposed approach is compared against the default SIFT approach and a recently developed approach (FERN) that is reported to be very efficient [27]. These approaches were evaluated using the same training and test set of images used for evaluating the MSER-SIFT approach. All methods were evaluated on the basis of their accuracy and running time. Specifically, 200 images were chosen at random from the test set, and the process was repeated 10 times in order to obtain the results tabulated in Tables IV, V.

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	16.5 ± 0.3 sec

TABLE IV: Running times of different methods. MSER-SIFT takes longer than FERN during testing but is significantly faster during training.

Table IV shows that MSER-SIFT is significantly faster than default SIFT, even though the training and test databases of SIFT are pruned to remove similar features, as described in [14]. MSER-SIFT performs better because it detects a smaller set of unique features per image. The FERN [27] classifier has a tree-like structure based on simple features and is hence fast during testing. However, unlike MSER-SIFT or SIFT, FERN takes several seconds per image to model the classifier from the training set. This makes it infeasible to make incremental revisions to the FERN classifier, a key requirement for a robot working in a dynamic environment.

Method	$Obs\|Obs(\%)$	$NObs\|NObs(\%)$
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 V: Accuracy of the different techniques. MSER-SIFT provides the best performance.

Table V compares the methods in terms of their classification accuracy (true positives and true negatives). The MSER-SIFT method provides the best overall accuracy in terms of recognizing obstacles and rejecting non-obstacles correctly. Based on Table IV and Table V, we observe that MSER-SIFT provides high accuracy while still being efficient enough to allow for incremental revisions of the training database. MSER-SIFT was therefore used as the gradient feature-based detection algorithm in all subsequent experiments.

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)	38.6 ± 41.0	1.8 ± 1.5	86.1
$US + VC + VM$	9.2 ± 5.1	4.8 ± 4.1	90.4

TABLE VI: The distance and bearing errors, and the detection accuracy of the processing schemes.

Next, Table VI summarizes the distance error, bearing errors and classification accuracy of the processing schemes (US, VC, VM, US+VC+VM). Similar to the model learning phase, obstacles were placed at different locations and the robot walked through fixed poses. The errors were computed by performing 15 trials over ≈ 20 different obstacle positions where the obstacles were detected correctly. The accuracy was computed over 400 images captured during this testing process. As expected, the individual processing schemes have different properties. For instance, VM provides accurate bearings but has high distance errors with large variance because sufficient features are not found in some test cases to accurately determine the bounding rectangle. Similarly, US measures distances accurately but has low classification accuracy due to its narrow field of view. However, the proposed merging strategy (final row in Table VI) provides low errors and high classification accuracy. In order to trade-off reliability against efficiency, MSER-SIFT is run once every 10-20 frames (instead of every time-step), but the better performance justifies this trade-off.

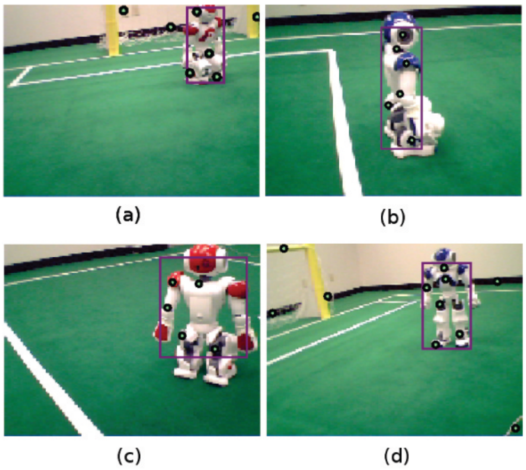

Fig. 5: Obstacle localization with MSER-SIFT—detected obstacles are enveloped in pink rectangles.

Figure 5 shows images with the detected obstacles enveloped in rectangular boxes. The key result is that visual information is exploited, and learned object representations and error models are used to merge information reliably.

Finally, we evaluated the robot's ability to navigate safely using the potential fields overlaid on the estimated obstacle locations. In order to do so, we had the robot navigate between a sequence of poses. During this motion, the robot had to tackle the presence of obstacles that were randomly placed in its path. The robot had to detect the obstacles correctly, and suitably modify its path to avoid the obstacles. Over a set of 100 such navigation attempts, the robot was able to able to navigate safely and smoothly in 95 trials. We label a trial as being successful if the robot avoids all existing obstacles and does not respond to spurious obstacles. In the trials where the robot fails, it detects a false positive (i.e. an obstacle when it does not exist) for a few frames. However, the robot soon corrects this error as it gets closer to the perceived obstacle location—there is very little observed difference in the motion of the robot. In other words, the potential field-based system works fine.

IV. Related Work

Section I discussed some existing methods for the challenges under consideration in this paper. This section briefly reviews some more related work in the fields of computer vision and robotics.

Computer vision research has provided several techniques that have used local gradient features for problems such as object recognition [28] and robot localization [29]. These approaches are based on local descriptors designed to be robust to factors such as scale, orientation, illumination and affine transformations [14], [15], [16], [22], [30]. Most of these methods are computationally expensive for robots or require significant manual training. Recent experimental evaluation of these methods [24] has identified efficient detectors (MSER [22]) and reliable feature descriptors (SIFT [14]). More recent approaches such as FERN [27] have used simpler features in order to achieve efficient operation, but they require significant training time.

In parallel to the research in computer vision, mobile robots are increasingly being used in a range of applications [6], [7], [8]. However, even state of the art robotics applications [6], [8] under-utilize the visual information. In addition, despite research on sensor fusion in related fields such as networks and multiagent systems [20], [21], most proposed techniques require manually specified heuristic constraints when used on mobile robots.

On humanoid robots, extensive research has been performed in recent years on challenges related to motion control [10], [11] and stereo vision-based navigation [31]. Research in the RoboCup framework [32] and the humanoid robots research community has resulted in several innovative techniques that have enabled humanoid robots to move towards robust and autonomous operation. However, as with other mobile robot platforms, the available information is not being fully exploited. For instance, within the standard platform league of RoboCup, teams have typically been using color-coded regions and range information to characterize obstacles [12], [13]. As a result, safe local motion in a dynamic environment in the presence of obstacles continues to be a challenge.

Obstacle avoidance is a well-researched area on mobile robots. The approach that we have used in this work, i.e. artificial potential fields, have been used before for obstacle avoidance [33] and multirobot coordination [19]. Potential fields are appealing because they use simple local knowledge about the environment—they are easy to maintain and update in dynamic environments. This technique can therefore been used to set up attractive and repulsive potentials to guide one or more robots to the desired areas of the environment.

V. Conclusions And Future Work

There has been considerable interest in recent years on the development of humanoid robots, particularly in the robot soccer community [2] and in the human-robot interaction scenarios [9]. Though humanoid soccer robots are increasingly becoming more autonomous as a result of the development of sophisticated approaches for vision, motion and team coordination, robust autonomous performance is still an unsolved problem. One major challenge in the standard platform league of RoboCup is the ability to detect and avoid the obstacles on the field (i.e. the other robots). In this paper, we present an image gradient-based scheme to efficiently and reliably characterize the obstacles in the environment. The obstacle estimates obtained from the MSER-SIFT scheme is merged with similar estimates obtained from other processing schemes, using learned models that predict the measurement errors of the individual processing schemes. Furthermore, an artificial potential field is incorporated to use the estimated obstacle locations, in order to enable the robot to achieve safe local navigation.

We have shown that the proposed MSER-SIFT approach uses a smaller set of unique features to characterize the target objects, leading to better accuracy and efficiency. Currently the robot learns the feature database of the target object only in the initial training phase. However, it is possible to enable the robot to revise the learned database and add representations for new objects such that the robot can incrementally improve its performance. Obstacles and objects that are found to be stationary can also serve as additional markers. These markers can be used by the robot to localize when the known field markers (e.g. the goals on the soccer field) are not visible.

One shortcoming of the MSER-SIFT technique is that the computational complexity is high despite our optimizations. The technique can be further optimized by only processing relevant image regions (e.g. regions below the horizon). The run-time can be further reduced by optimizing the performance of other computationally expensive modules. For instance, extended instruction sets found in the Nao processor can be exploited to significantly reduce the computational complexity of localization algorithms [34].

In this paper we have setup potential fields based on the estimates obtained by merging the information extracted from multiple sensors. We aim to extend this algorithm to a multirobot setting where the robot also uses the information communicate by its teammates. The robot can then build a representation of the environmental regions that are not within its field of view. The challenge would be to account for the uncertainty in the information communicated by the teammates. Eventually, the goal is to enable robots to autonomously learn environmental models, effectively merge information obtained from different sources, and operate robustly and safely in dynamic application domains.

References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup:The Robot World Cup Initiative," in *ICRA*, February 1997, pp. 340–347.

[2] SPL, "The robosoccer standard platform league," 2008, http://www.tzi.de/spl/.

[3] Nao, "The aldebaran nao robots," 2008, http://www.aldebaran-robotics.com/.

[4] V. Design, "Videre Design Camera," 2008, http://www.videredesign.com/stereoonachip.htm.

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

[6] DARPA, "The darpa urban robot challenge," 2007, http://www.darpa.mil/grandchallenge/index.asp.

[7] 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.

[8] S. Thrun, "Stanley: The robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[9] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[10] J. Pratt and B. Krupp, "Design of a bipedal walking robot," in *Proceedings of the SPIE*, 2008.

[11] J. Rebula, F. Canas, J. Pratt, and A. Goswami, "Learning capture points for humanoid push recovery," in *ICHR*, 2007.

[12] D. Becker, J. Brose, D. Göhring, M. Jüngel, M. Risler, and T. Röfer, "German Team 2008: The German National Robocup Team," in *Robot Soccer World Cup XII Preproceedings*, 2008.

[13] T. Hester, M. Quinlan, and P. Stone, "Ut austin villa 2008: Standing on two legs. technical report ut-ai-tr-08-8," The University of Texas at Austin, Tech. Rep., 2008.

[14] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[15] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.

[16] P. Viola and M. Jones, "Robust real-time object recognition," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.

[17] A. Murarka, M. Sridharan, and B. Kuipers, "Detecting Obstacles and Drop-offs using Stereo and Motion Cues for Safe Local Motion," in *The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[18] S. Chernova and M. Veloso, "Teaching multi-robot coordination using demonstration of communication and state sharing," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, 2008.

[19] D. Vail and M. Veloso, "Dynamic multi-robot coordination. in multi-robot systems: From swarms to intelligent automata," *Foundations and Trends in Human-Computer Interaction*, vol. II, no. 1, pp. 87–100, 2003.

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

[21] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art," *AAMAS*, vol. 11, no. 3, pp. 387–434, 2005.

[22] J. Matas, O. Chum, M.Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, 2002.

[23] M. Sridharan and X. Li, "Autonomous information fusion for robust obstacle localization on a humanoid robot," in *International Conference on Humanoid Robots*, 2009.

[24] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, USA: MIT Press, 2005.

[26] M. Sridharan, G. Kuhlmann, and P. Stone, "Practical Vision-Based Monte Carlo Localization on a Legged Robot," in *The International Conference on Robotics and Automation (ICRA)*, April 2005.

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

[28] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Simultaneous object recognition and segmentation by image exploration," in *Proc. Eighth European Conf. Computer Vision*, 2004.

[29] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *IROS*, 2002.

[30] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets," in *Proc. Seventh European Conf. Computer Vision*, 2002.

[31] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "Real-time path planning for humanoid robot navigation," in *IJCAI*, 2005.

[32] L. Iocchi, H. Matsubara, A. Weitzenfeld, and E. C. Zhou, *RoboCup-2008: Robot Soccer World Cup XII*. Berlin: Springer Verlag, 2009.

[33] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *In Proceedings of the 1985 IEEE International Conference on Robotics and Automation (ICRA-1985)*, 1985.

[34] P. Djeu, M. Quinlan, and P. Stone, "Improving particle filter performance using sse instructions," in *The IEEE/RSJ International Conference on Intelligent RObots and Systems*, October 2009.