Online structured learning for Obstacle avoidance

Adarsh Kowdle APK64@CORNELL.EDU
Cornell University

Zhaoyin Jia ZJ32@CORNELL.EDU
Cornell University

Abstract

Obstacle avoidance based on a monocular system has become a very interesting area in robotics and vision field. However, most of the current approaches work with a pre-trained model which makes an independent decision on each image captured by the camera to decide whether there is an impending collision or the robot is free to go. This approach would only work in the restrictive settings the model is trained in.

In this project, we propose a structured learning approach to obstacle avoidance which captures the temporal structure in the images captured by the camera and takes a joint decision on a group of frames (images in the video). We show through our experiments that this approach of capturing structure across the frames performs better than treating each frame independently. In addition, we propose an online learning formulation of the same algorithm. We allow the robot without a pre-trained model to explore a new environment and train an online model to achieve obstacle avoidance.

1. Introduction

Obstacle avoidance based on a monocular system has become a very interesting area in robotics and vision field (Kim et al., 2006; Odeh et al., 2009; Michels et al., 2005; Sofman et al., 2006; Chen & Liu, 2007). However, most of the current approaches work with a pre-trained model which makes an independent decision on each image captured by the camera to decide whether there is an impending collision or the robot is free to go. This approach would only work in the restrictive settings the model is trained in. For example, Michels et. al. proposed a system with a robot and a build-in camera (Michels et al., 2005). By analyzing the acquired images and training a model using the ground truth laser scan depth maps, the robot can estimate the rough depth of the scene in each frame and therefore navigate itself without collision.

In this project, there two main contributions. Firstly, the task of obstacle avoidance using images obtained using a monocular camera should not be treated as an independent decision problem working on each frame. We change this formulation to capture the structure in a group of frames and take a joint decision on these frames. We use a hidden markov model to capture this structure between the frames. Secondly, we want to relax the requirement of the labeled data to obtain a trained model for obstacle avoidance. We develop an online formulation of the structured learning approach proposed where we allow the robot without a pre-trained model to explore a new environment and train an online model to achieve obstacle avoidance. We achieve this by developing an approach to obtain positive samples (training examples of impending collision) while the robot is exploring the scene, using visual features. Once we have the training examples we can use the structured learning formulation to train an online model for obstacle avoidance. We update the model while the robot explores the environment as and when it receives more training examples.

The rest of the paper is organized as follows. We explain the feature design and visual features we use, in Section 2. We describe the dataset captured and the experiments with structured learning in Section 3 followed by the online implementation of structured learning approach in Section 4. We finally conclude in Section 5.

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 2010.

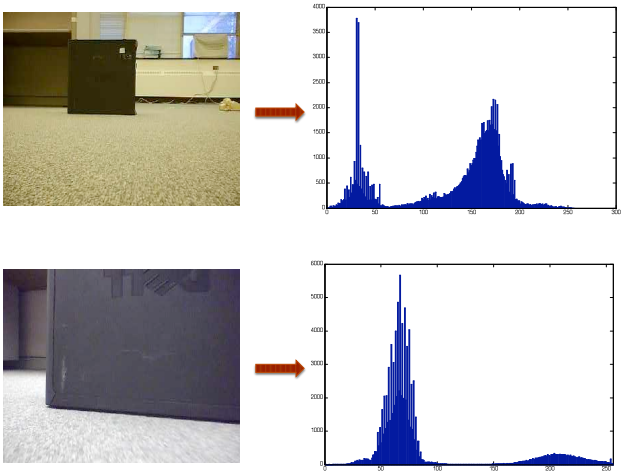


Figure 1. Feature design: Image histogram. Images on the left column and corresponding global image level histogram on the right.

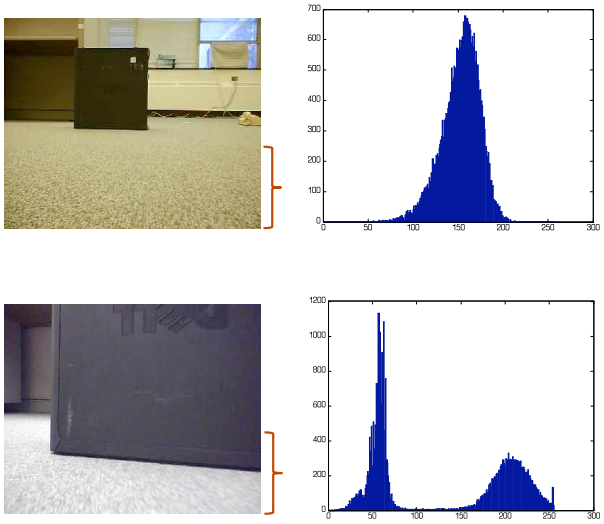


Figure 2. Feature design: Row wise color histogram. Images on the left column with a sub-block of a set of rows shown with a brace and color histogram corresponding to the braced region on the right.

2. Feature design

In this section, we describe the features we use. An intuition behind these features is that, when we consider a robot moving on the ground, an obstacle or an impending collision would be characterized by some sort of an occlusion in the lower portion of the image captured.

2.1. Histogram

We use the histogram of the pixel color intensities as a descriptor for each image. In particular, we consider two types of histograms to describe the image.

The first histogram is a global image level histogram of the pixel intensities. The intuition behind this feature is that when the robot gets very close to an obstacle the image would become occluded by the obstacle as it approaches it. This results in the feature moving from a distributed histogram to a histogram which peaks near darker intensities as shown in Figure 1.

The second histogram captures some spatial characteristics in the image. We split the image into rows of width 30 pixels. We then take a histogram over the three color channels in each of these sub-blocks. This concatenated histogram forms our second feature. This feature is motivated by the fact that the rows at the bottom of the image would have a spread out histogram unless there is an obstacle nearby. This is shown in Figure 2.

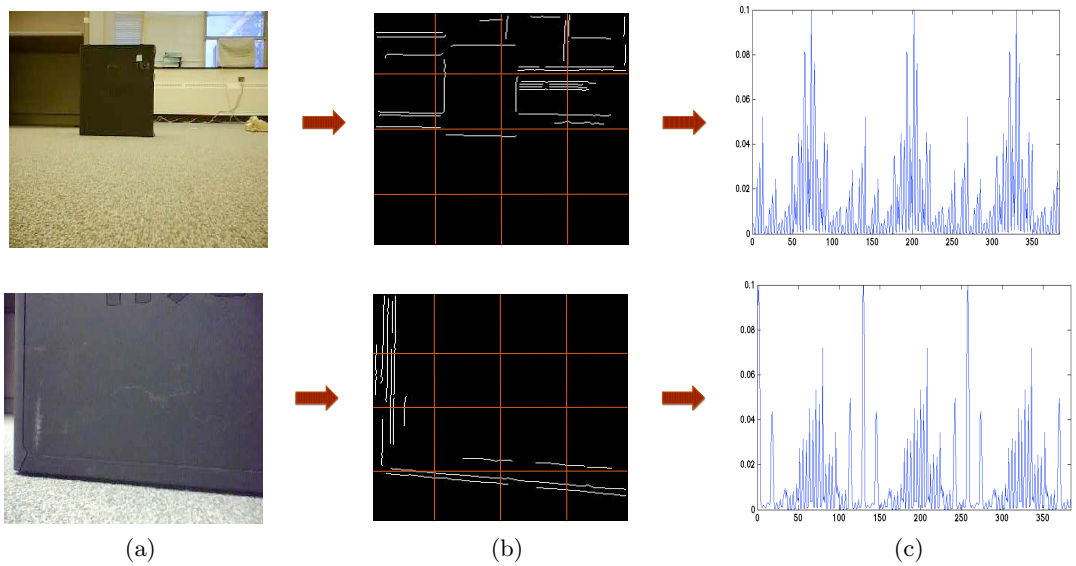
2.2. Edge gist

The edges in an image can also give a lot of information about obstacles in the path. In order to capture this in a descriptor, we use a well known descriptor called gist (Torralba et al., 2006).

We would want to describe only the dominant edges (lines) in the image and not all the finer edges (like edges which would show up on the ground). In order to achieve this, we first connect edges along the same direction together to result in a sparser set of dominant edges in the image as shown in Figure 3(b). The image is then split into 16 sub-blocks as shown. Within each sub-block an orientation histogram is used to describe the edges within the block i.e. a histogram of the orientations at each point within the sub-block. These set of 16 histograms concatenated together is what we call the edge gist. Based on the distribution of the dominant edges different bins in the descriptor dominate as seen in Figure 3(c).

3. Experiments

In this section, we first describe the dataset we use for the evaluation experiments. We first evaluate the performance of taking independent decisions on each frame treating it as a classification problem (Clear to go vs. Impending collision). We then evaluate the performance of casting this problem as a structured prediction problem using a hidden markov model and



(a) (b) (c)

Figure 3. Feature design: Edge gist (a) Some sample images; (b) The detected dominant edges in the image. The 4x4 sub-block division is shown overlayed on the image; (c) The orientation histogram from all the 16 sub-blocks concatenated together to form the edge gist descriptor.

Figure 4. Dataset: Some sample images from the 5 videos captured for the evaluation experiments.

(a) Clear to go (b) Impending collision

Figure 5. Some sample labeled frames.

taking a joint decision on the frames.

3.1. Dataset

We collected a set of 5 videos adding up to about 10,000 frames. Each frame was then manually labeled with a binary label 1/0 indicating an impending collision or clear to go respectively. Some sample frames as shown in Figure 4 with a sample labeling in Figure 5.

3.2. Binary classification using an SVM

We use the labeled data to first train a binary classifier to classify between impending collision (positive class) and clear to go (negative class). We use an SVM with a gaussian kernel to perform the classification. We concatenate all the features described in Section 2 as the feature vector in this case. We use the frames in 2 videos for training and the frames from the other 3 videos as testing and performed cross validation. The resulting accuracy in this case was $79.5 \pm 0.01\%$.

3.3. Structured prediction

The images captured from an autonomous robot are in effect highly correlated since it captures images at a steady rate while it moves around. Thus, taking an independent decision on each frame would not be the best approach.

We try to capture the structure in this output space by using a HMM (Tsochantaridis et al., 2004). We link together a group of 10 consecutive frames as shown in Figure 6 and take a joint decision on the group. We use the HMM implementation by Joachims et al. (2009)

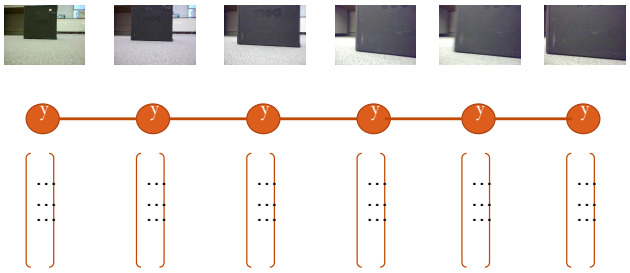


Figure 6. Structured learning: An illustration to show the linking up of consecutive frames to take a joint decision on the frames.

where the output space is again a 1/0 with a group of 10 frames linked together as one training instance.

We again use the frames in 2 videos for training and the frames from the other 3 videos as testing and performed cross validation. The resulting accuracy using the HMM was $83.5 \pm 0.01\%$ giving us a 4% increase from the binary classification. We thus implement an online formulation of the structured learning on the robot as described in the next section.

4. Online structured learning

In the previous sections, one drawback of the system would be its requirement of intensive labeling from human. However, the advantage of a mobile robot is its ability to interact with the environment. Therefore we develop an online updating scheme which requires no labeling and lets the robot develop its environment autonomously. The flowchart of the online updating system is shown in Fig.7. It could be roughly divided into two steps: training and testing. We will introduce them respectively in the following sections.

4.1. training

The training step works as an initialization of the system. During the training, we let the robot automatically label the acquired images by some simple vision features. The robot keeps going forward until it identifies that it is blocked. The blockage is defined in two ways:

1. The color distribution of the current image. If the robot is approaching into an obstacle, most likely the camera will be covered. We can calculate the color distribution to serve as an indication for blockage, i.e the color histogram. Once the camera is blocked by an obstacle, most color will be fallen into few bins, and the other bins will have very low values. Thus by counting the percentage of bins that is lower than a threshold will function as a signal for obstacle.

2. The average change of color in each pixel. There are situations that the robot encounters some small obstacles. In that case the camera is not blocked because the obstacle only stops the wheels, and thus the color distribution will not help identifying the obstacle. Then we check the average color difference between the current frame and the previous frame. If it is clear ahead, the color will keep changing because the robot is following the command of going forward. On the other hand when the robot is blocked, the color would not change much even if it tries to go forward.

Once the robot identifies the obstacle in the current frame, it labels this image as an obstacle (positive). Moreover, since it discovers the obstacle very lately by taking aggressive moving decision (keep moving forward), a few previous labels should also be considered as obstacle to avoid hitting obstacle in the future. So the system the previous 10 images as obstacles also, and then moves into another direction. When the robot can go forward, the acquired frames are labeled as clear (negative).

4.2. testing and updating the model

After the system have collected a few positive and negative samples, we are able to train an initial model for detecting the obstacle following the routine in the previous sections. Features are extracted for each frames and the model is trained by structure SVM. And then for every step, we test the model on the current frame. While the prediction is negative, i.e the path is clear, the robot will keep moving forward. Whenever the prediction becomes positive (obstacle ahead), the robot turns roughly 90^o and moves in another direction to avoid the obstacle.

In many cases the first few positive instances (obstacles) are not representative for detecting all the collision. Therefore we build an online updating model. Besides relying on the decision from the prediction of the model, we constantly evaluate the two criteria in the training step for identifying the obstacles, and record the labels. Once there is an false negative, i.e the prediction is negative (clear to go) but the previous two criteria identify that the robot is blocked, which means that the robot encounters a new obstacle. Then the current and a few previous frames are labeled as positive, and we re-train the model with all the images and labels. After this updating step, we evaluate the future images with the new model, and update the model again when the robot encounter another new obstacle. The routine can be interpreted in the loop in Fig.7.

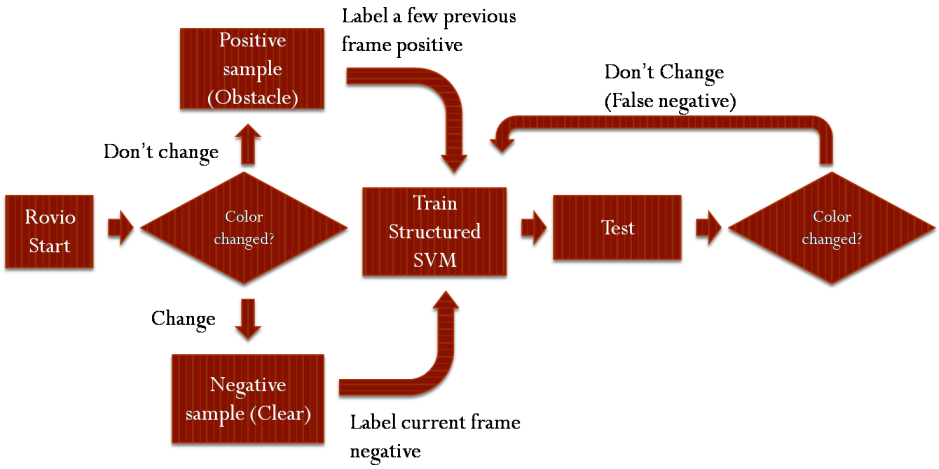
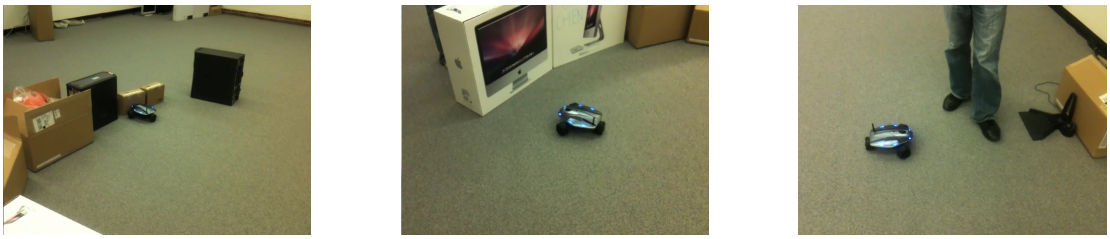


Figure 7. Flowchart of online updating the model



(a) (b) (c)

Figure 8. Experiment of online updating model in real environment. (a): training phase when the robot is collecting the initial training instances. (b) the robot can move away in distance before hitting the obstacle after training. (c) the robot can identify new obstacles and avoid hitting it.

4.3. experiment in real environment

We perform the online updating model in a real environment setting on Rovio robot. Shown in Fig.8, the environment includes some obstacles such as boxes in different colors, computer cases etc. Fig.8 (a) is the training phase of the Rovio robot. It is collecting the training instances and identifies the obstacle until the camera is blocked. After training the model with a few instances, the robot can predict the obstacle in advance before hitting it. Fig.8 (b) shows the relative position between the robot and the obstacle when the robot is about to make a turn and move into another direction. Also the robot can predict collision for some unseen objects, shown in Fig.8 (c), because the features we use are representative for general obstacles. The whole experiment is recorded as a video clip and could be viewed in the supplementary materials.

5. Conclusions and discussion

In this paper we present an autonomous robot system that can avoid obstacle based on vision only. We evaluate different features for predicting an obstacle on a single image, and select some that are useful for the task. We experiment for different classifying models. With the same features, the structure SVM can give a better performance compared to the binary classifier. We believe the main reason is that for predicting the obstacles, the collected images are not independent from each other. They are highly correlated and the chain model can well represent such relationship. Therefore the structure SVM achieves better result in quantitative evaluation.

Moreover, we build a system that requires no human labeling and allow the robot interact with the environment on its own, labeling each frames and online-updating the model. We experiment the system in a real environment setting and implement the algorithm on the Rovio robot. The experiment proves that our model works. The robot can predict collision in advance after initial training, and is even able to avoid an unseen obstacle.

One extension to the system would be multiple com-

mands/outputs after encountering an obstacle. Currently the robot will only turn a fix degree to the right and move to another different once it find itself blocked. The output y of the structure SVM is a binary value with 1 indicating the obstacle and 0 for clearance. However, multiple choice of y would be preferable, and they would be asking the robot to move left, right, or backward etc. Complicate decisions would make the robot appear more intelligent in avoiding the obstacle. And the robot can finish some complex tasks efficiently, like traveling from A place to B, rather than simply randomly exploring the environment.

Also one drawback of online updating model is the tendency of making a conservative system. Since we only update the model once it encounters a false negative, regardless of false positive, the model could become more conservative after several updates, because the safest movement would be predicting everything as positive. One solution could be setting another criteria for detecting the clear path, and updating the model when there is a false positive.

Since the robot can record the rough distance it has moved, one interesting extension would be predicting the distance in a single image. The goal is similar to (Michels et al., 2005), however in this case we no longer require laser data for estimating the depth. The robot can give the depth information through interaction with the environment. Another interesting extension would be applying the apprenticeship learning in this scenario. Although the robot is running automatically, labeling the instance by aggressively hitting the object is not favored in many situation. On the other hand, labeling each frame by human is tedious and intensive for there are usually thousands of frames in a short video clip. Apprenticeship learning is a neutral solution. We can guide the robot to move for a short period of time, and then the robot can learn policy of the human behavior. Therefore after that it can run autonomously, avoiding the obstacle without guidance.

References

Chen, H. T. and Liu, T. L. Finding familiar objects and their depth from a single image. In *ICIP*, pp. VI: 389–392, 2007.

Joachims, T., Finley, T., and Yu, C. N. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009. URL `http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html`.

Kim, Dongshin, Sun, Jie, Oh, Sang Min, Rehg, James M., and Bobick, Aaron F. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *ICRA*, pp. 518–525. IEEE, 2006.

Michels, J., Saxena, A., and Ng, A. Y. High speed obstacle avoidance using monocular vision and reinforcement learning. In Raedt, Luc De and Wrobel, Stefan (eds.), *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pp. 593–600. ACM, 2005. ISBN 1-59593-180-5.

Odeh, S., Faqeh, R., Eid, L. A., and Shamasneh, N. Vision-based obstacle avoidance of mobile robot usingquantized spatial model. 2009. ISSN 19417020.

Sofman, B., Lin, E., Bagnell, J. A., Cole, J., Vandapel, N., and Stentz, A. Improving robot navigation through self-supervised online learning. *J. Field Robotics*, 23(11-12): 1059–1075, 2006.

Torralba, A., Oliva, A., Castelhano, M. S., and Henderson, J. M. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychol Rev*, 113(4):766–786, 2006.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. 2004.