In proceedings: 27th Army Science Conference, Orlando, FL. Dec 2, 2010

FULLY ADAPTIVE VISUAL NAVIGATION FOR AUTONOMOUS VEHICLES

Marco Scoffier* and Urs Muller
Net-Scale Technologies
Morganville, NJ 07751

Yann LeCun, Pierre Sermanet, Benoit Corda and Clement Farabet
Courant Institute of Mathematical Sciences
New York University, New York, NY 10003

Abstract

Adaptive systems make for robust autonomous ground vehicle navigation able to react to changing environments and to operate in unknown areas. This paper demonstrates that adaptivity and machine learning techniques can reduce a system's sensitivity to sensor quality and calibration, enable the use of low cost and low power cameras, reduce the dependency on active sensors such as LIDAR, and simplify the process of reusing the same navigation system on different small and large hardware platforms. In particular, this paper presents an obstacle avoidance example that was solely implemented with end-to-end learning, a robust camera-based long range vision system which was integrated into a working robotics platform and employs a new technique called near-to-far learning as well as the results from countless outdoor field tests in natural environments.

1. INTRODUCTION

The goal of the Net-Scale/NYU collaboration is to create a robotic navigation system which is adaptable end-to-end. Today's systems consist of a large number of components. Each is separately optimized and has its own tolerances, strengths and weaknesses. Such systems require extensive fine tuning and are sensitive to calibration and to changes in the environment. Performance can be catastrophically affected by one failing part. A system which auto-tunes is more robust. We believe that a system with an end-to-end feedback loop and continuous automatic adaptation will bring about the next break through in robotics. Years of compartmentalized development optimizing individual modules of a system have taken us as far as we can go. We must now treat the whole system end-to-end.

Mobile robots that can navigate autonomously in a natural environment have wide ranging civilian and military applications. Current systems rely on a large number and incredible variety of sensors, followed by vast computing resources, yet their robustness is insufficient for commercial deployment and their cost, size, and power consumption is prohibitive for smaller vehicles. A key deficiency in current systems is that they rarely respond intelligently to the unexpected, and often do not adapt to new situations and new environments. To function reliably, mobile robots that encounter a new class of obstacle should adapt quickly so as to perform early detection and avoidance of similar obstacles in the future. Our teams participation in the LAGR program demonstrated advances in large-scale autonomous machine learning methods applied to visual recognition and control. In particular, we showed that the combination of convolutional networks, energy-based unsupervised learning, and on-line self-supervised adaptation holds great promises for highly robust and adaptive perception and control systems for mobile robots.

1.1 Low Cost Passive Sensing

Fully autonomous navigation in variable outdoor environments is the long-term goal of the Net-Scale/NYU collaboration. Furthermore we focus our efforts to create technology which can work both with active sensors, such as LIDAR, and passive sensors, such as regular video cameras. Processing of passive sensors, in particular vision sensors, is a harder problem compared to the use of active sensors but passive sensors have a number of important advantages: they require less power, are more difficult to detect, and are typically less expensive than active sensors. We believe the future will move towards passive low cost sensing and smart computation, rather than today's common combination of active expensive sensing and minimal computation.

Lower in power and cost, more difficult to detect, the advantages of passive systems are numerous but the computational difficulties of extracting meaningful information from simple vision sensors have hampered the deployment of vision based systems in the field. Within the LAGR program we made significant progress towards resolving the computational hurdles, enough that there has been interest in using elements of our technology in other systems which do not require full autonomy: to handle the short range obstacle avoidance for a remote operator over a noisy link or to drive a system back to a home base should the human driver have to abandon a piece of equipment. We hope to use a future STTR grant to move elements of our technology into the marketplace. There are many scenarios in the marketplace which are a perfect fit for our technology. This includes scenarios that require full autonomy, as well as partial autonomy.

1.2 Use Case Scenarios

A robust navigation system has many practical uses. Following are some typical scenarios of how the military could use such systems in the field. Tele-operation is difficult especially over noisy communication links. We have a system which can take over the low-level driving commands from a human operator allowing the human to perform higher level tasks.

Single operator surveillance. Currently human must tele-operate mobile robots such as the Talon series (Foster Miller, 2010) or the Packbot (iRobot, 2010). Driving the robot to its goal takes the full attention of a skilled human operator. With a robot capable of semi-autonomous navigation, the operator can set a goal for the robot which moves itself into position only requiring the operator's attention when it encounters a problem. The semi-autonomous navigation system handles all the low level driving and even allows a single operator to control multiple vehicles. For a surveillance task, a single operator can simply point out numerous targets on an image or a map and send multiple vehicles in parallel to move to those target locations autonomously. The operator can then concentrate on the actual mission, getting data from the multiple strategically placed mobile sensor platforms, for example. To retrieve all the vehicles would be as simple as pressing a come home button.

Air-land rescue In many operational scenarios, because of rough terrain or danger from ground fire, it is impossible to land an air craft precisely where it is needed. A ground based robot must complete the mission on land. For example, the vehicle must drive to a wounded soldier to deliver medical supplies, or move into a dangerous area to provide surveillance. A semi-autonomous navigation system can integrate a map recorded from the flight or satellite images with the actual images recorded from the ground. The ability to adapt allows our semi-autonomous navigation to operate in a radically changed landscape such as after a natural disaster or during a military operation.

2. OVERVIEW

In this paper we present the essential elements for a robust and adaptable system capable of autonomous navigation and planning in heterogeneous natural outdoor environments. Originally created within the DARPA funded LAGR program (Learning Applied to Ground Robots)(DARPA IPTO, 2005-2008) we are currently moving the system to a modular platform adaptable to a variety of robotic hardware platforms (this work was partially supported through an Army STTR program). Within the DARPA LAGR program, our system underwent monthly independent government tests (Jackel et al., 2006).

There are three sections. In the first we present an earlier prototype DAVE which was much simpler than the LAGR system but is an example of end-to-end learning: No parameters are hand turned. The system learns to avoid obstacles directly from video input. In the second section we present LAGR's near-to-far learning which extends the range of the passive visual sensors from less than 10m using the stereo system to 75m and beyond. Seeing further allows for much faster driving speeds. Our system completed an unknown outdoor obstacle course 3 times faster compared to the state of the art system used by DARPA for testing. In the final section we describe our recent work about adding a type of memory to the system to make it more suited to longer missions. Specifically we present the use of multiple experts to solve the "fast learning fast forgetting" problem.

2.1 DAVE

We built DAVE (DARPA Autonomous VEhicle), an early prototype, for DARPA under a specific mandate: to provide evidence that end-to-end learning works for autonomous ground vehicle navigation. As a proof of concept nothing was hand tuned. Pixels enter directly into a learning machine and steering directions are the output. After training on hours of examples of human driving the system masters an obstacle course better and faster than a human with a joystick.

In the DAVE project, Yann LeCun in collaboration with Net-Scale implemented the first use of convolutional nets for robot vision, and the first demonstration of *end-to-end learning* for vision-based obstacle avoidance. We built the reactive obstacle avoidance system around a convolutional network and trained to map raw pixel images from two video cameras directly to desired steering angles provided by a human driver (LeCun et al., 2005). The project report and a video clip are available at (Net-Scale Technologies, Inc., 2010). The results from the DAVE project led to the DARPA LAGR program.

2.2 LAGR: Near-to-far

The goal of the LAGR program was to develop a new generation of learned perception and control algorithms for autonomous ground vehicles, and to integrate these learned algorithms with a highly capable robotic ground vehicle. Eight teams participated in the program which went through two phases and lasted from 2005 - 2008. (DARPA IPTO, 2005-2008)

By combining long-range perception with learned behavior, LAGR was setup to make a qualitative break with the myopic, brittle behavior that characterizes most UGV autonomous navigation in unstructured environments. To overcome the difficulty of making an accurate, objective measure of performance in off-road environments, the Government Team managing the field tests created a relative measure: it tested the navigation software by comparing its effectiveness to that of a state-of-the-art, navigation software running on a standardized vehicle on a series of varied test courses. (Jackel et al., 2006)

The Net-Scale/NYU LAGR entry used one of the most ambitious, innovative, and unconventional approaches to perception and control. It was the entry that most heavily relied on learning and adaptation for its perceptual and control modules. See Figures (3,6,5)

The Net-Scale/NYU LAGR system demonstrated that it could accurately detect obstacle and traversable areas at long range, and that it could adapt to every kind of natural environment. Like DAVE the main computational network in the Net-Scale/NYU LAGR system uses convolutional networks. The composition of multiple stages that extract features that are increasingly abstract, increasingly global, and increasingly invariant to irrelevant variabilities in the input are the defining characteristic of the convolutional network architecture. An interesting consequence is that each obstacle/non-obstacle decision takes a large contextual window into account. By learning to reconstruct unlabeled samples from video log files, the convolutional network learns to extract useful features extracted in unsupervised fashion. We do no expensive labeling and train the system off-line on hours of recorded log files.

When the robot operates in a particular environment it trains a simple classifier on the output of the convolutional feature extractor. The classifier gets its labels for nearby objects from the stereo system which categorizes obstacles based not on their visual appearance but purely on the geometry of the point cloud generated by the stereo disparity map. The combination of unsupervised pre-training, supervised off-line learning, and on-line real-time adaptation proved to be effective. See Figures (3,7,8).

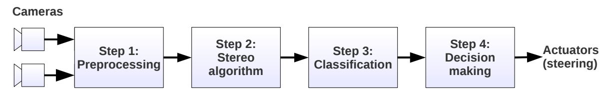
2.3 Multiple Experts

Guided by the field experience we gained during the LAGR program, the Net-Scale/NYU team has developed a plan to create a navigation platform. Partially funded through an Army STTR program we added several elements to the existing code base. In the final section of this paper we present the Mixture of Experts Classifier which solves a problem we named "fast learning - fast forgetting". The problem does not exist in short test runs, but arises on longer "mission length" runs of several hundred meters when the robot traverses multiple visual environments. With a single classifier, as we had for the short runs of the LAGR program, the robot learns to navigate in one environment. This learning replaces any learning of different environments which it might have seen in the past. In the worst case the robot would have to revisit a previously explored area in order to re-learn how to navigate in that environment. We implemented a simple pool of multiple classifiers, and a system which selects the classifier for the current environment. This allows the system to maintain a memory of previously learned environments, and learn over different time scales.

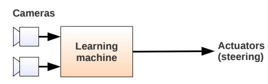
3. TECHNICAL DETAILS

3.1 DAVE: Obstacle Avoidance With End-to-End Learning

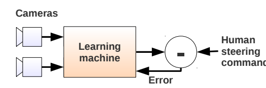
As an initial exploration into the potential of machine learning for autonomous ground vehicle navigation for DARPA we built a vision-based obstacle avoidance system for off-road mobile robots (LeCun et al., 2005; Net-Scale Technologies, Inc., 2010). The system consists of a single unified learning machine which is self-trained to map raw camera pixels directly to steering angles. No parts are manually designed nor hand tuned. We collected training data by a human driver to navigate the vehicle in a wide variety of terrains, weather conditions, lighting conditions, and obstacle types. The robot is a 50cm off-road truck, with two forward pointing low-cost web-cams (no high-end sensors are necessary). After training, the learning machine can detect obstacles and navigate around them in real time at speeds of 2 m/s. In fact, it outperformed all human drivers



(a) Typical processing steps of a traditional navigation system. The preprocessing step often includes noise filtering, image rectification and contrast normalization. The stereo algorithm calculates the disparity between left and right camera for small image patches to reconstruct a 3-D point cloud as accurately as possible. The classification step analyzes the point cloud to classify the terrain into traversable and non-traversable areas. This information is then used by the decision making step whether to steer left, right, or go straight.



(b) In contrast to the complicated traditions system, DAVE uses a single unified learning machine which maps raw camera pixels directly to steering angles.



(c) During training the unified learning machine receives steering commands from a human operator. Learning to map raw pixels to steering commands.

Figure 1: Comparison of DAVE's end-to-end learning machine to a traditional navigation system

who tried to steer the truck manually both in terms of speed and number of crashes with obstacles. We named the system DAVE, for DARPA Autonomous VEhicle.

In comparison, Figure 1(a) shows the typical processing steps of a traditional navigation system where traditional teams design and optimize each module individually. For example, a noise filter in the preprocessing step may try to eliminate as much noise from the image as possible regardless of which type of noise is actually detrimental to the following processing steps and which isn't. Likewise, the stereo algorithm will attempt to reconstruct a 3-D image as accurately as possible and therefore make itself sensitive to camera calibration even though the following classification step may be able to cope with inaccuracy if it was adapted accordingly.

Figure 1(b) illustrates the single unified learning machine used by DAVE.

Figure 1(c) shows the unified learning machine during training. The learning architecture is a 6-layer convolutional network (LeCun et al., 1998) with 3 million connections and 72,000 independent parameters. We collected training data by recording the actions of a human driver together with the video data. The human driver remotely drives the robot straight ahead until the robot encounters a non-traversable obstacle, then avoids the obstacle by steering the robot in the appropriate direction. We use the data at a resolution of 320×240 pixels at 15 frames per second. We collected a total of 1,500 clips on 17 different days during the Winter of 2003/2004 (the sun was low on the horizon). This resulted in a total of about 127,000 individual pairs of frames. No manual data cleaning took place. We use 95,000 frame pairs for training and 32,000 for validation/testing. The training pairs and testing pairs came from different sequences (and often different locations).

The approach is somewhat reminiscent of the ALVINN and MANIAC systems (Pomerleau, 1993; Jochem & Pomerleau, 1997). The main differences with ALVINN are: (1) our system uses stereo cameras; (2) we train it for off-road obstacle avoidance rather than road following; (3) Our train-able system uses a convolutional network rather than a traditional fully-connected neural net.

Figure 2: DAVE: Snapshots from the left camera while the robots drives itself through various environments. The black bar beneath each image indicates the steering angle produced by the system.

Figure 2 shows snapshots from the robot driving in different environments along with the steering output generated by the neural network. One can view further information and download video clips at (Net-Scale Technologies, Inc., 2010).

The main motivation for the use of end-to-end learning is to eliminate the need for hand-crafted heuristics. We train the whole system on the raw pixel input to produce the desired training angles directly. We eliminate the need for feature design and selection, geometry, camera calibration, and hand-tuning of parameters. Relying on automatic global optimization of an objective function from massive amounts of data produces systems that are more robust to the unpredictable variability of the real world. Another potential benefit of a pure learning-based approach is that the system can use cues other than stereo disparity to detect obstacles, possibly alleviating the short-sightedness of methods based purely on stereo matching.

3.2 LAGR: Near-to-far Learning

One of the key successes of the Net-Scale/NYU LAGR program was the creation, demonstration and independent validation of the near-to-far learning system. The system is capable of learning the visual appearance of nearby obstacles, and extending the recognition of obstacles well beyond the range of the stereo vision system. It creates navigable maps of obstacles up to 75m away from the mobile robot greatly increasing the long range planning capabilities and corrects the myopic behavior of the mobile robot which relies solely on 7 to 10m range of the stereo system. See Figure 7.

Near-to-far learning uses a dense stereo system to gather labels based on geometry and bootstrap a visual classifier. The stereo system generates a 3d point cloud. We find a ground-plane in this point-cloud. We label points above the ground-plane as obstacles. We project the points above the plane into the plane to create a traversability map for the close range navigation. The labels derived from the geometry of the 3D stereo point-cloud identify points in the original image as looking like obstacle or looking like traversable. We extract image patches around these labels and have the image data and labels which we use to train a classifier. We train this classifier on-line as the robot drives constantly adapting to different environments.

This technique is the essential element in making the system adaptable to different environments. A stereo vision system classifies obstacles and traversable areas based on the 3 dimensional structure which is independent of their visual aspect. In parallel a convolutions neural network extracts a distinctive feature vector for every 6x13 pixel patch in the image.

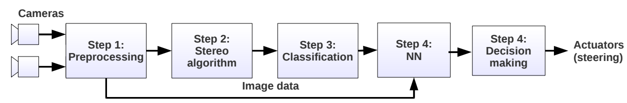


Figure 4: **Net-Scale/NYU LAGR: Near-to-far flow-chart** The near-to-far learning uses labels from a classical stereo algorithm to boot-strap the learning of a neural network which provides long range vision. The stereo provides training labels for image patches in the robot's stereo range.

We use the labels from the stereo classifier as a supervised signal to train a classifier to discriminate the visual differences in the form of the extracted feature vectors between obstacle and traversable areas of the environment. This classification generalizes to areas well beyond the stereo range of approximately 10m allowing the robotic system to "see" obstacles as far away as 75 or 100m.

Two articles in the Journal of Field Robotics describe the final LAGR system in full detail ref. (Hadsell et al., 2009; Sermanet et al., 2009). Several conference publications focus on particular elements of the system ref. (Hadsell et al., 2008; Sermanet et al., 2008a; Sermanet et al., 2008b).

3.3 Multiple Experts

From long field experience with the online classifier of the Near-to-Far learner, which we described above, we discovered that the "fast learning" which is so crucial to the adaptivity of the system also would result in "fast forgetting" of environments. This "fast forgetting" would cause problems observable only on longer range excursions. The system would cease to recognize environments which it had previously learned to classify, because the information from the new environment replaced that from previously learned environments. For example, after a drive in the forest where the system learns to make the fine discrimination between the texture of a dirt path with leaves on it and the thick leaves to the side of the path, the system would poorly classify short vs. long grass in an open field which it had previously perfectly discriminated. The solution was to implement a simple memory, to continue to allow the system to adapt quickly as that had already proven to be one of it's biggest strengths, but to cache sets of weights which had been previously learned. The system recalls the cached weights if it sees the same environment again.

We call this a multi-expert system. Instead of a single on-line classifier, we now have as many classifiers as we

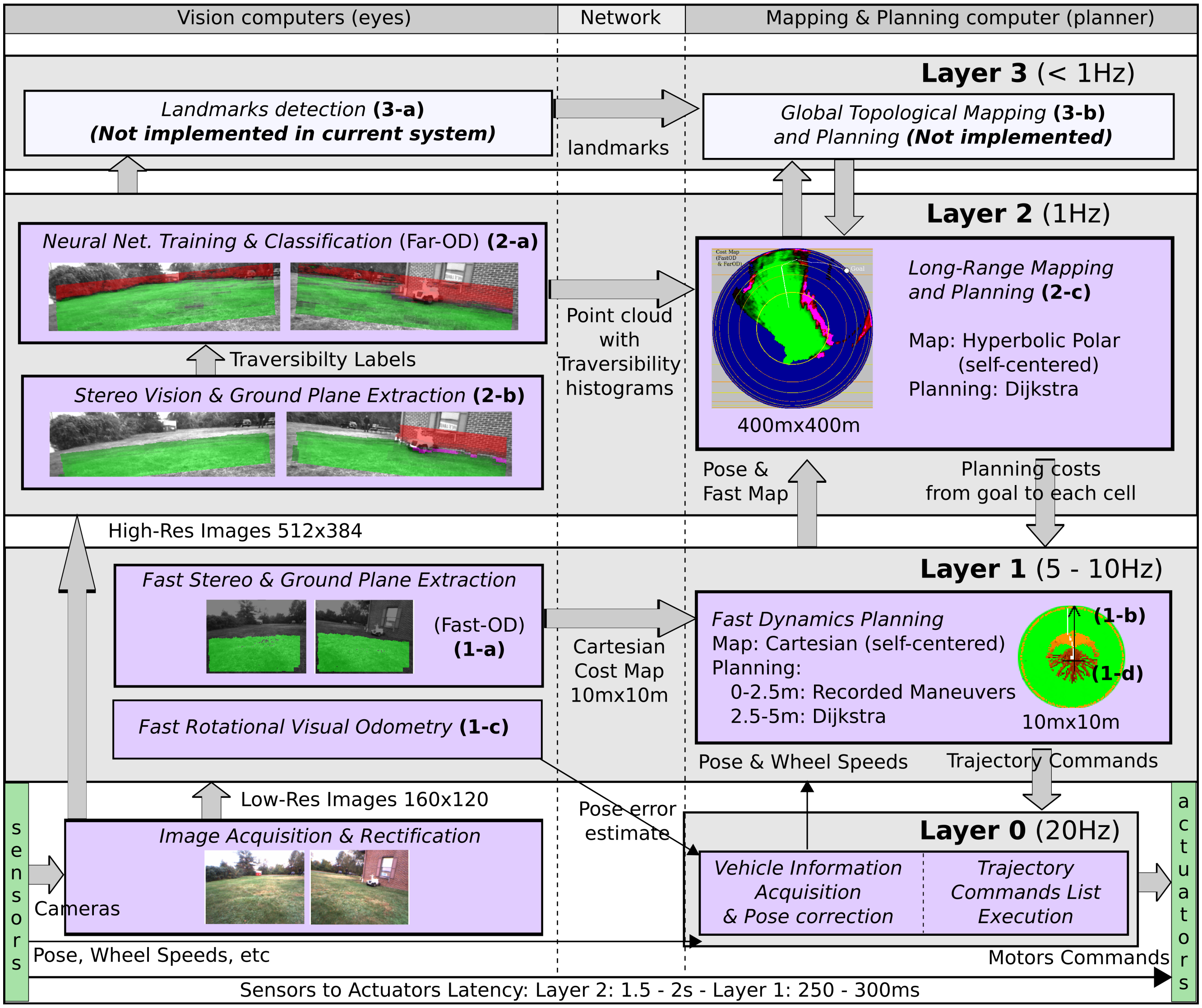


Figure 3: **Net-Scale/NYU LAGR : Overall system architecture.** Each layer contains perception, mapping and planning modules (except for layer 0). From bottom to top, sensors-actuators latency and period increase while resolutions decrease. Layers 1 and 2 are self-centered while layer 3 is global. Layer 0, also known as control loop, guarantees the flow of motor commands required at 20Hz. We have added Layer 3 since the LAGR program. Missing from the diagram is a feedback loop from Layer 3 which adds new capabilities including training using previously seen labels, map registration for additional pose stability and an environment signal for selecting location specific experts.

like. We have a set of default weights which we hold fixed. Previously we trained them off-line on hours of log footage to perform best on average over many highly varied environments. The current algorithm for initializing the experts starts them all at these default weights. We train the first expert on incoming image patches,exactly as we used to train the single on-line learner. It learns to discriminate between image patches from the environment in which the robot is moving.

The first frame in which the current expert does less well than the fixed default weights is the switch for changing to a new expert drawn from the pool of initialized experts. We store the first expert for later retrieval. Upon processing each frame, each expert classifies the patches for which there are labels from the stereo system. We then use the best performing expert to classify the whole image. The patches with labels from the stereo system get added to the ring buffer which we use to train that classifier. We train only the best expert on these new patches. This system creates a set of experts which differentiate themselves by their ability to classify different environments.

We are working on more intelligent switches for the experts beyond the current classification of a whole image in the near range. We will soon have a system which can classify different areas of an image with different experts. We want to use an expert trained in the woods to recognize the woods with the “woods” classifier when it is far away in the field. This system is hard to test as it requires full hand

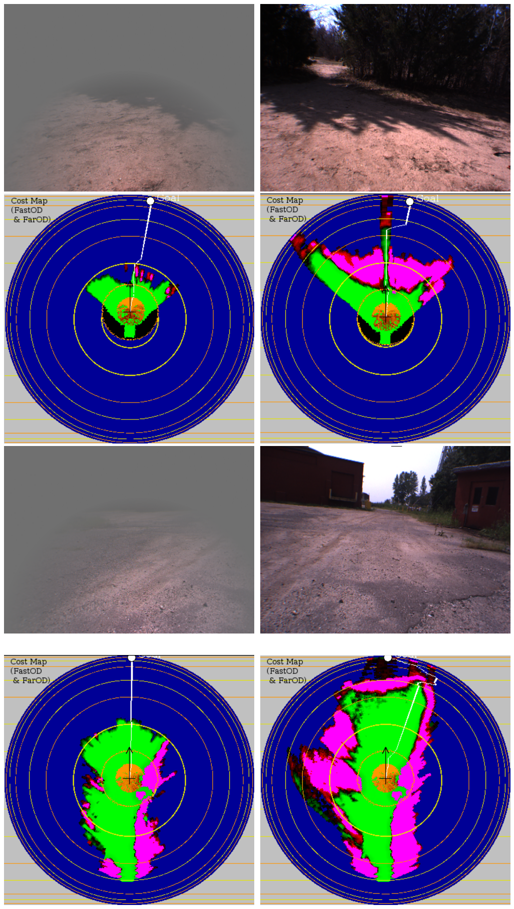


Figure 5: **Net-Scale/NYU LAGR : H-Polar Map** Two examples of Left: short range (stereo) obstacle map, and Right: the long range map. Above each example are the input images to the system. We overlay the images on the left with a grey fog which represents the distance to which the system can "see" using only the short-range stereo system. The online learning system on the right learns to classify the long-range obstacles in different environments.

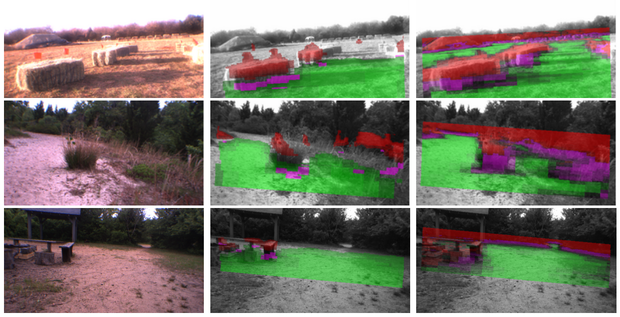


Figure 6: **Net-Scale/NYU LAGR :** Qualitative examples of the success of the long-range classifier in different terrain. **Left**: RGB input; **middle**: training labels; **right**: classifier output. Green is traversable, red is obstacle, and pink is foot-line. Notice that the classifier output in the far right column allows the navigation system to 'see' much further.

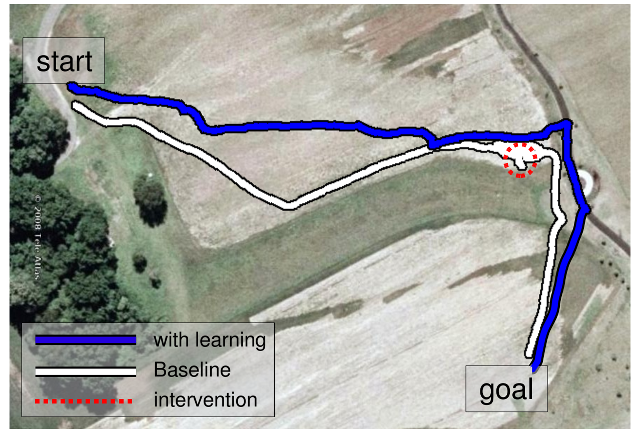


Figure 7: **Net-Scale/NYU LAGR : Aerial plots of comparison runs of long-range and Baseline systems.** This aerial view is approximately 300 meters wide. The long-range system in blue identified the large tall-grass area from the beginning and stays out of trouble while the Baseline system got close to the grass and got into trouble.

labeling of video log data. In the next section we present a test we set up to show the effectiveness of the current Multi-expert system.

3.3.1 Testing the Multiple Experts

We created a test using a store of 16 log files to emulate the situations in which environments change dramatically. Running the multi-expert system we would switch log files every 15 frames.

On average over 6000 frames in 16 different log files the multi-expert system outperforms the default weights by 7% and outperforms the on-line-weights by 9%. The on-line adaptive classifier performs poorly in this setup because we were specifically looking for a regime to confuse it. If we stay in a single environment the single classifier can train properly and match the performance of the current expert. The problems with a single learner become obvious when we change environments. A new expert created for this environment can train quickly because it is learning only on images patches from this environment. A previously trained expert which already performs well in this environment beats the single on-line learner because it is adapted specifically for that environment.

We present two cases which demonstrate the advantage of having the online learning system. For comparison we plot the classification error from three different systems. First we plot the default weights in black – which we trained on hundreds of hours of log files to perform as well as possible on average across all environments. These are the weights which we use to initialize the other two systems. They are static and don't adapt to new environments, but represent the best we have been able to train across all environments.

Next we plot the online weights in gray. We initialize the online weights to the default weights, each new frame causes and update of these weights. The stereo system labels image patches. We use these labels are to train the

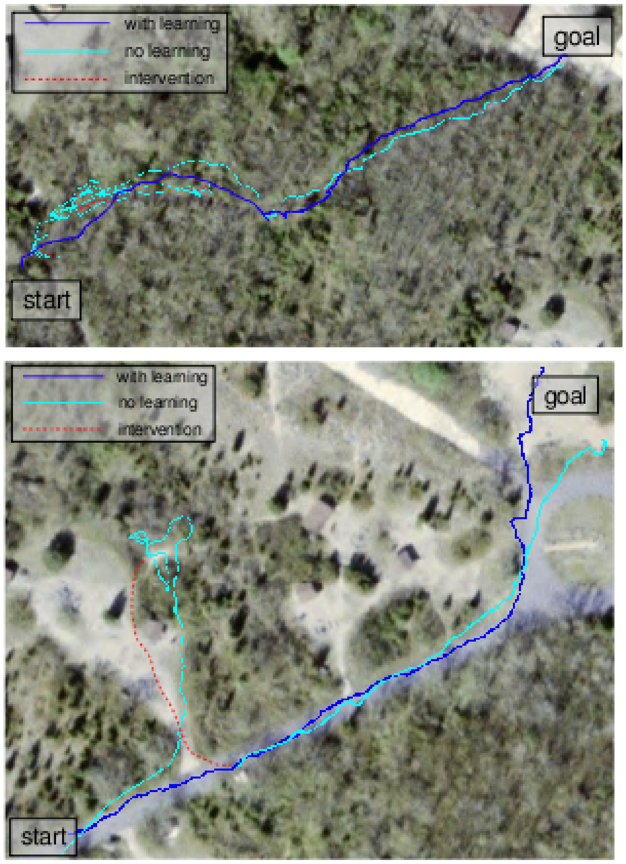


Figure 8: **Net-Scale/NYU LAGR : Aerial plots of comparison runs of long-range and Baseline systems.** This aerial views is are approximately 300 meters wide. In the top image the robot travels down a long forest path. Because the path gives strong visual cues in the near-range as to the direction to take, we expected the performance to be similar between the Baseline and the long range systems. But even here the long range vision adds stability to the planning making the robots path much more direct and much less tendency to get stuck in the brush on either side of the path. In the lower image the robot travels along a road and has the possibility of going into a cul-de-sac when using only the short range system can appear to be a more direct path to the goal. The long-range system sees to the end of the cul-de-sac and correctly does not enter, and thus makes it to the goal more quickly, more directly and does not require any manual intervention.

classifier. As the stream of new patches can be variable and there are often imbalances in the frequency of patches received for each different class (traversable, obstacle, etc.) we maintain a ring buffer of recently seen patches on which we train the classifier.

Finally we plot the performance of the new multi-expert system. Here we use a different color for each expert. The multi-expert system is made up of multiple online classifiers in parallel. Each has it's own ring buffer of recently seen patches for the environment on which we have asked it to specialize.

The plots have the frame number on the x-axis and the error rate (low is better) on the y-axis. To help keep track of the variety of environments seen, above each plot we have put two views of a single frame from the 15 most recently shown to the classifier. We overlay the labels from the stereo system on the top image of each frame, the lower image shows the output of the multi-expert classifier. Red is obstacle. Green is traversable. Purple is the baseline of an obstacle, which is most useful when planning in the map. When there are ample labeled windows coming from the stereo system, the single on-line classifier adapts to a new environment quickly sometimes in as little as one frame. When there are few labels coming from stereo, the system adapts much more slowly because there are no new patches on which to train.

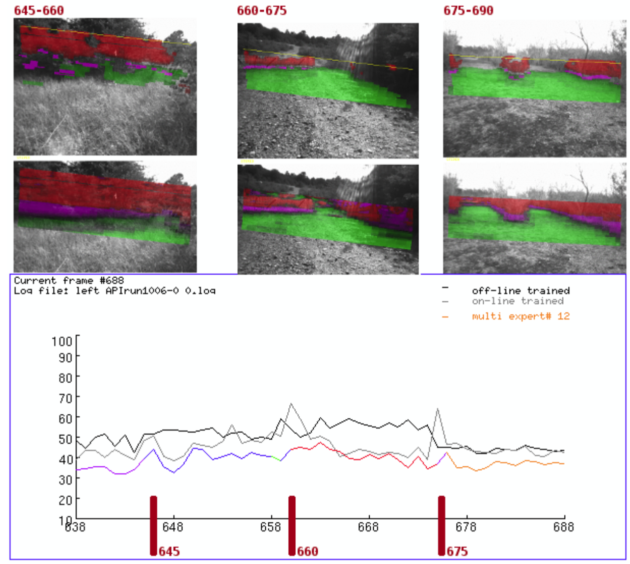


Figure 9: **Multi-expert example 1** Here we see the test system switch between three environments which the system has seen multiple times and to which it is already well adapted. We place images of the different environments above the graph. The first environment (frames 645-660) has high grass and a forest. The second (frames 660-675) has a rocky path with red earth, and orange construction fencing. The third (frames 675-690) is a network of paths mowed through the dry brush at SWRI in Texas. The default weights (black) and the on-line weights (gray) do pretty well. Though you can see a spike in the error rate of the online learner each time the environment changes. This spike is the time needed for the online learner to retrain for this new environment. We show each expert in a different color. For each environment an expert is automatically chosen which performs better than the either the default or the single on-line classifier. The experts shown as different colors stay stable through each environment, blue then red then orange, showing good separation and specialization among the experts.

We completed this work with partial funding from an Army STTR grant and is fully described in the Final Technical Report available at (Net-Scale Technologies, Inc., 2010)

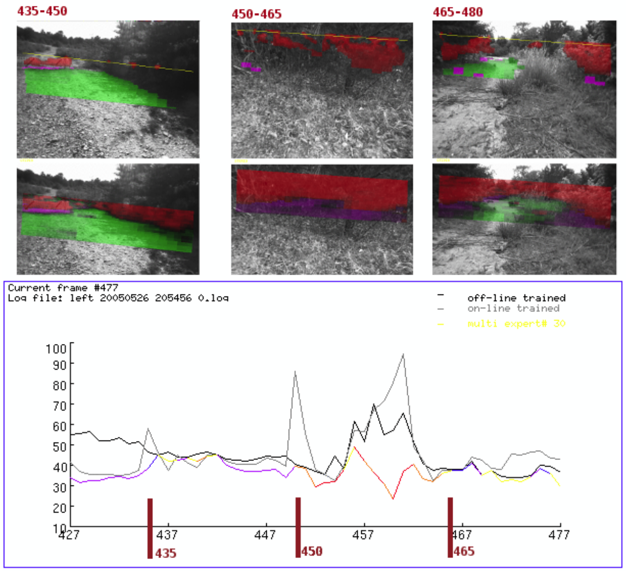


Figure 10: **Multi-expert example 2** Again we see 15 frames at a time from three different environments. We wish to highlight that in the middle environment (frames 450-465) the default weights and the on-line weights perform poorly. This is a difficult situation, the robot is facing directly into some close range brush. The stereo is performing poorly giving few labels. The on-line classifier gets few new labels and thus cannot adapt as the batch of training samples on which it adapts is full of samples from other environments. The multiple experts are having trouble, they are switching between experts, but still far out perform the other two systems.

4. CONCLUSION

We have presented practical examples and field test results for applying end-to-end adaptation to autonomous off-road navigation systems. We have further described a simple end-to-end system and two key elements from a larger system whose performance we evaluate not on any single module but on the overall system performance. Based on these results and our experience, we are convinced that designing systems with adaptation at every level where no single faulty module can catastrophically affect the overall mission, will lead to more powerful and robust solutions and will increase the number of areas where we can apply autonomous robotics technology.

5. ACKNOWLEDGEMENTS

We completed the above presented results with support from the Defense Advanced Research Project Agency Information Processing Technology Office (DARPA/IPTO) and from the US Army Tank-Automotive Research Development & Engineering Center (TARDEC). The authors would like to express their sincere gratitude towards these agencies for their support and for making this work possible.

References

DARPA IPTO (2005-2008) Web Site for DARPA LAGR Program, URL `http://www.darpa.mil/ipto/programs/lagr/lagr.asp`.

Foster Miller (2010) Web Site for Talon Robots, URL `http://foster-miller.qinetiq-na.com/lemming.htm`.

Hadsell, Raia, Ayse Erkan, Pierre Sermanet, Marco Scoffier, Urs Muller, & Yann LeCun (2008) Deep Belief Net Learning in a Long-Range Vision System for Autonomous Off-Road Driving. In *Proc. Intelligent Robots and Systems (IROS'08)*.

Hadsell, Raia, Pierre Sermanet, Marco Scoffier, Ayse Erkan, Koray Kavackuoglu, Urs Muller, & Yann LeCun (2009) Learning Long-Range Vision for Autonomous Off-Road Driving. *Journal of Field Robotics* **26**(2): 120–144.

iRobot (2010) Web Site for PackBot Ground Robots, URL `http://www.irobot.com/gi/ground/`.

Jackel, L. D., E. Krotkov, M. Perschbacher, J. Pippine, & C. Sullivan (2006) The DARPA LAGR program: Goals, challenges, methodology, and phase I results. *Journal of Field Robotics* **23**(11-12): 945–973.

Jochem, Todd & Dean Pomerleau (1997) Vision-Based Neural Network Road and Intersection Detection. In *In Intelligent Unmanned Ground Vehicles*, Charles Thorpe Martial H. Hebert & Anthony Stentz, eds., Kluwer Academic Publishers.

LeCun, Y., L. Bottou, Y. Bengio, & P. Haffner (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE* **86**(11): 2278–2324.

LeCun, Y., U. Muller, J. Ben, E. Cosatto, & B. Flepp (2005) Off-Road Obstacle Avoidance through End-to-End Learning. In *Advances in Neural Information Processing Systems (NIPS 2005)*, MIT Press.

Net-Scale Technologies, Inc. (2010) Supplemental Web Site for Paper: Fully Adaptive Visual Navigation For Autonomous Vehicles, URL `http://www.net-scale.com/papers/armyscience2010/`.

Pomerleau, Dean A. (1993) Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving. In *Robot Learning*, J. Connell & S. Mahadevan, eds., Kluwer Academic Publishing.

Sermanet, Pierre, Raia Hadsell, Marco Scoffier, Matt Grimes, Jan Ben, Ayse Erkan, Chris Crudele, Urs Muller, & Yann LeCun (2009) A Multi-Range Architecture for Collision-Free Off-Road Robot Navigation. *Journal of Field Robotics* **26**(1): 58–87.

Sermanet, Pierre, Raia Hadsell, Marco Scoffier, Urs Muller, & Yann LeCun (2008a) Mapping and Planning under Uncertainty in Mobile Robots with Long-Range Perception. In *Proc. Intelligent Robots and Systems (IROS'08)*.

Sermanet, Pierre, Marco Scoffier, Chris Crudele, Urs Muller, & Yann LeCun (2008b) Learning Maneuver Dictionaries for Ground Robot Planning. In *Proc. 39th International Symposium on Robotics (ISR'08)*.