Brian E. Mihok
2/16/05
16.412J/6.834J Cognitive Robotics
Homework 1

Part A: Topics of Fascination

Vision-based object recognition

I am interested in exploring vision-based object recognition in greater detail. The Sift algorithm created by David Lowe was one example presented in class. I am interested in learning about other scale and orientation invariant algorithms and their inherent advantages and disadvantages. However, I don't want to limit my investigation to just Sift-like algorithms. I came across the use of tree-structured belief networks (TSBNs) and dynamic tree-structured belief networks (DTSBNs) as other examples in doing the literature investigation and I am curious as to the breadth of algorithms used in visual recognition and why one algorithm would be used over another. Also in my investigation, I came across a great deal of research in facial detection and recognition. I would like to see how much of this work is applicable to detecting and recognizing man-made objects in natural environments. Finally, I would like to better understand what the state of embedded visual object recognition is currently and what work is being done to improve this state.

Multi-Agent Planning

Another area I would be interested to investigate further would be multi-agent planning. More specifically, I am interested in learning about the dynamic reallocation of resources based upon new information as it becomes available. I would like to learn about what is the best way to communicate this new information amongst the agents. Also, I want to learn about where the best place to compute a plan is for the system. Said differently, I am interested in learning the tradeoffs associated with each individual agent computing a plan and then comparing solutions versus having one centralized computation center for action and then distribute the plan to the agents. Similarly, I like to learn about the different methods under development regarding how decisions are best made in a multi-agent environment. I have worked briefly with genetic algorithms but would like to know more.

Neural Nets

The final topic of fascination is neural nets. I honestly do not know much at all about neural nets right now. As a result, I am primarily interested in learning topics other more advanced students might feel to be elementary. I would like to know the context of the origins of neural nets and what work has been done in the past. I am interested to know what the inherent advantages and disadvantages of neural nets are and why they would be used and in what scenarios. I would like to better understand what current research is being done and where participants in the research feel that the work will lead them over the course of the next 5-10 years.

Part D: Researching a Critical Reasoning Method

The first article that I read was entitled “Detection of Artificial Structures in Natural-Scene Images Using Dynamic Trees” and was written by Sinisa Todorovic and Michael Nechyba at the University of Florida. This article was accepted to the International Conference on Pattern Recognition in August 2004 and can be found online at http://www.mil.ufl.edu/mav/publications/papers/16.pdf.

I selected this article for two reasons. The first reason was that the detection of artificial structures in natural environments is a key component to the success of the UAV I described in Part B, which had a goal of maximizing the tracking of high priority targets while also maximizing the area surveyed. The first step towards achieving this goal is detecting man-made objects in a natural scene. Also, the authors are associated with the Center for MAV (Micro Air Vehicle) Research at the University of Florida, specifically the endeavor entitled Active Vision for Control of Agile Autonomous Flight. The University of Florida is a leader in MAV technology and research on the university level and has done work on vision-aided control of the MAVs they have developed, including horizon-based stabilization. It seems logical that the work discussed in this paper will eventually find its way into one of the MAVs.

The authors assume that the man-made objects are characterized “primarily by geometric regularities, and that artificial structures are rigid and composed of smaller, uniformly colored sub-parts.” They motivated the methods performed in the paper by giving a very brief reference to previous methods. The first method used an algorithm that extracted edges from images into larger geometric structures and the extracted edges served as nodes of a graph, where the geometric relations between the lines were links in the graph. The authors say that the graphs only accounted for nearest neighbor relations, limiting the complexity of the artificial structures. Next, the authors state that tree-structured belief networks (TSBN) have recently been used since they represent “pixel neighborhoods of varying size.” Though successful, the authors state that TSBNs give rise to “blocky segmentations.” The authors then propose to model man-made objects by dynamic tree-structured belief networks (DTSBNs) to eliminate this problem.

The DTSBN structures provide a way to detect both whole objects and parts of an object. The whole objects are modeled through the “root nodes,” while subcomponents are likened to the parent-child relationship on the tree. The authors thus believe that DTSBNs can lead to recognition of an object in the case of occlusion.

In addition to the choice of how to model objects, the authors also detail how they selected the image features. In looking for an efficient edge extraction method, they became convinced that traditional methods such as wavelets and Gabor filters were flawed and that a new method was required that took into account more than the “multiscale and localization properties” of wavelets. The authors suggest that geometric and color cues should be used to discriminate man-made objects from the surroundings. To do this, the authors proposed the use of multiscale linear-discriminant analysis (MLDA). The authors claim that MLDA “encodes both color and texture through a dynamic representation of image details” and that it “extracts edges over a finite range of locations, orientations and scales, decomposing an image into dyadic squares of uniform color."

The authors then describe the method for creating the dynamic trees and applying the MLDA to the image. Due to space constraints, I will leave that discussion to the authors in the actual paper. The analysis was applied to 100 256 x 256 natural-scene images with both artificial and natural objects captured by a ground camera at varied distances. This was done as a comparison between TSBNs and DTSBNs. The experimentation showed that DTSBNs outperformed TSBNs and that trained DTSBNs could be used for Bayesian image classification, leading to man-made object recognition in the future. The authors also concluded that DTSBNs could provide a "unified framework for unsupervised unknown object registration."

The strength of this paper is that it detected artificial objects in an unsupervised, natural environment with a method that demonstrated improvement over some existing research. The weaknesses of this paper in regards to the UAV project discussed was that the method relies on color images, a luxury night afforded to the UAV during night operations, and that the camera was ground based.

The papers I reviewed were selected independent of each other, but were instead based upon needs created by the cognitive robot definition. As stated above, the need that this paper addressed was the ability to detect and recognize targets in a natural environment. While this method could be investigated for daytime application, it would not work at night or if a color camera could not be used. Thus, a different approach would need to be found.

The second article I reviewed was entitled "A Low Cost Embedded Color Vision System" written by Anthony Rowe, Charles Rosenberg, and Illah Nourbakhsh at Carnegie Mellon University. This article was accepted to the IROS 2002 conference and can be found online at http://www-2.cs.cmu.edu/~cmucam/Publications/iros-2002.pdf. The reason I selected this article is because a soldier-portable UAV requires an embedded image system that is restricted to a payload-like size and weight requirement. When the entire vehicle only weighs 10 lbs, the vision system is necessarily restricted to being at the very most 1 lb. Thus, a typical desktop-frame grabber scenario is invalid. I wanted to see what the capabilities of an embedded image system and the article served as one such benchmark. Also, the CMU system allows for the tracking of a colored object. Tracking a detected object is another vital piece of the required functionality of the combat zone UAV described in Part B.

The article describes a product called the CMUcam, which is an embedded vision system which can perform basic color blob tracking at 16.7 frames per second. The motivation for the product was to extend vision system capabilities to applications that are restricted by size, complexity, and budget. In these types of applications, the traditional desktop computer with separate frame grabber is not feasible. Instead, the development and increased availability of CMOS cameras and microcontrollers combined to make an embedded vision system possible. The authors state that a major advantage of CMOS cameras versus CCDs is the “ability to integrate additional circuitry on the same die as the sensor itself.” The result is a vision system that is 1.75” x 2.25” and less than 2” deep with the camera module and lens attached, with a power requirement of 5 V at 200 mA.

The camera used in the product is an Omnivision OV6620 CMOS camera with an image array of 101,376 pixels, a supported resolution of 352 x 288, and a maximum refresh rate of 60 frames per second. The camera interfaces with the microcontroller using a standard serial port.

The microcontroller that is used to process the video is a Ubicom SX28 operating at 75 MHz. It is a RISC processor and operates at 75 MIPS, with a 2048 word flash programmable EPROM and 136 bytes of SRAM. The microcontroller also has fast and deterministic interrupts and three multi-bit I/O ports that were used to implement a serial UART port, a standard hobby servo PWM output port, and to control a status LED on the system. All firmware for the vision board was written in C and compiled using the ByteCraft SXC v2.0 compiler.

The color blob tracking was done using a simplistic algorithm that user to specify a the color limits of the object to be tracked and the bounding box in which the tracking was to be done. Then, analysis was performed to compare the number of color pixels within boundary to the colored pixels outside the box. More detailed analysis could be done, such as determining whether only one compact object was contained in the box or multiple objects, as well as a slew of color statistics and windowing options, but the details are best left to the paper due to length considerations here.

The authors reported that once the camera’s color bounds were set, that the CMUcam could track a blue 14” x 15” x 10” recycling bin confidently up to 35 feet away. With the appropriate IR coated lens, the authors reported that the system “performs well in a wide range of lighting conditions, including direct sunlight outdoors,” with nearly identical results outdoors as inside.

Finally, the authors created a 4” x 3” x 3” robot containing a small differential drive mobile base and a PIC microprocessor to further test the camera’s tracking abilities. The demonstration robot could successfully track a “small brightly colored red doll” at distances up to 15 feet.

The strengths of this article is that it details a system that is on the correct scale for size and weight as that which is required for a vision system that would be employed on a soldier-portable UAV. However, the weakness was that the system was completely incapable of providing the necessary level of functionality for the task. The tracking algorithm relied on color, which would not be available during night operation. The objects being tracked were either monochromatic or very simply colored to facilitate the simplistic algorithm instead of man-made objects in complex terrain. Even with the simplifications, the range, processing power, and memory were drastically undervalued for the desired application.

As previously stated, the three articles were not selected to have any relation to each other, but were instead selected to address functional requirements of the purpose cognitive robot. This article addressed the need for an embedded vision system that weighed less than 1 lb but could still perform object tracking. The article gives an idea of the state of technology for embedded systems on the scale of interest for the desired project.

The last paper I reviewed was a bit different than the other two. Instead of being directly related to the algorithms required to add the necessary intelligence to the proposed UAV project, the paper discusses the creation of an image database created for facial recognition. While the direct application of the topic does not apply, the issues dealing with capturing an image database across different poses, illumination patterns, and expressions can be considered analogous to the problems that would be encountered if an image database were to be created of military targets. The article was entitled "The CMU Pose, Illumination, and Expression Database" and was written by Terence Sim, Simon Baker, and Maan Bsat. It can be found online at http://www.ri.cmu.edu/pub_files/pub4/sim_terence_2003_1/sim_terence_2003_1.pdf.

The authors addressed the need for a database consisting of "a fairly large number of subjects, each imaged a large number of times, from several different poses, under significant illumination variation, and with a variety of expressions." To accomplish this, the authors suggest that imaging something from multiple poses either requires multiple cameras or multiple shots taken consecutively. They propose that the multiple camera option was more advantageous, if possible, for reasons including less data collection time, the fact that if the cameras are fixed in space, the relative pose is the same for every subject and there is less difficulty in positioning the subject to obtain a particular pose, and the fact that the imaging conditions are the same if the pictures are taken simultaneously. The disadvantage to this method is that multiple cameras, digitizers, and computers are needed, the cameras need to be synchronized such that the shutters all open at the same time, and that the cameras will have different properties.

The authors decided to use the multiple camera, simultaneous data collection method due to the "3D Room" at CMU. The 3D Room has 49 cameras, 14 of which were high quality Sony DXC 9000s. To produce similar image quality across the database, 13 of the 14 Sony cameras were used to take the pictures.

To obtain illumination variation in the pictures, the authors installed a "flash system" in the 3D room that was similar to one used at Yale by Georghiades et al in 2000. The authors used an Advantech PCL-734, 32-channel digital output board to control 21 Minolta 220X flashes. Generating a pulse on one of the output channels then caused the flashes to go off. The flash control code was integrated into the image capture routine such that flash occurred while the shutter was open. The authors modified the image capture routine so that they were able to capture 21 images, each with different illumination, in approximately 0.7 s. The capture routine took about 10 minutes per subject, allowing the authors to capture and store over 600 images from 13 poses, with 43 different illuminations, and with 4 expressions. The images were color and had a size of 640 x 486. This required approximately 600MB per person for a total of around 40GB.

While the topic of an image database of facial pictures does not have much direct application to a soldier-portable vehicle, the strength of the paper was that it provided background into some of the issues associated with creating an image database for objects. If such a database were created for military targets, some of the issues overcome by the authors of the article would need to be addressed. For instance, while the expression of a tank might not have any meaning, the illumination differences and the relative position of the viewer to the tank might make a difference, depending on the visual recognition algorithm. In order to obtain a full definition of all sides of an object, multiple pictures would have to be taken and analyzed, as was the case in this paper.

In addition to addressing some of the obstacles encountered in created such a database, the paper also gives a first approximation for time required to create such a database and the amount of storage required. While this certainly varies depending on the number of pictures taken per object and the amount of objects imaged, the paper at least provides a baseline estimate.

Part E: A Simple Project for your Cognitive Robot

I would like to pursue a project in either visual object recognition or in developing the preliminary work towards an automated landing capability for a UAV using information provided by a lidar system.

The first type of project mentioned above is still as of yet poorly defined in my mind primarily due to my lack of substantive knowledge on the subject. I have never worked with visual recognition of objects before so I do not have a good feel for the scope of a reasonable project given a 1-2 month timeframe. Based upon the requirements for the cognitive robot described in Part B, I would like to implement a visual object detection algorithm that can at the very least detect what objects are present in a given picture or video stream. Once these objects are detected it would be ideal to perform at least some nominal comparison of features to that of known objects to see if a match could be found. No requirement of tracking would be necessary for the project, as object detection and recognition serve as the foundation for the UAV functionality.

I do not know at this time if implementing such an algorithm is realistic in the given time frame and if so, which algorithm best lends itself to the application. To mitigate my lack of knowledge and specificity on a vision-based object detection system, another possible project of interest is working towards an automated landing procedure for a UAV based upon information by a lidar system. As mentioned in Part C, such a system would be applicable in a broad range of UAV application, would save in both operational cost and the efficiency of vehicle design, and is in demand by the military currently. A lidar system is a laser radar, which contains a laser that sweeps across a designated swath and returns the distance to the closest object at a given angle. If this tool was aimed downward at an angle such that it projected ahead of the vehicle, the resulting distances could be used to define a topological map of the area flown over by the aircraft. Combined with the UAVs position, the lidar could be used to map the landing area.

As discussed in Parts B & C, two different methods could be used to land the UAV with the lidar. In the first method, the UAV would attempt to land with no prior knowledge of the surroundings whatsoever. The autoland system would attempt to avoid obstacles as they are detected, but there would be no effort to find an ideal landing location.

In the second method, the UAV would utilize two passes over the designated landing region. During the first pass over the region, the lidar would collect the information and the topological map would be created. Then, as the UAV is in transit for the second pass, the autoland system would use the topological map to determine where the ideal location is for landing. It would then determine the optimum trajectory to guide the UAV to the selected landing site, thereby maximizing the chance for survival. At a certain elevation above the ground, as determined by the lidar system, the autoland system would execute a pitch-and-stall procedure to bring the vehicle down as gently as possible.

My proposal would be to use the second method to land the UAV. I would of course need several simplifying assumptions. The first obvious statement is that this would all be done in simulation and not hardware development. Next, I would assume that the simulated UAV would behave exactly as commanded in real-time without error. I would also assume that the first scan of the landing region was already performed and the lidar had properly detected and mapped the region. From these assumptions, I would then create a 3D topological map and select the best landing location. From this location, the surrounding terrain, and the known position, attitude, and velocity of the aircraft, I would then attempt to calculate a safe trajectory to the landing site. To be a bit more realistic with the given time constraints, I would say that a safe trajectory is more than adequate rather than an optimal trajectory and that the surrounding would need to be "reasonable," so that the UAV would not be attempting to land in ridiculous environments in which no pilot would ever attempt to land.