Proceedings of the 2011 ASME International Mechanical Engineering Congress & Exposition
IMECE2011
November 11-17, 2011, Denver, Colorado, USA

IMECE2011-63805

VISION GUIDED MOTION CONTROL OF A BIOMIMETIC QUADRUPED ROBOT —ROBOCAT[1]

Steven Carpenter, Xinming Yu, Melih Altun, James Graham, J. Jim Zhu*, Janusz Starzyk,

School of Electrical Engineering and Computer Science, Ohio University

Athens, OH, 45701 USA
*Contacting Author, zhuj@ohio.edu

ABSTRACT

This paper presents the vision system and visual processing for a biomimetic elastic cable-driven quadruped robot— RoboCat. This paper presents the vision system and visual *processing for a biomimetic elastic cable-driven quadruped robot—RoboCat. The paper is geared towards selection of appropriate visual servoing techniques for RoboCat such as vision algorithms, high-level cognition algorithms, software architecture and hardware implementation. The system uses two video cameras for stereo vision data acquisition and a SUMIT-ism form factor embedded computer for vision data processing. The vision system employs a color based target recognition algorithm, a neural network based shape recognition algorithm and a Color and Zernike moment based face detection algorithm. The paper presents the vision algorithms, vision guidance and motion tracking algorithms, rule-based decision making algorithms and the open architecture of the autonomous vision tracking system. Experimental testing results (including video clips) are also presented.*

INTRODUCTION

RoboCat is a biomimetic, or biologically inspired robot, so it follows that the control systems and software should also have biologically inspired elements. Legged Locomotion is attractive to researchers due to interests in realizing artificial systems similar to legged animals and also as a method of navigation over rough terrain [1]. Maneuvering and navigation in unstructured environments presents many challenges for mobile robots [2]. Although wheeled robots (which do not appear in nature) can be quicker and more efficient in traversing some terrain, legged robots can reach a much larger percentage of the terrain [3].

[1] This paper is based on research sponsored by the Air Force Research Laboratory, under agreements number FA8651-07-1-0010 and number FA8651-09-1-0016. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

Intelligent control systems which require accurate estimates of the state of the world around the robot is necessary to eliminate the tedious tasks of navigation and target recognition [4]. RoboCat is an elastic cable driven quadruped robot, which currently employs three target recognition algorithms for vision based autonomous navigation.

RoboCat navigates on the premise that if an identified target is within its view it will follow the target. This is a form of intelligent decision making. Household robots such as the iRobot Roomba are capable making an intelligent decision based on its current battery life. For example when its battery capacity drops below a threshold Roomba will begin looking for an IR beacon signal transmitted from its base station which Roomba can detect within a distance of 5 feet [5].

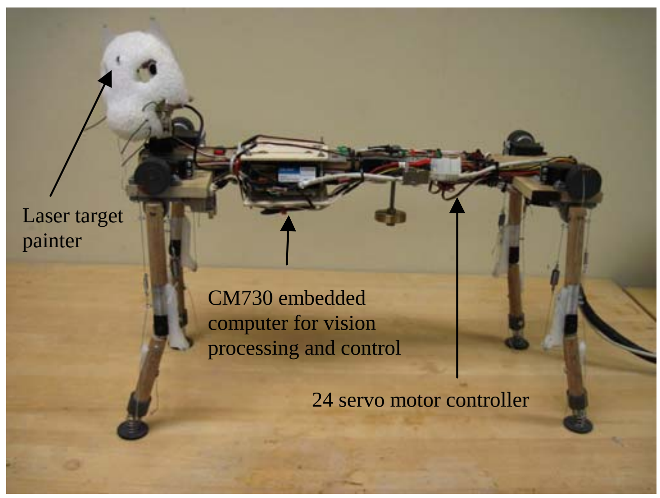


Figure 1. RoboCat won Ohio University's 2009 Student Research and Creative Activity Expo.

RoboCat implements a color based target recognition algorithm, a color and Zernike moment based face detection algorithm and a neural network based shape recognition algorithm. RoboCat's color detection algorithm allows it the capability of following an orange fish. While its face and shape detection algorithms allow it to track more advanced objects such as tennis balls, power sources and people.

While many early attempts at AI were (and still are) very crude, in recent years there has been an upswing in biologically inspired design. This makes sense when one considers the inherent difficulties in creating intelligent systems and the fact that there are many examples of varying degrees of "intelligence" present in the animal world. Various cognitive architectures have been proposed, all of differing complexities and focuses. Most of these architectures take inspiration from how we perceive the process of human and animal cognition. Older architectures such as SOAR [6][7] represent some of the initial forays into this area. SOAR was primarily rule based and had distinct "production" and "decision" mechanisms. Architectures such as CLARION [8], ICARUS [9], and LIDA [10] all have their roots in biologically inspired design. Dörner's Psi model, as presented in [11], is perhaps one the more fully-featured models and bears significant similarity to the Motivated Learning model [12] that the proposed cognitive mechanism shown in this paper is based upon.

VISION

Content analysis, the ability to identify what is in an image, became an active research area in the 1970's due to a thrust from both the database management and computer vision communities [13]. The database management community sorted images by text descriptions of the image, while the computer vision community sorted images by the visual features of the image.

The visual features of an image can be separated into general features and domain specific features. General features are those that are based on color, texture, and shape. Domain specific features are based on the location of the general features in the image. These are used for face, eye and fingerprint identification.

Human vision systems can track an object and recognize it perfectly, from the complicated background with various colors, textures and non-linear variations. However, for a machine, it is an extremely complex task. Object recognition is attempting to emulate a human (and biological) visual function. The objective of real-time object recognition is to give machines the ability to identify and label elements of a scene.

In the field of computer vision, there are lots of methods used for object recognition, i.e. gradient histograms [14], template matching [15], etc. Vision algorithms based on color are the least computationally expensive and therefore are the fastest to implement. Unfortunately vision algorithms based on color tend to fail under varying light intensities or when the background becomes more complicated.

Nevertheless, RoboCat's vision system employs an open architecture which allows researchers in content analysis to implement their vision algorithms on a biomimetic robot regardless of the programming language the code has been written (MATLAB, C, C++, etc.).

A. Color Based Target Recognition

Vision algorithms based on color are the least computationally expensive and are therefore the fastest to implement, but they fail at recognizing targets from clustering complex images. Slight changes in light intensity can ruin an entire algorithm based on color. Regardless, the color based target recognition algorithm filters an image of all pixels except the target color. It then finds the geometric center of the region containing the largest amount of the target color. Figure 2 shows RoboCat following an orange fish using color based target recognition

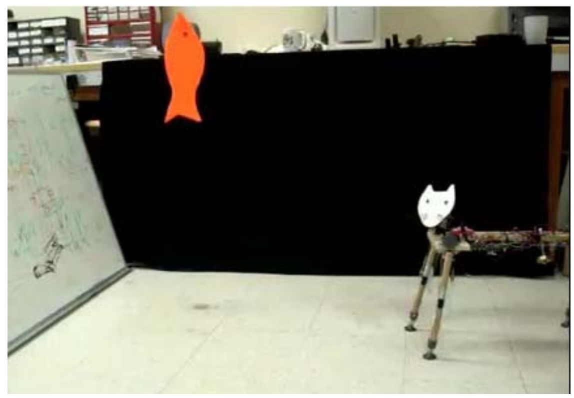


Figure 2. RoboCat following an orange fish.

B. Neural Network Based Shape Recognition

Neural network based real-time object recognition is a specific field of study in the area of computer vision. Compared with other methods, the greatest advantage of artificial neural networks (ANNs) is their ability to be used as an arbitrary function approximation mechanism that 'learns' from observed data.

For traditional artificial neural networks (ANNs), the training data sets are given, which means the ANNs can only learn some pre-specified objects. In order for RoboCat to have the ability to learn any object, a motion detection [16] algorithm is applied to produce a rough outline of the object, shown in Figure 3. By moving an object in front of the webcam, the motion detector can capture a training set for the artificial neural network from consecutive frames. However, the obtained training set could not be used directly because the sizes of the datasets captured from the real-time visual input can vary, causing problems for the neural network. The next step of data preprocessing is to sample the obtained visual data.

There are two advantages of data sampling. One is to make the size of obtained training data uniform. The other is to decrease the amount of input data to the ANN. The onboard processor PC104 is not as powerful as a desktop computer, so its processing ability is limited. Too much data processing might cause delay of real-time response. A retina sampling model was developed for this step.

Figure 3. (Left) The can is detected from the background. (Right) The data to be used as the training data.

The retina, unlike a camera, does not simply send a picture to the brain. The retina spatially encodes (compresses) the image to fit the limited capacity of the optic nerve. As shown in figure 4, the photoreceptors are not evenly distributed inside retina. Most of them are concentrated on or around fovea.

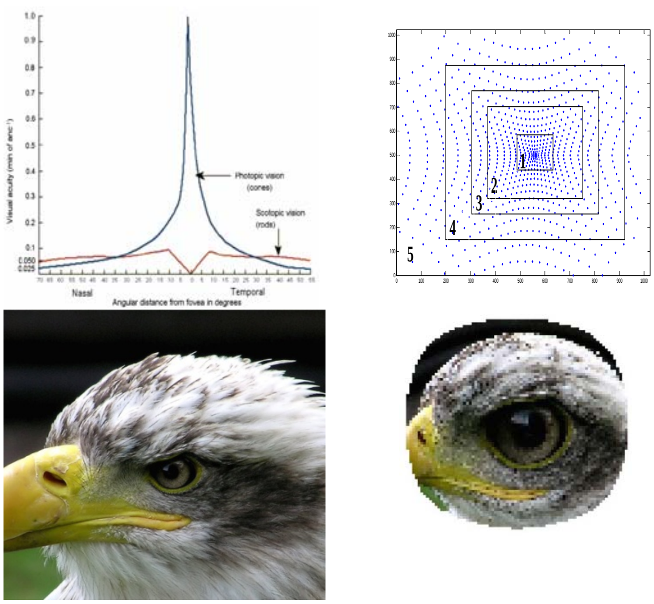


Figure 4. (Top-left) PDF of photoreceptors
(Top-right) Sampling points for retina model
(Bottom-left) Example: original image (900x900 pixels)
(Bottom-right) Example: sampled image (5198 pixels)

When this retina sampling model is applied, the size of the input data will be extremely limited, while most of the useful visual data is still available in the sampled training data set. Table 1 below, is a comparison between retina sampling and uniform sampling. The top right portion of figure 4 shows how the resolution in the center part of the retina sampling is much higher than that of uniform sampling.

Figure 5 shows consecutive frames from a frame sequence as examples of the application of retina sampling. The left column shows three different frames captured by the RoboCat camera; the middle column shows the motion detector captured objects of different sizes; the right column shows the sampled visual data by retina model. In addition to what has been discussed about the retina sampling, one more important aspect is to make the recognition process scale-invariant. The sizes of the captured data set (column 2) vary from frame to frame; however, the sizes of sampled visual data are identical (25x25), and can be used as an input to the ANN.

Table 1. Comparison (human vs. computer vision)

	Percentage of sampling points used	
	Retina Sampling (Human Vision)	Uniform Sampling (Computer Vision)
Inside Region 1	31%	4%
Inside Region 2	52%	14%
Inside Region 3	63%	25%
Inside Region 4	78%	50%
Inside Region 5	100%	100%

The final step of the data preprocessing is to convert the sampled data from RGB color space to HSV color space. RoboCat is tested in an indoor environment. The advantages of HSV over RGB are fully discussed for such indoor illumination [17].

It may be noted that a back-propagation neural network (BPNN) with hidden layers and one output layer can approximate any non-linear function. In order to reduce the complexity of a network, and increase computational efficiency, usually only one hidden layer is chosen. [18] So, a BPNN network with reinforcement learning is chosen for RoboCat, with which the RoboCat is trained to ‘remember’ the selected object by adjusting the weights of network connections.

The input layer of the BPNN is determined from the characteristics of the sampled dataset. There are 25x25 pixels for each set of input. Each pixel contains three elements (HSV). Thus, the total number of elements for the input layer is 1875. Compared with the original video frame, which contains 230,400 elements, the input dataset is much smaller and can be processed in real-time.

As discussed before, only one hidden layer is selected for this BPNN network. The hidden layer automatically extracts the features of the input data and further reduces input dimensionality. It was realized that 20 neurons for the hidden layer provided good performance.

Learning rules, for a connectionist network, are algorithms that govern changes in the connection weights . The most popular learning rules are learning rules that incorporate an error reduction procedure or error correction procedure. Learning rules incorporating an error reduction procedure utilize the discrepancy between the desired output pattern and an actual output pattern to change its weights during training. The error correction learning algorithm is the most popular rule learning method for object recognition with known training data. The objective of the error-correction learning algorithm is to start from an arbitrary point on the error surface and then move toward a global minimum, in a step-by-step fashion [18]. The equations for the error correction learning algorithm are shown below

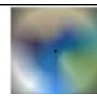
Frames(240x320)	Captured dataset	Sampled dataset
	 79x53	 25x25
	 95x62	 25x25
	 105x69	 25x25

Figure 5. Example of retina sampling application in RoboCat.

$$e_k(n) = d_k(n) - y_k(n)$$
$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$
$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$$

where $d_k(n)$ is the desired output of neuron k at step n, $y_k(n)$ is the actual output of neuron k at step *n*, $\Delta w_{kj}(n)$ is the weight adjustment of the connection between neuron *k* and *j* at step *n*, and η is a positive constant that determines the rate of learning.

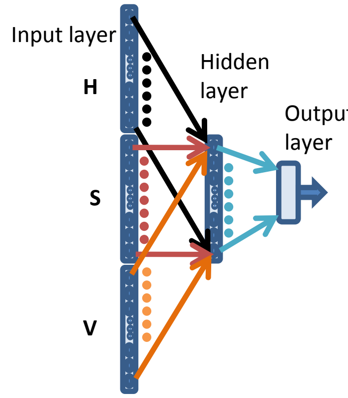


Figure 6. Structure of 3-layer neural network.

The training of the artificial neural network was performed using MATLAB and its image acquisition toolbox. The training dataset is generated and used simultaneously when the selected object is moving in front of the camera. To finish the training process, a 50-second WMV video clip was taken with a moving compressed air can. The frame rate of the camera is 15 frames/sec, so there are 750 frames captured for the training purpose. The network was trained for 1500 epochs (750 x 2). The mean-square-error curve plotted during the training process is shown in the figure below and shows how errors decreases from about 0.27 towards the pre-set threshold (0.01).

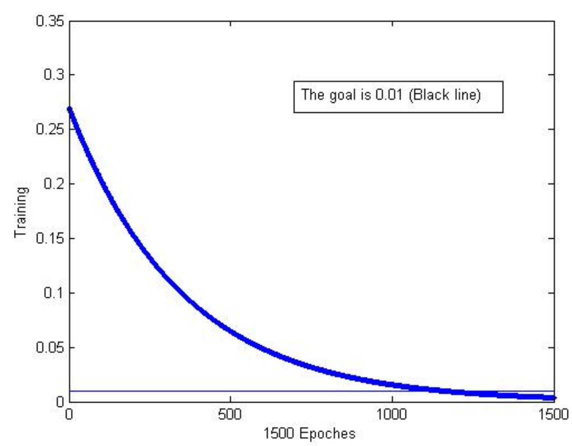


Figure 7. Mean square error curve.

After the training process is done, the neural network must be tested. Figure 8 shows some of the frames from the testing process. The centers of the object were marked by red-cross. It was found that the response of the neural network is accurate for most of the scenarios. After running 2000 consecutive frames with the desired object, the accuracy percentage was around 73%.

Figure 8. Some testing results.

C. Color and Zernike Moment Based Face Detection

There are various methods used for face detection. These include color based methods such as histogram thresholding [19], Principal Component Analysis (PCA) [20], Neural Networks [21], Zernike Moments [22], etc. To improve performance these methods are combined in many cases. In this paper previous face region extraction method are combined with Zernike Moments. The previous method [19] was based on color clustering using Self Organizing Feature Map (SOFM) neural network and histogram comparison. The method worked by clustering colors and selecting skin like regions in an image and detecting faces among them using histogram properties. This method is improved by adding Zernike Moments to detection process.

It is possible to find many solutions developed for skin color detection in the literature. Although there are different approaches using different color spaces [23][24] more recent studies have shown that color space has a little impact on performance of the skin color thresholding [25].

Extracting moments from an image is another method applied for face detection and recognition. Various moments can be used for this purpose such as Zernike Moments, Pseudo Zernike Moments and Legendre Moments [27]. Zernike Moments are known to reduce geometric features into projected vectors with orthogonal properties [22]. Features obtained by Zernike Moments have certain advantages over other feature extraction methods. The most important advantage is their rotation translation and scale invariance. Using Zernike moments with the previous work based on color clustering and histogram comparison will improve the methods robustness and accuracy.

The face detection algorithm filters the image of all colors that are not a candidate face color. The algorithm places a dot in the location of the geometric center of the area containing the largest number of pixels which is a candidate face. It then uses Zernike moments that are based on the domain specific features of the image (the location of the eyes and mouth within the image) to determine if the candidate is indeed a face. If it is a face, the algorithm draws a square around the face and paints a laser on it.

The face detection algorithm works as follows: Using the RGB values between 0 and 255 the dark areas and most non-skin color areas can be eliminated by checking if:

$$\left\{ R > 95 \text{ and } G > 40 \text{ and } B > 20 \right\}$$

If the red, green and blue pixels are within this region then the pixel is part of a candidate face. Since human skin colors have certain saturation it is checked that the color is not gray, which is true if *max{R,G,B} – min {R,G,B} > 15*

To further improve the algorithm, the value of the blue component of each pixel is considered since blue cannot be the dominant component in a skin color; a pixel is part of a candidate face if *(min {R,G,B} = B) AND (B<170)*

The above algorithm works best under uniform lighting conditions. To check the lateral illumination conditions such as shining a flash light upon the face it is checked that:

$$\left\{ R > 220 \text{ and } G > 210 \text{ and } B > 170 \right\}$$

Again the grayness and the value of the blue component is checked: *max{R,G,B}–min {R,G,B}>15 and min{R,G,B}=B.*

When the given input pixel with RGB format satisfies one of the illumination conditions above it is marked as a skin color. All the other areas are marked as non skin areas and can be replaced with a uniform white color in order not to be used in further processing.

After finding skin color regions, closed contours are detected to isolate each candidate face. This separates multiple faces and background areas with skin like colors.

Then geometric center of the connected component is calculated with geometric moments similar to the method in [26]. Geometric moments of the face mask are calculated using the equation:

$$M_{pq} = \sum_x \sum_y f(x,y)\, x^p y^q$$

where x, y are pixel coordinates and p, q are the order of moments. To find the center of the connected components we calculate M_{00} M_{10} and M_{01}. Then:

$$x_0 = \frac{M_{10}}{M_{00}}$$

$$y_0 = \frac{M_{01}}{M_{00}}$$

gives the center of gravity of the connected components.
The Zernike polynomials are calculated using

$$V_{nm}(p,\theta) = R_{nm}(\rho)\, e^{jm\theta}$$

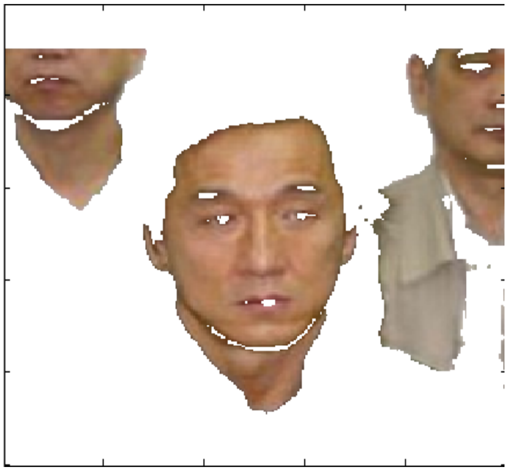


Figure 9. A skin color filtered image.

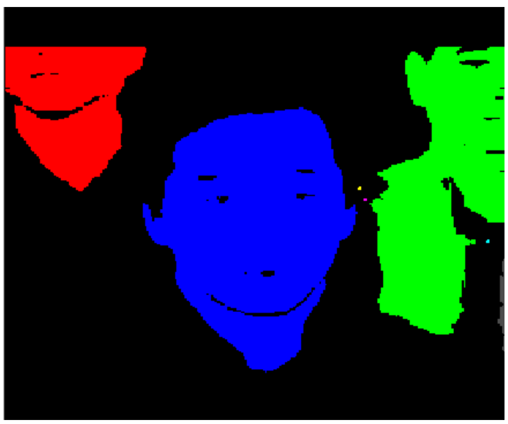


Figure 10. Areas separated by closed contours. Areas with less than 2000 pixels are considered too small for face detection and are eliminated.

The Zernike moments around the calculated center of the connected component are found using the formula:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f\left(x, y\right) V_{nm}^*\left(x, y\right), \quad x^2 + y^2 < 1$$

Where x, y are coordinates relative to the center of mass found above. That is x_0 and y_0 coordinates found from M_{pq} are used as the origin and Zernike moments are calculated for a unit circle around that origin. The calculated Zernike moments are given to a neural network as input. The neural network used for this project is a perceptron, which has been trained by test images. The neural net uses the Zernike moment polynomials up to A_{55} (A_{00} to A_{55}) and decides if the given image contains a face or not.

The images used for testing the proposed method are obtained from the Labeled Faces in the Wild (LFW) database (except Figure **11**), provided by the University of Massachusetts. Although a limited set of sample images have been tested so far, over 70% of them resulted in a successful detection. To improve the success ratio neural net can be optimized to produce better results. A more sophisticated neural net such as a back propagation network might produce better results. Also, for detection only five dimensions of real Zernike moments are used. Using more dimensions will improve the accuracy of detection. Moreover, complex Zernike moments can also be utilized to enhance the detection ratio. When the accuracy is high enough extracted Zernike moments can be used to add face recognition capabilities to the method.

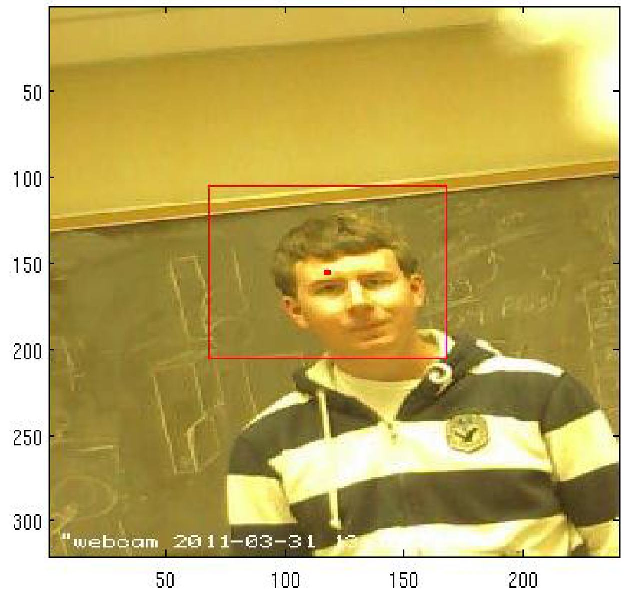


Figure 11. Detecting a candidate face dot and determining if it is indeed a face (red square).

TRACKING

Once a predetermined target is identified, the RoboCat's head will track movement of the target to keep it in the center of the view field. Once a decision is made to move towards the target based on some preset criteria, a body motion trajectory is generated and the corresponding gait sequence is sent to the lower-level gait controller to carry out the movements. The head will continue to track and stare at the target while the robot is in motion until either the target is reached, or a decision to abandon the target is made based on some preset criteria.

The current tracking controller uses simple proportional control. The controller is implemented by choosing a square region in the center of the image captured from the environment. If the object is within the square, RoboCat's head will remain stationary. If the object leaves the square, than RoboCat will move its head to bring the object back inside the square. If the object is outside of the quadruped's head motion range, it will keep the head in the current position until the object re-enters its viewing range.

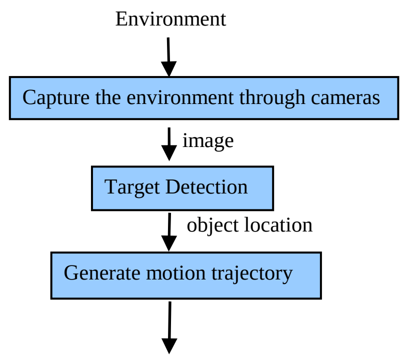


Figure 12. The vision guidance components consist of capturing an image from the environment through cameras, locating an object based on the features in the image, and using the object's location to generate a motion trajectory.

Since RoboCat's walking algorithm does not yet allow it to turn, it will only walk toward the object if it is directly in front of it. When the RoboCat is capable of turning a more advanced tracking control system based on line-of-site will be developed. This algorithm will move the body and the head to keep the object in the center of view.

INTELLIGENT DECISION MAKING

This section describes a paradigm and software architecture for primitive motivational cognition in autonomous intelligent agents, which serves to direct the RoboCat's (visual) attention and actions in a meaningful manner under autonomous operating conditions. More detailed exposition can be found in [28].

Intelligence and decision making in the RoboCat are currently very basic, however, there is a great deal of room for various forms of intelligent control in the design. Therefore, this section focuses on potential future improvements to the RoboCat's controlling "intelligence." The implementation of intelligent control depends on what functions we want to modulate, and how we want to motivate the robot. These basic motivations will determine how the robot responds, and eventually, when dealing with more complex environments and cognitive architectures, how its intelligence will grow. Initially, the RoboCat will be limited to basic motivations such as maintaining its power supply, ensuring that it stays at the correct operating temperature, and generally avoiding harm. This is analogous to the primitive needs of hunger, "comfort," and the avoidance of possible threats (unfamiliar/unexpected noises, unknown objects, and known harms) that a real biological cat might experience. It's easy to see how these seemingly simple motivations can blossom into some very complex behaviors. For example, the robot might have to weigh its choices between positioning itself out in the sun to recharge via solar panels against the possibility of overheating due to a damaged or failing cooling system. (Again, this could be seen as analogous to choosing between hunting for food or getting out of the sun and resting.)

In this approach, the robot has several primitive needs that provide it with its initial motivations. In the process of attempting to fulfill its needs, it will learn to balance the needs with what it is actually capable of and what is available to it. In more advanced implementations it will be capable of forming associations and developing further understanding of its environment, which will, in time, allow it to build upon its primitive motivations to form more complex abstract needs and motivations and develop more complex behaviors in order to fulfill these needs. **Error! Reference source not found. Figure 13** presents a simplified conceptual diagram of the robot's control functionality, where the motivations, combined with a memory system, a cognitive planning mechanism, and attention switching system will provide the robot cognitive functionality.

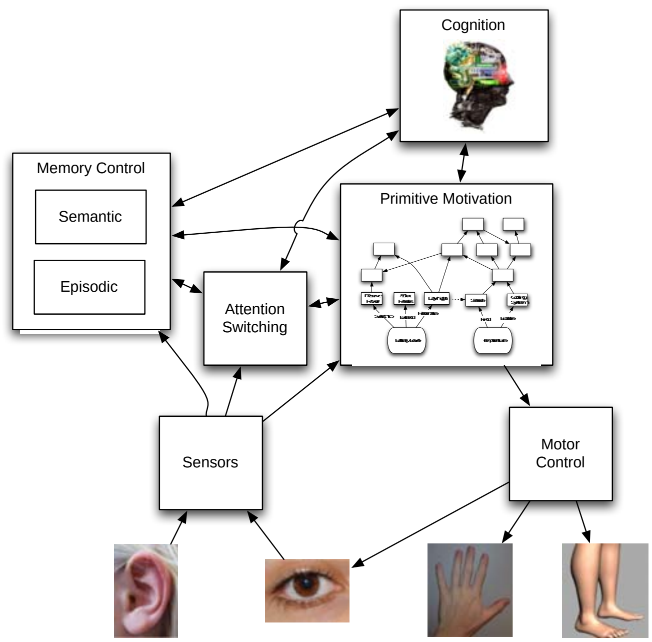


Figure 13. System Consciousness Diagram.

In **Figure 13Error! Reference source not found.**, the motivational system is what drives the robot to perform its actions. It receives its input and weighs everything to determine the best course of action. However, it does not work in a vacuum. First, an attention mechanism is needed to direct the robot's attention [28]. This is necessary because it is computationally demanding (as well as inefficient) to pay attention to each and every object present in the environment simultaneously. Attention can be affected by any of the controlling models and/or the sensors. For example, an abrupt appearance of an abject in the visual field, a memory evoked by a change in the environment, a change in the priorities as defined by the motivation module, or some facet of the current thought process in the cognition/planning module could all cause a switch in attention. The attention module itself controls what the robot "pays attention to." In terms of visual tracking, the robot will pay attention to the object it is tracking, but if in the "peripheral vision" something new shows up it may briefly decide to redirect its attention to identify the new object. If it is deemed irrelevant, the robot can quickly return to its original "task;" however, if it is relevant the robot may decide to follow the new object. Memory also plays a role. With it the machine will be able to keep track of and analyze past occurrences. Memory in conjunction with the planning module has the potential to allow a robot to plan a series of actions, adjust its priorities, or shift attention as needed for the current task.

However, the core of the decision making system is the motivational section. Figure 14 shows a pain-alleviating/harm-avoiding motivation network stemming from the most primitive needs of an autonomous agent such as hunger (battery life) and environmental elements (temperature). For example, when the battery of the RoboCat is lower than a certain threshold, it will be translated to the pain being above a threshold in the cognitive computer system, which will motivate the RoboCat to seek alleviation by extending its solar panel if it senses the presence of strong light, or engage an alternative solution, such as finding a charger. When the battery is charged sufficiently, the pain will have fallen below threshold allowing the RoboCat's attention to be directed to another pain. The process of charging, known to lower the pain sensation, will be associated with a pleasure sensation that will be reinforced in memory, so that when there are no outstanding pain sensations, the Robocat can direct its attention to satisfying its urge for pleasures, thereby demonstrating pleasure-seeking motivational behaviors like its biological counterparts.

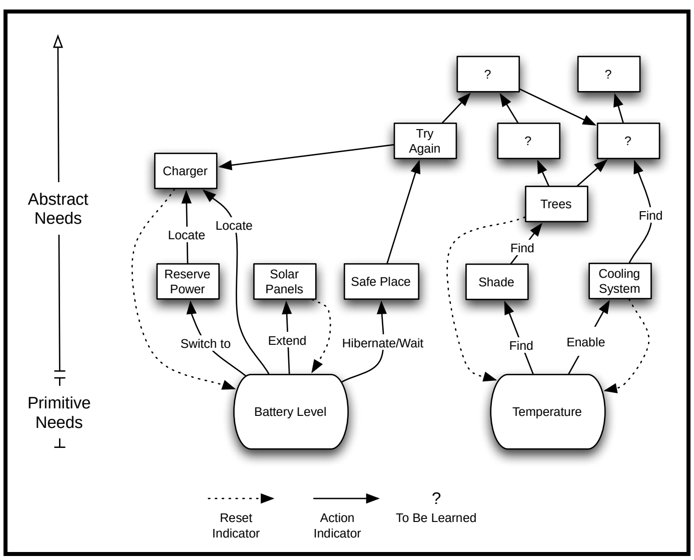


Figure 14. Motivation Diagram showing decisions that RoboCat might make based and temperature and battery life

Similarly, extreme environmental temperatures may cause malfunctions in the RoboCat's onboard subsystems, which will be detected and trigger the appropriate pain sensations. Other known and recognizable threats to the RoboCat's wellbeing, such as the presence of a certain shaped objects or unexpected movements of objects, will also be detected and trigger the corresponding pain sensations. When conflicting pain sensations are present at the same time, the levels of pains will be weighed and a decision will be made based on tradeoff of priorities for surviving and thriving. When multiple solutions exist, the agent will weigh the known variables and attempt to find optimal solutions to whatever need is currently driving it. For instance, once the Robocat discovers that it is able to regulate its temperature just as well by operating in the shade as utilizing its built-in cooling system it will "prefer" to use the shade since it will be more cost effective in terms of energy usage. It is noted that the possible actions shown in the figure are only representative, and specific actions depend on the capability of the autonomous agent. The unspecified actions in the figure signify the possibilities of learned reactions. The learning process will involve trial-and-error, accidental discoveries, evaluation of effectiveness, repetition and associative memory, which are not represented in Figure **14**.

The proposed ability to learn abstract motivations has the potential to lead to complex and novel behaviors. For example, real cats are known to associate certain people/behaviors with food and/or play and are perfectly capable of being taught to perform certain play actions before being fed. This type of learning/memory behavior should translate into similar scenarios in the RoboCat design. For instance, the RoboCat may learn to associate a certain person with danger, while another may be a source of "pleasure" (food or some other reward). It would be capable of learning to do such things as seeking energy sources, seeking "pleasure", avoiding known threats, and avoiding "potential threats" (unfamiliar/unknown noises for example). As the complexity of the RoboCat's body (embodiment) improves, its ability to learn and interact with its environment should also improve. Ultimately, it is hoped this research will lead to further improvements in embodied intelligence and robot design.

HARDWARE IMPLEMENTATION

The system uses two video cameras for stereo vision data acquisition; a SUMIT-ism form factor embedded computer for vision data processing, wireless communication and tracking guidance; two micro-controllers for gait control; and a 24-channel servo controller for head and leg actuation. The quadruped robot employs biologically inspired elastic cable-driven joints. Each leg has 3 Degrees-of-Freedom (DoF) actuation with a 2 DoF passive ankle joint, and the head has 2 DoF actuation while the video cameras are fixed relative to the head.

The image processing and motion trajectory generation of RoboCat is performed using the Sumit-ISM form factor embedded single board computer manufactured by ADLINK

Technology Inc. The embedded computer is a CoreModule 730 (CM730). The CM730 is located on the chest of the RoboCat and weighs 11.5 Oz with its heat sink. The CM730 operating system (OS) is a GNU/Linux desktop distribution: Linux Mint Helena. The OS is installed on a PATA 8GB solid state flash disk module which plugs into the 2.5 inch IDE slot on the CM730. The FDM is 2.5'' x 1'' x 0.3'' with a read/write capability of 80/75 MB per second. The FDM is a solid state drive therefore unwanted inertia is not added to the dynamics of the quadruped.

The CM730 allows for easy troubleshooting since it can be accessed via standard personal computer (PC) input/output peripherals. During testing the CM730 is operated as if it was a standard PC. During autonomous operation the keyboard, mouse, Ethernet, and monitor can be removed. The CM730 can then be accessed using a remote desktop application.

The CM730 communicates with the PIC control board via the CM730's general input/output (GPIO) pins and the PIC microcontroller's RB0 pin. The PIC control board continually monitors the RB0 pin. When the voltage on the RB0 pin is high RoboCat begins to walk and when the voltage on the RB0 pin is low the RoboCat assumes a stable position.

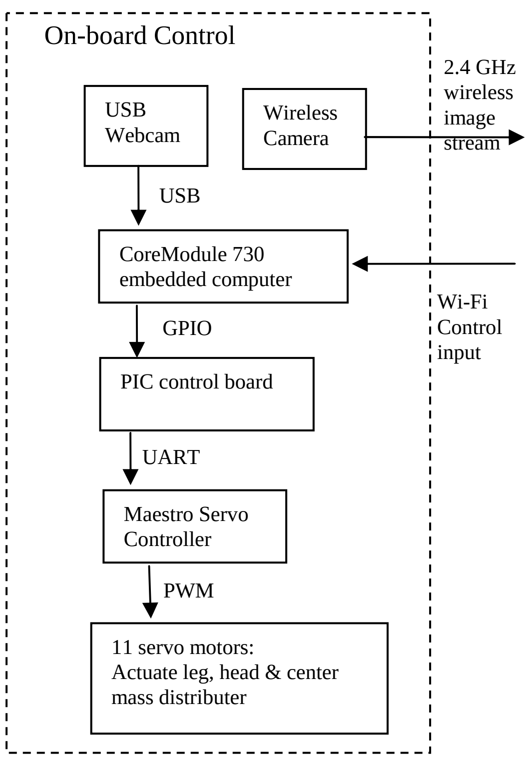


Figure 15. Block diagram defining the on-board autonomous communications stream of RoboCat.

During onboard control, a USB webcam captures an image and sends it to the CM730 over USB interface. The webcam program reads the image in a continuous stream and saves the image to a virtual hard drive (RAM). MATLAB then takes the image and performs image processing to detect if a specified target it present. If there is a target present, RoboCat will follow the target with its head.

If the object is directly in front of the quadruped the CM730 will send a signal to the PIC Control board to generate a walking sequence to be sent continuously to the Pololu Maestro servo motor controller via UART. The servo motor controller than processes the signal and sends the appropriate PWM signal to each of the 11 servo motors.

The off-board control consists of a laptop and a remote wireless receiver. RoboCat has two on-board cameras. Currently one is used for image processing and the other is sent wirelessly to an off-board display. The image from the second camera is sent to a 2.4 GHz wireless AV Receiver, in which the video stream can be sent to a standard television via composite video. In the future, both cameras will be used for stereo vision so that RoboCat can perceive depth. The off-board computer is used to send various command signals to the RoboCat. For example walk with vision tracking or without vision tracking. It also gives the user full access to the CM730 through the remote desktop.

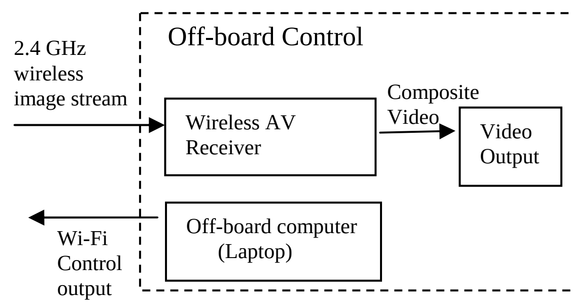


Figure 16. Block diagram defining the off-board communications stream of RoboCat.

CONCLUSION

In this paper a biomimetic quadruped robot was presented which employs an autonomous vision guidance system. The quadruped has been developed as an open architecture so that new target recognition algorithms and decision making algorithms can easily be implemented to increase the quadruped robot's intelligence. Future applications for RoboCat include: mobile surveillance, explosive ordinance disposal, victim detection in casualty situations along with military applications such as target detection and removal. A possible consumer application is to implement RoboCat as a mobile household surveillance system.

Although it was not discussed previously, RoboCat began as an aerial robotic transformer project in which there were hopes of RoboCat being capable of transforming into an aerial vehicle. This would be useful in a situation such as a burning building which it would be dangerous for firefighters to enter a second story; RoboCat would fly up to the second story window and search for victims. Although this is futuristic, RoboCat is capable of walking and detecting faces and therefore would be capable of finding casualty victims

The current state of development of RoboCat is that it can successfully track objects based on color. The following links show RoboCat tracking an orange fish cutout by walking toward it and following it with its head:

http://www.youtube.com/watch?v=sZZpKn_nDIc
http://www.youtube.com/watch?v=ZCydQB9Vyfo

The intelligent decision making and vision algorithms have been developed on a separate platform and will be tested on RoboCat in the future. The next step is to test RoboCat's tracking capabilities with the new shape and face detection algorithms discussed in this paper. As RoboCat's intelligence grows, the intelligent decision making algorithms will play a larger part in its autonomous capabilities such as finding a power source when it is low on power, playing with a toy ball when it feels playful, finding shelter when it is raining etc.

In conclusion, we created a quadruped robot that embodies an embedded computer which can easily implement new vision algorithms written in MATLAB, C, C++, or python and currently RoboCat has color, face, and shape detection. Ultimately, the vision algorithms will be used to employ RoboCat's intelligent decision making.

REFERENCES

[1] T. Miyashita, K. Hosoda, M. Asada, "Reflective walk based on lifted leg control and vision-cued swaying control," in *Proc. of Int. Sym. on Climbing and Walking Robots*, 1998, pp. 349-354.

[2] J. McClung, M.Cutkosky, J. Cham, "Rapid Maneuvering of a biologically Inspired Hexapedal Robot", *ASME International Mechanical Engineering Congress and Exposition*, Nov. 2004.

[3] J. McClung, III, "Techniques for Dynamic Maneuvering of Hexapedal Legged Robots" P.H.D Dissertation, Stanford University, Dec. 2006.

[4] R. C. He "Stereo Vision Mapping with Unsynchronized Cameras," M.S. thesis, Massachusetts Inst. of Tech., June 2008.

[5] T.-c. Chiueh, R. Krishnan, P. De, and J.-H. Chiang, "A networked robot system for wireless network emulation," in *RoboComm '07: Proceedings of the 1st international conference on Robot communication and coordination.* Piscataway, NJ, USA: IEEE Press, 2007, pp. 1–8.

[6] J.E. Laird, A. Newell, and P.S. Rosenbloom, "Soar: an architecture for general intelligence," Artificial Intelligence, vol. 33, pp. 1-64, 1987.

[7] J.E. Laird, "Extending the Soar cognitive architecture," in Artificial General Intelligence 2008, Memphis, TN: IOS Press, 2008, pp. 224-235.

[8] R. Sun, "The CLARION cognitive architecture: extending cognitive modeling to social simulation," in Cognition and Multi-Agent Interaction, R. Sun, Ed. New York, NY: Cambridge University Press, 2006.

[9] P. Langley, "An adaptive architecture for physical agents," in Proc. 2005 IEEE/WIC/ACM Int. Conf. Intelligent Agent Technology, Compiegne, France, 2005, pp. 18-25.

[10] U. Ramamurthy, B.J. Baars, S.K. D'Mello, and S. Franklin, "LIDA: a working model of cognition," in Proc. 7th Int. Conf. Cognitive Modeling, Trieste, Italy, 2006, pp. 244-249.

[11] J. Bach. *Principles of Synthetic Intelligence*. Oxford University Press, New York, 2009.

[12] J. A. Starzyk, J. T. Graham, P. Raif, and A-H.Tan, "Motivated Learning for Autonomous Robots Development", to appear in Cognitive Science Research, 2011.

[13] Y. Rui, T. Huang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 1, pp. 39-62, Mar. 1999

[14] Navneet Dala,(2005) "Histograms of Oriented Gradients for Human Detection", Computer Vision and Pattern Recognition, CVPR 2005, pp886~893

[15] Luke Cole, David Austin, Lance Cole (2004), "Visual Object Recognition using Template Matching", Australian Conference on Robotics and Automation

[16] Motion Detection and Segmentation Project http://www.ai.mit.edu/projects/cog/VisionSystem/motion_detection.html

[17] Iyad Aldasouqi, Mahmoud Hassan (2011), "Smart Huamn Face Detection System", International Journal of Computers, Issue 2, Volum 5, 2011

[18] Simon Haykin, "Neural Networks: A Comprehensive Foundation, 2nd Ed", Pearson Education, Delhi, 1999

[19] S. Aslan, Y. Uzer, O. Isik, M. Altun, "A simple and improvable method for face region extraction," *ISCIS '08. 23rd International Symposium on Computer and Information Sciences*, Oct. 2008

[20] M. Turk and A. Pentland, "Eigenfaces for Detection / Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991

[21] R. Feraud, O.J. Bernier, J.E. Viallet, and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 1, 2001

[22] A. Ono, "Face Recognition with Zernike Moments" Systems and Computers in Japan, Vol. 34, No. 10, 2003

[23] R. Hsu, M. Abdel-Mottaleb, A.K. Jain, "Face Detection in Color Images" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, 2002

[24] J.Y. Lee and S.I. Yoo, "An Elliptical Boundary Model for Skin Color Detection", Proc. of the International Conference on Imaging Science, Systems, and Technology, 2002

[25] A. Albiol, L. Torres, E.J. Del, "Optimum Color Spaces for Skin Detection", Proceedings of the International Conference on Image Processing, vol. 1, pp. 122–124, 2001

[26] J. Haddania, K. Faez, M. Ahmadi, "An Efficient Human Face Recognition System Using Pseudo Zernike Moment Invariant and Radial Basis Function Neural Network" *International Journal of Pattern Recognition and Artifcial Intelligence*, vol. 17, no. 1, pp. 41-62, 2003

[27] J. Haddadnia, K. Faez and P. Moallem, "Neural network based face recognition with moment invariants" IEEE Int. Conf. Image Processing, Thessaloniki, Greece, Vol. I, pp. 1018-1021, 2001.

[28] J. A. Starzyk and D. K. Prasad, "A Computational Model of Machine Consciousness" to appear in International Journal of Machine Consciousness, 2011.