Go with the Flow: Synthetic Vision for Autonomous Animated Creatures

Bruce M. Blumberg
MIT Media Lab
E15-311, 20 Ames St.
Cambridge Ma. 02139
bruce@media.mit.edu

Abstract

Real-time obstacle avoidance and low-level navigation is a fundamental problem for autonomous animated creatures. Here we present an ethologically inspired approach to this problem in which the creature renders the scene from its viewpoint (i.e. synthetic vision) and uses the resulting image to recover a gross measure of motion energy as well as other key features of its immediate environment, which are then used to guide movement. By combining this form of synthetic vision with an ethologically inspired model of action-selection, we are able to demonstrate robust obstacle avoidance and low-level navigation in Silas T. Dog, a virtual dog built by the author, when he is placed in complex scenery such as the "Doom" environment.

Introduction

Most autonomous animated creatures built to date use "direct sensing" in which sensing is performed via interrogation (or via direct access to other creature's state). While this approach is simple and fast, it has a number of limitations. First, it requires agreement on a common protocol with which to make and respond to inquiries. Second, it does not help with the problem of obstacle avoidance and low-level navigation.

A number of researchers have suggested using computer vision techniques to address the navigation and obstacle avoidance problem in computer graphics. Among them are Reynolds [Reynolds87], Renault [Renault90], and Terzopoulos [Tu94]. There are a number of motivations for doing so:

- First, it may be the simplest and fastest way to extract useful information from the environment. This may be particularly true if the system can take advantage of the underlying hardware.
- Second, synthetic vision may scale better than other techniques in complex environments.
- Third, this approach makes the creature less dependent on the underlying representation/implementation of its environment because it does not rely on other creatures and objects to respond to particular queries. All it needs to be able to do is render the world.
- Fourth, believable behavior begins with believable perception.

In this paper we discuss one approach to using synthetic vision in which we use a very approximate measure of motion energy combined with the extraction of a few simple features to guide navigation. While other authors have suggested the usefulness of optical flow for guiding movement in robots (in particular see [Duchon96]), our contribution is (a) to show how a cheap first order approximation of motion energy may suffice, and (b) to stress the usefulness of the approach as a general technique for performing obstacle avoidance and navigation in autonomous animated creatures. See [Blumberg96] for a detailed description of the work described below.

Implementing Synthetic Vision

The fundamental idea is very simple: a creature has a "vision" sensor which renders the scene using a graphics camera mounted at the position and orientation of the sensor. The resulting rgb image is extracted from the framebuffer and used as input into whatever perceptual mechanisms the creature possesses. For example, Silas has a single vision sensor which uses a perspective camera with a field of view of 90 degrees, as shown below. To aid in visual tracking, false coloring is typically used when rendering the creature's current object-of-interest. This use of false coloring is similar to the use of markers by Chapman and Whitehead [Chapman91, Whitehead92].

The Motion Energy Approach

Recovering motion energy from the visual field and using it to perform obstacle avoidance is inspired by research into bee navigation, in which it appears that bees flying down a corridor will position themselves so as to balance the motion energy in their right and left eyes [Srivansan96]. Similarly, they appear to use the perceived motion energy to control their altitude and ground speed [Mura94]. Desert ants use the perceived motion energy from their ventral retinas as input into their path integration system, which keeps track of their distance and bearing from the nest [Wehner96]. See [Duchon96] for an example of using optical flow to guide movement in a maze by a real robot.

We combine a very simple and approximate measure of motion energy with a measure of "mass" to arrive at a metric to guide low-level navigation. First, we divide the image in half. Then for each pixel in each half we calculate the following measure of energy:

$$FM_{ijt} = abs(fw \cdot (pixel_{ijt} - pixel_{ij(t-1)}) + (1 - fw) \cdot po(i, j))$$

where:

FM_{ijt} = Measure of energy at pixel(i,j) at time t

fw = flow weighting (between 0 and 1)

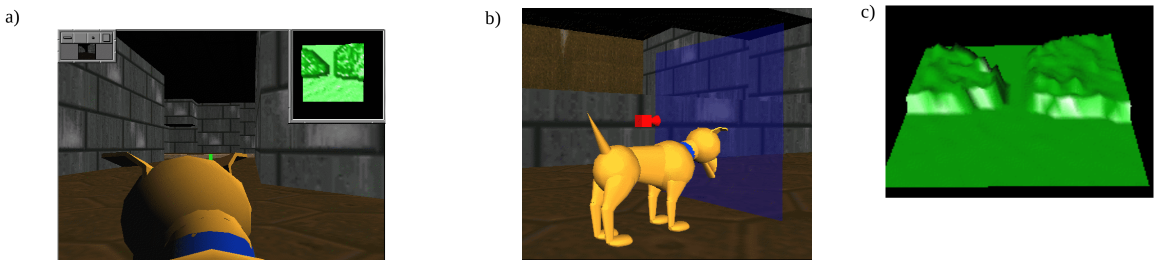
$pixel_{ijt}$ = rgb value of pixel (i,j)at time t

$po(i, j)$ = 1 if pixel non-black, 0 otherwise

The first term corresponds to the difference between successive frames, and is intended to be a gross measure of the motion energy at that pixel. The second term of our motion energy equation is not a measure of motion energy at all, but rather is added to deal with the scaling problems associated with computer-generated textures. Intuitively, it represents a binary measure of whether there is mass at that pixel. We use the coefficient *fw* to weight the respective contributions of the flow and mass. The difference between the total motion energy in the right and left halves of the retina can be used to determine the bearing to steer. That is:

$$bearingToSteer = K \cdot \left(\sum_{left} FM_{ijt} - \sum_{right} FM_{ijt} \right)$$

This simple control law will work in most cases for corridor following and obstacle avoidance. The notable exception is the case



Silas uses motion energy and other features recovered from his “vision” sensor to perform obstacle avoidance and low level navigation. In (a) we see the scene from a camera mounted on Silas’s back. The image in the window in the upper left is the actual (32x32) image rendered by the “vision” sensor using an Inventor perspective camera. The window in the upper right shows the recovered “motion energy map” (see text). In (b) we show his field of view. In (c) the “motion energy map” is displayed from a low angle to give a better sense of its structure.

in which the creature is moving directly toward a wall. In this case, the control law will tell the creature to steer straight, and the creature will eventually collide with the wall. Our solution to this problem is to keep track of the total motion energy, and when it is above some threshold and the energy is approximately the same on either side of the retina, then pick a direction to turn and turn.

In fact, we choose to augment this basic control law by extracting more information from the image and using it in conjunction with a simple reactive behavior system to provide more robust low-level navigation and obstacle avoidance. Specifically, using Horswill’s approach [Horswill93] we keep track of the height of the first non-floor pixel in each column. The resulting vector, or the “ridge” is used to recover additional information, including:

- Openings - By using a measure of the slope (i.e. the difference in heights between column$_{(i)}$ and column$_{(i+d)}$) the system identifies “left” and “right” corners. An “opening” is then defined to be a sequence of a “left” and “right” corner.
- Free Paths - The ridge can be used to see if there is a free path to a point in its visual field. The point may correspond to a pixel belonging to the object-of-interest, or it may be a point which is projected into the visual field. In either case, it tests to see if a straight line from the bottom center of the image to the point of interest intersects a non-floor pixel. If so, then the path is blocked. If not, then there is a possible free path to the object.
- Dangerously close obstacles - The system keeps track of the number of elements which are in a so-called “danger zone” at the bottom center of the image (and thus close to the creature). This is also a fail-safe in the case where the creature may be approaching a wall head-on and the motion energy on the right and left is the same.
- Free Space - If the minimum of the ridge is close to the horizon line, then for all intents and purposes the creature is in free space.

In addition to the “ridge” data described above we also keep track of the “target ridge”, which is composed of the lowest pixel in each column which is associated with the object-of-interest. This is used to answer questions such as: “is the object in my visual field”, “is there a free-path to the object”, and “how close am I to the object”.

The current approach is surprisingly robust, but it is not without its limitations. First, it assumes that objects such as walls are rendered using textures (flat, colored surfaces do not generate motion energy except at the edges). Second, the current implementation assumes that there is minimal texture on the floor. This is accomplished either by avoiding direct lighting of the floor, or by “inserting” a large constant colored polygon at floor level prior to rendering the scene. Third, it currently assumes a flat floor and will not handle over-hanging obstacles. Lastly, since it is a reactive approach to navigation, and no path planning is done, the resulting path is unlikely to be optimal.

Conclusion

Real-time obstacle avoidance and low-level navigation is a fundamental problem for autonomous animated creatures. In this paper we have presented a novel approach to this problem in which the creature renders the scene from its viewpoint (i.e. synthetic vision) and uses the resulting image to recover key features of its immediate environment, which are then used to guide movement. By combining this form of synthetic vision with an ethologically inspired model of action-selection, we are able to demonstrate robust obstacle avoidance and low-level navigation.

References

Blumberg, B.(1996). *Old Tricks, New Dogs: Ethology and Interactive Animated Characters.* Ph.D. Dissertation, MIT Media Lab.

Chapman D. (1991) *Vision, Instruction and Action*. MIT Press, Cambridge Ma.

Duchon, A.(1996). Maze Navigation Using Optical Flow. To be published in: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma.

Horswill, I. (1993). A Simple, Cheap, and Robust Visual Navigation System. In: *Second International Conference on the Simulation of Adaptive Behavior.* Honolulu, HI. MIT Press

Mura, F. and N. Franceschini. (1994). Visual Control of Altitude and Speed in a Flying Agent. In: *Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. In: *Proceedings of SIGGRAPH 87.*

Renault, O., and N. Magnenat-Thalmann, D. Thalmann (1990). A vision-based approach to behavioral animation. In: *The Journal of Visualization and Computer Animation* 1(1).

Srinvasan, M., S. Zhang, M. Lehrer, and T. Collett (1996). Honeybee Navigation en route to the Goal: Visual Flight Control and Odometry. *Journal of Experimental Biology.* 199, Jan. 1996.

Tu, Xiaoyuan and D. Terzopoulos (1994). Artificial Fishes: Physics, Locomotion, Perception, Behavior. In: *Proceedings of SIGGRAPH 94.*

Wehner R., B. Michel and P. Antonsen (1996). Visual Navigation in Insects: Coupling of Egocentric and Geocentric Information. *Journal of Experimental Biology.* 199, Jan. 1996.

Whitehead S.D. (1992). *Reinforcement Learning for the Adaptive Control of Perception and Action*. Technical Report 406, University of Rochester Computer Science Dept.