Analyzing Terrain Surfaces to Synthesize and Visualize Optimal-Coverage Tractor Paths for Conservation Farming

Category: Research

Abstract— There are competing economic goals when plowing a farm field. The rows must be level, to reduce erosion. The rows should also perfectly abut each other, so that no land is wasted on gaps and no fuel is wasted on overlapping areas. With todays GPS-guided tractors, it has become worthwhile to try to produce an optimal tractor path, knowing that a tractor can actually be programmed to follow it. This paper describes the use of visualization methods to analyze a farm field represented by a topologically organized neural network characterized by a shunting neural equation. A complete coverage tractor path is autonomously generated from the dynamic activity landscape of the neural network using GPU programming and framebuffer computing. The effectiveness of the proposed approach is verified through computer simulation.

Index Terms—Farming, GPU, neural networks, terrain surfaces.

1 INTRODUCTION

The field of farming called "Conservation Plowing" seeks to maximize crop production while minimizing both the energy needed to produce it and the waste due to soil erosion. This concept is not new – the Phoenicians originally developed the concept of plowing furrows to follow the field elevation contours in order to retain water and reduce erosion [25]. But, from visualization we know that evenly-spaced contour elevation values don't generally produce geometrically evenly-spaced contour lines. If successive contour lines are "too far" apart, gaps in field coverage will occur. If they are "too close" together, plowed areas of the field will overlap and waste fuel. Also, too much overlap between passes results in excessive soil compaction, and reduces field productivity [1].

For many years, this process has been done by a human tractor driver eyeballing the terrain. This is easiest if the field is relatively flat, which makes the overlap issues much more important than the contour-following issues. But, farmers no longer have the luxury of only growing crops on flat fields. In the effort to maximize land use, hilly crop fields are being pressed into service. This increases the importance of following the hill contours too.

Exactly following the hill contours would be difficult to do by hand. Now, however, GPS-controlled tractors are becoming common. A tractor GPS unit consists of three GPS receivers on the vehicle and one on a nearby base station [2]. This setup gives the tractor position to within an inch of actual. This means that, if an optimal path can be produced, a tractor can be programmed to follow it. The trick is in trading off the need to follow contours versus the need to avoid gaps and overlaps.

2 PREVIOUS WORK

Thus coverage path planning becomes a fundamentally important issue in farming, as it has long been for the greater field of robotics. Complete coverage path planning (CCPP) of robots is a special type of path planning in a two-dimensional environment, which requires the robot path to pass through every area in the workspace.

There are a lot of studies on the path planning for robots using various approaches. Some of the early models deal with static environments only, some suffer from undesired local minima. Most commonly applied to cleaning robots (think Roomba), many other robotic applications also require complete coverage path planning, e.g., vacuum robots [28], painter robots [5], autonomous underwater covering vehicles [12], de-mining robots [9], land mine detectors [5], lawn mowers [6], automated harvesters [7], agricultural crop harvesting equipment [8], and window cleaners [9]. Autonomous coverage robots are particularly useful in hazardous environments. There have been many studies on CCPP using various approaches, e.g., artificial potential field [22], approximate cellular decomposition, exact cellular decomposition, template-based model [14], neural networks, and fuzzy logic.

Approximate cellular decomposition models generally decomposed the workspace into discrete cells [19] or grids [30], while a recently proposed model subdivides the workspace into discrete cells and following a spanning tree of a graph induced by the cells [8]. There are many exact cellular decomposition based approaches to CCPP as well [6, 3]. The fundamental concept is to decompose the workspace into a collection of nonoverlapping cells, and then, the robot searches the connectivity graph that represents the adjacency relation among cells. Thus the complete coverage can be achieved by back and forth robot motions.

Neural network approaches introduce backpropagation [24] and learning [28, 20], but due to their computational complexity, have difficulty dealing with unstructured environments.

Fuzzy logic based methods [7, 16] can be employed for CCPP, but due to the difficulty in defining suitable fuzzy rules, the generated paths are generally not smooth enough at turning and traversing.

Various other approaches were also proposed for CCPP, such as approaches based on covering salesman problem (CSP) [4], and using heat trails as short-lived navigational markers [23].

Glasius et al. [10] proposed a neural network model for real-time trajectory formation with collision free in a nonstationary environment. However, this model suffers from slow dynamics and cannot perform properly in a fast changing environment [10]. Inspired by Hodgkin and Huxleys [13] membrane equation and the later developed Grossbergs shunting model [11], Meng and Yang [18, 27] proposed a neural network approach to dynamical trajectory generation with collision free in an arbitrary environment. These models are capable of planning real-time optimal path in non-stationary situations without any learning process. In later work Yang and Meng enhanced their model to take into account the clearance from obstacles [26], which is demanded in many situations.

Obstacle clearance can be very important in path planning. Many models for path planning concentrate on minimizing the distance between the starting position and target (e.g., [17], [10, 18, 27]) In a static environment, the path planned by models in [10, 18, 27] has the shortest distance as well, although they do not explicitly minimize any cost functions. They assume that the shortest path is the "best" path. The obstacle clearance is not considered during the path planning. Therefore, the path clips the corners of obstacles and runs down the edges of obstacles. This is the so called "too close" problem [29]. Such a "too close" problem can be avoided by expanding the obstacles by an extra size, but some possible solution paths are blocked. This strategy is not acceptable, particularly when all the possible solution path are blocked after the expanding. On the other hand, some models (e.g., [15]) maximize the clearance from obstacles while minimizing the distance from the starting position to the target. The found path passes through the middle of free space [15]. Therefore it may deviate significantly from the shortest path. This is the so called too far problem [29]. Several models were proposed to reduce or solve the "too far" and "too close" problems. For example, Zelinsky [29] proposed a path transform model for finding a neither too far nor too close path in a static environment by combining the distance transform and the obstacle transform.

In this paper, based on our adaptation of the shunting model developed by Yang and Meng [26], we propose a neural network model for real-time contour following coverage path generation on an arbitrary terrain. The dynamics of each neuron is characterized by a shunting equation derived from Hodgkin and Huxleys [13] membrane model for a biological neural system. There are only local lateral connections among neurons. The varying environment is represented by the dynamic activity landscape of the neural network. The optimal realtame path is planned through the dynamical neural activity landscape. The optimality in the real-time path planning with safety consideration is in the sense of a continuous, smooth path toward the objective of complete coverage. The model algorithm is computationally simple. The proposed model is capable of planning real-time complete coverage paths with obstacle avoidance in an unstructured environment. The term "real-time" is in the sense that the coverage path planner responds immediately to the dynamic environment including the robot, targets (uncovered areas) and obstacles. It is the first time that a nonlearning based neural network approach is developed for real-time contour farming.

3 Model

The fundamental concept of the proposed model is to develop a neural network architecture, whose dynamic neural activity landscape represents a static farm field. Fortunately, the neural shunting model developed by Yang and Meng, is readily adapted to this purpose. The real-time collision-free tractor motion is planned based on the dynamic activity landscape of the neural network and the previous tractor location, such that all areas of the field will be covered.

The 2-D Cartesian workspace in the proposed approach is discretized into squares as in most CCPP models. Unlike other approaches the diagonal length of each discrete area is not equal to the robot sweeping radius, as this would incur inefficient overlap in the tractor coverage.

When the tractor is moved, the direction is determined by a local sampling of the dynamic activity landscape, the "plow" endpoints are then found, and the coverage path is "swept" from its previous position, all discrete positions encompassed by this motion are then considered "covered."

The proposed neural network model is expressed topologically in a discretized workspace W. The location of the i-th neuron in the state space S of the neural network, which is denoted by a vector $q_i \in R^2$, uniquely represents an area in W.

Each neuron has local lateral connections to its neighboring neurons that constitute a subset R_i in S. The subset R_i is called the receptive field of the i-th neuron in neurophysiology. The neuron responds only to the stimulus within its receptive field. Thus the dynamics of the i-th neuron in the neural network can be characterized by a shunting equation as.

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e(t) - (D + x_i)S_i^i(t) \tag{1}$$

Where A represent the passive decay rate of neural activity, which solely determines the transient response to an input signal. The steady state-neural activity is nonlinearly dependent on the value of A. Smaller values of A result in a slower passive decay of neural activity. When the value of A is too small neural activity saturates and the model cannot function.

Functions $S_i^e(t)$ and $S_i^i(t)$ represents the excitatory and inhibitory inputs to the shunting model (1), respectively, and are defined as:

$$S_i^e(t) = \left(\sum_{j=1}^{k} w_{ij}[x_j]^+ + [I_i]^+ \right) \tag{2}$$

$$S_i^i(t) = \left(\sum_{j=1}^{k} v_{ij}[x_j - \sigma]^+ + [I_i]^- \right) \tag{3}$$

Parameter σ is the threshold of the inhibitory lateral connection. The threshold of the excitatory connection is chosen as zero. The value k represents the number of adjacent neurons, in our case $k = 8$. The weight w_{ij} represents the the excitatory connection from the i-th to the j-th neuron, and similarly v_{ij} represents the inhibitory connection weight and is dependent on w_{ij} as shown:

$$w_{ij} = f\left(|q_i - q_j|\right) \tag{4}$$

$$v_{ij} = \beta w_{ij} \tag{5}$$

where β is a positive constant, $\beta \in [0,1]$, and $|q_i - q_j|$ represents the Euclidean distance between vectors q_i and q_j in the state space.

Functions $[a]^+$ and $[a]^-$ are linear-above threshold functions.

$$[a]^+ = \max\{a, 0\} \tag{6}$$

$$[a]^- = \max\{-a, 0\} \tag{7}$$

Function $f(a)$ can be any monotonically decreasing function, for instance

$$f(a) = \mu/a \tag{8}$$

for all adjacent neurons, where the parameter μ determines the connection weight for adjacent neurons. A small value of μ results in weak lateral connections and can prevent possible saturation in neural activity. When $\mu > 1$, the propagated neural activity is amplified and the neural activity can easily saturate. Therefore, to prevent possible neural activity saturation a smaller μ is necessary; to strengthen the influence from the target, a larger μ is needed. Usually μ is usual chosen in the range $\mu \in (0,1]$.

External Input As mentioned previously the dynamics of the neural network are controlled though manipulation of the external input I_i, for example defining the external input as:

$$I_i = \begin{cases} -E, & \text{if it is in an obstacle area} \\ E, & \text{if it is in the target area} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

we generate a neural activity field, by which following steepest gradient ascent will lead to the target region, which is appropriate for goal directed path planning, but inappropriate for coverage planning.

Alternately by simply changing the criteria for the target area to include all uncovered areas as:

$$I_i = \begin{cases} -E, & \text{if it is in an obstacle area} \\ E, & \text{if it is in an uncovered area} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

we can guarantee that the neural activities of the uncovered areas are going to stay at the peak of the activity landscape while the obstacles are going to stay of the bottom. Note that parameter E determines the amplitude of the external inputs from the target and obstacles, thus E should be chosen as very large over the sum total of input from all adjoining lateral connections, i.e. $E \gg \sum_{j=1}^{k} w_{ij}$.

Finally by adding further refinement to the external input as:

$$I_i = \begin{cases} -E, & \text{if it is in an obstacle area} \\ h(u_i), & \text{if it is in the uncovered area} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

we define an activity landscape by which regions of the of the terrain of similar height, (same contour), attract the tractor most, where u_i corresponds to the field position of the i-th neuron and T represents the tractor position. The height function is essentially just a triangular function modulating the amplitude of the neural activity based on the absolute difference in elevation from the tractor's location on the terrain.

$$h(u_i) = E - \frac{|u_i.z - T.z|}{\rho} \tag{12}$$

where ρ represents the the size of the vertical range from which to accept excitatory input, $\rho \in [0,1]$.

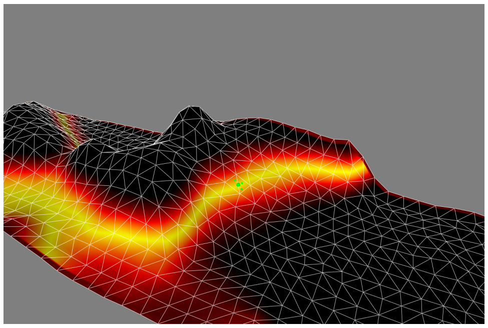

Fig. 1. Contour based neural activity with $\rho = 0.5$, all neurons corresponds to regions of the terrain of similar height to that of the tractor's current position receive the greatest excitatory input.

The proposed neural network characterized by (1) guarantees that the positive neural activity can propagate to the whole state space, while the negative activity is restricted locally, due to the existence of the threshold σ of the inhibitory lateral connections.

Therefore, the contour region *globally* influences the whole state space to attract the tractor, while the obstacles have only local affect to avoid collision. By adjusting the μ_{in} and σ values we can adjust the local influence from the obstacles and coverage regions to approximate the width of the tractors plow, providing an elegant mechanism to minimize overlap.

Since the neural activity is bounded in $[-D,B]$, when $\sigma < -D$, the inhibitory lateral connection term $\sum_{j=1}^{k} v_{ij}[x_j - \sigma]^+$ in (3) is equal to zero. Note that when $\beta = 0$, i.e., $v_{ij} \equiv 0$, this term becomes zero as well. Thus no negative neural activity is able to propagate to the other neurons.

4 Stability

In order to prove the stability and convergence of the proposed model we must satisfy all the three conditions required by Grossberg's general form (1) positivity, (2) symmetry, and (3) monotonicity. So first, from the definitions of $[a]^+$, $[a]^-$ and v_{ij}, we rewrite (1) into Grossberg's general form,

$$\frac{dy_i}{dt} = a_i(y_i)\left(b_i(y_i) - \sum_{j=1}^{k} c_{ij} d_j(y_j)\right) \tag{13}$$

by the following substitutions:

$$a_i(x_i) = \begin{cases} B + x_i, & \text{if } x_i \geq 0 \\ D - x_i, & \text{if } x_i < 0 \end{cases} \tag{14}$$

$$b_i(x_i) = \frac{1}{a_i(x_i)}\left(B[I_i]^+ - D[I_i]^- - (A + [I_i]^+ + [I_i]^-)x_i\right) \tag{15}$$

$$D_{ij} = -w_{ij} \tag{16}$$

$$d_j(x_j) = \begin{cases} x_j, & \text{if } x_i \geq 0 \\ \beta(x_j - \sigma), & \text{if } x_i < \sigma \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Since B and D are positive constants, then $a_i(x_i) \geq 0$ satisfying the positivity condition. Since $w_{ij} = w_{ji}$, then $c_{ij} = c_{ji}$proving symmetry. Finally, since $d_j'(t) = 1$ at $x_j > 0$, $d_j'(t) = \beta \geq 0$ at $y_i < \sigma$, and $d_j'(t) = 0$, otherwise, then $d_j'(t) \geq 0$ proving monotonicity.

$$d_j'(t) = \begin{cases} 1, & \text{if } x_j > 0 \\ \beta & \text{if } y_i < \sigma \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

Therefore, (1) satisfies all the three conditions required by Grossberg's general form. The rigorous proof of the stability and convergence can be found in [11]. Thus, the proposed neural network system is stable, and the dynamics of the network are guaranteed to converge to an equilibrium state of the system [11].

If the excitatory and inhibitory connections in the shunting equation in (1) are lumped together and the auto gain control terms are removed, then a simpler form can be obtained from (1)

$$\frac{dx_i}{dt} = -Ax_i + I_i + \sum_{j=1}^{k} w_{ij}[x_j]^+ + \sum_{j=1}^{k} v_{ij}[x_j - \sigma]^+ \tag{19}$$

This is an additive equation as seen in [10] The nonlinear functions $[a]^+$, $[a]^-$ and the threshold σ together guarantee the positive neural activity can propagate to the whole workspace while the negative activity can propagate locally in a small region only. From the definitions of $[a]^+$, $[a]^-$ and v_{ij}, (19) can be rewritten into a compact form as:

$$\frac{dx_i}{dt} = -Ax_i + I_i + \sum_{j=1}^{k} w_{ij} d\left(x_j\right) \tag{20}$$

5 Implementation

The discrete topologically organized map of the proposed neural network architecture is easily mapped to a discrete two dimensional texture. Lateral connections between neurons become adjacent connections to neighboring texels. The advantages of this mapping are many, (1) it allows us a simple way to discretize our field's terrain, and (2) it allows us to take advantage of the massive parallelism found in modern GPU's to implement the shunting equation efficiently for large networks.

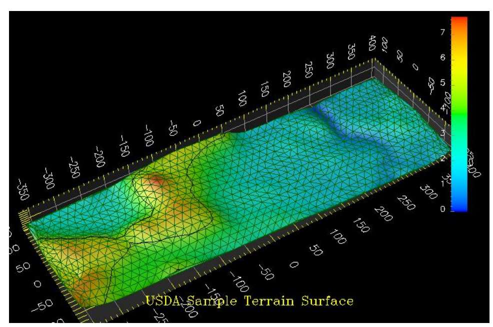

Fig. 2. The terrain data arrives as an untriangulated collection of scattered Digital Elevation Map (DEM) points.

5.1 Discretization

The first step is the discretization of our terrain mesh, which we assume to be vertically convex with height defined along the Z-axis. To accomplish this we set up an orthographic projection of the terrain bounded by its maximum and minimum X and Y elements. Through the use of a simple fragment shader, we render the $\langle x,y,z \rangle$ coordinates of the terrain to the $\langle r,g,b \rangle$ color channels of a floating point texture bound to the framebuffer, see figure 3. It is worth noting that it is useful to setup this projection such that an obstacle boundary surrounds the terrain.

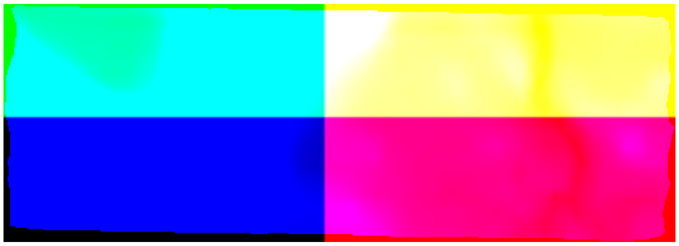

Fig. 3. A discretized farm field rendered to a texture, the $\langle x,y,z \rangle$ coordinates of the terrain are mapped to the $\langle r,g,b \rangle$ color channels of a floating point texture.

5.2 Ping-Ponging for Neural Activity Propagation

A further advantage of the shunting equation is that it is easily implemented in an iterative manner. This allows us to implement the shunting procedure in a fragment shader, by employing a framebuffer computing technique known as "ping-ponging." By binding two textures to a framebuffer, we alternately render the propagation of neural activity from one texture (the ping texture) as input to the other (the pong texture), before swapping the framebuffer's texture targets and repeating the process. Each texture is thus a representation of the state of the neural activity of the network, either at time t or $t+1$, this technique allows the allows the discretized neural activity field to progress in time.

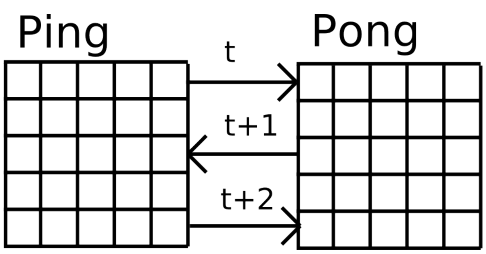

Fig. 4. Ping & Pong represent two textures bound to the framebuffer, they are alternately the input to and output of the shunting equation, allowing the neural activity to be iteratively propagated through time.

5.3 Tractor

Unlike previous approaches, we do not employ a point robot to sample the terrain. As a consequence of this, we do not discretize the tractor's direction. Instead, after sampling the direction at the tractors "center," we find the corresponding perpendicular directions with respect to the terrain's normal, and traverse the terrain in those directions in order to find the end point of our tractor "plow." Similarly, we trace forward to sample additional points ahead of the tractor. These points can then be sampled in the same manner as the tractor's center position. After each iteration of tractor movement the tractor position plow endpoints are stored, and the newly swept region is rendered into the coverage texture, see figure 5.

5.4 Coverage

In addition to textures representing the discretized 3-D terrain surface, and the neural activity of the field, we maintain an additional texture representing the covered and uncovered regions of the terrain. As we update the tractor's position we render each newly swept region into the coverage texture, thereby accumulating the tractor's accrued coverage of the farm field, see figure 6 for an example.

5.5 Sampling

At this point we have a texture representing the field, the neural activity, and the coverage. However, the location of our tractor still resides on the surface of a 3-D mesh, must devise an approach by which to accurately, and efficiently receive this texture data. Again, we employ

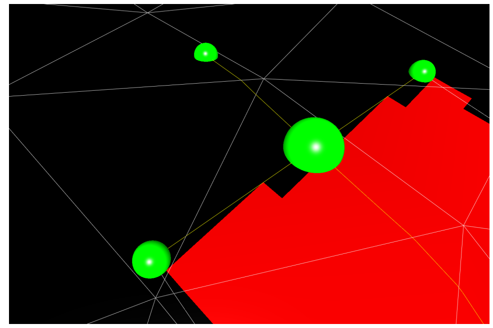

Fig. 5. Our simplified model of a tractor, the large sphere represents the the center of the tractor, while the smaller two the left and right represent the endpoints of the "plow," the forward sphere indicates direction, and can be used for sampling of data ahead of the tractor. The red areas indicate "covered" terrain.

Fig. 6. Sample contour-directed coverage texture, covered areas are in red, obstacles are in green, and black represents uncovered terrain.

framebuffers, but in this instance the attached texture is merely a single pixel, we pass the tractors 3-D position into a fragment shader, in addition to the maximum and minimum values of the terrain, and the resolution of the texture. This provides us with enough information to resolve the tractors position in texture coordinates. By providing our sampling fragment shader access to the previously defined textures, we can sample adjacent texels for their information. In contrast to discrete methods, we employ a Sobel filter in order to determine the tractors optimal direction, rendering the neural activity and direction components returned by the Sobel filter into the attached texture. When the neural activity at the current location is larger than that of its neighbors the tractor waits for neural propagation to indicate a clear direction of travel.

5.6 Direction

In the proposed CCPP model, the tractor path is generated from both the activity landscape and the previous tractor location, see figure 10, and figure 8 for images of a tractor headed toward its goal. Since we have described external inputs for goal-directed path, and coverage-directed paths, it should be noted that the output of the Sobel must be rotated 90° for the contour coverage directed approach approach to function, in this way, the tractor will run perpendicular to the gradient of the activity field, see figure 1 for an example.

6 Visualization

Processing the neural network on the GPU lends itself to efficiently generating images by which to analyze the neural activity field.

6.1 Vertex Displacement Graph

By creating a grid with vertices corresponding to each neuron of the activity field, using a Vertex shader we can displace the vertices of this grid by the neural activity at each position in the texture. In this manner we dynamically generate a real-time graph of neural activity for analysis of the network.

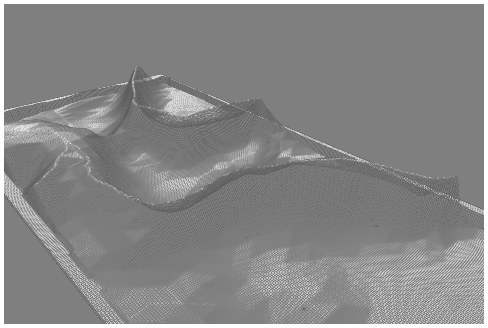

Fig. 7. A graph of the neural activity field created with a vertex displacement shader from the activity texture.

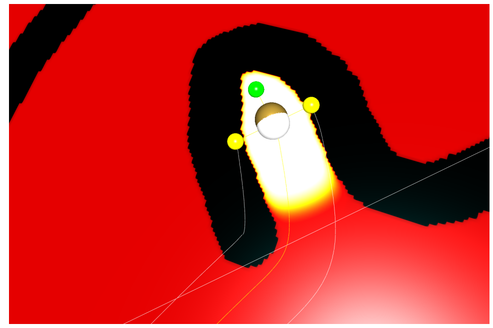

Fig. 10. A goal directed activity field rendered in heated object color scale, with a tractor travelling towards the goal.

6.2 Contour Lines

By employing a simple fragment shader we can generate smooth anti-aliased contour lines based on the neural activity stored in the textures, the height of the farm field, or the height of the activity graph. Line sharpness is easily controlled via an exponential impulse function. See figure 8 for an example on a goal directed field, and figure 9 for an example on a contour coverage based activity field.

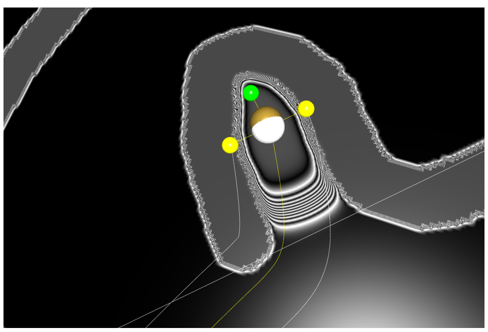

Fig. 8. Contour lines on a goal-directed activity field with obstacles.

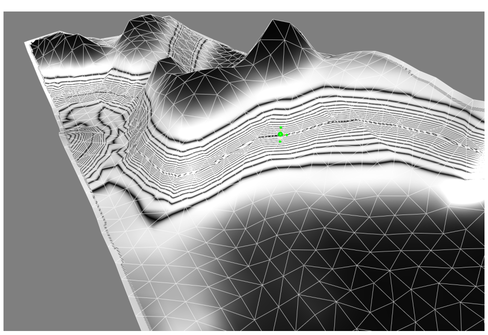

Fig. 9. Contour lines on a contour coverage based activity field without obstacles.

6.3 Heat Object Scale

Another useful method by which to visualize neural activity is to display neural activity using a heated object color model, again this is easily implemented in shader, and easily applied to field, the activity texture, and the activity graph. See figure 10 for an example on a goal directed field, and figure 1 for an example on a contour coverage based activity field.

7 Simulations

The proposed neural network model is capable of generating many tractor trajectories. In this section the proposed model is applied to a tractor on a sample farm field. Various parameters and external input models are shown for comparison.

Goal   By employing the external input as defined in (9), the proposed shunting model we generate a neural activity field which globally attracts the tractor towards the target region through steepest gradient ascent, see figure 10.

Coverage   By changing the criteria for excitatory input to be generated in all uncovered areas as in (10), the shunting model generates a neural activity field which will attract the tractor to all uncovered areas until a coverage path has been obtained.

Contour Coverage   Finally, by varying excitatory input to the neural activity field by a height function based on the elevation difference between the tractor's current location and the terrain elevation, our model generates an activity field which attracts the tractor to follow contour lines. Once a contour line has been covered by the tractor it no longer generates neural activity, thereby encouraging the tractor to seek out new contour elevations and consequently shifting the networks excitatory input to match the tractor elevation on the field, see figure 1 for an example.

8 Conclusion

In this paper, a biologically inspired neural network approach is proposed for the dynamic tractor coverage path generation in an arbitrary terrain surface. Several model variations are presented and the differences are compared by descriptive analysis and simulation studies. The proposed approach is applied to the real-time coverage path planning for a tractor on a static terrain. The optimal real-time trajectory is generated through the dynamic neural activity landscape that represents the farm field environment. The stability and convergence of the proposed models are guaranteed by a qualitative analysis and a rigorous Lyapunov stability analysis.

Some points are worth mentioning about the proposed neural network approach to dynamic collision-free coverage path generation.

- This paper presents a new visual approach to analyzing terrain surfaces, through the use of framebuffer computing and GPU programming.

- This model is biologically plausible. It is originally derived from Hodgkin and Huxleys biological membrane model [13]. The neural activity is a continuous analog signal and has both upper and lower bounds.

- The model algorithm is computationally efficient. The optimal tractor coverage path is generated without explicitly searching over the free workspace or the collision paths, without explicitly optimizing any global cost functions, without any prior knowledge of the dynamic environment, and without any learning procedures.

- The computational complexity linearly depends on the state space size of the neural network. Each neuron in the neural network has only local lateral connections, which does not depend on the size of the overall neural network. Utilization of commodity graphics hardware has allowed us to maintain real-time results on far larger networks than previous implementations.

- This model can perform properly in an arbitrarily dynamic environment, even with sudden environmental changes, such as suddenly adding or removing obstacles or targets. The neural network system is characterized by a continuous shunting model, it is stable and keeps sensitive to variations in the environment [11]

- The proposed model is capable of generating real-time collision-free trajectories of an agent with multiple moving targets and the trajectories of multiple robots in a common workspace.

- By choosing suitable strength of obstacle clearance the proposed model can plan a path that allows tractor passes to comfortably abut each other. Thus it does not suffer from either the "too close" (narrow safety margin) or "too far" (waste) problems [21, 29, 31].

References

[1] *Western Farm Press*, March 2000.

[2] *Farm Industry News*, May 2003.

[3] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002.

[4] Arkin and Hassin. Approximation algorithms for the geometric covering salesman problem. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 55, 1994.

[5] P. Atkar, H. Choset, A. Rizzi, and E. Acar. Exact cellular decomposition of closed orientable surfaces embedded in $\mathfrak{R}^3$. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 1:699–704, 2001.

[6] H. Choset and P. Pignon. Coverage path planning: The boustrophedon cellular decomposition, 1997.

[7] Y. Fu and S. Lang. Fuzzy logic based mobile robot area filling with vision system for indoor environments. *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 326–331, 1999.

[8] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2:1927–1933, 2001.

[9] D. Gage. Randomized search strategies with imperfect sensors, 1993.

[10] R. Glasius, A. Komoda, and S. C. A. M. Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8(1):125–133, 1995.

[11] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[12] S. Hert, S. Tiwari, and V. Lumelsky. A terrain-covering algorithm for an auv, 1996.

[13] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. 117:500–544, 1952.

[14] C. Hofner and G. Schmidt. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, 1:610–617, 12-16 Sep 1994.

[15] J. Ilari and C. Torras. 2d path planning: a configuration space heuristic approach. *Int. J. Rob. Res.*, 9(1):75–91, 1990.

[16] S. Lang and B.-Y. Chee. Coordination of behaviours for mobile robot floor cleaning. *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, 2:1236–1241, 13-17 Oct 1998.

[17] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, 32(2):108–120, 1983.

[18] M. Meng and X. Yang. A neural network approach to real-time trajectory generation [mobile robots]. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 2:1725–1730, 16-20 May 1998.

[19] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2:116–121, Mar 1985.

[20] F. Muiz, E. Zalama, P. Gaudiano, and J. Lpez-Coronado. Neural controller for a mobile robot in a nonstationary environment.

[21] H. Noborio, T. Naniwa, and S. Arimoto. A feasible motion-planning algorithm for a mobile robot based on a quadtree representation. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 327–332 vol.1, 14-19 May 1989.

[22] A. Pirzadeh and W. Snyder. A unified solution to coverage and search in explored and unexplored terrains using indirect control. *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 2113–2119, 13-18 May 1990.

[23] R. Russell. Heat trails as short-lived navigational markers for mobile robots. *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, 4:3534–3539, 20-25 Apr 1997.

[24] P. Tse, S. Lang, K. Leung, and H. Sze. Design of a navigation system for a household mobile robot using neural networks. *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, 3:2151–2156, 4-9 May 1998.

[25] Wikipedia. Contour plowing — wikipedia, the free encyclopedia, 2008. [Online; accessed 26-March-2008].

[26] S. X. Yang and M. Q.-H. Meng. An efficient neural network method for real-time motion planning with safety consideration. *Robotics and Autonomous Systems*, 32(2-3):115–128, 2000.

[27] X. Yang and M. Meng. Dynamical trajectory generation with collision free using neural networks. *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, 3:1634–1639, 13-17 Oct 1998.

[28] F. Yasutomi, M. Yamada, and K. Tsukamoto. Cleaning robot control. *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, 3:1839–1841, 24-29 Apr 1988.

[29] A. Zelinsky. Using path transforms to guide the search for findpath in 2d. *IJRR*, 13:315–325, 1994.

[30] A. Zelinsky, R. Jarvis, J. Byrne, and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot, 1993.

[31] D. Zhu and J.-C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *Robotics and Automation, IEEE Transactions on*, 7(1):9–20, Feb 1991.