Algorithms for Stereo Image Creation from a Sequence of Two Dimensional Images

Identification and Significance of the Problem

Ground-based space surveillance sensors capture sequences of images of a satellite. As the satellite passes overhead, each image is taken from a slightly different angle. As a result, it should be possible to select pairs of images that have appropriate degrees of disparity to form stereo pairs, allowing a viewer to see the satellite three-dimensionally. This feature would enhance the viewer's ability to discern the satellite's shape and features, and aid in identification of the satellite's function. The problem, however, is that many satellites do not maintain a consistent orientation relative to the earth, but rather exhibit tumbling or slewing motion. As a result, the task of selecting stereo pairs requires not only choosing the optimum amount of disparity, but also finding pairs with matching orientation. Existing methods for achieving this require humans to search through the images for matching pairs, which is too time-consuming, subjective, and error-prone. What is needed is a computer algorithm that can search for stereo pairs with minimal supervision. Assuming that the rotation rate and image frame rate are consistent, once a suitable pair has been identified, the offset should apply to all image pairs, allowing stereo pairs to be viewed in sequence so that tumbling and slewing can be observed.

<u>Analysis of the Problem.</u> Image analysis has long been a challenge for computer algorithms. There are several issues that must be addressed in order to find an ideal solution.

1. Solving the correspondence problem.

   When two disparate views are to be matched, the positions of features differ between the views. It is not just a matter of matching identical features, but of determining what goes with what. This problem is complicated by the fact that the two images are seen from different angles. As a result, the relative positions of features differ between the two images.

   Humans are able to solve the correspondence problem easily. Disparity can be detected readily even in random-dot stereograms, which have no discernable features in a single image, but which differ only by having a portion of the array of dots shifted in position between the left- and right-eye views.

   This ability to detect disparity is attributed to disparity detector cells in the primary visual cortex. These disparity detectors respond maximally when stimulation in a given receptive field of one eye occurs at the same time as stimulation in a horizontally shifted receptive field of the other eye. (Hubel & Weisel, 1962).

   The most widely accepted computational models of disparity are based on facilitation among cells responding to stimulation in the same position in the same depth plane, and inhibition of cells responding to different depth planes (Blake & Wilson, 1991; Coren, Ward, & Enns, 2004).

   The present task involves not so much detecting and interpreting disparity between individual features as determining whether an appropriate level of disparity exists overall—such as assigning a score describing the adequacy of the image pair to produce stereopsis.

2. Determining the rate of rotation, or finding frames with matching orientations

   The object may be tumbling or slewing in space, so that pairs of images may not match in orientation. A complicating factor here is again that disparity is present in the images, which means that the ideal pair of images will not match point by point. Image comparisons require identifying a frame in which the object has returned to its original orientation relative to the earth, but because the image is taken from a different angle the features will not be in the same positions in relation to each other.

Two possible approaches could be used to identify appropriate matches. The first would be to search for image pairs that have matching features with an appropriate level of disparity. This approach could easily lead to false-alarm matches, however, especially for a symmetrical object that might have multiple views that appear similar. If the object is completely symmetrical, such a solution might be acceptable. If the object is not entirely symmetrical, however, it could lead to stereo pairs that are confusing to observe. (For example, if it is symmetrical except that it has a window or an antenna on one side, that feature would appear in only one of the images in the pair.)

A second approach would be to measure the rate of rotation, and to use that information to predict which frame will contain a suitable orientation for a stereo pair. Doing this with a computer is not easy, although humans are able to estimate rotation rates about as well as they can estimate linear velocities (Petersik, 1991) Judgment of rotation speeds is influenced by object size and by number of facets, but there is a component that is independent of these factors (Kaiser, 1990; Kaiser & Calderone, 1991). Neural cells tuned to respond to specific rotation speeds have been identified in the brains of monkeys. These cells respond optimally to movement consistent with rotation, rather than to linear velocities or shear rates (Sakata et al., 1994).

In humans, rotation not only does not interfere with depth perception, but actually facilitates it. Shadows or patterns of dots moving in trajectories consistent with rotary movement are perceived as three-dimensional shapes (Ullman, 1979; Wallach & O'Connell, 1953). It has even been suggested that rotation and stereopsis are processed in the same parts of the brain (Nawrot & Blake, 1989, 1991).

Computational models of structure from motion take advantage of rotation to develop three-dimensional representations of objects. These can work quite well, especially if a rigid structure is assumed at least for consecutive frames (Caudek & Rubin, 2001; Ullman, 1984).

3. Segregating figure from ground

   The task of determining what is part of the object and what is not should be relatively easy for telescope images, where the object is almost always against a uniform black background. That means that any luminosity in the image can be considered to be part of the object. For other types of images, however, this can be one of the most complex problems of computer image processing. Some methods have been successful, such as detecting edge crossings using the second derivative of luminosity, but when edges intersect the problem can get messy (Marr, 1982; Marr & Nishihara, 1978).

4. Changes in image size

   If the distance between the object and the telescope changes, the size of the image can change. As long as figure/ground separation can be determined, it is fairly easy to correct for this problem by scaling the image (Ullman, 1979).

5. Movement across the image

   This problem can again be solved fairly easily by applying a translation to the object, assuming that it can be segregated from the background (Ullman, 1979).

6. Resolution limits

   Satellite distances can range from a few hundred miles in low earth orbit to 23,000 miles in geosynchronous orbit. The Hubble Space Telescope, for example, is 43.5 ft. long × 14 ft. diameter. At an altitude of 353 miles, it subtends a visual angle of 4.8 × 1.5 arc seconds. Even the best telescopes are limited by atmospheric effects and lens diffraction to about 1 arc second, which results in images that are fuzzy at best. These seeing limits mean that few features can be discerned, and even identical orientations will appear different because of atmospheric distortion.

Phase I Technical Objectives

All of the above issues are challenging for computer algorithms, but are generally easy for humans. This suggests that a neural network model might provide an appropriate solution to the problem. By attempting to simulate the parallel communication and learning rules of neurons in the brain, neural network models are able to display more human-like performance than traditional artificial intelligence algorithms. They excel at completing patterns with missing or degraded inputs, and they can discover implicit rules that are difficult to state or incorporate explicitly in computer algorithms. For example, in one successful application a neural network was able to distinguish between rocks and mines in sonar signals, a task that requires considerable skill in humans. It achieved this through exposure to various examples of the signals, with no need to program the model explicitly to search for specific features in the signals (Gorman & Sejnowski, 1988).

For time-dependent problems and those involving sequences of inputs, recurrent neural networks are a good choice. Recurrent neural networks incorporate feedback through arrays of units that are interconnected to all other units, including themselves. The ability of these networks to change activation levels by communication over multiple time steps allows them to respond to sequences or time-dependent inputs, and to vary with time.

A technique that has been successful in a number of motion-detection applications uses genetic algorithms to develop recurrent neural networks optimally suited to the required task. This technique has been applied to a number of tasks involving motion detection, and has been shown to develop abilities to perform complex behaviors with remarkably simple networks. The resulting networks have been able to perform such tasks as predicting the locations of moving objects and intercepting them, intercepting pairs of falling targets requiring divided attention, differentiating between parts of their own bodies and other objects, deciding whether they can fit through an opening, orienting, tracking and avoidance, and shape discrimination (Beer, 1996; Goldenberg, Garcowski, & Beer, 2004; Slocum, Downey, & Beer, 2000). Other abilities that have been demonstrated include finding and following directions to a suitable mate (Werner & Dyer, 1991), generating a sinusoidal output (de Garis, 1992), classifying binary strings or following an optimum path (Spiessens & Torreele, 1992), or guiding robots around rooms without hitting walls or obstacles (Cliff, Husbands, & Harvey, 1992; Floreano & Mondada, 1994).

Two variables can be used to identify matching stereo images in satellite surveillance images. The first is the disparity between two images. Networks might be able to be evolved that can respond maximally to pairs with optimal disparity, picking out the best match from a sequence of images. The second approach would be to evolve an ability to detect rotation. If the rate of rotation can be estimated, then the frame providing the correct match in orientation can be predicted from the sequence. Presenting a few frames would lead to a trajectory toward the correct frame, or a rotation rate that can be used to calculate when a match can be expected to occur.

A third possibility is to evolve a network that can combine disparity and rotation information in a single process, outputting a trajectory leading to the correct frame or responding optimally to the ideal match.

Others have addressed solving the correspondence problem in stereo image processing using genetic algorithms, but the general approach involves using a genetic algorithm to find an optimum solution for a given pair of images. This approach differs from what is proposed here, in which a genetic algorithm is used to evolve a disparity detection or rotation detection neural network that is optimized to detect disparityh with a minimum amount of computation. The difference lies in the approach that the genetic algorithm is applied only once, to arrive at a neural net disparity detector, whereas the typical approach is to use a genetic algorithm for each image.

These solutions are generally concerned with solving the correspondence problem and deriving a 3-D representation based on the input image. In other words, the algorithms must determine which feature of view A matches which feature of view B, and then compute the amount of disparity between these features and use that information to derive a shape representation. In contrast, the goal of th proposed research is simply to estimate the degeree of disparity in two images, and ideally to identify quickly whether the correspondence problem can even be solved for a given pair of images.

The ideal solution differs from a simple disparity detector, in which corresponding points in a stimulus are identified and the degree of disparity for those points is determined. In the present case, the desired solution is a score for an overall disparity level and suitability to form a matching stereo pair.

The plan for evolving appropriate neural networks will be detailed in the next section.

Evolving disparity detectors and rotation detectors is likely to be a more complex problem than the genetic algorithm problems described above. As a result, a large number of units may be required, leading to slow processing in the evolution stage and the possibility that the obtained solution will be computationally slow. That would be a problem for a robotic application or other application requiring real-time responsiveness. In this case, however, there is no need to arrive at a solution quickly, and long computations can be allowed.

Recurrent neural nets have been used previously for solving the correspondence problem. Ananth Raj & Parthasarathy (1995) used a separate layer for each disparity level, with nodes in each layer correspondinig to each pixel. {See the paper for additional historical references—paper in file.] This led to a large network that had to be solved in parts. The network was trained and tested using random-dot gray-scale stereograms.

<u>Research Goals.</u> The goal of the proposed research is to investigate the feasibility of using genetic algorithms to develop recurrent neural networks that can identify matching stereo pairs from a sequence of images, and to determine the level of complexity required and the computational needs to obtain a solution. Specifically, the following questions will be investigated:

1. Can a recurrent neural network evolve in a genetic algorithm in such a way that it can detect level of disparity in a pair of images, and distinguish image pairs with appropriate disparity levels from those with no disparity or an inadequate match of features?

   a. What degree of complexity (i.e., number of neural units) is required for such a network?
   b. Are there benefits to added complexity?
   c. How much computation is required for the evolution of a satisfactory neural network model?
   d. If a satisfactory network is obtained, how much computation is required by the network to obtain an output?
2. Using the same procedure, can rotation-sensitive neural networks be evolved?
   a. Is it better to evolve a single neural network that can output a level signaling the rate of rotation, or to evolve an array of neural networks tuned to respond maximally to specific rates of rotation?
   b. Can the rotation be detected independent of object size and shape?
   c. Can rotation be detected equally well for all axes of rotation?
   d. What is the optimum level of complexity for such a network, and what are the computational requirements?
3. Using the same method, can a combined rotation and disparity detector be evolved that will output a prediction of which frame will hold the best match?
   a. What is the optimum level of complexity, and what are the computational requirements?
   b. Can this type of network provide a more useful or efficient output than the separate rotation or disparity detectors?
4. Do the results of developing these types of networks generalize to new object shapes, distances, sizes, and rotation speeds?

Phase I Work Plan

Experiment 1

Experiment 1 will investigate the first question: whether a genetic algorithm can be used to generate recurrent neural networks that can calculate disparity scores for various images, choosing the pair most suitably matched for stereo vision.

Images

A set of inputs will be created to simulate the types of images that might be obtained through ground-based surveillance telescopes. In order to simulate the limited resolution possible from a ground-based telescope hundreds to thousands of miles from the target, and to keep the order of calculations small, images will be limited to a square 6×6-pixel array. Images will be derived from a set of 3-D models projected onto this array using a standard anti-aliasing routine, which will result in a coarse, blurred image similar to a digital photograph taken through a telescope. All images will be 256-bit gray scale.

Several basic satellite shapes will be used for 3-D models, representing various configurations of antennae, body shapes, and features. Because the ability to detect disparity should be independent of both shape and size, inputs will be created in three image sizes: the large size will occupy most of the pixel array; medium size will occupy approximately ½ of the pixel array, and the small size will occupy approximately ¼ of the pixel array. Each of these will be rotated at rates of 1°, 5°, 10°, 15°, and 20° per frame around each axis (X, Y, and Z relative to the image plane). The viewing angle will change also in a manner consistent with the changing camera angle as the object tumbles across the sky.

In each set, the initial orientation of the object will be used as a reference standard. Each of the other orientations, including the initial orientation, will be input to the network in random order, and the output will be obtained for that orientation.

Model

Genetic algorithms attempt to simulate the evolutionary process, using random strings of numbers that represent features of a real-life problem. A population of these strings is created, and each string is evaluated for its fitness, which is a score indicating how well it solves the problem. New populations are developed for subsequent generations by allowing the strings from the prior generation to mate and reproduce, with any given string's probability of mating determined by its fitness score. The reproduction process involves a crossover of two mating strings, in which the strings are split at random points and combined to form new strings. Over time, populations become composed mostly of strings with high fitness levels and more subtle differences between individual strings. A small rate of mutation, in which a single number is randomly modified, also occurs and allows for the possibility of new variations that could possibly perform better.

When applied to neural networks, the typical approach is to use each number, or feature, in the string to stand for one of the weights between two neural units. Thus, in a network with nine neural units, there will be 81 weights, among which 9 will be weights between units feeding back on themselves. It is common to assume that the weight matrix is symmetrical, resulting in a total of only 45 unique weight values. Other parameters, such as buffer values, can also be represented as features in the strings, and can be varied during evolution as well. The evolution process tends to choose the best values of these features and pass them on to subsequent generations.

A disparity detecting neural network need not be time-dependent, as the network must simply choose the best match to a reference standard. Therefore, the model to be used for Experiment 1 is a feedforward neural network with 72 input units, corresponding to the 36 pixels of the input image and the 36 pixels of the reference standard, a hidden layer with a variable number of units, and one output unit. Activation of the input units will be equal to the pixel luminance value, ranging from 0 for black to 1 for white. Activation of the hidden layer will be

$$b_i = \sum_{j=1}^{N} a_j w_{ji} + \theta_i ,$$

where b_i is the activation of hidden unit i, a_j is the activation of input unit j, and θ_i is a bias term. The hidden units will connect to one output unit, following the same activation rule. The weights and bias terms will be features of a genetic algorithm. An initial population of 100 individual strings will be created with random normal deviates (mean = 0; standard deviation = 1) for the feature values.

Evolution

The most important determinant of fitness is the ability of the network to output a high value when an appropriate level of disparity is present. In addition, the network should output a low value for other, inappropriate matches. Therefore, fitness will be weighted according to the number of possible right and wrong outputs according to the rule

$$F = A_{correct} - \sum_{k} \frac{A_k}{M-1}, \quad k \neq correct ,$$

where $A_{correct}$ is the output unit activation in response to the correctly matched image, A_k is the activation in response to the k^{th} incorrect match, and M is the total number of images to be matched to the standard image. Any negative fitness score will be assigned a value of zero.

Each subsequent generation is produced in a manner in which individuals with higher fitness have a higher probability of reproducing, as follows:

1. Each individual in the population will be assigned a probability of reproducing equal to $F/\Sigma F$, where ΣF is the sum of all fitness scores for the population. Thus, individuals with high fitness scores have a higher probability of reproducing than those with lower fitness scores.
2. Fifty pairs of strings are selected at random, weighted by their probability of reproduction.
3. For each pair of strings, a crossover point is selected at random. The two strings in the pair are split at the crossover point. The portion of the first string after the crossover point is appended to the portion of the second string before the crossover point to make a new string, and vice-versa. Each pair thus produces two new strings.
4. The new generation is evaluated in the same manner as the previous generation, presenting each of the images to each individual and obtaining a fitness score for that individual.
5. A low rate of mutation will also be introduced to the process. Mutation occurs when an individual feature of a string, selected at random, is changed to a random value. A probability of mutation of 0.0001 will be selected as a starting value, but may need to be adjusted in order to obtain satisfactory results.
6. Evolution will continue until the change in maximum fitness value is small for three successive generations. A high mean fitness and small standard deviation will also be required. The values to be used for these criteria will be determined as experience is obtained with the algorithms' performance.

Five network configurations will be tested, with the number of hidden units equal to 1, 2, 4, 8, 16, and 32, respectively. The purpose for doing this is to determine the minimum network size able to solve the problem. If a satisfactory solution is not obtained using this number of hidden units, networks with 64 and 128 hidden units will be evaluated as well.

Results

If this model is successful, it is expected to be a computationally efficient solution to the problem of finding suitable stereo pairs. Unlike other genetic algorithm solutions to the correspondence problem, it would not be necessary to go through multiple generations of evolution in order to arrive at a solution each time. Instead, the result of this experiment would be a single neural network optimized to solve this problem. Only one cycle would be required, and it would only be necessary to compute one sum for each hidden unit and one for the output unit. It is reasonable to believe that a simple neural network could be applied here because it does not actually have to solve the correspondence problem and create a 3-D representation; it only has to discriminate good stereo pairs from bad stereo pairs.

The obtained solution might have theoretical value, shedding light on how disparity can be detected from stereo pairs.

Although the network obtained in Experiment 1 might be sufficient for the desired task, it might be possible to improve its performance by detecting rotation speeds separately and using that information to predict which frame will form the first satisfactory match. Doing this could improve speed by narrowing the number of frames that have to be examined. In addition, false matches in largely symmetrical satellites might be eliminated by judging rotation rates.

Experiment 2

The purpose of Experiments 2 through 4 will be to investigate the second question above: Can a genetic algorithm create rotation-sensitive neural networks? Each experiment will try a different design of network with a different desired result. In Experiment 2, the goal will be to create a network that responds maximally to a specific rate of rotation, namely 5° per frame? It is assumed that if a network can be sensitive to a single rate of rotation, then an array of networks could be created, each one responding maximally to a specific rate of rotation. In Experiment 3, a different approach will be tried, in which networks will be evolved with output proportional to the rate of rotation. Experiment 4 will adapt the more successful approach in an attempt to extend it to more complex rotation around multiple axes.

Images

The images to be used are the same as those in Experiment 1, except that 360 views will be generated for each rotation rate, even though this will result in multiple rotations for most sequences. This is necessary in order to prevent the possibility of using the number of images in the sequence to determine the rate of rotation.

Model

For Experiment 2, rotation will be around only one axis at a time. Neural networks will be evolved for maximum response to a 5° per frame rotation rate for each of the *X*, *Y*, and *Z* axes. The model to be used is a recurrent neural network, of the form

$$\Delta y_i = \frac{1}{\tau_i}\left[-y_i + \sum_{j=1}^{N} \frac{w_{ji}}{1+e^{-g_j\left(y_j+\theta_j\right)}} + I_i \right],$$

where y_i is the activation of unit i, τ_i is the unit's time constant, w_{ji} is the weight between units j and i, g_j is a gain term, θ_j is a bias term, and I_i is the input value, which is zero for all but input units (Beer, 1996; Goldenberg et al., 2004; Slocum et al., 2000). It will have 36 input units, corresponding to the 36 pixels of the input images. These input units will feed to an array of hidden units and one output unit. Each unit will be connected to all the others, including itself, by an array of weights. The weights, time constants, gains, and bias terms will be features of a genetic algorithm. An initial population of 100 individual strings will be created with random normal deviates (mean = 0; standard deviation = 1) for the feature values.

Evolution

Sequences of images will be presented to the input units for each rate of rotation and each shape and size of object. The most important determinant of fitness is the ability of the network to output a high value when an appropriate rotation rate is present. In addition, the network should output a low value for other, inappropriate matches. Therefore, fitness will be weighted according to the number of possible right and wrong outputs according to the rule

$$F = \sum A_{correct} - \sum_k \frac{A_k}{M-n}, \quad k \neq correct\,,$$

where $A_{correct}$ is the output unit activation in response to the correct rate of rotation, A_k is the activation in response to the k^{th} incorrect rotation rate, M is the total number of sequences, and n

is the number of 5°/frame sequences. Note that there will be one correct sequence for each size and shape of satellite. Any negative fitness scores will be assigned a fitness of zero.

Reproduction will proceed by the same method as in Experiment 1, with 100 new strings in each generation. A probability of mutation of 0.0001 will be selected as a starting value, but may need to be adjusted in order to obtain satisfactory results. Evolution will continue until the change in maximum fitness value is small for three successive generations. A high mean fitness and small standard deviation will also be required. The values to be used for these criteria will be determined as experience is obtained with the algorithms' performance.

As in the previous experiment, networks with 1, 2, 4, 8, 16, and 32 hidden units will be evaluated in separate evolutionary trials, with 64 and 128 units if needed to obtain successful results.

Results

It is expected that the most successful application will be for rotation about the Z-axis, as this is fully visible in the image plane, and small-diameter features, if discernable, will rotate in an identical pattern even on larger-diameter objects.

If the network is to provide an accurate prediction of the frame properly matching the orientation of the initial position at a rotation rate of 5° per frame, it must have a discrimination threshold less than 0.7° per frame. This ability should be tested using rotation near the 5° per frame rate, to determine whether it can reliably produce its highest output at the correct rate, regardless of satellite shape and size. In order to be useful for this application, an array of rotation detectors would need to have a resolution on that order—even smaller for smaller rotation rates. This might require a lot of computational power to develop a full-scale application. On the other hand, if the network could provide a close approximation then it might be combined with the disparity detector of Experiment 1 to obtain the closest possible match. This could be especially useful if the satellite is symmetrical, which could cause the disparity detector to come up with a false positive.

Even if the rotation model cannot discriminate at this fine resolution, any ability to recognize rotation could be of theoretical value, because there has been little research on rotation detection and the mechanism for it in humans is not well understood. The ability to detect rotation is likely to have other practical value as well, even if it could not be applied to this problem.

Experiment 3

The goal of Experiment 3 is to test an alternative model for rotation detection. In this model, the network output will be proportional to the rate of rotation, with all rotation rates detected by the same neural network.

Images and Model

The same images, neural network model, and procedures will be used in Experiment 3 as in Experiment 2. The principal difference will be in the fitness measure used. The network needs to output a value proportional to the rate of rotation, so the fitness should be based on the error in rotation rate. Output for each of the sequences will therefore be fit to a least squares regression line, and fitness will be based on the standard error of estimate for the regression, which needs to be minimized. Fitness will be defined as

$$F = 1 - SE,$$

where *SE* is the standard error of estimate. Networks with negative fitness scores will be assigned a fitness value of zero.

Evolution will take place by the same procedure as in Experiments 1 and 2. As in the previous experiments, networks with 1, 2, 4, 8, 16, and 32 hidden units will be evaluated in separate evolutionary trials, with 64 and 128 units evaluated if needed to obtain successful results.

Results

If this model is successful, it will be more computationally efficient than the model of Experiment 2, because it requires only one neural network to estimate all rotation rates, instead of an array of networks tuned to separate rates. As with the other model, it must have a fine rate of discrimination if it is to make an accurate prediction of the frame that is best suited for a stereo pair.

As with the previous model, even if this one cannot provide a precise rotation rate, it might be useful combined with the disparity detector of Experiment 1. Any ability to detect rotation would also have theoretical value and other possible practical applications, even if not suited for this problem.

Experiment 4

Experiments 2 and 3 trained the model for a single axis of rotation. Assuming that one or both of these experiments produces satisfactory results, the next step is to use more complex input values, with rotation about all three axes. The goal of this experiment is to determine whether the networks can correctly identify a rotation rate about a single axis in the presence of rotation about all three axes.

Model

The network model to be used in this experiment is the more successful of the models developed in Experiments 2 and 3. If both experiments produce usable measures of rotation rate, the model from Experiment 3 is expected to be more useful, and will be the first choice for use in this experiment.

Images

Various combinations of rotation rates about all three axes will be generated for this experiment.

Procedure

All sequences showing rotation will be presented to the network. The fittest network will be the one that best detects the correct rate of rotation around a single axis. Separate evolution processes will be run for each of the three axes, in order to obtain networks optimized to detect rotation around each axis. Fitness will be calculated as in Experiment 3 or 2, depending on which model is used.

Results

It is rare that an object would be rotating about only a single axis, so the ability to detect rotation about all three axes, and to identify the rates on each axis, is essential if rotation rate is to be used as a predictor of the correct frame for making a stereo pair.

Experiment 5

As a final experiment, the ability to create a combined rotation and disparity detector will be tested. If such a network could be developed, it could provide the fastest and most accurate prediction of the correct frame for making a stereo pair.

Images and Model

The same set of images will be used as those for Experiment 4. The recurrent network of Experiment 3 will be used. In this case, the fitness will be determined by the ability to select the correct frame for a match, using the fitness definition in Experiment 1.

Work Breakdown Structure

It is anticipated that a large portion of the time required for this project will be dedicated to devising and coding the simulations. Two primary programs will be required: One program for generating the various images to be used in training and testing the simulations, and another program to perform and evaluate the genetic algorithm simulations. A single Windows-based program should be capable of performing all of the simulations.

Progress reports will be submitted every two weeks.

Approximate time frame

1. Weeks 1-5
   - 1.1. Kickoff meeting within 30 days of contract start
   - 1.2. Create images to be used for evolutionary development
     - 1.2.1. Create 3-dimensional mathematical representations for the images
     - 1.2.2. Code program to generate bitmap images of these various objects, based on mathematical description of shape, orientation and position, projection onto image plane, and antialiasing routine for coarse pixel size.
2. Weeks 6-14
   - 2.1. Create program to perform genetic algorithm for evolution of appropriate neural nets.
     - 2.1.1. This will probably be a large chunk of the project.
     - 2.1.2. Requires choosing optimum parameters for model, may require several iterations to obtain well-behaved program.
3. Weeks 15-20
   - 3.1. Execute genetic algorithm to obtain optimum sensors
     - 3.1.1. Expected to require some time to reach solutions.
     - 3.1.2. Several iterations may be necessary in order to obtain satisfactory results
   - 3.2. Analyze and interpret results
     - 3.2.1. Determine whether results are satisfactory
     - 3.2.2. Examine final networks to determine whether it is possible to identify rules describing how they function.
4. Weeks 20-26
   - 4.1. Finish analyzing and interpreting results
   - 4.2. Technical review within 6 months
5. Weeks 27-33
   - 5.1. Prepare final report
   - 5.2. Prepare Phase II proposal
6. Weeks 34-39
   - 6.1. Examine ability of network to generalize to images outside the training set.

6.2. Determine computation requirements for scaling up to larger and more complex images
  6.2.1. Extend network design through modification, if necessary, to apply to these situations.
  6.2.2. Devise training protocol for using actual images or realistic simulations of images
6.3. Investigate possibility of using rotation to distinguish figure from non-uniform backgrounds

Related Work

No prior work has been done with genetic algorithms, with rotation detection or disparity detection. Indirectly related work: Principal investigator developed a neural network model of auditory pitch perception and evaluated its performance in relation to human data for Ph.D. dissertation, completed November, 2003. He also programmed a neural network model to assist his advisor and a fellow graduate student at Kent State University.

Relationship with Future Research or Research and Development

If the project is successful, then it can be applied to the analysis of all space surveillance imagery collected by the Air Force, primarily imagery collected from the sensors at the Maui Space Surveillance System.

The research proposed in Phase I is limited to gray-scale images with uniform black backgrounds. An important objective in Phase II will be to develop the ability to separate figure from non-uniform backgrounds, and to process color images. The ability to detect rotation and disparity might assist in achieving this goal. Most commercial applications would require this ability, as uniform backgrounds are rare in environments other than space exploration. This ability would allow the applications to be extended to satellite surveillance photos, such as data collected from the XSS-10 and XSS-11 missions, space exploration photographs, surveillance videos of public locations, and studies of fluid flow eddy currents.

Commercialization Strategy

The technology to be developed in Phase I could be used to enhance any sequence of images. This may require modification of the algorithm to account for differences in image size and characteristics from the simulated images used in Phase I. It is expected that this procedure will be successful for images with uniform black backgrounds.

Application to other types of images will require the ability to process larger, higher-resolution images and to separate figure from complex non-uniform backgrounds. The ability to process color will also need to be developed.

Rotation and disparity detectors, if they are successful, are likely to have many commercial applications, including measurement and detection of motion.

Key Personnel

Clifford F. Lewis is the sole proprietor, and will serve as the principal investigator. He is employed full-time in this capacity, and will conduct all of the research in Phase I. He holds a Ph.D. in Experimental Psychology from Kent State University and has experience in Visual C++ programming, statistical analysis, and neural network modeling. A resume is attached to the end of this document.

Facilities/Equipment

Two PC-compatible computers with Microsoft Windows operating systems are available for programming, simulation, data analysis, and report writing. One is a 450MHz Intel Pentium II-based computer with 256 MB RAM and 40GB hard drive. The other is a Winbook XL[3] notebook computer with 500 MHz Pentium Pro processor, 64 MB RAM and 11GB hard drive. These computers are expected to be adequate to perform the simulations proposed for Phase I. SPSS statistical software, Microsoft Visual C++, and Microsoft Office are available on these computers.

All work will be performed at the Proprietor's home at 3269 Altamont Ave., Cleveland Hts., OH. It is believed that the property is in compliance with all environmental laws and regulations of federal, State of Ohio, and City of Cleveland Hts. governments. The proposed work will not pose any environmental hazards.

Subcontractors/Consultants

No outside assistance is expected to be needed for Phase I research. Online newsgroups can provide programming and modeling assistance and if needed.

Prior, Current, or Pending Support of Similar Proposals or Awards

No prior, current, or pending support for proposed work.

References

Ananth Raj, D. P., & Parthasarathy, G. (1995). Disparity estimation from a stereo pairs using recurrent neural network, *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century* (Vol. 5, pp. 3896-3901): Dept. of Electr. & Comput. Eng., Osmania Univ., Hyderabad, India.

Beer, R. D. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes & M. J. Mataric & J.-A. Meyer & J. Pollack & S. W. Wilson (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: Bradford Books.

Blake, R., & Wilson, H. R. (1991). Neural models of stereoscopic vision. *Trends in Neurosciences, 14*(10), 445-452.

Caudek, C., & Rubin, N. (2001). Segmentation in structure from motion: modeling and psychophysics. *Vision Research, 41*, 2715-2732.

Cliff, D., Husbands, P., & Harvey, I. (1992). *Evolving visually guided robots* (Cognitive Science Research Paper CSRP 220). Brighton, England: University of Sussex.

Coren, S., Ward, L. M., & Enns, J. T. (2004). *Sensation and Perception* (6th ed.). Hoboken, NJ: John Wiley & Sons.

de Garis, H. (1992). The genetic programing of steerable behaviors in GenNets. In F. J. Varela & P. Bourgine (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: The MIT Press.

Floreano, D., & Mondada, F. (1994). Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In D. Cliff & P. Husbands & J. Meyer & S. Wilson (Eds.), *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: The MIT Press.

Goldenberg, E., Garcowski, J., & Beer, R. D. (2004). May we have your attention: Analysis of a selective attention task. In S. Schaal & A. J. Ijspeert & A. Billard & S. Vijayakumar & J. Hallam & J.-A. Meyer (Eds.), *From Animals to Animats 8: Proceedings of the Eighth*

International Conference on the Simulation of Adaptive Behavior. Cambridge, MA: Bradford Books.

Gorman, R. P., & Sejnowski, T. J. (1988). Learned classification of sonar targets using a massively parallel network. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 36*, 1135-1140.

Hubel, D. H., & Weisel, T. N. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology, London, 160*, 106-154.

Kaiser, M. K. (1990). Angular velocity discrimination. *Perception & Psychophysics, 47*(2), 149-156.

Kaiser, M. K., & Calderone, J. B. (1991). Factors influencing perceived angular velocity. *Perception & Psychophysics, 50*(5), 428-434.

Marr, D. (1982). *Vision*. San Francisco: Freeman.

Marr, D., & Nishihara, H. K. (1978). Visual information processing: Artificial intelligence and the sensorium of sight. *Technology Review, 81*, 2-23.

Nawrot, M., & Blake, R. (1989). Neural integration of information specifying structure from stereopsis and motion. *Science, 244*, 716-718.

Nawrot, M., & Blake, R. (1991). The interplay between stereopsis and structure from motion. *Perception & Psychophysics, 49*(3), 230-244.

Petersik, J. T. (1991). Perception of three-dimensional angular rotation. *Perception & Psychophysics, 50*(5), 465-474.

Sakata, H., Shibutani, H., Ito, Y., Tsurugai, K., Mine, S., & Kusunoki, M. (1994). Functional-properties of rotation-sensitive neurons in the posterior parietal association cortex of the monkey. *Experimental Brain Research, 101*(2), 183-202.

Slocum, A. C., Downey, D. C., & Beer, R. D. (2000). Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In J.-A. Meyer & A. Berthoz & D. Floreano & H. L. Roitblat & S. W. Wilson (Eds.), *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: Bradford Books.

Spiessens, P., & Torreele, J. (1992). Massively parallel evolution of recurrent networks: An approach to temporal processing. In F. J. Varela & P. Bourgine (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: The MIT Press.

Ullman, S. (1979). *The interpretation of visual motion*. Cambridge, MA: M.I.T. Press.

Ullman, S. (1984). Maximizing rigidity: The incremental recovery of 3-D structure from rigid and rubbery motion. *Perception, 45*, 205-217.

Wallach, H., & O'Connell, D. N. (1953). The kinetic depth effect. *Journal of Experimental Psychology, 45*, 205-217.

Werner, G. M., & Dyer, M. G. (1991). Evolution of Communication in Artificial Organisms. In C. G. Langton & C. Taylor & J. D. Farmer & S. Rasmussen (Eds.), *Artificial Life II: Proceedings of the Workshop on Artificial Life Held February, 1990 in Santa Fe, New Mexico*. Redwood City, CA: Addison-Wesley.

Resume: Clifford F. Lewis, Ph.D.

ADDRESS

3269 Altamont Avenue
Cleveland Heights, OH 44118
(216) 371-3981
clewis@kent.edu

EDUCATION

Ph.D., Experimental Psychology, 2003, Kent State University, Kent, OH
M.A., Experimental Psychology, 1998, Kent State University, Kent, OH
Graduate coursework in mathematics, 1980-81, Kent State University, Kent, OH
B.S., Engineering: Fluid & Thermal Sciences, 1977, Case Western Reserve University, Cleveland, OH

PROFESSIONAL AFFILIATIONS

International Neural Network Society
Cognitive Science Society

RESEARCH INTERESTS

Motion perception with applications in virtual reality.
Connectionist and other computational models, especially as applied to perception.
Models of learning.

PROFESSIONAL EXPERIENCE

1995-2003 — *Part-time instructor,* Kent State University, Baldwin-Wallace College, and Cleveland State University. Courses taught: Sensation & Perception, SPSS Lab, Statistical Inference (graduate level), Quantitative Methods in Psychology I & II, Computers in Psychology, General Psychology

1991-1994 — *Free-lance writer*. Specializing in technically oriented marketing literature, instruction manuals, and magazine articles.

1985-1991 — *Editor,* Materials Engineering. Penton Publishing, Cleveland, OH. Wrote more than 60 published feature articles about applications of materials for the development of a variety of engineered products. Edited more than 30 articles submitted for publication. Supervised staff of three editors. Coordinated efforts of art department and production staff with editorial staff to get magazine published on time.

1981-1985 — *Chief Engineer,* Superior Carbon Products, Inc., Cleveland, OH. Promoted to Chief Engineer after six months as assistant to Chief Engineer. Supervised other engineers, draftsmen, maintenance department, and laboratory. Worked with customers to select appropriate carbon brushes and slip ring assemblies for their applications, and to solve problems.

1977-1980 — *Analytical Engineer,* Pratt & Whitney Aircraft, East Hartford, CT. Computer-aided engineering of airfoils for jet engine fans and compressors. Quasi-3D modeling of airflow patterns around airfoils.

Resume: Clifford F. Lewis, Ph.D. **Page 2**

PUBLICATIONS

Lewis, C. F. & McBeath, M. K. (2004). Bias to experience approaching motion in a 3-dimensional virtual environment. *Perception, 33,* 259-276.

Lewis, C. F., McBeath, M. K., & Fountain, S. B. (1999) Lay-language version of *Finding the missing fundamental: A connectionist model of harmonic detection.* Invited paper, Acoustical Society of America World Wide Press Room [On-line], 138th meeting. Available: http://www.acoustics.org/press/138th/lewis.htm

Lewis, C. F. (2003). *A neural network model of virtual pitch.* Doctoral dissertation, Kent State University.

PAPERS PRESENTED

Lewis, C. F. (2003, April). *A neural network model of harmonic detection.* Poster presented at the annual meeting of the Acoustical Society of America, Nashville, TN.

Lewis, C. F. (2002). *A neural network model of virtual pitch.* Paper presented at the Brown Bag Seminars, Dept. of Psychology, Kent State University, Kent, OH.

Lewis, C. F. (2000). *A neural network model of pitch perception.* Paper presented at the Brown Bag Seminars, Dept. of Psychology, Kent State University, Kent, OH.

Lewis, C. F., McBeath, M. K., & Fountain, S. B. (1999, November) *Finding the missing fundamental: A connectionist model of harmonic detection.* Paper presented at the annual meeting of the Acoustical Society of America, Columbus, OH.

REFERENCES

Stephen B. Fountain, Ph.D.	Maria S. Zaragoza, Ph.D.	Michael K. McBeath, Ph.D.
Dept. of Psychology	Dept. of Psychology	Dept. of Psychology
Kent State University	Kent State University	Rm. 301 Technology Center
P.O. Box 5190	P.O. Box 5190	Arizona State University
Kent, OH 44242-0001	Kent, OH 44242-0001	Tempe, AZ. 85287-1104
(330) 672-3826	(330) 672-2018	(480) 965-8930
sfountai@kent.edu	mzaragoz@kent.edu	m.m@asu.edu