Conf-9411144--3

SAND94-2817C

Chemical recognition software*

John S. Wagner, Michael W. Trahan, Willie E. Nelson,
Philip J. Hargis Jr., Gary C. Tisone

Sandia National Laboratories, Albuquerque, NM 87185

ABSTRACT

We have developed a capability to make real time concentration measurements of individual chemicals in a complex mixture using a multispectral laser remote sensing system. Our chemical recognition and analysis software consists of three parts: 1) a rigorous multivariate analysis package for quantitative concentration and uncertainty estimates, 2) a genetic optimizer which customizes and tailors the multivariate algorithm for a particular application, and 3) an intelligent neural net chemical filter which pre-selects from the chemical database to find the appropriate candidate chemicals for quantitative analyses by the multivariate algorithms, as well as providing a quick-look concentration estimate and consistency check. Detailed simulations using both laboratory fluorescence data and computer synthesized spectra indicate that our software can make accurate concentration estimates from complex multicomponent mixtures, even when the mixture is noisy and contaminated with unknowns.

Key Words: chemical recognition, chemometrics, neural nets, genetic algorithms, evolutionary algorithms

1. SCIENTIFIC BACKGROUND AND GOALS

The purpose of chemical recognition software is to estimate the individual chemical concentrations in a multicomponent mixture using a premeasured chemical catalog of calibrated spectral signatures. The software should be able to reject noise and optical contaminants such as scattered light, and be able to reject unknown or uninteresting chemicals. A multispectral laser remote sensing system generates a range-resolved 2-D spectral signature of a remote object or plume. The 2-D spectra can be visualized as a surface: the surface height representing return signal intensity, with the excitation wavelength from the laser source along one axis and emission (or fluorescence) wavelength from the interrogated object along the other axis.Two typical uv spectra taken by a uv fluorometer are shown in Figure1, for acetone and benzene. Note that, at least in

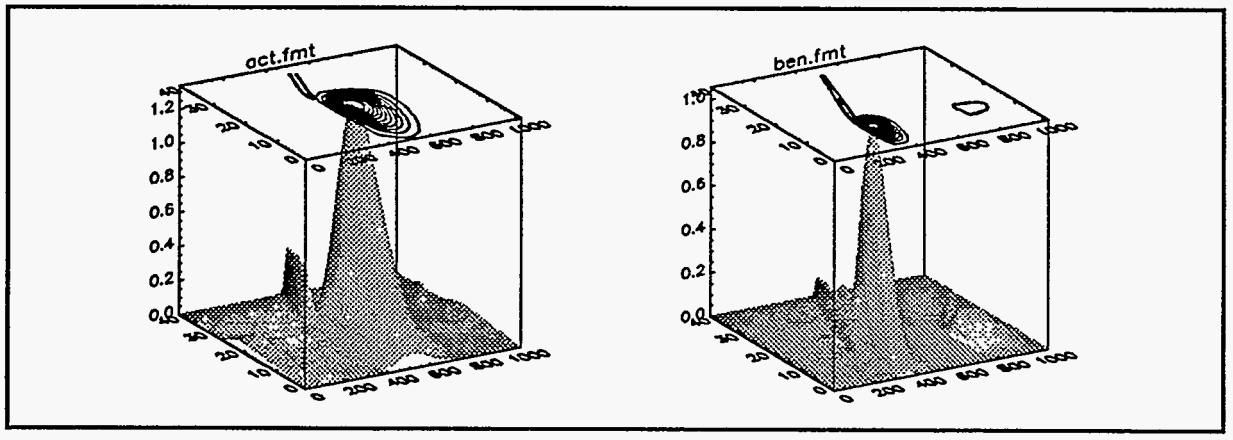

Figure 1. 2-D fluorescence spectra of acetone and benzene. There are 41 excitation (laser) wavelengths and 1024 fluorescence (return) wavelengths

the ultraviolet, spectral features can be broad and featureless: this is not necessarily so in the visible and infrared. Our objective is to develope software that can use both broad and featureless as well as sharply defined spectra.

When the object or plume is made up of many components, the returning 2-D mixture signal is assumed to be a linear superposition of the contributions from each component. The signal contribution from each component depends on its unique spectral shape amplified by its relative concentration.

Historically, algorithms designed to estimate concentrations of components in mixtures used a chemometric approach involving 1-D least-square minimizations[1], and in some cases normal-component analysis[2]. These techniques do tend to work well for many applications. Nonetheless these conventional methods do not utilize all the available information in

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

spectra, such as the correlation of one pixel's intensity with another. For example when using a conventional 1-D method, the pixels of the mixture can be arbitrarily scrambled without changing the resulting concentration estimates, as long as the identical pixels in the database catalog are scrambled in the same way. In 2-D (and in higher dimensions) the spectra can be converted into a 1-D problem by stringing the pixels out, as shown in Figure 2. In higher dimensions, such as in 2-D, the

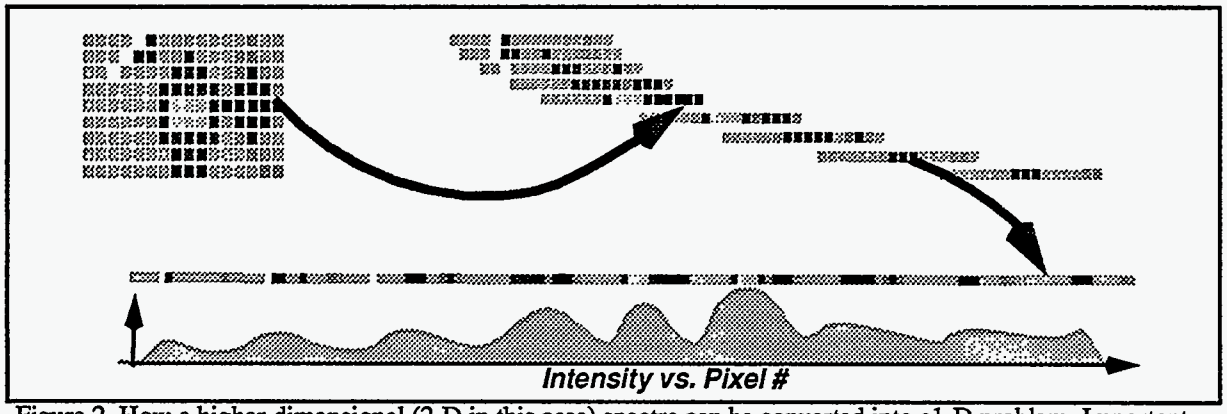

Figure 2. How a higher dimensional (2-D in this case) spectra can be converted into a1-D problem. Important information relating near neighbor pixels is potentially lost.

pixel-to-pixel correlations are even more structured to the human eye, and more complex in the sense that pixels along diagonals, and well-separated pixels can be correlated with each other.

In order to take advantage of the additional information available in spectral signatures we have been developing new techniques of analyzing spectral signatures, including new ways of adaptively extracting minute chemical concentrations in complex multicomponent mixtures. Our chemical recognition and analysis software research effort is concentrating on multivariate methods, neural nets and genetic optimization. We are addressing future concerns as well, for example massively parallel implementations, accurate uncertainty estimates on concentrations, and intelligent techniques to search for chemicals when the database catalog becomes very large (~hundreds). We expect these techniques will have important uses that go well beyond laser remote sensing.

2. THE MULTIVARIATE APPROACH

The core algorithm in our chemical recognition software package is the multivariate 'patch' algorithm. This algorithm computes the chemical concentrations and estimates the uncertainty of those concentrations. Conceptually the multivariate method is like a least squares fit, where the fitted parameters (the concentrations) are chosen to minimize the residuals between the measured unknown mixture and the fitted solution. The 'patch' algorithm extends this approach in the sense that it minimizes the residuals on pixel sets which collectively contain the most important features in a particular chemical spectrum. Since neighboring pixels often contain valuable correlated information, this approach enables better concentration estimation and better noise rejection. The algorithm incorporates conventional 1-D chemometrics as a limiting case. Visually these correlated sets of pixels appear as patches when overlaid onto a 2-D spectrum, hence the name. This is

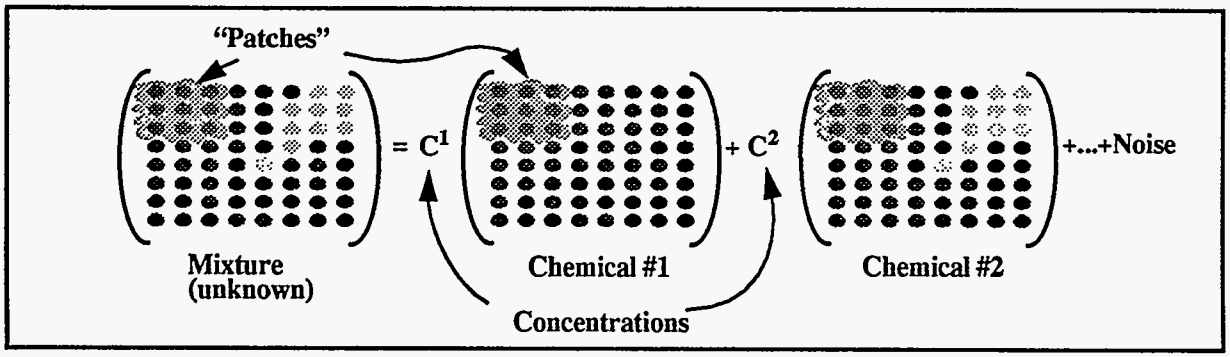

Figure 3. A schematic diagram of a 2-D problem. The mixture is a linear superposition of components (individual chemicals).

illustrated in Figure 3, where the mixture is on the left, and is equal to a sum of chemical samples multiplied by their respective concentrations, and may be contaminated with noise. One patch occupies the same pixels in the mixture and in the chemical components.

Mathematically, the 'patch' algorithm computes the residuals between the mixture and a hypothetical solution, for all pixels in each patch, then minimizes this residual, or a function of this residual, for every patch independently. The user can chose to minimize the squared sum, the sum of the squared (as in least squares), or the summed absolute value (as in robust estimation) of the residuals in every patch. We are currently assessing trade-offs associated with these choices. The system of equations is inverted by computing the pseudo-inverse of the patch-sum matrix using singular value decomposition. The concentrations are then computed when the pseudo inverse is multiplied by the mixture and component-sum matrix. The mathematical details are unfortunately too complicated to derive in this short paper.

After the concentrations are estimated the algorithm then computes an uncertainty estimate. This uncertainty estimate is derived from the first three terms of the Taylor series expansion of the rate of change of concentration from both mixture, chemical database and mixture-chemical (cross product) uncertainty. This is shown in Figure 4. Conventional algorithms

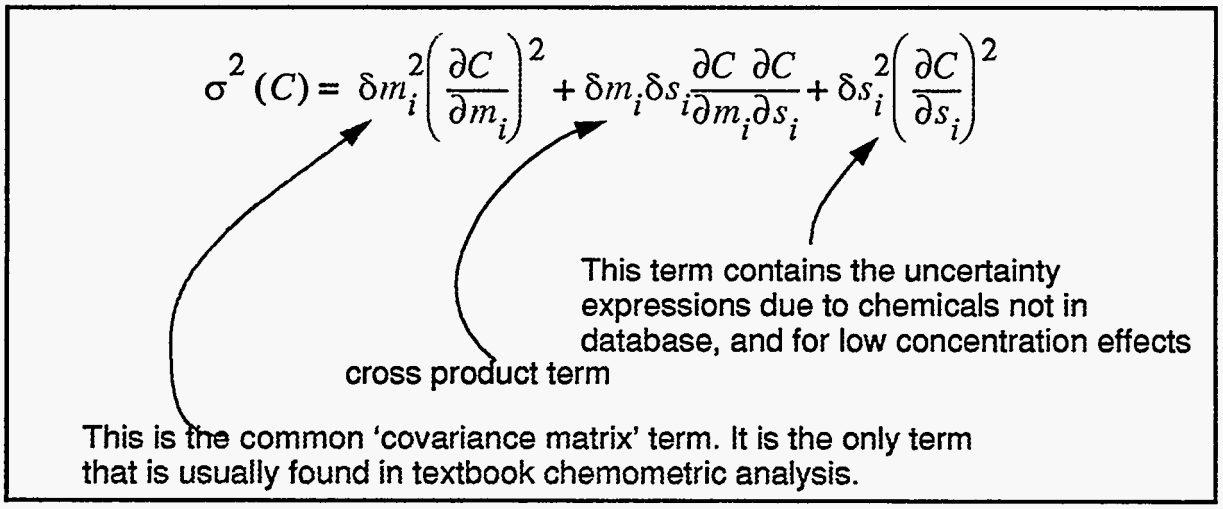

Figure 4. The terms of the uncertainty of a concentration estimate.'C' is the concentration, 'm' is a mixture measurement, and 's' is a chemical component sample measurement

usually assume only the mixture has measurement uncertainty, with the curious result that the uncertainty in the concentrations is computed from a covariance matrix that is independent of the mixture itself. In other words all mixtures would have the same uncertainty in the concentrations. The covariance matrix is the first term in our Taylor expansion, with the other new terms exhibiting the expected loss of confidence as the concentrations become relatively small or when there is a chemical missing from the database. We have also implemented a Monte Carlo uncertainty estimate, so that if the mixture measurements do not obey a normal statistical distribution we can still estimate concentration errors with a real system response.

Patches are lists of correlated pixels, but how are these lists chosen? This question of patching strategy is a complicated one. Patches can overlap, be sparse, or widely separated. In fact a patch does not have to made of contiguous pixels at all. Although any reasonable patching strategy will yield very good results, for many problems the optimization of the patch algorithm is too labor intensive and too subjective. Some of the possibilities are shown in Figure 5. The optimization of the patches will be discussed in the next section on genetic optimizations.

As an example of the benefits of the flexibility of the patch algorithm, consider the following problem. A set of three gaussian-shaped 'computer synthesized' chemicals are created with their peaks on a diagonal, and the spectra of these three chemicals are stored in the multivariate algorithm's chemical database. The image is made up of 40x40 pixels. Mixtures are computer created by combining these three chemicals with concentrations 0.5, 1.0, and 1.5 respectively. Then a fourth chemical is added to the mixtures, off diagonal, with a very large concentration of 100.0. Both the conventional 1-D algorithm and the 'patch' algorithm are given this problem, with increasing (0% to 100%) random noise added. The results

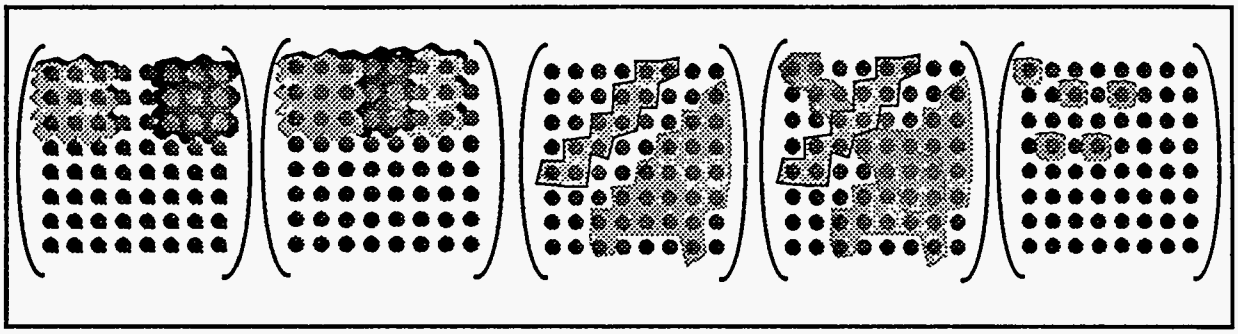

Figure 5. An illustration of different patch strategies. A patch is a list of related pixels. Patches can be of any shape. Patches can contain contiguous pixels but don't have to, and pixels can be in more than one patch. Pixels not in any patch are ignored.

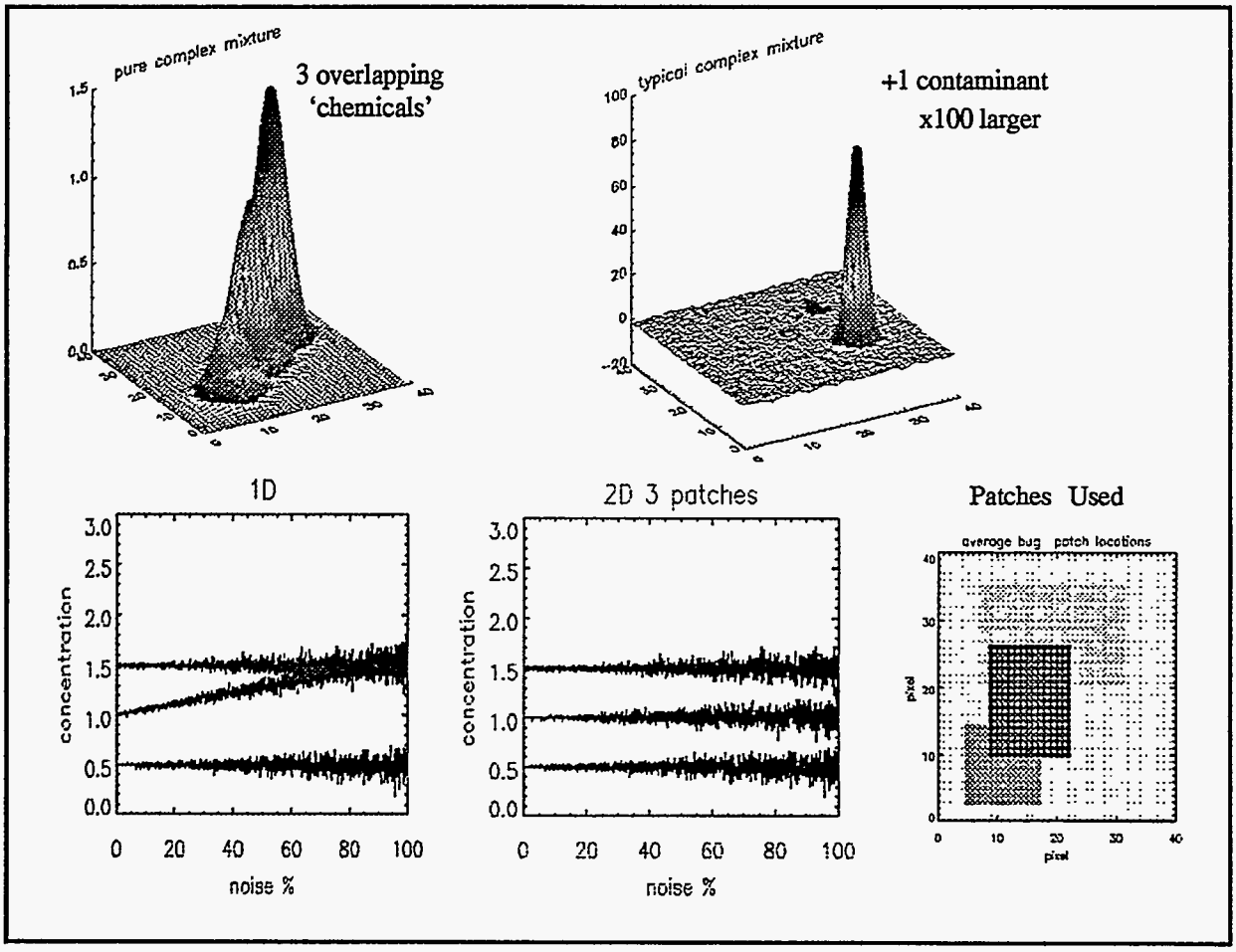

Figure 6. A comparison of 1-D and 2-D multivariate algorithms. The patches used in the 2-D solution are shown in the inset.

are shown in Figure 6. The 1-D algorithm is unable to reject the contaminate as noise is increased.

To demonstrate the algorithm on 'real' data, Figure 7 shows the results of the 'patch' algorithm identifying methanol, xylene, and toluene with relative concentrations of 0.5, 1.0, and 1.5 respectively, as up to 100% noise is added. The

spectrum for each chemical was taken with a uv fluorometer, and the mixtures were generated by the computer by adding the spectra according to the concentrations, and then adding additional random noise. The random noise was uniformly distributed from -1.0 to 1.0 multiplied by the percentage. The size of the images was 41 x 1024 pixels. No attempt was made to remove the scattered light from the raw data. The code achieves good concentration estimates (accurate to a few percent) even at 100% noise.

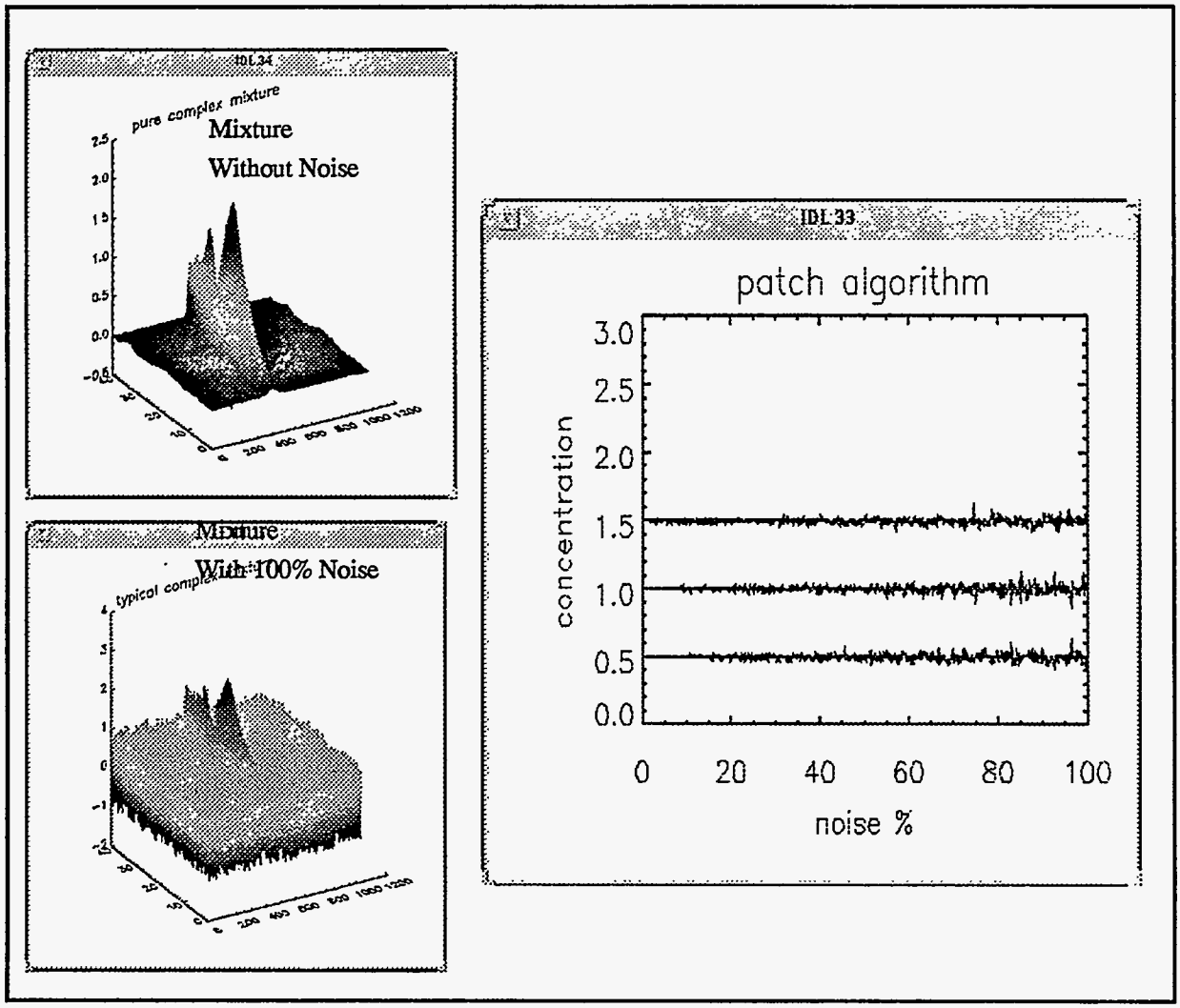

Figure 7. Patch Algorithm performance: Methanol, Xylene, and Toluene as noise increases to 100%

3. GENETIC OPTIMIZATION OF THE PATCH ALGORITHM

To overcome the need for trial and error optimization of the 'patch' algorithm we have developed a novel Genetic Algorithm (GA) to optimize the lists of correlated pixels for any set of candidate chemicals. In a genetic algorithm a set of genetic-like sequences are created in which each sequence can completely describe a possible solution to a problem. Every genetic sequence, or 'bug', must compete with its peers on the problem, and the most fit are allowed to generate offspring for the next generation, with occasional mutation to introduce new genetic material into the population. Generation after generation the population relentlessly improves its fitness. Genetic algorithms are exceptionally good at search and optimization when applied to problems with very large multidimensional solution spaces. GA's, used carefully, are not easily trapped in local minima or maxima, a problem that can plague hill climbing methods or variations on Newton's

method. A schematic diagram of a genetic algorithm is shown in Figure 8.

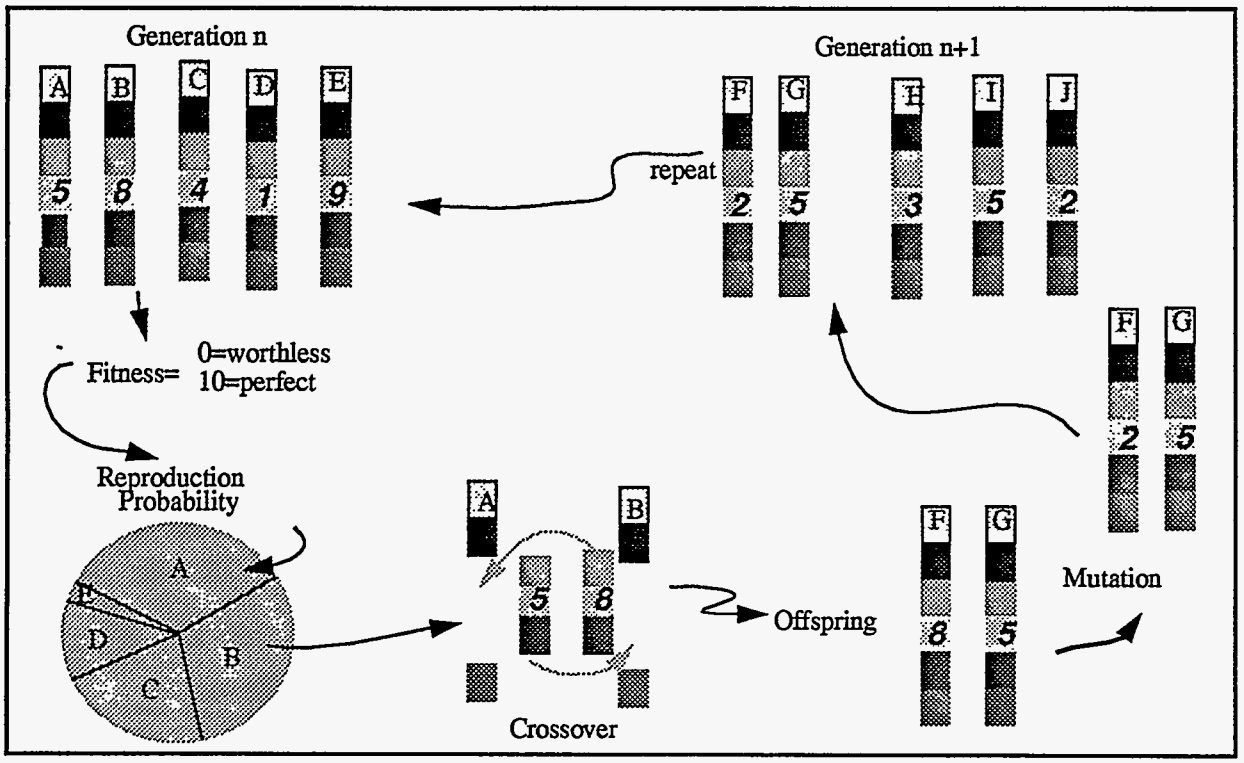

Figure 8. A schematic diagram of a genetic algorithm is shown

Applied to the patch algorithm problem, the GA's genetic sequences are the patch-lists, which describe which pixels are in a given patch. The GA ferrets out which pixels contribute to the solution and organizes them optimally. Figure 8 shows a

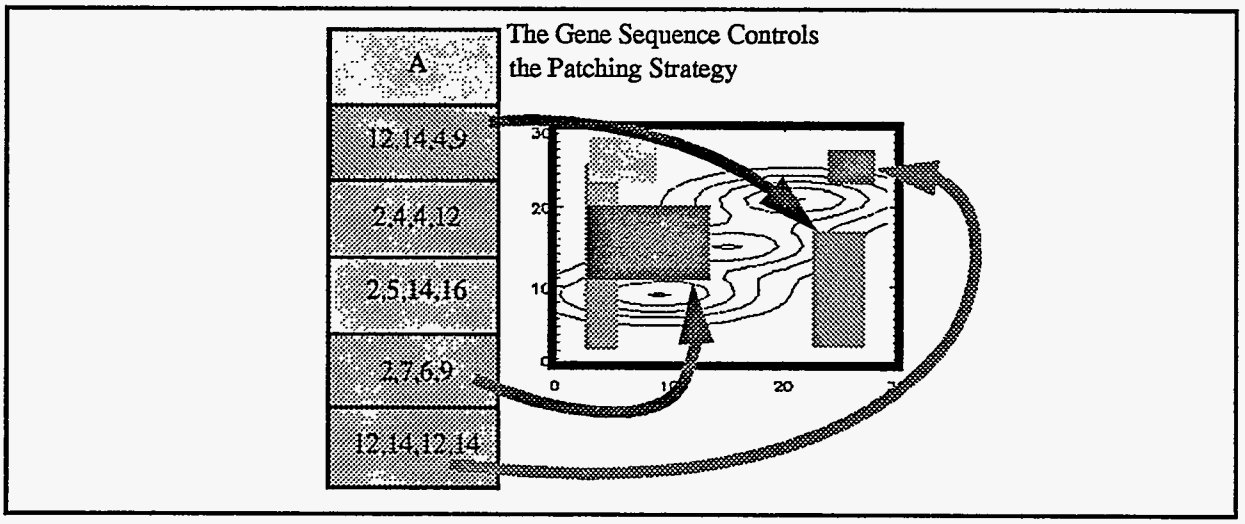

Figure 9. How a genetic sequence can control a patched multivariate algorithm

GA controlling rectangular patches. We have also written GA's not restricted to rectangular patches, and are currently evaluating the trade-offs.

In the training sessions we have conducted so far, the genetically optimized code can frequently achieve an order of

magnitude improvement in speed and accuracy, and frequently comes up with novel and better solutions to complex problems than those anticipated by the authors. The speed increase comes from the fact that the genetically optimized 'patch' algorithm typically uses only a small fraction of the total pixels in the solution, greatly reducing the number of operations required.

Figure 9 illustrates a GA optimizing 24 patches on toluene, xylene, and methanol. The upper left panel is a historical plot

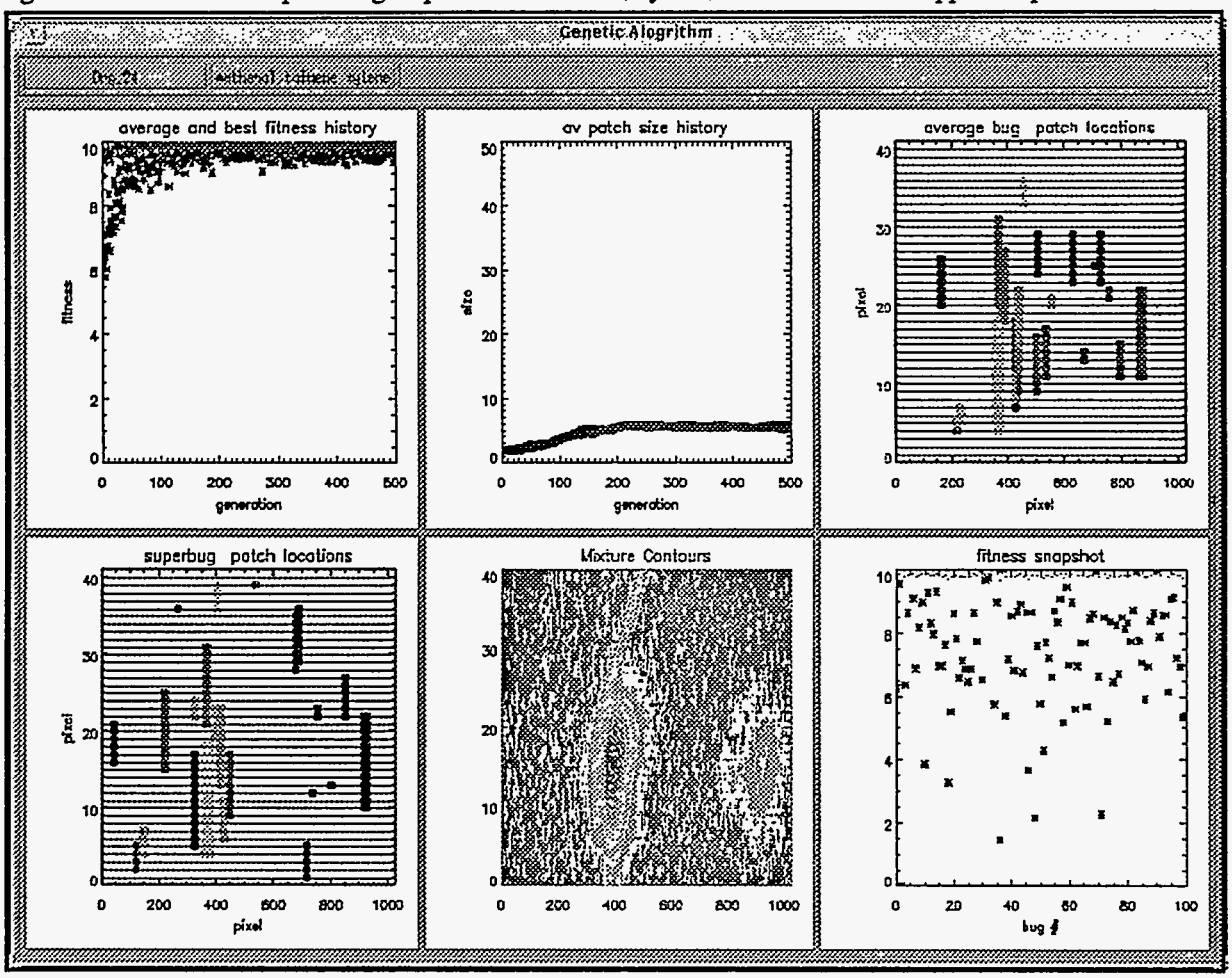

Figure 10. A GA optimizing 24 patches on toluene, xylene, and methanol

over 500 generations of the average fitness (lower curve) and superbug fitness (upper curve). A superbug is the most fit bug in a generation. The fitness is defined so that a perfect bug gets a 10: $fit(bug) = \frac{1.}{\left(\sum_{chemicals} |C_{trainingset} - C_{bug}| + 0.1\right)}$. The middle upper panel is a history of the patch size, the fact that the curve flattens out is a good indication the GA had found the optimal solution after about 200 generations. The right upper panel shows the location of the 24 patches corresponding to an 'average' bug. The patches are not really as distorted as the figure indicates. There is a severe aspect ratio distortion since there are 41 rows of pixels and 1024 columns. There are 100 bugs competing. The lower left panel shows the superbug patches after 500 generations. Experience has taught us that it is usually better to use the average solution rather than the super solution since the average is adapted to a wide range of conditions whereas the super solution is adept only on a special case. The lower middle panel shows a snapshot of the training set: a linear combination of toluene, xylene, and methanol added together using random concentrations and 10% noise added as well. The right bottom panel is a snapshot of the fitness of all 100 bugs taken at the end of the run. The actual raw fitness is indicated by the dots at the top of the panel. The asterisks are the weighted fitnesses. Weighted fitnesses are a tool used in GA's to force competition very early and very late in the run when all bugs are almost equally fit.

It is important to note that the GA is *run only once* for a given application. Once the patches are found that optimally solve for a given set of chemicals, that patch set can be used over and over. Also note that a variety of strategies can be used to train the GA. For example patches can be optimized to recognize certain chemicals while at the same ignoring others and so on. As system modeling improves, and as we gain experience with a real remote sensing system, it will be possible to train the GA using the real system response to optimize concentration estimates while rejecting systemic distortion, noise and contaminates.

4. CHEMICAL RECOGNITION USING NEURAL NETS

As our chemical database grows we will need to intelligently choose which chemicals in the database will be actually used in the quantitative analysis. Some chemicals of interest are mission determined, but an intelligent neural net scanning the input could automate this selection, note unusual occurrences, and provide a useful double check on concentration estimates. Neural nets are attractive in the sense that they can be very fast to evaluate. A feed forward neural net has an easy to follow structure, has favorable scaling with number of chemicals (linear) and can be evaluated in a straightforward series of multiplies and adds. Neural nets do not require an iterative solve or a matrix inversion as do multivariate methods. The disadvantage of a neural net is that it is difficult to construct (train), and does not provide a rigorous uncertainty estimate. These deficiencies may someday be relaxed as research in neural nets continues at a rapid pace. A neural net designed for chemical recognition is potentially quite different from neural nets used in pattern recognition, which are basically classifiers. In the chemical recognition problem similar spectra are superimposed, so instead of asking 'what letter is this?', we are asking 'how much of every letter?', is superimposed on top of every other in the image.

We are currently developing and prototyping a new genetically trained neural net for chemical recognition, as shown in Figures 12 and 13. The most common training technique of neural nets is back propagation. Our novel use of a genetic algorithm to train the net may have advantages in that a GA can also design the net. Whether or not there are any speed or accuracy advantages compared to the conventional training method is under investigation.

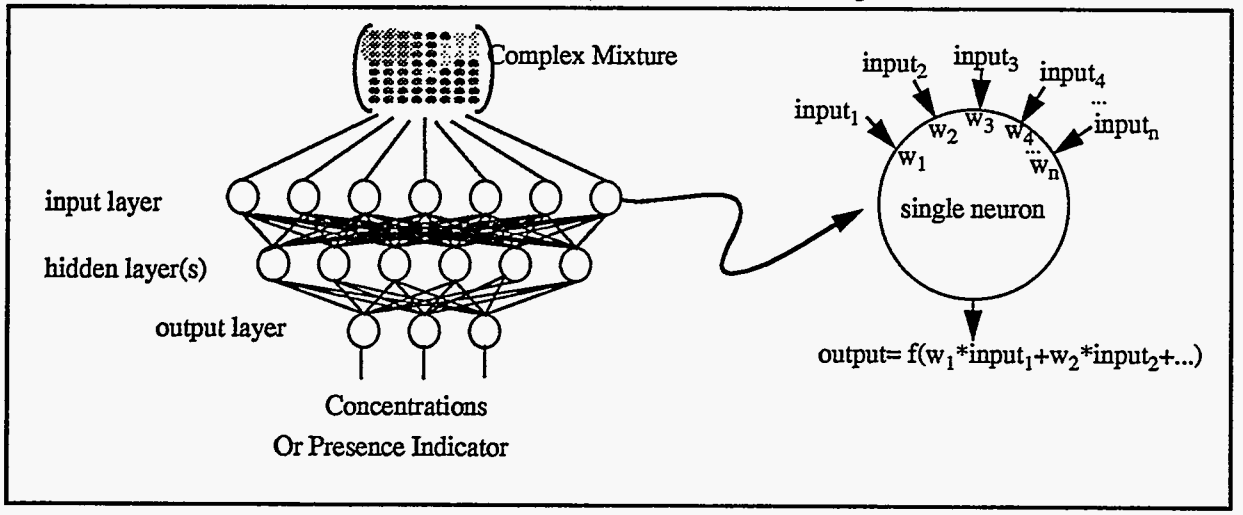

Figure 11. A feed-forward multilayered neural net for chemical recognition

In Figure 12, a 4 layer neural net is training on three gaussian-shaped chemicals along the diagonal, similar to the test case displayed in the earlier Figure 6. The upper left panel is a snapshot of the training set used during generation 50. The training set is recomputed every generation, using a different set of concentrations on each chemical, chosen randomly between 0.0, and 1.0 The second panel on the top row is a history of the raw fitness throughout the first fifty generations. The lower curve is the average fitness of all the 100 nets that are competing, while the top points are the fitness of the superbug at each generation. The gap between the superbugs and the average population is an indication that the general population has a lot of learning to do. The good performance of a superbug compared to the average on these early training sets indicates that it is too overspecialized for a particular special case. The next panel is a snapshot of the raw and scaled fitness at generation 50. The right most panel is a plot of the weights on all 400 neurons in the topmost layer. The weights are initially random, and are not showing much structure this early in the training session. The middle panel plots the 400 weights of each of the 12 neurons in the second layer. The bottom panel plots the 12 weights of each of the 12 neurons in the

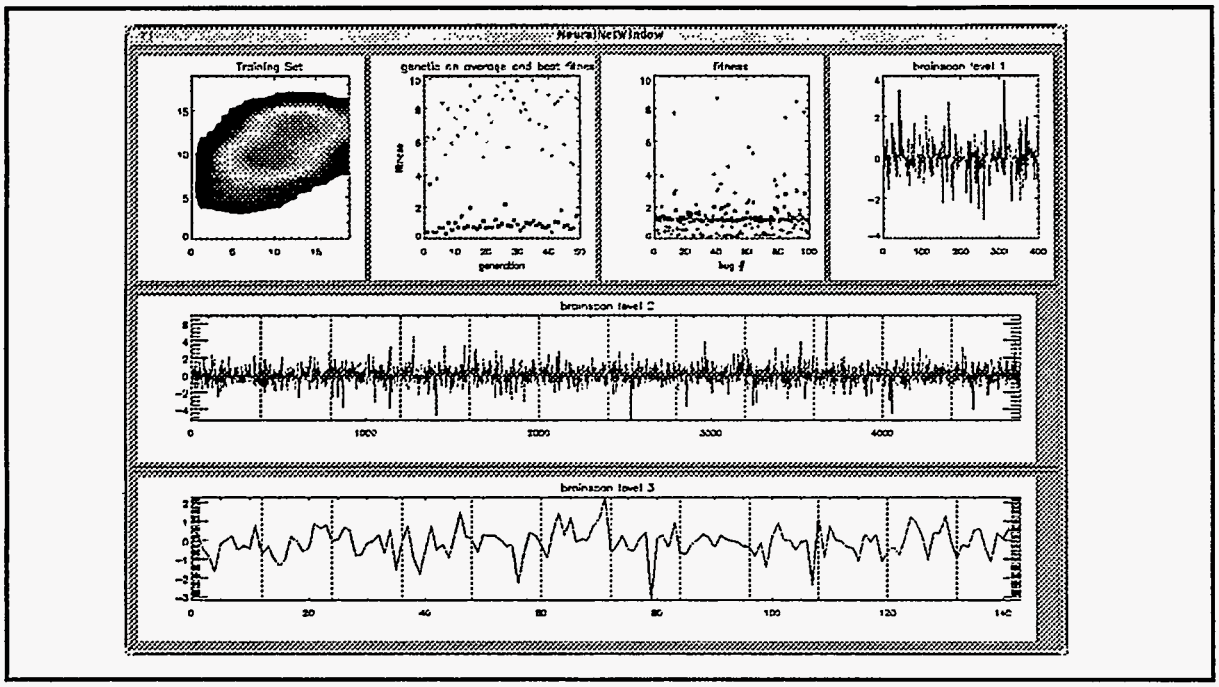

Figure 12. A 4 layer neural net genetically training on 3 gaussian-shaped chemicals after 50 generations. The weights are not yet showing visible structure.

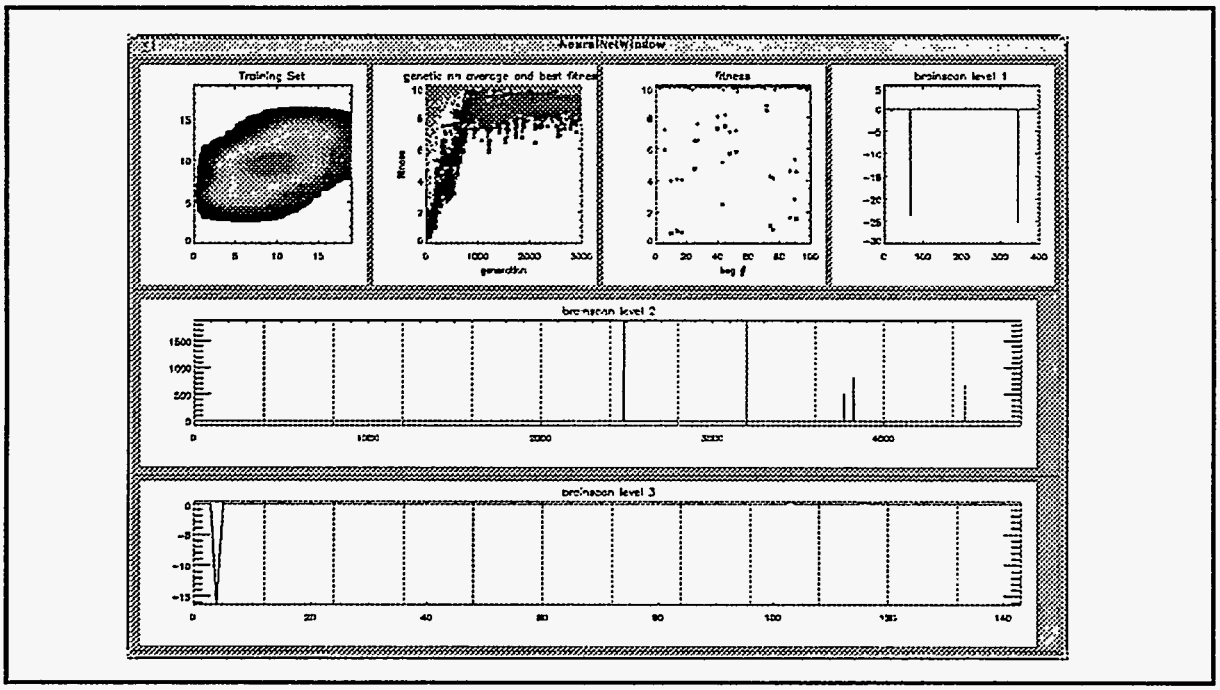

Figure 13. The same 4 layer neural net after 3000 generations. The weights are showing considerable structure.

bottom layer. The bottom 4th layer is not plotted, its output is the net's concentration estimates and are used to compute the fitness.

Figure 13 shows the same net after 3000 generations. The fitness of both the superbugs and the overall population are

achieving fitnesses over 8, which is very good. The population is still learning, and we are not sure of its ultimate capability. Notice the plots of the weights now exhibit considerable structure; the information in certain pixels being amplified and differentiated by the net, while other pixels are multiplied by zero and are effectively discarded.

5. SUMMARY AND FUTURE ACTIVITIES

We have completed the major routines for a chemical recognition software package. The main components of the package are a multivariate 'patch' algorithm, a genetic optimizer for the multivariate routine, and a genetically engineered neural net.

We envision combining the routines in the near future into an expert system, as shown in Figure 14. An expert system as

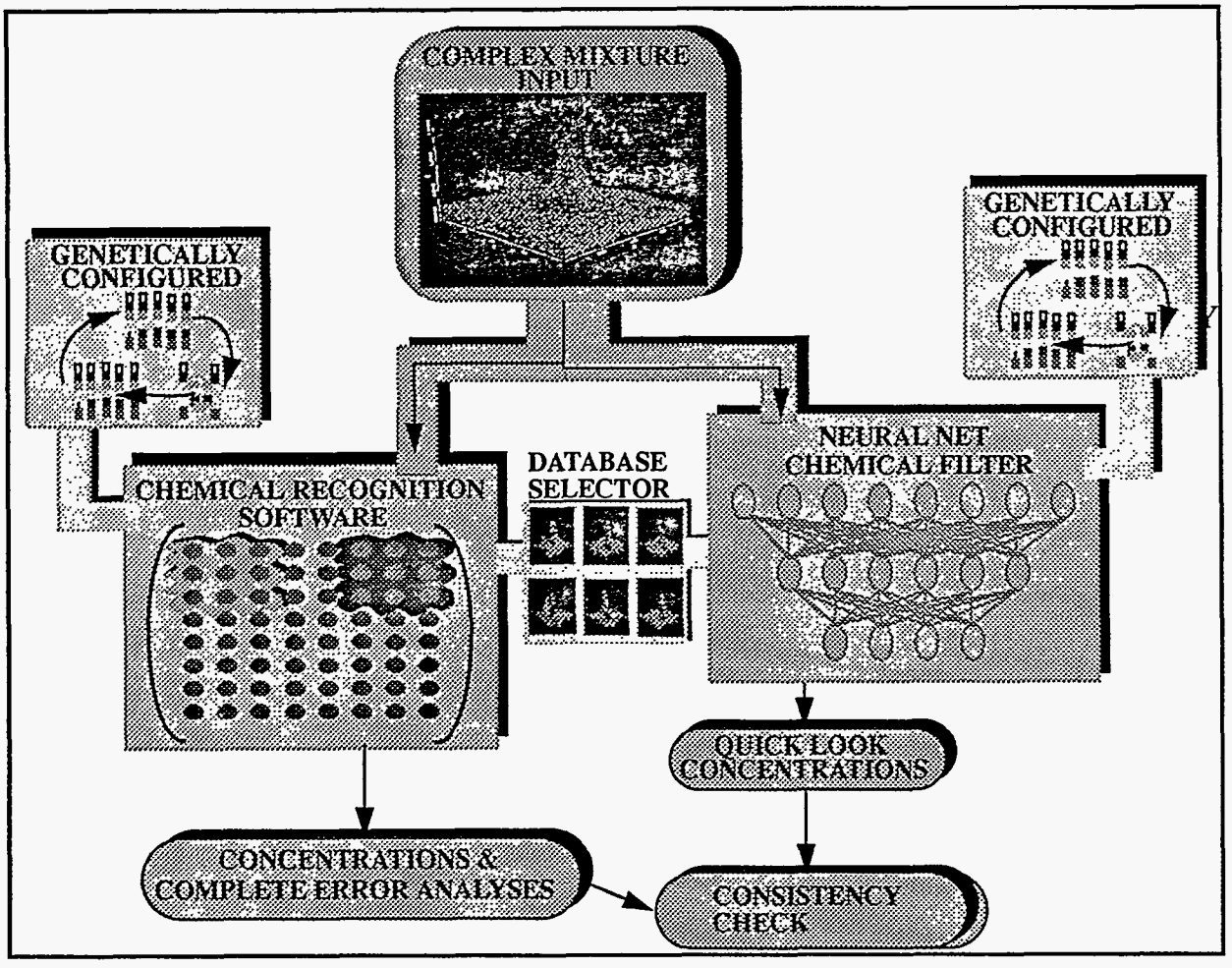

Figure 14. The configuration of a future expert system using multivariate algorithms, genetic optimization, and Intelligent neural nets

would feed the remote sensing signal first to the neural net, which would quickly identify the most important chemicals in the signal and estimate their concentrations. The neural net would hand off the candidate list to the multivariate algorithm, which then uses this list plus any user-specified chemicals in it's analysis and uncertainty estimates. The neural net keeps the work load on the multivariate algorithm to a reasonable level. Good agreement between the neural net and the chemical concentration estimates from the multivariate algorithm will provide a compelling consistency check.

In the future we will be using and testing our software in a multispectral uv fluorescence system. We will continue to research newer more advanced recognition techniques such as constrained multivariate, spectral, and maximum entropy methods and will investigate the properties and performance of new types of neural nets such as holographic, fuzzy and

internally structured Kohonen nets.

6. REFERENCES

1. W. H. Press et. al., Numerical Recipes: The Art of Scientific Computing, page 509, Cambridge University Press, 1986.
2. E.R. Malinowsky and D.G. Howery, Factor Analysis in Chemistry, pg141, Krieger Publishing Co., 1989.

*This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER