AURA Universal Pattern Matching Technology for Multiple Data Types

White Paper

Cybulas approach to integrated pattern matching using the AURA technology is described, highlighting how AURA may be applied to many pattern matching tasks. The use of multiple data types is highlighted incorporating data fusion.

Cybula Ltd.

Aug 2004

Cybula Ltd., IT Centre,
York Science Park
Innovation Way, Heslington
York, YO10 5DG. UK
Phone 01904 567686
Fax 01904 567685
enquiries@cybula.com

1 Introduction.

This paper presents an over view of the use of Cybulas patented AURA technology in a wide variety of pattern matching tasks, highlighting the universal nature of the technology. It describes how, by allowing this, the process permits simple fusion of information resulting from pattern matching on different data sources. Typical applications include sensor management systems, biometric systems, machine monitoring etc. This is built upon AURAs intrinsic strengths, including high performance and scalable operation. Many of the systems described in this paper are detailed in other Cybula documents.

2 Overview.

AURA is centred around the use of a simple, but powerful, pattern recognition engine based on a Correlation Matrix Memory (CMM). The CMM is a well known type of pattern matching method. Since 1986, Prof. Austin has developed CMM based technologies for many applications. In 1999 it was clear that CMM based methods, encapsulated in the AURA technology had great potential for many companies. Cybula was set up to take this universal pattern matching technology to market. The AURA technology is based in software or optional hardware which can be used on a very wide variety of pattern matching problems requiring speed, and scaleability to large data. Since 1986, the AURA methods have constantly been shown to perform as well as other competitive techniques in the areas where it has been applied. Over 150 papers and reports have been published to demonstrate this, and are available from the University of York web site where the technology is still under constant development, expanding its uses to many other problems (www.cs.york.ac.uk/arch/neural).

The now ubiquitous neural network methods such as Kohonen Networks, Radial Basis Function networks and Kohnen networks all allow users develop good pattern matching systems for small problems, where they excel. However, when the problems grow to large datasets, and where very high performance is needed, they become limited. The AURA technology has its origins in neural networks but draws upon pattern recognition methods and parallel processing for its fundamental operation. The well known k-Nearest Neighbour methods (k-NN) is a relatively good pattern matching method that has been constantly shown to operate well on many problems, however, it suffers from slow operation on large data problems. By using a CMM and other advanced methods the

AURA technology allows k-NN based pattern recognition methods to scale to large problems and operate quickly. When combined with other statistical and pattern recognition techniques AURA becomes widely applicable and more robust than the standard k-NN methods.

AURA exists as a C++ software library containing the core pattern match engine and then optional pre-processors and back check functions that are designed to deal with the different data types (discussed below). In addition, dedicated PC and workstation hardware is available that can accelerate AURA applications. AURA is available on a wide variety of platforms and can be ported to any platform that supports the C++ language.

3 AURA applicability

AURA can be applied to almost any data type. Currently, the technology has the following application components:

Signal Data (time varying data)
Text strings (strings of symbolic data)
Document sets
Form Data
Graphs (applicable to images and multidimensional data)

Each of these applications can be illustrated through a number of demonstrators built by the company. These include a Trademark database system, chemical database and face recognition (Graph Matcher), Financial time series prediction, engine health monitoring (Signal data), Address database (text strings) and document retrieval systems (document sets). Each of these applications have allowed the technology to be developed and evaluated and deployed.

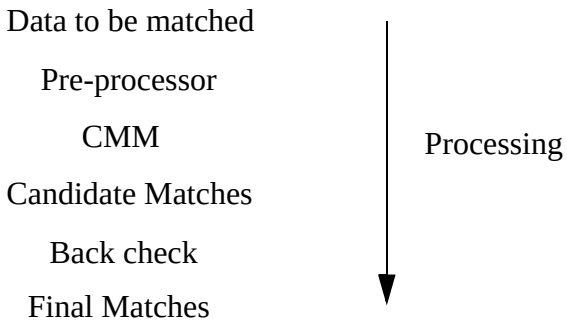
To achieve this wide applicability the technology has a number of core elements, that are combined with a number of adapters to make it work with these different data types. The following describes how the core engine operates to achieve scalable pattern matching, we then describe each data adapter, following this the speed and memory efficiency are described followed by the implementation of AURA.

4 The Core AURA engine.

The core of AURA is a storage and retrieval engine based on a Correlation Matrix Memory. This system allows large amounts of data to be saved and retrieved quickly and efficently. Unlike a database, AURA is designed first and foremost to deal with large incomplete data. That is data with items missing, added or changed. Databases have been developed specifically for clean data. Thus AURA is capable of searching for data held in its storage engine (the CMM) very quickly. The AURA methods have been designed specifically to get the most out of modern computers by using low level operations in the computer and managing data and compute resources very efficiently. If that was it, many others would have developed CMMs. The power of Cybulas approach is to combine the CMM with methods that prepare the data correctly to get the best out of the network and to use the CMM as a part of a more sophisticated data access system.

The AURA technology solves the pattern matching problem on large datasets by not trying to do every thing at once, it uses a two stage method instead. The first stage finds the items that are similar to the input, but does not worry too much about the exact similarity. By doing this rough search a very fast approach can be used. The data items that are candidate matches are then passed to a second stage for detailed matching, again using a number of different carefully developed methods.

The stages that are used are summarised below:



The system takes the target data set and stores this in the CMM. When a data item to be matched is presented to AURA, a pre-processor is used that is suitable for that data. The

pre-processed data is fed to the CMM, results are collected and then refined by the 'back-check' operation. This final stage returns the exact items that match the input. The user can control the level of similarity and the number of items returned as required. The core CMM combined with the pre-processing stages allows both universal and fast pattern matching capabilities.

AURA contains a number of pre-processors broadly grouped in to string and document matching, graph matching and signal matching. The first of these applies to data that is made up of a string of symbols for example text and documents (1D sets of symbols, either words or letters). The second is data that can be represented as an attributed graph. The nodes and the arcs of a graph represent data and relationships between data respecively. This can be applied to images, where the nodes are parts of the image and the relations are the relative positions. Alternatively it could represent people as nodes and how they are related by the relations in the graph. The system allows you to search for matches between graphs that are incomplete. Finally, signal matching is where you have a time related signal that must be searched for, examples are financial time series, sounds and complex signal data. In this case the data is made up of 1D strings of real values.

The diagram below shows the various components making up the AURA technology that are described in the following sections.

<table>
<tr><td>Pre-processors</td><td>Text</td><td>Document</td><td>Graph</td><td>Signal</td></tr>
<tr><td>Storage engine</td><td colspan="4">CMM</td></tr>
<tr><td>Back check</td><td colspan="2">Data retrieval</td><td colspan="2">Classification</td></tr>
</table>

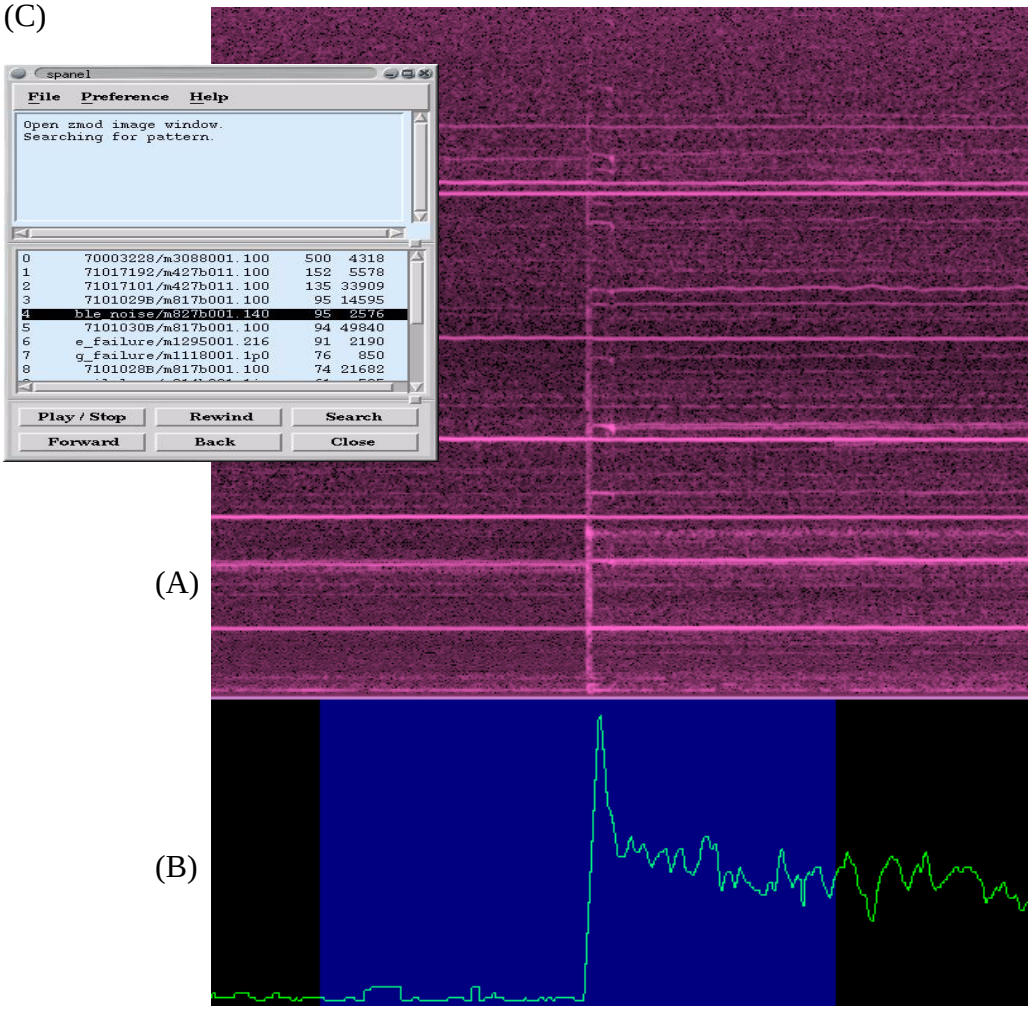
AURA components.

5 The AURA application components.

The following describes the components that can be used with the core AURA pattern match engine to allow it to be applied to the different data types.

Signal Data.

Signal data can be stored and searched using the AURA technology by using the signal pre-processor and back check components. Signal data is typically represented as a string of real valued items, usually indexed over time. The image below shows a system that has been developed for searching for vibration data from Rolls Royce engines. The engines can produce up to 1Gb of vibration data per engine, per flight. AURA allows a user search using samples of vibration data.

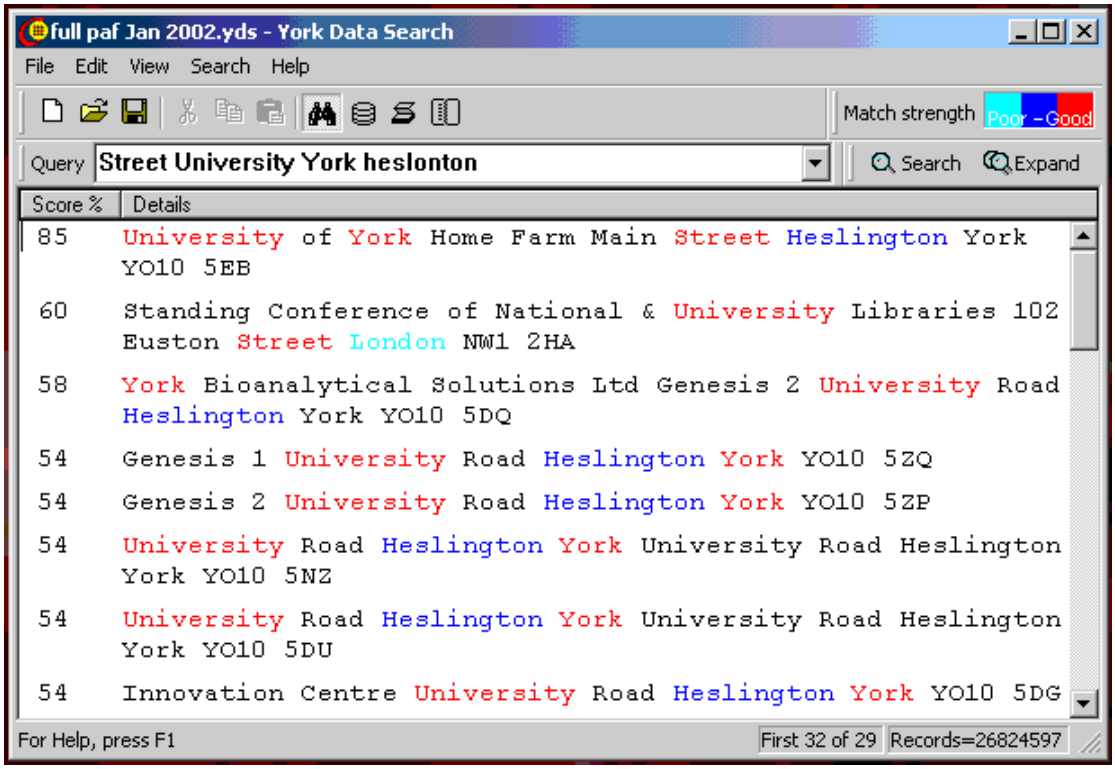


This figure shows how the signal components are used to search vibration data from engines. Section (A) shows a power frequency plot of the data, section (B) shows a signal searched for in this data. (C) shows the search dialogue with a list of matching data segments.

This technology can be used for finding any string of variables. The methods have been designed specifically to use memory very efficiently.

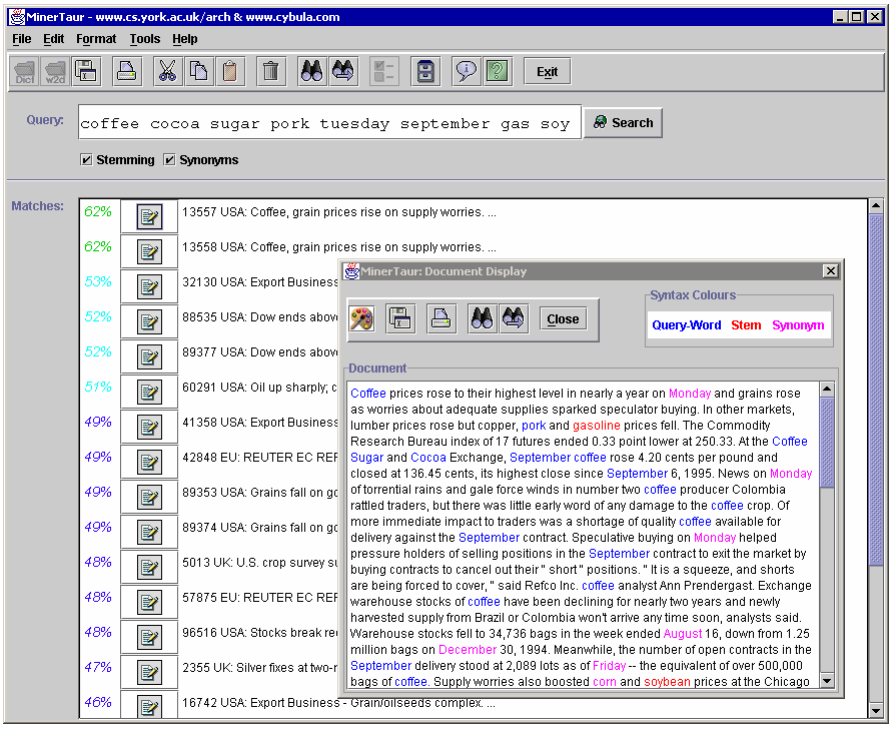
Text Matching.

Text data can be stored in AURA using the string pre-processor and back-check components. The main emphasis of text searching is that it allows the user to match parts of words, i.e. the data items used by the system are composed of individual letters, rather than whole words as found in the document components. These components have been used to build a list search engine (TSE) used to search for matches in lists of addresses (for example). The image below shows a system holding all 27 million UK addresses, allowing a free text search in under a second on a 1GHz PC. The memory needed for the system is about 400Mb. The text matching components allow for deletion, addition and changing of letters in the text as well as incorrect word ordering. More details of this application can be found in the YDS information sheet.



Document searching

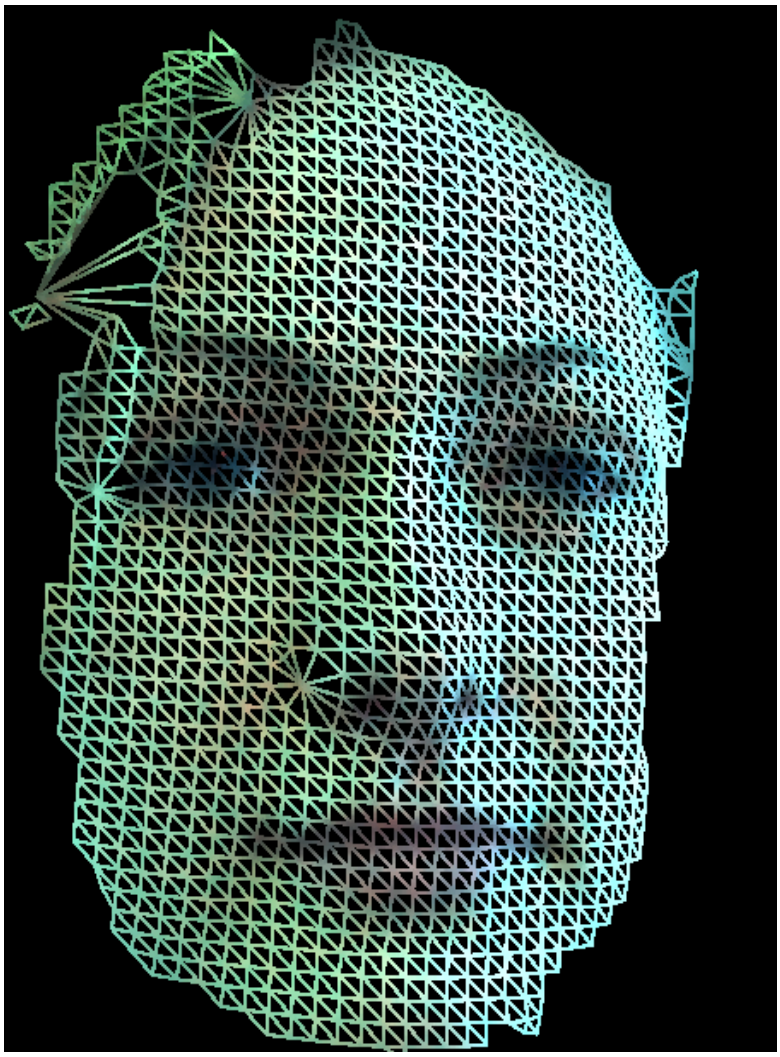
Documents can be stored in AURA using the document pre-processor and back check components. These allow documents to be stored and retrieved by supplying example words to the system. In contrast to the text matching components, the document components use words as the atomic elements of the search, rather than the individual letters. These AURA components have been used to construct a document retrieval system, MinerTaur. An example screen shot of the system is shown below. This tool allows access to large numbers of news articles based on a key word search. The technology also allows spelling errors and synonym matching, added as separate components to the main AURA system.



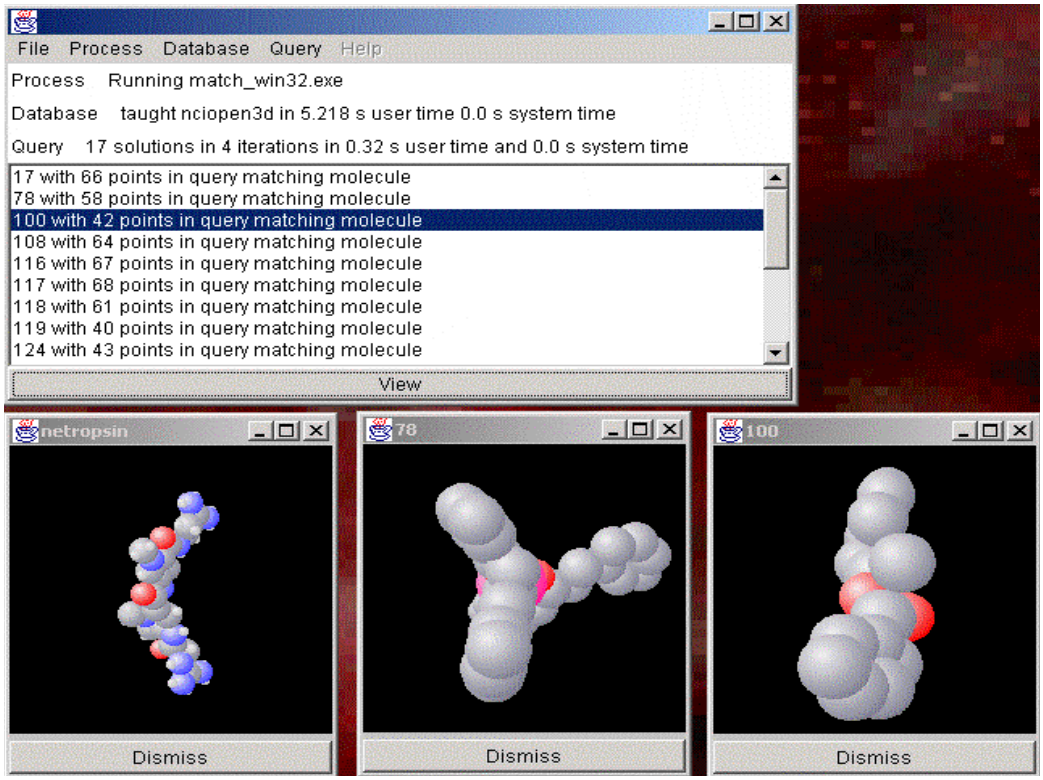
Graph matching.

The AURA graph matching components allow the storage and matching of graphs that may be incomplete. These components are extended through the use of pre-processors that take data and represent it as a graph. They have been used to build image database systems, molecular databases and face recognition systems. A graph is made up of a set

of nodes and vertices. The image below shows a face coded as a graph as read by a 3D camera. This data is used in Cybulas face matching system, FaceEnforce.



Another application of the graph matcher components is the AURA molecular matcher. In this application a molecule is converted to a graph using a special converter. Then it is sent to the AURA graph matching system built from the graph matcher components. It allows a user select to a molecule and search the database for similar shapes. The system encodes the full 3D shape of the object. Other objects could be used such as faces, CAD drawings and graphics models. An example of the system is shown on the next page.



Clearly these components can be applied to many other areas such as finger prints, photographs, engineering drawings etc.

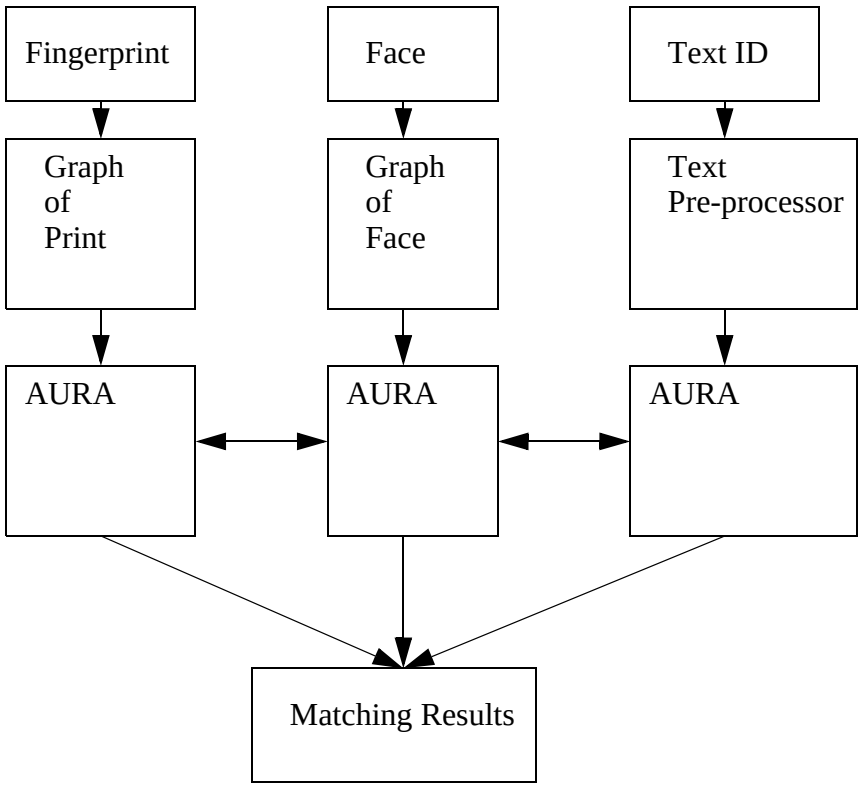
Pattern Classification

The examples given so far use AURA as a powerful database system. AURA is also a versatile pattern classifier, allowing the identification of an unknown item of data. For example, it has been applied to detecting fraud in social security payments where form data is captured and must be given a risk rating to identify the likelihood of fraud. The user enters the form data and the system gives the probable level of fraud. To enable classification, any of the components given above can be used (graph, text, signal, document), but the back check function is replaced by a classification component.

AURA differs from other classification methods in that it allows data to be added at any time to the classifier. No rebuilding of the classifier is required. This allows its use in on line applications where new data is constantly arriving.

6 Data Fusion over different data types.

It will be clear that AURA is a highly flexible pattern matching technology that can be applied to any of the data types listed above. Because of this the ability to search based on multiple data types is easily achieved. There are two ways to do this, either individual AURA systems are constructed for each data type, and in operation the results from each combined. Alternatively, a single pattern match engine is constructed and the AURA system combines the results for you. The latter brings the full power of AURA to bare on the problem. Examples where this can be used is where text and images must be matched. The system can be configured to store both types of data and match, simultaneously on both the image and text data. The exact balance between the two data types may be controlled. The figure bellow illustrates such an approach used for a fusion biometric system incorporating fingerprint, 3D face and text ID for identification.



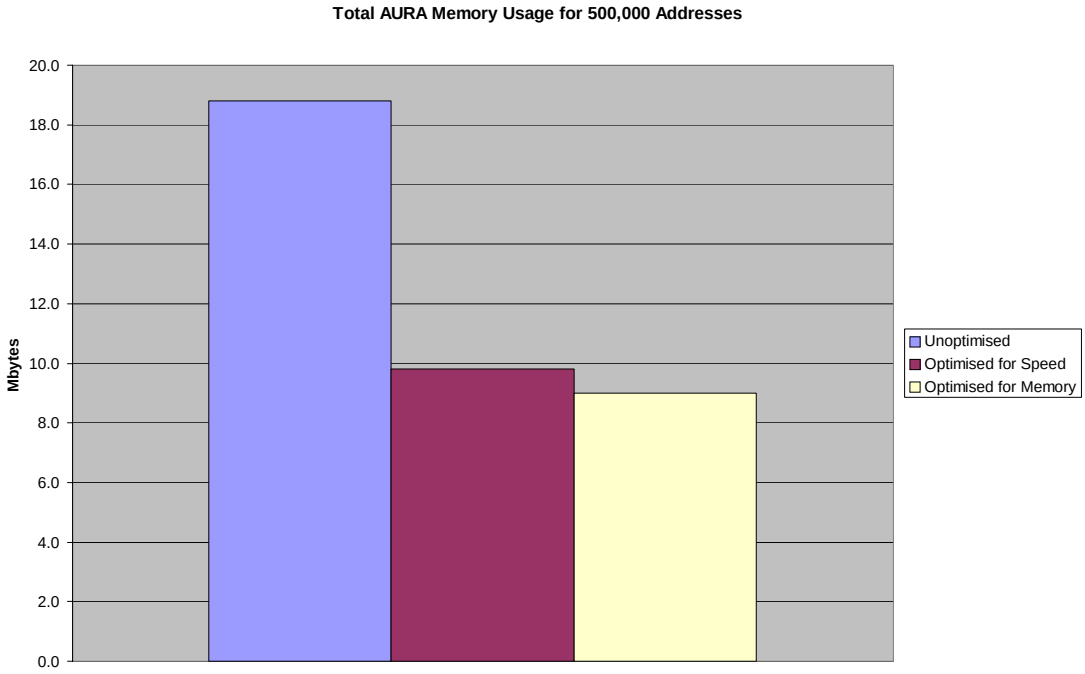
7 Speed and scaleability guidelines.

The advantage of AURA is not only in its flexibility but also in its speed and its efficient use of memory. The following gives a guide to the speed and memory use of AURA

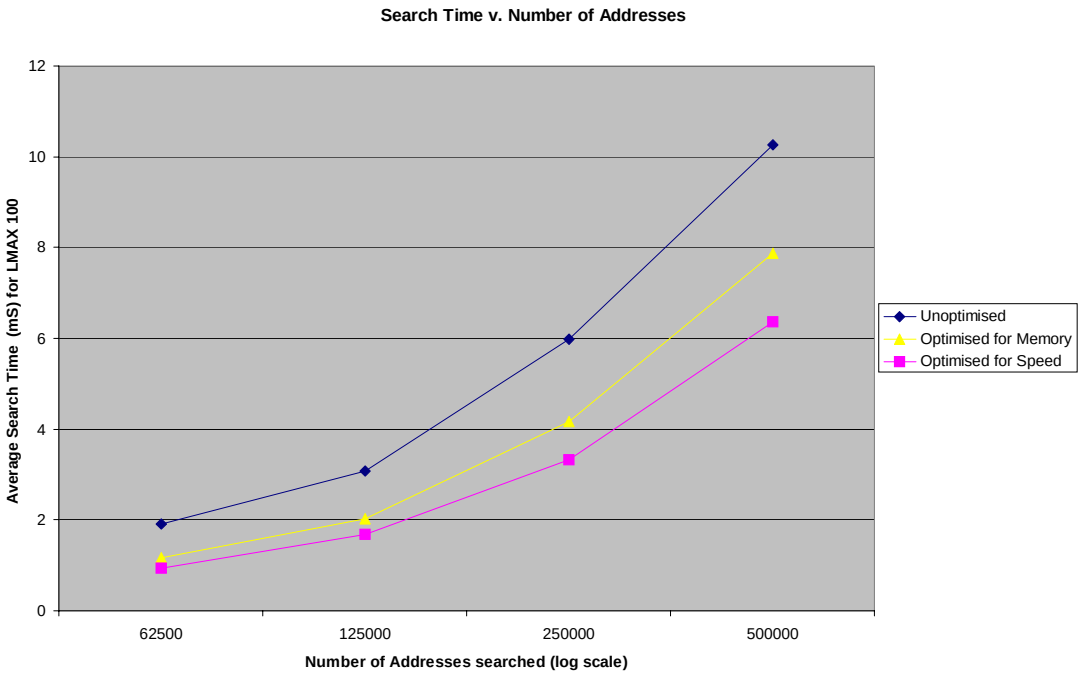
when applied to text searching and searching for molecules in a molecular databases (AURA graph matcher).

Speed and Memory use of a text searching system.

The application example is a system used to store and retrieve address data (the YDS tool). The system has been build using the AURA text components. The system is loaded with various numbers of addresses, the memory used by AURA and the time taken to recall an address are noted. In each case an example of a known address is input to the system and the system recovers the 100 most similar addresses.



The bar graph above shows how the AURA technology can be optimized for speed or memory use, and gives figures for YDS storing 500,000 addresses. The figure on the next page shows the search time for each level optimization (in ms - $x10^{-2}$) for various data sizes when recovering the 100 closest matches to the input. The results show that 500,000 examples can be searched in 0.62 second. These results were collected on a 1.6GHz Pentium 4 processor, 256 MBytes RAM.



Speed and Memory use of the AURA graph matcher.

The following evaluates the AURA graph matcher components when used in a molecular matching system. Timings and Memory use of the software on Sun UltraSparc III 900Mhz Cu processors and 44GB RAM. Each molecule consists of a graph stretched over the surface of the molecule in 3D.

Memory Use.

Table 1 on the next page gives the memory used by the system. DB size is the number of molecules in the database.

Table 1: Memory used.

DB Size	Memory used in Mb
1000	6.5
2000	13
3000	20
4000	26
5000	33
6000	40
7000	46

Timings

The following gives the time to load the database into the system and the time to recall a given example on the Sun platform. The typical graph size representing the molecule is 20 to 120 nodes. In all evaluations the query was the netropsin molecule. The time is given for the main query and generation of result list.

Table 2: Loading and recall times (seconds) for graph matcher

DB Size	Loading	Query
1000	88.7	1.7
2000	411.4	4.4
3000	989.8	6.9
4000	1817.3	9.7
5000	2877.0	12.4
6000	4233.8	15.2
7000	5839.0	17.9

Table 2 shows that 7000 graphs held in the AURA system can be searched in 17.9 seconds given an unknown input graph.

8 Implementation

The AURA components are available as a C++ library and are available to run on a wide variety of platforms (consult our web site for the latest list). To allow high performance

the AURA system has been designed to operate across a number of computers. These can be closely coupled (such as in a cluster, SMP or shared memory multi-processor) or loosely coupled such as machines in geographically different locations. The loosely coupled implementation runs under Globus, an open source distributed system for multiple computers. More details of these capabilities are available in other documents.

For the most demanding applications a special purpose processor card has been developed, based on the PCI bus, this allows a single processor machine containing a PCI bus to support multiple cards. In large installations a very small system with extremely high processing levels can be provided. More details of this hardware is available in other Cybula documents.

9 Glossary

CMM component: Correlation Matrix Memory used to store and find data in AURA based systems.

AURA: Advanced Uncertain Reasoning Architecture. The set of methods that make up Cybulas high performance pattern matching system.

Back Check components: The software in AURA that refines the data that is returned from the CMMs.

Pre-processor components: The software in AURA that prepares data for the CMM.