THE USE OF HADAMARD TRANSFORM AS A DATA COMPRESSION TECHNIQUE IN THE DEVELOPMENT OF A 3-DIMENSIONAL FLUORESCENCE SPECTRAL LIBRARY FOR QUALITATIVE ANALYSIS

by

Fumiko Ishihara

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements fo the degree of

DOCTOR OF PHILOSOPHY

in

Chemistry

APPROVED:

R. E. Dessy, Chairman

B. E. Hanson

G. L. Long

J. G. Mason

J. P. Wightman

March, 1989

Blacksburg, Virginia

To my parents, Mr. & Mrs. , my sister,
and my husband, for their continuous support
and their understanding

THE USE OF HADAMARD TRANSFORM AS A DATA COMPRESSION TECHNIQUE IN THE DEVELOPMENT OF A 3-DIMENSIONAL FLUORESCENCE SPECTRAL LIBRARY FOR QUALITATIVE ANALYSIS

by

Fumiko Ishihara

Committee Chairman: Raymond E. Dessy
Chemistry

(ABSTRACT)

In recent years, chemical instrumentation has become much more sophisticated. Most analytical equipment now incorporates a microprocessor or is interfaced to a microcomputer. As a result, chemists can collect an immense amount of data on a single sample in a short period of time. While there may be an advantage to gathering such a great deal of information, problems can arise from too much information. Today, analysts commonly are faced with the dual problems of storing and analyzing the resulting flood of information.

The goal of this research has been to address the problems of data storage and data analysis. Specifically, data compression techniques and spectral search and match algorithms have been developed. The data compression techniques developed utilize the Hadamard Transform and the modified zero-crossing clipping algorithm. The spectral search technique utilizes the unique format of the

compressed and clipped data to greatly accelerate spectrum identification.

To demonstrate the feasibility of this technique, three-dimensional fluorescence spectra of polynuclear aromatic compounds have been used.

The results indicate data compression techniques and the application of these techniques to a library search system for three-dimensional fluorescence spectroscopy were both successful.

ACKNOWLEDGEMENTS

This research could not have been completed without the inputs and support of many people. There are so many who deserve appreciation for their contributions toward the completion of this research that a list is difficult to generate. Some of them deserve a special recognition which follows.

A special thanks goes to of Perkin-Elmer Corporation. Without the fluorescence spectrophotometer, which granted us, this research could not have been completed.

Another thanks goes to of IBM Almaden Research Center who gave me technical advice and insight for the theoretical and practical aspects of fluorescence spectrophotometry and the fluorescence spectra of polynuclear aromatics

All the members of our research group were a great source of inspiration and created a friendly atmosphere for which I will always be greatful. Especially, I would like to thank Dr. Ian Chapple and Dr. Steve Conder for their helpful advice during the early stages of the research. I also would like to thank Dr. Raymond Dessy and Mrs. Lee Dessy who provided a necessary environment in which this research was developed and completed.

Finally, I would like to thank my husband,.
and my parents for their moral and financial support and providing me with the inspiration to complete this doctoral dissertation.

Table of Contents

List of Figures

List of Tables

Table page

I. INTRODUCTION

In recent years, chemical instrumentation has become much more sophisticated. Most analytical equipment now incorporates a microprocessor or is interfaced to a microcomputer. As a result, analysts can collect a large amount of data in a short time. While there may be an advantage to gathering such a great deal of information, problems can arise from too much information. Today, analysts commonly are faced with too much data and lack adequate methods to handle such burdens in a short time. It becomes important to develop techniques to process data efficiently.

One approach to this problem involves data compression. This technique stores data without loss of critical information. Even casual personal computer users are familiar with ARC programs that can reduce text files by a factor of 4 with no difficulties or loss of material. This research involves the development of such compression techniques for multidimensional spectra and constructing of an efficient library search system.

The project specifically focuses on three dimensional fluorescence spectra (excitation, emission, intensity). The main application of fluorescence spectrophotometers has been for quantitative analysis, with only limited application to

qualitative analysis. The goal of this work was to develop an efficient method for identifying an unknown sample in a reasonable time. This method requires the creation of a library which contains a collection of reference spectra that can be searched reliably but quickly.

The data compression techniques can be applied to each individual spectrum in the reference library. The similarly compressed unknown is then compared in some way with the library. Such compression not only reduces the required storage space but also minimizes the search time.

Two types of compression were applied to each reference spectra and the results for each method were stored in different reference libraries. These are called, after the applied technique, a "filtered Hadamard Transform" and a " clipped" library. These processes will be discussed in detail in succeeding chapters. The "clipped" reference spectral library, whose compression is much greater than the other, is used for preliminary comparisons in order to discard the most unlike candidates. Only those selected at this stage are subjected to the next comparison which uses the "filtered Hadamard transform" reference library. This two stage comparison makes library search efficient and reliable. Before examination of these techniques an historical review is necessary.

II. HISTORICAL

A. LIBRARY SEARCH

Because of the concurrent development of sophisticated analytical instruments and computer interfacing technologies, chemists are faced with an overabundance of data. It is becoming essential to efficiently treat these data to produce a distilled set of information and to interpret the results quickly and correctly.

A classical approach to spectral identification involves library searching methods. Many chemists have studied library searching techniques for Mass Spectrometry and Infrared Spectrometry.[1-11] These techniques are discussed individually in the following section.

Identification of Unknowns using the Discrepancy Factor

One of the early computerized library searching techniques was developed by L. Crawford and J. Morrison[1] in 1968. To evaluate the similarity of an unknown mass spectrum and a reference mass spectra, they utilized the discrepancy factor, which is expressed as follows,[1]

$$D=\Sigma \mid Pn_{ref} - Pn_{unknown} \mid \qquad (1)$$

where P is the normalized peak height at a m/e value of n. and k is the number of observed peaks. However, before the discrepancy factor can be calculated, the peak heights need to be normalized. One of the simplest normalization methods

is "to make the sum of all the observed peak heights, Pn, equal unity", i.e.,

$$\Sigma \; Pn = 1 \qquad (2)^{1}$$

If the unknown and the reference spectra match completely, then D will be Ø.Ø. If all of the peaks from unknown spectrum are absent in the reference spectrum, then D will be 2.Ø. Such a negative correlation is very unlikely. To perform a library search, an unknown mass spectrum is compared with reference spectra and the discrepancy factor associated with each reference spectrum is calculated. The reference spectrum with the lowest discrepancy factor is a likely candidate for identification of the unknown. Because the normalization method described above puts too much emphasis on the largest peaks, more advanced types of normalization methods have been introduced for comparison purposes. The following normalization methods will more evenly distribute the weight of importance to smaller peaks.

$$\Sigma \; \sqrt{Pn} = 1 \qquad (3)$$

$$\Sigma \; \sqrt[3]{Pn} = 1 \qquad (4)$$

$$\Sigma \; Pn^{2} = 1 \qquad (5)^{1}$$

As the normalization method is changed, the discrepancy formula should be changed accordingly. For example, if the last normalization method (5) is used, then the discrepancy factor formula is changed to the following:

$$D = \Sigma \; (Pn_{ref} - Pn_{unknown})^{2} \qquad (6)^{1,2}$$

This particular dissimilarity is referred as "the least-squares comparison metric".[2] This least-squares comparison metric and the discrepancy factor based on the absolute difference methods have been used by Grotch.[15-18] His study is explained in detail in a later section.

The results reported by Crawford and Morrison support the applicability of discrepancy as a suitable criterion for library searching. These workers also studied the effect of random errors in peak heights on the discrepancy factor. These random errors do increase the discrepancy value by 10% to 20%, depending on the normalization method employed. The effect of random errors can be minimized if the second normalization (eq. 3) method is utilized.[1]

Crawford and Morrison have applied their technique to two types of mixtures. One mixture consisted of 90% triethylamine and 10% 2-heptanol, while the other consisted of 70% methyl sec-butyl ether, 20% 2-methyl-3-pentanone and 10% hexanoic acid. The reciprocal of D was used as the criterion for the search. Thus, the two or three highest 1/D values should give the correct identification of the components of the mixture. In both cases studied, the most predominant compound in each mixture was correctly identified. However, the 1/D values for the second and the third compounds were similar to that of many other reference spectra, and the results are not conclusive. The use of the

second type of normalization did seem to improve the success of the search method.[1]

These comparisons, between an unknown and reference spectrum, have been done using the entire set of peaks in the spectra. In an effort to improve the efficiency and speed of the spectral search, Crawford and Morris chose the six strongest peaks from the unknown and reference spectra and calculated the discrepancy factors after normalization. Unfortunately, the percentage of correctly identified unknown spectra was considerably lower than the results obtained using the full data sets.[1]

Other attempts have been made to reduce the search times and storage space required. Early studies introduced filtering methods before the actual search started. One method is called "noncommon mass rejection". In this method, the two or three strongest peaks are selected from an unknown spectrum and reference spectra. If these strongest peaks from the unknown and the reference do not have the same mass number, then the match process for that library member is terminated. This method eliminates the comparison of the unknown with extremely dissimilar reference spectra. Therefore, it greatly reduces extra search time.

Another approach has utilized molecular weight information. For each reference spectrum, the molecular weight of the compound is calculated and the reference

spectra are stored in order of descending molecular weight. Before the actual search, the range of the molecular weight of the unknown compound is approximated by the use of the highest recorded mass molecular ion peak. By searching only those reference spectra having a molecular weight within this range, the search time can be reduced.[1]

Identification of Unknown using Similarity Index

An alternate library search technique was developed by H. Herts, R. Hites and K. Biemann[3] in 1971. This technique utilizes a factor called "similarity index". This similarity index deals with the intensity ratio of unknown and reference peaks at the same mass number. These workers used ratios instead of the sum of the differences discussed earlier. The detailed method used to calculate the similarity index is discussed later. Before the actual search, Biemann and co-workers abbreviated the spectrum. They divided each mass spectrum into 14 mass unit interval starting at mass number six. They then extracted the two most intense peaks in each interval, and retained only these peaks as a reference library spectra.[3] This method of compression discards less structural information than a selection involving only the 6 strongest peaks in the entire spectrum.

These workers used a "presearch" to eliminate the comparison of the unknown with reference spectra of unlike candidates. Several requirements have to be met before a

full search is started. First the most intense peak in the reference spectrum must have at least 25% the relative intensity in the unknown spectrum, or vice versa. Next, the number of peaks in the unknown spectrum should be close to that of the reference spectrum. Then the "abundance of homologous series of ions" should be similar in both the unknown and the reference spectra. If the comparison between the unknown and the reference pass the presearch step, then the full search routine is implemented.[3]

To perform this search, a similarity index must be calculated. The detailed procedure to calculate this index is described below.

$$\text{weighted ratio}_i = \frac{P_{\text{reference},i}}{P_{\text{known},i}} \qquad i=\text{mass numbers} \quad (7)^3$$

where $P_{ref.}$, P_{known} are intensities at mass number i.

$$\text{corrected ratio}_i = \frac{\text{weighted ratio}_i}{\text{average ratio from large peaks (>10\% relative intensity)}} \quad (8)^3$$

If the corrected ratio is greater than one, then the reciprocals of that ratio are used.

$$\text{ave.weighted ratio} = \frac{\text{corr.ratio}_i * \text{weighing factor}}{\text{number of the peaks observed}} \quad (9)^3$$

$$\text{similarity index} = \frac{\text{average weighted ratio}}{\text{fraction of unmatched intensities} + 1} \quad (10)^3$$

Theoretically, this similarity index gives 1.00 if the unknown spectrum and the reference spectrum are completely identical. A reference spectrum with a high similarity index indicates that the mass spectrum of the unknown compound is similar to that of the reference spectrum.

The reference compounds with the top ten highest similarity indices are reported by the program. Most of the time, this search routine resulted in correct identification of the unknown. In the cases involving incorrect identification, the correct compound always came in the second place. Also, if the unknown compound and the reference compounds have similar chemical structures then the resulting similarity indices were high. It is interesting to note that of the compounds studied, the range of the similarity indices for the correct identification varies from 0.109 for methyl hexane-1,6-dioate to 0.695 for isopropylbenzene.[3]

Identification of unknown using Euclidean distance and comparison with other methods

Isenhour and Rasmussen[4] have reported the results of a study which used three different methods as comparison criteria of mass spectral data. These methods are the use of an "absolute value distance" (equivalent to the discrepancy factor which is studied by Crawford[1]), the use of a similarity index (equivalent to Biemann's work), and the use of Euclidean distance which can be obtained by

taking the square root of "the least-squares comparison metrics" factor.[4]

$$\text{Euclidean distance} = [\Sigma(I_{Uj} - I_{Lj})^2]^{1/2} \quad (11)$$

where I_U and I_L are the peak heights of the unknown and library spectra at mass number j. The study indicates that all three methods give equally successful searches.[4]

Probability Based Matching Techniuqes

One of the recent developments in library search techniques is the introduction of a Probability Based Matching (PBM) system. PBM is a statistical technique that calculates the probability of each peaks appearance in a spectrum. A series of reports based on this technique has been published by Venkataraghavan, McLafferty and co-workers[5-11]. PBM is based on the "General Rule of Multiplication" of probability theory[12], whose basic principle is stated in Equation 12.

$$\text{overall probability} = \pi\, p_i \quad (12)^6$$

This equation states that "if n independent events occur with probabilities p_1, p_2,...,p_n then the probability of all n of these events occurring"[6] is the product of all independent probabilities. In the case of Mass Spectrometry, p_i is the probability of occurrence of a peak at mass m_i with the intensity of i_i. Therefore the probability of two peaks with independent probabilities p_1 and p_2 appearing in the same unknown spectrum is given by $p_1 \times p_2$.[6]

The PBM system examines an unknown spectrum for the presence of a library reference compound and calculates the probability that the compound is present, which is called the 'Confidence Index', K. It is important to note that this system involves a 'reverse search', indicating it examines for the presence of a reference spectrum in the unknown compound. Forward searching systems examine for the presence of peaks from an unknown spectrum in the reference spectra. In this latter approach, if some extra peaks are found in the unknown spectrum, then there is a lowering of the degree of similarity, which may be considered as evidence for contamination. However, for the reverse search, it does not matter how many extra peaks are present in the unknown spectrum as long as all the peaks from the reference spectrum are present in the unknown. The advantage of a reverse search is that not only pure compounds but mixtures can be identified.[6]

The Confidence Index, K, which is used as a measure of this library search system, is expressed below. Each probability is expressed in base two logarithms.[5]

$$K = \Sigma K_j = \Sigma\ (U_j -, A_j - D + W_j) \qquad (13)$$

K_j is calculated for the selected peaks in an unknown spectra, and these are then summed to evaluate K. In general, the presence of peaks at a certain mass regions is much less common than in other parts of the spectra. For example, it is less common to have peaks at higher masses

than to have peaks around mass units of 30 to 50. This suggests that if peaks at higher mass units are present, then they can be used as a unique characteristic of the unknown spectra. The probability U_j, shows the degree of importance and the uniqueness of a given peak. The higher the mass of a peak, the larger its U_j values will be. [A_j is based on the abundance of the peak in the reference spectrum which modifies the uniqueness calculated in U_j. (U_j - A_j) are determined using the reference spectra.] A dilution factor, D, takes into account the fact that an unknown is a mixture. This requires an overall reduction of peak intensities due to the presence of other compounds. For a pure compound, this dilution factor is zero.

If an unknown and the reference spectra involve the same compound, then the relative intensities of peaks within the reference spectrum and those of the unknown spectrum should agree within a reasonable experimental and instrumental error. Window tolerance, W, takes into account that all measured spectra will have some kind of error. A higher tolerance limit allows more room for experimental and instrumental error. This higher tolerance limit is expressed as a smaller W value, which is expressed in percentile form.

The actual searching technique described in the initial report by McLafferty[5] is discussed below. Those peaks with highest (U_j - A_j) value are examined first. In the initial

report 15 peaks were used for the identification. Window tolerances used were 20% and 37%.

After the peak order is determined, K_j can be calculated starting with the peak with the highest $(U_j - A_j)$ value. After the specified number of peaks are examined, then the final confidence value, K, can be calculated by totaling the individual K_j values. The reference spectrum with the highest K value should be the correct identification. The library size used in this search was less than 100 spectra.

The results of this technique applied to a library search involving drugs such as heroin, methamphetamine and amobarbital dimethylacetal have been described. For these substances, the correct compounds from the reference library gave the highest K value, in comparison to those provided by unrelated compounds.

Factors such as sample size, impurities, U and A values, and window tolerance all affect the resultant K values. For example, a large sample size (10^{-5}g) and a high molecular weight unknown result in a high K value, generally in the range of 75 to 125. Even for a smaller molecular weight unknown, if a large amount of sample is used, the resultant K value is about 80 to 100. For a smaller sample size (10^{-7}g), the K value found is normally about 20 to 40. However, regardless of conditions, the K values of unrelated compounds are much lower than those for correct compounds.

Typical values are less than 10. This system is, however, very susceptible to impurities in the compounds. The author suggests that to improve these results a much larger number of peaks needs to be examined.[5]

The second report by McLafferty, Venkataraghavan and co-workers shows the use of the PBM system with a large library of about 30,000 spectra.[6] This system was applied to pure compounds as well as mixtures. To accommodate the use of reference spectra from diverse sources, which result in different reproducibilities and possible introduction of impurities in the reference, some improvements in the PBM system were made[6]. In the previous system the unknown peaks were examined for the presence of reference peaks. If a peak was present, but the intensity ratio of unknown to reference was smaller than a specified level, the comparison of the unknown spectrum with the reference spectrum was terminated. The improved version also introduced the following improvement. If a peak is missing in the unknown spectrum, or the intensity ratio of peaks is smaller than the minimum level allowed, then that peak is "flagged".[6] However, the comparison of other peaks continues. If the number of missing peaks (i.e., flagged peaks) surpasses a predetermined number, then the comparison of the rest of the peaks is terminated. After the K value calculation, if the resultant value is larger than the threshold, then it is

stored as a result. The threshold level used in the literature report was K=25.

Some new terminologies were introduced to evaluate the results of this system; these are delta K, recall and reliability. Delta K is the difference between the calculated K value and the maximum K value possible, which would result from a perfect match between unknown and reference spectra. Recall is the proportion of all possible matches which are actually retrieved. Reliability is the ratio of the number of correct compounds retrieved to the number of all retrieved compounds.

The unknown compounds are divided into " a 'Low Molecular Weight Set' (LMWS, mol wt 144-160 amu) and a 'High Molecular Weight Set' (HMWS, mol wt 232-312 amu)". Plots of recall/reliability were made for different threshold values of K and delta K values. Increasing the threshold K value or decreasing the delta K value increases the reliability, but decrease the recall. However, the recall/reliability results using K and delta K show some discrepancies between LMWS unknown and HMWS unknown. For example, in the 50-80% recall range, using K as threshold criteria for the HMWS unknown gives higher reliability than using delta Kas the threshold criteria. For the LMWS, using delta K gives higher reliability.[6]

As noted earlier, this system was applied to unknown mixtures. One type of mixture consisted of three low

molecular weight organic compounds, while the other included one from among high molecular weight compounds. The ratio of compounds in the mixture was 60%, 30%, and 10% in both types. For LMWS, all three compounds were retrieved correctly and had the highest K values. All other compounds retrieved from the library as possible matches were structurally similar to the unknown. In the case of HMWS, again, all three mixture were retrieved. However, the compounds with the three highest K values often did not match with the unknown, although they were structurally similar to the unknown compounds. Similar searches were done using only two high molecular weight compounds, Methyl n-Octadecanoate(90%) and Methyl cis-9-Octadecenoate. The results were improved drastically by examining more peaks, ranging from 16 peaks for 170 amu compounds to 26 peaks for $\geq$600 amu compounds.[6]

Spectral Search by Using a Modified Least-Square Method

Thus far all these library search developments have concentrated on mass spectra. However, many scientists have investigated search techniques for infrared spectral identification. One such study has been performed by S. Lowry , D. Huppler and C. Anderson.[13] The original criterion used in this search technique is a least-squares calculation that was modified so as to use the absolute value difference;

$$M_{sq} = \Sigma (x_i - y_i)^2 \qquad (14)^{13}$$

$$M_{ab} = \Sigma |x_i - y_i| \qquad (15)^{13}$$

The reason for the modification was to reduce the computation time. To evaluate the search performance of M_{sq} and M_{ab} functions, an unknown spectrum which is not a member of the reference libray was selected. Ideally, the search technique should return reference spectra which are structually and chemically related to the unknown spectrum. The closer the similarity between the selected reference spectra and an unknown spectrum, the better the search technique is. The results seem to indicate that the use of M_{ab} gives a closer relationship between the selected reference spectra and the unknown.

A second modification was required to accommodate base-line drift. Instead of using the absolute intensity, the derivative of intensity between two wave numbers was used. These are expressed as follows:

$$M_{sd} = \Sigma [(x_i - x_{i+1}) - (y_i - y_{i+1})]^2 \qquad (16)13$$

$$M_{ad} = \Sigma |(x_i - x_{i+1}) - (y_i - y_{i+1})| \qquad (17)^{13}$$

Search results using p-bromofluorobenzene as the unknown spectrum with base-line drift indicates that the modified versions, M_{sd} and M_{ad}, both correctly identify the compound with a 99% match rate. However search results using the unmodified version, M_{sq} and M_{ab}, could not identify the correct compound at all. Flexibility was introduced by adding a spectral blanking ability so that, if desired, only

the fingerprint regions can be compared (instead of the full spectrum). This increases the speed of the search. In the case of p-bromofluorobenzene, using only the finger print region gives a correct identification with a 99% match rate.[13]

Spectral Search by Fourier-Phase Correlation

One of the most recent investigations on Infrared Spectra library search techniques has been published by S.Kawata, T.Noda and S. Minami.[14] This technique uses the results of cross-correlation between the reference and unknown spectra in the Fourier transformed domain as the searching criterion. The cross-correlation between two functions which are Fourier transformed is expressed as follows:

$$r_{gh}(v) = F^{-1}[G(t)H^{*}(t)] \qquad (18)^{14}$$

where H*(t) is a complex conjugate of H(t), which, in turn, is a Fourier transform of function h(v). F^{-1} indicates an inverse Fourier transform operation. The degree of similarity, n, shown between two spectra can be calculated starting at the origin:

$$n = r_{gh}(0) \qquad (19)^{14}$$

The higher the similarity between the unknown and a reference spectra, the higher the n value becomes. If the two spectra are identical, then n becomes 1. Results for this technique show that the n value becomes 1 only when a comparison is made with between two identical spectra. The

rest of the time, the n value ranges from 0.39 to 0.92. The compounds examined were xylene, benzene, toluene, hexane, cyclohexane and methyl ethyl ether.[14]

These workers then modified the technique so that the search results of any two different spectra would give much lower n values than presented above. In the first technique all the reference spectra are first Fourier transformed. Then, using cross-correlation, they are compared with an unknown spectrum which has also been Fourier transformed. The Fourier transform of any spectra or function can be represented by two different types of curves; these are the amplitude (or power) spectrum and a phase spectrum. This modified approach uses only phase spectra. After the reference spectra are Fourier transformed, only the phase spectra are extracted and stored in the library. The same process is done for the unknown spectrum and cross correlation is applied between phase spectra of the reference and unknown spectra. The calculated n value is 1.00, if, and only if, the two spectra are identical. The n values which result from a comparison using two different spectra are lower than those obtained using the original method. The original n values ranged from 0.39 to 0.92. The improved n values ranged from 0.01 to 0.29 showing a tremendous improvement over the original n values.

B. DATA COMPRESSION

To improve the efficiency of library search techniques, many different methods have been studied. We have just examined a few of the algorithms. However, a different approach can be taken to improve the search technique. This approach involves compressing the reference data without losing characteristic information about the spectra. If such a compression technique is applied to the library search system, then not only the storage space for the reference spectra, but also the search speed can be improved dramatically.

Several approaches have already been reported. These include choosing only a selected number of stronger peaks from a spectrum, reducing the number of bits used to store intensity information, and the use of transform techniques. The following historical study discusses such data compression techniques

Data Compression by Spectral Abbreviation

Herts, Hites and Biemann and their co-workers explored the data compression technique known as "spectral abbreviation".[3] As described in an earlier section, they divided each mass spectrum into 14 mass unit interval starting at mass number six, extracted the two most intense peaks in each intervals, and retained only these peaks as reference library spectra. Compared to choosing a few

strongest peaks from the entire spectra, this method of compression does not discard potentially important structural information .

Comparison of Different Data Compression Methods

Isenhour and Rasmussen have reported an extensive study comparing different data compression techniques.[4] This paper included the spectral abbreviation technique described above (e.g. 2 peaks/14 amu), a modified version of the spectral abbreviation technique which retain only 1 peak per 14 amu, a method which retains the 10 strongest peaks, and a method which retains the 10 most significant peaks. The "significance" of a peak is described as follows:

$$\text{significance} = \text{intensity} * \text{mass number} \qquad (20)^4$$

This technique emphasizes the importance of larger mass positions as well as peak intensities. Without any data compression technique, the number of peaks in an average reference mass spectrum was about 50. When only 2 peaks are retained in every 14 amu, the average spectrum in the library has about 20 peaks. The search technique criterion used on this compressed set was a discrepancy factor, D. This involved the absolute difference between unknown and reference peaks, calculated at each mass unit, and summed over the entire spectrum. The smaller the resultant value, the closer the unknown spectrum is to the reference spectrum. After the search, the reference spectra were listed in the order of increasing D values. Ideally, the

correct reference spectrum should be the one with the smallest D value.[4]

At first the reference spectrum which gave the smallest D value was considered as the correct identification for the unknown. Search results using this criterion are developed below. Without any data compression, 29 out of 40 unknown spectra were correctly identified. With the use of spectral abbreviation, such as 2 peaks per 14 amu, or 1 peak per 14 amu, the method also correctly identified 29 and 28 unknown compounds out of 40 unknowns, respectively. However, when the 10 most intense peaks and the 10 most significant peaks were used for the comparison, only 14 and 15 unknowns, respectively, were correctly identified. Overall performance was improved when the correct identification was chosen from not just one, but the 15 reference spectrum which had the 15 smallest D values. A search involving the full spectra identified 38 out of 40 unknowns correctly. Finally, with the spectral abbreviation methods, which involve 2 peaks per 14 amu and 1 peak per 14 amu, 36 and 34 unknowns, respectively, were correctly identified. When the 10 most intense peaks and the 10 most significant peaks were used for the comparison, 18 and 22 unknowns, respectively, were correctly identified. This study, clearly shows that there is a trade off between the amount of data compression and the performance of the search. The goal of data compression methodologies is to increase the compression

rate without losing significant information about each spectrum.[4]

Data Compression by using the Data Encoding methods

Another type of data compression technique has been extensively studied by Grotch.[15-17] His technique involved reducing the number of bits required to store the intensity of a mass spectrum. Without data compression, 14 bits are necessary to store one peak intensity. His first approach was to use only one bit for each peak intensity, so that all the spectra would be expressed by a simple binary bit string. If the intensity of a peak is greater than the specified threshold level then it is encoded as 1, otherwise it becomes a Ø. This technique put emphasis on the position of the peaks rather than intensity.[15]

He then demonstrated that these one bit coded binary patterns still contained significant information about each compound. In order to use these binary patterns for reference spectra, they need to retain specific structural uniqueness for each chemical compounds. A library file contained 3246 mass spectra which were all encoded into binary patterns with a threshold level set at 1.Ø% of the base peak. Then each of these spectra was compared with every other spectrum in the library file, resulting in approximately 5.2×10^6 comparisons. Among these, only 15 cases reported a perfect match between two different spectra. It is interesting to note that all pairs with the

same binary patterns were close isomers, such as 1-butene and 2-butene. This suggested that these binary coded data sets still retained chemical information.

Grotch's next approach was to introduce another type of compression technique.[16] In this work, the peak intensities were encoded into one of 8 levels (Ø-7) instead of Ø and 1. The threshold levels were set at Ø.5, 1, 2, 4, 8, 16 and 32% of the sum of the peak heights. Although, this technique is called a three-bit encoding, each peak intensity was packed into 4 bits instead of 3 bits. At each pack, the most significant bit was set to "1" for the reference spectra and "Ø" for the unknown spectra.

Comparison of Different Data Compression Methods

The search results using these two different types of data compression techniques were compared with results obtained using spectra without any data compression. A total of 125 unknown spectra were used in this study. A modified version of the discrepancy factor technique was used as a searching criterion, which involves:

$$C = uN + \Sigma[(XOR)_i - u(AND)_i] \qquad (21)^{16}$$

where N is the number of the peaks in one spectrum. The value of u determined the types of logical functions used for the criterion. If u=Ø, then this criterion uses the XOR function. If u becomes larger, reaching a threshhold value, then the criterion used becomes the XAND function. The combination of two functions (i.e., XOR and XAND functions)

gave better searching results than XOR or XAND function alone. The u value used for this particular search was 3.Ø. After the comparison, reference spectra were listed in descending order of the most likely candidate for the unknown.

Initially, the reference spectrum which ranked first was considered as the correct identification of the unknown. With the use of a one-bit encoding system, 64.8% of the 125 unknowns were correctly identified. With the use of three-bit encoding system, 73.6% were correctly identified. Without any data compression technique, 8Ø.8% were identified.

Next, not only the reference spectrum which ranked first but the spectra which ranked above tenth position were considered as possible correct identifications of the unknown. The results, as might be expected, improved. The percentages increased to 86.4%, 93.6% and 97.6% respectively. These results also indicate the trade off between compression rate and search performance.[16]

Grotch's final modification was to combine the several data compression techniques discussed above.[17] He first used the spectral abbreviation technique,[3] where only one peak in every 14 amu interval is retained. Then all the intensity information was discarded, so that only the positions of the peaks remained. At each interval, it is necessary to encode 14 unique mass positions. Only 4 bits

were necessary to encode these positions, since 4 bits produces 16 possible positions. To perform the library search, the modified discrepancy factor (i.e, eq 21) technique was used as the search criterion. Several u values were tested. However, the use of u=3+log(intensity) seemed to give the best results.

The next step was to encode the peak intensity into 4 levels (Ø-3), which would add 2 bits per peak. The threshold levels using Ø.Ø1, 4.Ø and 22.Ø% of a base peak gave the best results. The search criterion which showed the best results were described as:

$$C = \frac{\Sigma \left| l_{unknown} - l_{reference} \right|}{\Sigma (l_{unknown} - l_{reference})} \qquad (22)^{17}$$

where l denotes the encoded peak intensity which ranges from Ø to 3.

By using only the peak position of one peak per 14 amu, the search system correctly identified 77.6% of 125 unknown spectra. Adding 2 bits of peak intensity information improved the results to 84.8%. If the correct identification could be selected from the first ranked reference spectrum to the tenth ranked reference spectra, the search results in both cases improved to 94.4% and 96.8%, respectively. Although the addition of the intensity information significantly improved the search results, the use of only the peak position of 1 peak/14 amu still could

identify the majority of the unknowns correctly with much lower data storage requirement.

Width-Enhanced Binary Representation Method

Recent work by Delaney and co-workers applied Grotch's one bit encoding technique to Vapor Phase Infrared Spectra.[18] For the case of VPIR spectra, not only the position and the intensity, but also the width of the peaks should be included. Figure 1 demonstrates the generation of "width-enhanced peak representation". This figure shows the generation of 100%, 90% and 50% width-enhanced peak representation. To include the peak width information, the peak was encoded at various fractions of the peak heights. All the encoded peaks are expressed in the shapes of boxes where the width of the boxes correspond to the width of the peak at a specified fraction of the peak heights. If the excessive width is encoded (i.e., encoding at a lower peak height), then the resolution of the peak might be lost. Therefore it is important to determine the optimal fraction of peak height at which level the peak was encoded. They reported such a "width-enhanced binary representation" approach and applied it to Vapor Phase Infrared Spectra.[18]

This compression technique was applied to a library search system using a normalized XOR metric as the comparison criterion. This normalized XOR is calculated as follows;

$$\text{normalized XOR} = \text{XOR/OR} = \text{XOR/IOR} \qquad (23)^{18}$$

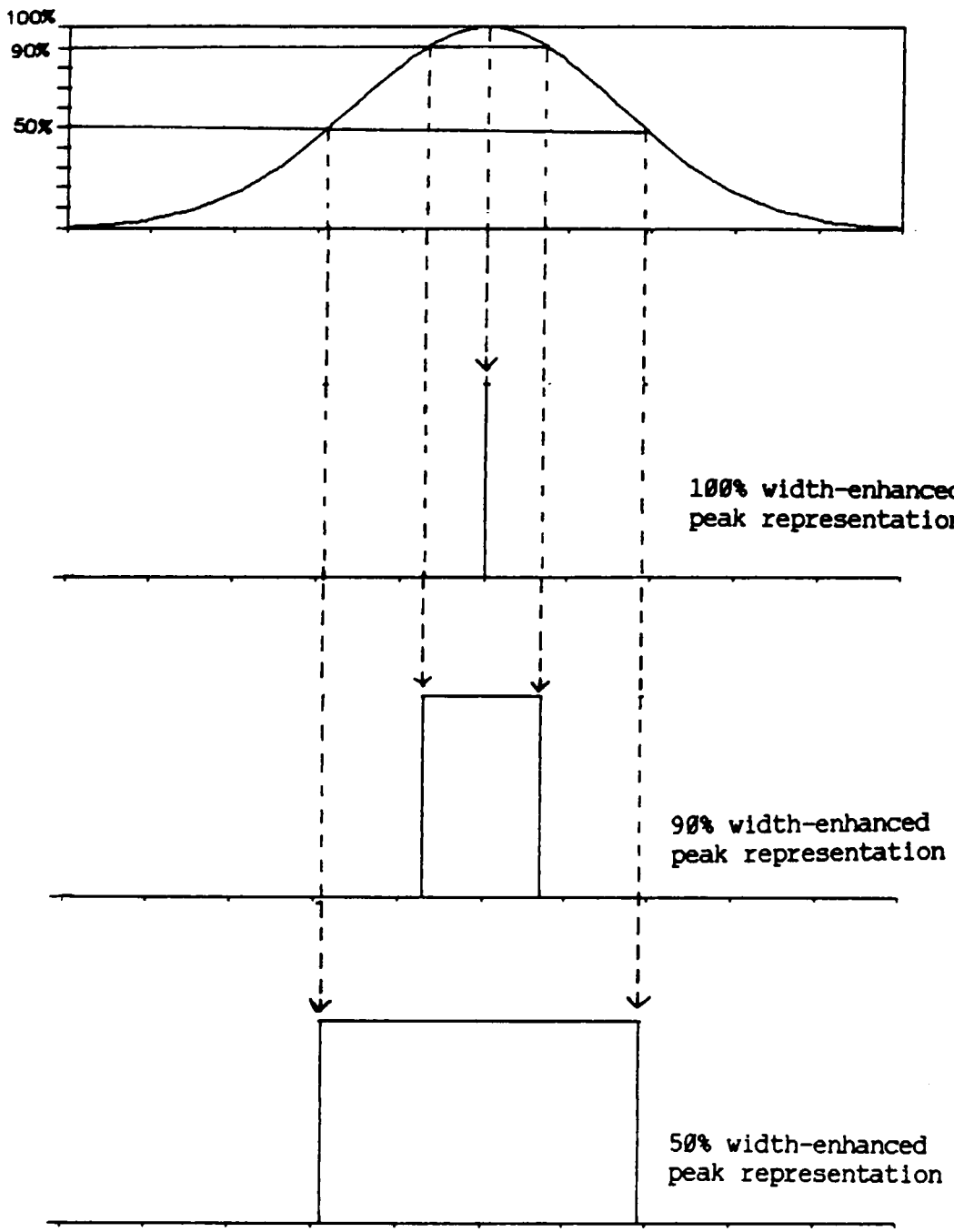


Figure 1. Generation of Width-Enhanced Peak Representation[18]

The results of these comparisons are stored as scores which range from Ø to 1. If the unknown and reference spectra are identical, then the normalized XOR should give a Ø score. The more dissimilar the two spectra are the higher the score value is. Comparisons were done using one of the reference spectra as an "unknown". Therefore if a compression was successful, the search system should correctly identify the unknown with a score value of Ø. Results show that the identification of 1-tridecanol and 1-chloro-2,4-dinitrobenzene both are correctly identified with score values of Ø. The score values with the rest of the reference spectra range from .Ø55556 to .4687500 for the 1-tridecanol search and Ø.5185185 to Ø.5692308 for 1-chloro-2,4-dinitrobenzene search. This comparison was done using the 50% width-enhanced library.

Delaney's next experiment was to determine the optimal width-enhanced representation. As noted earlier the use of a 50% width-enhanced peak representation reference library was successful. However, it is necessary to determine whether 50% was optimal or not. Delaney and co-workers again used 1-tridecanol as an "unknown" and repeated the comparison using different width-enhanced library. The width percentages used were 10%, 30%, 50%, 60%, 70%, 90% and 100%, where 10% contains the maximum amount of width information and 100% contains no width information. The results of the comparison are shown in the Table 1.

Table 1

Library search results using a width-enhanced peak representation as a database[18]

Width (%)	XOR/IOR score		Same score (%)
	nearest	average	
100	0.000	0.289	9
90	0.167	0.334	7
70	0.154	0.294	2
60	0.118	0.357	0
50	0.056	0.355	0
30	0.138	0.276	0
10	0.212	0.268	0

Since the previous experiment showed that 1-tridecanol was correctly identified with a score of Ø, Delaney stored the second lowest score under "nearest" column. The scores from all the reference spectra (excluding that of 1-tridecanol) are averaged and stored under "average" collumn. If these "average" scores are high, then many of the reference spectra are very dissimilar to 1-tridecanol. A higher average score value suggests the uniqueness of an individual spectrum. In some cases, more than one reference spectra generate the same score value. The number of these reference spectra which do not have unique score values are totaled and the percentage calculated as shown below:

$$\text{same score}(\%) = \frac{\left[\begin{array}{l}\text{The no. of reference}\\ \text{spectra which does not}\\ \text{have unique score value}\end{array}\right] * 100\%}{[\text{The total number of spectra in the reference library}]} \quad (24)^{18}$$

The results are stored under "same score". A low "same score" value indicates the uniqueness of the individual spectra. The results show that the average score was maximized and the percentage of similarity dropped to Ø when 6Ø% width-enhanced peak representation reference library was used.

However, a more extended study indicates that the most efficient and optimum reference library can be constructed

when 70% width-enhanced peak representation are used for the reference library.

Other scientists have taken different approaches to the data compression problem. They have utilized transformation techniques, such as the Fourier Transform and Karhunen-Loeve Transform.[19-23]

The use of the Fourier Transform

One technique using Fourier transform was introduced by Isenhour.[19] Here, all the reference spectra are first Fourier transformed, which is denoted as follows:

$$S(f) = \int S(t)\, e^{i2\pi/t}\, dt \qquad (25)^{19}$$

where S(t) is the mass spectra and S(f) is the Fourier domain representation. Then, the transformed data is clipped, using the following rules. A horizontal line across the transformed data is chosen so that average of data above the line and below the line is zero. A peak position which is above the line is encoded as a 1, and a peak position below the line as a 0. The formula is denoted as:

$$C(f) = \frac{S(f)}{|S(f)|} \qquad (26)^{19}$$

An original mass spectrum contains 256 points. After Fourier transformation, the imaginary components were set to zero. The real components were clipped producing 256 data points represented as a logic 1 or 0. The unknown was also Fourier transformed, clipped and compared with the reference

spectra. The comparison method utilized was an exclusive-nor (XNOR) logical method. After the comparison of an unknown and a reference spectrum, the results of the XNOR at each data point are totaled. If the two apectra were identical, the total should be 256.

Their library search system was first tested using one of the reference spectra as an "unknown". Results using 4-methyl-1,3-thiazole and 1-bromododecane as unknowns are described below. In both cases the compounds were correctly identified with a perfect match. For 4-methyl-1,3-thiazole the reference spectra which placed 2nd and 3rd were 4-methyl-1,2-thiazole and 5-methyl-1,2-thiazole. For 1-bromododecane, the compounds 1-bromoundecane and 1-bromodecane were placed 2nd and 3rd, respectively. This suggests that the search system recognizes reference spectra that are similar to the unknown in structure.

Next, this system was applied to 40 real, unknown compounds. The comparisons were performed using complete spectra, clipped Fourier transformed spectra and unclipped Fourier transformed spectra. The results of the search performance showed no significant differences among these three reference libraries. By using these different spectral forms, 29 to 30 unknowns were matched correctly with the first ranked reference spectra. When the correct identification was chosen from not just one, but the fifteen reference spectra which had the fifteen largest totals,

overall performance was improved. With the three different spectral forms mentioned above, 36 to 37 unknown compounds out of 40 unknowns were correctly identifed. These results are comparable to other compression methods, such as the Biemann search and 1 peak/14 amu methods.

Isenhour also introduced the concept of a "prefilter". In a prefilter, the reference library and unknown are compared using only the first 16 bits of the clipped form. If 12 out of 16 data bits match, then a full 256 point comparison is done; otherwise the reference spectrum is regarded as an unlikely candidate and the search is terminated at the prefilter step. This prefilter step eliminated about 85% of the reference spectra from full comparison, and thus improved the speed of the search.

Isenhour's group compared the compression rate of their technique with others. A result shows that Biemann technique and the 1 peak/14amu method can compress data to 42% and 21% respectively. Isenhour's technique compresses data to 16%.[20]

The use of the Karhunen-Loeve Transformation

Another data compression method uses the Karhunen-Loeve Transformation.[21] The Karhunen-Loeve transform is defined below. Let f(x,y) be the orginal data set with i by j dimensions, then f(x,y) is converted into one vector, f(z). The correlation matrix of the data set is estimated as follows;

$$[R] = 1/(i*j) \sum [f_k(z)]^T [f_k(z)] \qquad (27)^{24}$$

Then [A] is calculated so that

$$[A][R][A] = \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ \emptyset & & & \alpha_{i*j} \end{bmatrix} \qquad (28)^{24}$$

where $\alpha 1 \geq \alpha 2 \geq \cdots\cdots \geq \alpha_{i*j}$ are the eigenvalues of [R] and are arranged in descending order. The Karhunen-Loeve transform is as follows;

$$[F(w)] = [f(z)][A] \qquad (29)^{24}$$

where F(w) is the transformed data of the vector form of the orginal data set. Although a matrix [A] is a i by j matrix, the actual dimensions utilized are i by m where m is always smaller than j. Because the final dimension of the matrix F(w) depends on the dimension of [A], the choice of the value m is important. Since the value m is always smaller than the original dimension, data compression is achieved. However, the optimum compression value is not defined. Isenhour used a trial and error method where library search performance was measured using library sets with different degree of compression (i.e., the different values of m). One of their studies which used an IR spectral library showed that the dimension of a matrix [A] was optimally reduced from 185 (i.e., j value) to 37 (i.e., m value). When a library search was performed using this spectral

libray, 7 unknowns out of 12 unknown spectra were correctly identified. When the correct identification was chosen from not just one, but the first fifteen reference spectra "hits", 10 unknowns were correctly identified.

Isenhour and co-workers eventually presented three different studies using the Karhunen-Loeve Transform to compress an IR spectral library and a mixed IR and Mass spectral library. Their studies showed about an 80% reduction in library size was possible.

C. THREE-DIMENSIONAL FLUORESCENCE SPECTROMETRY

Studies using three dimensional fluorescence spectrometry have been done by G. Christian and co-workers and I. Warner and his co-workers[25-31].

In clinical chemistry, two important criteria for analyzing samples are fast analysis time and small sample size. The determination of ethylene glycol in serum and seleniun in urine are examples.[25,26] Recent developments in fluorescence spectrometry has lead to a new technique which is called Rapid Scanning Fluorescence Spectroscopy, or Video Fluorescence Spectrometer.[26] The sample is excited, not just at one wavelength, but at all wave lengths in the region of 200-400nm simultaneously. Luminescence, excited by different wavelengths, then goes through a polychromator and is detected by a vidicon detector. The vidicon is a

detector which will respond to the proton flux intensity throughout the wavelength region. As a result, three-dimensional data can be obtained. The three variables are excited wavelength, emitted wavelength, and intensity. The three-dimensional fluorescence spectrometer provides more information for the identification of compounds (such as emission intensity decay and the excitation wavelength change) than a single conventional spectrum (i.e., intensity vs. excitation wavelength or intensity vs. emission wavelength).

Professor I. Warner and his co-workers have done pioneering work on the application of Fourier transform on the three-dimensional fluorescence spectra for signal to noise improvement, in both qualitative and quantitative analysis.[27-31] All the studies were done by utilizing video fluorometric data.

One such investigation involved the application of a Fourier Transform filtering technique to two-dimensional fluorescence data.[27,30] They applied different types of low pass filters (smoothing filters) to Fourier transformed two-dimensional fluorescence data. The study included the effects resulting from changing the filter cutoff frequency. If the cutoff frequency is too large, then the data becomes over-smoothed. Once the transformed data are filtered, they were inverse transformed back to the original domain and compared with relatively noise-free data. The calculated

value of mean square error between filtered data and the noise-free data was utilized to find the optimum frequency cutoff for the low pass filter. This optimum was found when about 90% of the higher frequency regions were filtered out. An application of this filtering technique was to extract spectral information from data derived from low concentration samples. The spectrum obtained from a 8.28×10^{-9}M solution of perylene in cyclohexane was buried in the noise. However, after optimal filtering, spectral information characteristic of perylene was successfully extracted.

Another study involved the development of a spectral matching technique applied to Fourier Transformed two-dimensional fluorescence data.[28,31]

Three criteria were used; two of which are developed from cross-correlation-based evaluation and the third one is intervector distances between standard and unknown Fourier transformed data. The cross-correlation-based evaluation uses the results of cross-correlation between the reference spectrum, f(x,y), and an unknown spectrum, g(x,y), in the Fourier transformed domain as a searching criterion. If two spectra are identical, then the cross-correlation function becomes identical to the auto-correlation function..

The cross-correlation between two functions which are Fourier transformed are expressed as follows:

$$\text{Cross-correlation} = [F(u,v) * G^{*}(u,v)] \qquad (30)^{28}$$

where F(u,v) and G(u,v) are the Fourier transform of the excitation-emission matrix data, f(x,y) and g(x,y). Also G*(u,v) represents the complex conjugate of G(u,v). Auto-correlation can then be denoted as follows:

$$\text{Auto-correlation}=[F(u,v)*F^{*}(u,v)] \qquad (31)$$

Auto-correlation in the Fourier domain is calculated by multiplying an array of complex numbers by their own complex conjugates. The resulting matrix consists solely of real, positive numbers; no imaginary and negative real coefficients are included.

After the cross-correlation is performed between an unknown and a reference spectra, the sum of the absolute values from the imaginary coefficents of the resultant cross-correlation and the sum of the negative values from the real coefficents of the cross-correlation analysis are calculated. Since the result of auto-correlation consists of real, positive numbers, if the two spectra are identical both sums should result in zero. The smaller the values of these sums are, the closer the two spectral shapes are to each other.

The third criteria used to evaluate the spectral matching technique was "intervector distances between abbreviated Fourier transforms"[28]. Before intervector distances are calculated, ninety percent of the data are excluded. Intervector distances between standard and unknown Fourier transformed data are denoted as follows:

$$D=[\Sigma\Sigma[Re(i,j)-Re'(i,j)]^2+[Im(i,j)-Im'(i,j)]^2]^{1/2} \quad (32)^{28}$$

where Re(i,j) and Im(i,j) are the real and imaginary values of the Fourier transformed data. Intervector distance is essentially a sum of the intensity differences between the standard and unknown data.

As discussed earlier, these three parameters, the sum of the absolute values of the imaginary coefficents of the cross-correlation, the sum of the negative values of the real coefficents of the cross-correlation analysis, and intervector distances, were used as criteria in the subsequent spectral matching.

Warner and his co-workers stored two-dimensional fluorescence data for twenty five polynuclear aromatic (PNA) compounds as standards. Eleven anthracene derivatives were chosen as unknowns; all eleven were also present as standards. By using the sum of the absolute values of imaginary coefficients and intervector distances as criteria for the matches, all eleven were correctly identified. By using the sum of the negative real coefficients as criterion, about 75% were correctly identified by first place hits and the rest (25%) were identified by second place hits. Many of the anthracene derivatives ranked higher than the rest of the PNA compounds. This method is obviously successful and certainly recognizes spectrally similar compounds.

D. HADAMARD TRANSFORM

Two quite distinct chemical applications of Hadamard Transform techniques are known. One usage of this transform technique applies it to an instrument. Several attempts have been made to utilize the Hadamard Transform technique in the same manner as Fourier transform (e.g. FTIR, FTNMR). This will be discussed in detail in the following section. The second usage of Hadamard Transform is to apply the technique to the data after the acquisition is completed for signal processing purposes.
An extensive discussion of Hadamard transform is given in the theory chapter of this thesis.

Hadamard Transform Applied to Instrumental Design

The development of a Hadamard Transform Spectrometer was originally attempted with the goal of designing a multiplexing spectrometer with an improved signal to noise ratio.[32]

In 1949, Golay designed an instrument which utilized multislits instead of conventional narrow entrance and/or exit slits[33,34]. This system consisted of a grating or prism to disperse radiation, and a mechanical chopper or mask was used to modulate the dispersed radiation. The increase in the amount of entrance and exit radiation results in the improvement in the signal to noise ratio. Although Golay's design was not a multiplexing spectrometer,

it showed the advantage of using multislits (encoding masks) for the instrumental design.

In 1970, Harwit[35] and co-workers and Nelson [36] reported a more sophisticated version of the Golay's instrument. This instruments consisted of a light source, an entrance mask which was series of multislits (encoding mask), collimating optics, grating, decollimating optics, exit mask, postoptics and a detector.

The entrance and exit mask consist of an array of open and closed slits. Radiation passes through only the open slits. The light thus passes through not just one entrance slit but multiple slits (let this be N slits). In any given mask position, half of the slits are open and the rest are closed. Many different combinations of entrance mask pattern are possible. If M different mask patterns are available, then the input light is encoded differently by each of the M mask positions. Decoding of the output by the exit mask is achieved in the same way. By measuring the intensity passing through these two masks with many different combinations of patterns, the radiation spectrum can be recovered. Although the numbers M and N may be different, the authors studied the situation where N = M.

The results shows that the signal-to noise ratio is improved by a factor of order N in comparison with single slit spectrometer. The total energy throughput is also increased by a factor of N. The term Hadamard Spectroscopy

has been used to describe this instrument, because the mathematics and masking process are related to the Hadamard Matrices.

An interesting preliminary study was reported by Phillips and Briotta. The work measured the atmospheres of Earth and Jupiter with a doubly multiplexed Hadamard transform spectrometer.[37] The system contained three entrance slits and nineteen exit slits with a resolution of 0.004. The measurements were taken over a 0.64um segment at a time. The spectrum of Jupiter from 880 cm^{-1} (11.4um) to 770cm^{-1} (13.0um) are obtained. Strong absorption due to the ammonia's vibration band were observed at 870cm^{-1}, 851cm^{-1} and 833cm^{-1}. The authors concluded that the overall profile of the experimentally obtained spectrum agree with the theoretically calculated spectrum. Although this is the preliminary study, this is the first attempt to use the Hadamard Transform Spectrometry outside of the laboratory controlled environment. A more sophisticated version with 15 entrance and 255 exit slits is being constructed.

A second usage of Hadamard Transform has been described by Dillard. He reports a successful application to data compression purpose.[38] An image was Hadamard transformed, and then the intensity of the absolute values of each coefficient were compared. Only a few of the most intense coefficients were retained and the rest were put to zero.

Thus the image is compressed. To reconstruct the image, the Inverse Hadamard transform was applied.[38]

The images used were obtained by a standard TV camera. They were digitized so that they consisted of 6 bit per element of a 480 by 640 pixel array (307200 pixels). As shown in Figure 2, this array (480 rows by 680 columns) was further divided into a 4 by 4 subarray. Thus the images now consisted of 120 (i.e., 480/4=120) by 160 (i.e., 640/4=160) 4 by 4 subarrays (i.e., total of 120x160=19200 of 4 by 4 subarrays). These subarray were then Hadamard transformed separately. The transformed data were represented as W_0, W_1, ..., W_{15}. Since the first coefficient, W_0, contains the most of the energy (i.e., the strongest intensity), W_0 is always transmitted back to reconstruct the image. The absolute values of all the other coefficient are obtained and are arranged in the descending order. The author's preliminary assumption was that not all of the data was necessary to reconstruct the original image.

The first attempt in image reconstruction retained only the two largest coefficient (W_j and W_{j+1}) and W_0. The author allocated 4 bits for the index information, j and j+1, for each selected coefficients (excluding W_0 index information). The other coefficients were put to zero. Before the compression, the total number of bits per subarray was 96 (6 bits x 4 x 4). After the compression, the total was 26 [6 bits x 3 coefficients) + (4 bits x 2)]

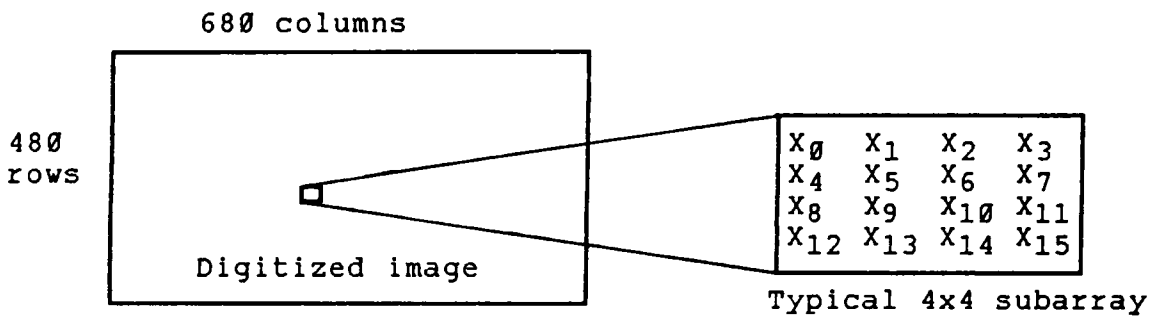


Figure 2. Method of subdividing the digitized image[38]

with a compression ratio of approximately 3.7 to 1. The selected data sets were inverse transformed to reconstruct the original images. The author used "a simple subjective evaluation"[38] of the images to determine the degree of the degredation. The results suggested that the no noticeable degredation was observed.

The second attempt retained only the one largest coefficient (Wj), and $W_{\emptyset}$, for the image reconstruction. The total number of bits became 16 [(6 bits x 2 coefficients) + (4 bits x 1)] with a compression ratio of 6 to 1. The results showed that degredation of the images did occur but the loss of information was minor. The author suggested that, depending on the usage of the transmitted images, this 16 bits per subarray is adequate.

A similar study was done by Pratt. The transformed images were transmitted over a communications channel rather than the image itself.[39] The first attempt used the Fourier Transform. However, any other type of transform can be used as long as certain properties are fulfilled. These properties are that the two-dimensional transform "has an inverse, posseses the averaging property, and redistributes the image energy properly". The Hadamard Transform was eventually selected because it has all the necessary properties as well as the fastest, simplest computational algorithm.

Photographs taken from a cathode ray tube of Surveyor scenes were used on this study. They were Hadamard transformed, transmitted to the receiver end, then inverse transformed to create the original images. The degree of the degredation of the reconstructed images were negligible and the advantage of sending the transformed images, rather than the images themselves, is a high tolerance level to channel errors. The next section describes the Hadamard Transform in detail.

III. THEORY

A. HADAMARD TRANSFORM

The Walsh-Hadamard Transform is generally called a Hadamard Transform. While the Fourier transform decomposes a spectrum into appropriate sine and cosine waves (sinusoids) of different frequencies and amplitudes, the Hadamard transform decomposes the spectra into appropriate square waves of different periodicities (sequencies) and amplitudes.

Because we are dealing with a discrete Hadamard transform, the conventional analog representation of square waves needs to be converted into a digital form. The digital forms of such square waves, the basis of the Hadamard transform, are called Walsh functions, and the allowed levels are represented by the values +1 and -1. As an example, eight sampled Walsh functions are listed below.

Walsh functions	Sequency
1 1 1 1 1 1 1 1	Ø
1 -1 1 -1 1 -1 1 -1	1
1 1 -1 -1 1 1 -1 -1	2
1 -1 -1 1 1 -1 -1 1	3
1 1 1 1 -1 -1 -1 -1	4
1 -1 1 -1 -1 1 -1 1	5
1 1 -1 -1 -1 -1 1 1	6
1 -1 -1 1 -1 1 1 -1	7

Each row indicates the sequency of the corresponding Walsh function. The sequency shows the number of sign changes within the function. This 8 by 8 square matrix is called the Walsh-ordered or " 'ordered' form"[40,42] of the Hadamard matrix.

However, the rows of this matrix can be rearranged to produce another types of Hadamard matrix which has unique properties. This matrix is called Hadamard-ordered or " 'natural' form"[40,42] of the Hadamard matrix. This is the basis of the Hadamard transform used in the present research. A large matrix can be broken up into smaller units. The three smallest (or lowest ordered) Hadamard-ordered Hadamard Matrices are denoted as follows:

Hadamard-Ordered Matrix Sequency

$$H_{2,2} = H(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \begin{matrix} 0 \\ 1 \end{matrix} \tag{33}$$

$$H_{4,4} = H(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad \begin{matrix} 0 \\ 3 \\ 1 \\ 2 \end{matrix} \tag{34}$$

$$H_{8,8} = H(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \qquad \begin{matrix} 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{matrix} \tag{35}$$

where H(1), H(2) and H(3) are 2^1x2^1, 2^2x2^2 and 2^3x2^3 Hadamard-ordered Hadamard matrices. The important thing to note is that the sequency is no longer in an increasing order. If a given matrix is divided into four quadrants, then the first to the third quadrants are identical and the fourth one is the complement of the others. Because of this unique property, a matrix of a high order (meaning a high number of rows and columns) can be reduced to a combination of lower order matrices. For example the matrix, H(3) can be expressed by using matrix H(2);

$$H_{8,8} = H(3) = \begin{bmatrix} H(2) & H(2) \\ H(2) & -H(2) \end{bmatrix} \qquad (36)$$

In general, Hadamard matrices of higher orders can be produced as follows:

$$H_{N,N} = H(n) = \begin{bmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{bmatrix} \qquad (37)^{40}$$

where $n = \log_2 N$

Another property of Hadamard Matrices is that they are both orthogonal and symmetric. This means that,

$$H_{NN} * H_{NN}^T = N*I \qquad (38)^{41}$$

and,

$$H_{NN} * H_{NN} = N*I \qquad (39)^{41}$$

where I is the N by N identity matrix.

One of several possible formulations of the various discrete Hadamard transforms is given as follows:

For 1-dimension

$$F(u)_N=1/N\ H_N(n)\ f(x)_N \qquad (40)^{40}$$

where $n=\log_2 N$

where f(x) is a data vector, F(u) is a transformed representation (sequency domain) of the data vector, $H_N(n)$ is a Walsh Hadamard Matrix and N is the number of data point.

For 2-dimension

$$F(u,v)_{NN}=1/N^2\ H_{NN}(n)\ f(x,y)_{NN}H_{NN}(n) \qquad (41)^{40}$$

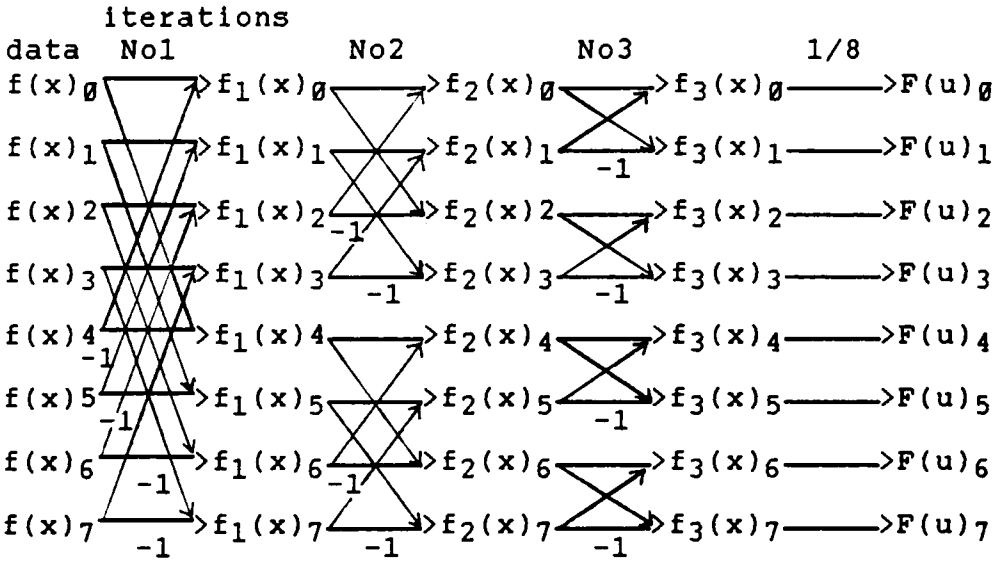
Again f(x,y) and F(u,v) are a data matrix and a transformed representation of the data matrix, respectively.

One can see that this Hadamard transform involves a matrix calculation, which can be time consuming process. However, just as the conventional Fourier transform has an associated Fast Fourier transform method, the Hadamard Transform also has an available method which can perform the operation without involving actual matrix calculations. This method is called the Fast Hadamard Transform (FHT). An example of this FHT algorithm is given below using a one-dimensional transform with 8 data samples.

Conventional calculation;

$$F(u)_8=1/8\ H_8(3)\ f(x)_8 \qquad (42)^{40}$$

FHT algorithm[40,41];



The no. of iterations = $\log_2$(number of data points)=$\log_2 8=3$
The number of partitions = $2^{(\text{present iteration number}-1)}$
Thus for the first iteration, only one partition exists and for the third iteration, four partitions, $2^{(3-1)}=4$, exist.

For the first step, each datum is paired, and additions and subtractions are repeated. This step is repeated for the required number of iterations. The number of such arithmetic operations needed for the example shown is 32. Of these 32 operations, 24 are simple additions and subtraction. On the other hand, if conventional matrix calculations were used for this same example, the number of the arithmetic operations would increase. If the conventional method were used, 8 separate operations are needed to represent the 8 different datum of the resulting 8 by 1 matrix. For each operation, 8 multiplications and 7

additions are involved. Therefore the total number of these arithmetic operations are 64 multiplications and 56 additions. It is obvious that the Fast Hadamard Transform algorithm drastically reduces the computation time as well as the complexity of the calculations.

It is also important to note that while the Fast Fourier Transform involves trigonometric calculations, this FHT algorithm primarily involves simple additions and subtractions. The last step involves modulo 2 division which is simple for binary computers.

B. COMPRESSION

Two different types of compression techniques are used in this research. These are a) a Hadamard Transform plus filtering process and b) a Hadamard Transform plus clipping method. Spectral data compressed by the former method are stored in the "filtered spectra library" (Figure 3) and used for the second step full/complete search of the unknown. Spectral data compressed by the latter method are stored in the "search library" (Figure 3) and used for the first step comparison, called a prefilter process.

1. Hadamard Transform plus Filtering Method

The Hadamard transformation by itself does not compress data, however the transformation converts data into a new domain, called sequency, so that the compression of the data

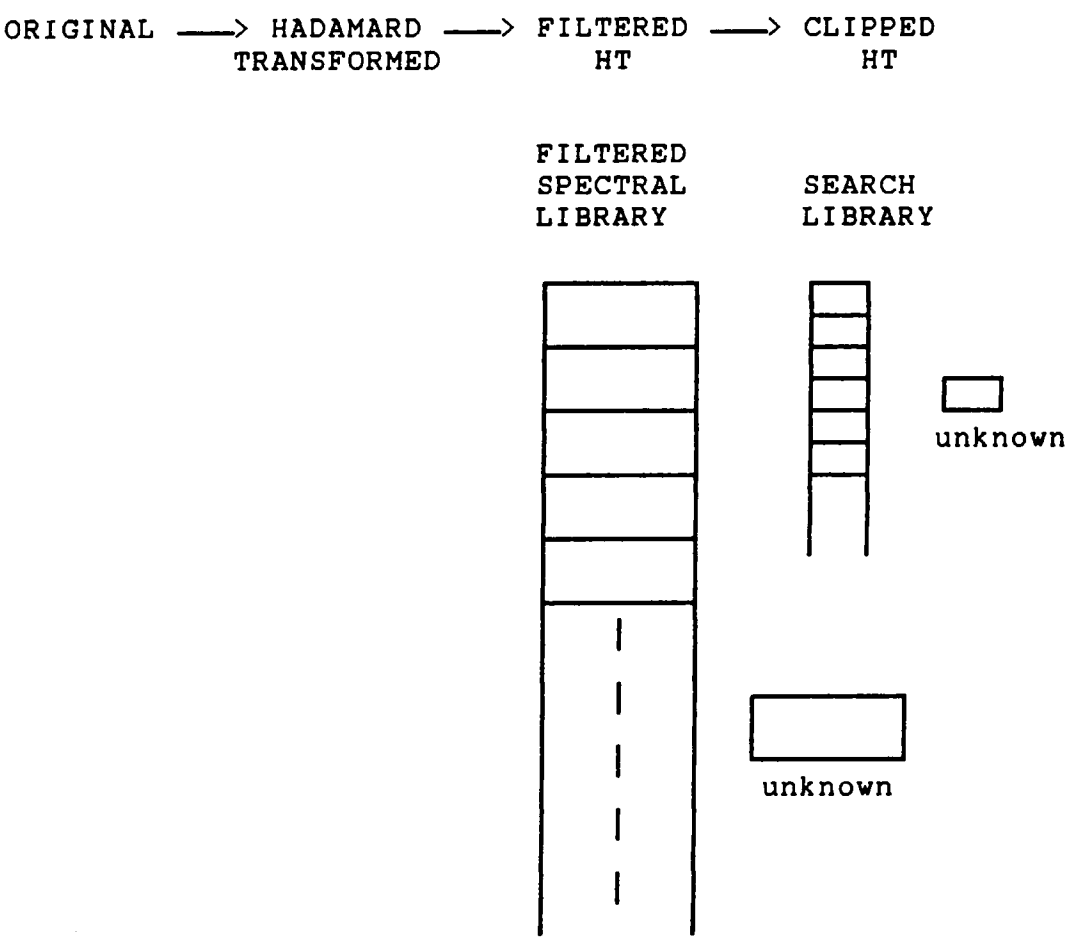


Figure 3. Library Scheme

becomes possible. After the transformation of the data much of the random noise resides at the higher sequency region of the data. Thus, elimination of this higher sequency region should not distort the main feature of the spectral data of the original domain. This compression technique is applied to a three dimensional fluorescence spectrum of anthracene as shown in Figure 4a. The data was Hadamard transformed in to sequency domain (Figure 4b). The next process is to filter out the high sequency region. As shown in Figure 5b, only the lowest thirty two sequencies are retained and the remaining 75%, involving the higher sequency region, was filtered out. This processed data was then Inverse Hadamard transformed into the original domain, which is in wave length. Figure 5a shows that although the resolution of the peaks are not quite as high compared to the unprocessed data, the main features of the anthracene fluorescence data are still displayed. From this study, the compressed data retains characteristic information about features of the spectra and can be stored in the "filtered spectra library" (Figure 3).

Each excitation/emission fluorescence data consists of 64 by 64 data points, stored as 16 bit integer values. As shown below, they are compressed by 75% after the Hadamard Transform.

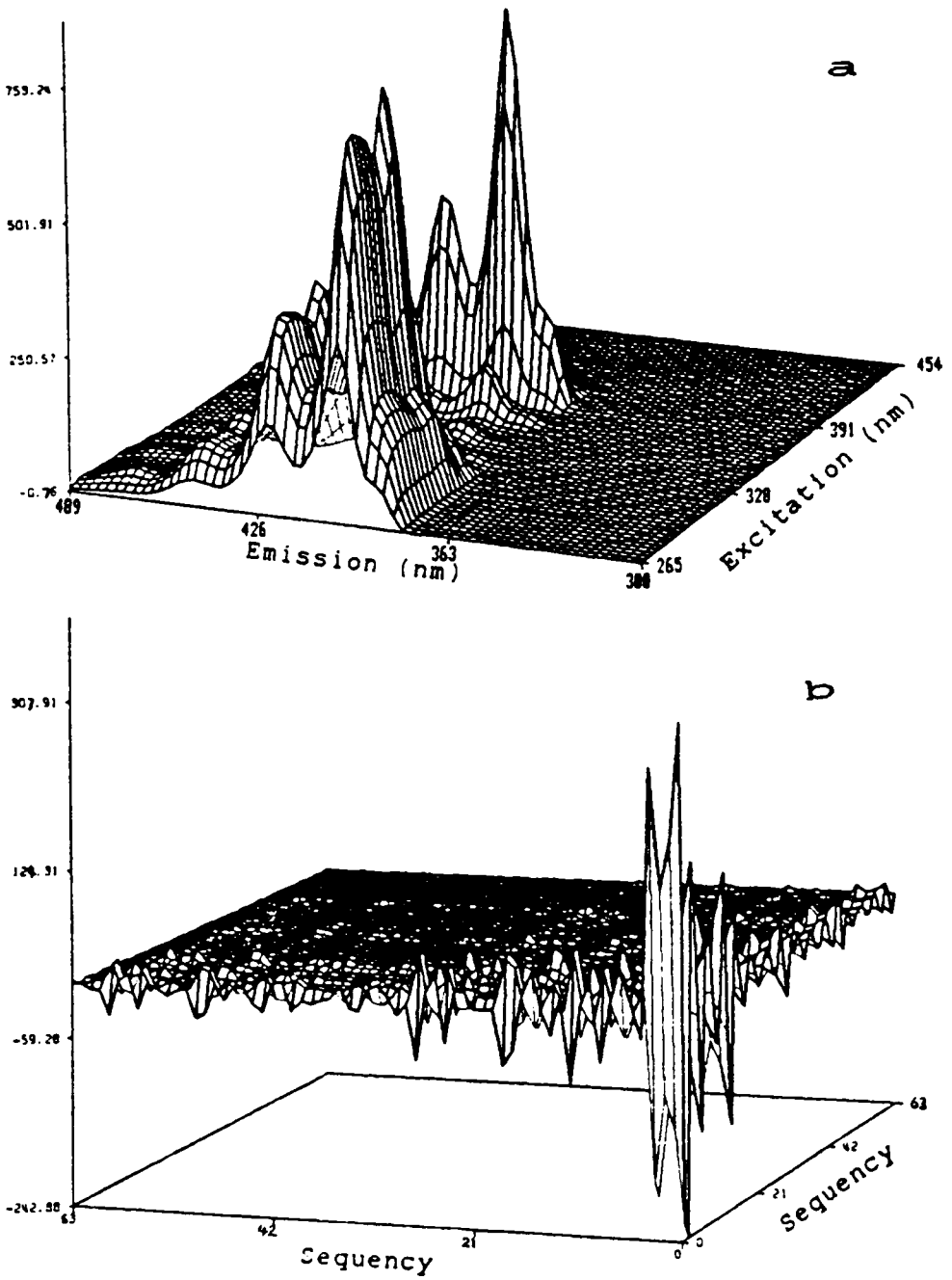


Figure 4a. Anthracene: Original spectrum
4b. Anthracene: Unfiltered transformed data

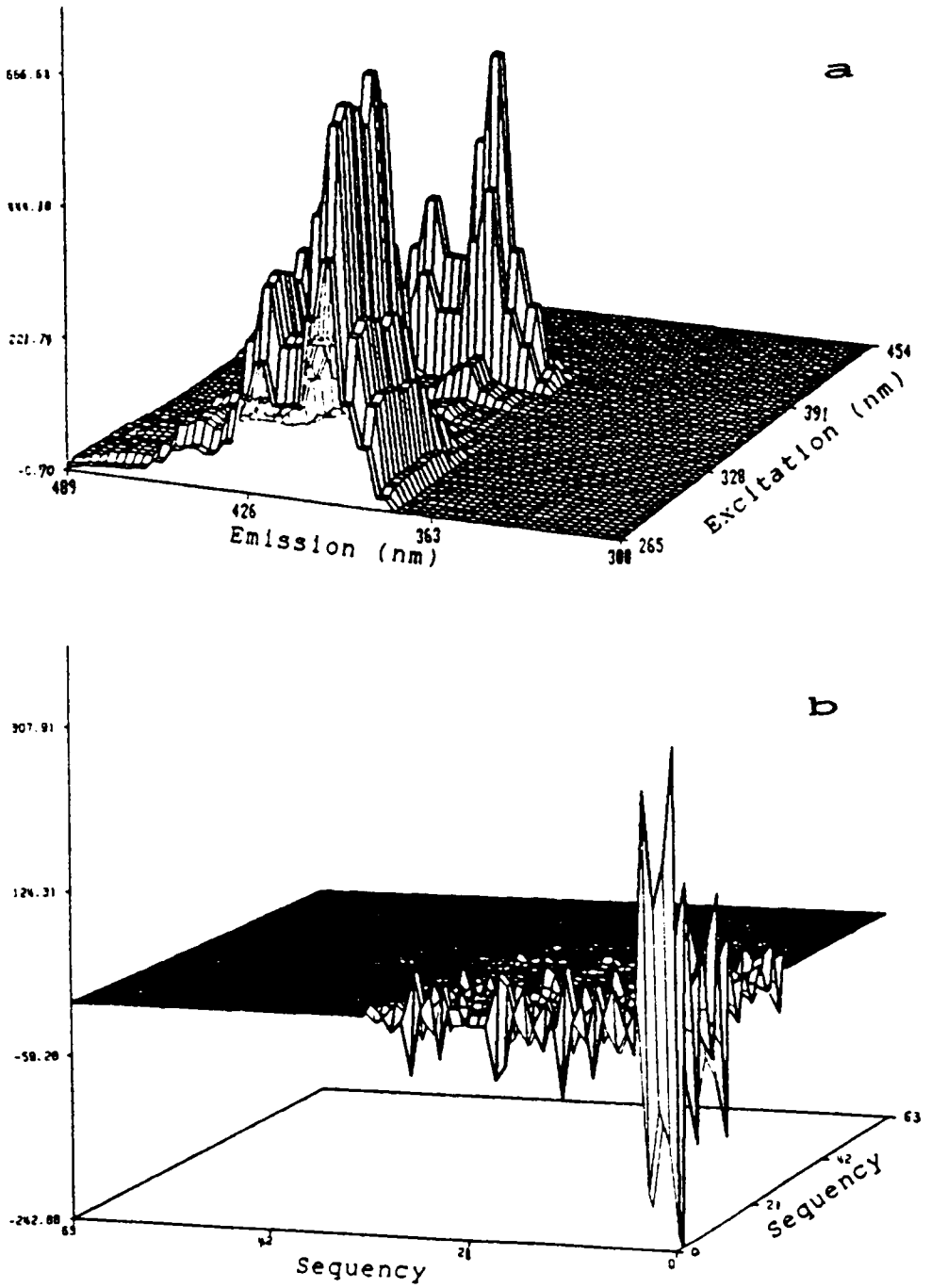


Figure 5a. Anthracene: Inverse transformed spectrum using 6.25% filtered transformed data (32x32 data points retained).

5b. Anthracene: 6.25% transformed data (32x32 lower sequency retained).

64 x 64 x 16bits =4k*16 32 x 32 16bits = 1k*16

100% data base 25% full data base

In the rest of this work, the intact data base is called the "100%" data base and the compressed data is the "25%" data base.

2. Hadamard Transform and Clipping method

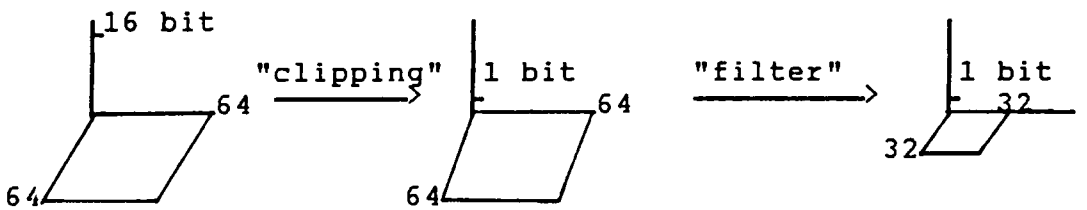
A second type of the compression may be applied to the Hadamard transformed spectral data. The technique is called clipping. It converts the intensity information into 1's and 0's depending on the magnitude of the intensity. This clipping method takes peaks involving large magnitude changes and reduces them to two levels, or binary information. Thus, after the compression, the intensity information, originally expressed by multilevels, is simplified into a binary pattern.

To perform clipping, a horizontal line across the transformed data is chosen so that the average of data above the line and below the line is zero.

if $S(h) \geq$ Average value, $C(h) = 1$
if $S(h) <$ Average value, $C(h) = 0$

where S(h) and C(h) are transformed spectral data and clipped data respectively. Since each datum is expressed as a binary bit, the storage space required is drastically reduced.

The Figure below demonstrates the power of the clipping method as a compression technique.



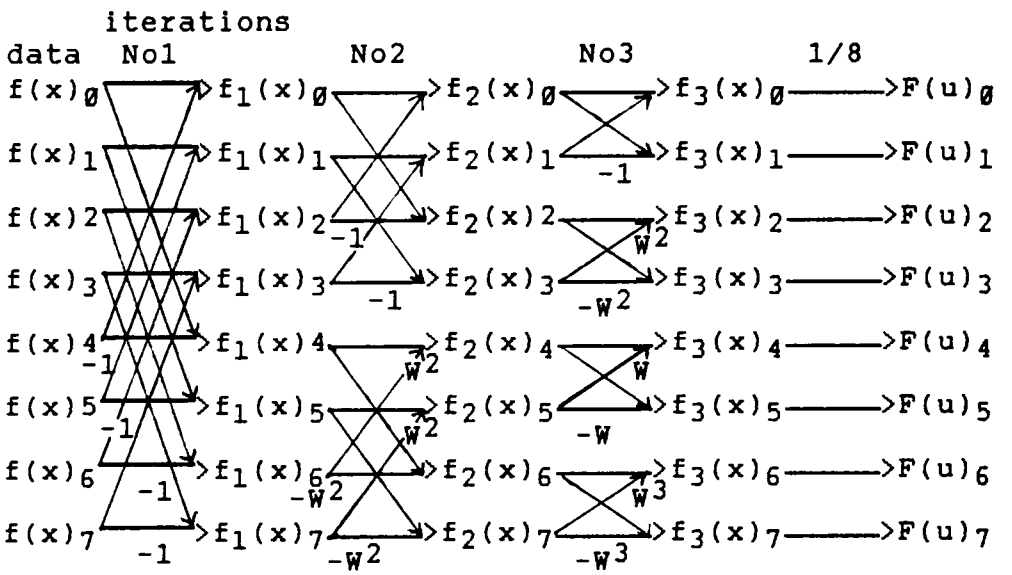
64 x 64 x 16bits	64 x 64 x 1bit	32 x 32 x 1bit
=4096*16 = 4k*16	=4096*1 = 4k*1	=1024*1 = 1k*1
100% data base	**6.25% data base**	**1.56% data base**

Before the compression, the data consisted of 64 by 64 data points stored as 16 bit integer values. This requires 4096*16 bits of memory space. After the clipping, however, the memory space necessary to store this data is only 4096 bits, a mere 6.25% of the original space. Further compression can be achieved by combining the filtering process with this compression method. By filtering out 75% of the higher sequency region, the memory needed to store this data becomes 1024 bits or only 1.56% of the original space.

C. HADAMARD TRANSFORM AND FOURIER TRANSFORM

The use of the Hadamard Transform gives several advantages over the use of the Fourier Transform. The first one is the simplicity of the calculations involved. The following chart shows the signal flows of the Fast Fourier Transform (FFT) algorithm.

FFT algorithm for $N=8$[40];



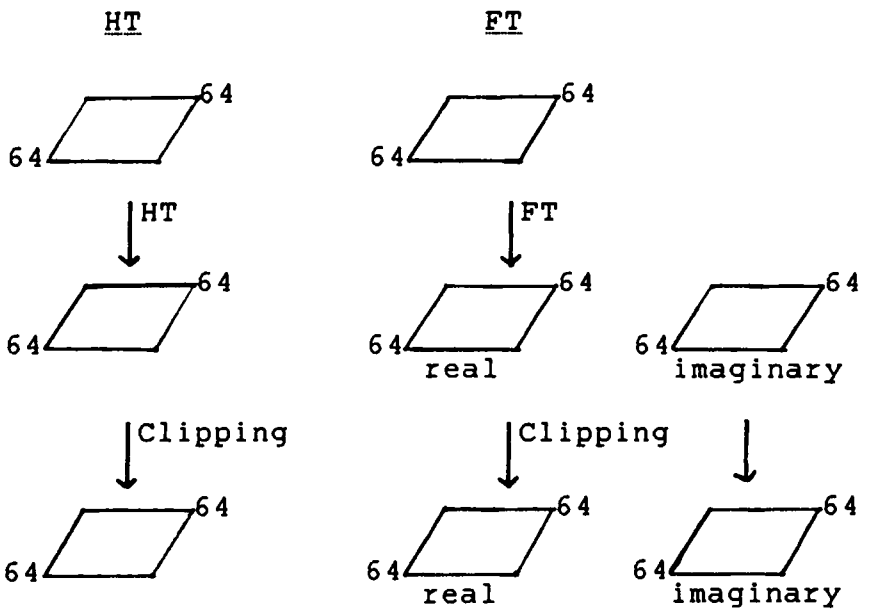
where $W = e^{-i2\pi/N}$

The no. of iterations = $\log_2$(number of data points)=$\log_2 8$ =3

The signal flow of FHT is developed from that of FFT, thus these two are similar to each other. However, for the FHT algorithm, additions and subtractions are the only types of calculations involved except the last step. For the FFT algorithm, additions, subtractions and exponential

calculations are required which makes entire computation more complicated than that of the FHT algorithm.

The second advantages of the use of the Hadamard Transform over the Fourier Transform is the efficiency of the search system.



As shown above, after the Fourier Transform is performed on a data base, to fully retain information, one has to deal with both real and imaginary arrays. The Fourier Transform results in twice as much data as the Hadamard Transform. The use of a Fourier transformed data base implies an increased data base size to be examined during subsequent searches, and a corresponding increases in search time. The use of the Hadamard transformed data base

in the development of a library search system is therefore justified on several grounds.

IV. EXPERIMENTAL

A. MATERIALS

Polynuclear Aromatics (PNA's) and other fluorescent dyes were used as received in this research. A detailed list is given in Table 2. These compounds all generate fluorescence spectra within a desired excitation and emission wavelength region, which is described in the next section. All the solutions were prepared in spectra-grade cyclohexane which was obtained from American Scientific. Some of the selected PNAs were also prepared in 100% ethanol, methanol and in benzene which was also obtained from American Scientific.

B. EQUIPMENT AND DATA ACQUISITION

The fluorometer used in the research was a Perkin-Elmer Model MPF-66 fluorescence spectrophotometer. It was controlled by a Perkin-Elmer 7500 data station.

The excitation wavelength region was from 265nm to 454nm and the emission wavelength range was from 300nm to 489nm. The data acquisition software worked as follows: the excitation wavelength of the fluorometer was set at 265nm and a one-dimensional fluorescence spectrum was taken at 3nm intervals, which results in 64 data points per spectrum.

Table 2. The list of Polynuclear Aromatic compounds

Names	
Anthranilic acid	Salicilic acid
1,1-Diphenylethylene	4-Biphenylphenyl ether
4-Methylbiphenyl	4-Vinylbiphenyl
triphenylamine	Indole
Quinoline	Azulene
Naphthalene	1-Methylnaphthalene
2-Methylnaphthalene	2,3-dimethylnaphthalene
2,6-dimethylnaphthalene	1-Naphthol
2-Naphthol	1-Phenylnaphthalene
2-Phenylnaphthalene	Fluorene
Acridine	Anthracene
1-Aminoanthracene	2-Aminoanthracene
9,10-Dichloroanthracene	9-Methylanthracene
9-Phenylanthracene	9,10-Diphenylanthracene
9-Vinylanthracene	2-Methylanthracene
Tetracene	Phenanthrene
Chrysene	1,1-Binaphthyl
2,2-Binaphthyl	p-Quaterphenyl
PPO	PPD
PBD	BBO
BBD	α-NPO
α-NPD	β-NPD
POPOP	DimethylPOPOP
4,5-Diphenylimidazul	Triphenylene
Pyrene	1,3,6,8-Tetraphenylpyrene
Perylene	Diphenylstilbene
BBOT	Esculin
Anthraquinone	Phenanthraquinone

After the first fluorescence spectrum was obtained, the excitation wavelength was increased by 3nm and another one-dimensional fluorescence spectrum with the same emission range was scanned. This cycle was continued until the fluorescence spectrum at an excitation wavelength of 454nm was taken. At this point, 64 one-dimensional fluorescence spectra with increasing excitation wavelength had been stored in the PE 7500 data station. These 64 individual fluorescence spectra were then incorporated into a single set of 2-dimensional fluorescence data set consisting of 4096 (64 points by 64 points) data points per sample.

C. DATA TRANSFER PROCESS

These two-dimensional fluorescence data files were then transferred from the Perkin-Elmer data station to the VAX system through a RSTS/E operating system running on a PDP-11/23 (Digital Equipment Corporation). All the software developed for these operations is listed in the appendixes. The development of the data compression techniques, and building of the standard spectral library for the qualitative analysis of the fluorescence spectra can now be described.

D. CREATION OF REFERENCE LIBRARY

For the qualitative identification of three-dimensional fluorescence spectra, it is necessary to develop a computer assisted library search system. Furthermore, it is important to make such a library search system as efficient as possible. One method is to compress the spectral data in the reference library so that not only the storage space, but more importantly, the search time will be minimized. The data compression techniques developed in this work uses the Hadamard Transform and clipping method which are described in the "Theory" section. A combination of unique data compression and library search system techniques produces an efficient means of handling three-dimensional fluorescence spectra. The following describes that library search system as well as the data compression techniques.

As shown in the Figure 3, all three dimensional fluorescence spectra underwent several signal processing steps before they were stored in a reference library. First, they were Hadamard transformed (HT), then 75% of the area representing the higher sequencies was eliminated and the remainder stored in the "Filtered Spectra Library". This process is valid because, as we shall see, the majority of the characteristic spectral information occurs at lower sequencies. The final step involves the "clipping" technique described previously. The resultant binary

pattern is stored in the "Search Library".

As in Figure 3, an unknown spectrum goes through the same procedures as the reference spectra. The filtered HT data representation and the clipped HT representation are used for comparison with those of reference spectra.

E. HADAMARD TRANSFORM ALGORITHM

Algorithms of all the software developed in the research are given in the appendix. However, it is important to explain Hadamard Transformation algorithm in detail at this point.

A Two dimensional Hadamard transform is done in the following manner. First, the column transformation is performed, then the row transform is done. It is important to set a flag that indicates column/row transformation. Then, based on the dimension of the data set, the number of iterations necessary for the transform is calculated. The rest of the procedure is listed below.

1. Start the procedure with the first iteration.
2. For a given iteration, the number of partitions, the total number of the data points involved within one partition, and the number of additions/subtractions required within the partition are calculated.
3. Set the default calculation to Addition.
4. Determine which two data points should be added and

subtracted then perform the actual addition and subtraction.

5. Repeat step 4 until all the pairs of data points within a partition are added and subtracted.
6. Repeat steps 4 and 5 for all the partitions within a given iteration.
7. Repeat all steps from 2 through 6 for the next iteration.
8. After all the steps are repeated for the number of iterations, divide the results by the dimension of the data set.
9. If the column transform is completed then reset the column/row flag to row transformation. Repeat the above steps. When the row transform is completed, then two dimensional Hadamard Transform is completed.

F. LIBRARY SEARCH SCHEME

An explanation of the library search scheme follows. As shown in Figures 6 and 7, a comparison between an unknown and the reference library takes place in two steps. The first step is called the "prefilter". This is to eliminate the most unlikely candidates from the reference spectra. It selects only those reference spectra which are most likely to be a proper identification of the unknown. Since this

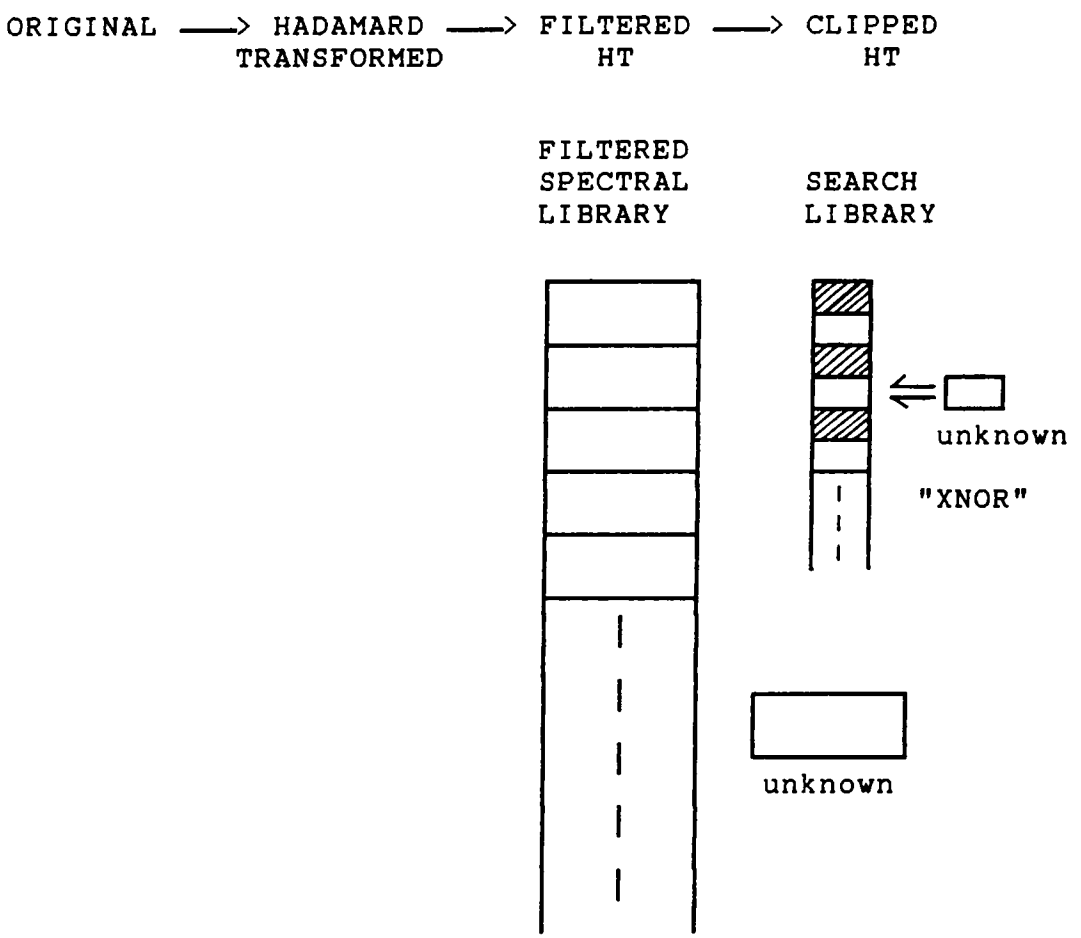


Figure 6. Library Scheme with a Prefilter process

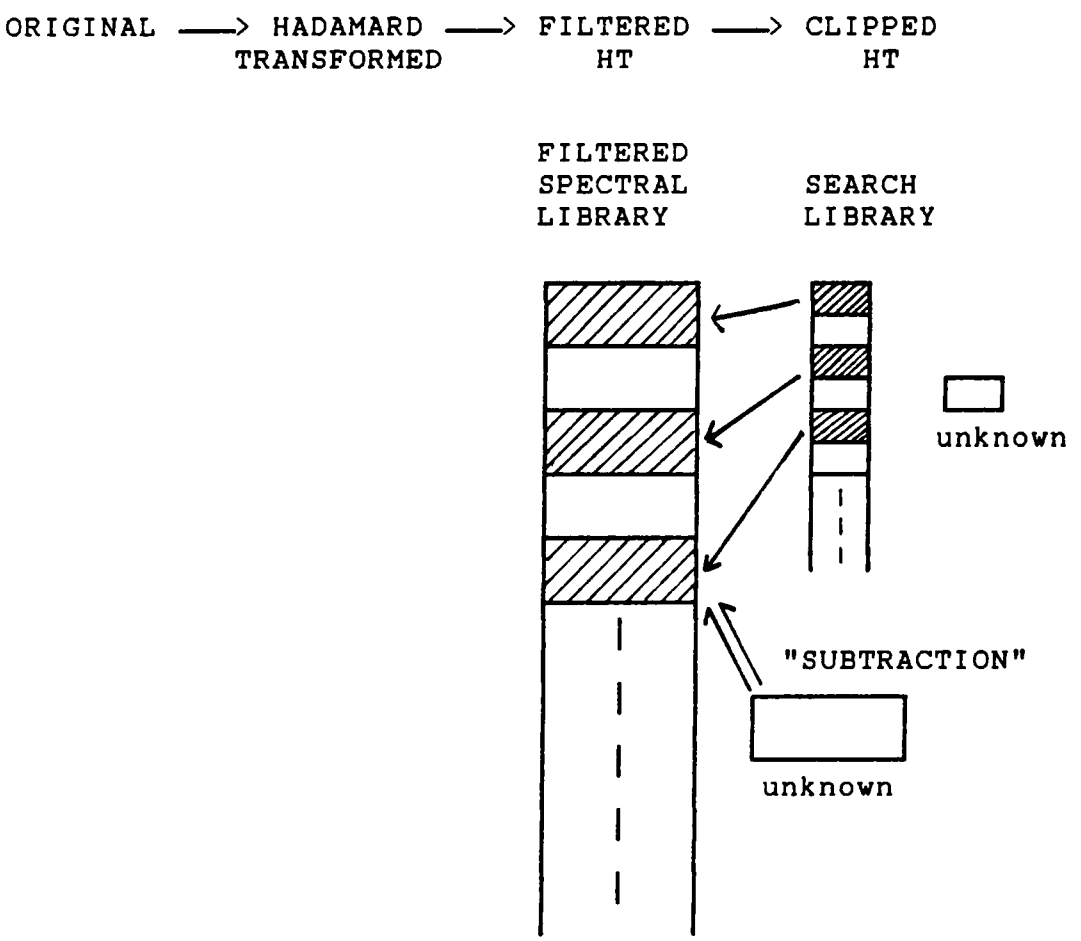


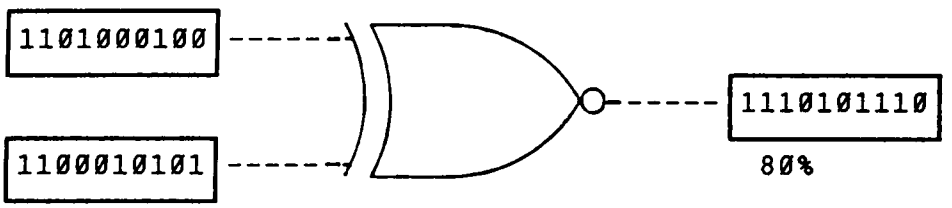
Figure 7. Library Scheme with a Second Stage Comparison

step involves a comparison with the entire library, the procedure has to be quick, yet accurate. Because the clipped HT form is a binary pattern, comparisons employing it can be very fast. The method utilized is called XNOR logic (Figure 6). This XNOR logic Table is shown in Table 3. After this comparison, the values in the output are totaled and the percentage of correct matches is calculated. As shown in Figure 8, if the reference and unknown spectra are identical, then the comparison result should yield a 100% perfect match rate. On the other hand, if two spectra are very different (i.e., two spectra have no relation to each other), then the result should be statistically around a 50% match rate. The reference spectra are placed in a decreasing match order sequence. Those with the highest match rates are considered as possible candidates for the unknown.

The next step involves the use of reference spectra in the "Filtered Spectral Library", which is made up of the 25% compressed HT representation (Figure 7). The comparison method utilized was simple subtraction between the reference and unknown spectra. A small difference between the unknown and a reference spectra identifies similar spectra. This comparison is more computational intensive than the XNOR method. However, because of the "prefilter", only a few selected reference spectra need to be examined at this step in the identification of an unknown.

Table 3. XNOR logic table

ref	unknown	output
Ø	Ø	1(correct match)
Ø	1	Ø(incorrect match)
1	Ø	Ø
1	1	1



	percentile
perfect hit(identical pair)	100%
example(above)	80%
statistically unrelated (very different pair)	50%

Figure 8. An example of comparison procedure at a prefilter step

V. RESULTS AND DISCUSSION

RESULTS OF THE SEARCH

The first study mainly concentrated on the clipped data base set which was used for the preliminary comparison stage known as a "prefilter". Statistics were acquired assessing the degree of compression that could be imposed on the system and yet still identify unknowns correctly.

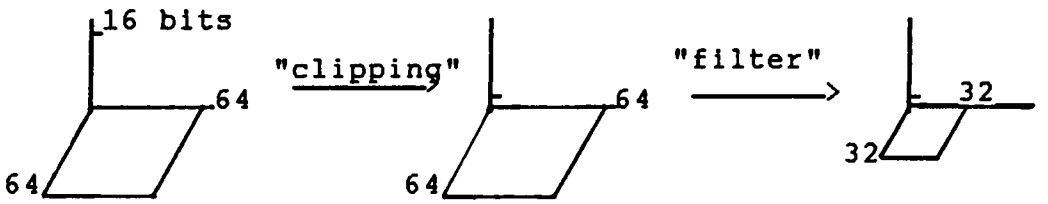
A. UNIQUENESS OF THE SYSTEM

As discussed in the earlier section, the first stage comparison utilized the clipped data base and was done by using the XNOR logical method. At the end of each comparison, the percentage of the match rate was calculated.

If the data compression technique/search method is to be useful, three criteria must be satisfied. First, it should correctly identify the unknown with a 100% match rate at the first comparison step. Second, to assure the uniqueness of that spectrum, the correct reference spectrum should be the only one that gives 100% match. Finally, the search results with the rest of the reference spectra should lie around 50%.

This search technique was tested using two types of clipped data base sets. With clipped data, one needs to

store 1 bit per data point instead of the 16 bits per data point required by integer math storage of amplitude data. The first set used the entire 64 by 64 matrix of clipped data points for each library member of the data base. The second set comprised a 32 by 32 data point matrix occurring at the lowest sequency region. These two data bases were examined to observe any improvement or deterioration of the search results.



64 x 64 x 16bits	64 x 64 x 1bit	32 x 32 x 1bit
=4096*16 = 4k*16	= 4096*1 = 4k*1	=1024*1 = 1k*1
100% data base	6.25% data base or 100% clipped data base	1.56% data base or 25% clipped data base

As shown in the above figure, the two clipped data bases are named the "6.25% data base" and "1.56% data base". These values demonstrate the amount of compression. This awkward non-integer terminology can be simplified. This is accomplished by renaming the "6.25% data base", which consists of the entire 64 by 64 matrix of clipped data points, as the "100% clipped data base". The second set which consists of the 32 by 32 clipped data point matrix thus becomes the "25% clipped data base".

The first tests were performed using one of the spectra from the reference library as an unknown. This library consisted of 56 PNA compounds. Table 4 shows the summary of the results for each of the 56 compounds using the "100% clipped data base". All of the 56 PNA compounds correctly identify themselves with a 100% match rate. Also, for all the fifty-six cases, the correctly identified compound was the only one which produced a 100% match rate. This indicates the uniqueness of that spectrum after such drastic compression of the data.

The last column in Table 4, "Average Search Results", indicates the average match rate of the 2nd through 56th hit compounds. Table 5 shows the total average of the "Average Search Results" which is 54.4% with standard deviation of 4.3%. The statistical normal distribution of this results are shown in the Figure 9. If the two samples compared are randomly related, the match rate should lie around 50%. The results obtained indicate that the binary patterns obtained by clipping are unique signatures for the compounds.

The "Difference of 1st & 2nd " column in Table 4 shows the difference of the match rate between a correctly identified compound and the compound which has the second highest match rate (denoted as the 2nd hit). The larger this difference, the more unique the 1st hit compound is from the rest of spectra in the reference library. Table 5 indicates the average difference of the 1st and 2nd hit is 31.0%.

Table 4. Results of the library search using a 100% clipped data base

NO.	Names of 1st hit	Diff. of 1st & 2nd (%)	Ave. Search Results (%)
1	Anthranilic acid	38.3	54.0
2	Salicilic acid	28.9	57.5
3	1,1-diphenylethylene	31.5	54.2
4	4-biphenylphenyl ether	22.8	55.0
5	4-methylbiphenyl	29.2	53.4
6	4-vinylbiphenyl	24.2	54.3
7	Triphenylamine	34.1	55.6
8	Indole	22.8	54.7
9	Quinoline	31.1	55.5
10	Azulene	34.1	53.9
11	Naphthalene	26.2	56.1
12	1-methylnaphthalene	14.6	55.5
13	2-methylnaphthalene	28.1	55.5
14	2,3-dimethylnaphthalene	26.2	55.1
15	2,6-dimethylnaphthalene	38.5	53.5
16	1-naphthol	35.8	52.1
17	2-naphthol	31.7	53.9
18	1-phenylnaphthalene	14.6	55.6
19	2-phenylnaphthalene	25.8	55.2
20	Fluorene	43.7	52.1
21	Acridine	38.3	54.9
22	Anthracene	43.0	52.0
23	1-aminoanthracene	32.9	55.0
24	2-aminoanthracene	30.1	55.8
25	9,10-dichloroanthracene	36.4	53.3
26	9-methylanthracene	16.5	52.4
27	9-phenylanthracene	16.5	53.6
28	9,10-diphenylanthracene	23.9	54.4
29	9-vinyllanthracene	23.9	54.3
30	2-methylanthracene	30.6	54.3
31	Tetracene	38.6	52.6
32	Phenanthrene	38.2	53.4
33	Chrysene	36.3	54.5
34	1,1-binaphthyl	37.5	54.8
35	2,2-binaphthyl	33.5	55.0
36	p-quaterphenyl	34.2	54.2
37	PPO	33.9	52.6
38	PPD	24.2	53.4
39	PBD	31.7	54.6
40	BBO	30.6	55.4
41	BBD	34.1	54.4

Table 4. Results of the library search using a 100% clipped data base (Cont'd)

NO.	Names of 1st hit	Diff. of 1st & 2nd (%)	Ave. Search Results (%)
42	α-NPO	32.4	53.8
43	α-NPD	40.4	52.2
44	β-NPD	35.3	53.8
45	POPOP	27.2	54.7
46	DimethylPOPOP	38.7	52.5
47	4,5-diphenylimidazule	34.2	57.5
48	Triphenylene	32.1	53.5
49	Pyrene	29.6	56.6
50	1,3,6,8-tetraphenylpyrene	36.5	52.9
51	Perylene	39.3	53.2
52	Diphenylstilbene	25.8	54.9
53	BBOT	24.6	53.5
54	Esculin	30.1	55.6
55	Anthraquinone	28.9	57.6
56	Phenanthraquinone	32.1	55.1

Table 5. Average results of the library search using a 100% clipped data base

Ave. diff. of 1st & 2nd hit	Average of "Ave. search results"	Average Standard deviation
31.0%	54.4%	4.3%

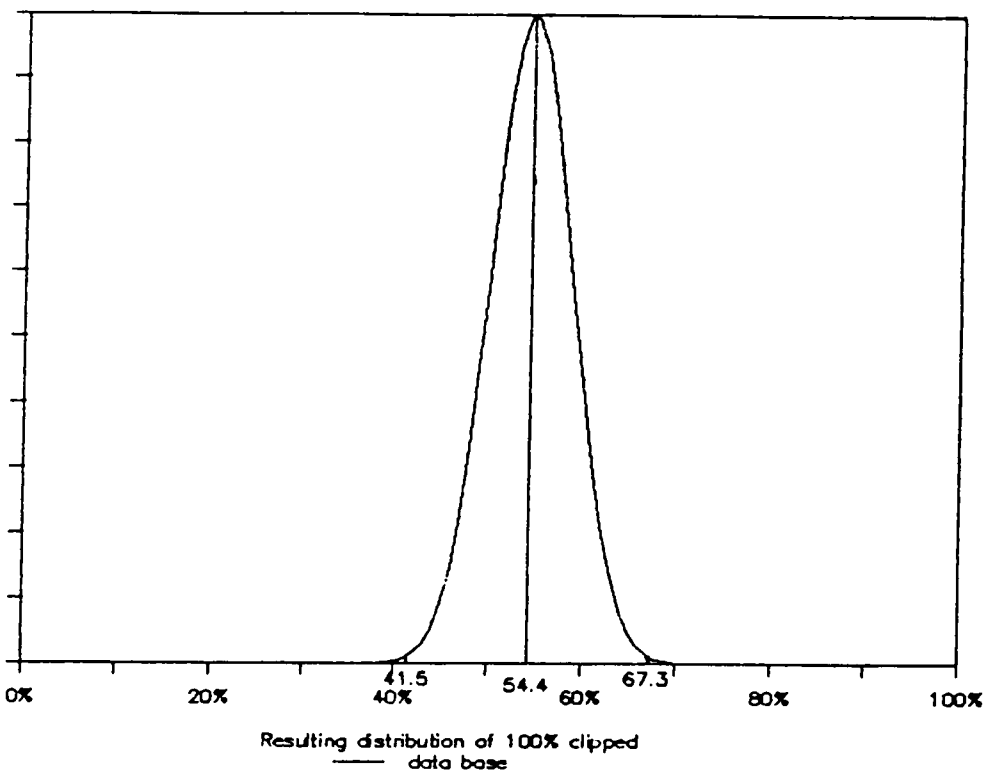


Figure 9. The statistical normal distribution of the search results using a 100% clipped data base.

The total average match rates of the spectra which ranked from 2nd to 56th was 54.4%. The average difference between 1st and 2nd hits was 31.0%. The average match rate for the 2nd hit then would be 69.0%. Since this value is higher than the three-standard-deviations boundary (which is from 41.5% to 67.3% (54.4±4.3%)), it is interesting to investigate the types of compounds which came in 2nd.

Table 6 shows the results for some derivatives of anthracene and naphthalene compounds as unknowns, and the names of those compounds which ranked 2nd. It is important to note that those ranking 2nd are spectrally similar to the unknowns of interest. The fluorescence spectra of many structurally related derivatives are similar. It is not surprising then that those compounds with a ranking high hit positions are structurally similar to the unknown.

One notes that even after a considerable amount of compression, the binary pattern still contains characteristic information about that spectrum.

Table 7 through 8 shows the results in similar study using the "25% clipped data base. Table 8 shows the total average of the "Average Search Results" from Table 7. This is 52.7% with a standard deviation of 4.7%. The three-standard-deviations boundary in this case is from 38.6% to 66.8% (52.7±4.7%). The normal distribution is shown in the Figure 10. These latter values indicate that the majority

Table 6. Search results of derivatives of naphthalene and anthracene compounds showing 2nd place hits using the 100% clipped data base.

Name of 1st hit	Names of 2nd hit
Naphthalene	2,3-dimethylnaphthalene
1-methylnaphthalene	1-phenylnaphthalene
2-methylnaphthalene	2,3-dimethylnaphthalene
2,3-dimethylnaphthalene	Naphthalene
2,6-dimethylnaphthalene	1-naphthol
1-naphthol	2,3-dimethylnaphthalene
1-phenylnaphthalene	1-methylnaphthalene
2-phenylnaphthalene	Diphenylstilbene
1-aminoanthracene	2-aminoanthracene
9,10-dichloroanthracene	9,10-diphenylanthracene
9-methylanthracene	9-phenylanthracene
9-phenylanthracene	9-methylanthracene
9,10-diphenylanthracene	9-vinylanthracene
9-vinylanthracene	9,10-diphenylanthracene
2-methylanthracene	

Table 7. Results of the library search using a 25% clipped data base

NO.	Names of 1st hit	Diff. of 1st & 2nd (%)	Ave. Search Results (%)
1	Anthranilic acid	40.9	51.8
2	Salicilic acid	38.7	54.2
3	1,1-diphenylethylene	35.0	52.7
4	4-biphenylphenyl ether	20.6	53.6
5	4-methylbiphenyl	36.2	51.8
6	4-vinylbiphenyl	17.9	51.7
7	Triphenylamine	34.7	53.5
8	Indole	20.6	52.9
9	Quinoline	32.5	53.5
10	Azulene	33.5	52.6
11	Naphthalene	27.4	53.8
12	1-methylnaphthalene	22.7	53.8
13	2-methylnaphthalene	31.2	53.7
14	2,3-dimethylnaphthalene	27.5	53.2
15	2,6-dimethylnaphthalene	31.4	52.3
16	1-naphthol	23.4	52.1
17	2-naphthol	23.1	52.5
18	1-phenylnaphthalene	22.7	54.0
19	2-phenylnaphthalene	27.8	54.2
20	Fluorene	35.0	51.5
21	Acridine	37.6	52.7
22	Anthracene	42.1	52.2
23	1-aminoanthracene	34.4	52.2
24	2-aminoanthracene	30.3	53.1
25	9,10-dichloroanthracene	35.1	51.7
26	9-methylanthracene	14.3	51.5
27	9-phenylanthracene	14.3	52.4
28	9,10-diphenylanthracene	21.5	53.0
29	9-vinyllanthracene	21.5	52.8
30	2-methylanthracene	25.7	52.1
31	Tetracene	41.6	51.5
32	Phenanthrene	34.7	53.0
33	Chrysene	32.5	53.8
34	1,1-binaphthyl	36.7	52.9
35	2,2-binaphthyl	26.5	53.7
36	p-quaterphenyl	37.4	50.9
37	PPO	30.0	50.8
38	PPD	17.9	52.4
39	PBD	18.8	53.0
40	BBO	27.4	53.9
41	BBD	32.5	52.7

Table 7. Results of the library search using a 25% clipped data base (Cont'd)

NO.	Names of 1st hit	Diff. of 1st & 2nd (%)	Ave. Search Results (%)
42	α-NPO	32.9	51.5
43	α-NPD	34.4	51.3
44	β-NPD	26.5	53.1
45	POPOP	25.4	52.8
46	DimethylPOPOP	33.3	50.8
47	4,5-diphenylimidazule	37.4	53.4
48	Triphenylene	18.8	53.5
49	Pyrene	37.5	53.1
50	1,3,6,8-tetraphenylpyrene	31.4	52.1
51	Perylene	43.7	51.8
52	Diphenylstilbene	25.4	53.1
53	BBOT	22.7	52.5
54	Esculin	30.3	52.5
55	Anthraquinone	33.4	55.1
56	Phenanthraquinone	33.4	54.3

Table 8. Average results of the library search using a 25% clipped data base

Ave. diff. of 1st & 2nd hit	Average of "Ave. search results"	Average Standard deviation
29.7%	52.7%	4.7%

Table 9. Search results of derivatives of naphthalene and anthracene compounds showing 2nd place hits using the 25% clipped data base.

Name of 1st hit	Names of 2nd hit
Naphthalene	4-vinylbiphenyl
1-methylnaphthalene	1-phenylnaphthalene
2-methylnaphthalene	Naphthalene
2,3-dimethylnaphthalene	Naphthalene
2,6-dimethylnaphthalene	1-naphthol
1-naphthol	1-methylnaphthalene
1-phenylnaphthalene	1-methylnaphthalene
2-phenylnaphthalene	
1-aminoanthracene	2-aminoanthracene
9,10-dichloroanthracene	9,10-diphenylanthracene
9-methylanthracene	9-phenylanthracene
9-phenylanthracene	9-methylanthracene
9,10-diphenylanthracene	9-vinylanthracene
9-vinylanthracene	9,10-diphenylanthracene
2-methylanthracene	POPOP

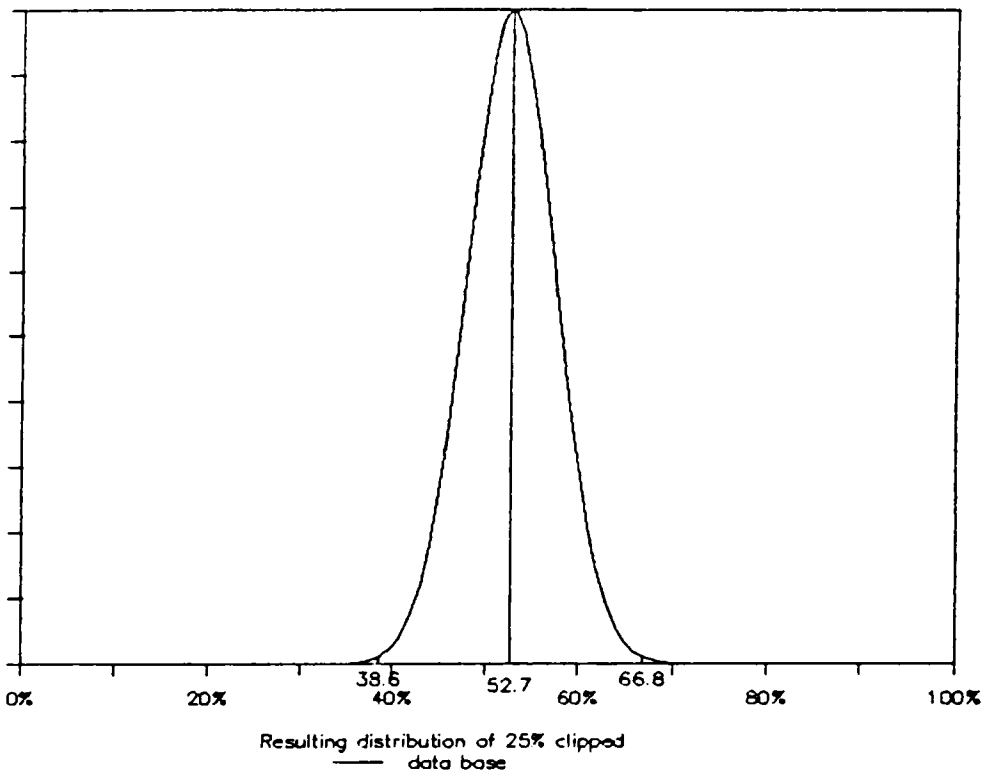


Figure 1Ø. The statistical normal distribution of the search results using a 25% clipped data base.

of the match rates generated by the comparison of an unknown and the incorrect reference spectra will be between 38.6% and 66.8% (i.e., within the three standard deviation range).

A typical result is shown in Figure 11a and Figure 11b, where diphenylstilbene was selected from the reference library and used as an unknown. This figure shows the distributions of the match rates for the spectra. The x-axis indicates the match rates between the unknown and the reference spectra. The match rates are expressed in percentile. The y-axis indicates the number of reference spectra meeting this match requirement. The one spectrum with a 100% match rate is from the comparison of the diphenylstilbene with itself. For the "100% clipped data base", the results of the comparison of diphenylstilbene with the rest of the reference spectra lie from a 49.6% match rate to a 74.2% match rate. For the "25% clipped data base", the values are from 44.8% to 74.6%. For both the 100% clipped and 25% clipped data bases, the majority of the reference spectra produced match rates around 50% with a specific average match rate of 54.9% and 53.1% respectively. This indicates that most of the reference spectra have binary patterns statistically unrelated and different from that of diphenylstilbene. The figures for the rest of the spectra are in the appendix.

These results are very comparable to those obtained

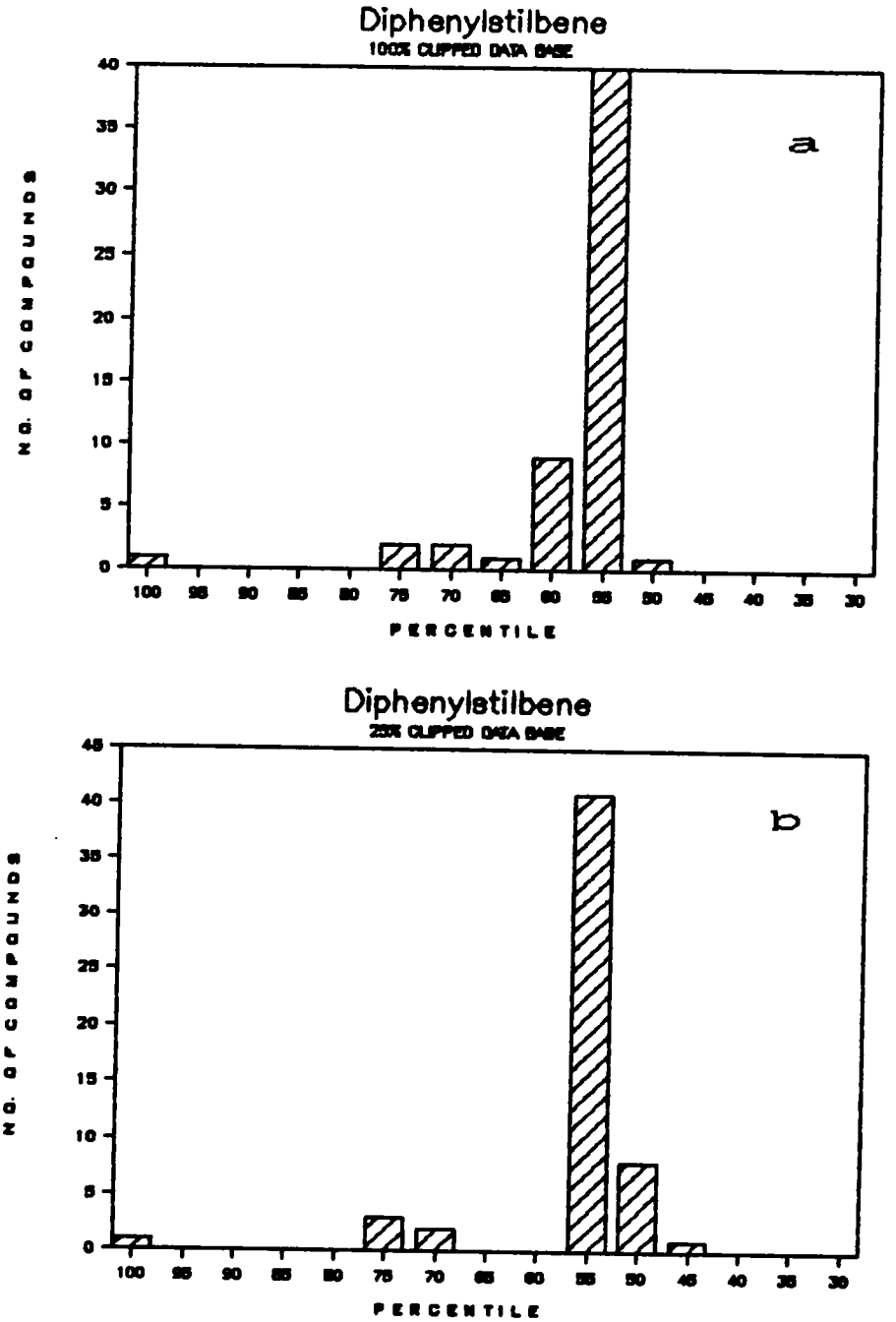


Figure 11. the search results of diphenylstilbene using
(a). a 100% clipped data base,
(b). a 25% clipped data base.

from the 100% clipped data base. One can conclude that the secondary compression by 75% does not adversely affect search performance.

B. USE OF REAL "UNKNOWNS"

Next a study was initiated using 21 "real" unknown spectra where the compounds' spectra are to be found in the reference library, but the unknown data were measured at totally different times. Our library search system should satisfy the factors mentioned earlier. Because these spectra, the unknown and reference, have random noise, it is impossible to obtain a 100% match rate. However, a reliable system should give a high match rate. Second, search results with the rest of the reference library should again lie around 50%. Finally, if the 1st hit and 2nd hit rate are close, then the structural relationship between compounds generating these spectra should be investigated.

The search technique was tested again using the two types of clipped data base sets, which are the "100%" and "25%" clipped data matrices.

The Table 10 shows the summary of the results for 21 "real unknowns" using the 25% data base. Out of 21 unknowns, 18 were correctly identified, two additional unknowns ranked as the second hit, while one unknown ranked 3rd. The search system can apparently withstand the normal

Table 10. Search results for 21 real unknown compounds using a 25% clipped data base

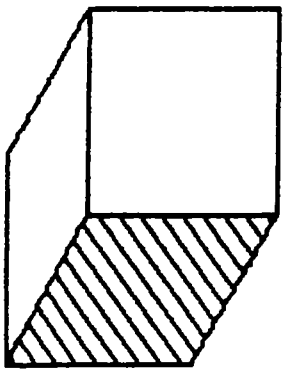
No.	Names of unknowns	Match Rates (%)	Rank	Ave. Search Results (%)
11	Naphthalene	77.0	1	51.8
12	1-methylnaphthalene	71.9	1	50.5
16	1-naphthol	74.9	1	51.7
22	Anthracene	97.5	1	52.0
23	1-aminoanthracene	92.1	1	52.4
25	9,10-dichloroanthracene	85.6	1	51.8
26	9-methylanthracene	83.1	1	50.4
29	9-vinyllanthracene	87.0	1	53.4
31	Tetracene	96.2	1	50.4
32	Phenanthrene	83.8	1	58.9
33	Chrysene	70.1	1	52.4
34	1,1-binaphthyl	88.1	1	52.7
37	PPO	72.8	1	52.4
45	POPOP	87.2	1	53.1
49	Pyrene	96.1	1	52.3
50	1,3,6,8-tetraphenylpyrene	85.9	1	52.6
51	Perylene	70.6	1	51.7
53	BBOT	82.4	1	52.6
27	9-phenylanthracene	79.9	2	51.8
38	PPD	71.9	2	51.5
18	1-phenylnaphthalene	69.4	3	51.9

Average of "Ave. Search Results" = 52.1%
Standard Deviation = 4.1%

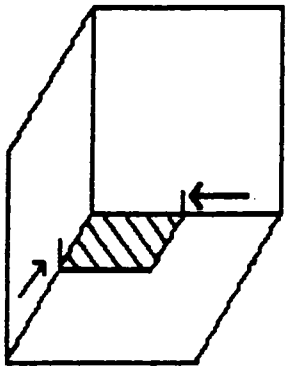
instrumental and chemical noise injected into laboratory results.

In the previous study, the majority of the values of the match rates generated by comparison of an unknown with an incorrect reference spectra were between 38.6% and 66.8% range. These results also show that all the match rates for correctly identified spectra are higher than 66.8%. Those compounds which were not identified by the first rank hit had the correct hit at a somewhat lower rank, but which still showed a match rate greater than 66.8%. This demonstrates that the compression did not result in the loss of significant amounts of information.

The above results were compared with those obtained using the 100% data base (Figure 12). Figure 13 shows the results. If only the first hit is accessible for the identifications of an unknown, only 66.7% were correctly identified using the 100% data base. With the 25% clipped data base (Figure 12) 85.7% were correctly identified. If the first five hits were included, all unknowns were correctly identified with 25% clipped data base, while with 100% clipped data base, 90.5% were correctly identified. These results indicate that great improvement in search performance was obtained by using the smaller data base! The 25% data base includes only the lower sequency regions of the 100% data base. The higher sequency regions tend to include random noise and information about subtle peak shape



100% data base



25% data base

Figure 12. Representation of 100% and 25% data base

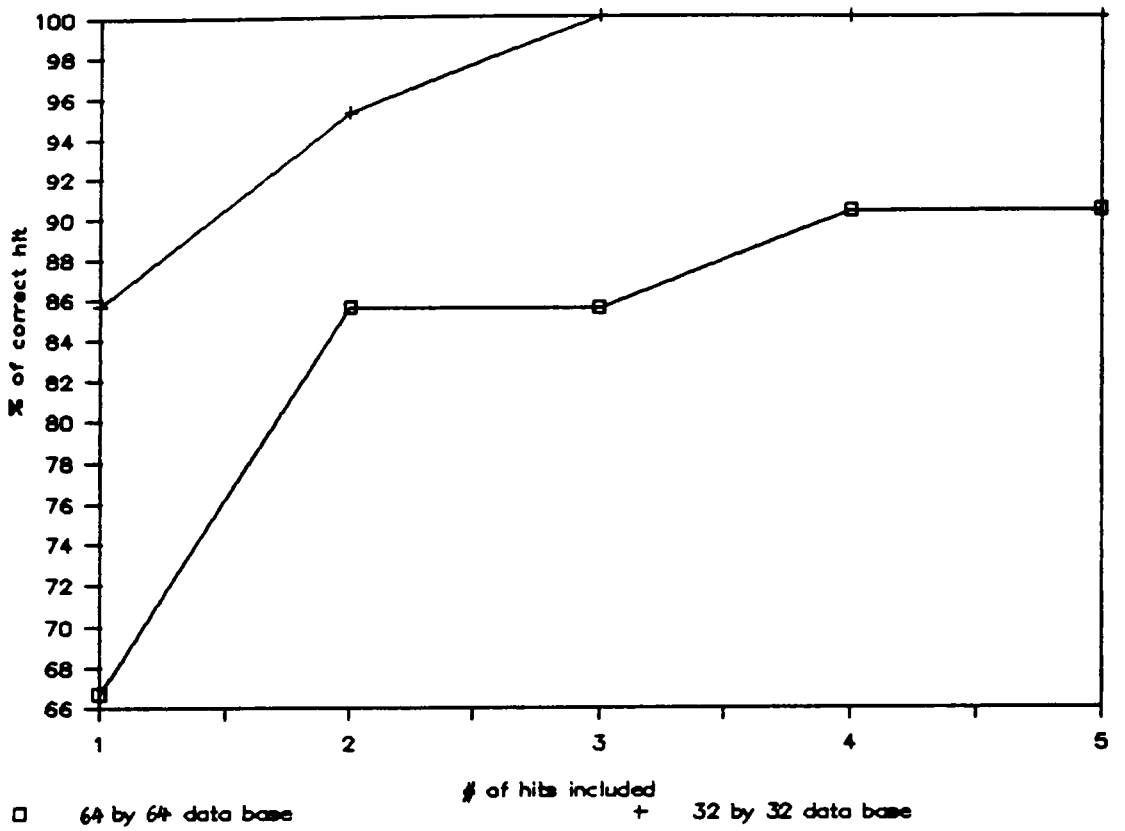


Figure 13. Comparison of performance between 100% and 25% clipped data base

that is easily altered by experimental conditions. The characteristic information regarding the fundamental spectral shape tend to lie in the lower sequency regions. The results above prove that less is more.

As shown in the Table 10, average search results were calculated for each unknown search. Also the total average search results for all unknowns calculated (Table 10) which is 52.1% with a standard deviation of 4.1%. Even using the real unknowns the average results lie around 50%. This shows the search method does not produce a high value unless the spectral shapes involved are similar.

This brings up another important factor of this search results; the types of spectra. Table 9 shows the names of 1st and 2nd ranked compounds for the series studied. The compounds in the data bases were derivatives of naphthalene and anthracene and related compounds. Spectral relationships between the 1st and 2nd ranked compounds show that they are structurally similar to one another. The library search system gives higher match rates to these reference spectra with similar spectral shapes.

Tables 11 through 14 show the complete list of unknowns with their companion matches. Hits up to the fourth rank are shown. These lists indicate that not only the second ranked compounds but also the third and fourth ranked compounds are not always, but are usually spectrally and structurally similar to one another.

Table 11. Search results of naphthalene, 1-methylnaphthalene, 1-naphthol and 1-aminoanthracene

Naphthalene

Rank	Percentile	Spectra
1	77.Ø	Naphthalene
2	74.5	4-vinylbiphenyl
3	67.4	PPD
4	66.4	2,3-dimethylnaphthalene

1-methylnaphthalene

Rank	Percentile	Spectra
1	71.9	1-methylnaphthalene
2	65.7	2-methylnaphthalene
3	63.5	1-naphthol
4	6Ø.Ø	2,6-dimethylnaphthalene

1-naphthol

Rank	Percentile	Spectra
1	74.9	1-naphthol
2	7Ø.4	1-methylnaphthalene
3	61.9	2-methylnaphthalene
4	61.6	Chrysene

1-aminoanthracene

Rank	Percentile	Spectra
1	92.1	1-aminoanthracene
2	66.7	2-aminoanthracene

Table 12. Search results of 9-methylanthracene 9-vinylanthracene and 9,10-dichloroanthracene

9-methylanthracene

Rank	Percentile	Spectra
1	83.1	9-methylanthracene
2	70.8	9-phenylanthracene
3	61.1	Acridine

9-vinylanthracene

Rank	Percentile	Spectra
1	87.0	9-vinylanthracene
2	71.6	BBOT
3	70.4	9,10-diphenylanthracene

9,10-dichloroanthracene

Rank	Percentile	Spectra
1	85.6	9,10-dichloroanthracene
2	64.1	1,3,6,8-tetraphenylpyrene
3	62.3	9,10-diphenylanthracene
4	61.5	BBOT

Table 13. Search results of 1,1-binaphthyl chrysene, PPD, and POPOP

1,1-binaphthyl

Rank	Percentile	Spectra
1	88.1	1,1-binaphthyl
2	69.5	1-naphthol
3	68.2	1-phenylnaphthalene
4	62.7	1-methylnaphthalene

Chrysene

Rank	Percentile	Spectra
1	70.1	chrysene
2	63.4	2,2-binaphthyl
3	61.3	PBD
4	61.1	β-NPD

PPO

Rank	Percentile	Spectra
1	72.8	PPO
2	68.3	β-NPD
3	62.0	Anthraquinone

POPOP

Rank	Percentile	Spectra
1	87.2	POPOP
2	75.2	2-phenylnaphthalene
3	75.1	Diphenylstilbene
4	74.2	2-methylanthracene

Table 14. Search results of 1,3,6,8-tetraphenylpyrene, BBOT, pyrene and phenanthrene

1,3,6,8-tetraphenylpyrene

Rank	Percentile	Spectra
1	85.9	1,3,6,8-tetraphenylpyrene
2	67.6	9,10-diphenylanthracene
3	64.8	BBOT
4	62.2	9,10-dichloroanthracene

BBOT

Rank	Percentile	Spectra
1	82.4	BBOT
2	69.1	9-vinylanthracene
3	65.8	1,3,6,8-tetraphenylpyrene
4	64.6	9,10-diphenylanthracene

Pyrene

Rank	Percentile	Spectra
1	95.9	Pyrene
2	61.2	Anthraquinone
3	60.5	Quinoline

Phenanthrene

Rank	Percentile	Spectra
1	83.8	Phenanthrene
2	69.8	Triphenylamine
3	63.9	4,5-diphenylimidazule
4	63.0	1,1-diphenylethylene

As mentioned above, three unknowns were not correctly identified by the highest ranking hit. Table 15 shows the list of these search results. When 9-phenylanthracene was used as an unknown, the search produced the highest match with 9-methylanthracene, and 9-phenylanthracene came in second. Both compounds are substitutes of anthracene at the same position. They are spectrally and structurally closely related. In the case of 1-phenylnaphthalene, the correct match came in third, after 1-methylnaphthalene and 1-naphthol. In the case of 1-phenylnaphthalene both 1-methylnaphthalene and 1-naphthol preceded the correct identification. This was followed by 2,6-dimethylnaphthalene. These sets involve similarly substituted naphthalene and hence are spectrally very similar to each other.

The compression/searching technique developed is based on the shapes of the spectra. But the fluorescence spectrum of a compound is related to its structure. Compounds with very similar structure often yield similar spectra. The compression/searching technique is affected by the structure of compounds only through the similarity of their spectra. There is, however, an indirect connection between high hit rank and structural similarity.

Table 15. Search results of 9-phenylanthracene, PPD, 1-phenylnaphthalene

9-phenylanthracene

Rank	Percentile	Spectra
1	80.3	9-methylanthracene
2	79.9	9-phenylanthracene
3	63.2	Acridine
4	58.9	Dimethyl POPOP

PPD

Rank	Percentile	Spectra
1	89.2	4-vinylbiphenyl
2	71.9	PPD
3	70.1	Naphthalene
4	61.4	2-methylnaphthalene

1-phenylnaphthalene

Rank	Percentile	Spectra
1	77.1	1-methylnaphthalene
2	76.1	1-naphthol
3	69.4	1-phenylnaphthalene
4	63.2	2,6-dimethylnaphthalene

C. APPLICATION OF A SECOND STAGE COMPARISON

After the prefilter process is done, a second stage comparison is employed to resolve ambiguities within the hit set derived from application of the prefilter. This second stage comparison involves the subtraction method discussed earlier.

It is important to confirm that the 25% filtered spectra which are used for the second stage comparison retain characteristic spectral information. To demonstrate this, a comparison of two methods are necessary, the prefilter vs. comparison of filtered but not clipped data sets. First, those unknowns which were correctly identified in the first stage comparison (prefilter method) were taken and subjected to a second stage comparison using the 25% filtered reference spectra. Tables 16 through 19 shows the results of both the prefilter process and the second stage comparison for the above unknowns. With the exception of one compound, the results from both methods agreed. It is not surprising that the 25% filtered spectra retain characteristic information about each spectra.

Next, it is important to demonstrate the searching power of this second stage comparison for resolving conflicts that occur after application of the prefilter. To show this point, again a comparison of the prefilter and the subtraction methods is necessary. Those unknowns which were

Table 16. Search results of unknowns from a prefilter process and a second stage comparison

Prefilter:			Second stage Comparison:		
NAPHTHALENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	77.0	Naphthalene	[1]	826006	Naphthalene
2	74.5	4-vinylbiphenyl	2	1428394	4-vinylbiphenyl
3	67.4	PPD	3	1626349	PPD
4	66.4	2,3-dimethyl-naphthalene	4	1631406	2,3-dimethyl-naphthalene
1-METHYLNAPHTHALENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	71.9	1-Me-naphthalene	[1]	712310	1-Me-naphthalene
2	65.7	2-Me-naphthalene	2	1573688	2-Me-naphthalene
3	63.5	1-naphthol	3	1639775	2,6-dimethyl-naphthalene
4	60.0	2,6-dimethyl-naphthalene	4	2955738	1-naphthol
1-NAPHTHOL					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	74.9	1-naphthol	[1]	1630724	1-naphthol
2	70.4	1-Me-naphthalene	2	2747580	1-Me-naphthalene
3	61.9	2-Me-naphthalene	3	3519175	β-NPD
4	61.6	Chrysene	4	3556428	2-Me-naphthalene
1,1-BINAPHTHYL					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	88.1	1,1-binaphthyl	[1]	707610	1,1-binaphthyl
2	69.5	1-naphthol	2	2033241	1-Me-naphthalene
3	68.2	1-phenylnaphthalene	3	2079369	1-phenyl-naphthalene
4	62.7	1-Me-naphthalene	4	2528787	1-naphthol

Table 17. Search results of unknowns from a prefilter process and a second stage comparison

Prefilter:			Second stage Comparison:		
9-METHYLANTHRACENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	83.1	9-Me-anthracene	[1]	487946	9-Me-anthracene
2	70.8	9-phenylanthracene	2	1277092	9-phenyl-anthracene
3	61.1	Acridine	3	1856858	Anthranilic acid
4			4	2484251	9,10-dichloro-anthracene
9-VINYLANTHRACENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	87.0	9-vinylanthracene	[1]	683924	9-vinylanthracene
2	71.6	BBOT	2	2048752	BBOT
3	70.4	9,10-diphenyl-anthracene	3	2075519	9-Me-anthracene
4			4	2085545	9,10-diphenyl anthracene
9-DICHLOROANTHRACENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	85.6	9,10-dichloro-anthracene	[1]	990434	9,10-dichloro anthracene
2	64.1	1,3,6,8-tetraphenyl pyrene	2	2305851	9-phenyl anthracene
3	62.3	9.10-diphenyl-anthracene	3	2336688	BBOT
4	61.5	BBOT	4	2452046	9,10-diphenyl anthracene

Table 18. Search results of unknowns from a prefilter process and a second stage comparison

Prefilter:			Second stage Comparison:		
1-AMINOANTHRACENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	92.1	1-aminoanthracene	[1]	213975	1aminoanthracene
2	66.7	2-aminoanthracene	2	1325882	4-biphenyl-phenyl ether
3			3	1538823	2aminoanthracene
4			4	3044215	BBOT
CHRYSENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	70.1	chrysene	[1]	1830711	chrysene
2	63.4	2,2-binaphthyl	2	3148459	PBD
3	61.3	PBD	3	3223166	β-NPD
4	61.1	β-NPD	4	3299091	2,2-binaphthyl
PPO					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	72.8	PPO	[1]	717323	PPO
2	68.3	β-NPD	2	1380767	2,2-binaphthyl
3	62.0	Anthraquinone	3	1520914	β-NPD
4			4	2633850	naphthalene

Table 19. Search results of unknowns from a prefilter process and a second stage comparison

Prefilter:			Second stage Comparison:		
1,3,6,8-TETRAPHENYLPYRENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	85.9	1,3,6,8-tetraphenyl pyrene	[1]	909666	1,3,6,8-tetra-phenylpyrene
2	67.6	9,10-diphenyl-anthracene	2	2562350	9-vinylanthracene
3	64.8	BBOT	3	2705496	9,10-dichloro-anthracene
4	62.2	9,10-dichloro-anthracene	4	2806203	9,10-diphenyl-anthracene
BBOT					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	82.4	BBOT	[1]	928191	BBOT
2	69.1	9-vinylanthracene	2	1522075	9-vinylanthracene
3	65.8	1,3,6,8-tetraphenyl pyrene	3	2651167	9,10-dichloro-anthracene
4	64.6	9,10-diphenyl-anthracene	4	2677472	9,10-diphenyl-anthracene
POPOP					
Rank	%	Spectra	Rank	Total Diff.	spectra
[1]	87.2	POPOP	1	2754258	2-phenyl-naphthalene
2	75.2	2-phenylnaphthalene	2	2854125	diphenylstilbene
3	75.1	Diphenylstilbene	3	2967676	BBO
4	74.2	2-methylanthracene	[4]	3137099	POPOP

not correctly identified in the first stage comparison were taken and subjected to a second stage comparison. Table 20 shows the results for these unknowns. After the second stage comparison, all of these unknowns were correctly identified. The prefilter method can, therefore, satisfactorily eliminate unlikely contenders. In 85.7% of the cases (Table 10), it leads directly to a correct identification of unknowns. In 93.8% of the cases (Tables 16-20), application of the second comparison step leads to a correct identification.

The dual level comparison saves considerable time and space. The prefilter library is 1.6% the size of the original data set. The second stage library is 25% the size of the original data set. Based on this library example, one can save 74.4% of the original space. The prefilter data base uses a binary pattern and the comparison process is extremely fast and efficient. The second stage comparison is more complicated, but only a few spectra need to be compared at this stage.

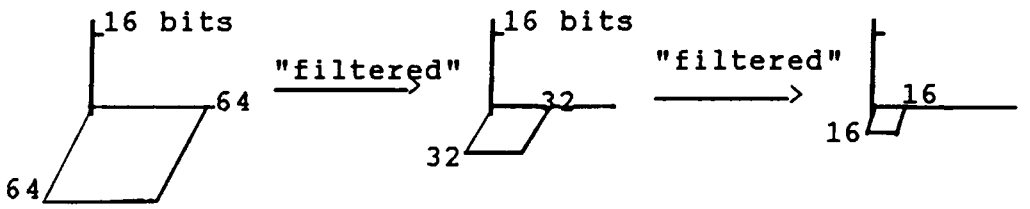
The above process is based on a prefilter which rejects 90% of the compounds in the library, leaving only 10% to be subjected to the second stage comparison. The percentage of compounds which are subjected to the second stage comparison would, of course, depend on the number of the library members. This would be an easy criteria to program into an automatic system.

Table 20. Search results of unknowns from a prefilter process and a second stage comparison

Prefilter:			Second stage Comparison:		
9-PHENYLANTHRACENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
1	80.3	9-Me-anthracene	[1]	653786	9-phenylanthracene
[2]	79.9	9-phenylanthracene	2	1186446	9-Me-anthracene
3	63.2	Acridine	3	1584577	2-Me-anthracene
4	58.9	Dimethyl POPOP	4	2143822	Dimethyl POPOP
PPD					
Rank	%	Spectra	Rank	Total Diff.	spectra
1	89.2	4-vinylbiphenyl	[1]	662985	PPD
[2]	71.9	PPD	2	1154442	4-vinylbiphenyl
3	70.1	Naphthalene	3	1416390	Naphthalene
4	61.4	2-methylnaphthalene	4	1503570	2,3-dimethyl-naphthlene
1-PHENYLNAPHTHALENE					
Rank	%	Spectra	Rank	Total Diff.	spectra
1	77.1	1-Me-naphthalene	[1]	173526	1-phenyl-naphthalene
2	76.1	1-naphthol	2	859165	2,6-dimethyl-naphthalene
[3]	69.4	1-phenylnaphthalene	3	1142886	1-Me-naphthalene
4	63.2	2,6-dimethyl-	[4]	3240218	1-naphthol

Investigation of a Filtering Rate

Figure 4a and 4b show an original spectrum of anthracene and its Hadamard transformed spectrum. This transformed spectrum was then filtered to a different extent in the previous application of the second stage comparison, the filtering rate utilized was 25% (32 by 32 data points are retained) which is shown in the Figure 14a. As shown in the Figure 14, the filtering rate was increased. Figure 14b, 14c and 14d represent 6.25% [(16 by 16 data points/64 by 64 data points of the original data base)*100%], 1.56%, and 0.39% of the original data base, respectively.



64 x 64 x16bits	32 x 32 x 16bits	16 x 16 x 16bits
=4096*16 = 4k*16	= 1024*16 = 1k*16	=256*16
100% data base	25% data base	6.25% data base

These were then they were Inverse Hadamard Transformed into the original domain (i.e., emission vs. excitation). Figures 15a and 15b are the results of inverse transformation of the 25% and 6.25% filtered transformed spectra and Figure 16a and 16b are the results of inverse transformation of the 1.56% and 0.39% filtered transformed spectra. As the filtering rate increases from 25% to 0.39%, the inverse transformed spectra loses more of the higher

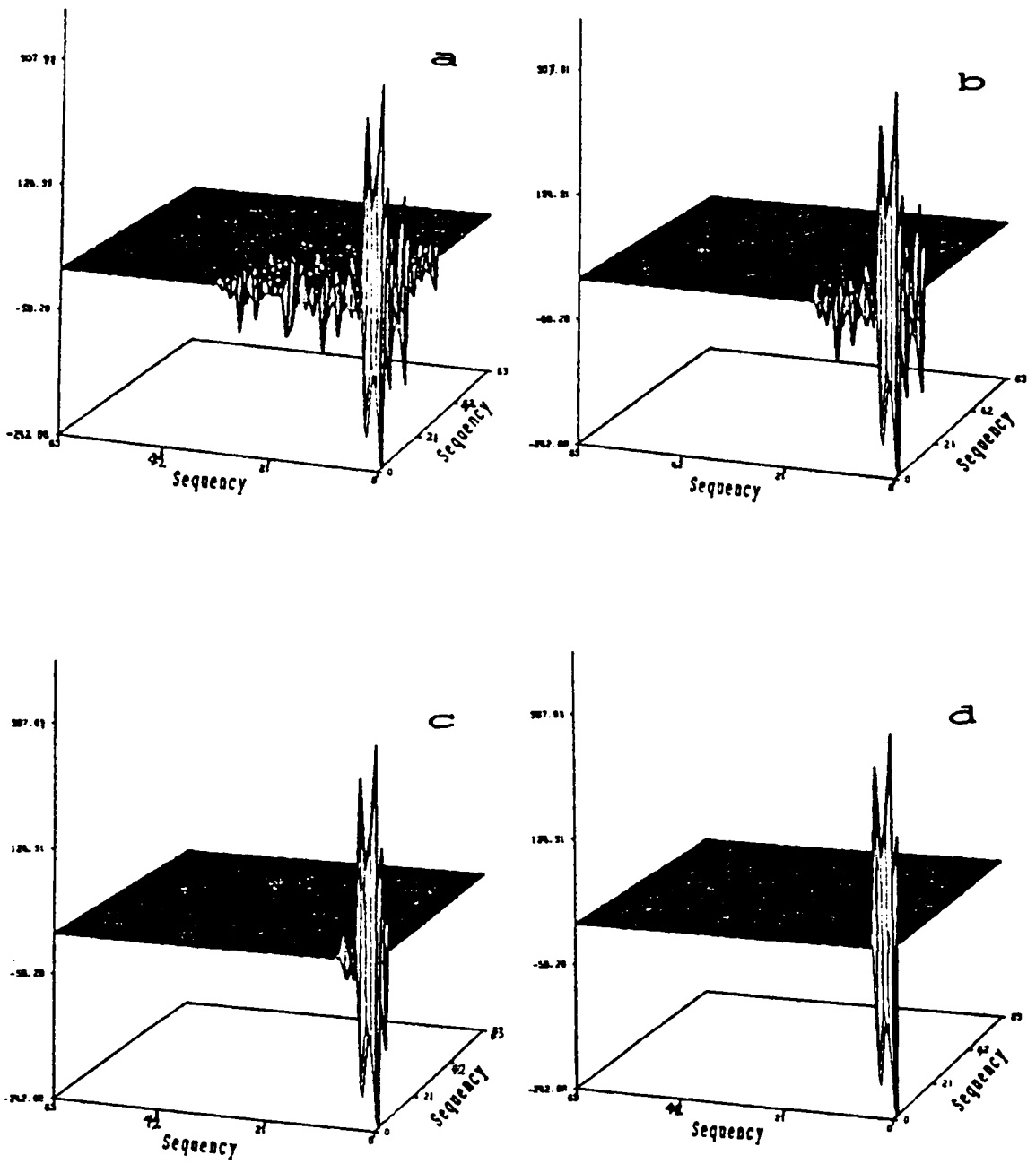


Figure 14. Anthracene transformed spectra
a) 25% filtered (32x32 data points retained)
b) 6.25% filtered (16x16 data points retained)
c) 1.56% filtered (8x8 data points retained)
d) 0.39% filtered (4x4 data points retained)

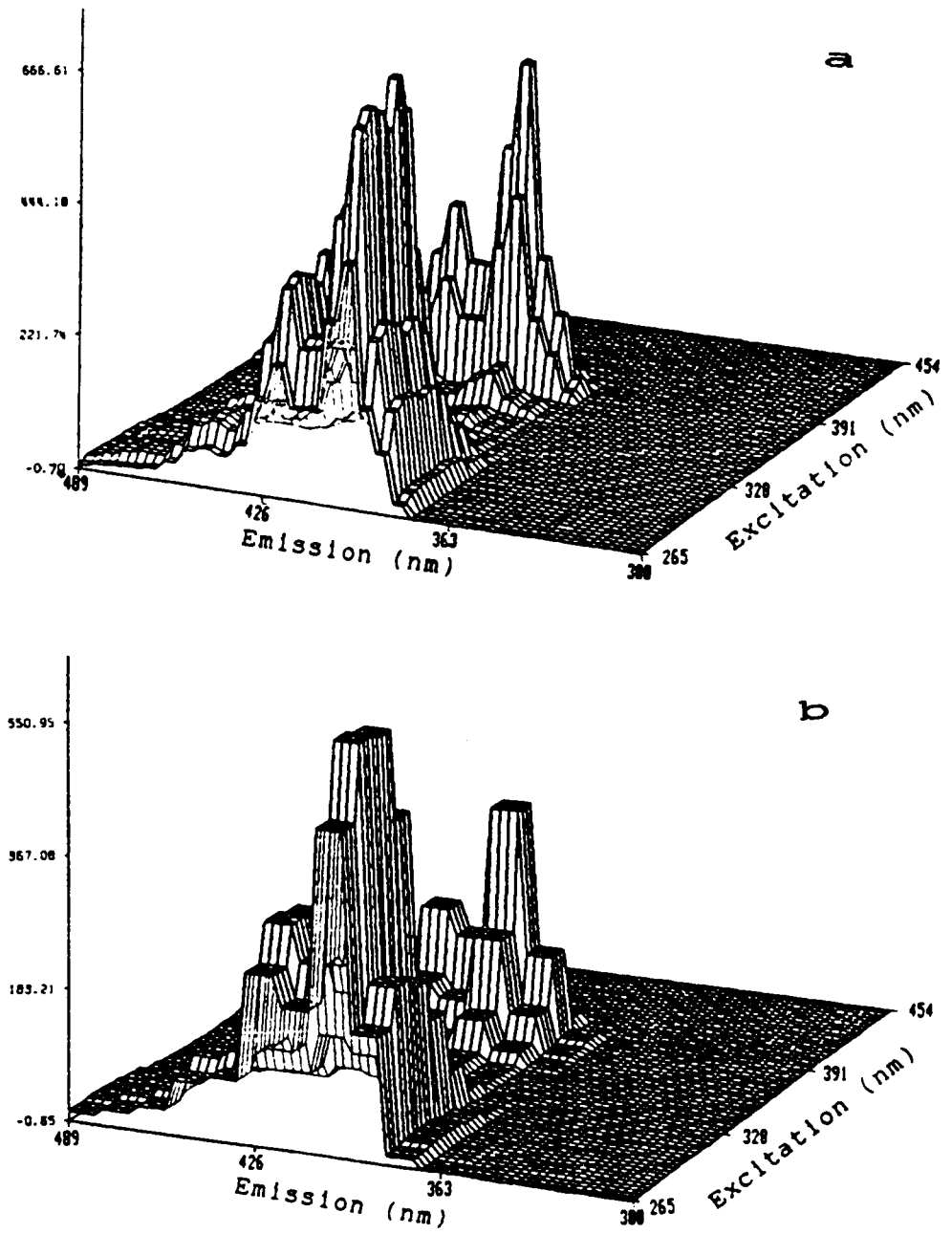


Figure 15. Anthracene inversed transformed spectrum
a) using 25% filter (32x32 lower sequency region)
b) using 6.25% filter (16x16 lower sequency region)

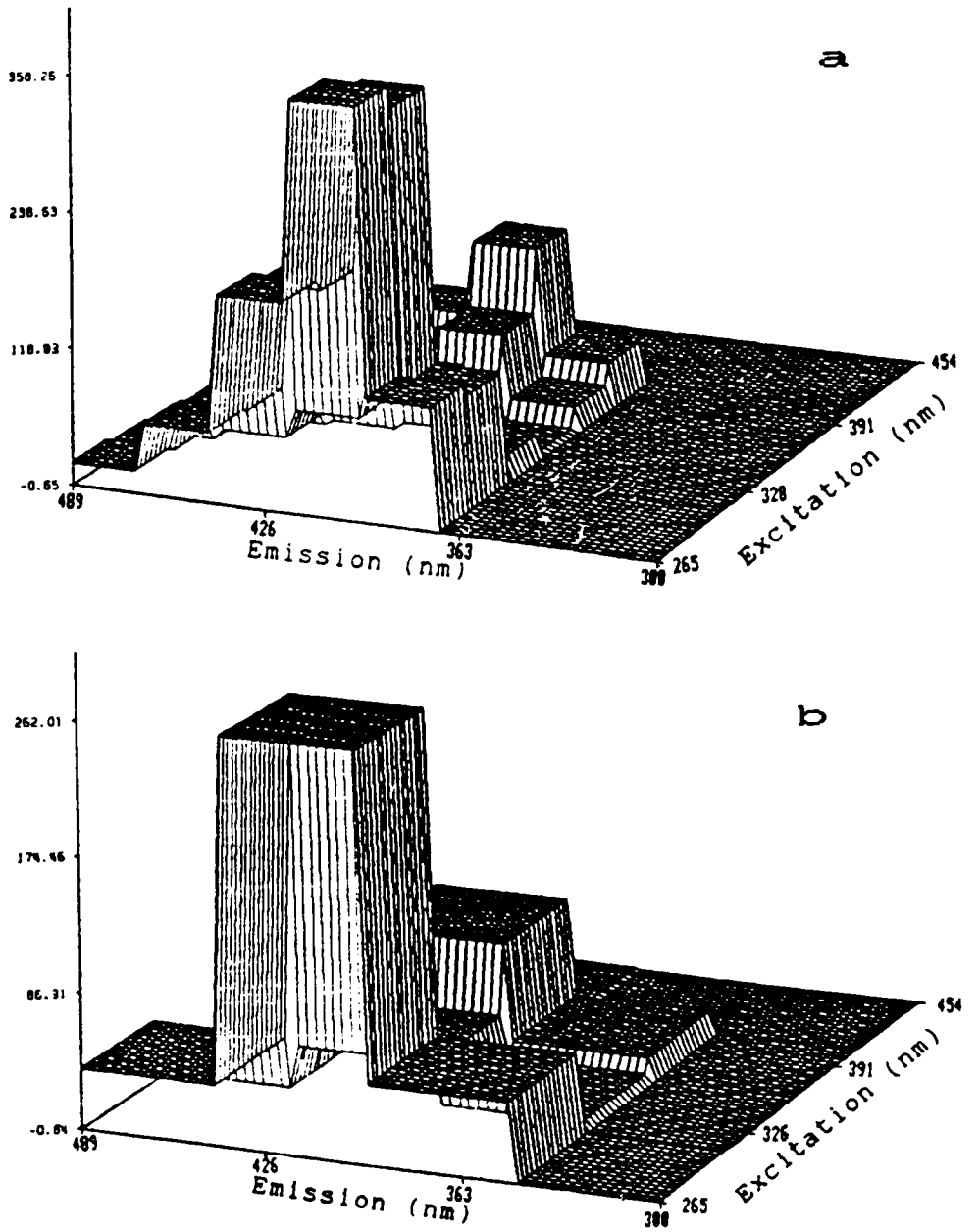


Figure 16. Anthracene inversed transformed spectrum
a) using 1.56% filter (8x8 lower sequency region)
b) using 0.39% filter (4x4 lower sequency region)

resolution information. It is obvious that the inverse transformed spectrum of the 25% filtered transformed spectrum is the closest to the original spectrum. However very sever spectral deterioration exists in the inverse transformed spectra which are the results of the 1.56% and the Ø.39% filtered transformed spectra (Figure 16a and 16b). One is left with the examination of the performance of the second stage comparison (i.e., the subtraction method) using the 6.25% filtered transformed spectra as a data base.

The Tables 21 through 3Ø show the results of this second stage comparison using the 25% filtered transformed data base and the 6.25% filtered data base. These results indicate that the use of the 6.25% filtered data base generates the same results as the the 25% filtered data base. Further compression is possible by using 6.25% filtered data base. This suggests that if the appearance of the inverse transformed spectra are not important, then the 6.25% filtered transformed spectra can be used as the data base for the second stage comparison instead of the 25% filtered transformed spectra.

D. EFFECTS OF NOISE TOWARDS THE FIRST STAGE SEARCH RESULTS

An important factor to consider is the effect that noise might have on the efficiency of the search results. It is impossible to obtain spectra without superimposed

Table 21. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF naphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
11	826006	Naphthalene
6	1428394	4-Vinylbiphenyl
38	1626349	PPD
14	1631406	2,3-Dimethylnaphthalene
9	11841563	Quinoline

16 x 16 data points are used
SEARCH RESULT OF naphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
11	382323	Naphthalene
6	704919	4-Vinylbiphenyl
14	873247	2,3-Dimethylnaphthalene
38	923765	PPD
9	7159147	Quinoline

32 x 32 data points are used
SEARCH RESULT OF 1-Methylnaphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
12	712310	1-Methylnaphthalene
13	1573688	2-Methylnaphthalene
15	1639775	2,6-Dimethylnaphthalene
16	2955738	1-Naphthol
9	11753944	Quinoline

16 x 16 data points are used
SEARCH RESULT OF 1-Methylnaphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
12	294263	1-Methylnaphthalene
15	805604	2,6-Dimethylnaphthalene
13	808131	2-Methylnaphthalene
16	1845625	1-Naphthol
9	6992237	Quinoline

Table 22. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF 1-Naphthol

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
16	1630724	1-Naphthol
12	2747580	1-Methylnaphthalene
44	3519175	bNPD
13	3556428	2-Methylnaphthalene
33	3867622	Chrysene

16 x 16 data points are used
SEARCH RESULT OF 1-Naphthol

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
16	653927	1-Naphthol
12	1626769	1-Methylnaphthalene
44	1840791	bNPD
33	2053284	Chrysene
13	2074305	2-Methylnaphthalene

32 x 32 data points are used
SEARCH RESULT OF 1-Phenylnaphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
18	173526	1-Phenylnaphthalene
15	859165	2,6-Dimethylnaphthalene
12	1142886	1-Methylnaphthalene
16	3240218	1-Naphthol
9	11818032	Quinoline

16 x 16 data points are used
SEARCH RESULT OF 1-Phenylnaphthalene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
18	92540	1-Phenylnaphthalene
15	503059	2,6-Dimethylnaphthalene
12	513166	1-Methylnaphthalene
16	2035326	1-Naphthol
9	7122120	Quinoline

Table 23. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF 1,1-Binaphthyl

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
34	707610	1,1-Binaphthyl
12	2033241	1-Methylnaphthalene
18	2079369	1-Phenylnaphthalene
16	2528787	1-Naphthol
19	2558336	2-Phenylnaphthalene

16 x 16 data points are used
SEARCH RESULT OF 1,1-Binaphthyl

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
34	416715	1,1-Binaphthyl
12	1277895	1-Methylnaphthalene
16	1387761	1-Naphthol
18	1526803	1-Phenylnaphthalene
19	1723632	2-Phenylnaphthalene

32 x 32 data points are used
SEARCH RESULT OF Anthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
22	81622	Anthracene
30	993549	Methylanthracene
21	3216916	Acridine
55	3438790	Anthraquinone
49	3885320	Pyrene

16 x 16 data points are used
SEARCH RESULT OF Anthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
22	62707	Anthracene
30	529459	Methylanthracene
21	2082691	Acridine
55	2244374	Anthraquinone
49	2567389	Pyrene

Table 24. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF 1-Aminoanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
23	213975	1-Aminoanthracene
4	1325882	4-Biphenylphenylether
24	1538823	2-Aminoanthracene
53	3044215	BBOT
3	13391526	1,1-Diphenylethylene

16 x 16 data points are used
SEARCH RESULT OF 1-Aminoanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
23	142749	1-Aminoanthracene
4	959376	4-Biphenylphenylether
24	1162158	2-Aminoanthracene
53	2022732	BBOT
3	6985260	1,1-Diphenylethylene

32 x 32 data points are used
SEARCH RESULT OF 9-Methylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
26	487946	9-Methylanthracene
27	1277092	9-Phenylanthracene
1	1856858	Anthranilicacid
25	2484251	9,10-Dichloroanthracene
21	3021656	Acridine

16 x 16 data points are used
SEARCH RESULT OF 9-Methylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
26	197805	9-Methylanthracene
27	696474	9-Phenylanthracene
1	1206319	Anthranilicacid
25	1463387	9,10-Dichloroanthracene
21	2021111	Acridine

Table 25. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF 9,10-Dichloroanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
25	990434	9,10-Dichloroanthracene
26	2305851	9-Methylanthracene
53	2336688	BBOT
50	2452046	1368-Tetraphenylpyrene
28	2590195	9,10-Diphenylanthracene

16 x 16 data points are used
SEARCH RESULT OF 9,10-Dichloroanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
25	486248	9,10-Dichloroanthracene
53	1454412	BBOT
28	1471711	9,10-Diphenylanthracene
26	1564282	9-Methylanthracene
50	1571004	1368-Tetraphenylpyrene

32 x 32 data points are used
SEARCH RESULT OF 9-Phenylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
27	653786	9-Phenylanthracene
26	1186446	9-Methylanthracene
30	1584577	Methylanthracene
46	2143822	DimethylPOPOP
21	2800792	Acridine

16 x 16 data points are used
SEARCH RESULT OF 9-Phenylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
27	311353	9-Phenylanthracene
26	743324	9-Methylanthracene
30	954234	Methylanthracene
46	1412876	DimethylPOPOP
21	1903080	Acridine

Table 26. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF 9-Vinylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
29	683924	9-Vinylanthracene
53	2048752	BBOT
26	2075519	9-Methylanthracene
28	2085545	9,10-Diphenylanthracene
40	3039697	BBO

16 x 16 data points are used
SEARCH RESULT OF 9-Vinylanthracene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
29	383947	9-Vinylanthracene
26	1161774	9-Methylanthracene
53	1303020	BBOT
28	1338603	9,10-Diphenylanthracene
40	1896644	BBO

32 x 32 data points are used
SEARCH RESULT OF Chrysene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
33	1830711	Chrysene
39	3148459	PBD
44	3223166	bNPD
35	3299091	2,2-Binaphthyl
48	3635134	Triphenylene

16 x 16 data points are used
SEARCH RESULT OF Chrysene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
33	761221	Chrysene
39	1840014	PBD
44	1949426	bNPD
48	2057776	Triphenylene
35	2088800	2,2-Binaphthyl

Table 27. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF Pyrene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
49	1830296	Pyrene
1	3786916	Anthranilicacid
47	4454068	4,5-Diphenylimidazule
7	4566376	Triphenylamine
9	12694063	Quinoline

16 x 16 data points are used
SEARCH RESULT OF Pyrene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
49	838181	Pyrene
1	2302129	Anthranilicacid
47	2875419	4,5-Diphenylimidazule
7	2886171	Triphenylamine
9	7588002	Quinoline

32 x 32 data points are used
SEARCH RESULT OF 1368-Tetraphenylpyrene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
50	909666	1368-Tetraphenylpyrene
29	2562350	9-Vinylanthracene
25	2705496	9,10-Dichloroanthracene
28	2806203	9,10-Diphenylanthracene
53	3047234	BBOT

16 x 16 data points are used
SEARCH RESULT OF 1368-Tetraphenylpyrene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
50	627358	1368-Tetraphenylpyrene
25	1581584	9,10-Dichloroanthracene
29	1718127	9-Vinylanthracene
28	1827709	9,10-Diphenylanthracene
53	2023148	BBOT

Table 28. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF Perylene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
51	1698998	Perylene
23	4064738	1-Aminoanthracene
26	4314565	9-Methylanthracene
27	4391801	9-Phenylanthracene
22	4501369	Anthracene

16 x 16 data points are used
SEARCH RESULT OF Perylene

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
51	1068212	Perylene
23	2788122	1-Aminoanthracene
26	3000765	9-Methylanthracene
27	3001828	9-Phenylanthracene
22	3111759	Anthracene

32 x 32 data points are used
SEARCH RESULT OF PPO

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
37	717323	PPO
35	1380767	2,2-Binaphthyl
44	1520914	bNPD
11	2633850	Naphthalene
55	3177756	Anthraquinone

16 x 16 data points are used
SEARCH RESULT OF PPO

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
37	282524	PPO
35	561838	2,2-Binaphthyl
44	586460	bNPD
11	1566986	Naphthalene
55	2127884	Anthraquinone

Table 29. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF PPD

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
38	662985	PPD
6	1154442	4-Vinylbiphenyl
11	1416390	Naphthalene
14	1503570	2,3-Dimethylnaphthalene
13	1529857	2-Methylnaphthalene

16 x 16 data points are used
SEARCH RESULT OF PPD

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
38	316822	PPD
6	465064	4-Vinylbiphenyl
11	853974	Naphthalene
14	953792	2,3-Dimethylnaphthalene
13	976660	2-Methylnaphthalene

32 x 32 data points are used
SEARCH RESULT OF BBOT

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
53	928191	BBOT
29	1522075	9-Vinylanthracene
25	2651167	9,10-Dichloroanthracene
28	2677472	9,10-Diphenylanthracene
50	2864617	1368-Tetraphenylpyrene

16 x 16 data points are used
SEARCH RESULT OF BBOT

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
53	454494	BBOT
29	1005449	9-Vinylanthracene
25	1440174	9,10-Dichloroanthracene
28	1728029	9,10-Diphenylanthracene
50	1920306	1368-Tetraphenylpyrene

Table 30. Search results of second stage comparison using 25% and 6.25% filtered Hadamard transformed data base

32 x 32 data points are used
SEARCH RESULT OF POPOP

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
19	2754258	2-Phenylnaphthalene
52	2854125	Diphenylstilbene
40	2967676	BBO
45	3028431	POPOP
30	3137099	Methylanthracene

16 x 16 data points are used
SEARCH RESULT OF POPOP

SPECTRUM NUMBER	TOTAL VALUE OF DIFFERENCE	NAME OF THE SPECTRUM
19	1776057	2-Phenylnaphthalene
40	1919757	BBO
52	1928040	Diphenylstilbene
30	2051821	Methylanthracene
45	2135001	POPOP

random and fixed pattern noise. This suggests that any two fluorescence spectra of the same compound will not be exactly identical. The upper part of Table 31 shows the effect of random noise on search results. This random noise was deliberately added to each spectrum. The intensity of this added noise ranged from 0.5% of the most intense peak (S/N = 200) in the spectrum to 10.0% (S/N = 10). As expected, with each addition of noise, the match rate (expressed as percentile) decreases. Two types of data bases were used; a 100% clipped data set and a 25% clipped data set. The 25% clipped data base results in better match rates. In both cases the match rates decrease slightly until the intensity of added noise reaches 10.0%. At this point, the match rate in both cases decreases dramatically, indicating that the search system is tolerant to noise up to S/N ratios of 10.

The last column of Table 31 shows the differences (in percentile) for the two types of data base at a given noise intensity. The difference between the two data bases at no added noise is 1.3%. As the noise intensity increases the difference becomes larger. At 10.0% noise level, the difference is 12.2%. The data suggest that the effect of random noise is greater with the 100% clipped data base than 25% clipped data base. The reasons for this are discussed below.

Table 31. Effects of noise

First step search results of
Anthracene with randam noise added

Noise Added	Percentile 100% clipped	25% clipped	[25% clipped]-[100% clipped]
0.0%	96.2%	97.7%	1.3%
0.5%	94.7%	97.5%	3.0%
1.0%	84.8%	94.0%	9.2%
5.0%	81.5%	92.3%	10.8%
10.0%	64.5%	76,7%	12.2%

First step search results of
Anthracene with Gaussian peak added

Rank	Percentile 100% clipped	25% clipped
1	90.7%	87.8%

After the Hadamard transformation, most of the characteristic features of the spectra are located around the lower sequency region, and random noise components exist at higher sequencies. It is therefore not surprising that the 100% clipped data base gives less satisfactory results than the 25% clipped data base which has discarded the higher sequencies.

The lower region in Table 31 shows the effect of the search results when a broad peak, typical of background fluorescence, is added to the spectra. The broad peak was simulated by a gaussian peak. This broad three-dimensional gaussian peak had a dimension of 32 by 32. It was added to the anthracene spectra at the center of the data region. Because this is a broad peak, after the Hadamard transformation, most of the characteristic information resides in the lower sequency region.

The effect of this types of noise is surprisingly small compared to the random noise discussed above. With the 25% clipped data base the match rate is 87.8% and with the 100% data base, the match rate is 90.7% (compared to no noise). The match rate with the 100% clipped data base is much higher than that of 25% clipped data base. This is because the importance of the lower sequency region is emphasized when the 25% clipped data base is used. However, even with the 25% data base, the results are still satisfactory and the correct compound is ranked as the 1st hit. Since it is

Table 32. Search result of unknown phenanthrene

Name of spectra	March rates (%)	Rank	Ave. Search Results (%)
Phenanthrene_A	81.9	1	53.7
Phenanthrene_B	83.8	1	53.9
Phenanthrene_C	83.4	1	53.7
Phenanthrene_D	83.Ø	1	53.7
Phenanthrene_E	83.1	1	53.7
Phenanthrene_F	83.9	1	53.7
Average	83.2		53.7
Stnd. Deviation	Ø.7		Ø.1

important to eliminate those regions which might contain more random noise information than spectral information, the smaller data base should be utilized in library searches.

To show the applicability of the compression technique and the search system, phenanthrene was measured in six separate experiments, and the results were compared. The Table 32 shows the results of this study. They all identified the compound correctly at the first hit with an average match rate of 83.2±Ø.7%. The results of this study show that the system is very stable and consistent.

E. SOLVENT EFFECTS

Because of solute-solvent interactions, the shapes of the fluorescence spectra of organic compounds can be changed dramatically, depending on the solvents used[42-46]. These effects are greater if the PNA compounds of interest have polar substituents on their aromatic rings. Two different types of the effects are observed. One is a "general solvent effect" and the other is termed a "specific solvent effects"[43]. General solvent effects result from the interaction of the dipole moment of the solvent molecules and the free electron motion in the solute molecules. Specific solvent effects are due to specific chemical interactions between the solute and the solvent molecules, such as hydrogen bonding and charge transfer interactions.

In general, if a solvent is changed to a more polar material, the resulting emission shifts to longer wavelengths due to the change in the energy difference between the ground and the excited states. If the only solvent-induced change is in the position of the emission, then the shape of the spectra is not altered.

However, positional shifts are not the only solvent effects observed. In many cases, emission spectra in polar solvents lose structural resolution due to vibrational states[43,44]. This effect dramatically changes the shape of spectra as one moves from a nonpolar solvent.

As noted earlier these effects are minimal if the PNA compounds lacks polar substituents. Table 33 shows the results of the 2,3-dimethylnaphthalene (which has no polar substituents) in cyclohexane vs. ethanol. The reference library contained spectra taken in cyclohexane and ethanol. Another set of the 2,3-dimethylnaphthalene spectra were taken in both solvent and used as unknowns. A fluorescence spectral shift due to a change in the polarity of the solvent (from cyclohexane to ethanol) in this case is less than 3nm. When 2,3-dimethylnaphthalene dissolved in ethanol is subjected to compression/search, the first hit was 2,3-dimethylnaphthalene in ethanol with a match rate of 81.Ø%. The second hit was 2,3-dimethylnaphthalene in cyclohexane with a hit rate of 8Ø.4%. In this case, the solvent does not affect the search results. However it is important to

Table 33. The effect of solvent on a non-polar compounds

Spectra/Solvent	Percentile/Solvent	Difference
2,3-dimethylnaphthalene/C	78,5%/C 77.9%/E	Ø.6%
2,3-dimethylnaphthalene/E	81.Ø%/E 8Ø.4%/C	Ø.6%

C == Cyclohexane
E == Ethanol

examine the effect of the shift it is greater than 3nm.

From the reference library, a 2,3-dimethylnaphthalene spectrum (dissolved in cyclohexane) was first selected. Then the spectrum was manually red shifted (toward longer emission wavelength) for 3nm and added to the library. The same process was repeated to create a 6nm and 12nm red shifted spectra. They were compressed to generate the 25% filtered and clipped data bases and subjected to a comparison with 2,3-dimethylnaphthalene. The results are shown in Table 34. The search with unshifted 2,3-dimethylnaphthalene as an unknown should generate a 100% match with itself which is shown in the top row of the table. The result shows that the match rate decreased dramatically for the 6nm shifted spectra. However, since the resultant match rate, 67.3%, is still greater than 66.8% (three standard deviations away from the average match rate), this indicates that a significant amount of similarity between the spectra still exists. The match rates for spectra with larger shifts are much smaller. From these result, the compression/search technique can handle spectral shifts up to 6nm. If the spectral shift due to the polarity of solvent for a particular PNA compound is large, then a complete spectral library would have to include representatives of that compound with several solvents.

Polynuclear aromatic compounds which contains polar substituents on their rings dissolved in cyclohexane show

Table 34. The effect of the wavelength shift on a Hadamard Transformed data base

Spectra/wavelength-shift	Percentile	
2,3-dimethylnaphthalene/no shift	100.0%	
2,3-dimethylnaphthalene/3 nm	83.4%	
2,3-dimethylnaphthalene/6 nm	67.3%	
	-----	66.8%
2,3-dimethylnaphthalene/9 nm	62.0%	
2,3-dimethylnaphthalene/12 nm	54.0%	

sharper vibrational structure in fluorescence spectra than when dissolved in polar solvents like ethanol or methanol. This change in the shape of their spectra is detected by the library search system. Fluorescence spectra of a series of polar substituted PNAs were taken using different polarity solvents, (e.g., cyclohexane, ethanol and methanol). They were placed in the reference search library. Other measurements were taken in ethanol or methanol and they were used as unknowns.

Table 35 show the search results for those spectra. The specific solvent effects occurring between these polar PNA's and ethanol and methanol are due to hydrogen bonding. It is obvious that the percentile differences shown for these PNA's between nonpolar and polar solvents are much greater than the one obtained from 2,3-dimethylnaphthalene. These differences ranges from 8.3% to as high as 38.9%, indicating that the shapes of these spectra are very different from their counterparts in cyclohexane. This study demonstrates that the library search system is very sensitive to solvent effects. A complete library would have to have representative spectra for polar PNA's in several solvents.

F. CONCENTRATION EFFECT

The effect of concentration on spectral shape is another important factor to consider. Generally, the effect

Table 35. The effects of solvent on the polar compounds

Spectra/Solvent	Percentile/Solvent	Difference
PPO/E	79.9%/E	
	71.6%/C	8.3%
PPD/E	84.4%/E	
	72.5%/C	11.9%
PBD/E	80.9%/E	
	71.0%/C	9.9%
α-NPO/E	89.4%/E	
	79.6%/C	9.8%
α-NPD/E	84.6%/E	
	73.9%/C	10.7%
β-NPD/E	87.8%/C	
	73.8%/E	14.0%
1-naphthol/E	78.1%/E	
	53.3%/C	24.8%
2-naphthol/E	70.3%/E	
	51.4%/C	19.0%
POPOP/M	77.9%/M	
	57.2%/C	20.7%
POPOP/M	82.2%/B	
	43.3%/C	38.9%

of concentration on a fluorescence spectrum is a change in the intensity; higher concentration means more molecules to be excited, thus, more molecules fluoresce as they relax to the ground state. However, the clipping method removes intensity information. If the only difference between two spectra were their intensity, then after the first step comparison, the match rate should be 100%. Unfortunately, in certain cases, concentration (especially a high concentration solution) causes the shape of fluorescence spectra to alter[44,47,48].

1. Effect on the shapes of fluorescence excitation spectra

Because the data sets utilized are three-dimensional fluorescence spectra (excitation, emission and intensity), any effects that high concentration solutions have on the fluorescence excitation spectrum will affect the entire data set.

Such an effect on the fluorescence excitation spectrum is shown in the case of naphthalene. Figures 17 and 18 are fluorescence excitation spectra of naphthalene at concentrations of 1.00×10^{-5}M and 1.00×10^{-3}M respectively. Although the fluorescence spectra are not affected by the concentration, a dramatic influence on the fluorescence excitation spectra is shown.

Fluorescence spectra of naphthalene at different concentrations (1.00×10^{-5}M, 5.00×10^{-5}M, 1.00×10^{-4}M, 5.00×10^{-4}M, 1.00×10^{-3}M) were taken and stored in the reference

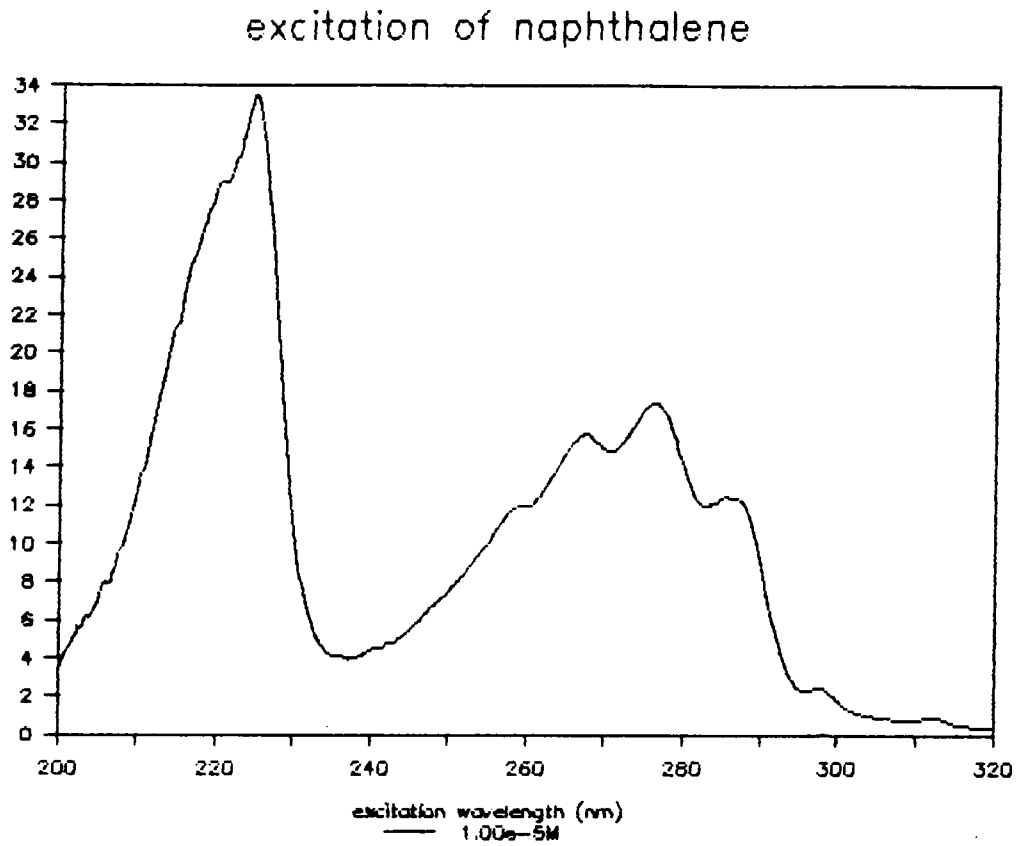


Figure 17. Fluorescence excitation spectrum of naphthalene at concentration level of 1.00×10^{-5}M

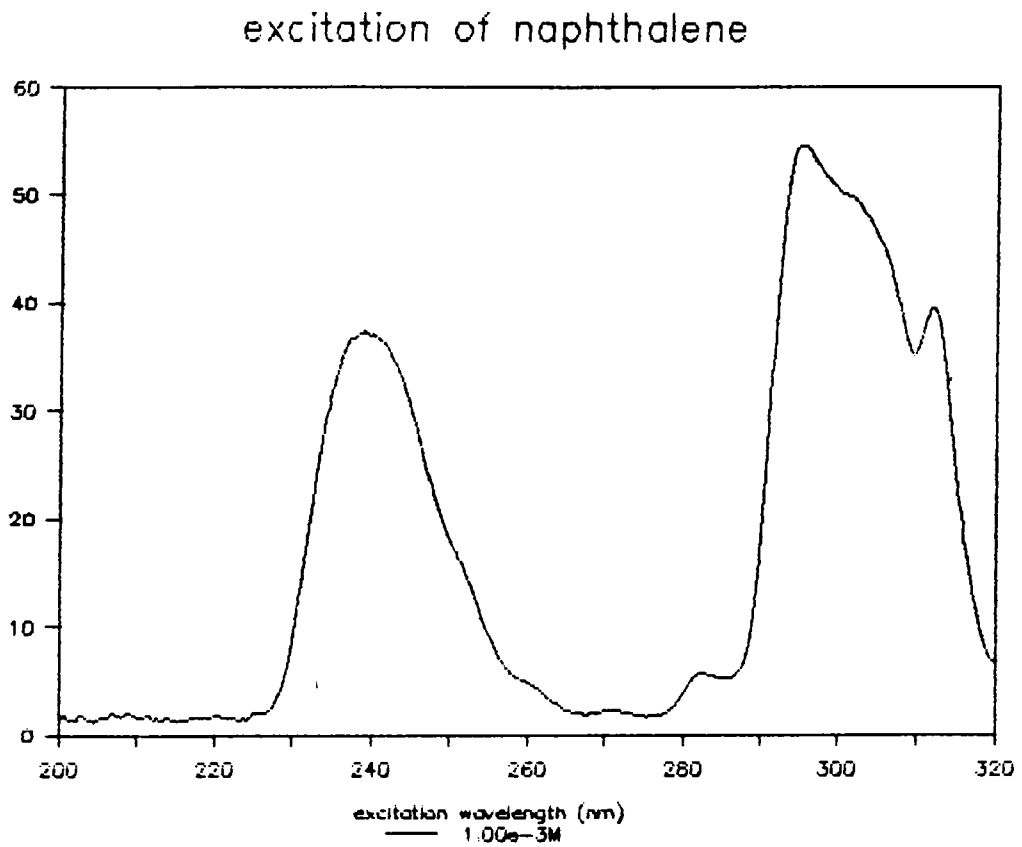


Figure 18. Fluorescence excitation spectrum of naphthalene at concentration level of 1.00×10^{-3}M

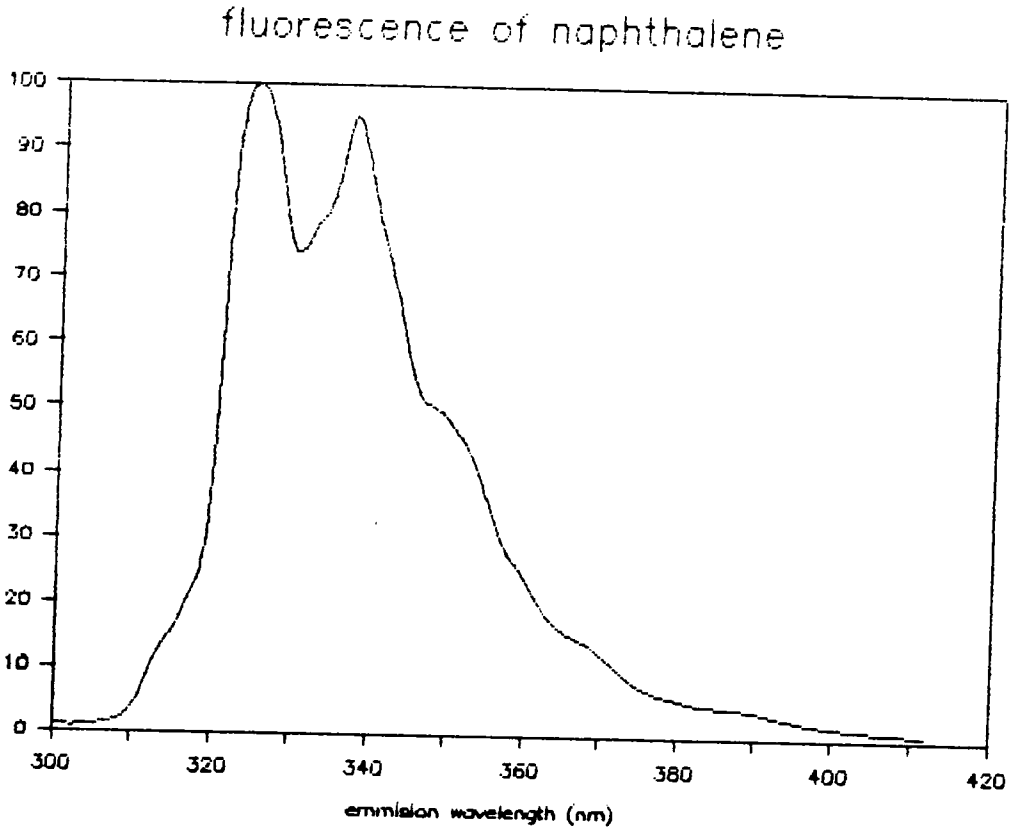


Figure 19. Fluorescence spectrum of naphthalene

search library. Another set of naphthalene solutions at five different concentration were taken and used as the unknowns. The results are shown in Table 36. The first column is the unknowns' concentration. The second column is the result of the search. Finally, the third column shows the rank ordering.

For the most highly concentrated unknown ($1.00x10^{-3}M$), the first rank hit was the identical library element. Reference spectrum with the second highest concentration ($5.00x10^{-4}M$) came in 2nd. However, the rest of the lower concentrations had significantly lower rank scores. For naphthalene at $5.00x10^{-4}M$ as an unknown, the various lower concentrations showed rank scores from second to fifth. Concentration is a factor that cannot be ignored. For the lower concentration unknowns, the results give high hit rank to library members with low concentrations, but significantly lower rank scores for high concentration library members. This implies that the searching technique is somewhat tolerant of different concentrations. But a complete library would have to contain representative spectra for both high concentration and low concentration solutions.

2. Effect on the shapes of fluorescence spectra

Another typical effect on the fluorescence spectrum is demonstrated by anthracene. If a peak at the shortest wavelength in a fluorescence spectrum overlaps with a peak

Table 36. The effect of the concentration on the first step library search (25% clipped data base of naphthalene used)

Naphthalene

Concentration	Percentile/Conc	Rank
1.00e-3M	98.7%/1.00e-3M	1
	78.4%/5.00e-4M	2
	57.6%/1.00e-5M	
	53.2%/5.00e-5M	
	51.9%/1.00e-4M	
5.00e-4M	99.3%/5.00e-4M	1
	77.2%/1.00e-3M	2
	71.6%/1.00e-4M	3
	69.4%/5.00e-5M	4
	67.6%/1.00e-5M	5
1.00e-4M	99.4%/1.00e-4M	1
	96.3%/5.00e-5M	2
	85.1%/1.00e-5M	3
	71.7%/5.00e-4M	4
	50.8%/1.00e-3M	
5.00e-5M	98.0%/5.00e-5M	1
	97.3%/1.00e-4M	2
	87.0%/1.00e-5M	3
	69.9%/5.00e-4M	4
	51.6%/1.00e-3M	
1.00e-5M	90.2%/1.00e-5M	1
	88.0%/5.00e-5M	2
	86.6%/1.00e-4M	3
	68.3%/5.00e-4M	4
	54.8%/1.00e-3M	

at a longer wavelength in the absorption spectrum (fluorescence excitation spectrum), then the intensity of the fluorescence peak at the shortest wavelength decreases due to the inner filter effect (Figure 20).

The fluorescence spectra of anthracene was measured at different concentrations ($1.00x10^{-5}M$, $5.00x10^{-5}M$, $1.00x10^{-4}M$, $5.00x10^{-4}M$, $1.00x10^{-3}M$) and stored in the reference search library. Another set of anthracene at five different concentrations were taken and used as unknowns. The results are shown in Table 37. The first column is the unknowns' concentration. The second column is the result of the search. Finally, the third column shows the ranks. For the case of the most highly concentrated unknown ($1.00x10^{-3}M$), the reference spectra for anthracene ($5.00x10^{-4}M$) came in 2nd. However, the rest of the lower concentration anthracene members of the library set had significantly lower ranks. For anthracene at $5.00x10^{-4}M$, the reference spectra of anthracene at $1.00x10^{-3}M$ ranked second. The rest of the lower concentrations showed significantly lower ranks. Lower concentrations showed high ranking hits for library members with low concentrations, but very low ranks for high concentrations. The results indicate that a significant amount of shape change in the spectra occurred between a concentration of $5.00x10^{-4}M$ and $1.00x10^{-4}M$.

A complete library would have representative spectra for high concentration solutions and lower concentration

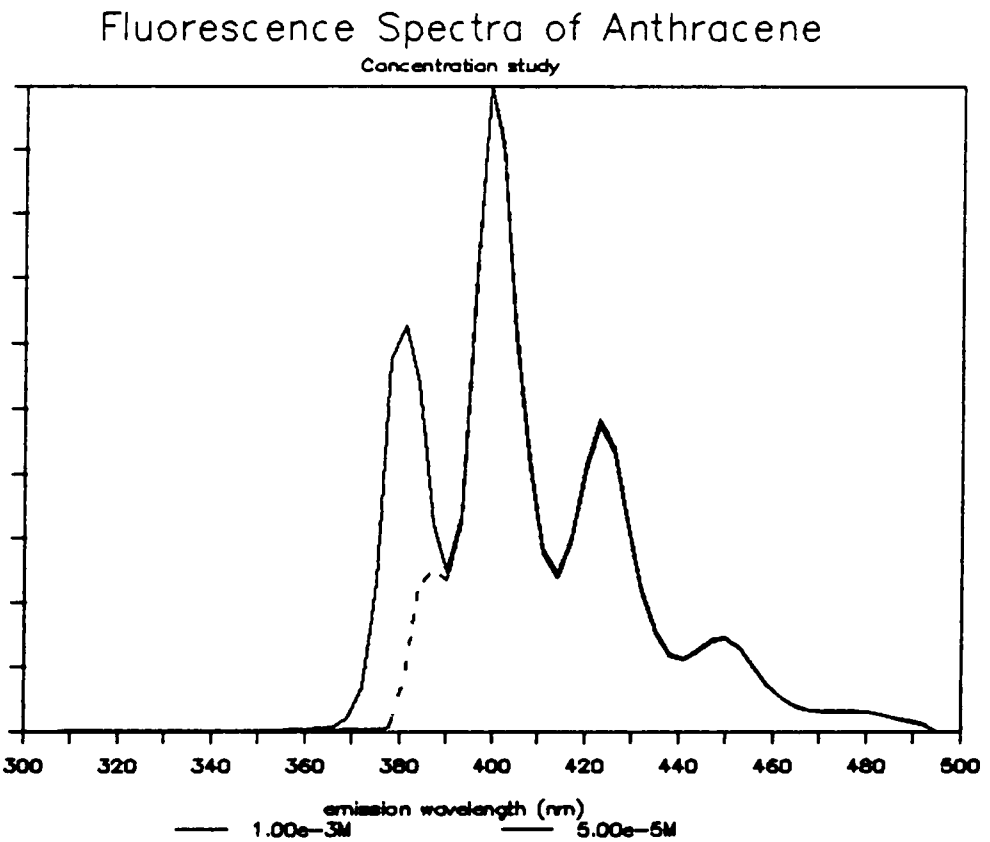


Figure 20. Fluorescence spectrum of anthracene (inner filter effect)

Table 37. The effect of the concentration on the first step library search (25% clipped data base of anthracene used)

Anthracene

Concentration	Percentile/Conc	Rank
1.00e-3M	98.7%/1.00e-3M	1
	80.3%/5.00e-4M	2
	46.8%/1.00e-5M	
	41.2%/5.00e-5M	
	40.6%/1.00e-4M	
5.00e-4M	99.5%/5.00e-4M	1
	79.3%/1.00e-3M	2
	54.6%/1.00e-4M	
	51.3%/5.00e-5M	
	49.0%/1.00e-5M	
1.00e-4M	90.9%/1.00e-4M	1
	84.5%/5.00e-5M	2
	64.5%/1.00e-5M	3
	54.2%/5.00e-4M	
	40.4%/1.00e-3M	
5.00e-5M	97.8%/5.00e-5M	1
	86.6%/1.00e-4M	2
	74.4%/1.00e-5M	3
	50.5%/5.00e-4M	
	41.2%/1.00e-3M	
1.00e-5M	97.9%/1.00e-5M	1
	75.6%/5.00e-5M	2
	65.2%/1.00e-4M	3
	48.2%/5.00e-4M	
	46.2%/1.00e-3M	

solutions.

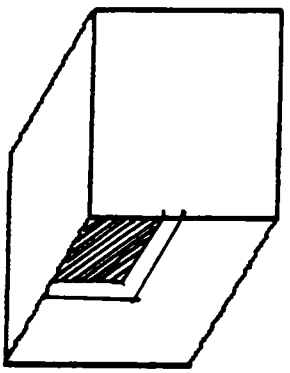
G. OPTIMAL COMPRESSION RATE

1. Elimination of high sequency area

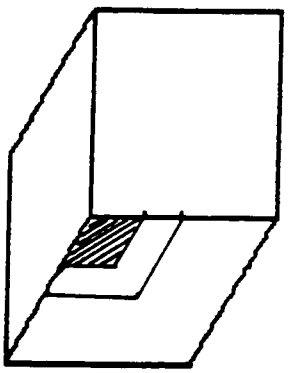
A series of experiments were undertaken to determine the optimal compression rate. The Figure 21 shows data bases representing 21%, 15%, 10% and 5% structures. Comparisons of an unknown with reference spectra were performed using these reduced clipped data bases.

Of the twenty unknowns used, twelve unknowns were identified correctly at the first hit for all the data bases used. However, as the amount of reduction increased, the remaining unknowns started to be correctly identified only at lower ranks. For example, naphthalene was correctly identified at the first hit with the 21% clipped data base. However, with the 15% clipped data base it was correctly identified as the second hit.

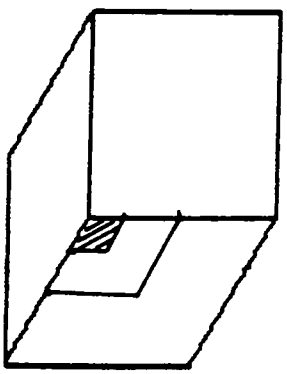
Those twelve unknowns that were identified correctly at the first hit for all the data bases were used to determine the optimum compression rate for the reference library used. After the comparisons, the difference between the correct first hit and the second hit was recorded. If each binary pattern still contained unique and characteristic information about a spectrum, then this difference should be large. As the pattern loses uniqueness, then the difference



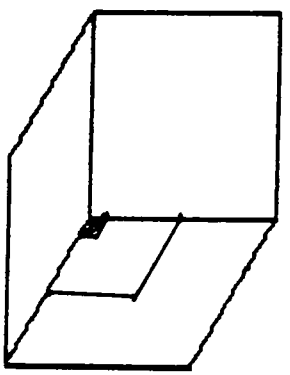
21% data base



15% data base



10% data base

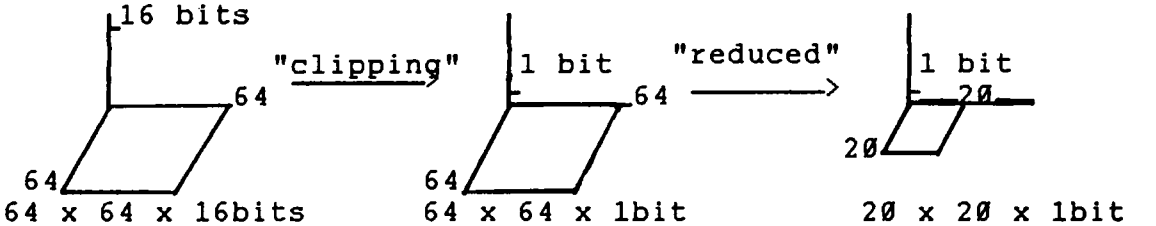


5% data base

Figure 21. Data base representations of a different compression rate

becomes smaller.

Table 38 shows the results using these reduced data bases. The Table 38 includes not only the 21% to 5% clipped data base sets, but also the 4%, 3%, 2%, 1%, and 0.6% clipped data base sets. For some compounds, when the reduced data bases were used, the difference between the first and second hit fell to zero. This indicates that each spectral binary pattern no longer contains unique information about the spectrum. The last row in the table shows the average difference for each reduced data base. The average difference between the first and second hit is largest when the 10% clipped data base is utilized suggesting that this is the optimal compression rate. It is important to note that at the first step, prefilter step, since each data point uses only 1 bit instead of 2 bytes, the actual compression rate is not 10%.



64 x 64 x 16bits = 4096*16 = 4k*16	64 x 64 x 1bit = 4096*1 = 4k*1	20 x 20 x 1bit = 400*1
100% data base	6.25% data base or 100% clipped data base	0.61% data base or 10% clipped data base

The ten percent clipped data base corresponds to 20 by 20 data points. Only one bit per datum is stored as the prefilter pattern. The original data base was 64 by 64 data

Table 38. Prefilter search results using different compression rate for data bases

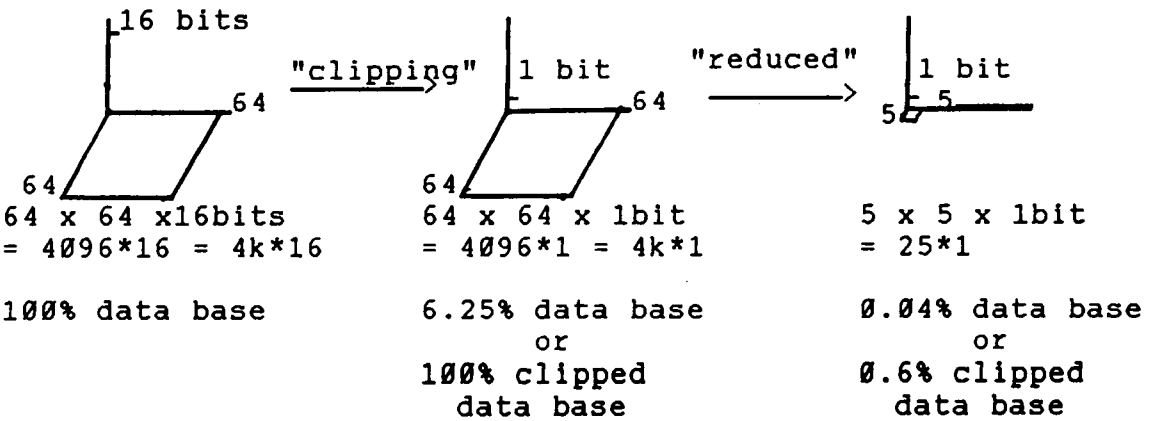
Name of spectra	Data bases used (compressed & clipped) 21%	15%	10%	5%	4%	3%	2%	1%	0.6%
1-Me-anthracene	13.8	15.7	19.0	10.7	11.2	12.4	16.1	8.3	0.0
1-amino anthracene	34.2	32.0	35.0	34.2	27.8	29.8	24.7	33.3	20.0
9-vinyl anthracene	14.6	16.0	25.2	14.3	7.7	9.1	11.1	0.0	0.0
9,10-dichloro anthracene	18.8	22.5	29.3	26.5	20.1	27.3	22.2	16.7	20.0
9-methyl anthracene	13.8	15.7	19.0	10.7	11.2	12.4	16.1	8.3	0.0
Anthracene	39.3	41.6	39.8	32.1	30.2	29.0	28.4	16.7	0.0
1-naphthol	5.2	3.7	6.3	10.2	10.1	9.1	12.4	11.1	8.0
1-methyl naphthalene	9.3	12.0	14.3	16.3	13.6	9.9	8.6	0.0	0.0
1,1-binaphryl	16.6	13.0	11.7	13.8	14.8	18.2	18.5	0.0	0.0
1,3,6,8-tetra phenylpyrene	18.1	17.0	22.5	27.6	22.5	25.6	30.8	25.0	12.0
BBOT	13.0	12.8	14.8	21.4	15.4	18.2	22.2	13.9	0.0
POPOP	13.2	12.5	13.0	12.3	4.7	9.1	9.9	16.7	16.0
Average	17.5	17.9	20.8	19.2	15.8	17.5	18.4	12.5	6.9

points, and sixteen bit integer representation was used. Thus, the prefilter library member is 0.61% the size of the original. At this optimum rate, the compression is 99.4%.

2. Elimination of lower sequency area

Up to this point all the reduction of the data bases has been concentrated on the higher sequency region. However, it is equally important to consider the importance of the lower sequency region. Although most of the important information lies around the lower sequency region, it is possible that a few members of the lowest sequency area may not be unique. For example, some of the lowest sequency region might be common to most of the spectra. A data base which excludes a few of members the lower sequency region needs to be examined.

As shown in the Table 38, when only the lowest sequency region, a 0.6% clipped data base consisting of 5 by 5 data points, is used as a data base, most of the match rate differences between the first and second ranked compounds are zero.



This indicates that the lowest sequency region does not contain any unique information and could be discarded. However, since these represent only a fraction of the entire data base, the elimination of such a small area would not increase the efficiency of the search performance.

Elimination of larger percentages of the lower sequency region was then examined. The starting data bases were 21%, 15%, and 10% clipped data sets. From these matrices, 5%, 1% or 0.6% of the low sequency region was eliminated. The blank area around the origin shown in Figure 22 indicates the low sequency region which is excluded from each data base. The shaded area in the Figure is the data base used for the library search.

Figure 23 shows the results of this study. Only those that were identified correctly at the first hit are shown in this graph. The y-axis is the difference between the first and second hit. Three boxes indicates the three starting data bases of 21%, 15% and 10%. Within each box is presented the results for the subtraction of 5%, 1% and 0.6%. Each connected line indicates how the search results perform for a given compound. Three types of observations can be made. As the lower sequency region eliminated increases, the differences may increase, decrease, or remain the same. It is obvious that the performance of the search results really depends on the compounds. Each compound shows a different tendency. For example in the case of

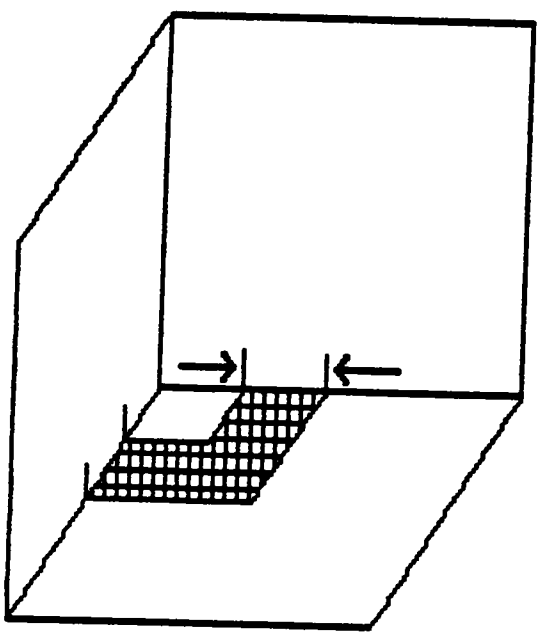


Figure 22. Data base representation of the compression window

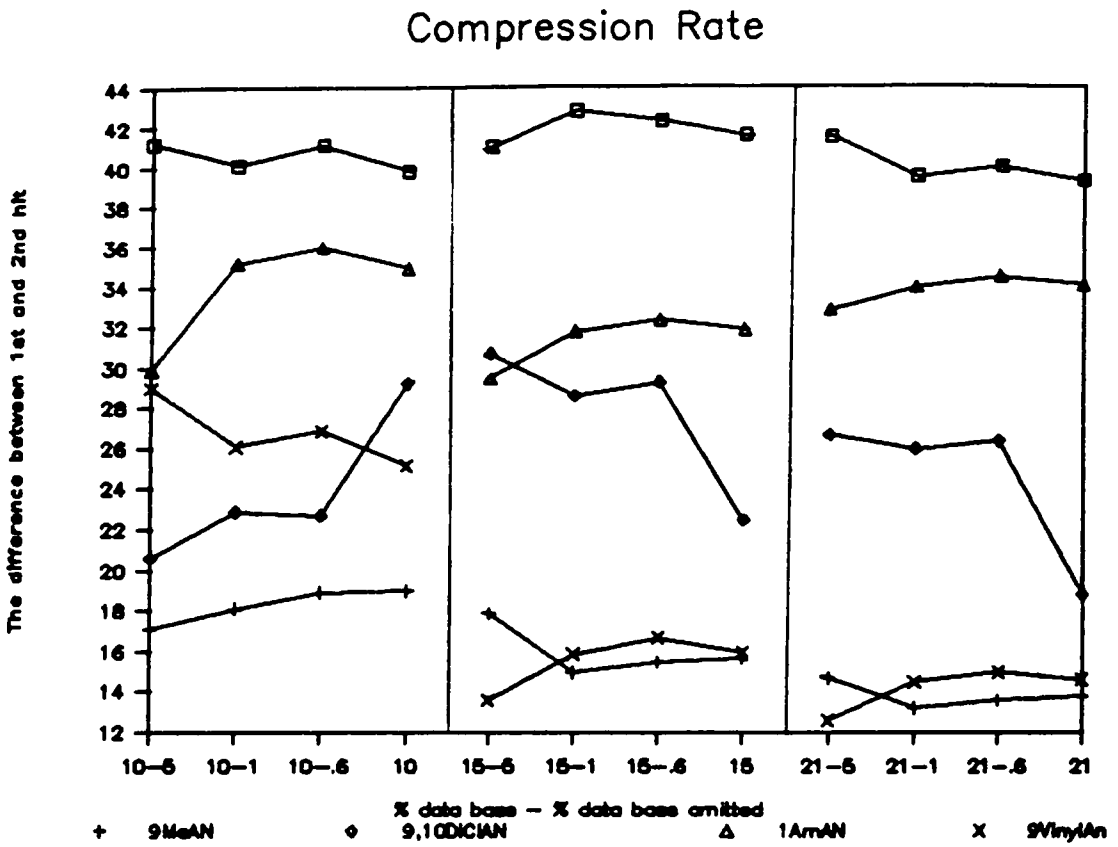


Figure 23. Comparison of a search performance of different data bases

9-vinylanthracene, no matter what the starting data base is, the differences increase as more lower sequency regions are eliminated. 1-aminoanthracene shows the exact opposite behavior. If all the unknowns showed the same tendency, either increasing or decreasing, one could conclude that a specific area of the data base contained the most unique pattern for all the unknowns. However, it is obvious that each unknown shows a unique tendency. This indicates that the characteristic information for each spectral data does not lie in one area. Each spectrum has a different combinations of sequency regions which represents the characteristic feature of that spectrum. Therefore, the eliminations of either 5%, 1% or even 0.6% does not enhance the searching ability for the unknowns. To generate an optimally compressed data base, all the lower sequency data need to be included.

H. SEARCH TIME AND COMPRESSION RATES

Different sizes of compressed (clipped) data bases were subjected to a prefilter process and the search times of these various data bases were measured. Search time is the time it takes to compare an unknown with the clipped data bases. The data bases utilized for this study are 100%, 25%, 21%, 15%, 10%, and 5% clipped data bases.

Figure 24 shows the results of this study. The result

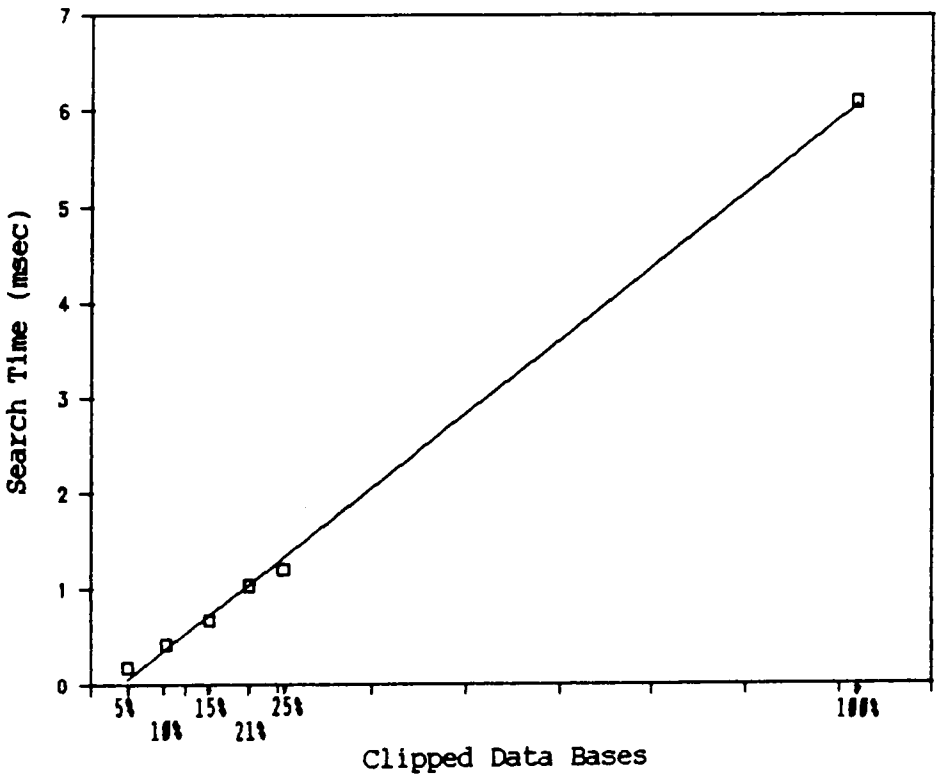


Figure 24. Search Time vs. Clipped Data Bases

indicates that the search time and the compression rate are lineary related with a correlation of Ø.998. The compression of the data set clearly minimizes the search time neccesary for the identification of unknowns.

I. APPLICATION OF THE SEARCH SYSTEM TO A DIFFERENT FLUORESCENCE SPECTROPHOTOMETER

It is important to note that a preliminary pilot study of this research was conducted using a Fluorispec fluorescence spectrophotometer, Model SF-1ØØ, by Baird-Atomic.

Table 39 shows the results of the study which examines the uniqueness of binary patterns at the prefilter process. Also Table 4Ø shows the comparison of the Baird-Atomic fluorescence spectrophotometer with the Perkin-Elmer's MPF-66. As a pilot study, five aromatic nuclear compounds were examined. "Average Search Results", indicates the average match rate of the 2nd through 4th hit compounds.
The total average of the "Average Search Results" is 5Ø.9%. As shown in Table 4Ø, although the signal to noise ratio of Baird-Atomic Fluoresence spectrophotometer is 1Ø, the results of the pilot study were equally successful. The library search system developed is applicable to different types of fluorescence spectrophotometers.

Table 39. Prefilter Search Results Using Baird-Atomic Fluorescence Spectrophotometer

Name of the 1st hit	Average Search results(%)
Anthracene	51.1
Chrysend	51.5
Pyrene	50.8
Tetracene	50.2
2-Naphthol	50.8
Average	50.9

Table 40. Comparison of Fluorescence Spectrophotometer between Baird-Atomic Model SF-100 and Perkin-Elmer MPF-66

	Resolution	S/N
B-A SF-100	2 nm	10
P-E MPF-66	0.30 nm	100

VI. CONCLUSION

The purpose of this research was to develop new data compression techniques and utilize them in construction of an efficient library search system for three dimensional fluorescence spectroscopy.

The actual comparison among unknown and reference spectra took place in two steps. The first step is called a "prefilter" and uses a "clipped data base". This step eliminates the most unlikely candidates from the reference spectral library and chooses only a few reference compounds which are the most likely candidates for the unknown. This procedure, which involves the entire library, is quick and yet accurate. The next step is to use a "filtered Hadamard transform data base". This involves a much larger file for each candidate, but only a few preselected compounds are involved.

The performance of the prefilter process was examined. The first search probe was performed using one of the spectra from the reference library as an unknown. For all cases, the system correctly identified the unknown and the correct reference spectrum was the only one with a 100% match rate. The total average search result using a 25% clipped data base was 52.7%. The match criterion employed would give a match rate value of 50% for statistically

unrelated spectra. The results show that even after a considerable amount of compression, the binary pattern used still contains the characteristic information about each spectrum.

Next the experiments were repeated using 21 real unknowns. These unknowns were measured at totally different times than the corresponding library member. The results showed that in 85.7% of the cases, this prefilter process correctly identified unknowns. To resolve ambiguities within the hit set derived from application of the prefilter, a second stage comparison was applied to unknowns. The results indicate that in 93.8% of the cases, the application of the second stage comparison, using a 25% filtered data base, led to a correct identification.

Effects of two types of noise on the prefilter process were also investigated. When random noise was added to a spectrum, the search system was tolerant of noise up to S/N ratio of 10. If a broad three-dimensional gaussian peak was added to a spectrum, the results showed that the search system had a surprising tolerance for this type of noise.

The effects of solvent changes were also studied. If an unknown is non-polar, then the effect of the polarity of the solvent is minimal. If an unknown compound is polar, then the effect became significant. For a complete library, it is necessary to include reference spectra derived from a series of different polarity solvents.

Concentration effects were also investigated. The results show that the search system is somewhat tolerant of different concentration levels at lower concentrations (i.e., up to 5.00×10^{-4}M). At higher concentration levels (i.e., greater than $5.00 \times^{-4}$M) effects became significant. Again, it is necessary to include representative spectra for high concentration solutions and lower concentration solutions in a reference library.

The optimal compression rate and filtering rate were also investigated. For the second stage comparison, these showed that one could use a 6.25% filtered data base (i.e., a 16 by 16 data point matrix from the lower sequency region) for the comparison. For the prefilter process, the optimal compression rate was a 10% clipped data base, which is only 0.61% the size of the original data base.

In conclusion, new data compression techniques based on Hadamard transform and clipping techniques were successfully developed. The application of these techniques to a library search system for three dimensional fluorescence spectroscopy was successful. The overall library may be about 7% of the space required by the full original wavelength domain data set. Corresponding savings in time are also found.

Finally, like all search system this computer supported library search system should not be used as an ultimate tool for the identification of unknown samples. It should rather

be utilized as an aid for chemists to minimize the time which is required for the identication of unknowns, eliminating many of the library members which are inappropriate.

VII. REFERENCES

1. Crawford, L.R.; Morrison, J.D. *Anal. Chem.* 1968, 40, 1464-1469.

2. Delaney, M.F.; Warren, F.V.Jr.,; Hallowell, J.R.Jr. *Anal. Chem.* 1983, 55, 1925-1929.

3. Hertz, H.S.; Hites, R.A.; Biemann, K. *Anal. Chem.* 1971, 43, 681-691.

4. Rasmussen, G.T.; Isenhour, T.L. *J. Chem. Inf. Comput. Sci.* 1979, 19, 178-185.

5. McLafferty, F.W.; Hertel, R.H.; Villwock, R.D. *Org. Mass Spectrom.* 1974, 9, 690-702.

6. Pesyna, G.M.; Venkataraghavan, R.; Dayringer, H.E.; McLafferty, F.W. *Anal. Chem.* 1976, 48, 1362-1368.

7. Atwater (Fell), B.L.; Venkataraghavan, R.; Mclafferty, F.W. *Anal. Chem.* 1979, 51, 1945-1949.

8. McLafferty, F.W.; Stauffer, D.B. *J. Chem. Inf. Comput. Sci.* 1985, 25, 245-252.

9. Ellis, R.D.; Peterson, D.W. *Anal. Chem.* 1985, 57, 771-773.

10. Atwater, B.L.; Stauffer, D.B.; McLafferty, F.W.; Peterson, D.W. *Anal. Chem.* 1985, 57, 899-903.

11. Stauffer, D.B.; McLafferty, F.W.; Ellis, R.D.; Peterson, D.W. *Anal. Chem.* 1985, 57, 1056-1060.

12. Freund, J.E. "Mathematical Statistics"; Prentice-Hall: Englewood Cliffs, N.J., 1962, p 46.

13. Lowry, S.R.; Huppler, D.A.; Anderson, C.R. *J. Chem. Inf. Comput. Sci.* 1985, 25, 235-241.

14. Kawata, S; Noda, T.; Minami, S. *Appl. Spectrosc.* 1987, 41, 1176-1182.

15. Grotch S.L. *Anal. Chem.* 1970, 42, 1214-1222.

16. Grotch S.L. *Anal. Chem.* 1971, 43, 1362-1370.

17. Grotch S.L. *Anal. Chem.* 1973, 45, 2-6.

18. Warren, F.V. Jr.; Delaney, M.F. *Appl. Spectrosc.* 1983, 37, 172-181.

19. Isenhour, T.L.; Foulk, S.J. *Anal. Chem.* 1978, 50, 1679A.

20. Lam, R.B.; Foulk, S.J.; Isenhour, T.L. *Anal. Chem.* 1981, 53, 1679.

21. Hangac, G.; Wieboldt, R.C.; Lam, R.B.; Isenhour, T.L. *Appl. Spectrosc.* 1982, 36, 40-47.

22. Williams, S.S.; Lam, R.B.; Isenhour, T.L. *Anal. Chem.* 1983, 55, 1117-1121.

23. Harrington, P.B.; Isenhour, T.L. *Appl. Spectrosc.* 1987, 41, 449-453.

24. Andrews, H.C.; Pratt, W.K. *Proc. Symp. on Computer Processing in Communications* 1969, pp 63-84

25. Warner, I.M.; Callis, J.B.; Davidson, E.R.; Christian, G.D. *Clin. Chem.* 1976, 22, 1483-1492.

26. Johnson, D.W.; Callis, J.B.; Christian, G.D. *Anal. Chem.* 1977, 49, 747A-757A.

27. Rossi, T.M.; Warner, I.M. *Appl. Spectrosc.* 1984, 38, 422-429.

28. Rossi, T.M.; Warner, I.M. *Appl. Spectrosc.* 1985, 39, 949-959.

29. Rossi, T.M.; Warner, I.M. *Appl. Spectrosc.* 1986, 40, 44-52.

30. Vicsek, M.; Neal, S.L.; Warner, I.M. *Appl. Spectrosc.* 1986, 40, 542-548.

31. Pau, C.; Warner, I.M. *Appl. Spectrosc.* 1987,41,496-502.

32. Hrwit, M. In "Transform Techniques in Chemistry"; Griffiths, P.R. Ed.; Plenum Press: New York, London, 1978, pp 173-197.

33. Gollay, M.J.E. *J. Opt. Soc. Am.* 1949, 39, 437-444.

34. Gollay, M.J.E. *J. Opt. Soc. Am.* 1951, 41,468-472.

35. Harwit, M.; Phillips, P.G.; Fine, T.; Sloane, N.J.A. *Appl. Opt.* 1970, 9, 1149-1154.

36. Nelson. E.D.; Fredman, M.L.; *J. Opt. Soc. Am.* 1970, 60, 1664-1669.

37. Phillips, P.G.; Briotta, D.A. Jr. *Appl. Opt.* 1974, 13, 2233-2235.

38. Dillard, G.M. *Proc. Walsh Functions Symposium* 1970, 22-18 - 22-22.

39. Pratt, W.K.; Kane, J.; Andrews, H.C. *Proc. of the IEEE* 1969, 57, 58-68.

40. Ahmed, N.; Rao, K.R. "Orthogonal Transforms for Digital Signal Processing"; Springer-Verlag: New York, Heidelberg, Berlin, 1975, pp 85-152.

41. Ahmed, N.; Rao, K.R. ; Abdussattar, A.L. *IEEE Trans. Audio Electroacoust.* 1971, AU-19, 225-233.

42. Beauchamp, K.G. "Applications of Walsh and Related Functions with an Introduction to Sequency Theory"; Academic Press: London, 1984.

43. Guilbault, G.G. " Practical Fluorescence Theory, Methods, and Techniques"; Marcel Dekker, Inc.: New York, 1973.

44. Ellis, D.W.; Wehry, E.L.; Rogers, L.B. In "Fluorescence and Phosphorescence Analysis"; Hercules, D.M. Ed.; Interscience Publichers: New York, 1966.

45. Turro, N.J. "Modern Molecular Photochemistry"; The Benjamin/Cummings Publishing Co., Inc.: Menlo Park, 1978.

46. Schulman, S.G. In "Chemical Analysis"; Elving, P.J.; Winefordner, J.P. Ed.; John Wiley & Sons: New York, 1985, 77.

47. Lloyd, J.B.F. In "Standards in Fluorescence Spectrometry"; Miller J.N. Ed.; Chapman and Hall: New York, 1981.

48. Parker, C.A.; Rees, W.T. *Analyst*, 1962, 87, 83-111.

VIII. APPENDICES

APPENDIX A

Three-Dimensional Fluorescence Spectra of Polynuclear Aromatic Compounds

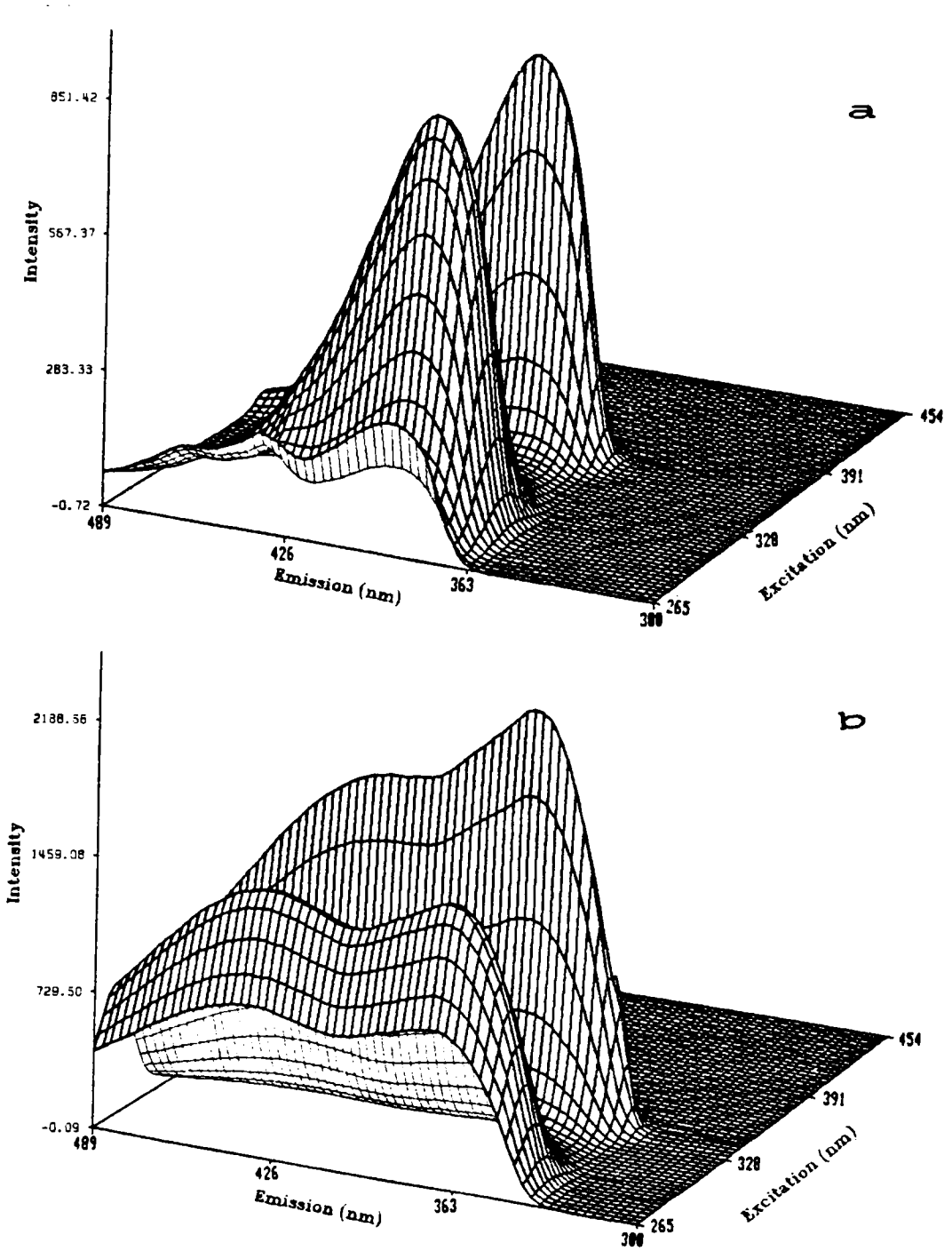


Figure A-1a. Anthranilic Acid
b. Salacilic Acid

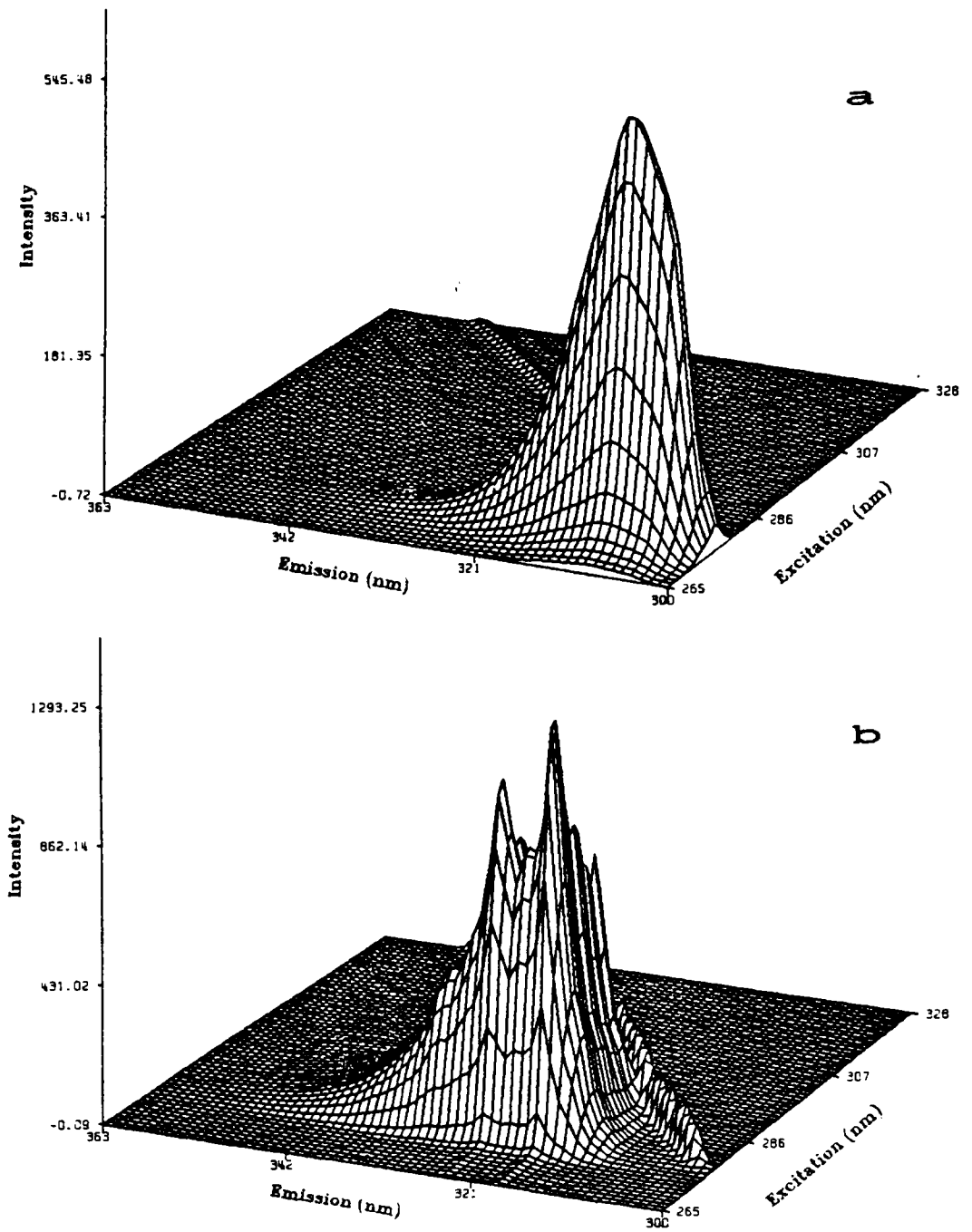


Figure A-2a. 4-Biphenylphenyl Ether
b. Triphenylamine

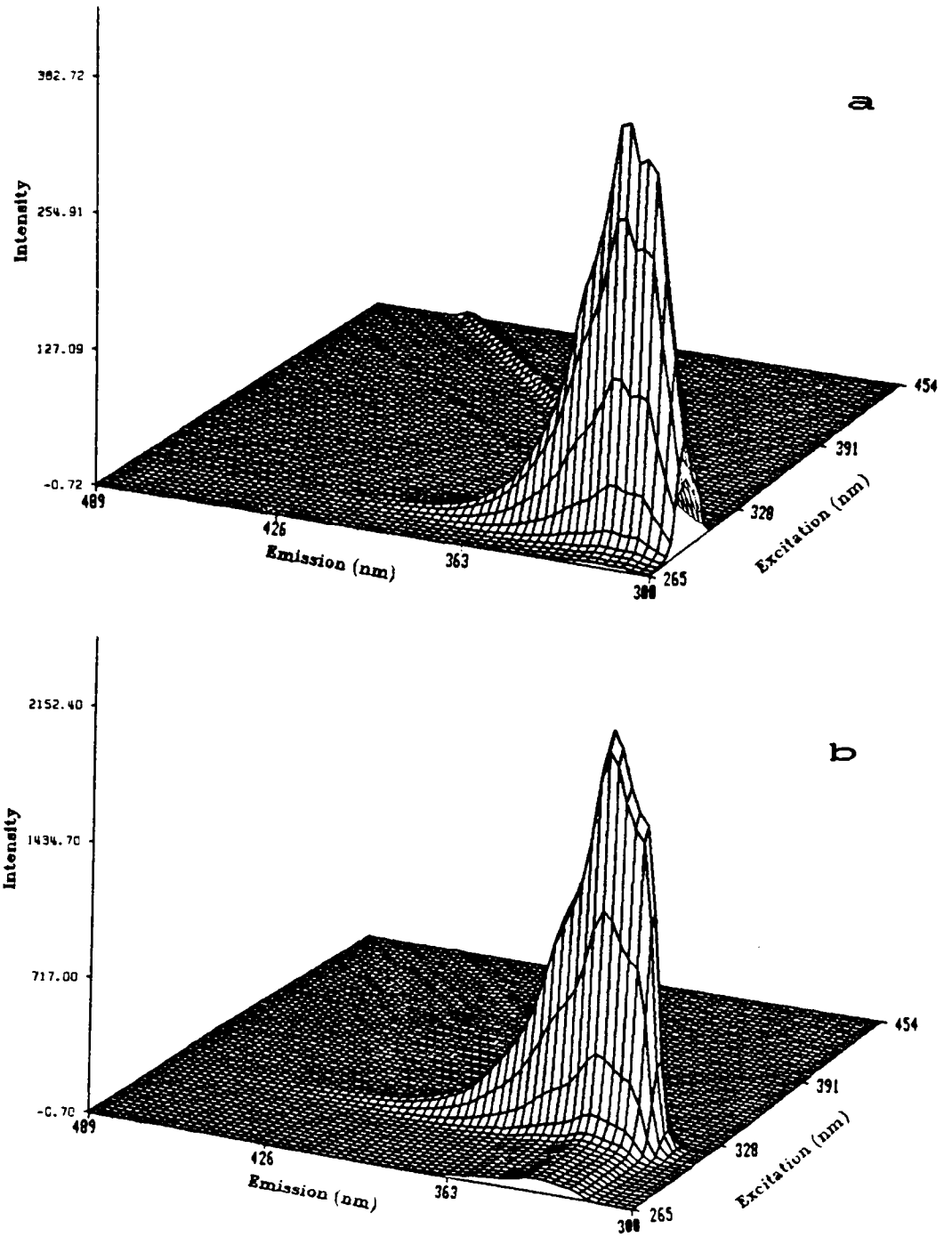


Figure A-3a. 4-Methylbiphenyl
b. 4-Vinylbiphenyl

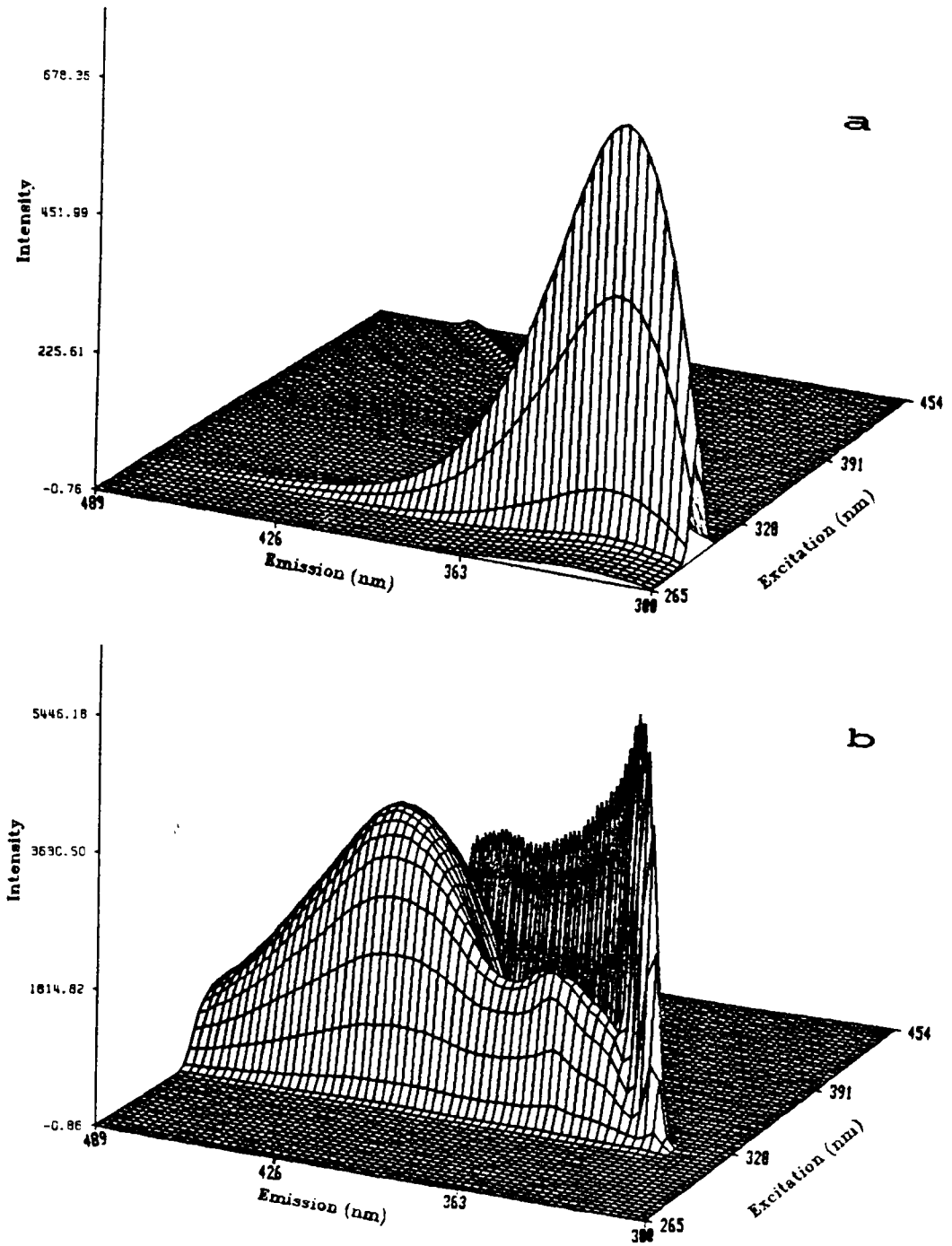


Figure A-4a. Indole
b. Quinoline

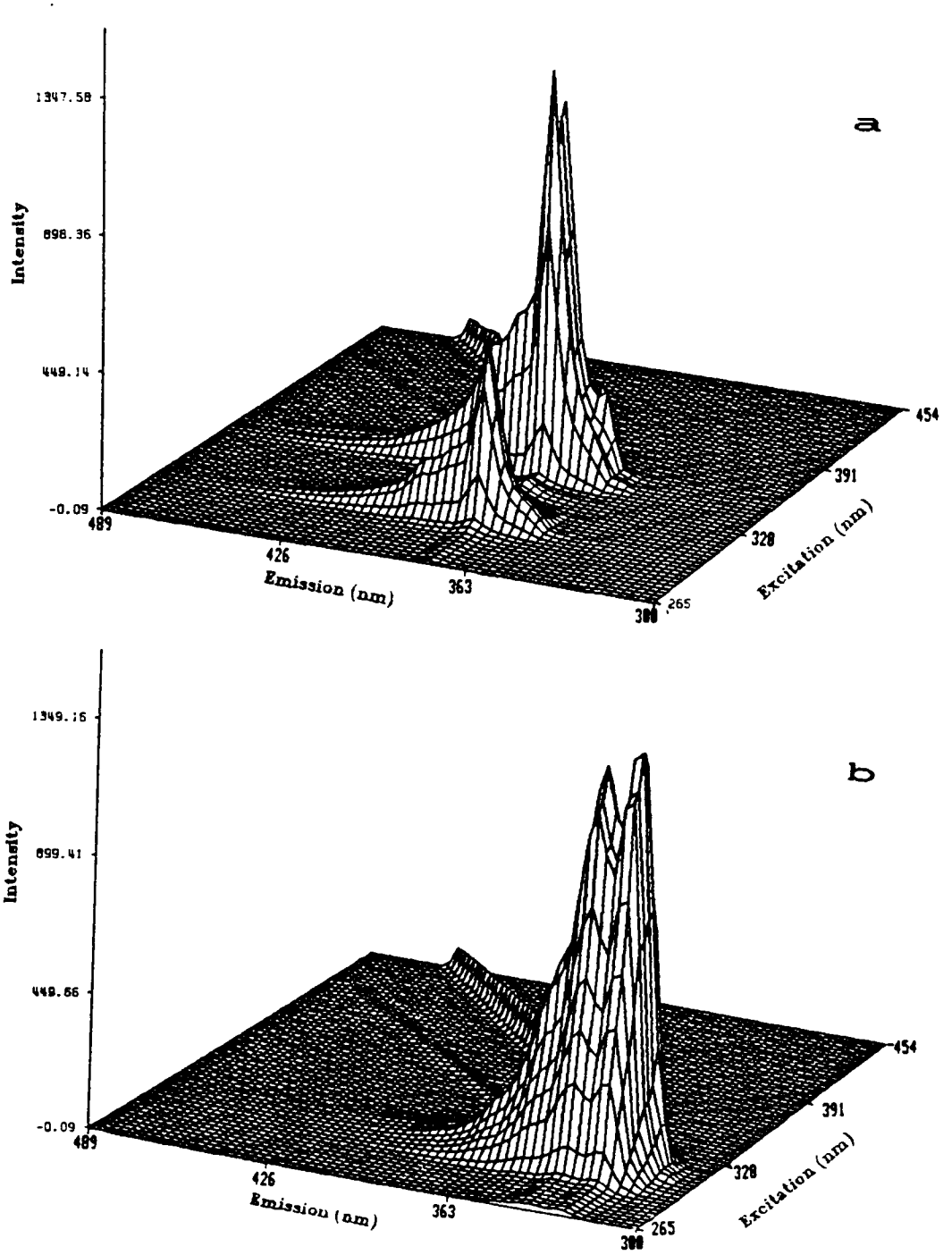


Figure A-5a. Azulene
b. Naphthalene

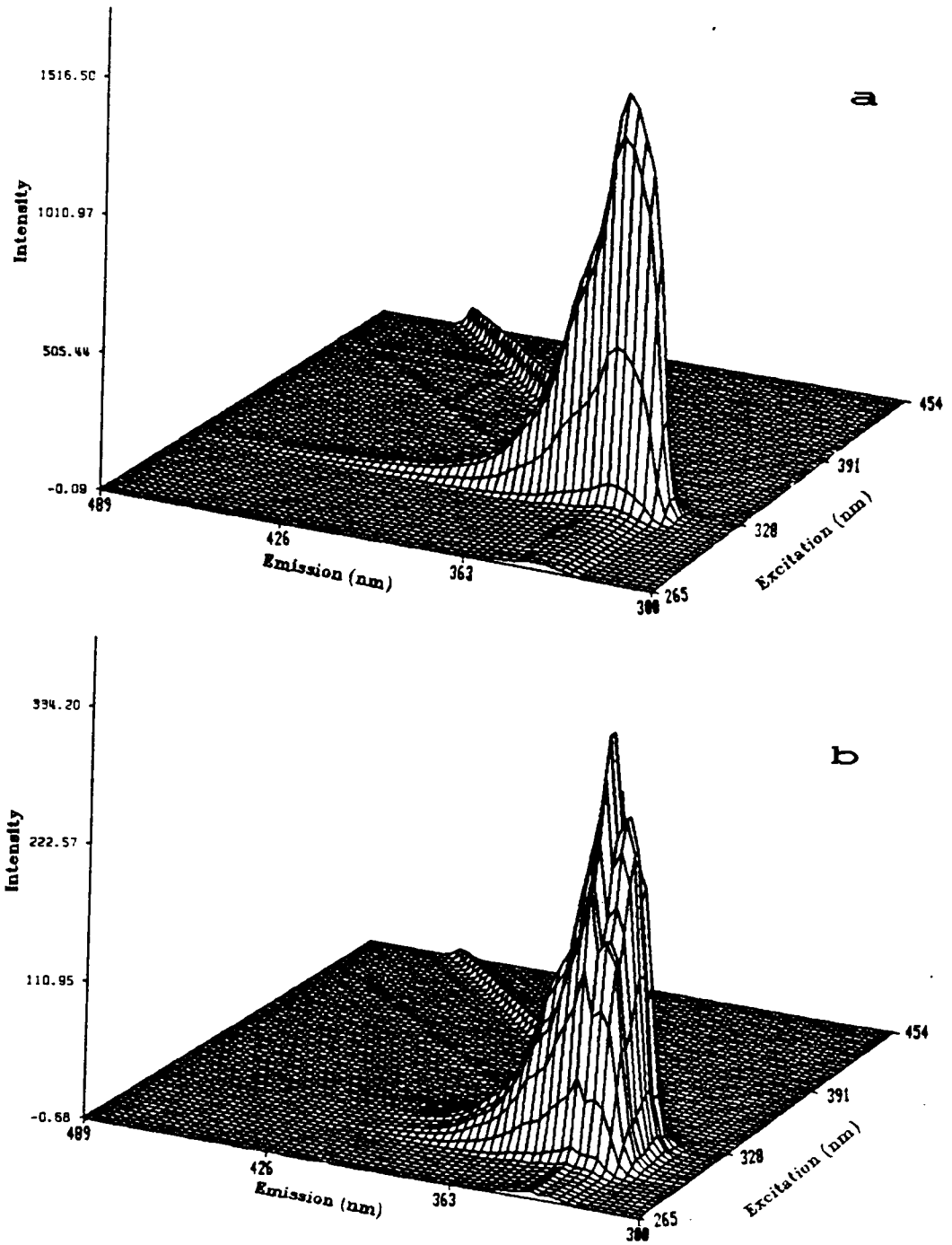


Figure A-6a. 1-Methylnaphthalene
b. 2-Methylnaphthalene

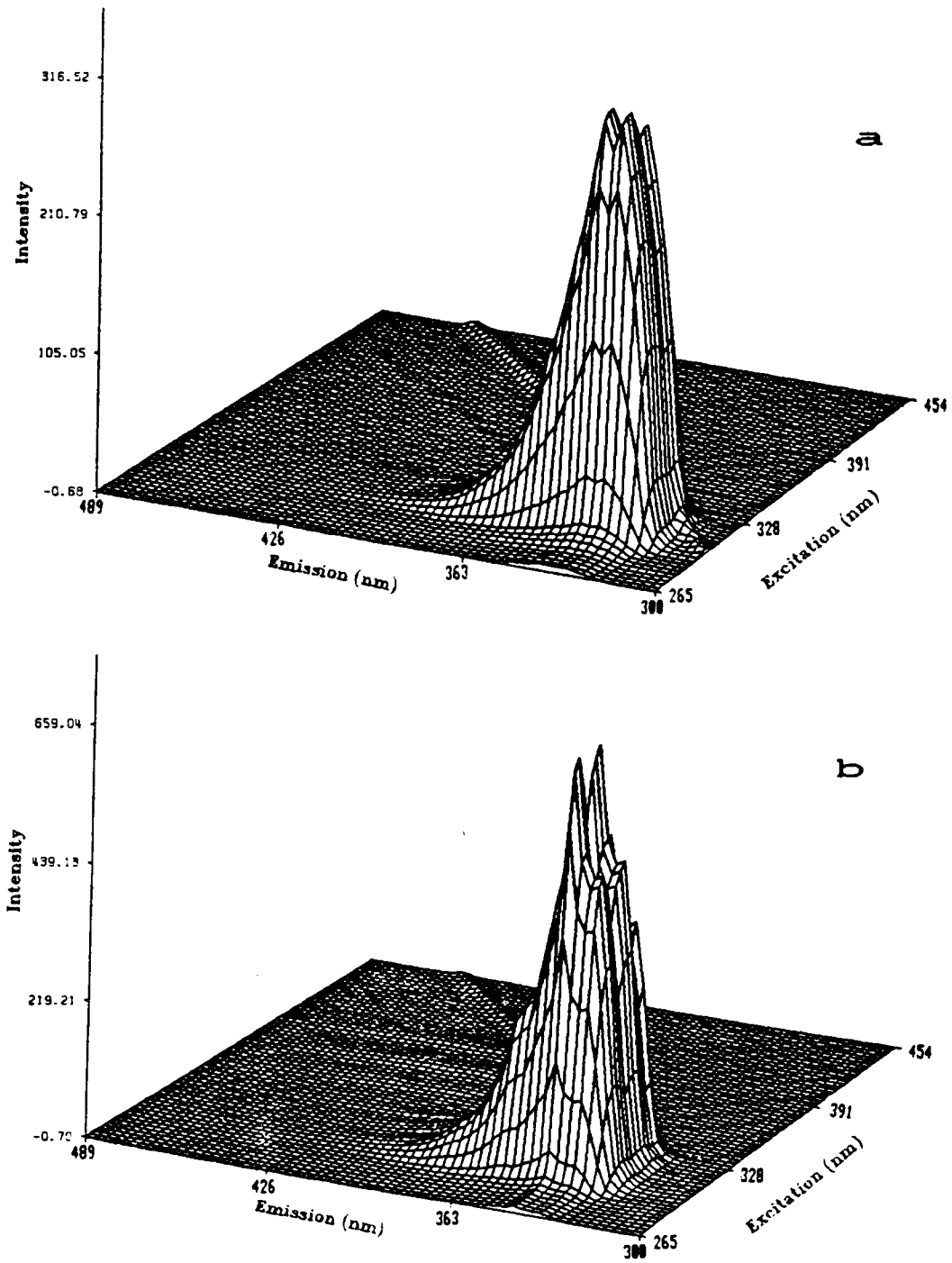


Figure A-7a. 2,3-Dimethylnaphthalene
b. 2,6-Dimethylnaphthalene

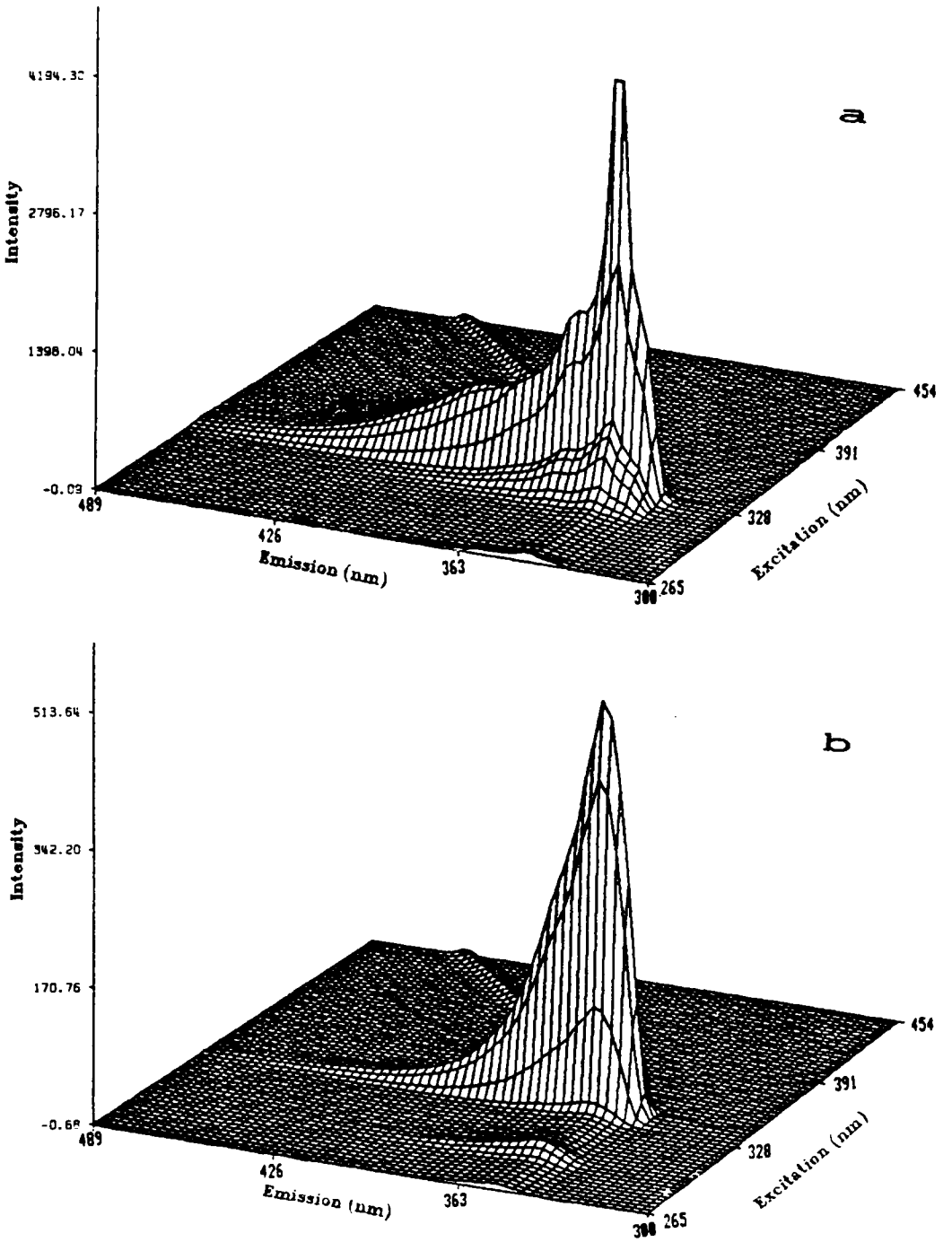


Figure A-8a. 1-Naphthol
b. 2-Naphthol

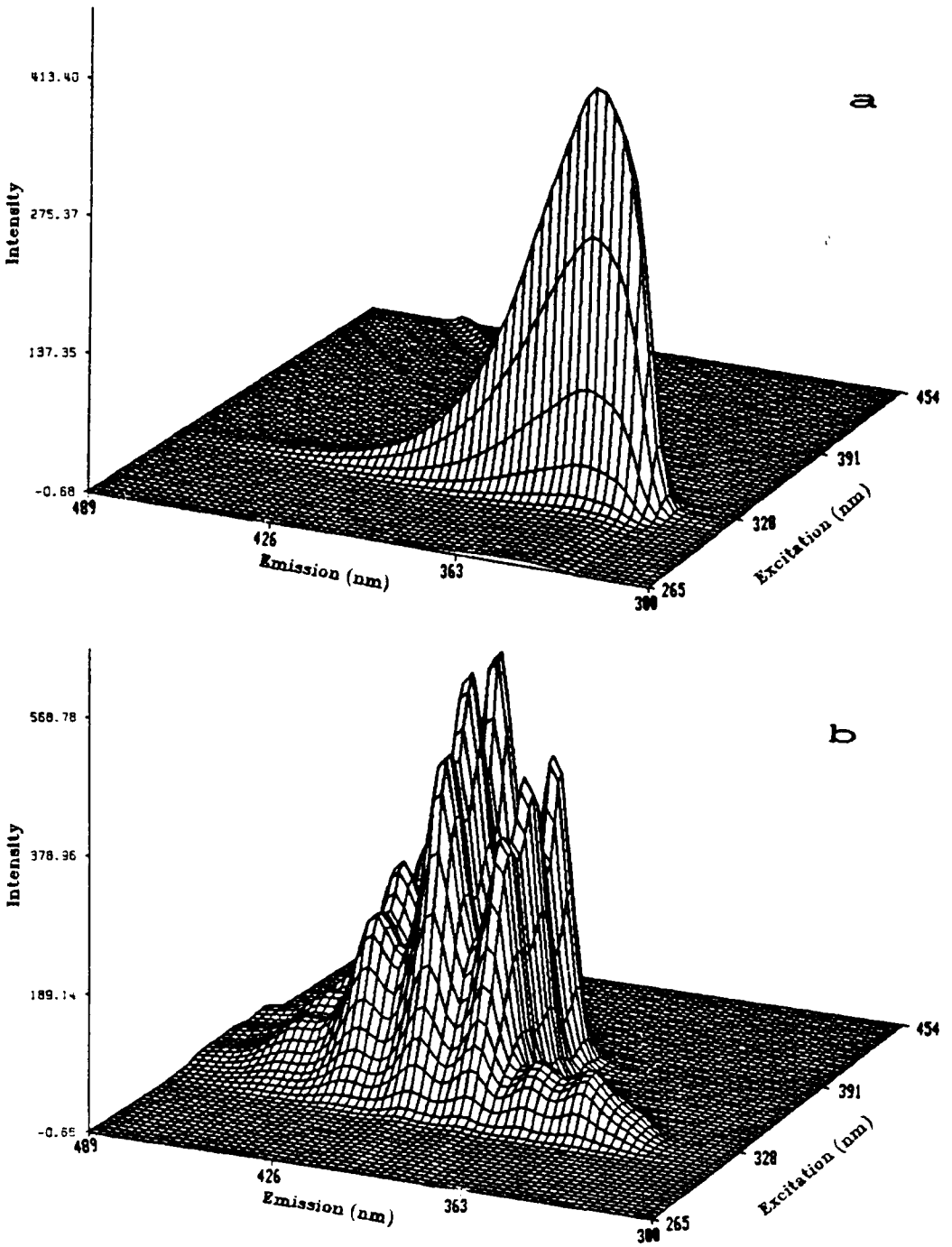


Figure A-9a. 1-Phenylnaphthalene
b. 2-Phenylnaphthalene

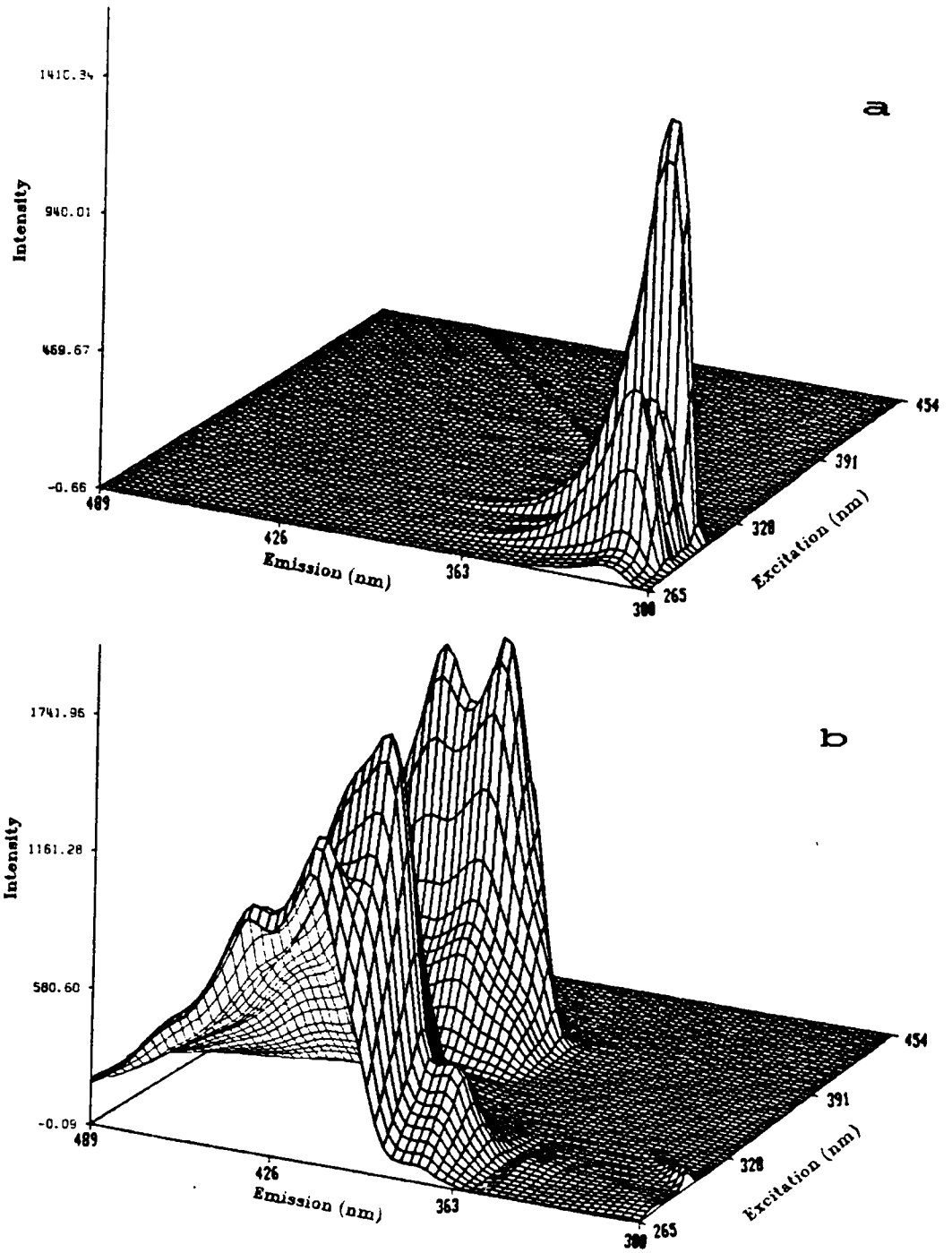


Figure A-10a. Fluorene
b. Acridine

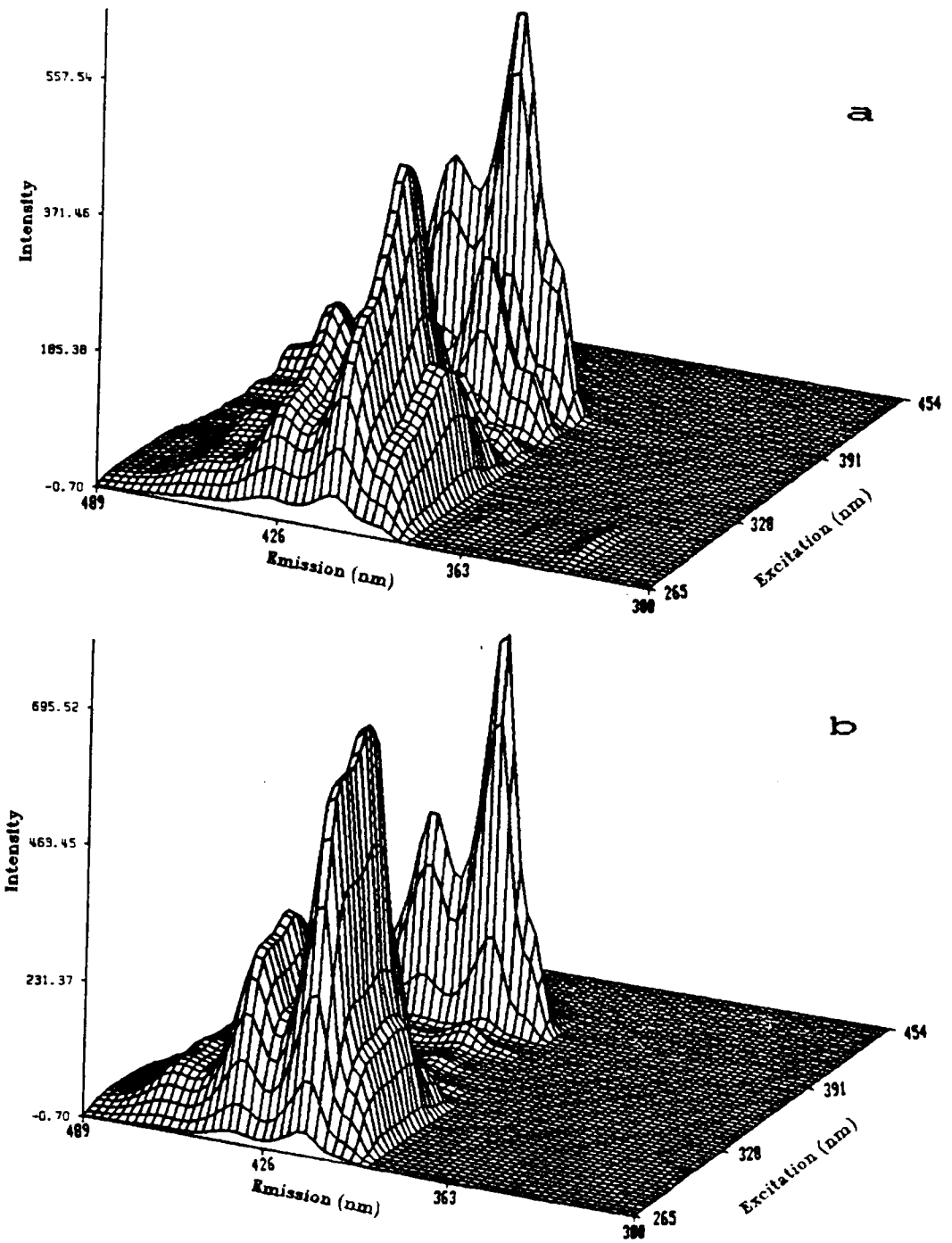


Figure A-11a. 2-Methylanthracene
b. 9-Methylanthracene

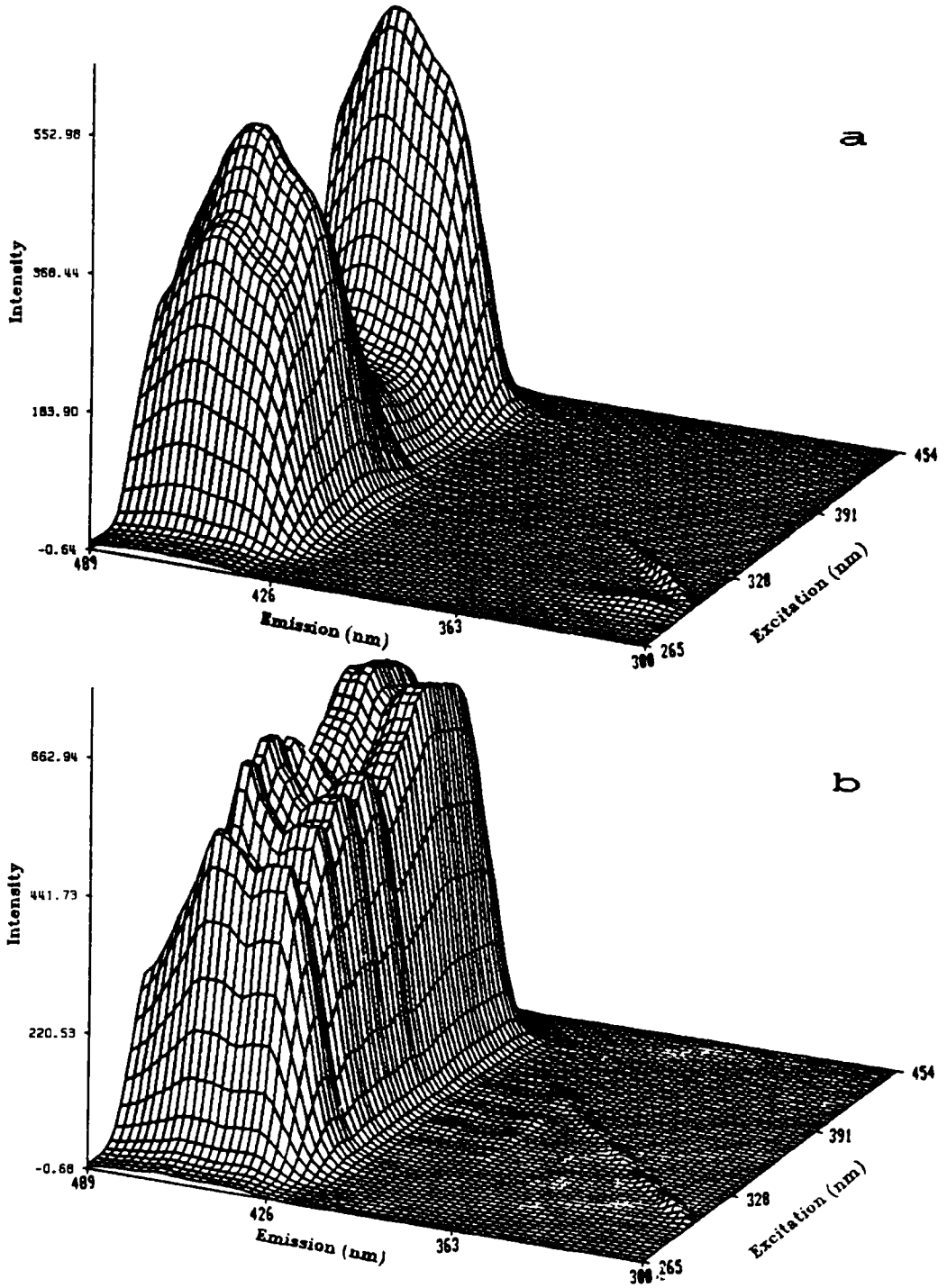


Figure A-12a. 1-Aminoanthracene
b. 2-Aminoanthracene

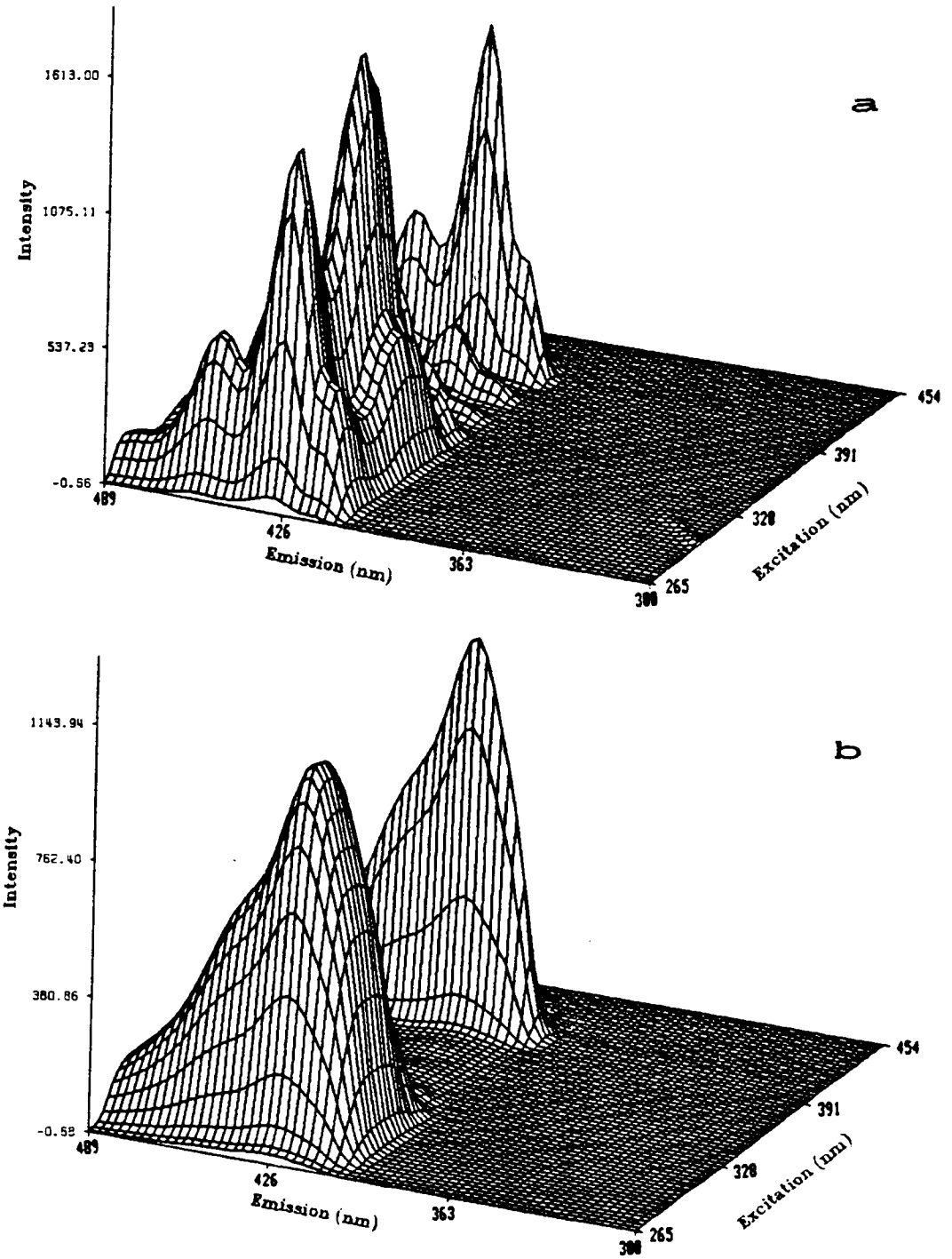


Figure A-13a. 9,10-Dichloroanthracene
b. 9-Vinylanthracene

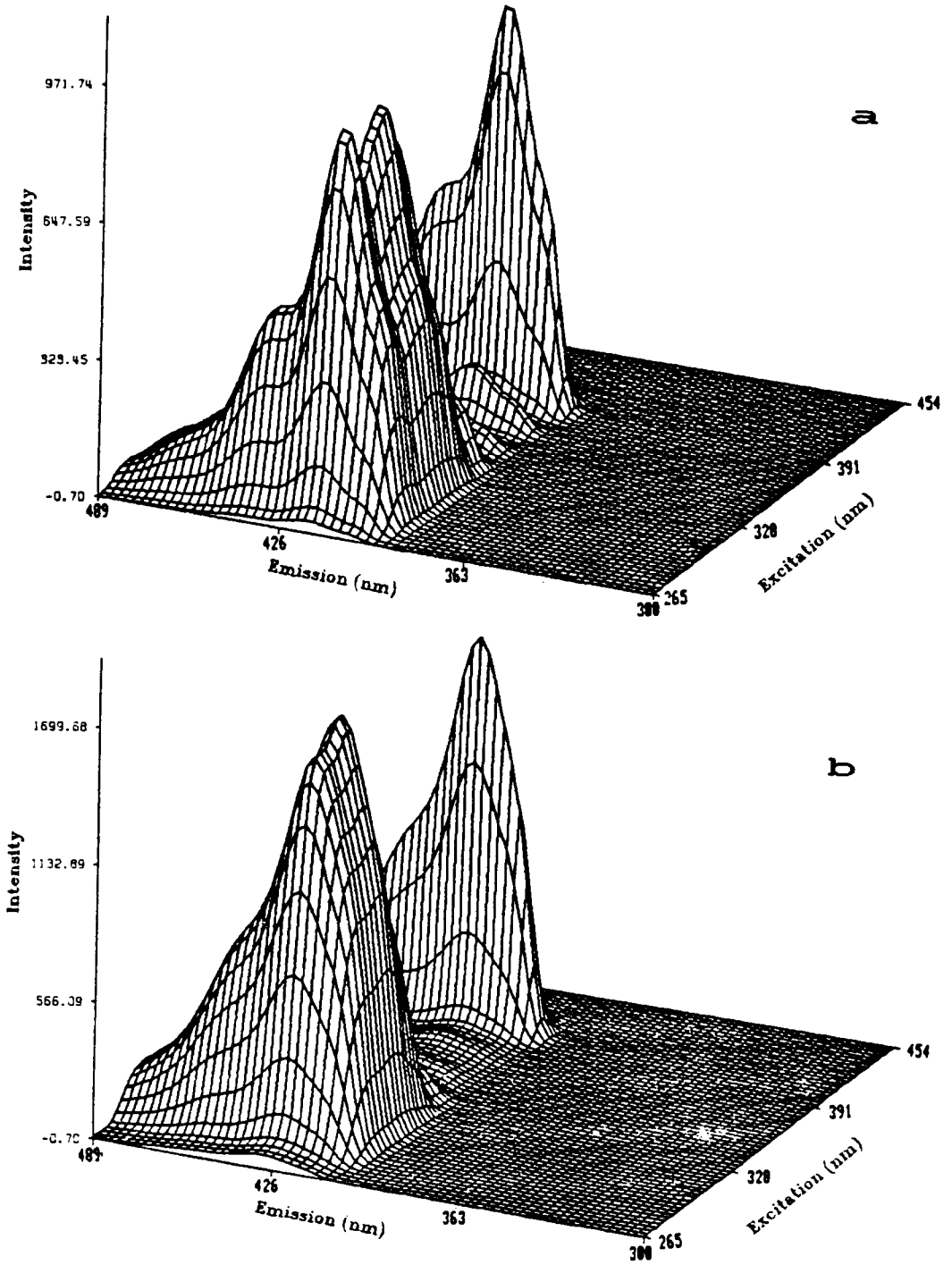


Figure A-14a. 9-Phenylanthracene
b. 9,10-Diphenylanthracene

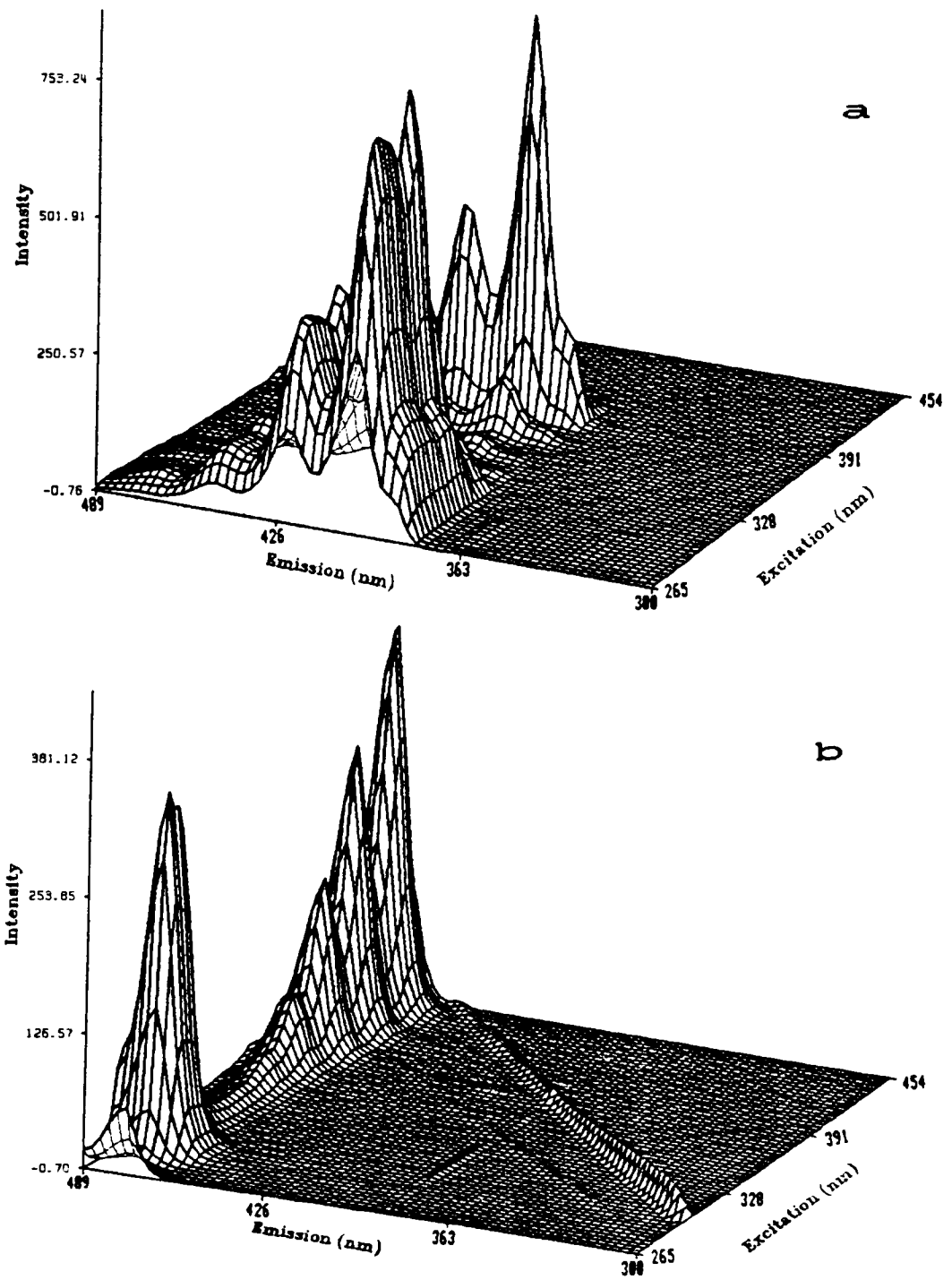


Figure A-15a. Anthracene
b. Tetracene

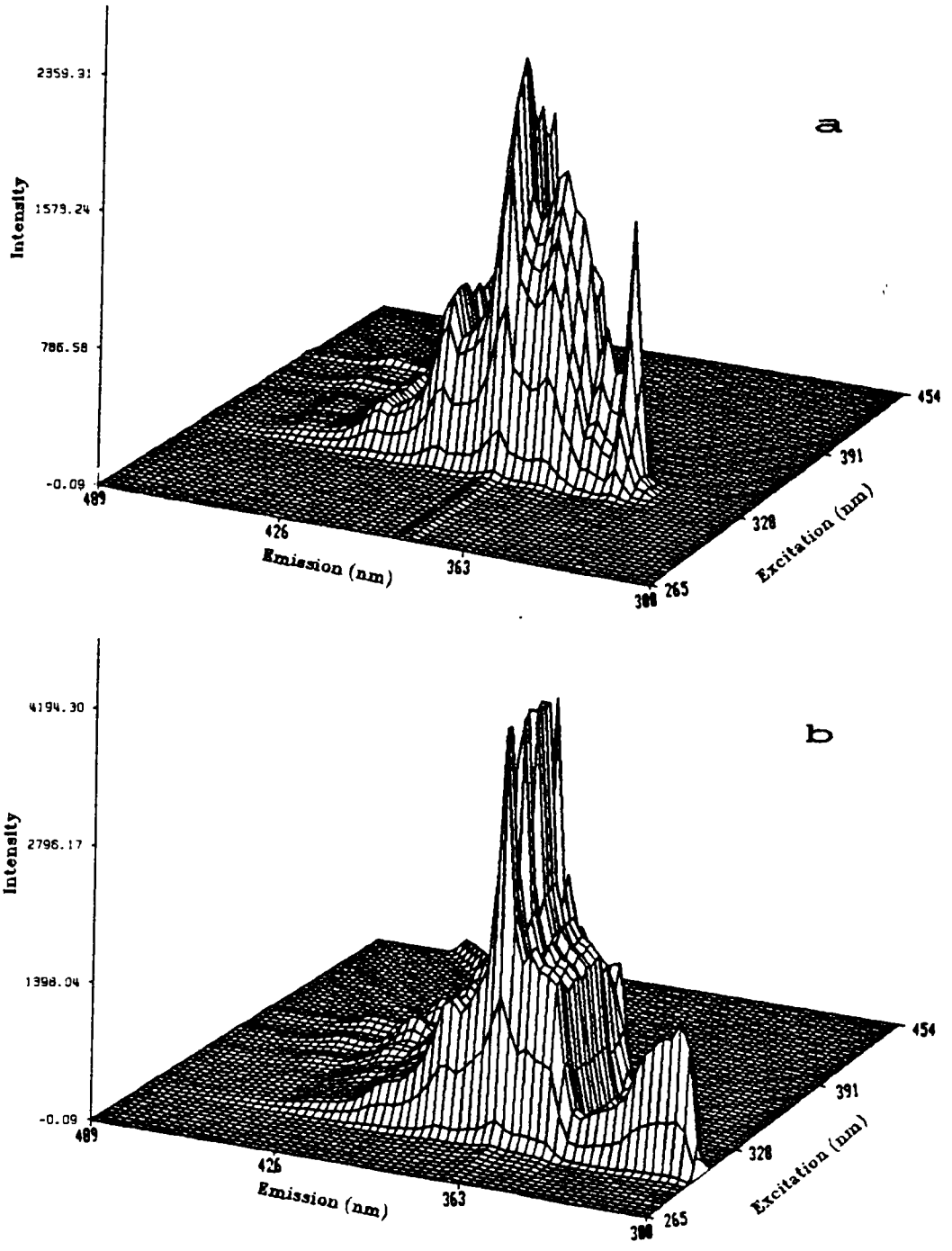


Figure A-16a. Chrysene
b. Phenanthrene

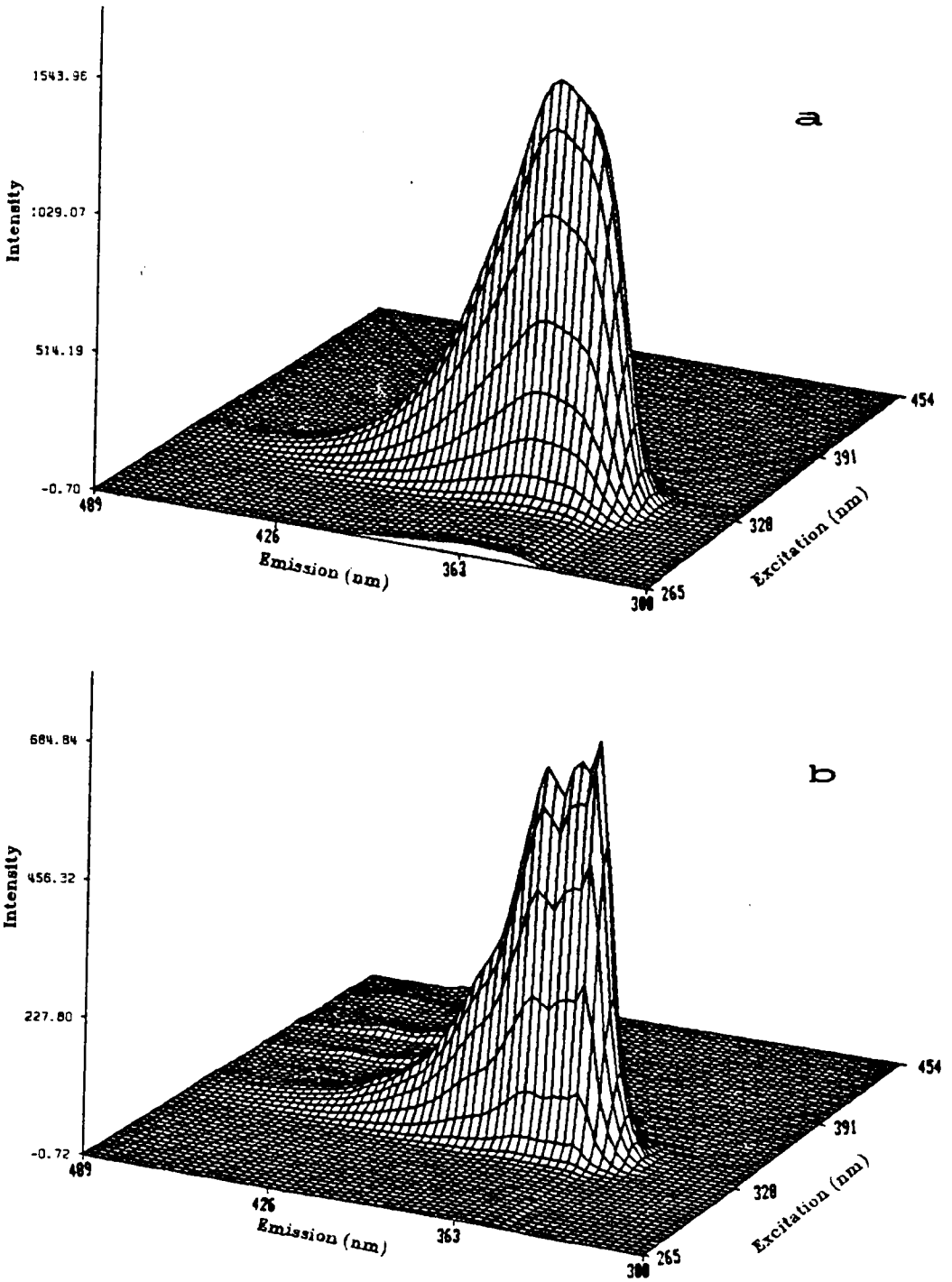


Figure A-17a. 1,1-Binaphthyl
b. 2,2-Binaphthyl

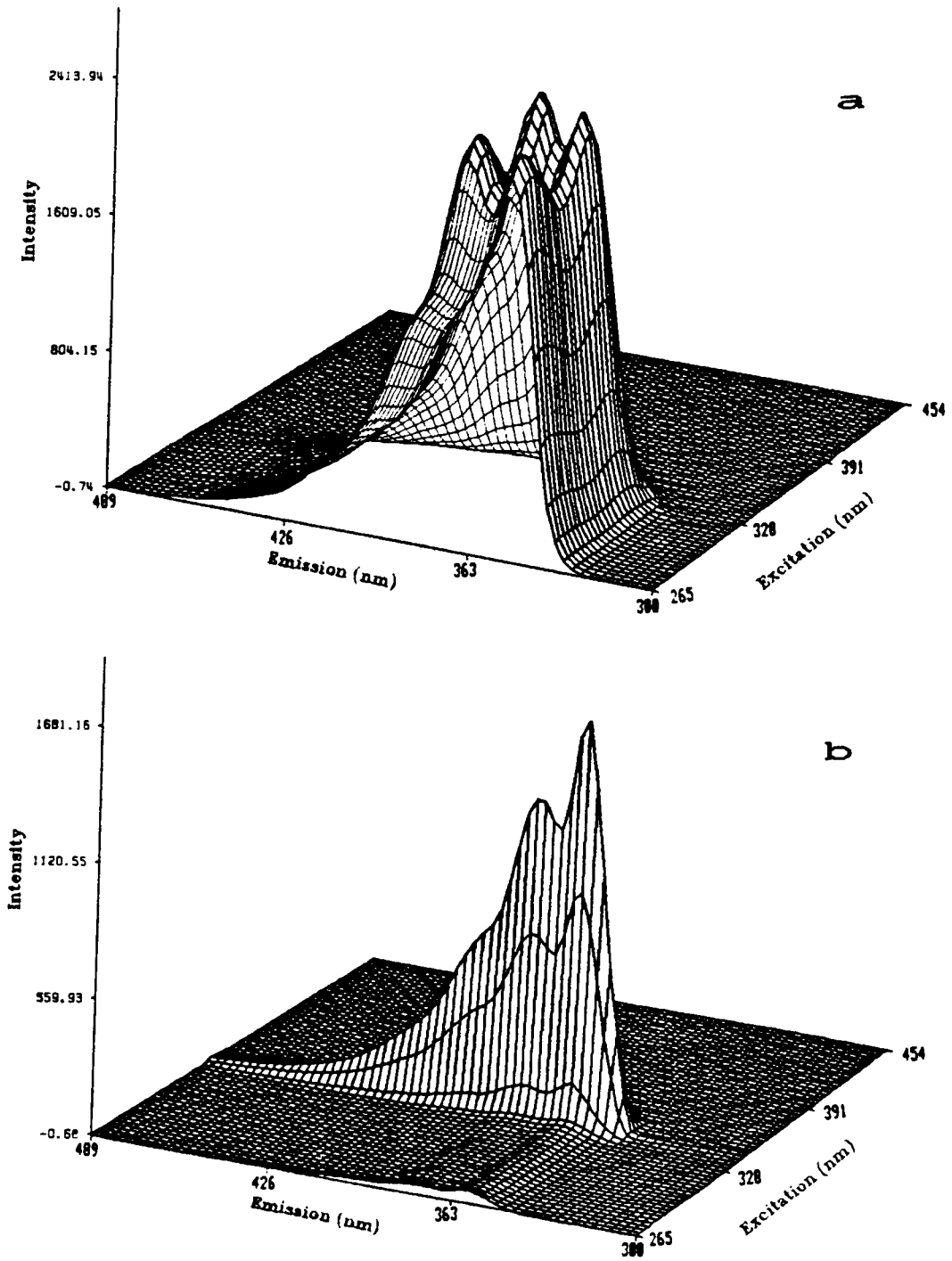


Figure A-18a. pQuaterphenyl
b. PPO

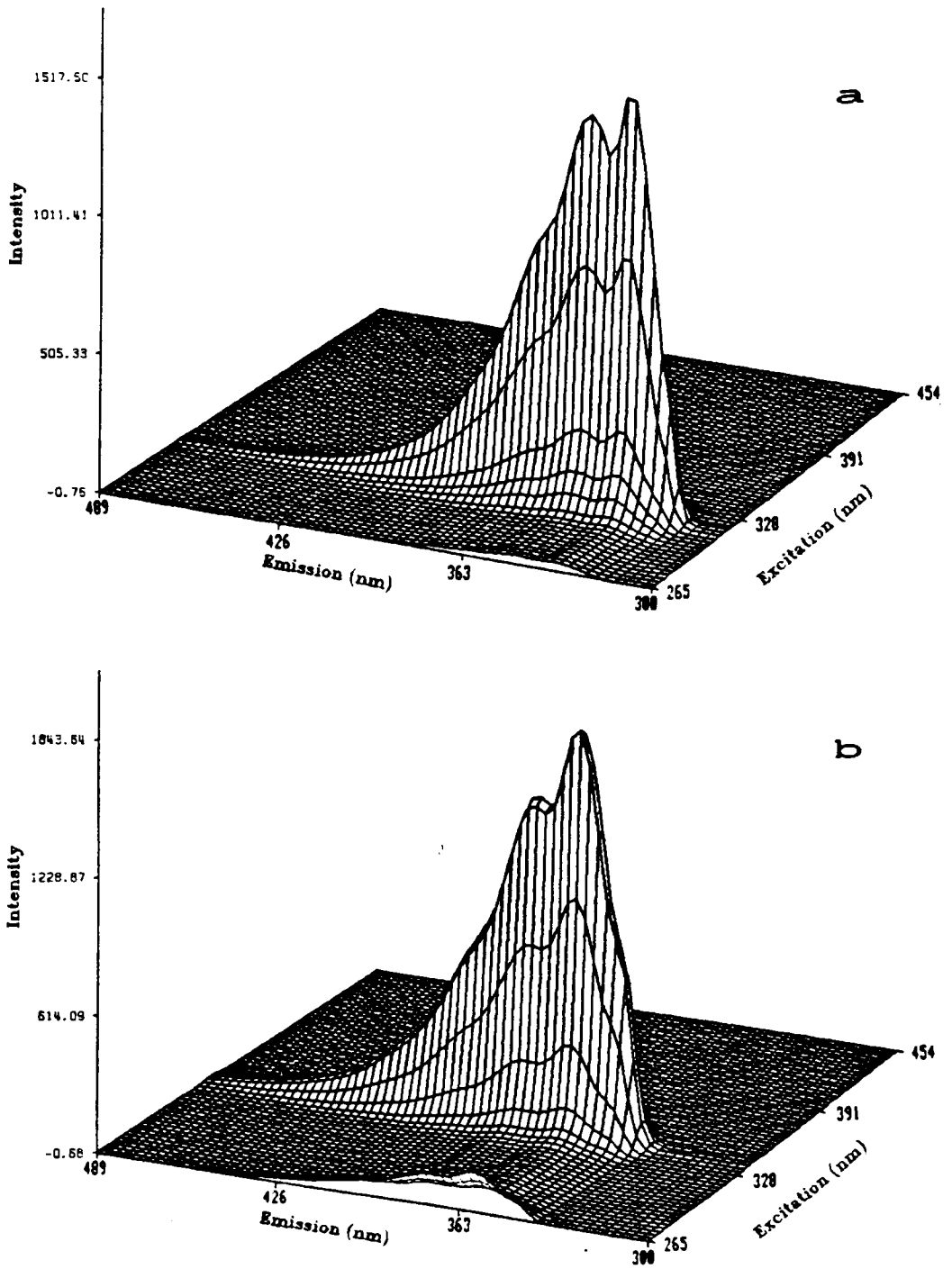


Figure A-19a. PPD
b. PBD

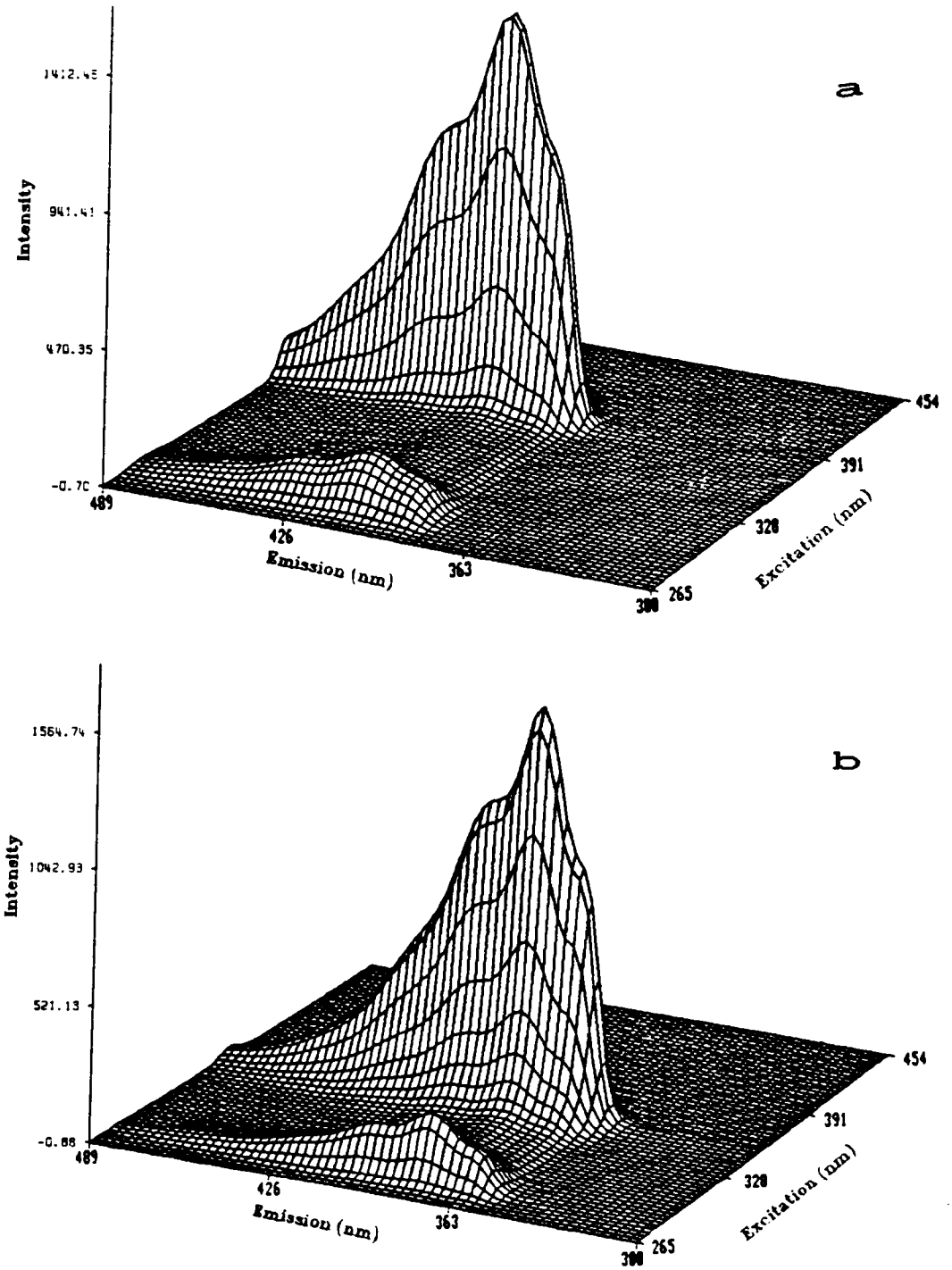


Figure A-20a. BBO
b. BBD

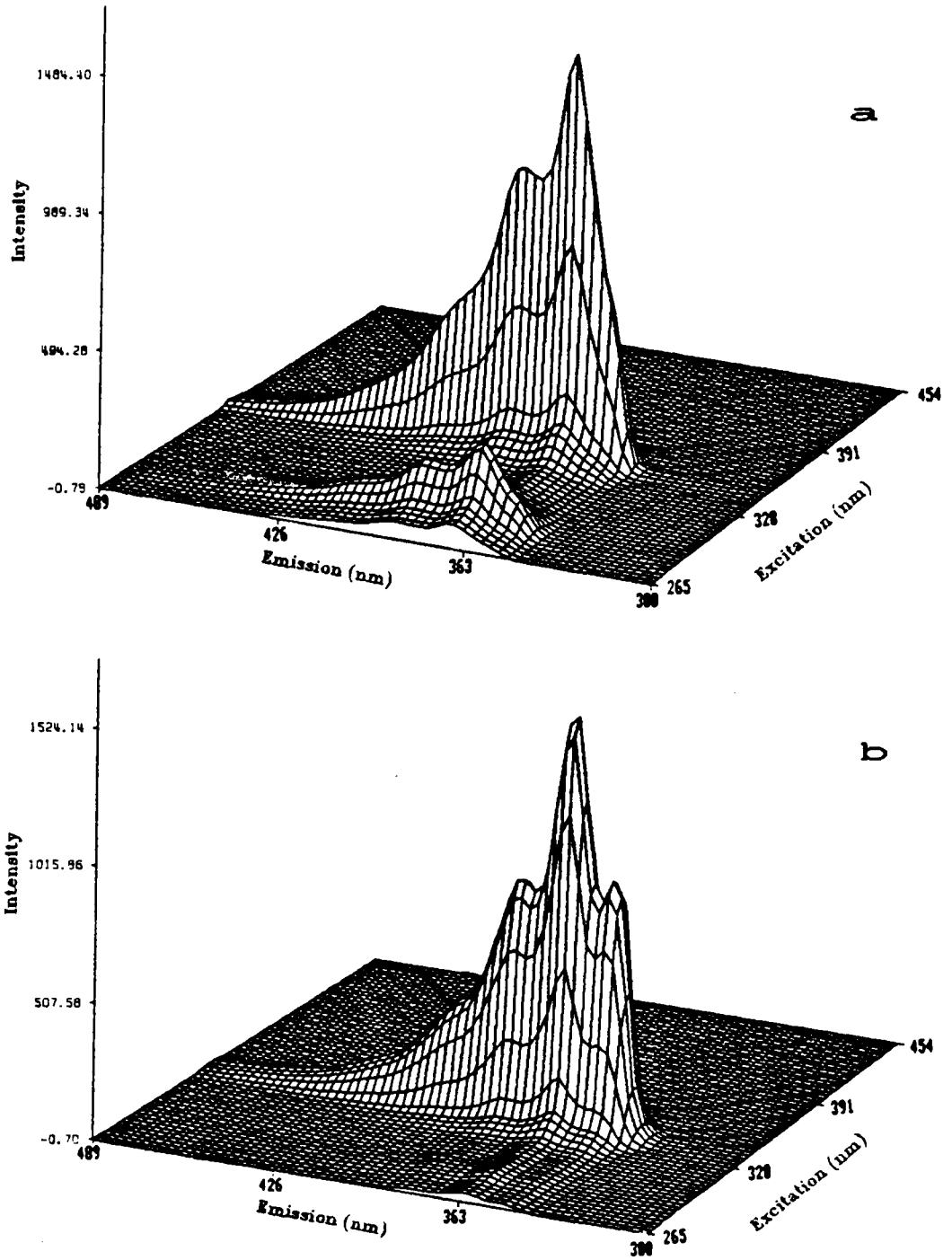


Figure A-21a. α-NPD
b. β-NPD

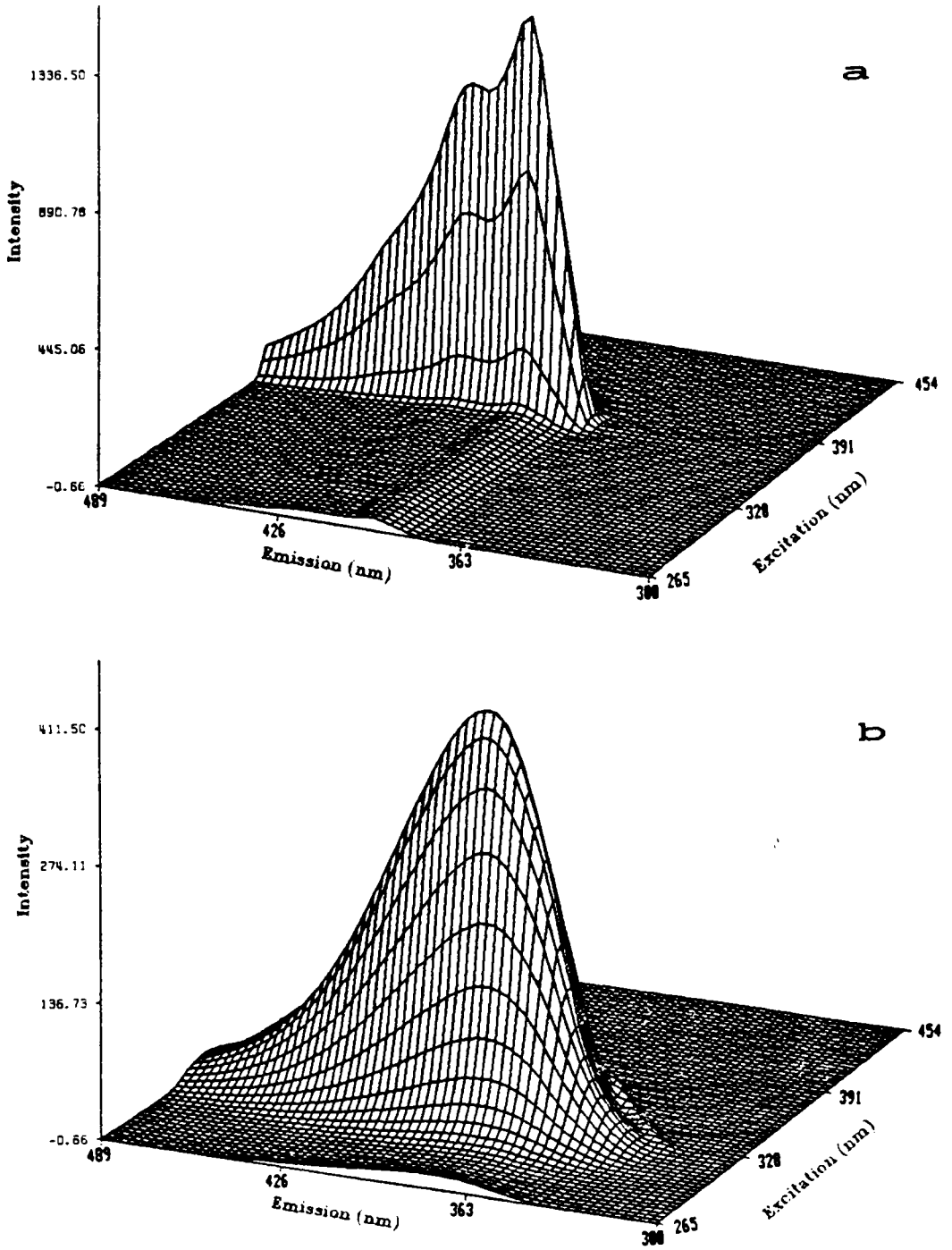


Figure A-22a. α-NPO
b. 4,5-Diphenylimidazule

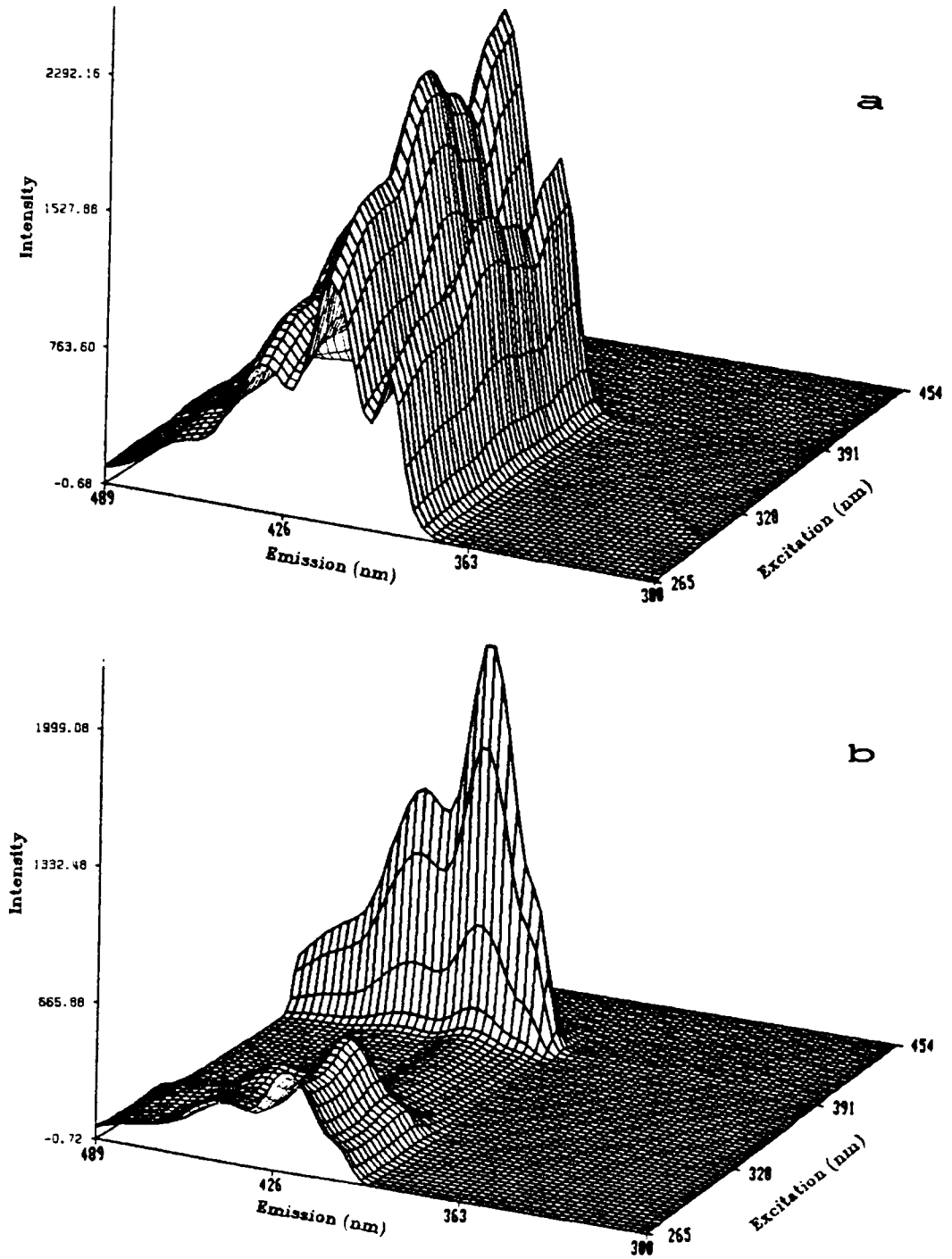


Figure A-23a. POPOP
b. Dimethyl-POPOP

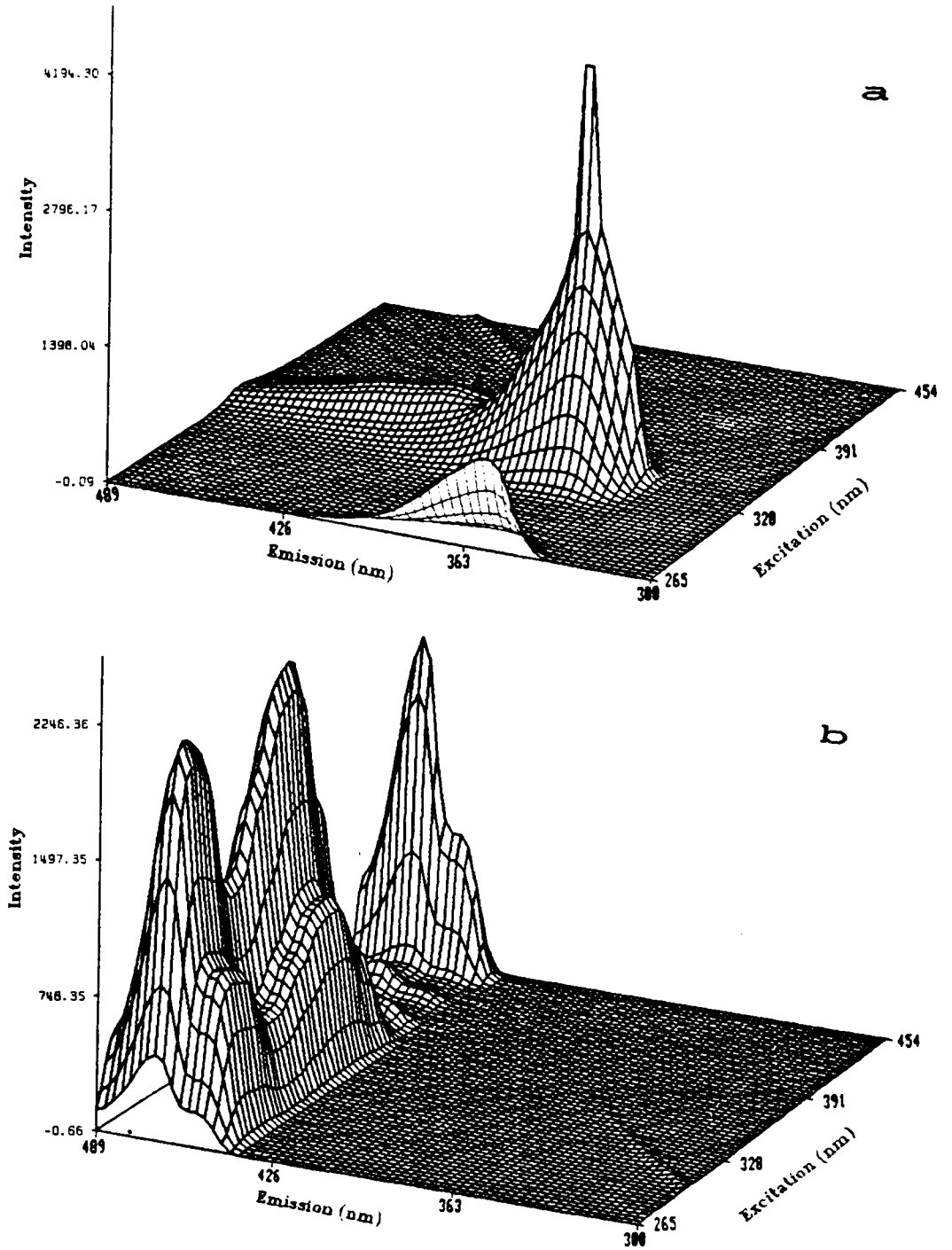


Figure A-24a. Triphenylene
b. Perylene

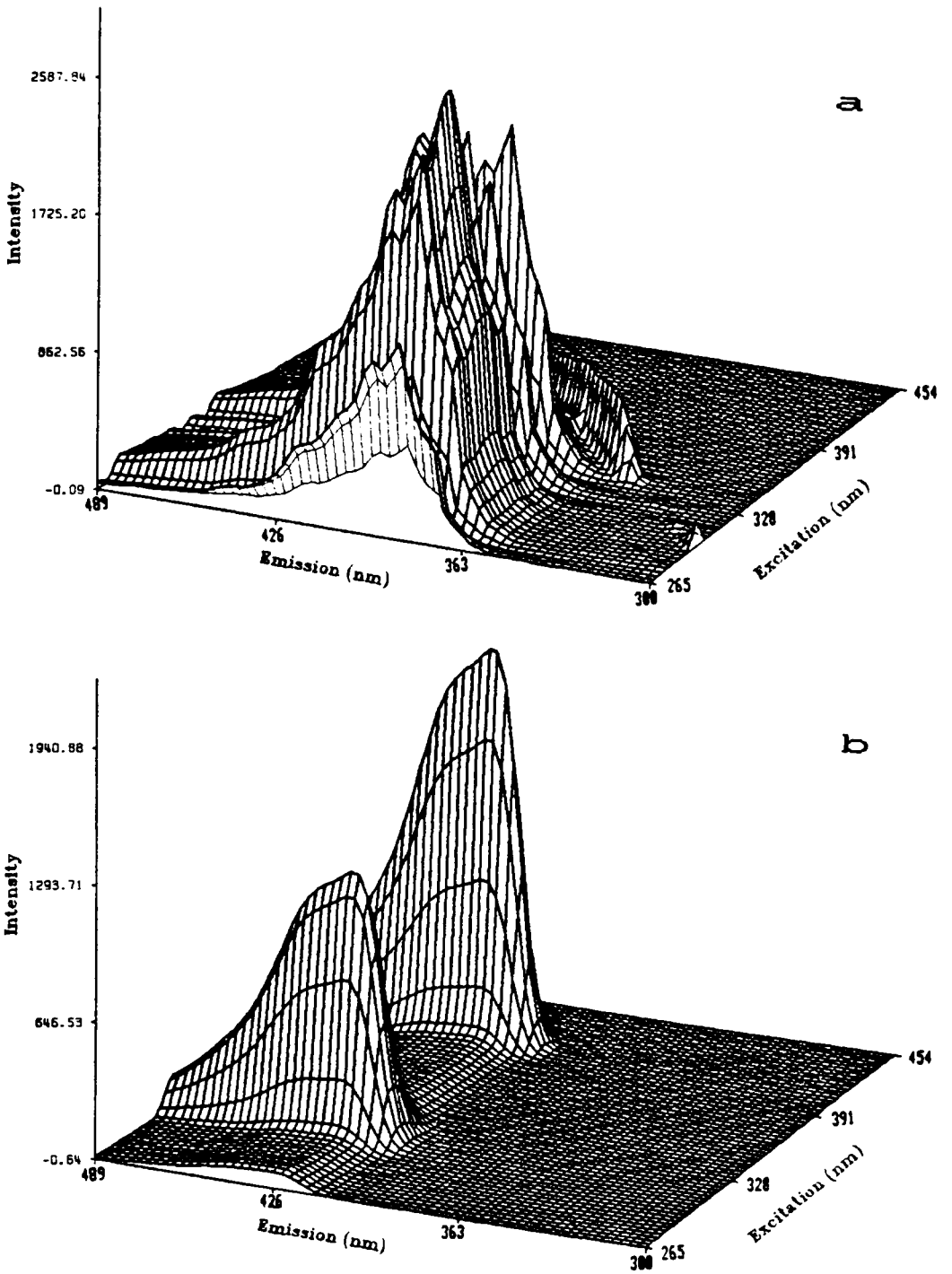


Figure A-25a. Pyrene
b. 1,3,6,8-Tetraphenylpyrene

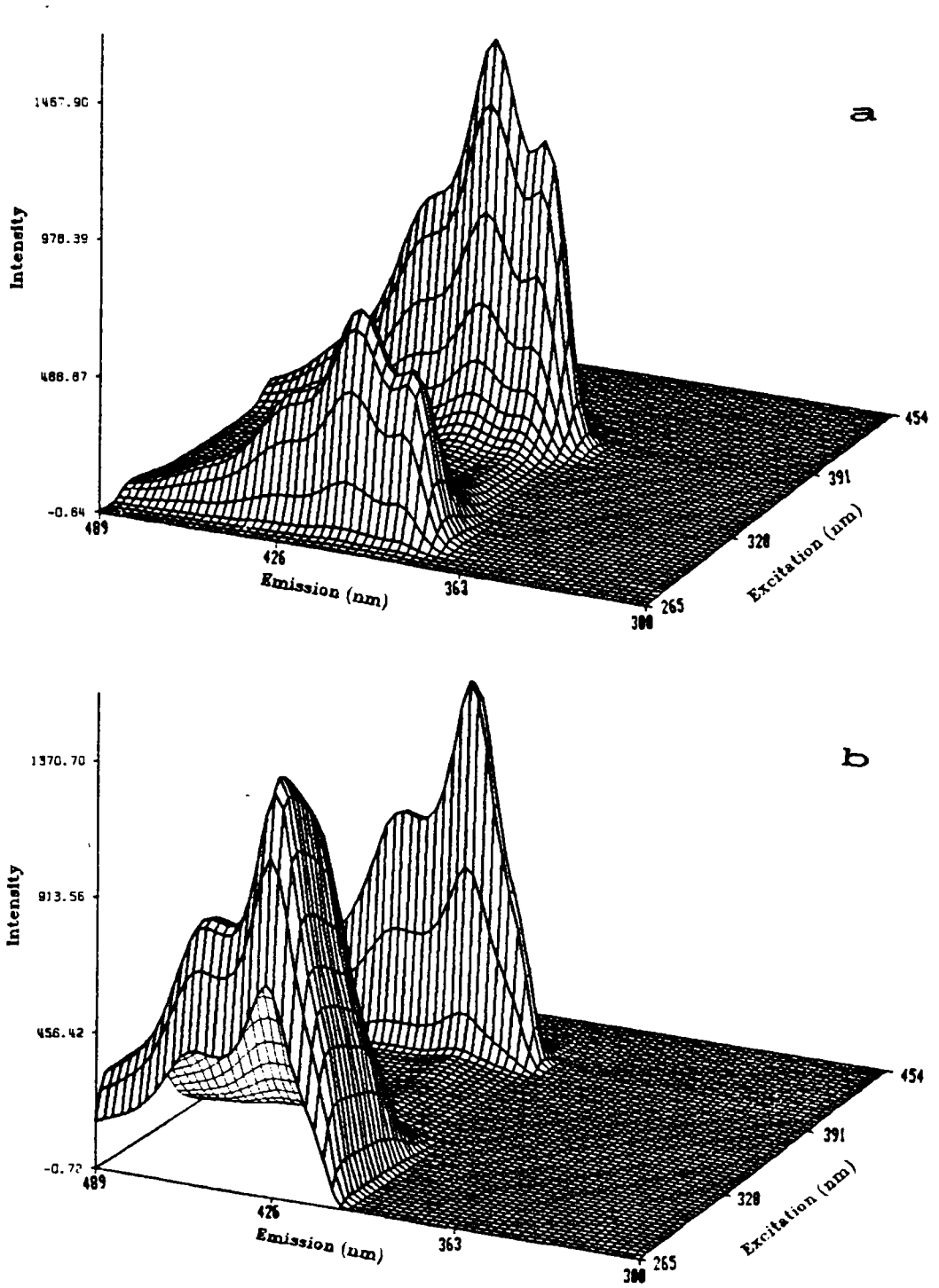


Figure A-26a. Diphenylstilbene
b. BBOT

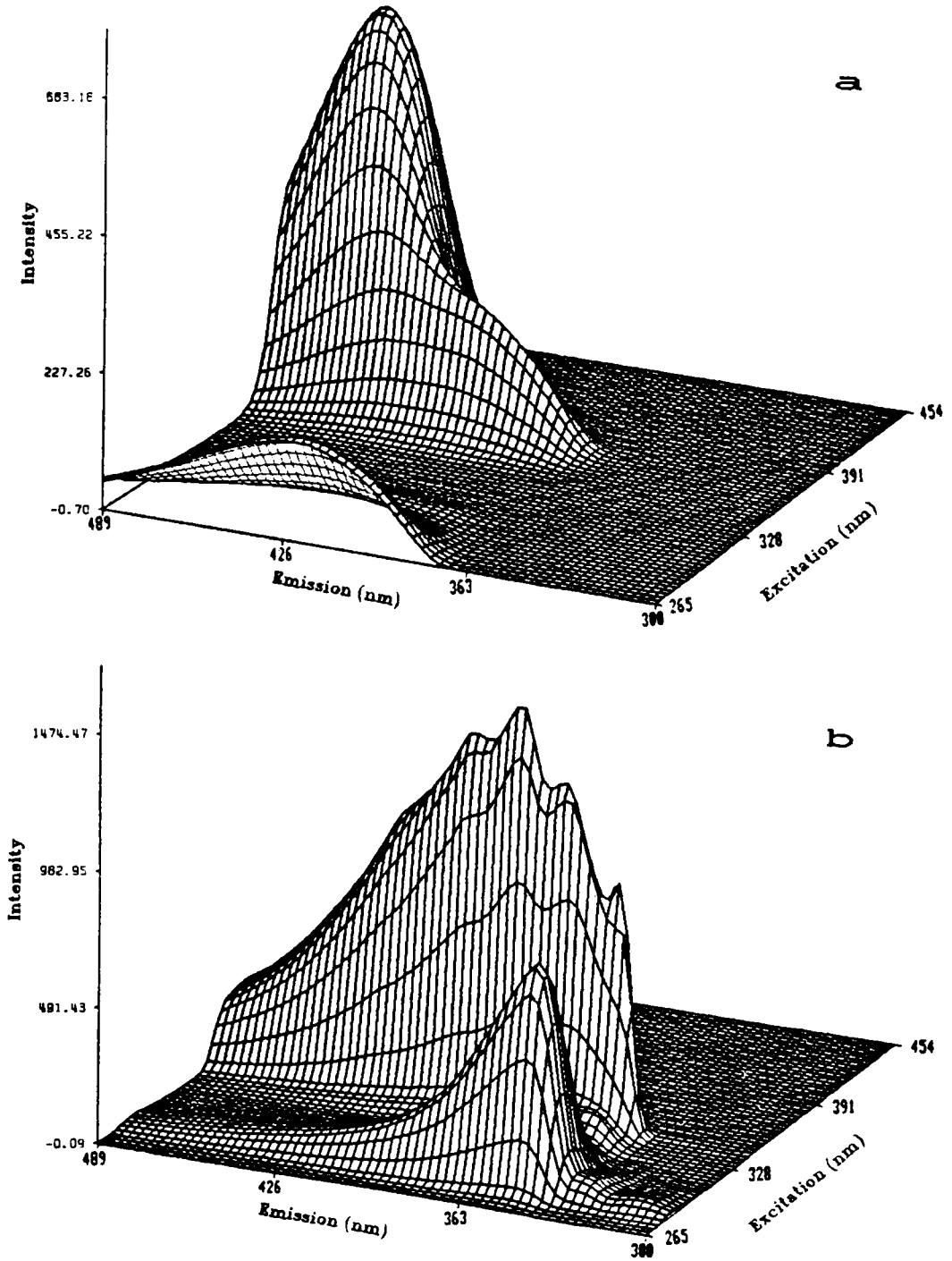


Figure A-27a. Esculin
b. Anthraquinone

APPENDIX B

The First Step Search Results (Prefilter Process) by Using 25% Clipped Data Bases

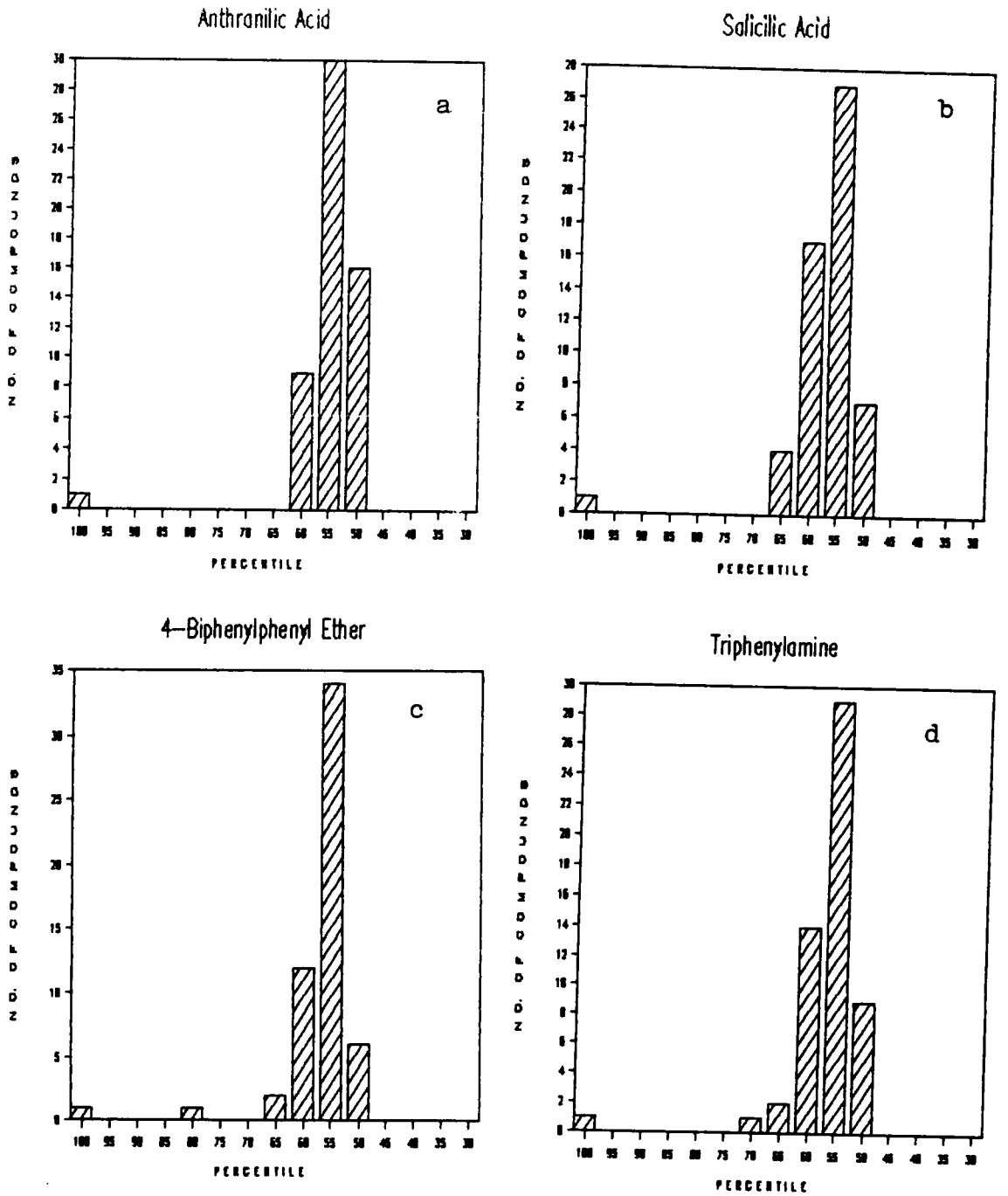


Figure B-1a. Anthranilic Acid
b. Salacilic Acid
c. 4-Biphenylphenyl Ether
d. Triphenylamine

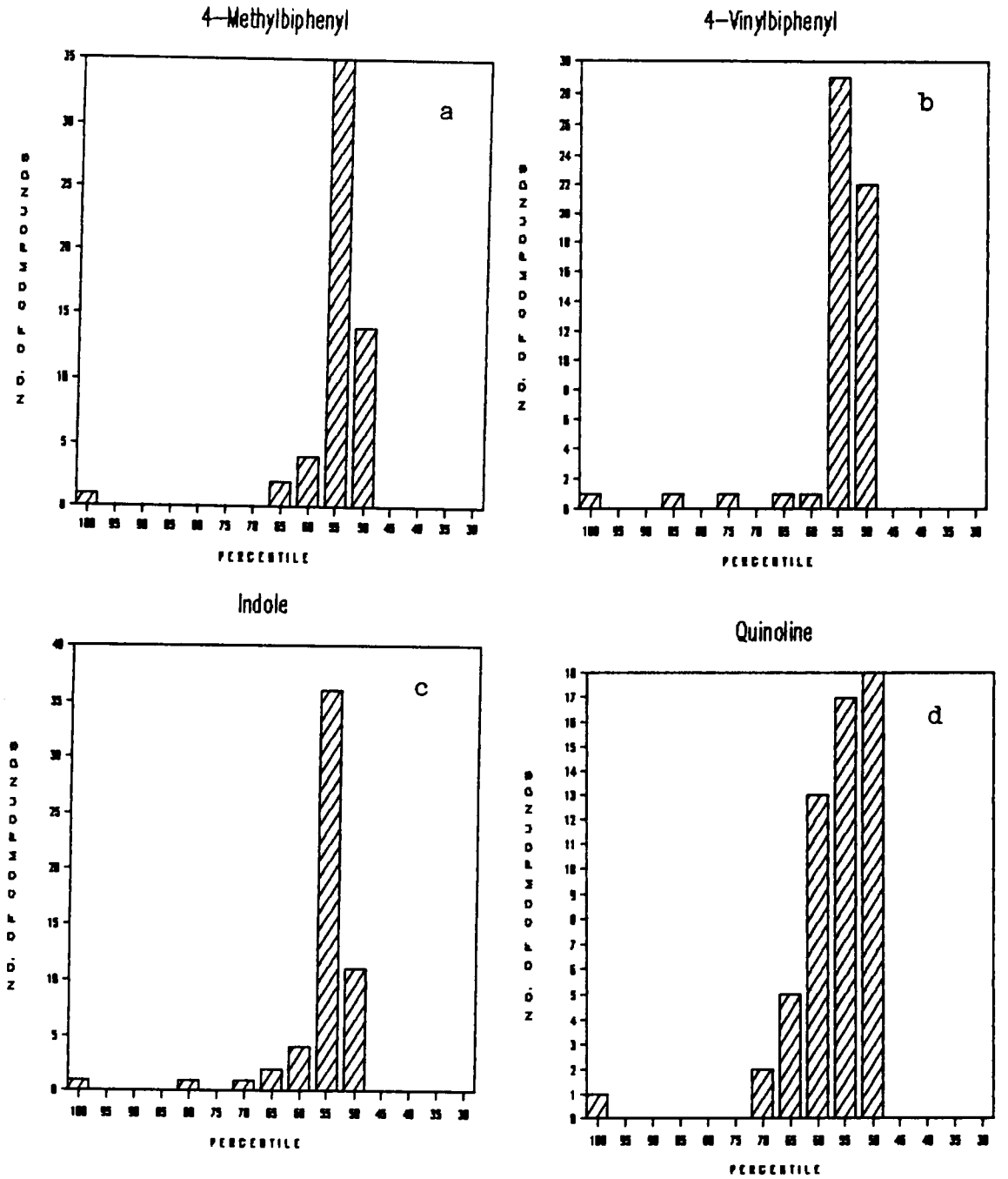


Figure B-2a. 4-Methylbiphenyl
b. 4-Vinylbiphenyl
c. Indole
d. Quinoline

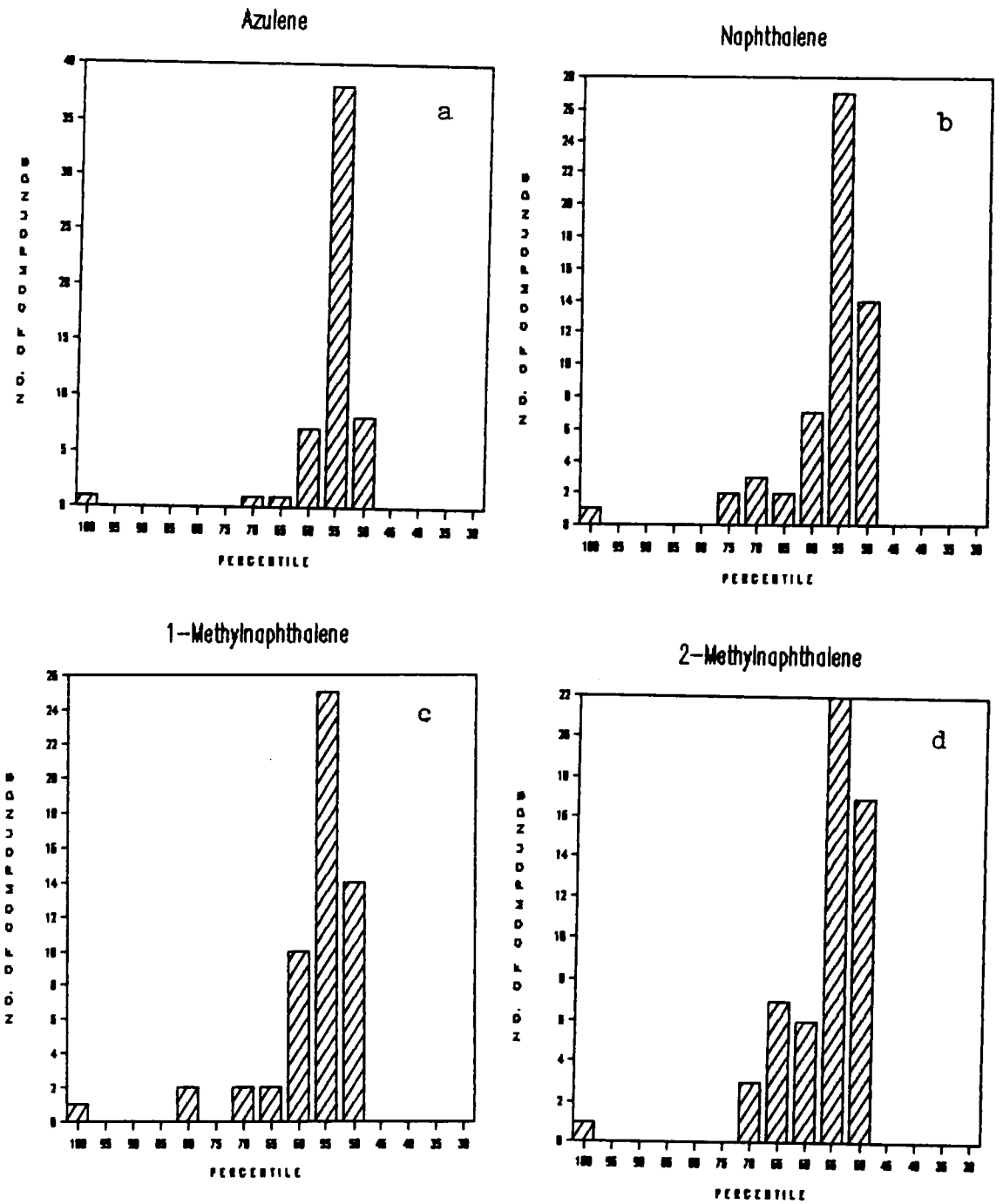


Figure B-3a. Azulene
b. Naphthalene
c. 1-Methylnaphthalene
d. 2-Methylnaphthalene

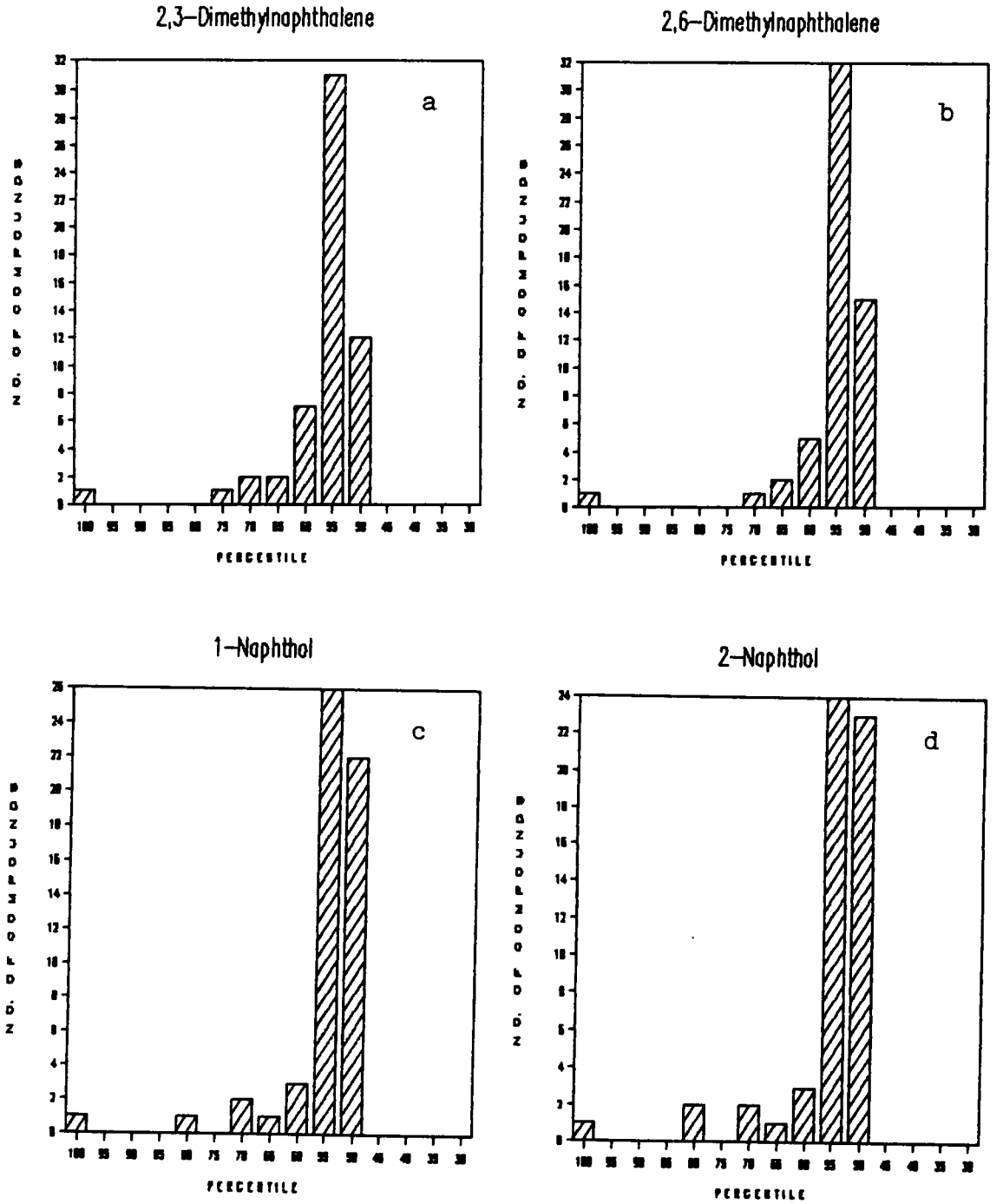


Figure B-4a. 2,3-Dimethylnaphthalene
b. 2,6-Dimethylnaphthalene
c. 1-Naphthol
d. 2-Naphthol

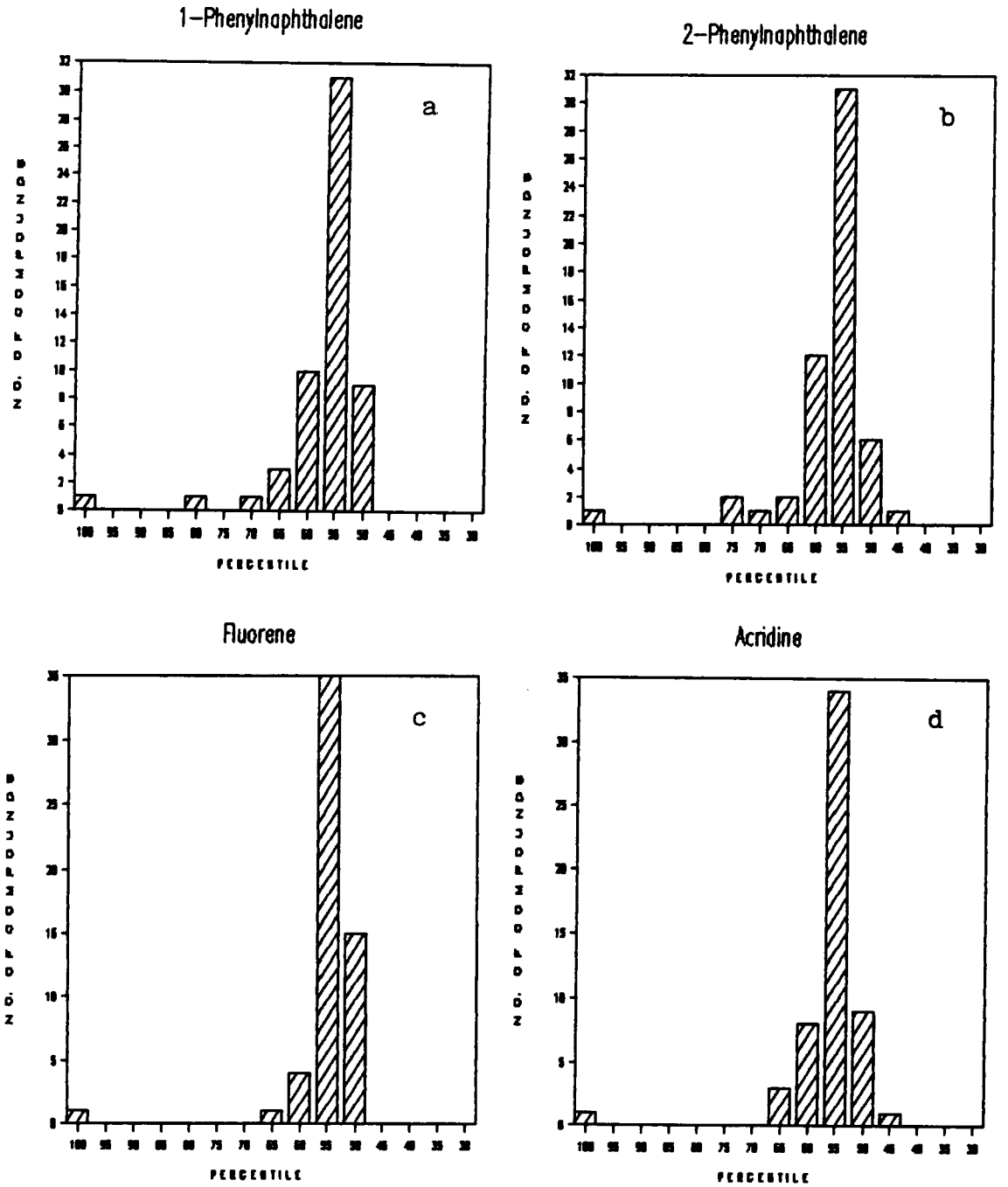


Figure B-5a. 1-Phenylnaphthalene
b. 2-Phenylnaphthalene
c. Fluorene
d. Acridine

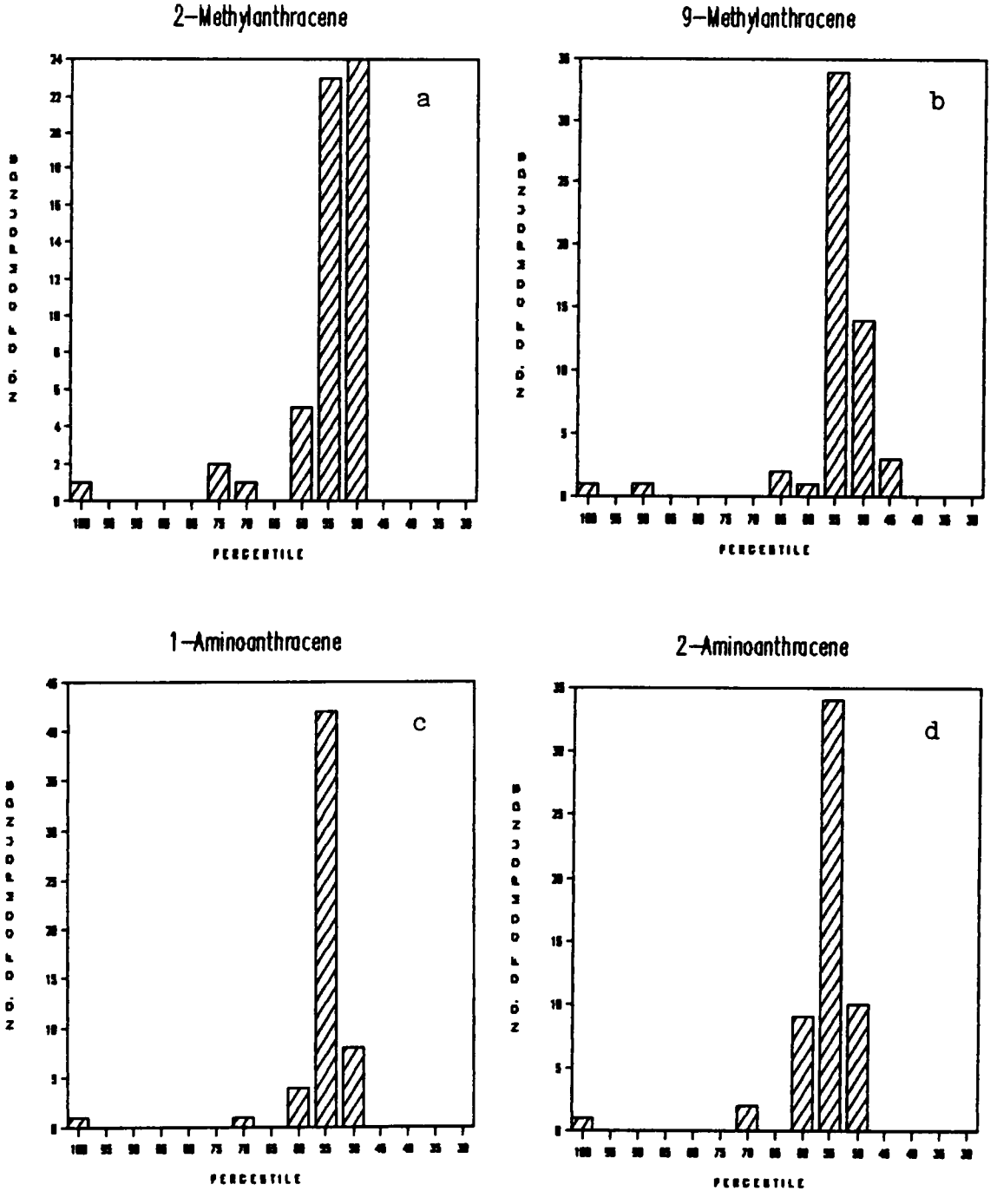


Figure B-6a. 2-Methylanthracene
b. 9-Methylanthracene
c. 1-Aminoanthracene
d. 2-Aminoanthracene

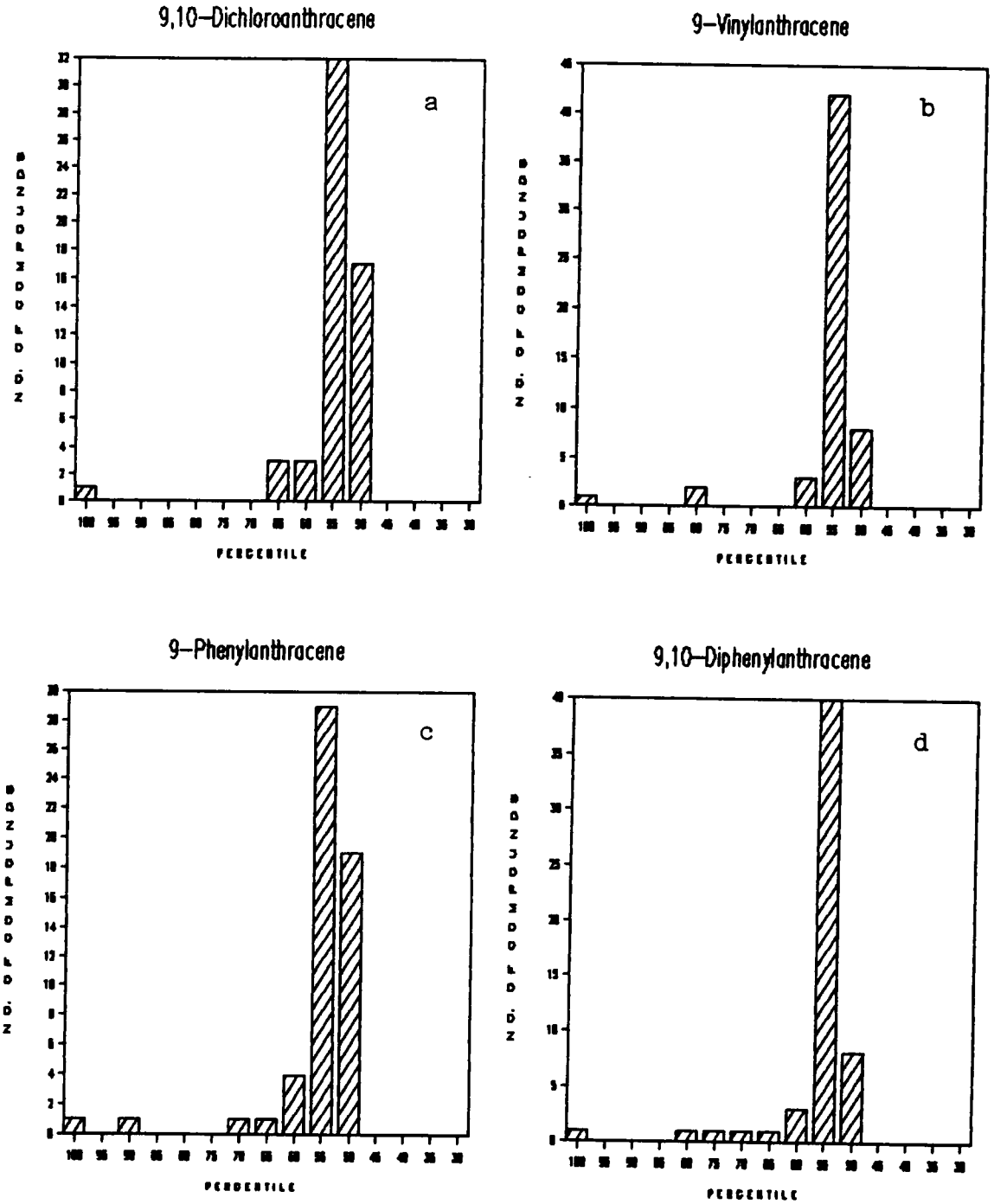


Figure B-7a. 9,10-Dichloroanthracene
b. 9-Vinylanthracene
c. 9-Phenylanthracene
d. 9,10-Diphenylanthracene

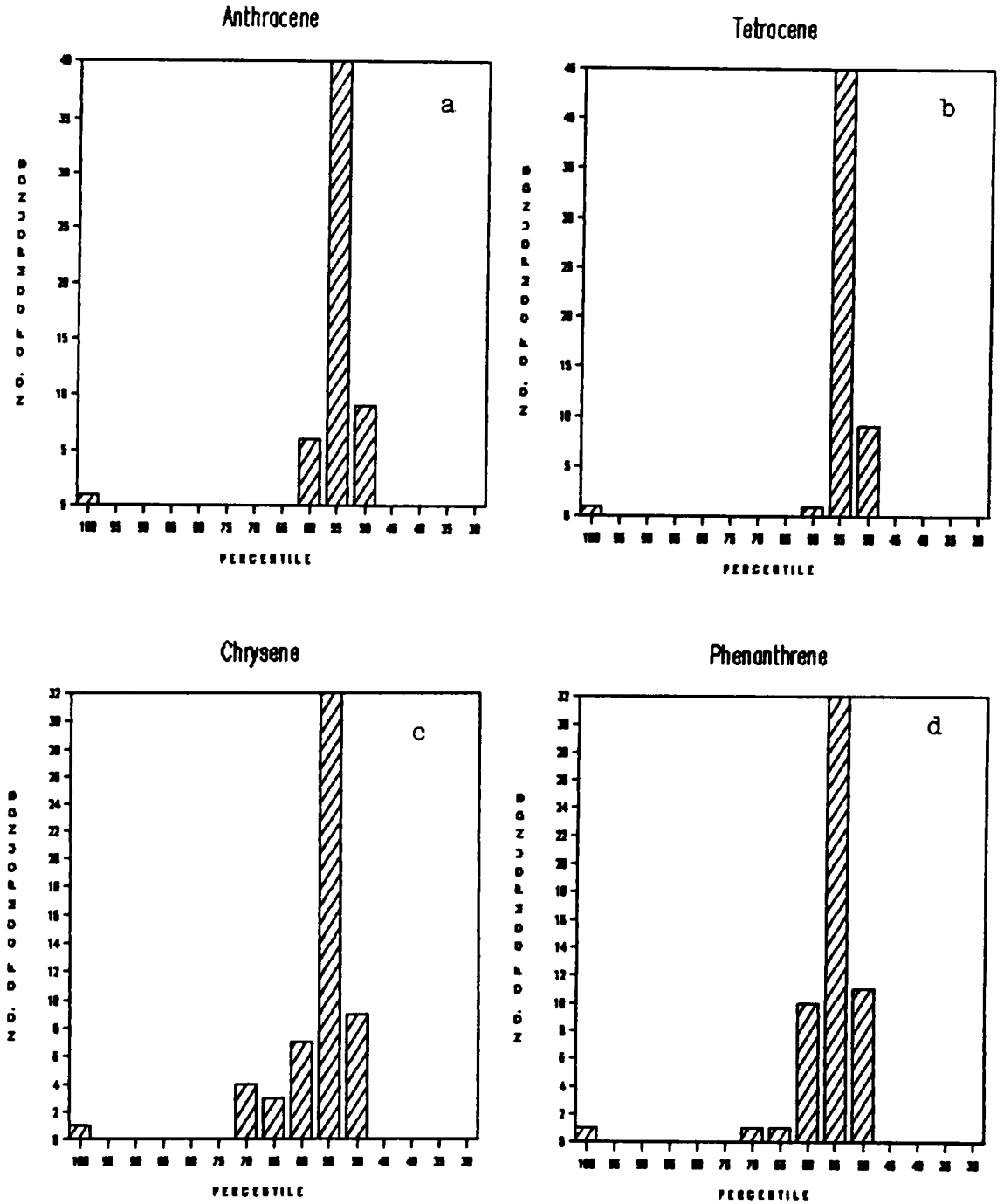


Figure B-8a. Anthracene
b. Tetracene
c. Chrysene
d. Phenanthrene

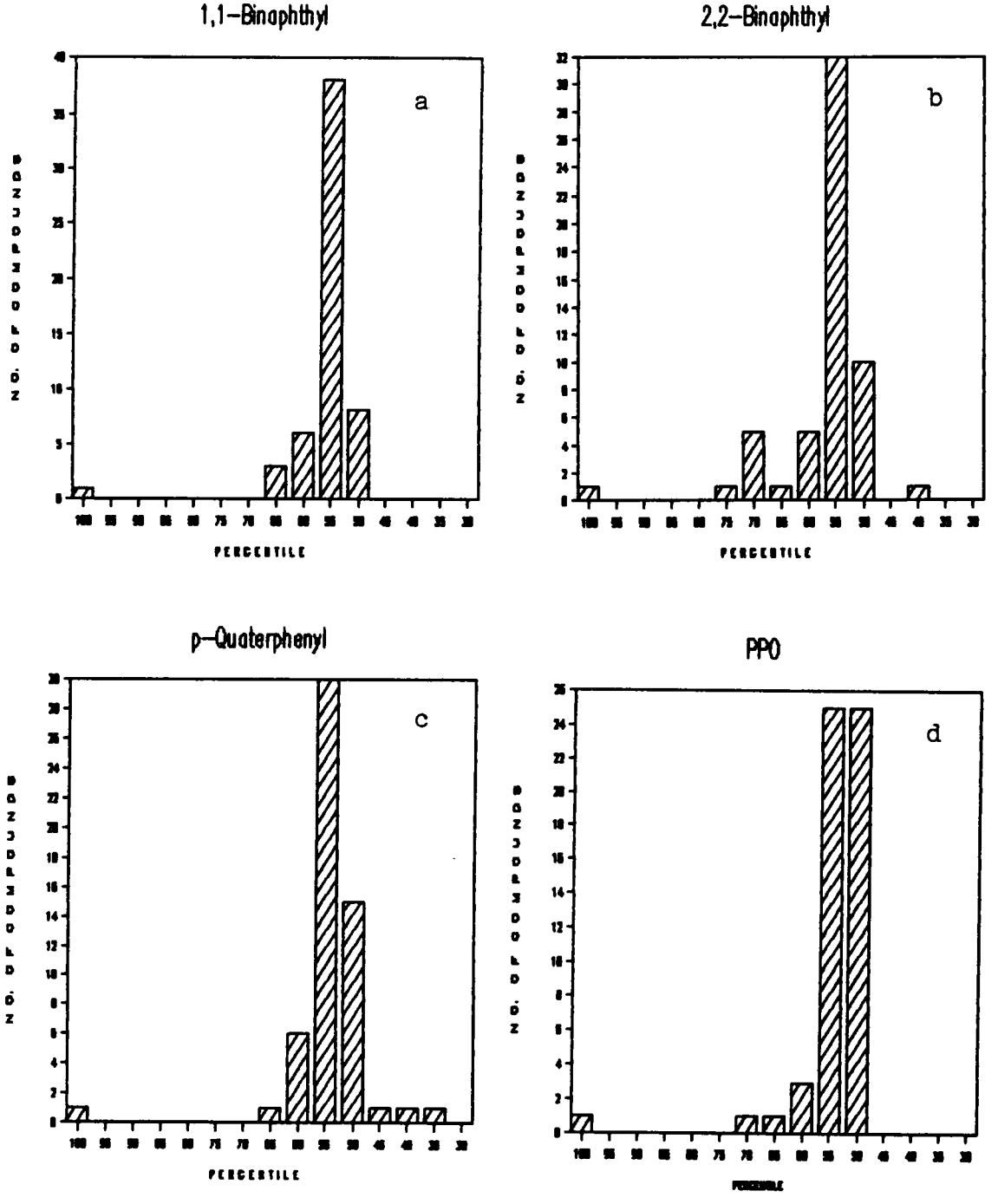


Figure B-9a. 1,1-Binaphthyl
b. 2,2-Binaphthyl
c. p-Quaterphenyl
d. PPO

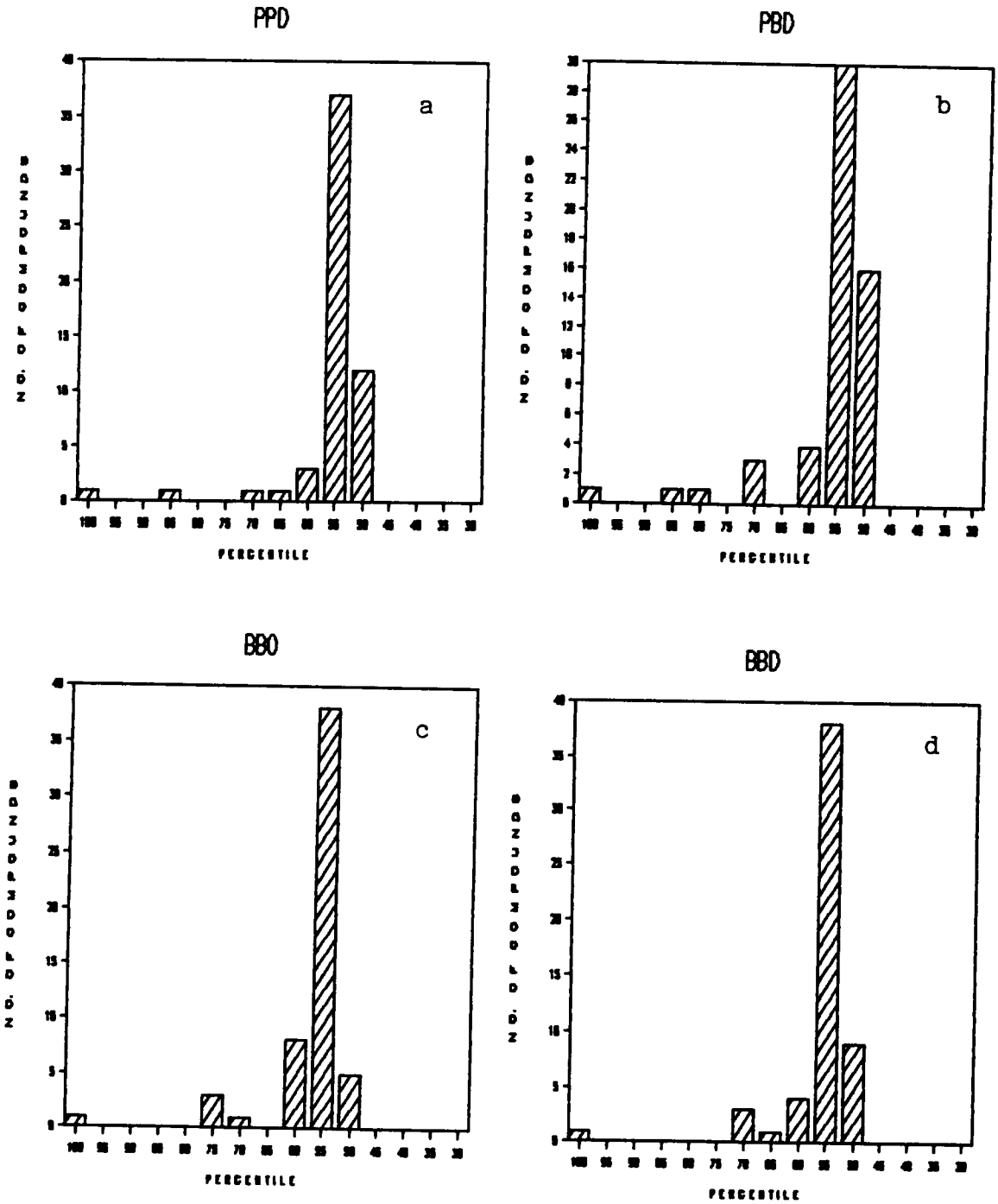


Figure B-10a. PPD
b. PBD
c. BBO
d. BBD

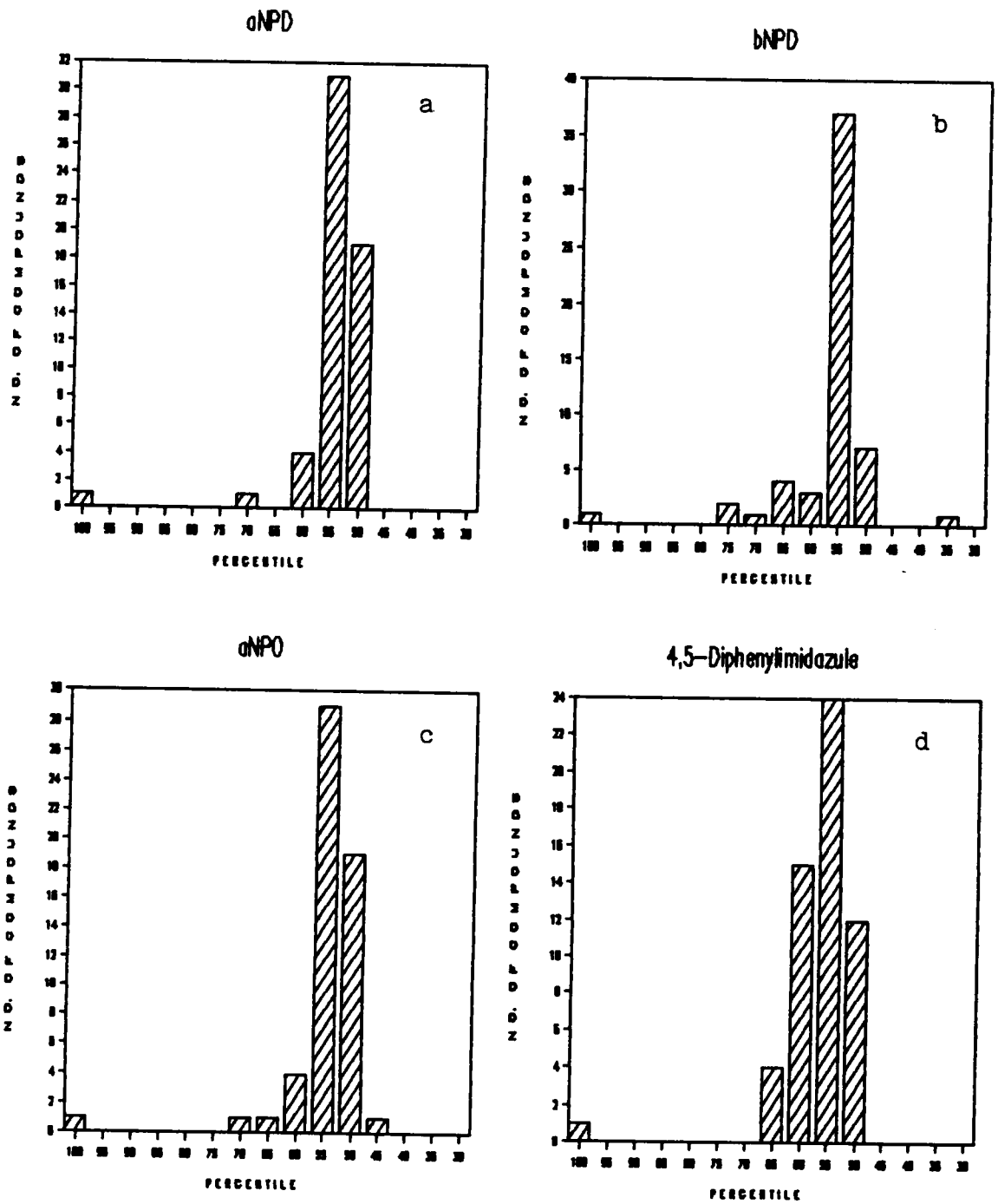


Figure B-11a. α-NPD
b. β-NPD
c. α-NPO
d. 4,5-Diphenylimidazule

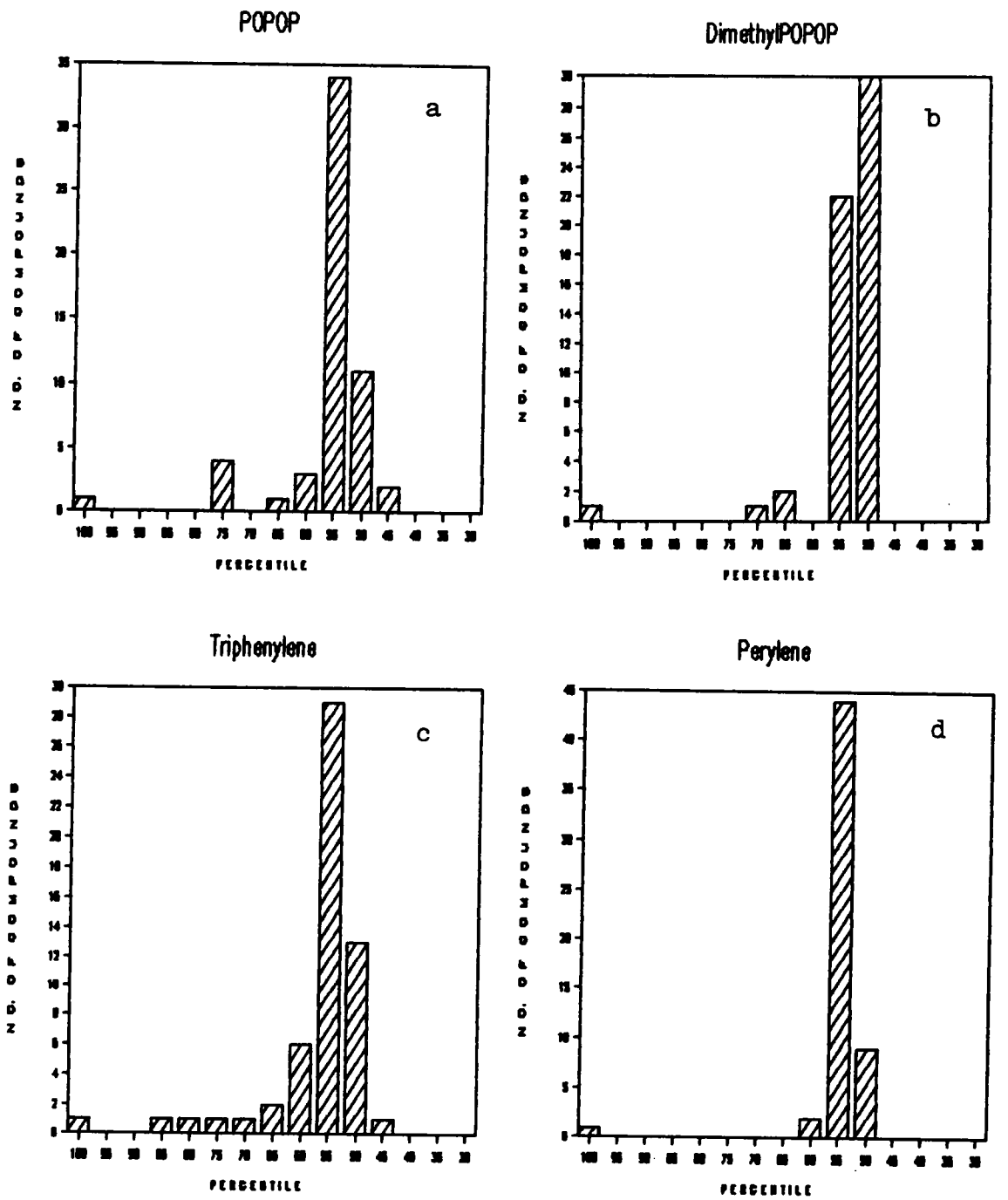


Figure B-12a. POPOP
b. Dimethyl-POPOP
c. Triphenylene
d. Perylene

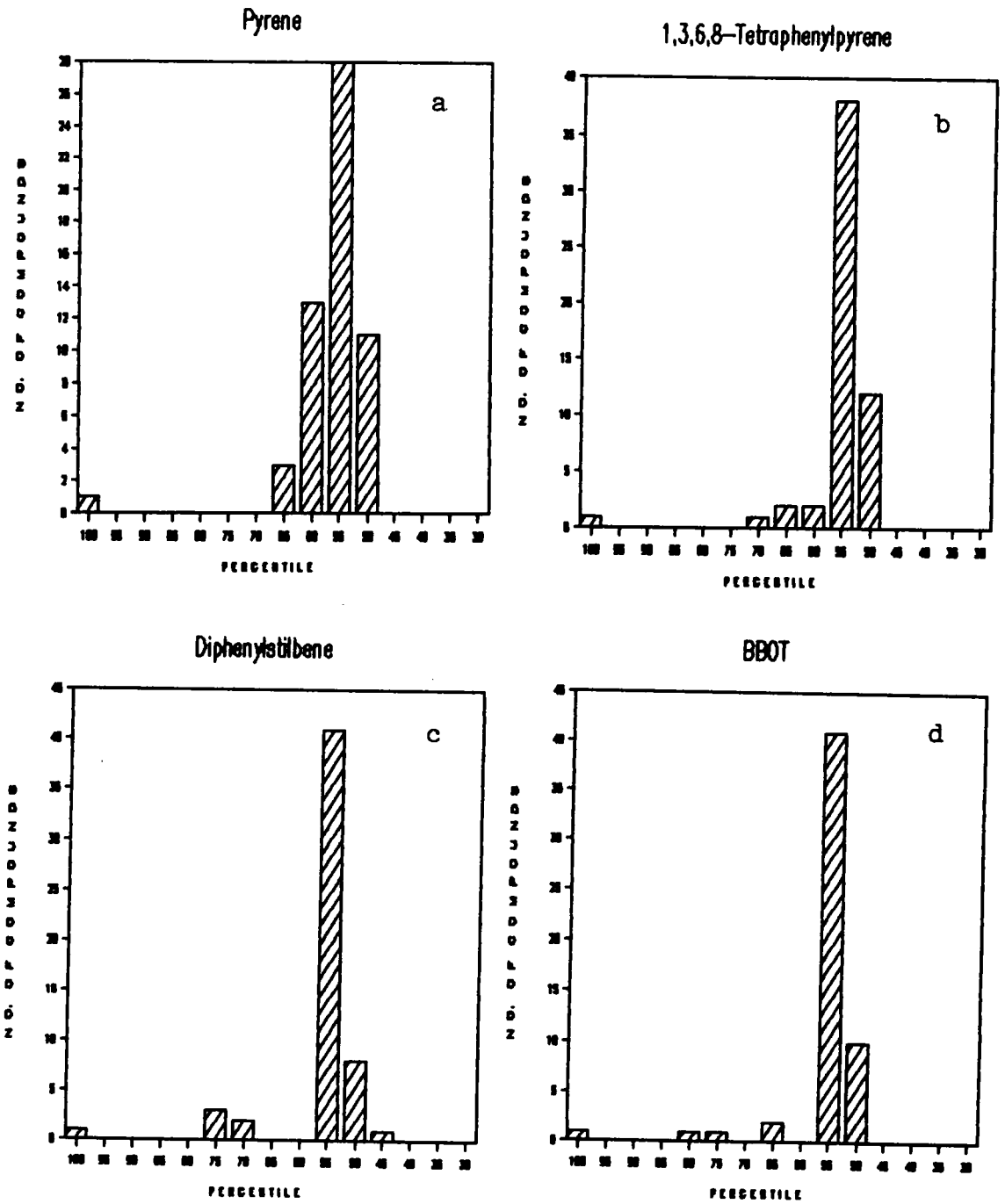
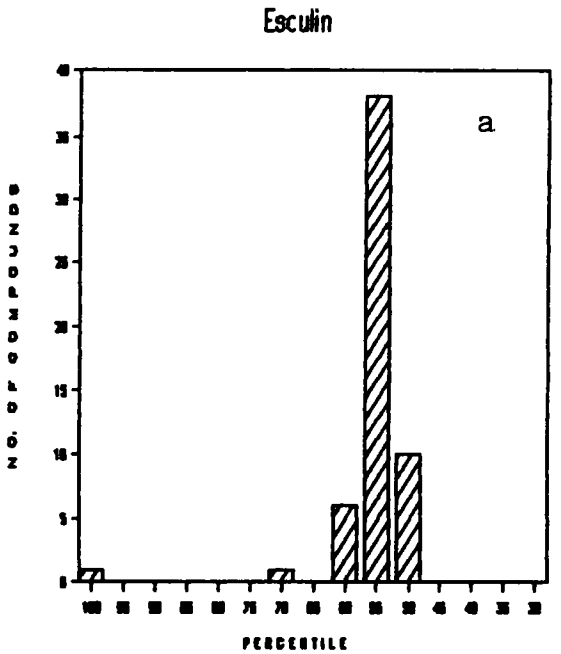


Figure B-13a. Pyrene
b. 1,3,6,8-Tetraphenylpyrene
c. Diphenylstilbene
d. BBOT



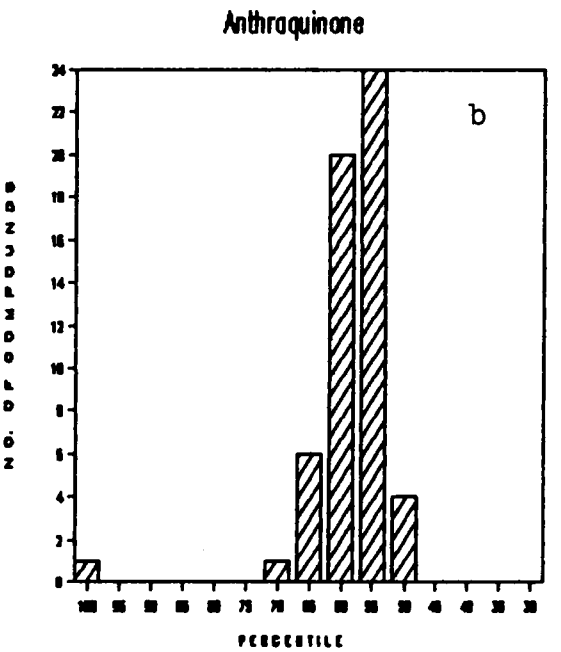


Figure B-14a. Esculin
b. Anthraquinone

APPENDIX C

Selected Major Software Listings
Developed During This Research

DATAQ.OY

```
&11Ø do display off
&break 1999
output clear vØ
output clear aØ
*************************************************************
*                          DATAQ.OY                         *
* This program is modified from SCOUT.OY (Rev. 4.ØØ         *
* developed by J.X.D., 11/12/84) which is one of the MPF-66*
* application software.  The modified program acquires      *
* three-dimensional fluorescence spectra of fixed emission *
* and excitation range (em=3ØØ-489nm; ex=265-454nm).        *
*                    by  F.I. 11/11/85                      *
* Soft key-driven version; load and run from /appl/fl only.*
* Works ONLY with MPF-66 and PECLS III M Rev. D software.   *
*************************************************************
set wait off
&keydef Ø
&def a2="scØØ1"
do sclear
do gclear
do display on
*                    b2c7DATAQ.OYbØc7
*    Three dimensional fluorescence survey analyses.
*    Standard SURVEY scan conditions are used (RATIO MODE,HI
*    GAIN,5nm SLITS)
*
*    c6INSTRUCTIONS:c7
*
*         1) Place the SAMPLE in the MPF-66.
*
*         2) All data in REGIONs X, Y, and Z will be
*            overwritten.
*
*
* c5SELECT SURVEY OPTION:c7
&keydef 6 "   EM   "," SURVEY ",2,7
&keydef 8 "  EXIT  "," TO CLS ",4,7
&keys. v1
* do display off
&if v1=6 then 13Ø
&if v1=8 then 1999
&if v1=-1 then 1999
&if v1=Ø then 1999
&13Ø *
&14Ø do display on
do sclear
* c5Place sample in MPF-66 then enter SAMPLE CODE (32
*   characters max.):c7
*
```

```
&enter a3
do sclear
* c5Linking to MPF-66c7
*
* do display off
&def a25="1"
* set prescan counter to 1
set space 3821 3821
*
retrieve x /usr/mpf/data/&a2
&error145       } retrieve error; goto retrieve error trap
&goto 147       } retrieve of dummy file succeeded, bypass
*                 error trap
&145 do sclear  } retrieve fatal error trap
do display on
*
*         c4**** FATAL ERROR !!!! ****c7
*       c5Unable to retrieve /usr/mpf/data/&a2.c7
*       This file is necessary for DATAQ operation.
*
*         Terminate DATAQ; return to CLS mode...
do pause,3
&goto 1999
&147 do display on  } dummy file retrieved successfully,
*                     attach to INST mode
inst
&error L50          } detect LINK error, if error goto link
*                     error trap
&BREAK L950
&GOTO L60           } LINK succeeded, bypass link error trap
&L50 do sclear      } link error trap
*         c4Lab computer unable to link with MPF-6c7
*
*         c5Check RS232C cable: ensure photometer and source
*          are turned on.
*
*
*         Choose option when ready to continue...
&keydef 0
&keydef 1 " RE-LINK",2,7
&keydef 8 "  EXIT  "," TO CLS ",4,7
&keys. v1
*    do display off

&if v1=1 then L47
&if v1=8 then L1000
&if v1=0 then L1000
&if v1=-1 then L1000
&L60 DO SCLEAR      } LINK succeeded; now in INST mode
DO DISPLAY ON
*         c5Setting instrument conditions for:
```

```
*
*     c1EXCITATION PRESCAN - 0th order EMISSIONc7
*
*               c1Prescanc3 # &a25
DO DISPLAY OFF
*          } goto emscan and update conditions to those of
*          } the sc001.sp in region X
do display on
MENU
EMSCAN
UPDATE
MENU
*          } exprescan to locate the max EX WL
PRESCAN
EXPRSCN
EDITSCAT 10
RANGE 265 454 0
RUN
&ERROR L70      } branch to L70 if "over range" error
*                 detected

&IF V31=0 then L50
&GOTO L100      } prescan succeeded, bypass the over range
*                 error trap
&L70 DO DISPLAY OFF       } determine how many times
*                           exprescan is attempted
&IF A2="sc001" THEN L80   } over range occurred in the 1st
*                           exprescan attempt
&IF A2="sc002" THEN L90   } over range error in 2nd exprescan
*                           attempt
&L80 DO SCLEAR            } 1st exprescan over range error
*                           trap
DO DISPLAY ON
*     ***** c4PRESCAN OVERRANGEc3 !!! *****
*
*        ** c5Reducing EM SLIT to 3nm c3**
*
MENU
EMSCAN
&INCR A2        } incr the dummy file so that the condition
*                 can be updated
&DEF A25="2"    } increase the exprescan counter to 2
RETRIEVE X /usr/mpf/data/&a2
&ERROR L85      } retrieve error occurred goto L85
&GOTO L87       } retrieve succeeded, bypass error trap
&L85            } goto CLS mode then goto retrieve fatal
*                 error trap
DETCLS
&goto 145
&L87 *          } rerout program to L60 to return to
&GOTO L60       } exprescan for the 2nd time with 3nm EM SLIT
```

```
&L9Ø DO SCLEAR       } 2nd over range error trap
DO DISPLAY ON
*         ***** c4Overrange number &a25 !!!c3*****
*
*     c5You MUST dilute sample or attenuate the fluorescence
*              c5signal before proceeding.
*
*     c5Press b2c7ENTERbØc5 when ready to continue:
DO PAUSE
MENU
DETCLS
&goto 11Ø       } after dilution, or attenuation, restart the
*                 entire program
* exprescan succeeded, goto CLS mode to recalculate EM scan
* range
&L1ØØ MENU
DETCLS
&error 1999
&break 1999
do sclear do display on
*
*     c5Calculating Emission SCAN RANGE...c7
*
*
* do display off
* v2=optimal EXWL for the emprscan, and for emscan
* v3=starting EMWL; v4=ending EMWL; max v4 possible is 895nm
* do display on
calc v2=int(v31+Ø.5)
calc v3=int(v31+15+Ø.5)
calc v4=int(v2+v2-1Ø+Ø.5)
&if v4 > 895 then 111Ø
&goto 112Ø
&111Ø calc v4=895              } if v4>895nm, then v4=895nm
&L12Ø *set for EM PRESCAN.     } goto INST mode

inst &error 1999
&L125 DO SCLEAR                } get ready for the EMPRESCAN
DO DISPLAY ON
*
*     c5Set conditions for EM PRESCANc7
*
*      c1(prescan number:c5&a25c1)c7
*
&ERROR L95Ø
&BREAK L(%)
DO DISPLAY OFF
*          } EMPRESCAN to locate max EM WL and its intensity
PRESCAN
EMPRSCN
EDITSCAT 1Ø
```

```
RANGE &V3 &V4 &V2
RUN
&ERROR L130     } goto L130 if "over range" error detected
&GOTO L140      } emprescan succeeded, bypass over range
error                }   trap
&L130 DO DISPLAY OFF          } determine how many times
*                               emprescan is attempted
&IF A2="sc001" THEN L133      } overrange error occurred in
*                               1st emprescan attempt
&IF A2="sc002 THEN L135       } over range error occurred in
*                               2nd emprescan attempt
&IF A2="sc003 THEN L90        } 3rd time over range error
*                               goto L90 to do dilution or
*                               attenuation
&L133 DO SCLEAR               } 1st emprescan over range
error trap
DO DISPLAY ON
*
*    *****c4Signal Overrange !!!c3 *****
*
*      c1** Closing EM SLIT to 3nm **c7
*
MENU
EMSCAN
&DEF A25="2"          } incr dummy file to get ready for the
*                       2nd prescan attempt
&INCR A2              } increase emprescan counter to 2
RETRIEVE X /usr/mpf/data/&A2
&ERROR L137           } if retrieve error detected goto
*                       retrieve error trap
* do display off
*                     } retrieve succeeded, update conditions
*                       to the 2nd dummy file
UPDATE
MENU
&GOTO L125            } now go try emprescan for the 2nd time
&L135 DO SCLEAR       } 2nd emprescan over range error trap
DO DISPLAY ON
*
*         c4*** Signal Overrange !!! ***c7
*
*           c5 ** Set GAIN to LOW **c7
MENU
EMSCAN
&DEF A25="3"          }incr dummy file to get ready for the
*                      3rd prescan attempt
&INCR A2              } increase emprescan counter to 3
RETRIEVE X /usr/mpf/data/&A2
&ERROR L137           } if retrieve error detected goto
                        retrieve fatal error trap
*                     } retrieve succeeded,update conditions
```

to the 3rd dummy file

```
UPDATE
MENU
&GOTO L125          } now go try emprescan for the 3rd time
&L137 DO SCLEAR     } retrieve fatal error trap
DO DISPLAY ON
*
*    c4***FATAL ERROR!! Can't fine /usr/fl/data/&A2 ***c7
*
DO PAUSE,3
&GOTO L950
&L140 *             } emprescan succeeded goto CLS mode
DETCLS
*
*    c5 Recalculating EM scan START WL...c7
*
*       c1 &v3nm - 5 c7
*
* do display off
&break 1999
calc v3=v3-5
*
&1600 * REM SAVE/RERUN options
do sclear
do gclear
do display on
status
*
&keydef 0
&keydef 1 "RUN 3-D ","EM SPECT",2,7     } ready to scan 3D
*                                         em spectra
&keydef 4 "  RERUN "," PRESCAN",2,7     } need to prescan
*                                         again
&keydef 8 "  EXIT  ","TO MENU ",4,7     } need to start over
&keys. v1
do display off
&if v1=1 then 1230
&if v1=4 then 1210
&if v1=8 then 1999
&if v1=0 then 1999
&if v1=-1 then 1999
&1210 *                  } need to prescan again
&goto L10
&1230 *                  } ready to scan 3-D Em spectra
&1300 * RUN THREE DIMENSIONAL FLUORESCENCE SPECTRA
do sclear
do display on
status y
*
*
```

```
*    c5Enter DESTINATION for save of EMISSION SPECTRUM:c7
&keydef 0
&keydef 1 " FLOPPY "," DISK 0 ",2,7
&keydef 2 " FLOPPY "," DISK 1 ",2,7
&keydef 3 "  HARD ","  DISK  ",2,7
&keydef 8 "  EXIT  "," TO CLS ",4,7
&keys. v1
do display off
&if v1=1 then 1305
&if v1=2 then 1310
&if v1=3 then 1315
&if v1=8 then 1999
&if v1=0 then 1999
&if v1=-1 then 1999
&1305 *
&def a9="f0:"
&goto 1320
&def a9="f1:"
goto 1320
&1315 *
&def a9="w4:"
&1320 do sclear
do display on
* c5Enter the first FILID for EMISSION spectrum
*              (8 characters max.):c7
*c5FILID should be in the form of AA###c7
&enter a10
*                         * &a10 *
*
do sclear
* c1Checking &a9 for &a10...c7
do display off
&def a30=a9+a10+".sp"
&check a30
&error L322
&goto L325
&1322 do sclear
do display on
*
********** c4Error !!! &a10.sp already exists c3 **********
*
*
* c5Press b2c7ENTERb0c5 to re-enter SAVE parameters...c7
do pause
&goto 1300
&L325 *
do display off
calc v60=265
calc v61=1
&def a4=clk
&def a5=date
```

```
&for v61=1,64,1
&if v61<>1 then copy y z
inst                    } goto INST mode
&error 1999
&BREAK L950
DO DISPLAY ON
*
*     c5... Setting conditions for EM SURVEY scan.c7
*
*
MENU
EMSCAN
EXWL &v60
SCROLL
RANGE 300 489
RESPONSE AUTO
SCAN SPEED 120
RUN Y
MENU
DETCLS
&break 1999
&error 1999
&if v61=1 then L420       } if very first scan, skip storing
*                           process
do sclear
do display on
save z &a9&a10,&a3,sc14.0,&a4,&a5
&error 1530
&goto 1420
&1530 do sclear
***** c4ERROR !!!!! Write to disk failed c3 *****
*
*     c5Press b2c7ENTERb0c5 to retry SAVE...c7
do pause
&goto 1600

&L420 *
do display off
&INCR a10                 } increase the filid name, and EXWL
calc v60=v60+3            } to the next step
do display on
*     v60
do display off
&next v61

save z &a9&a10,&a3,sc14.0,&a4,&a5
&error 1531
&goto 1430
&1531 do sclear
***** c4ERROR !!!!! Write to disk failed c3 *****
*
```

```
*    c5Press b2c7ENTERb0c5 to retry SAVE...c7
do pause
&goto 1600

&1430 do sclear

*    ******** c1End of one set of 3-Dimensional ********
*               fluorescence spectral scanc3
&keydef 0
&keydef 1 "  RERUN "," DATAQ ",2,7
&keydef 8 "  EXIT  "," TO CLS ",4,7
&keys. v1
do display off
&if v1=1 then 110
&if v1=8 then 1999
&if v1=0 then 1999
&if v1=-1 then 1999

&L950 DO SCLEAR
DO DISPLAY ON
* ***c4INSTRUMENT ERROR or BREAK encounteredc3 ***
*
*         c5Terminate DATAQ; return to CLS...c7
*
DO PAUSE,2
MENU
DETCLS

&1999 do display off
* END DATAQ
view scale 0 100
output clear v0
output clear a0
&keydef 0
set wait on
do gclear
do sclear
do display on
inst
MENU
DATA

&11000 set wait on
do sclear
do display on
```

RECVPE.BAS

```
     !-----------------------------------------------------&
     ! Fumiko Ishihara                                     &
     ! /revised          28-JUN-85 through 7-NOV-85        &
     !-----------------------------------------------------&
                                                           &

     !-----------------------------------------------------&
     ! This program receives the data from PE7000 data     &
     ! station into new file in RSTS/E.  It receives the   &
     ! data one line at a time; each received line is put  &
     ! in the buffer (260 line-max).  When the buffer      &
     ! becomes full or finished receiving all the line, it &
     ! will write into the file on the disk.  If not the   &
     ! end of the data file, then reinitialize the         &
     ! parameters and start receiving the line again.  If  &
     ! the end of the file, then close the new file.  The  &
     ! data will be sent to RSTS/E in 4 bits Hexadecimal   &
     ! Ascii character, so after receiving the data, bit   &
     !  rearranging is necessary.                          &

10   EXTEND
11   ON ERROR GOTO 30000                                   &
          ! universal error handler                        &

15   DIM BUFF%(10400),LIN%(80),ALINE%(40),LENGT%(260)      &

35   INPUT" Input the KB# of which the data is coming in ";&
     \ K$      K$="KB"+CVT$$(K$,-1%)+":"                   &
          ! ask for the KB# of which the PE7000 is        &
          ! connected                                      &

37   OPEN K$ AS FILE #1, RECORDSIZE 1024%, MODE 4%         &

40   INPUT" filename "; FILE$                              &
          ! FILE$ contains the file name                   &

50   OPEN FILE$ AS FILE #2%, RECORDSIZE 1024%, MODE 4%     &
          ! open new file which will receive the data as  &
          ! file #1 mode 4% to suppress automatic CR/LF at &
          ! right margin                                   &

60   PRINT " Ready to receive the data "                   &

70   GOSUB 1000          ! initialize all the parameters   &

80   INPUT #1%,LIN$                                        &
     \ CHANGE LIN$ TO LIN%                                 &
     \ N%=(LIN%(0)-9%)/8%                                  &
          ! LIN$ has the following format;                 &
```

```
          !    Hnnnnccddddddss      3 bytes = 6 HXD char's  &
          ! H=actual letter H; nnnn = 4 decimal digit line &
          !                           count                &
          ! cc=# of data byte on the line in hexadecimal   &
          ! dd=actual data bytes in hexadecimal pairs      &
          !    (ex. above)                                  &
          ! ss=check sum                                   &
          ! Receive the one line of data into line_buffer  &
          ! [LIN$] as a string then convert each character &
          ! in a string to its ascii equivalent and store  &
          ! it to LIN%(i).                                  &
          ! N%=# of hexadecimal character received 8/line &

90   FOR I=4 TO N%+3%                                      &
     \ IL=2*I                                              &
     \ IH=IL+1                                             &
               ! the actual data is from LIN%(8) to N%+3% &
               ! IL=low 4 bits:    IH=high 4 bits:         &

100  IF LIN%(IL) < 65% THEN LIN%(IL)=LIN%(IL)-48%          &
               ELSE LIN%(IL)=LIN%(IL)-55%                  &
               ! Change each ascii equivalent to the      &
               ! decimal  number which will correspond to  &
               ! the original binary value.  This is for a &
               ! lower 4 bits.                             &

110  IF LIN%(IH) < 65% THEN LIN%(IH)=LIN%(IH)-48%          &
               ELSE LIN%(IH)=LIN%(IH)-55%                  &
               ! Do the same as above for a higher 4 bits  &

120  LIN%(IH)=LIN%(IH)*16%                                 &
     \ ALINE%(I-3)=LIN%(IH)+LIN%(IL)                       &
               ! Left shift the higher 4 bits so that the  &
               ! value will correspond xxx as high 4    &
               ! bits in a byte.  By adding the higher 4   &
               ! bits and lower 4 bits, create a byte and  &
               ! store it as an integer.  ALINE%(i)        &
               ! contains the ascii equivalent of the byte.&

130  NEXT I     ! Repeat the data manipulation for the     &
               ! rest of the characters in a line.         &

135  ALINE%(0)=N%                                          &

137  D%=FNIDONE%(ALINE%)                                   &
140  IF D%=1%       THEN GOTO 270                          &
               ! Finished receiving all the lines ?        &
               ! If so close the file.                     &

200  J%=1%
     \ FOR  I=INDEX% TO INDEX%+N%
```

```
     \ BUFF%(I)=ALINE%(J%)
     \ J%=J%+1%
     \ NEXT I
               ! If not, write the received data-line into &
               ! buffer [BUFF].  INDEX%=character pointer  &
               ! J=pointer for the lin_buffer              &

210  LENGT%(LINE%)=N%      ! LENGT%=the # of characters      &
                           ! received for the current line   &

220  INDEX%=INDEX%+N%      ! update the character pointer    &

230  ILINE%=ILINE%+1       ! update the line pointer for    &
                           ! the next line

240  IF(ILINE% < 261%) GOTO 260
               ! BUFF can contain max. of 260 lines.  Are  &
               ! we received 260 lines?  If not, get the   &
               ! next line.                                 &

250  GOSUB 2000            ! output subroutine              &
     \ GOSUB 1000          ! initialize subroutine          &
               ! Yes, start writing on the disk and        &
               ! initialize all the necessary parameters   &
               ! then get ready for the next set of lines  &
               ! of data.                                   &

260  PRINT #1%,CHR$(127%);CHR$(127%);                       &
     \ GOTO 80                                              &
               ! No, we did not receive 260 lines yet.     &
               ! Send <CR><LF> to receive the next line.   &

270  GOSUB 2000                                             &
     \ CLOSE #1%,#2%                                        &
               ! Finished receiving all the line so write  &
               ! the contents of remaining buffer and      &
               ! close the files                           &

290  PRINT " LINE# = ";ILINE%;"   CHAR POINTER = ":INDEX%; &

300  GOTO 4000             ! End the program                &

999  !-----------------------------------------------------&
     !         Initialization subroutine                  &
     !-----------------------------------------------------&

1000 FOR I=1 TO 10400
     \ BUFF%(I)=0%                                          &
     \ NEXT I                                               &
               ! clear LENGT% which will contains the # of &
               ! characters received each line             &
```

```
1020 INDEX%=1          ! initialize character pointer          &
1030 ILINE%=1          ! initialize line pointer               &

1040 RETURN                                                    &

1999 !-------------------------------------------------------&
     !          Output subroutine                            &
     !-------------------------------------------------------&

     ! If the buffer is full or all the line is received,    &
     ! then need to write the contents of the buffer onto    &
     ! the disk.                                             &

2000 INDEX%=1          ! set character pointer on top          &

2010 FOR I=1 TO ILINE%-1%                                      &
     \ FOR J=INDEX% TO INDEX%+LENGT(I)-1%                      &
     \ PRINT #2%, BUFF%(J);                                    &
     \ NEXT J                                                  &
     \ PRINT #2%,CHR$(13);CHR$(10)                             &
     \ INDEX%=INDEX%+LENGT%(I)                                 &
     \ NEXT I                                                  &
          ! Write one line into the file.  Update the         &
          ! character pointer to the beginning of the next    &
          ! line.  Repeat for all the line.                   &

2200 RETURN                                                    &

2999 !-------------------------------------------------------&
     !          Function to determine the end of the file    &
     !-------------------------------------------------------&

     ! FNIDONE% returns 1 if the end of file, 0 if not.      &

3000 DEF FNIDONE%(ALINE%)                                      &
     \ FNIDONE%=0%                                             &
     \ CHANGE ALINE% TO ALINE$                                 &
     \ IF ALINE$="EOT" THEN FNIDONE%=1%                        &

3010 FNEND                                                     &
          ! Set FNIDONE% to 0.  Convert the first             &
          ! 3 characters of the line into a string.           &
          ! Is incoming line "EOT"?  If so reset FNIDONE%     &
          ! to 1.                                             &

4000 END
```

HADMD.FOR

```
      PROGRAM HADMD
C
C     HADMD.FOR
C     !-------------------------------------------------!
C     ! F.I.                                            !
C     ! /REVISED   2Ø-MAY-84 through 2Ø-JAN-88         !
C     !-------------------------------------------------!
C
C     This program includes input/output process, Hadamard
C     Forward/Inverse transform, different degree of
C     filtering process and clipping process.  Each
C     resultant data matrices can be stored in a designated
C     file in integer or floating format with an appropriate
C     heading.  The number of the data matrices to be
C     processed can be changed by changing the size of LFILE
C     This program was revised to process and to create
C     specific types of resultant data matrices.
C     II=2;  Walsh-Ordered WHT
C     II=3;  Inverse-Walsh Ordered WHT
C     X=Data array
C     N=# Of row and column of the data matrix
C     LFILE=filename of the library which contains the
C           original filenames and resultant filenames
C
      COMMON X(64,64)
      LOGICAL*1 M1,M2,M3,M4,M5,M6,M7,MR,N1,NR,LFILE(15)
      MR='Y'
      NR='F'
C
5     CONTINUE
C
      PRINT 2Ø
2Ø    FORMAT(' Input the dimension of the data matrix: ',$)
      ACCEPT 25,N
25    FORMAT ( I2)
C
C     Read the data; each block contains each row of data
C     matrix
C     Call subroutine to get the data form the different
C     file
C
      PRINT 2
2     FORMAT(' Enter the filename of the library which
     1 contains the original filename and the resultant
     1 filename: ',$)
      ACCEPT 8,NLU,(LFILE(I),I=1,NLU)
8     FORMAT(Q,14A1)
      PRINT 4
4     FORMAT(' Enter the number of files to be transformed:
```

```
     1 ',$)
      ACCEPT 6,NUNF
6     FORMAT( I3)
C
      OPEN (UNIT=11,NAME=LFILE,TYPE='OLD')

      DO 500 ILF=1,NUNF
C
         CALL GETSP (N)
C
         PRINT 40
40       FORMAT(' Input the type of transform (F=Forward,
         I=Inverse): ',$)
         ACCEPT 260,N1
         II=2                          ! set a flag to forward
                                         transform
         IF (N1 .EQ. 'I') II=3         ! change a flag to inv.
                                         transform
C
186      CONTINUE
C
C        Call the Walsh Hadamard Transform Subroutine
C
         CALL WHT (N,II)
C
C        If doing an inverse transform, do not clip
C
         IF (II .EQ. 3) GOTO 187
C
C        If doing a forward transform, determine if data
C        compression &/or clipping is desired
C
C        Ask if data compression is desired
C
         PRINT 252
252      FORMAT(' Need data compression [Y/N] ? ',$)
         ACCEPT 260,M7
         IF (M7 .EQ. 'Y') CALL COMPR(N,NC)
C
C        Call data compression subroutine
C
         PRINT  255
255      FORMAT(' Need clipping [Y/N] ? ',$)
         ACCEPT 260,M3
260      FORMAT(A1)
C
         IF (M3 .NE. 'Y') GOTO 3  ! if response is not 'Y',
C                                 ! skip clipping subroutine
C
         CALL CLIP(N,TT,AVE,NC,M7) ! Perform the clipping
C
```

```
        PRINT 265
265     FORMAT(' Do you want to store the clipped data on
       1 a disk [Y/N] ? ',$)
        ACCEPT 260,M6
        IF (M6 .EQ. 'Y') CALL SENDSP(N,II)
        GOTO 310
C
C       See if forward transformed data needs to be saved
C       on a disk
C
3       IF (M7 .EQ. 'Y') GOTO 266
C
        PRINT 300
300     FORMAT(' Do you want to store the transformed data
      1  on a disk 1 [Y/N] ? ',$)
        ACCEPT 260,M4
        GOTO 302
C
266     PRINT 301
301     FORMAT(' Do you want to store the transformed
      1  compressed data [Y/N] ? ',$)
        ACCEPT 260,M4
C
302     IF (M4 .EQ. 'Y') CALL SENDSP(N,II)
C
C       Ask user if an Inverse Transform is desired
C
310     PRINT 184
184     FORMAT(' Do you need to Inverse Transform [Y/N] ?
      1 ',$)
        ACCEPT 260,M1
        IF (M1 .NE. 'Y') GOTO 190
C
C       If not 'Y' jump to end of program
C
C       If 'Y' then set the direction flag II to 'inverse'
C       and jump back to a Walsh Hadamard subroutine
C
        II=3
        GOTO 186
C
C       If the transform just completed was an inverse
C       transform, ask if the resultant data is to be
C       written to a disk
C
187     PRINT 320
320     FORMAT(' Want to store Inv. Transformed data on
      1 a disk [Y/N] ? ', $)
        ACCEPT 260, M5
        IF (M5 .EQ. 'Y') CALL SENDSP(N,II)
C
```

```
190       CONTINUE
C
C         Check to see if the program is to be run again
C
C         PRINT 90
C 90      FORMAT(' Another Transform [Y/N] ? ',$)
C         ACCEPT 260,M2
C         IF (M2 .EQ. 'Y') GOTO 5
C
500    CONTINUE
C
       STOP
       END
C
C      ********************************************************
C      Subroutine to calculate Walsh Hadamard Transform
C
       SUBROUTINE WHT(NUM,II)
C
C      II=2 Walsh-Ordered WHT
C      II=3 Inverse Walsh-Ordered WHT
C
       COMMON X(64,64)
       DIMENSION IPOWER(10),Y(64)
       INTEGER*2 ALPH
C
C      This routine performed a forward and an inverse Fast-
C      Walsh-Hadamard transform.  A dimension of the data
C      matrix should be a power of two.
C      NUM = # of row and column
C      NT = 1, column transform; NT = 2 row transform
C
C      Fist perform a column transform, then perform a row
C      transform
C
       DO 10 NT=1,2
C
C          As J increases, it transforms the next column/row

           DO 20 J=1,NUM
C
C             If a Hadamard Ordered Transform is desired, skip
C             a bit reverse
C
              IF (II .LE. 1) GOTO 15
C
C             For a Walsh Hadamard Transform, a bit reversal
C             is required.  Bit reversal will calculate an
C             index IP which will indicate where the next
C             element will be stored in the processing array
C             Y, so that the processing array will be Walsh
```

```
C           ordered.
C
            DO 3Ø I=1,NUM
               IB=I-1
               IL=1
35             IBD=IB/2
               IPOWER(IL)=1
               IF (IB .EQ. (IBD*2)) IPOWER(IL)=Ø
               IF (IBD .EQ. Ø) GOTO 25
               IB=IBD
               IL=IL+1
               GOTO 35
25             CONTINUE
               IP=1
               IFAC=NUM
               DO 4Ø I1=1,IL
                  IFAC=IFAC/2
4Ø                IP=IP+IFAC*IPOWER(I1)
C
C              Now that IP has been calculated, use it to
C              load the next value from the data array into
C              the processing array
C
               IF (NT .EQ. 1) GOTO 45
               Y(IP)=X(J,I)            ! get a row element
               GOTO 3Ø
45             Y(IP)=X(I,J)            ! get a column element
3Ø          CONTINUE
C
C           Now that a complete row or column has been Walsh
C           Ordered, return the row or column to the
C           original data matrix
C
            DO 5Ø I=1,NUM
               IF (NT .EQ. 1) GOTO 55
               X(J,I)=Y(I)                ! row element
               GOTO 5Ø
55             X(I,J)=Y(I)                ! column element
5Ø          CONTINUE
C
15          CONTINUE
C
C           Now calculate the number of iterations
C
            ITER=Ø
            IREM=NUM
75          IREM=IREM/2
            IF (IREM .EQ. Ø) GOTO 65
            ITER=ITER+1
            GOTO 75
65          CONTINUE
```

```
C
C           Begin a loop for (Log to base two of NUM)
C           iterations
C
            DO 60 MM=1,ITER
C
C              Calculate number of partitions
C
               IF (MM .EQ. 1) NUMP=1
               IF (MM .NE. 1) NUMP=NUMP*2
               MNUM=NUM/NUMP
               MNUM2=MNUM/2
C
C              Begin a loop for the number of partitions
C
               ALPH=1
               DO 70 MP=1,NUMP
                  IB=(MP-1)*MNUM
C
C                 Begin a loop through this partition
C                 Done at each column or row
C
                  DO 80 MP2=1,MNUM2
                     MNUM21=MNUM2+MP2+IB
                     IBA=IB+MP2
C
C                    If a row transform, then do an addition
C                    & a subtraction for row data
C
                     IF (NT .EQ. 1) GOTO 85
                     Y(IBA)=X(J,IBA)+ALPH*X(J,MNUM21)
                     Y(MNUM21)=X(J,IBA)-ALPH*X(J,MNUM21)
                     GOTO 80
C
C                    Perform the column transform
C
85                   Y(IBA)=X(IBA,J)+ALPH*X(MNUM21,J)
                     Y(MNUM21)=X(IBA,J)-ALPH*X(MNUM21,J)
80                CONTINUE
C
C                 If this is a Walsh Hadamard transform,
C                 change the sign of Alpha
C
                  IF (II .GE. 2) ALPH=-ALPH
70             CONTINUE
C
C              After a column or a row transform, put the
C              transformed data into the original data
C              matrix
C
               DO 90 I=1,NUM
```

```
                IF (NT .EQ. 1) GOTO 95
                X(J,I)=Y(I)             ! repack the row after
C                                       ! a row transform
                GOTO 90
95              X(I,J)=Y(I)             ! repack the column
C                                       ! after a transform
90            CONTINUE
C
60          CONTINUE
20        CONTINUE
10      CONTINUE
C
C       If inverse transform, end of transform
C
        IF (II .EQ. 1 .OR. II .EQ. 3) RETURN
C
C       If forward transform, divide all elements by the
C       dimension of the data array
C
        DO 100 NT=1,2
           DO 110 J=1,NUM
              DO 120 I=1,NUM
C
                 IF (NT .EQ. 1) GOTO 105
                 X(J,I)=X(J,I)/(FLOAT(NUM))
                 GOTO 120
105              X(I,J)=X(I,J)/(FLOAT(NUM))
120           CONTINUE
C
110        CONTINUE
100     CONTINUE
        RETURN
        END
C
C       ******************************************************
C       Subroutine to clip the transformed data
        SUBROUTINE CLIP(N,TOTAL,AVE,NC,m7)
C
        COMMON X(64,64)
        LOGICAL*1 M7,MT       ! M7 is for data compression type
C
        IF (M7 .EQ. 'N') GOTO 10

C       If no data compression is done NC=n
C
        PRINT 20
20      FORMAT ( ' Do you want to clip the entire data set or
       1 just the compressed data set [E/C] ? ',$)
        ACCEPT 30,MT
30      FORMAT( A1)
        IF (MT .EQ. 'C') GOTO 40
```

```
10      NC=N
C       Calculate the TOTAL and average(Ave) of the data
C       points
C
40      TOTAL=0
        AVE=0
C
        X(1,1)=0
C
        DO 210 I=1,NC
           DO 211 J=1,NC
              TOTAL=TOTAL+X(I,J)
211        CONTINUE
210     CONTINUE
C
        NSQ=NC**2
        AVE=TOTAL/NSQ
C
C       Set all values which are above the average to 1.0
C       and the rest to 0.0
C
        DO 220 I=1,NC
           DO 221 J=1,NC
              IF (X(I,J) .LE. AVE) GOTO 222
              X(I,J)=1
              GOTO 221
222           X(I,J)=0
221        CONTINUE
220     CONTINUE
C
        IF (NC .EQ. N) RETURN  ! If an entire data is clipped,
        DO 230 I=1,NC          ! return.  If a portion of data
           DO 240 J=NC+1,N     ! is clipped, fill the rest
              X(I,J)=0.        ! with zeroes
240        CONTINUE
230     CONTINUE
        DO 250 I=NC+1,N
        DO 260 J=1,N
        X(I,J)=0.
260     CONTINUE
250     CONTINUE
C
        RETURN
        END
C
C       ********************************************************
C       subroutine to filter the data
C       It takes the desired filtering rate, and fill the
C       rest of the matrix with zeroes
C
        SUBROUTINE COMPR(N,NC)
```

```
      COMMON X(64,64)
      LOGICAL*1 LR
      NC=N/2                    ! NC is the dimension of 75%
C                               ! filtered data
      PRINT 10
10    FORMAT( ' What filtering rate is desired ? ')
      PRINT 15
15    FORMAT( ' (Q FOR 1/4, S FOR 1/16) ')
      ACCEPT 20,LR
20    FORMAT( A1)
      IF (LR .EQ. 'S') NC=N/4
C
C     If 1/16 filtering rate is desired, change the
C     dimension to N/4
C
      DO 30 I=1,NC
         DO 40 J=NC+1,N
            X(I,J)=0            ! fill the rest of the data
40       CONTINUE               ! field with zeroes
30    CONTINUE
C
      DO 50 I=NC+1,N
         DO 60 J=1,N
            X(I,J)=0            ! fill the rest  of the data
60       CONTINUE               ! field with zeroes
50    CONTINUE
C
      RETURN
      END
C
C     ********************************************************
C     Subroutine to read the data file
C
      SUBROUTINE GETSP(N)
C
      COMMON X(64,64)                     ! X=data matrix
C
      logical*1 mt,filen(15)
C
      DIMENSION IX(64,64)                 ! buffer array
C
C     Obtain the ascii data file.
C
      READ (11,230) NSP,(FILEN(I),I=1,NSP)
230   FORMAT(Q,14A1)
      OPEN (UNIT=9,NAME=FILEN,TYPE='OLD')
      READ (9,*) ((IX(I,J),J=1,N),I=1,N)
C
      DO 300 I=1,N
         DO 280 J=1,N
            X(I,J)=IX(I,J)
```

```
280       CONTINUE
300    CONTINUE
C
C      Close everything
       CLOSE(UNIT=9)
C
       RETURN
       END
C
C      ***************************************************
C      Subroutine to write the transformed or inverse
C      transformed data into the new data file on a disk
C      During the process of writing into a new data file,
C      it changes from a binary data to an ascii data
C
       SUBROUTINE SENDSP (N,II)
C
C      X=row, trans, clipped, or inv. trans data matrix
C
       COMMON X(64,64)
       LOGICAL*1 FILE(15),IFLAG,MCOMP,MCLIP
       DIMENSION IRAW(256)                ! buffer array
C
       NC=N

       READ(11,10) NCF,(FILE(I),I=1,NCF)
10     FORMAT(Q,14A1)
C
       PRINT 22
22     FORMAT (' Do you want to store an entire data set or
      1 just the filtered data [E/C] ? ',$)
       ACCEPT 40,MCOMP
C
       IF ( MCOMP .EQ. 'E') GOTO 28
       NC=N/2
28     CONTINUE
C
       PRINT 30
30     FORMAT (' Write data in integer or floating format
      1 [I/F] ? ',$)
       ACCEPT 40,IFLAG
40     FORMAT (A1)
       PRINT 41
41     FORMAT(' Is this a clipped data set ? ',$)
       ACCEPT 40,MCLIP
C
C      Open the new file to store the ascii data
C
       OPEN (UNIT=10,
      2  recordtype='fixed',recl=80,
      2 NAME=FILE, TYPE='NEW',
```

```
     2 carriagecontrol='FORTRAN')
C
C     begin to write the data onto the new data file
C
      IF (IFLAG .EQ. 'I') GOTO 46
      WRITE (10,45)
45    FORMAT (' ','(E11.3)')
      GOTO 48
C
46    CONTINUE
      WRITE (10,47)
47    FORMAT (' ','(  8I8)')
48    DO 100 I=1,NC
C
         DO 110 J=1,NC
            IRAW(J)=X(I,J)     ! IRAW contains one row of the
110      CONTINUE              ! data
C
C     Write one row of the data into a new data file which
C     causes data to be ascii characters
C
      IF (IFLAG .EQ. 'F') GOTO 70
      IF (MCLIP .EQ. 'Y') GOTO 56
      WRITE (10,54) (IRAW(J), J=1,NC)
54    FORMAT (' ',8I8)
      GOTO 100
C
56    CONTINUE
      WRITE (10,58) (IRAW(J), J=1,NC)
58    FORMAT (' ',32I2)
      GOTO 100
C
70    WRITE (10,71) (X(I,J),J=1,Nc)
71    FORMAT (' ',E11.3)
C
100   CONTINUE          ! repeat until the end of the file
C
C     Close everything
C
      CLOSE (UNIT=10)
      RETURN
      END
```

SEARCH.FOR

```
      PROGRAM SEARCH2
C     !-----------------------------------------------------!
C     !    Fumiko Ishihara                                  !
C     !    /Revised            39-JUL-85 through 23-JAN-88  !
C     !-----------------------------------------------------!
C
C     The first stage comparison (prefilter process) by using
C     an XNOR logic.
C     This program takes an unknown data file and compare it
C     with the reference library .  For the first stage
C     comparison, clipped data bases are used.  The number of
C     unknown data files can be changed by changing the
C     size of the library which contains the unknown data
C     files.  Also the different size of the clipped data
C     base can be used by changing the NDIM value.
C     Resultant files will contain match rates of an
C     unknown with each reference spectra as well as their
C     names in a descending order.
C     This program was revised to investigate the "optimal
C     compression rate"
C
      DIMENSION IX(64,64),IY(64,64) ! IX<IY are data matrix
      DIMENSION ICOUNT(100),ICH(100),INLCH(100),PERCF(100)
      DIMENSION NSN(100)
C                           ! ICOUNT = total # of match
C                           ! ICH = best matches, up to 10
C                           !       choices
C                           ! INLCH = stores the spectrum
C                                     number
      LOGICAL*1 LSFILE(11),SFILE(10),LUFILE(11),UFILE(11)
      LOGICAL*1 LAFILE(11),AFILE(11),M,MMM,MMMM
      LOGICAL*1 SNAME(25),UNAME(25),FSAVE(100,25)

C     LSFILE contains the filename which contains the
C           filenames of reference data files
C     SFILE contains the filename of the reference data in
C          the library
C     LUFILE contains the filename which contains the
C           filenames of unknown data files
C     UFILE contains the unknown data file
C     LAFILE contains the filename which contains the
C           filenames that will be used to hold search
C           results
C     AFILE contains the result of the search

      PRINT 10
10    FORMAT (' Enter the filename of the library which
     1 contains the unknown data to be compared: ')
      ACCEPT 22,NLU,(LUFILE(I),I=1,NLU)
```

```
      PRINT 12
12    FORMAT (' Enter the number of unknown to be compared:
     1 ',$)
      ACCEPT 14,NUNK
14    FORMAT ( I3)
      PRINT 16
16    FORMAT (' Enter the dimension of the data: ',$)
      ACCEPT 18,NDIM
18    FORMAT ( I2)
C
      ITOTAL=NDIM*NDIM
      FLDIM=FLOAT(NDIM)
C
      PRINT 20
20    FORMAT (' Enter the filename of the library which
     1 contains the results of the search: ')
      ACCEPT 22,NLA,(LAFILE(I),I=1,NLA)
22    FORMAT (Q,11A1)
C
      PRINT 30
30    FORMAT (' Enter the filename of the reference library:
     1 ')
      ACCEPT 22,NLS,(LSFILE(I),I=1,NLS)
      PRINT 80
80    FORMAT (' Enter the # of the spectrum in the reference
     1 library: ',$)
      ACCEPT 18,NL
90    CONTINUE
C
      OPEN (UNIT=11,NAME=LUFILE,TYPE='OLD')
      OPEN (UNIT=10,NAME=LAFILE,TYPE='OLD')
C
      DO 500 III=1,NUNK
C
         DO 600 ICHRC=1,25
            UNAME(ICHRC)=' '
600      CONTINUE
C
         DO 610 ICHR=1,77
            DO 620 ICHRC=1,25
               FSAVE(ICHR,ICHRC)=' '
620         CONTINUE
610      CONTINUE
C
         PRINT 630,III
630      FORMAT (' UNKNOWN=',I3)
         READ (11,100) NU,(UFILE(I),I=1,NU)
100      FORMAT (Q,11A1)
         READ (11,102) NUNM,(UNAME(I),I=1,NUNM)
102      FORMAT (Q,25A1)
C
```

```
      OPEN (UNIT=8,NAME=UFILE,TYPE='OLD')
C
      READ (8,110)          ! read the heading from the file
      READ (8,*) ((IX(I,J),J=1,64),I=1,64)
110   FORMAT ( A7)          ! read the data into IX matrix
C
      CLOSE (UNIT=8)
C
      INL=0
C
      DO 140 L=1,NL         ! ICH is the # of matches in
         ICH(L)=0           ! a descending order
         ICOUNT(L)=0        ! ICOUNT is the # of matches
         PERCF(L)=0         ! PERCF is % of matches in order
140   CONTINUE
C
      OPEN (UNIT=9,NAME=LSFILE,TYPE='OLD')
C
      DO 150 K=1,NL
C
         DO 640 ICHRC=1,25
            SNAME(ICHRC)=' '
640      CONTINUE
C
         INL=INL+1
C
         READ (9,141) NS,(SFILE(I),I=1,NS)
141      FORMAT (Q,11A1)
         READ (9,142) NSNM,(SNAME(I),I=1,NSNM)
142      FORMAT (Q,25A1)
C
         OPEN (UNIT=8,NAME=SFILE,TYPE ='OLD')
C
         READ (8,110)       ! read the heading from the file
         READ (8,*) ((IY(I,J),J=1,64),I=1,64)
C                           ! read the data into IY matrix
C
         CLOSE (UNIT=8)
C
         DO 650 ICHRC=1,NSNM
            FSAVE(INL,ICHRC)=SNAME(ICHRC)
650      CONTINUE
C
         INDEX=0            ! INDEX is a pointer in the line
         IOLD=1
C
C
C        comparison starts
C
240      DO 250 I=1,NDIM
            DO 260 J=1,NDIM
```

```
                IF (IX(I,J) .NE. IY(I,J) ) GOTO 260
                ICOUNT(INL)=ICOUNT(INL)+1
260          CONTINUE                ! store the matches in
250       CONTINUE                   ! ICOUNT
C
          PERCTT=ICOUNT(INL)
          PERCT=PERCTT/ITOTAL
          PERC=PERCT*100
C
C         sought through ICH values, so that each time new
C         ICOUNT is reported and if this value is high
C         enough relative to the existing ICH values, it
C         can be placed in ICH with appropriate order
C
          IF (INL .EQ. 1) GOTO 300
          IF (ICOUNT(INL) .LT. ICH(NL)) GOTO 150
C
          DO 290 NT=1,NL-1
             IF (ICOUNT(INL) .LT. ICH(NL-NT)) GOTO 310
             ICH(NL-NT+1)=ICH(NL-NT)
             INLCH(NL-NT+1)=INLCH(NL-NT)
             PERCF(NL-NT+1)=PERCF(NL-NT)
             NLTP=NL-NT+1
             NLT=NL-NT
             NSN(NL-NT+1)=NSN(NL-NT)
290       CONTINUE
C
300       ICH(1)=ICOUNT(INL)
          INLCH(1)=INL
          PERCF(1)=PERC
          NSN(1)=NSNM
          GOTO 150
C
310       ICH(NL-NT+1)=ICOUNT(INL)
          INLCH(NL-NT+1)=INL
          PERCF(NL-NT+1)=PERC
          NSN(NL-NT+1)=NSNM
          NLTT=NL-NT+1
C
150    CONTINUE
C
       CLOSE (UNIT=9)
C
       COMPER=(100*FLDIM*FLDIM/4096)+.5
       IPERC=IFIX(COMPER)
C
       READ (10,160) NA,(AFILE(I),I=1,NA)
160    FORMAT (Q,11A1)
C
C      a search of one unknown is finished
C
```

```
C       open the file which will contain the results of the
C       search and write the necessary headings
C
        OPEN (UNIT=8,NAME=AFILE,TYPE='NEW')
C
        WRITE (8,303) IPERC
303     FORMAT ( '             ',I3,'% OF THE DATA BASE IS
       1 USED')
        WRITE (8,304) (UNAME(I),I=1,NUNM)
304     FORMAT ( '             ',' SEARCH RESULT OF ',25A1)
        WRITE (8,305)
305     FORMAT ( '   ')
        WRITE (8,320)
320     FORMAT( '             ',' SPECTRUM  ',' NUMBER OF  ',
       1 ' PERCENTAGE  ','   NAME OF   ')
        WRITE (8,330)
330     FORMAT( '             ',' NUMBER   ',' MATCHES   ',
        1 ' OF MATCHES  ',' THE SPECTRUM')
        WRITE (8,331)
331     FORMAT ( ' ')
C
C       start writing the search results with appropriate
C       spectral names
C
        DO 400 II=1,NL
           LLL=INLCH(II)
           WRITE (8,340) INLCH(II),ICH(II),PERCF(II),
       1   (FSAVE(LLL,J),J=1,25)
340        FORMAT ( 12X,I6,6X,I6,8X,F5.1,6X,25A1)
400     CONTINUE
C
        CLOSE (UNIT=8)
C
500  CONTINUE
C
     CLOSE (UNIT=10)
     CLOSE (UNIT=11)
C
     END
```

SUBTRACTION.FOR

```
      program SUB4_FOR
C     !-----------------------------------------------------!
C     ! Fumiko Ishihara                                     !
C     ! /Revised              17-MAY-87 through 1-FEB-89    !
C     !-----------------------------------------------------!
C
C     Second stage comparison process (by subtraction)
C     This program takes an unknowns data file and calculate
C     the difference between the unknown spectrum and the
C     selected reference spectra in the reference library.
C     The results are then stored with appropriated reference
C     spectral names in a descending order.  The Hadamard
C     transformed data set is used for the comparison.
C
      DIMENSION IX(32,32),IY(32,32)
      DIMENSION ICH(5),INLCH(5),NSN(100),ITOTAL(5),ISPECT(5)
C                    ! IX,IY are data matrices
C                    ! ICH = best matches, up to 10 choices
C                    ! INLCH = stores the spectrum number
C
      LOGICAL*1 LUFILE(11),UFILE(11),UNAME(25),UFILE2(11)
      LOGICAL*1 LSFILE(11),SFILE(78,11),SNAME(78,25)
      LOGICAL*1  LAFILE(11),AFILE(11),SFILEI(11),M,MMM,MMMM
C
C     LUFILE contains the filename of the library which
C            contains the filenames and their chemical name
C            of the unknown data
C     UFILE  contains the filenames of the unknown data in
C            LUFILE library
C     UNAME  contains the chemical name of the unknown data
C            in LUFILE library
C     UFILE2 contains the *.txt files which contains the
C            results of the search program.  This is also in
C            LUFILE library
C     LSFILE contains the filename of the library which
C            contains the filenames and their chemical names
C            of the standard data
C     SFILE  contains the filenames of the data in the
C            LSFILE library
C     SNAME  contains the chemical names of the std data in
C            LSFILE library
C     LAFILE contains the filename of the library which
C            will contain the filenames of the results
C     AFILE  contains the result of the subtraction; this is
C            in LAFILE library
C
      PRINT 10
10    FORMAT(' Enter the filename of the library which
     1 contains the unknown data to be compared: ')
```

```
      ACCEPT 22,NLU,(LUFILE(I),I=1,NLU)
      PRINT 12
12    FORMAT(' Enter the number of unknown to be compared: ',
     1 $)
      ACCEPT 14,NUNK
14    FORMAT( I3)
      PRINT 16
16    FORMAT(' Enter the dimension of the data to be
     1 compared: ',$)
      ACCEPT 18,NDIM
18    FORMAT( I2)
C
      PRINT 20
20    FORMAT(' Enter the filename of the library which
     1 contains the results of the search: ')
      ACCEPT 22,NLA,(LAFILE(I),I=1,NLA)
22    FORMAT (Q,11A1)
C
      PRINT 30
30    FORMAT (' Enter the filename of the library which
     1 contains the standard known data: ')
      ACCEPT 22,NLS,(LSFILE(I),I=1,NLS)
C
      PRINT 80
80    FORMAT (' Enter the # of the spectrum in the library: '
     1 ,$)
      ACCEPT 18,NL
C
      OPEN(UNIT=9,NAME=LSFILE,TYPE='OLD')
C
      DO 600 K=1,78
         DO 610 KK=1,11
            SFILE(K,KK)=' '
610      CONTINUE
600   CONTINUE
C
      DO 97 K=1,NL
C
         DO 620 ICHRC=1,25
            SNAME(K,ICHRC)=' '
620      CONTINUE
C
         READ (9,141) NS,(SFILE(K,I),I=1,NS)
         READ (9,142) NSNM,(SNAME(K,I),I=1,NSNM)
         PRINT 625,(SFILE(K,I),I=1,11)
625      FORMAT ( ' ',11A1)
C
97    CONTINUE

141   FORMAT(Q,11A1)        ! get the filenames of the refernce
142   FORMAT(Q,25A1)        ! data and their chemical names
```

```
      CLOSE (UNIT=9)
C
      OPEN (UNIT=11,NAME=LUFILE,TYPE='OLD')
      OPEN (UNIT=10,NAME=LAFILE,TYPE='OLD')
C
      DO 500 III=1,NUNK
C
         DO 630 ICHRC=1,25
            UNAME(ICHRC)=' '
630      CONTINUE
C
         READ (10,160) NA,(AFILE(I),I=1,NA)
160      FORMAT (Q,11A1)
C
         PRINT 640,III
640      FORMAT (' UNKNOWN=',I3)
         READ (11,100) NU,(UFILE(I),I=1,NU)
100      FORMAT (Q,11A1)
         READ (11,102) NUNM,(UNAME(I),I=1,NUNM)
102      FORMAT(Q,25A1)
         READ (11,100) NU2,(UFILE2(I),I=1,NU2)
C
         OPEN (UNIT=8,NAME=UFILE,TYPE='OLD')
         READ (8,110)          ! read the heading from the file
         READ (8,*) ((IX(I,J),J=1,NDIM),I=1,NDIM)
110      FORMAT ( A7)          ! read the data into IX matrix
C
         CLOSE (UNIT=8)
C
         OPEN (UNIT=8,NAME=UFILE2,TYPE='OLD')
C
         DO 94 I=1,5
            READ (8,110)94       CONTINUE
C
         DO 96 I=1,5
            READ (8,95) ISPECT(I)
95          FORMAT ( 12X,I6)
            PRINT 99,ISPECT(I)
99          FORMAT(' SPECTRA NO.=',I6)
96       CONTINUE
C
         CLOSE (UNIT=8)
C
C        *.dif will contain the result of subtraction between
C        the selected reference spectra and the unknown data
C
         INL=0
C
         DO 138 L=1,5
            ICH(L)=0           ! ICH is the # of matches in
            ITOTAL(L)=0        ! order
```

```
            INLCH(L)=Ø
138         CONTINUE
C
            ICOUNT=Ø
C
            DO 15Ø K=1,5
               ICH(K)=Ø
               INL=INL+1
C
               DO 65Ø ICHRC=1,11
                  SFILEI(ICHRC)=' '
65Ø            CONTINUE
C
               DO 66Ø ICHRC=1,11
                  SFILEI(ICHRC)=SFILE(ISPECT(K),ICHRC)
66Ø            CONTINUE
C
               OPEN (UNIT=8,NAME=SFILEI,TYPE='OLD')
               READ (8,11Ø)      ! read the heading from the file
               READ (8,*) ((IY(I,J),J=1,NDIM),I=1,NDIM)
C                                ! read the data into IY matrix
C
               CLOSE (UNIT=8)
C
               INDEX=Ø           ! INDEX is a pointer in the line
               IOLD=1
C
C              comparison starts
C              first, you need to determine whether you want
               the position of matches as well as the # of
               matches
C
               DO 25Ø I=1,NDIM
                  DO 26Ø J=1,NDIM
                     IF (IX(I,J) .GE. IY(I,J)) GOTO 242
                     ITOTAL(INL)=ITOTAL(INL)+(IY(I,J)-IX(I,J))
                     GOTO 26Ø
242                  ITOTAL(INL)=ITOTAL(INL)+(IX(I,J)-IY(I,J))
26Ø               CONTINUE
25Ø            CONTINUE
C
C              sought through the results so that the results
C              will be in an ascending order.
C
               IF (INL .EQ. 1) GOTO 3ØØ
C
               DO 29Ø NT=1,INL-1
                  ICOUNT=ICOUNT+1
                  IF (ITOTAL(INL) .GT. ICH(INL-NT)) GOTO 31Ø
                  ICH(INL-NT+1)=ICH(INL-NT)
                  INLCH(INL-NT+1)=INLCH(INL-NT)
```

```
290         CONTINUE
C
            ICH(INL-ICOUNT)=ITOTAL(INL)
            INLCH(INL-ICOUNT)=ISPECT(INL)
            GOTO 149
C
300         ICH(INL)=ITOTAL(INL)
            INLCH(INL)=ISPECT(INL)
            GOTO 150
C
310         ICH(INL-NT+1)=ITOTAL(INL)
            INLCH(INL-NT+1)=ISPECT(INL)
149         ICOUNT=0
C
150      CONTINUE
C
         OPEN (UNIT=9,NAME=AFILE,TYPE='NEW')
C
C        write the appropriate headings to a file which
C        contains the results
C
         WRITE (9,304) (UNAME(I),I=1,NUNM)
304      FORMAT ( '             ',' SEARCH RESULT OF ',25A1)
         WRITE (9,305)
305      FORMAT ( '  ')
         WRITE (9,320)
320      FORMAT ( '             ',' SPECTRUM  ',' TOTAL VALUE
     1   ',' NAME OF  ')
         WRITE (9,330)
330      FORMAT( '            ',' NUMBER   ',' OF DIFFERENCE
     1  ',' THE SPECTRUM')
         WRITE (9,331)
331      FORMAT ( ' ')
C
C        write the results with appropriate spectral names
C
         DO 400 II=1,5
            LLL=INLCH(II)
            WRITE (9,340) INLCH(II),ICH(II),
     1      (SNAME(LLL,J),J=1,25)
340         FORMAT ( 12X,I6,8X,I8,6X,25A1)
400      CONTINUE
C
         CLOSE (UNIT=9)
C
500   CONTINUE
C
      CLOSE(UNIT=10)
      CLOSE(UNIT=11)
      END
```

REDSHIFT.FOR

```
      PROGRAM RSHIFT_FOR
      !-----------------------------------------------!
      ! Fumiko Ishihara      23-JAN-89                !
      !-----------------------------------------------!
C
C     This program red-shifts a data matrix.

      DIMENSION IX(64,64),IY(64,64)
C     IX = the original matrix
C     IY = red shifted data matrix
C
      LOGICAL*1 LOFILE(15),OFILE(15),AFILE(15)
C
C     LOFILE = filename of the library which contains the
C              original and resultant filenames
C     OFILE = filename of the original data matrix
C     AFILE = resultant filename

      PRINT 10
10    FORMAT (' Enter the filename of the library which
     1 contains the data to be shifted and the resultant
     1 filenames ')
      ACCEPT 22,NLO,(LOFILE(I),I=1,NLO)
22    FORMAT (Q,11A1)
C
      PRINT 80
80    FORMAT (' Enter the number of the spectrum in the
     1 library',$)
      ACCEPT 18,NL
18    FORMAT ( I2)
C
      OPEN (UNIT=11,NAME=LOFILE,TYPE='OLD')
C
      DO 200 II=1,NL
C
         READ(11,25) NSP,(OFILE(I),I=1,NSP)
25       FORMAT (Q,11A1)
         READ(11,25) NAF,(AFILE(I),I=1,NAF)
C
         OPEN (UNIT=9,NAME=OFILE,TYPE='OLD')
         READ (9,*) ((IX(I,J),J=1,64),I=1,64)
30       FORMAT ( 8I8)
C
         CLOSE (UNIT=9)
C
         DO 50 I=1,64
            IY(I,1)=IX(I,1)
            DO 60 J=2,64
               JJ=J-1
```

```
                IY(I,J)=IX(I,JJ)
60            CONTINUE
50          CONTINUE
C
            OPEN (UNIT=9,NAME=AFILE,TYPE='NEW')

            WRITE (9,90) ((IY(I,J),J=1,64),I=1,64)
90          FORMAT (' ',8I8)
            CLOSE(UNIT=9)
C
200       CONTINUE
C
          STOP
          END
```

The vita has been removed from the scanned document