

# Towards a Digital Library of Popular Music

David Bainbridge, Craig G. Nevill-Manning, Ian H. Witten, Lloyd A. Smith, and Rodger J. McNab  
University of Waikato, Hamilton, New Zealand & Rutgers University, New Jersey, USA  
{d.bainbridge,i.witten,l.smith,r.mcnab}@cs.waikato.ac.nz & neville@cs.rutgers.edu

## ABSTRACT

Digital libraries of music have the potential to capture popular imagination in ways that more scholarly libraries cannot. We are working towards a comprehensive digital library of musical material, including popular music. We have developed new ways of collecting musical material, accessing it through searching and browsing, and presenting the results to the user. We work with different representations of music: facsimile images of scores, the internal representation of a music editing program, page images typeset by a music editor, MIDI files, audio files representing sung user input, and textual metadata such as title, composer and arranger, and lyrics.

This paper describes a comprehensive suite of tools that we have built for this project. These tools gather musical material, convert between many of these representations, allow searching based on combined musical and textual criteria, and help present the results of searching and browsing. Although we do not yet have a single fully-blown digital music library, we have built several exploratory prototype collections of music, some of them very large (100,000 tunes), and critical components of the system have been evaluated.

**KEYWORDS:** Music libraries, music representation, melody matching, optical music recognition, MIDI

## INTRODUCTION

From the point of view of the non-scholar, non-specialist, non-technophile, digital libraries have yet to come down to earth. Existing digital library collections cover technical topics, or are highly specialized in particular scholarly domains. Computer and information science is well-represented; as are certain very tightly-circumscribed non-technical research areas. But virtually all current digital libraries are intended for the scholar, the researcher, the specialist. As far as the general public is concerned, digital libraries are obscure and (dare we say it?) irrelevant.

Our ultimate aim is to transcend this stereotype and build a digital library that appeals to a wide cross-section of the community. It goes almost without saying that this is extremely ambitious: it is debatable whether millenia of traditional library development have succeeded in achieving such a goal! However, we have chosen a domain in which digital library technology has many important potential advantages over conventional libraries: music. Music is suitable because it is of interest to a wide cross-section of the community (including youth); it is a (largely) language-independent expression of popular culture. The World-Wide Web is ideally-suited to the delivery of music, and it will not be long before hand-held digital wireless devices free us from the shackles of physical interconnection [8]. Moreover, of all artistic endeavors, music is the one that has benefited most greatly from technological advances: all arrangers, most composers, and many performers use computers routinely in their work. If successful, a popular digital music library would significantly raise the profile of digital libraries and digital library research, which would benefit our community greatly.

This paper describes the progress we have made in realizing such a library. We have built several preliminary prototypes, ranging from a small collection of 1000 jazz tunes through a medium-sized one of 10,000 folk tunes to a massive one of 100,000 MIDI files. (These collections can be accessed at <http://www.nzdl.org/musiclib>.) The small collection is rich: it includes original scanned sheet music, internal representations of all melodies within a music editor, and titles, composers, arrangers, and lyrics (where they exist) for all tunes. From the music-editor representation, tunes can be synthesized into audio, and individual pages of sheet music can be rendered as images. The medium-sized collection contains more tunes but less information about each: it includes internal music-editor representations and song titles; again tunes can be synthesized and sheet music produced. The large one is relatively impoverished in terms of metadata but extremely rich in terms of content. Even titles are not necessarily represented correctly (though they often are); composers' names are mixed up with the title (if they are present at all); there is uncontrolled duplication of tunes (although we remove exact copies); renderings are of extremely variable quality; and the music is usually multiphonic, making it hard to extract themes. Nevertheless its massive size compensates: anyone interested in music

finds this to be a fascinating—and extremely compelling—collection to browse.

Music libraries pose new and interesting challenges for digital library research. We divide them into four categories: acquisition, searching and browsing, presentation, and evaluation. The paper is structured accordingly. First are the problems of acquiring sizable collections of freely available music. There are many possible sources, and we concentrate on three: MIDI files that are publicly available on the Web; music recognition software applied to scanned pages of music; and digitizing raw audio. The prototypes we have built do not address the collection and storage of music in audio form, for this technology is being explored vigorously by others—though obviously comprehensive music libraries will include much information in audio form [5]. However, we do utilize audio representations of musical queries. The second class of problems concern searching and browsing, and coordinated ways of incorporating both modes of access seamlessly. We describe music searching based on sung queries, text searching based on metadata, and combinations of queries on different fields and of audio and textual queries. Browsing very much hinges on the kind of metadata that is available. The third area is presentation: the various forms in which musical information can be communicated to the user. Finally, we consider evaluation. Evaluating music libraries presents interesting challenges. While we have not yet attempted any overall assessment of the utility of our prototype libraries, we have performed individual studies of some of the facilities they provide and the technologies we have developed. This paper describes both our accomplishments and our plans. Since our plans are substantial, instead of outlining future work in a separate section, we have included it in appropriate sections throughout the paper.

Our earlier work in this area was reported at the 1996 Digital Libraries conference [9], where we presented a stand-alone interactive system for retrieving tunes from acoustic input, and the design considerations that underlie it. Since that time, interest in techniques for melody matching has increased greatly (e.g. [7]). We have now incorporated this technique into our digital library software and it forms one of several access mechanisms to the collections described in the present paper. But our focus here is on other, more systemic, aspects of the digital music library. We expect that the next three years will see digital music libraries transformed into a popular end-user technology.

### **ACQUIRING A MUSIC COLLECTION**

The first consideration when constructing any digital library is acquiring the source material. For a music library—particularly a library of popular music—the principal sources are printed and recorded music. A third source, which is particularly important for research libraries, comprises textual information on musical topics—biographies of composers, treatises on music theory, and so on. We omit this from our

discussion because it can be well-handled using existing text-based digital library technology. We also omit consideration of original hand-written music, which, while constituting a treasured source of information in some music libraries, is almost impossible to treat in any way other than as ordinary images.

Acquisition of a sizeable collection of freely available music is not easy because musical data is not readily available in electronic form. We have worked with three kinds of source: automatic conversion of sheet music using techniques of optical music recognition, on-line MIDI files, and existing databases of music. Collections built from these sources will be discussed in turn.

### **Optical Music Recognition**

Optical music recognition (OMR) is the automatic conversion of scanned sheet music to an on-line symbolic form [3]. It provides a flexible approach for building new digital library collections. Although the operator inevitably needs some computer experience, optical music recognition is far less labor intensive than manual entry of music and does not require any specialist music skills.

OMR has been an active area of research since its inception in 1966 [11]. Many systems have been developed with accuracy rates quoted from 89% to “nearly 100%”, and commercial systems have been available since 1993. To assess its potential for creating digital libraries of music, we used it to build an on-line collection of sheet music, selecting as source material a book containing 1,200 popular tunes. Known as a “fake book,” the collection represents a cross-section of frequently requested tunes so that a band can fake a requested tune if they do not already know it. For the most part the music is monophonic (just the tune); guitar chords, lyrics and some bibliographic metadata are also given.

We digitized each page of the book and processed it using CANTOR, an OMR system that we have developed [2]. Figure 1 shows a tiny excerpt from a typical result. On the left is the original scanned music; on the right is the reconstructed score. As you can see, a small error has occurred in the rhythm of the second complete bar, but the gist of the tune is preserved. We quantify the overall accuracy of the process below, under “Evaluation”. OMR is a computation-intensive image-processing operation. Using a 133 MHz Pentium processor, it took around 48 hours to process the 1,200-tune collection.

Because the computer now has the music represented symbolically rather than pictorially, it is possible to manipulate it in musical terms. Reconstruction of the image is just one example: the tune can also be played back, its key can be altered, it can be searched for musical motifs, and so on.

The images also contain textual information: the title of the tune, composer, arranger, lyricist, and the lyrics themselves (as well as various musical annotations such as tempo and



Figure 1: Application of CANTOR to convert an excerpt from the Fake Book into symbolic form from which the score has been reconstructed.

chord sequences.) Although the technology of OCR is well developed, no practical OMR system yet includes the ability to recognize text occurring amidst the music. Because we wanted to explore how a rich, high-quality, digital library collection might be used, textual information was entered manually, by a secretary. However, musical annotations and chord sequences were not included. Although in a future version it would be worth capturing chord sequences and generate guitar tabulature from them as an alternate form of output, this was not deemed to be worth the labor in our prototype system. Examples from the resulting digital library collection are shown in Figures 2, 3, and 7; they will be discussed later.

#### Acquiring MIDI Files

MIDI (Musical Instruments Digital Interface) is a standard for controlling and communicating with electronic musical instruments. It represents music as timed events that determine note onsets and offsets, and includes a standard representation of the Western musical scale for specifying pitch. The music is polyphonic: different channels can be used for different instruments, and notes can be played simultaneously on the same channel to produce chords. In addition, timed events can be specified that contain ordinary text. Although there is no associated metadata standard, these are commonly used to name the song, to name the musical instruments associated with each channel, and even to include textual lyrics that are correctly positioned relative to the music. Other events include instrument changes, key and time signature changes, and binary downloads for particular hardware devices.

An astonishingly large and diverse array of MIDI files are available on the World-Wide Web, for a huge variety of music: popular, rock, classical, and jazz, as well as many more specialist genres. It can be argued that the selection of music represented on the Web provides a faithful reflection of popular music tastes. By and large, MIDI files are created by amateurs who choose to enter music that they admire. Of course, music entry is labor-intensive and requires both music and computer skills, and people think carefully about what pieces of music are worth spending their time and energy on. The resulting files are highly variable in musical quality, in the textual metadata that is included, and in the degree of Web site organization. Nevertheless, the result is a very significant musical resource that is continually and autonomously growing.

Many MIDI sites store several thousand files (for instance [www.midiphiles.com](http://www.midiphiles.com)) and our investigations with Internet search engines demonstrate that there are hundreds of thousands of such files scattered all over the world. There is a useful distinction between music files gathered from an established MIDI site and ones collected from the Web at large. In general, the former is more amenable to the construction of a quality digital library because it already encapsulates the crucial notion of selectivity and distillation; however, collection sizes are typically rather small. Conversely, the Web at large offers a vast quantity of music files, but they are unorganized, vary widely in quality, and include duplicates in various guises—only a few of which can be detected automatically.

We have gathered examples of both types and built sample collections from each. Downloading files from a single site is straightforward. We identified a well-organized site containing approximately 1,200 MIDI files; in this case acquisition was trivial since the data was already grouped into a handful of archived files that could be downloaded manually. In this site, composers and names of tunes were clearly identified by the file names, which even included explicit spaces, and we used these to build composer and title indexes.

Examples from the resulting digital library collection are shown in Figures 5–6; again, they will be discussed later.

Much larger collections of MIDI files could be obtained from the Web at large by modifying a Web crawler to download MIDI files, whose filenames invariably have the extension `.mid`, and starting it off from a handful of MIDI sites—which invariably include links to other MIDI sites. As with any Web indexer, there is no need to retain the files centrally once they have been located and indexed, though doing so usually improves performance. However, there is an easier way. The Hotbot search engine ([www.hotbot.com](http://www.hotbot.com)) can locate pages that include links to files with certain extensions. This provides a simple way to locate MIDI files, and Hotbot reports that 315,000 pages contain links to such files. Unfortunately, search engines limit the number of hits returned by a single query; in the case of HotBot the limit is 1000. Our solution is the following: we ask for all MIDI files on pages that include a particular word, chosen at random from an online dictionary. For example, there are 77 pages that contain MIDI files and the word “abduct.” We repeated this procedure many times, choosing a different random word each time. Most queries produce hits, even for unusual words, be-

cause of the stemming that Hotbot performs. The number of novel pages returned by each query starts to decrease as more pages files are gathered. We stopped gathering files after attempting to download 120,000 pages (13,000 web page links were unavailable). This produced links to 325,000 MIDI files, of which 36,000 were unavailable, leaving 289,000. We then removed files that were *exact* duplicates, yielding 99,000 files. However, the remaining collection still contains duplicates. For example, there are 25 different arrangements of J.S. Bach's *Jesu, joy of man's desiring*, and 27 arrangements of the Beatles' *Yesterday*. Each MIDI file consists of multiple channels that are assigned to different instruments: typically including piano, bass, drums and strings. The 99,000 MIDI files contained an average of 7.4 channels per file, for a total of 740,000 channels. The tunes contain 528 million notes, or about 700 notes per channel. Reasonable estimates for tempo and notes per bar gives an estimate for the length of the average tune of about 5 minutes.

Obtaining textual metadata for MIDI files taken from the Web at large presents a greater challenge than for a single well-organized site. Currently, we extract all text from the textual MIDI events, and include the filename too. The resulting list often includes composer, performer, and/or title—or snatches from the title—as well as some other information—instrument names, chord symbols, lyrics—that may or may not be easy to interpret. When such information is displayed on a query response page, the result is not unlike that of present-day search engines, which show the first few characters of each hit—sometimes meaningful, sometimes garbled. The metadata also includes the URL of the tune, or several URLs if there are multiple copies on the web. This textual data amounted to 130 Mb, of which 108Mb was URLs. This leaves 22 Mb of metadata from within the MIDI files, for an average of 220 bytes per tune. Like search engines, clicking each member of the list rapidly brings up more meaningful information, except that in the case of MIDI files the information is an audio rendition of the tune, which has the advantage of not interrupting the user's visual context. Sound replay begins very quickly and is instantly interrupted by clicking elsewhere; thus it is very quick and easy to scan the list despite the fact that the textual information is imperfect and sometimes incomprehensible.

### Existing Music Databases

Though scarce, databases containing musical material do exist. Two such examples are the Digital tradition [6] and the Essen database [12], both of which have built up, by hand, a large collection of folksongs collected from Britain, Ireland, North America, Germany, and China. As another example of a digital music library collection, we have combined these two sources to form a collection of over 10,000 public domain songs that can be searched melodically, textually or both in combination. No examples of this collection appear here because it has been described in a previous paper [9].

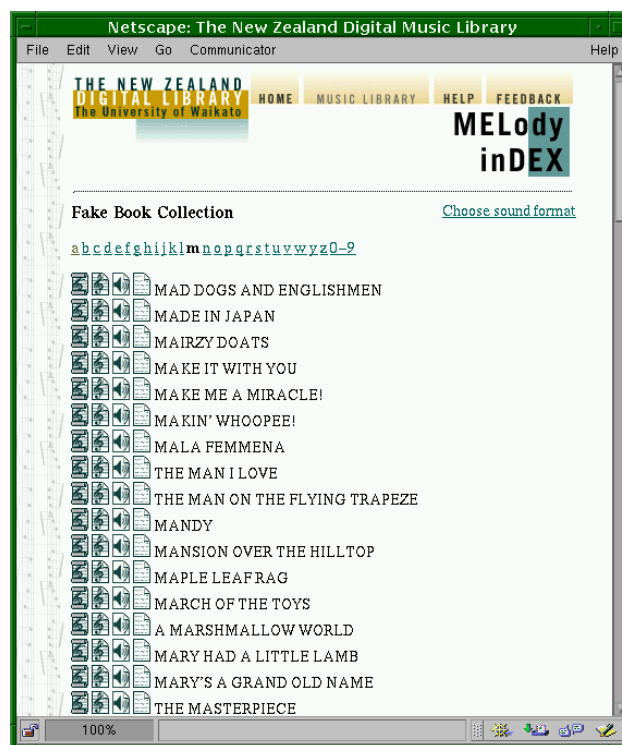


Figure 2: Browsing the Fake Book collection by title.

### BROWSING AND SEARCHING

Although searching gets the lion's share of research attention, because it presents interesting technical challenges—even more so in music than in textual domains—we believe that browsing is an equally important means of accessing digital libraries so far as the end user is concerned. Rich browsing possibilities are contingent on rich metadata. Music stores offer shoppers intriguing visual representations of particular recordings in the form of CD covers, and engaging textual information in the form of accompanying notes. They arrange their stock by broad category (popular music, rock, jazz, classical music, etc), and, within each category, by composer or performer. Most clients find it easy to pass enjoyable hours simply browsing the collection.

By comparison, the browsing facilities that we currently provide are relatively impoverished. Figure 2 shows a screen from the Fake Book collection where the user is browsing alphabetically by title. The icons beside each title give different ways of presenting the full song: the original page image (represented by a scroll); the reconstructed score, rendered from the internal music representation; an audio synthesis of the music; and a display of the textual data associated with the song—title, composer, arranger, lyricist, and lyrics. Similar indexes could permit browsing by composer and arranger.

It should be possible to obtain more descriptive, critical, and

anecdotal information on particular pieces and recordings of music from the Web, which is a rich source of musical information, opinions, catalogs, reviews, factoids, and miscellaneous trivia. To do so requires new techniques of information mining. We are working on these [13], but they are still in their infancy and we have not begun to apply them to harvesting information about music. Moreover, items should be cross-referenced to other musical sites.

Complementing browsing is the ability to target queries more directly through searching. Current public libraries that offer computer access to musical data are text based, and operate at the level of standard bibliographic metadata: we can provide more fine-grained access. Moreover, given the nature of the data, text is not always the most natural form for a query, and we allow melody-based querying too.

### Text-Based Querying

Textual queries to the digital music library are accomplished using our existing digital library software, which is based on full-text retrieval. An index is built to all the textual information associated with each piece of music; in addition, subsidiary indexes are built to subsets of the information to provide fielded queries with partial matching. Depending on the collection, the indexes are formed from information entered by hand (for example, the Fake Book collection), or extracted automatically (for example, the MIDI collections). The indexing process is the same regardless of the source of text.

Suppose a user recalls a song by George Gershwin that mentions “feet” (a typical, though non-academic, query!). Figure 3 shows the result of searching the Fake Book collection for a composer whose name matches “Gershwin” and for lyrics that include the word “feet.” These two words are sought in separate indexes—composer and lyrics respectively—and the results are combined using the same mechanism that is used to combine text- and music-based queries (described below). In this example, *Fidgety Feet* comes top of the list with a 100% match since it includes both query terms. Other matches have lower scores since they only include the composer “Gershwin.”

### Melody-Based Querying

Depending on circumstances, a more natural form of query might be to sing, hum, or whistle it—or enter it on a music or computer keyboard. Music librarians are often asked to find a piece of music based on a few hummed or whistled notes. The melody indexing system that we developed earlier is capable of interpreting audio input as a sequence of musical notes and searching for that sequence in a database of melodies [9]. Originally a Macintosh based program, we have ported our melody indexing system to run under Unix [10, 14] within the same software framework as our text-based digital library system.

The first stage in the process is to transcribe the acoustic

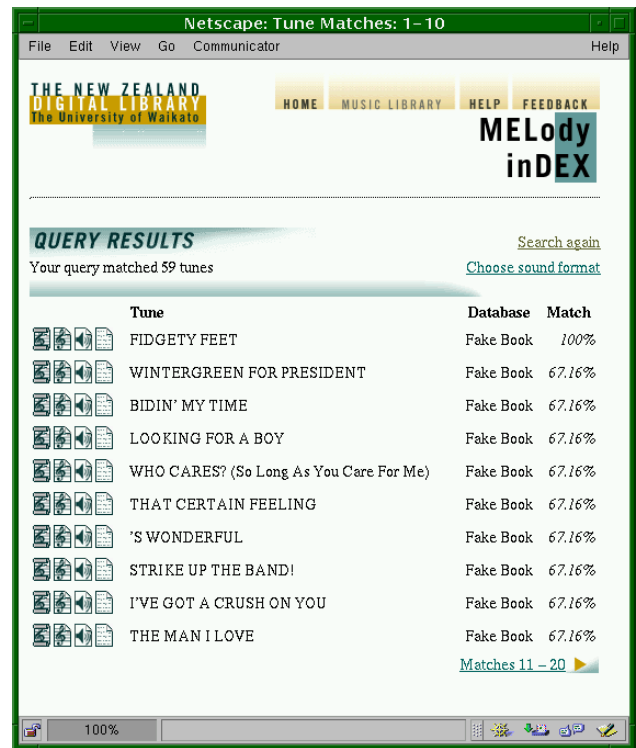


Figure 3: Ranked results from searching the Fake Book for songs written by Gershwin and lyrics that contain the word “feet.”

query into symbolic musical notes. The left-hand side of Figure 4 shows the acoustic profile of a rendition of the first three notes of *Three Blind Mice*. We perform a frequency analysis of this input using a standard pitch tracker and segment the notes based on the amplitude profile to generate the musical representation at the right of the figure. This stage is, of course, unnecessary if the input is entered on a keyboard or other MIDI instrument—which is also likely to be a popular mode of access. The next step involves measuring the similarity between two melodies, a research problem that has received much attention recently [7]. We search through the database of melodies using an approximate string matching algorithm based on dynamic programming to determine the degree of match of each melody to the input. Error-tolerant matching of queries to tunes is essential for several reasons:

- the input is noisy: pitch tracking can fail and people often sing badly;
- people do not remember tunes exactly;
- most melodies have several different versions—in particular, rhythmic variations; and
- the database may contain errors, especially if melodies have been acquired using OMR from difficult source material.

Melody retrieval using dynamic programming involves calculating the edit distance from the query to every melody in the database. While this works well for small databases (up

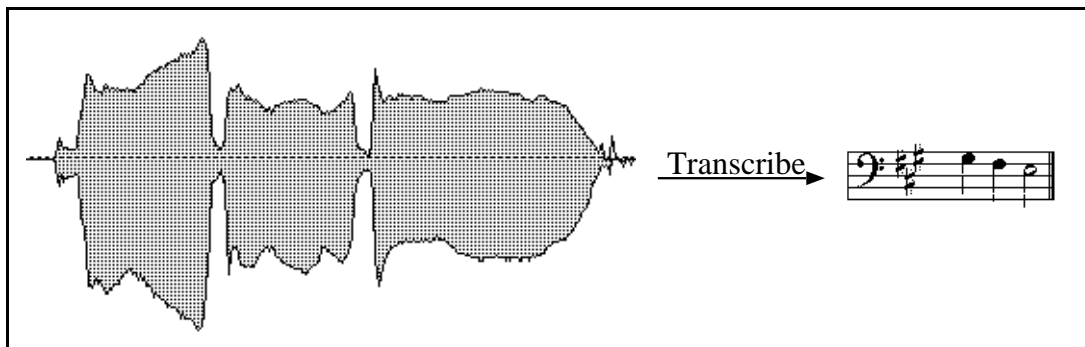


Figure 4: Converting a sung sample from audio to symbolic notes.

to several tens of thousand tunes), it scales badly. Dynamic programming is quadratic in the size of the query, which does not usually cause a problem because typical queries are quite short. But the main problem is the linear scan of the database. Because of the nature of edit distance, simple inverted indexes cannot be computed in advance as they are for textual queries. We intend to adapt techniques from another field, bioinformatics, where fast approximate matching is used for DNA and protein sequence matching millions of times daily. The most popular technique for this kind of search is BLAST [1], which utilizes a trigram index, a variation of an inverted index, that returns a subset of the database most likely to have low edit distance from the query string. Further heuristics are applied to window this set further until full dynamic programming is performed to obtain exact rankings for the remaining hundred or so sequences.

Music matching differs in several important respects from the protein/DNA problem. The music alphabet consists of about 84 absolute pitches (seven octaves, quantized into semitones), or 60 intervals, that is, relative pitches (two and a half octaves, both up and down)—as opposed to 20 amino acids or 4 nucleotides. The substitution cost of one interval or pitch for another depends on their distance simply defined as an arithmetic difference (though more subtle measures are possible), whereas with protein matching, the substitution matrix must be experimentally determined. In music searches, the query will usually be much smaller than the resulting tunes: a user is most likely to enter the beginning couple of bars, or a prominent theme. In bioinformatics, the queries are usually the full sequence of a gene or protein, so the query size is similar to the retrieved sequence. This is likely to affect the tradeoff between the size of the index and the guarantees that can be placed on the recall and precision that it offers.

A useful side-effect of using technology from bioinformatics in a music library is the ability to use clustering and phylogeny techniques developed for determining evolutionary relationships. These relationships certainly exist in music, from close neighbours (different versions of the same tune), through tunes that share motifs (either by inspiration or plagiarism), to distant relationships such as the influence of jazz

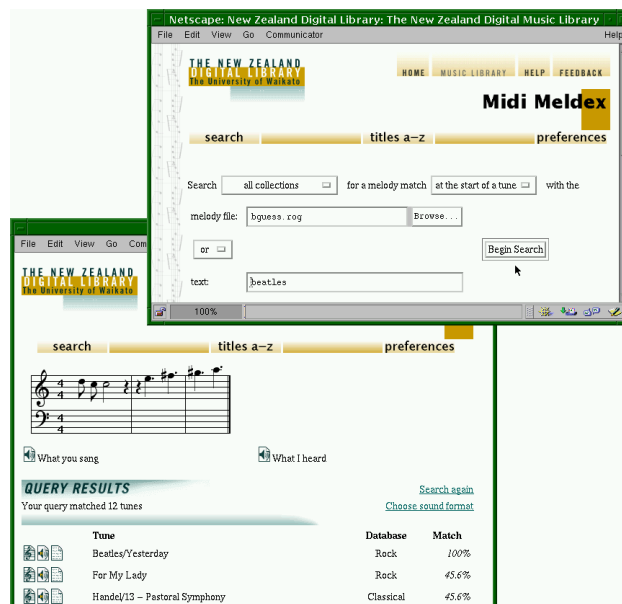


Figure 5: Result of searching for MIDI files containing the word “beatles” combined with a melody match to the opening refrain of *Yesterday*.”

on popular music. In short, although future research is necessary to make melody retrieval into a practical technology for large-scale music libraries of the kind that we envisage, there are strong indications that this will be possible.

### Combined Searches

Text searches can be combined with melody matching to yield a more comprehensive search technique. For example, Figure 5 shows the standard query page for the collection formed from a single MIDI site (approximately 1,200 tunes). As specified on a “preferences” page (which is not shown but can be accessed from a button at the top of the search page), text matching is case insensitive with stemming disabled. Melody matching is also subject to a set of options not described here (see [9]): in this case it is restricted to the start of each track (ignoring leading rests), compares the interval between notes, and ignores note duration.



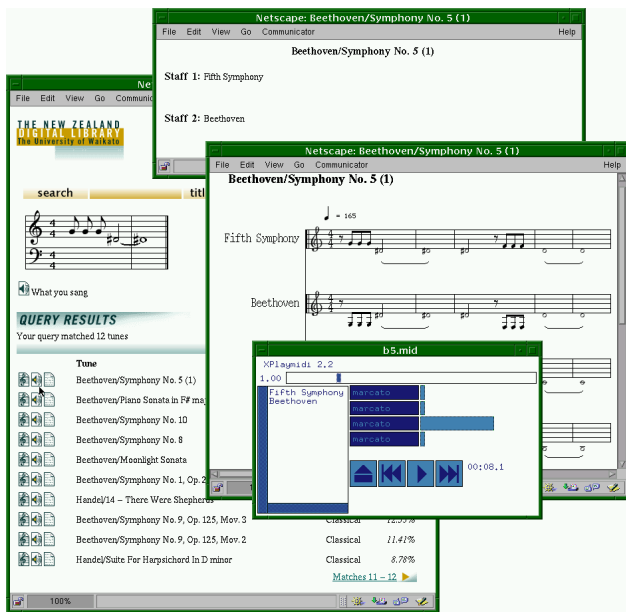


Figure 6: Searching the MIDI collection for Beethoven's Fifth Symphony.

In this case, the textual search is for the word “beatles,” and a melody has been sung that resembles the first few notes of the tune *Yesterday*. The music displayed in Figure 5 is the computer’s rendition of the user’s sung input: note incidentally that the rhythm of the notes is disturbed because the output module, which resynthesizes the music-editor notation into a GIF image, has assumed, incorrectly in this case, that the tune starts at the beginning of a bar. This does not affect melody matching. From the results page, part of which is shown in Figure 5, an item in the collection can be viewed in various forms, symbolized using icons on the left-hand side: leftmost is a link to the MIDI file reconstructed as sheet music, next the MIDI file itself (resulting in audio playback if the browser is configured appropriately), and finally an HTML page presenting the text extracted from the MIDI file.

Combination searches are implemented by performing separate searching for the appropriate material—full-text search using the appropriate index for text, and melody matching using the specified options for music—and merging the results together. Match scores are scaled to express them as a percentage in each case, and then documents with scores present in more than one index are added together. Finally, *all* scores are divided by the number of indexes searched, and the list is sorted for presentation.

### PRESENTATION OF MUSIC DATA

Our prototype digital music library systems call for many different types of information to be presented to the user. To illustrate these, two further examples appear in Figures 6–7. Figure 6 shows a query to the small (1,200 item) MIDI collection, where the famous opening bars of Beethoven’s Fifth

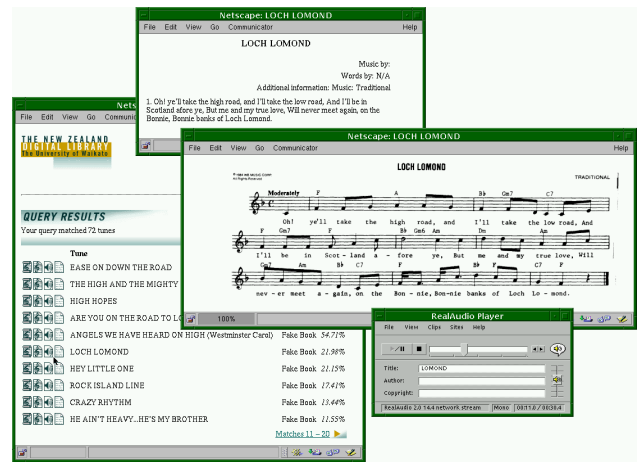


Figure 7: Searching for the folksong *Loch Lomond*.

Symphony have been sung. The correct work appears at the top of the query results list. The user has asked for the extracted text to be shown (it happens to contain the name of the work and the composer as textual events associated—for no apparent reason—with the first two staff lines), the music to be displayed (it is rendered directly from the MIDI file), and the MIDI file to be played (controlled from the small player window at the front). Figure 7 shows a second example of a search session, this time with the 1,200-item Fake Book collection. The text “high road,” a remembered phrase from a particular song, was entered and searched across all text fields (title, composer, lyrics and additional information). The query returned 72 matches, with *Loch Lomond* (the song being sought) appearing sixth in the list. The user has opened the text file and seen words that appear familiar, called up the sheet music which helps to confirm that the document is the one sought, and finally requested it to be played.

Standard technology for music conversion and display has allowed us to create these prototypes fairly easily. However, music presents different challenges from text and graphics, chiefly because it is time-based. We plan to tackle two issues as we develop our music libraries further. The first is how to present a list of hits to the user. Currently, the system only displays metadata such as title and composer. In some cases, however, the user might like to see and hear the matching notes in the context of the tune, analogous to a textual keyword-in-context display. The hits and their context should be short, so that the user can hear a number of them quickly to decide which one is the desired tune. Excerpting parts of MIDI is not completely straightforward, because the rendition of an excerpt is dependent on earlier events such as parameter changes and instrument assignments.

We expect to encounter new challenges in designing the user interface to the collection of tune fragments. Should they be in the same key, to aid comparison? Is there a sensible

visual representation that can be used to obviate listening to all the samples? Standard musical notation might be a little cumbersome and slow to interpret. Can we show an overview of the piece, showing where the hit occurs in context? Users are accustomed to scanning a list of textual hits, spending a small amount time on each one. They should therefore be able interrupt the playing of one fragment quickly when they decide that it is not of interest. We may be able to draw on results of interfaces to other time-based media such as video [4] and apply them to MIDI music.

The second issue is how to present the full tune to the user. It is often difficult to determine the relevance of a hit from an excerpt. In text, the user can retrieve an entire document and scan it very quickly, but the musical analog of scanning is not obvious. The tune could be played back at a much faster rate, but this is likely to produce something completely unintelligible. Dropping notes would again destroy the tune. Clearly, some kind of music summarization is necessary. Perhaps salient parts of the music can be detected automatically. For example, repeated motifs might indicate a theme in the music. The form of the music (*e.g.* 12-bar blues) might indicate where the signature melodies occur. Also, aligning the sequence of notes with itself as is done for protein and DNA sequences should reveal internal near-repetitions that indicate choruses and verses. This analysis would provide delimiters for the various sections that can then be used to identify salient phrases. As we develop the music libraries, we plan to turn these speculations into concrete features and conduct experiments to answer these questions.

## EVALUATION

**Accuracy of OMR.** Statistical analysis was performed to establish the accuracy of the OMR operation. All pages were printed out with the stafflines detected by the computer electronically removed. No erroneous examples were encountered, confirming that staffline location was 100% correct for the 600 page book. A page of music, however, does not necessarily constitute a tune. In the case of the Fake Book, tunes are often short enough to fit more than one to a page; other tunes can span more than one page. Using a heuristic based on the location of “title sized text,” pages of music were restructured as separate tunes. The process was not infallible. Displaying each separated/collated tune revealed five erroneous documents. These were corrected by hand before proceeding to the OMR stage.

Tabulating the accuracy of subsequent OMR processing steps is labor intensive [2]; consequently we chose 20 reconstructed scores at random, and studied those. Correcting these scores *completely* required 348 editing operations, or (alternatively) 17 errors for every 100 notes played: a sizeable task. More effort could be expended customizing CANTOR to the particular fonts used in the Fake book, however because only a small fraction of these errors are crucial to melody matching (which is already an approximate comparison) the data

used in building the collection was left uncorrected. Based on mistakes affecting duration and pitch, 9 operations per 100 notes are necessary to correct durational mistakes, and only 1 operation per 100 notes for pitch, suggesting that the default values for the Preferences page (“match by interval” and “ignore duration”) are a sensible choice.

**Query sizes in large collections.** How long an excerpt will a user need to enter to uniquely identify a particular tune in a large MIDI collection? This is not entirely straightforward to estimate. There are 528 million notes, that is, places where an excerpt might match. If the alphabet consists of about 60 intervals, then it will take  $\log_{60} 528 \text{ million} = 4.9$  notes on average to form a unique substring.

However, the intervals are not uniformly distributed. In fact, the zero-order entropy (*i.e.* assuming notes to be *i.i.d.*) of the notes is only 4.5 bits each, so 6.5 notes would be required. Of course, intervals are not independent: many channels exhibit a high degree of regularity, especially percussion and bass. To quantify this, we compressed the sequence of notes using *gzip*, which takes account of repeated strings of notes. This gives an entropy estimate of 1.6 bits/note, indicating a significant amount of repetitive structure in the sequences. At this entropy, 18 notes would be required to uniquely specify a subsequence.

This is too many to expect a casual user to enter; however, it is a pessimistic estimate. From manual inspection of the tunes, we believe that most of the regularity is in the bass and percussion channels, and a user is unlikely to search for tunes in non-melody channels. Deleting all but the melody line for indexing purposes should result in an improvement from two sources. First, the number of notes will decrease by about a factor of seven. Second, the entropy of the melody will be higher. We believe that these two effects will reduce the notes required for a specific query to a reasonable number.

One factor that we are unable to estimate is the entropy of queries: it is likely that salient tune fragments that users remember have higher than average entropy. This would reduce the necessary query length still further; user studies will be necessary to quantify this effect. We have also ignored rhythm in these calculations. If rhythm were taken into account in matching (which is an option in our matching technique), fewer notes would be required. Because this adds extra complexity to the pitch tracker and matching procedures, we will only take advantage of rhythm if it is clearly necessary.

**Melody retrieval.** Our earlier experiments on how accurately people sing well-known melodies [9] have been used to establish design criteria for the melody retrieval component. Among the findings were that subjects readily add or delete notes corresponding to syllables in the lyrics, that singers tend to compress wide intervals and stretch small ones, that they frequently exhibit gradual pitch drift, and that



they frequently begin singing at the song's "hook" (a memorable line designed to capture listeners) instead of at the beginning. We have taken these into account in the approximate matching strategies that are implemented and placed under the user's control on the "Preferences" page.

## CONCLUSIONS

We have described two prototype digital music libraries: one small, high quality collection created using optical music recognition and manual text entry, and one large, low quality collection derived from a web crawl for MIDI files. Each collection has strengths that we would like to reproduce in a future production-quality digital music library: on the one hand quality metadata, searching by sung input and output in multiple formats, and on the other hand, breadth of content. Each has stretched our current technology and demonstrated the need for further research in acquisition, searching and browsing, presentation and evaluation.

The future for digital music libraries seems bright. There is already enormous popular interest in music on the Web. For example, Diamond Multimedia recently caused a furore in the music publishing industry by marketing a low-cost compact portable music player capable of playing high-quality music files in the MP3 format downloaded from the Internet. Lycos, a large search engine, has announced a search engine for MP3 files (<http://mp3.lycos.com>), and has indexed 500,000 of them. The existence of such an index is proof enough that digital music libraries are an idea whose time has come.

Even more interesting is the motivation for providing the index: search engine queries for "MP3" are almost as popular as "sex". Assuming that a pornographic digital library is beyond the purview of academic research, music seems to be the best route to a truly popular digital library. The inclusion of digitized recordings would certainly be a tremendous asset to a digital music library, although there are many important, and difficult, issues of intellectual property to be considered and resolved, particularly when dealing with commercial recordings by well-known artists. However, we have identified an arena in which commercial recordings are not an essential component. For example, it is highly likely that the creators of MIDI files would readily give permission for them to be included in a larger collection, providing due credit was given, and that music publishers would find it in their interests to allow wider access to the works they produce, providing access to the music images was suitably controlled and there was some provision for buying paper copies directly from publishers.

In summary, we believe that music will constitute a "killer app" for digital libraries.

## REFERENCES

1. S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped blast

and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

2. D. Bainbridge. *Extensible Optical Music Recognition*. Ph.D. thesis, Department of Computer Science, University of Canterbury, NZ, 1997.
3. D. Bainbridge and N. P. Carter. Automatic reading of music notation. In H. Bunke and P. S. P. Wang, editors, *Handbook on Optical Character Recognition and Document Image Analysis*, pages 583–603. World Scientific, Singapore, 1997.
4. M. Christel, D. Winkler, and R. Taylor. Multimedia abstractions for a digital video library. In *Proceedings of ACM Digital Libraries '97 Conference*, Philadelphia, PA, July 1997.
5. J. W. Dunn and C. A. Mayer. VARIATIONS: A digital music library system at Indiana University. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, 1999.
6. D. Greenhaus. About the digital tradition. <http://www.deltablues.com/>, 1992.
7. W.B. Hewlett and E. Selfridge-Field, editors. *Melodic Similarity: concepts, procedures and applications*. MIT Press, 1998.
8. G. Lawton. Vendors battle over mobile-os market. *IEEE Computer*, 32(2):13–15, 1999.
9. R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham. Towards the digital music library: tune retrieval from acoustic input. In *Proc Digital Libraries '96*, pages 11–18, 1996.
10. R.J. McNab, I.H. Witten, and S.J. Boddie. A distributed digital library architecture incorporating different index styles. In *Proc. IEEE International Forum on Research and Technology Advances in Digital Libraries*, pages 36–45, Santa Barbara, California, 1998. IEEE Computer Society Press.
11. D. Pruslin. *Automatic Recognition of Sheet Music*. Sc.D. dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, June 1966.
12. H. Schaffrath. The esac databases and mappet software. *Computing in Musicology*, 8, 1992.
13. I.H. Witten, Z. Bray, M Mahoui, and W. Teahan. Text mining: a new frontier for lossless compression. In *Proc. Data Compression Conference*, Snowbird, Utah, 1999. (to appear).
14. I.H. Witten, R.J. McNab, J. Jones, M. Apperley, D. Bainbridge, and S.J. Cunningham. Managing complexity in a distributed digital library. *IEEE Computer*, pages 74–79, February 1999.