# **Moodtrack:** practical methods for assembling emotion-driven music

**G. Scott Vercoe**
B. Mus. Jazz Composition
Oberlin Conservatory of Music
June 1995

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
**Masters of Science in Media Arts and Sciences** at the
**Massachusetts Institute of Technology**
September, 2006

Author: **G. Scott Vercoe**
Program in Media Arts and Sciences
August 14, 2006

Certified by: **Walter Bender**
Senior Research Scientist
Program in Media Arts and Sciences

Accepted by: **Andrew Lippman**
Chair, Departmental Committee on Graduate Studies
Program in Media Arts and Sciences

# **Moodtrack:** practical methods for assembling emotion-driven music

**G. Scott Vercoe**

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, on September, 2006, in partial fulfillment of the requirements for the degree of Masters of Science in Media Arts and Sciences

## Abstract

This thesis presents new methods designed for the deconstruction and reassembly of musical works based on a target emotional contour. Film soundtracks provide an ideal testing ground for organizing music around strict timing and emotional criteria. The approach to media analysis applies Minsky's ideas of frame-arrays to the context of film music, revealing musical and emotional cue points of a work. Media deconstructions provide a framework for the assembly of new musical material, leading to the creation of soundtracks using musical commonsense.

Thesis supervisor: Walter Bender, Senior Research Scientist

# **Moodtrack:** practical methods for assembling emotion-driven music

**G. Scott Vercoe**

The following people served as readers for this thesis:

**Martin Marks**
Senior Lecturer in Music
Massachusetts Institute of Technology

**Richard Boulanger**
Professor of Music Synthesis
Berklee College of Music

*Acknowledgements:* I am grateful to the Media Lab faculty, staff and students for supporting my work with their time, ideas and conversation. My readers have offered invaluable help: Walter Bender for his patience, guidance and breadth of knowledge; Richard Boulanger for being a mentor with incredible insight and belief in a 'future-proof' world of music; and Marty Marks for kindly sharing his knowledge, love and philosophy of film music. I'd like to thank a few members of the Media Lab community: Henry Lieberman, for helping me begin my journey; Push Singh, for his visionary research and unforgettable kindness; Roz Picard, for opening my eyes to the study of emotion; Whitman Richards, for helping me see a creative path; 'popes' Bo Morgan and Dustin Smith for being there through the best of times and the worst of times; Jae-woo Chung, for inspiring collaborations; Ian Eslick, Hugo Liu, Hyemin Chung, Mako Hill, Brian Whitman, Tristan Jehan, Alex Vasilescu, Cameron Marlowe, Erik Blankinship, Joanie DiMicco and everyone else who has been kind to me throughout my time at MIT. Thanks to Ken Bailey, Lydia Harrell, Kemp Harris, Mike Caglianone, Jim Lucchese, Brian Verocchi, Greg Ogan, Denpo & Ninka, Tiffany Harris, Dave Kudrowitz, Adam Oh, Justin Hadley, and all my partners in creative endeavors—the whole Zen Bastards/Hyptonic/Fade-Outs/Rollover Baby/Neon Grandma extended family—it is my musical roots that make this all meaningful. Special thanks go to Dane Scalise, for being my biggest fan and the best editor anyone could ask for; his friendship and input contributed a large part to this thesis. Extra special thanks go my father Barry Vercoe for his love and wisdom, for believing in me, being there, and never letting me think that there's anything I can't do. Lastly, I'd like to thank the great American composer, Elizabeth Vercoe, who I am lucky to call my mother. She is the strongest, kindest, most inspiring and wonderful person I have ever known. Her love and devotion has enabled me to pursue my dreams, so naturally this thesis is dedicated to her.

# **Moodtrack:** practical methods for assembling emotion-driven music

**G. Scott Vercoe**

# 1. INTRODUCTION

N.B.: Although much of the terminology in this thesis is used in the study of film music, some new terms are introduced. The glossary in Appendix 1 defines terms written in **boldface**.

## 1.1 Methodology

This research explores the analysis and re-synthesis of musical narrative, specifically **non-diegetic** film music scoring. Software tools are developed as an approach towards emotion-driven soundtracks. To generate a score to a section of film comparable to the original scoring, segments of sound are retrieved and rated using a concise set of musical features. The segments are organized and assembled to match the musical contour of the original score. Alternatively, Moodtrack can take emotion-cue points as input, and use data relating musical features to emotions for retrieval, rating and assembly. Individual components of the system include on-line collection of music-emotion data, soundtrack annotation and music-assembly techniques. The goal is to demonstrate multiple components working together to build emotion-driven soundtracks, effectively **re-purposing** a film's musical score for use in another context.

## 1.2 Hypotheses

Music Information Retrieval (MIR) research addresses the analysis, classification, low-level feature-extraction, storage, index and retrieval of music. In MIR, natural language is primarily used for low-level description of audio features, not as a means for describing upper-level semantics such as aesthetic features, context and mood. The semantic gap refers to the rift between high-level meaning and low-level features (Dorai et al., 2003). The research here is offered to help bridge the semantic gap in the study of film music with an approach to musical and emotional narratives. Three hypotheses are tested in this thesis:

**1.** Annotation software can assist a music editor to assess a film's musical and emotional content.

2. Comparable sections of music can be realistically assembled by matching musical features.
3. Data correlating emotions to musical features can be applied to automatic assembly of emotion-driven music.

### 1.3 Background

The modern film music professional may theoretically have immediate access to hundreds of original compositions, but retrieving music appropriate for a specified mood, scene or setting is a constant issue. In many instances, appropriate film music to accompany an important scene may sit forgotten deep in a file system, for which traditional search methods are ineffective. Entertainment conglomerates owning film production studios consistently draw from their in-house music division's catalog for licensing benefits. For that reason, intimate knowledge of a very large catalog is vital for successful music retrieval.

For each new project, composers, **music supervisors**, editors and other and music staff members support the director's vision of the film's emotional and musical requirements. Every genre of film (horror, comedy, action, etc.) has its own musical sensibility with which the music team must be intimately familiar. A film director may request a specific mood, the setting may imply a music genre, and visual events may determine musical cue points. The restrictions can become extensive and complex, requiring the music staff to form a representation of the musical requirements. Even if a composer carefully organizes musical material, it can be difficult to recall each associated mood.

In the last two decades, research on automatic extraction of musical features from audio signals has made significant progress (Yang et al., 2004). Machines can extract instrumentation, performance parameters, musical form, tempo, dynamics, rhythm and genre, to varying degrees of accuracy. However, despite the recent development of these methods, surprisingly few creative ways of using the resulting data have been explored. Most importantly, the set of extractable musical features may not correspond to an ideal set of aesthetic features used by composers and directors.

# 2. SYSTEM OVERVIEW

## 2.1 Approach

Moodtrack consists of a set of music-annotation and music-arrangement tools. In short, annotated music is used to build sections of music that satisfy a supplied set of criteria. While the research area covers music-emotion and film analysis, Moodtrack's tools are designed for film music professionals, such as supervisors, editors, composers and directors. The low-level and technical nature of supplying the annotation and required criteria may deter the casual user. However, this thesis is offered as a step towards creating automatic systems that can provide context-driven audio with musical common sense.

## 2.2 Input data

Because Moodtrack involves the arrangement (not composition) of musical sections, a music library is required. Adding a soundtrack to the library requires manual segmentation of soundtracks into short musical phrases, saved as audio samples.

Music and emotion data is collected from two sets of volunteers, 1) players of an online game, and 2) experts annotating musical stimuli. The online game collects general data correlating musical features to emotions, while the expert media-analysis provides a systematic musical and emotional annotation.

To assemble music, the user provides a set of criteria describing the contour and timing of a desired section of music, and sets parameters controlling the type and operation of the music-arranging technique. For music retrieval and assembly, the system requires data representing musical feature levels, contour and timing. If an annotated musical score exists, musical features can be used as the target **temporal musical feature cue**. If a musical score for the scene does not exist or has undesirable features, then emotion levels, contour and timing (**temporal emotion cues**) can be provided which are mapped to the necessary musical features.

### 2.3 Output data

Online music-emotion observations are stored as a data table relating musical features to emotions. The data in the table represents the number of respondents agreeing to the correlation between a general level (high or low) of a musical feature and an emotion. The resulting **feature-emotion map** represents the popular responses to each feature-emotion relationship within the observed data.

Results from expert media-analysis are stored as a database file, accessible to multiple components of the system. Each database represents annotation of a single creative work, normally film media (DVD) and its associated soundtrack album (CD).

Moodtrack's **music-arranger model** output arrives as a set of solutions to the supplied criteria using the specified parameters. Each solution consists of an array of audio segments that are concatenated to build a section of music.

### 2.4 System Components

Three user environments were created in order to discover and organize critical components of the overall system. These components are able to share data by using a common emotion model, musical feature set and data structures.

*Data-collection site: Film Game*

The Film Game contains a set of three web-based activities for players to express thoughts on music and emotions. Data collected during game play is used to determine associations between musical features and induced emotions, creating a **feature-emotion map**.

*Annotation GUI (Graphical User Interface): Media Content Librarian*

The **Media Content Librarian** is an application designed to allow music editors to annotate music and film. Source music is segmented into short musical phrases. Using the interface, a music expert evaluates the soundtrack's audio and film content for use by the system. The application exports a data file containing an annotated library for use by the **music-arranger model**.

*Music-arranger model: Media Content Shell*

The **Media Content Shell** provides access to methods for custom-building music to fit the user's specifications. An interactive shell and scripting language offers methods for the manipulation of emotion, musical feature and cue data. Using the shell, users can load soundtrack libraries, provide emotion, feature and timing requirements, and parameters.

## 3. THEORY AND RATIONALE

*Watch a film run in silence, and then watch it again with eyes and ears. The music sets the mood for what your eye sees; it guides your emotions; it is the emotional framework for visual pictures.* –D. W. Griffith, director (quoted in Karlin, 1994)

### 3.1 Purpose, Re-purpose

The major difference between film music[1] and purely concert music lies in their original purpose. Music for film need not command complete attention of the audience. Needless to say, film music often stands perfectly well on its own, a fact that is revealed by the large amount of soundtracks that have effectively made a transition from screen to concert stage. However, the underlying concept of film music is to accompany or 'play up' on-screen drama. Music may provide setting, portray characters' emotions, remind the audience of past events or alert the audience to future events. While music originally intended for the concert stage may express a similar range of feeling as music for film, the concert composer has autonomous control of emotional contour. While film composers have the freedom to decide the intensity and rate of approach towards desired emotions, the film's timing and the director's vision provide relatively stringent requirements of emotional contour.[2]

Music-emotion researchers commonly record a subject's emotional response to a musical stimulus and correlate the response with known changes in musical features. Musical selections used in such experiments include concert music[3], jazz and popular music styles. Anecdotal evidence supports the idea that children can identify significant moments in a known narrative

[1] This could also include the generic precursors to film music, such as music for opera and ballet, and incidental music for theater.

[2] With the exception of music videos and some cartoons, film is normally not cut to music.

from music alone. A kindergarten teacher played excerpts from Max Steiner's classic 1933 *King Kong* score, and the young students could quickly and easily determine the point in the story (as King Kong falls in love, gets attacked by planes, etc.) with considerable accuracy (Harris, 2006). Music-emotion researchers seeking musical excerpts with explicit emotional content need look no further than the great film scores of the last century.

The focus of this project lies at the intersection of film music analysis and music-emotion research. Towards a greater understanding of emotion in both music and film, musical narratives are explored through analysis and reconstruction (**re-purposing**) of musical underscoring. The research described here may prove useful to Music Information Retrieval, and music perception, cognition and therapy.

Historically, music-emotion research has revealed many distinct connections between musical features and induced emotions (Gabriellsson/Lindstrom, 2001). To design experiments to discover such correlations, researchers must adopt representations of both emotions and musical features. This project is no different in its need for accurate and intuitive representations.

### 3.2 Representation of music affect (emotion)

Emotional representations typically involve either categorical or graphical approaches. Categorical representations include sets of primary moods, such as those proposed by Ekman (1999) and Plutchik (1980). Graphical representations of emotion include two-dimensional *Arousal-Valence* space (Russell, 1980), or three-dimensional *Pleasure-Arousal-Dominance* space (Mehrabian, 1996). Researchers have found that *Arousal-Valence* offers sufficient complexity for describing emotion, while being simple enough for users to evaluate effectively

[3] Commonly known as *classical music*, though not necessarily from the *Classical Period*

(North/Hargreaves, 1997). In addition, research has demonstrated that up to 80% of emotion-response variance is effectively represented using only the two dimensions *arousal* and *valence* (Whissell et al., 1986).

Music-emotion scholars have adopted their own representations and systematic approaches. Kate Hevner used an eight-adjective circle to categorize emotion described by subjects in her experiments (1936), shown in Figure 1 with additional terms added by David Huron (2006). Emery Schubert experimented with continuous self-report of emotional response to musical stimuli based on the *Arousal-Valence* two-dimensional emotion space, or 2DES (1999).

To be useful to the current project, a representation of emotions must allow for intuitive and detailed description by music experts. In addition, the representation must provide a means for effective comparison between emotions. To satisfy these two requirements of the system, a hybrid categorical-graphical representation is used. While Schubert had much success with the 2DES (1999/2001), the training required of the user seemed to be an unnecessary addition to the already complex task of music and film annotation. For the annotation stage, a categorical representation was thought to be most useful. Hevner's emotion-group representation is extended with the addition of a vector representing the reported *intensity* of the emotion, providing another level of detail for annotations and queries. A selected emotion-group and *intensity* is translated to a two-dimensional emotion space to allow for comparison of emotions within the system.

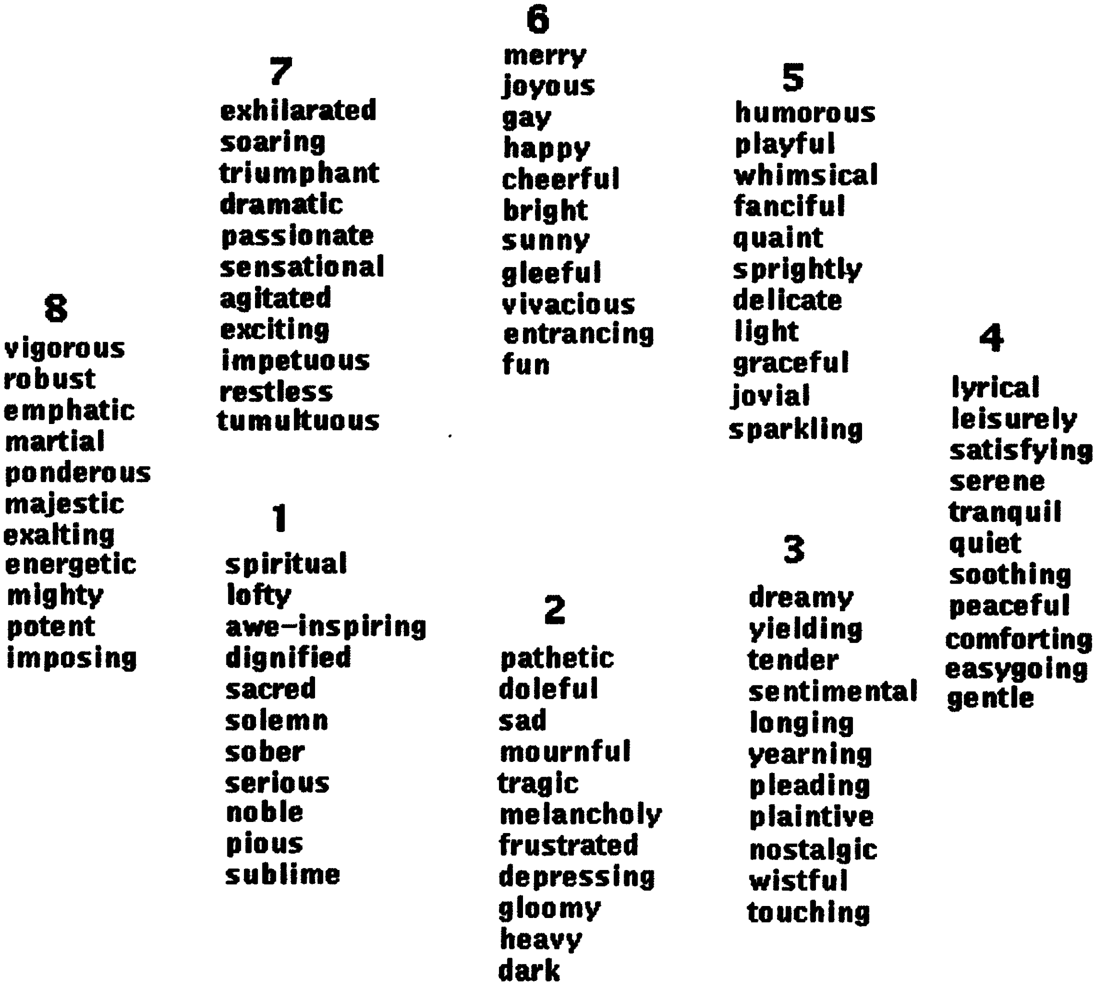


Figure 1. Hevner's music-emotion adjective-circle as revised by Huron (2006)

The use of **Hevner octant-intensity vectors** is introduced as a novel approach to categorical representation of emotions. The addition of intensity levels allows for more effective translation and comparison within *Arousal-Valence* space. The reported *intensity* level of an emotion-group is used to determine the radial distance of an octant vector. The end point of an octant vector is used to translate the reported emotion from octant vector to two-dimensional representation. While Schubert and others have proposed that Hevner octants map directly to *Arousal-Valence* akin to Russell's circumplex model, the current use of octant *intensity* applied as *arousal* contradicts this idea. There are two resolutions of this apparent conflict: either the intensity levels used in the octant representation should not be considered equivalent to arousal levels, or the two-dimensional emotion space being mapped to is not *Arousal-Valence*. One possible label for this space is a flattened *Pleasure-Arousal-Dominance* space, in which the two dimensions are *pleasure* and *dominance* and radial distance represents *arousal*. While the conflict may remain an open issue, octant strengths (vector lengths) are referred to as *intensity* rather than *arousal* to avoid confusion.

Like a large amount of music-emotion research, the analytical approach of this project owes a great deal to the pioneering work of Kate Hevner through its use of correlations between musical features and emotions. In addition, Moodtrack uses the octant model proposed by Hevner, with some additional terms provided by David Huron (2006). Although each emotion-group consists of a set of adjectives, the emotion categories used in this project can be simplified as: *Serious, Sad, Dreamy, Soothing, Playful, Happy, Exciting* and *Vigorous*.

### 3.3 Representation of musical features

Central to this thesis is the adoption of natural language descriptors to effectively describe music regardless of genre or style. The proposed terminology aims to codify the technical language of music into general, but highly descriptive, dimensions. These dimensions of music pique our interest and emotions in a complex and wonderful way that is our musical sensibility. While each of us has our own unique preference, it is argued that our descriptions of music share a common ground.

Western music notation has a long tradition of using symbolic and natural language descriptors. Written music traditionally uses symbols (e.g. '<' as *crescendo*) and descriptive terms (often in Italian, such as *forte, staccato, andante, rubato* or *subito*), to describe parameters within various musical feature dimensions. The language problem alone is complex, while semantic meanings can be vague and often vary due to musical genre, context or preference. Dimensions of music can include a wide range of descriptive terms, with varying levels of subjectivity: *Tempo, Mode, Form, Dynamics, Rhythm, Timbre, Melody, Harmony, Register, Dissonance, Intensity, Texture, Articulation, Density, Melodic direction*, etc.

In his seminal book, *What to Listen For in Music*, the great American composer (of concert and film music) Aaron Copland cites the four major elements of music as *Rhythm, Melody, Harmony and Tone Color* (1939/1957). Each element is given careful critical treatment both historically and musically. Copland regards *Rhythm* as the most essential element, followed by *Melody*. He notes that *Harmony* is a relatively recent invention, as music before the ninth century consisted solely of single melodic lines. *Tone Color* he observes as analogous to color in painting. While these observations are remarkably useful to provide music appreciation techniques for the non-musician, much modern music would not follow the priorities he describes. A composer such as Ligeti[4] makes scant use of *Melody*, preferring to operate within the elements of *Tone Color* and *Harmony*. Although their importance may vary greatly between musical works, Copland astutely chose vital elements of music that can be understood and discussed by musician and non-musician alike.

Using Copland's approach as a model, the set of musical elements selected for the current system must be clear and explicit to the user, regardless of musical background and experience. An element of music does not imply a dimension with opposite poles necessary for this project. For this, a set of musical feature-dimensions is required, each having an adjective-pair of polar

[4] Ligeti's music is widely known in film as the atmospheric music chosen by Stanley Kubrick for use in his film *2001*

opposites (Table 1). The set of musical features used in this thesis is distilled from literature by music-emotion researchers, composers, critics and scholars of film and concert music. To adjust the lists provided by Hevner and Copland, priority was given to features common in discussions of film music. The resulting musical feature-dimensions are listed in Table 2.

| **Feature** | **Negative pole** | **Positive pole** |
|---|---|---|
| Mode | minor key | major key |
| Tempo | slow | fast |
| Rhythm | firm | flowing |
| Melody | descending | ascending |
| Harmony | simple | complex |
| Register | low pitch level | high pitch level |

Table 1. Feature-dimension polar opposites used by Hevner

This research offers these musical features and polar opposites as a concise list of musical parameters suitable for subjective evaluation by experts. The intention was to create a set of general features that are applicable across cultures and musical genres. Musical experts clearly adjust their analysis of music into these dimensions depending on the musical genre, but the process of data-entry can remain constant. Collecting subjective musical analyses can provide a foundation for researchers to learn how musical dimensions vary between genre and culture.

| Feature | Negative pole | Positive pole |
|---|---|---|
| Tempo | slow | fast |
| Rhythm | simple | complex |
| Melody | simple | complex |
| Harmony | consonant | dissonant |
| Register | low pitch level | high pitch level |
| Dynamics | soft | loud |
| Timbre | pleasing | harsh |
| Density | thin | thick |
| Texture | distinct | blended |

Table 2. Feature-dimension polar opposites used by Moodtrack

### 3.4 Self-report data

Musical features and emotions are evaluated by music experts and self-reported to provide annotation. By using subjective evaluation as a means to annotate emotion and musical features, retrieval is based more on the annotator's preferences than on algorithmic means. Annotation by **Hevner octant-intensity vectors** requires a pair of numbers as input–one representing the selected octant, and another representing the self-reported *intensity* of the selected emotion-group. Because of their ubiquitous nature, percentiles are used as scales for both emotion-group *intensity* and musical feature level. The self-reported intensity of emotion thus ranges on a scale from 0 (minimal *intensity*) to 100 (peak *intensity*). Musical features follow a similar scheme, although polar opposites are located at the ends of each scale. For example, the musical feature *Tempo* ranges from 0 (*slow*) to 100 (*fast*), with a midpoint at 50 (average *Tempo*). Editors using the system were instructed to calibrate their responses within the context of each work. This contextual approach allows the annotation of a film with predominantly slow music to have an equal range of self-reported *Tempo* values as a film whose soundtrack has predominantly fast music. By providing context-sensitive self-reporting, film music can be compared and **re-purposed** more effectively.

### 3.5 Narrative elements

The film director provides a visual and dialogue-driven narrative, but it is up to the composer or **music supervisor** to deliver a relevant musical narrative. Film composers often attach themes directly to specific characters, settings, objects, ideas or emotions. The thematic *leitmotif* is inherited from Wagnerian opera, theater and ballet–all deep troves of emotional content. *Leitmotivic* film techniques were perfected by such composers as Steiner and Korngold, whose films sometimes contain eight or more main themes (Karlin, 1994). The composer's placement of *leitmotifs* has been shown to be useful for finding structure in film (Hawley, 1993). Though highly effective in film music, the referential nature of motifs itself does not provide immediate direction of emotions. It is the musical features that act as vehicles to transport emotions from the musical underscoring to the percepts of the audience members. Film composers and supervisors must effectively manipulate the musical features to produce an emotion for the narrative, with or without the aid of thematic material.

In addition to a mastery of musical features, creators of film music must have a strong emotional intuition. While our everyday emotions may not change at highly dramatic rates, building and transforming emotions in musical narrative must carry a sense of emotional realism in their presentation to the audience. In this respect, film music can be seen as a potential tool to help understand the ways emotions change over time. Through the systematic segmentation and re-assembly of film music, emotions are annotated and restructured, and in the process, better understood.

The approach used here is to treat homogenous musical phrases as a frame, ordered as a sequence of frame-arrays (Minsky, 1986). Partitioned music libraries may prove useful for music-emotion researchers. Provided with pre-partitioned segments, researchers can examine and experiment with living bodies of music, in a kind of sonic vivisection. By operating on a shared set of musical frame-arrays, music-emotion researchers may be able to compare results more effectively.

### 3.6 Media timing

Music can have varying degrees of synchronization with the emotion and visual cues it underscores. Music is often synchronized to the film it accompanies, having no temporal displacement. In the 1958 Hitchcock classic *Vertigo*, Scottie's dramatic bouts of acrophobia are represented on-screen as the cameraman zooms in while moving the camera back, creating a dizzying visual effect. Each of these episodes is consistently accompanied by Bernard Herrmann's dizzying music, with its swirling harp glissandi and dissonant harmonies. Music can punctuate the emotion of an event from the recent past. An example of this positive temporal displacement, in another Hitchcock classic, is Bernard Herrmann's musical finale to the unscored sequence in *North By Northwest* where Thornhill narrowly escapes an attack by a crop-duster (1959). Conversely, music can provide advance notice of upcoming drama, as in John Williams' well-known two-note motif representing the approaching shark in *Jaws* (1975). In this case, the music foreshadows an impending event through negative temporal displacement. Horror movies typically make extensive use of musical foreshadowing to build suspense and provide a sense of danger. Through the use of positive and negative temporal displacements, and tightly-synchronized accompaniment (no displacement), film composers can effectively provide contrast between **diegetic** and **non-diegetic realms** to provide effective emotional underscoring of the film.

Because temporal displacement is such a vital aspect of film scoring, to effectively analyze film music the temporal representation scheme must allow separate storage of diegetic and non-diegetic emotion timings. The annotation tools developed for this project provide two identical accounts for analysis representing diegetic (on-screen, off-screen dialogue or implied) emotions and non-diegetic emotions of the musical underscore.

### 3.7 Emotional congruence

Displacement can be applied to the emotional congruence between film and music, that is, the degree of alignment between emotions seen or implied on screen and those implied by the musical score. An emotionally-incongruent soundtrack is said to "run counter to the action." In this case, music can be used for ironic effect, such as the selection of Louis Armstrong's "What a Wonderful World" set to horrific scenes of violence and destruction in the film *Good Morning Vietnam* (1988). Most composers, editors and supervisors use the effect sparingly, preferring music with higher levels of emotional congruence with the film. While emotional-incongruence is an interesting film music effect, this project aims to provide music that is emotionally-congruent to the film. Users can simply provide incongruent emotions to the system to create music that "runs counter to the action."

### 3.8 Music retrieval

Music can be retrieved from the library according to its musical feature or emotion. Feature-based retrievals allow audio segments annotated with an emotion different than that of the search query to be considered during the search process. Attaching weights to each member in the set of musical features provides the user parameters to adjust the retrieval process. Four types of music retrieval methods are offered to the user: two feature-based and two emotion-based techniques. An **annotated feature-based retrieval** performs a cross-correlation between human-annotated features in the reference music (typically the original musical score for the scene) and each possible candidate in the library. **Extracted-feature-based retrieval** involves a similar process, except machine-extracted features are examined. For these two processes, providing a reference musical section is required prior to initiating a query. An **emotion-based retrieval** performs a cross-correlation between the provided **temporal emotion cue** and the musical section candidates. A **hybrid feature-emotion retrieval** method is possible wherein the emotion cue is first translated to a set of musical feature cues through a **feature-emotion map**. The hybrid method involves all three components of the system to build a section, while the other methods omit use of a **feature-emotion map**.

### 3.9 Mapping features to emotions

While Hevner was the first to systematically examine musical feature-dimensions and emotional response, this approach as a research direction has been fruitful, particularly in recent years. Empirical data relating emotions and features can be reduced to reflect only popular responses within each musical feature/emotion-group relationship. In Hevner's experiments for example, given the musical feature *Tempo,* more subjects associated the dimension-pole *fast* rather than *slow* with the emotion-group *Exciting* (Table 3). Because the goal of this project is to arrange music using commonsense, unpopular responses are safely dropped.

| **Feature** | **Serious** | **Sad** | **Dreamy** | **Playful** | **Happy** | **Exciting** | **Dramatic** | **Vigorous** |
|---|---|---|---|---|---|---|---|---|
| Mode | -4 | -20 | -12 | 3 | 21 | 24 | 0 | 0 |
| Tempo | -14 | -12 | -16 | -20 | 6 | 20 | 21 | 6 |
| Rhythm | -18 | -3 | 9 | 2 | 8 | 10 | -2 | 10 |
| Melody | 4 | 0 | 0 | 3 | -3 | 0 | -7 | -8 |
| Harmony | -3 | 7 | -4 | -10 | -12 | -16 | 14 | 8 |
| Register | -10 | -19 | 6 | 8 | 16 | 6 | -9 | -13 |

Table 3. Hevner data reflecting popular respondents in musical feature dimension-poles

### 3.10 Context-driven music

Systems designed to generate music based on emotion can employ a variety of techniques. Systems can be categorized based on format of the input emotions and musical output. Specific emotions can be inferred from physiological signals, and, using the inferred emotion, produce a track-playlist (Dabek, et al., 1986), or drive real-time arrangements (Chung/Vercoe, 2006). Emotions can be inferred based on events in a game environment and used to adjust MIDI performance parameters (Livingston, 2005). Visual, emotional and contextual clues can be used to set parameters of an algorithmic composition (Jewell/Prugel-Bennett, 2003).

The *arrangement* of music involves the adaptation of a piece of existing music into a new instrumental or stylistic domain. A jazz arranger, for instance, may arrange a piece originally for solo piano to be played by a big-band. A *re-mix* is a term used by electronic musicians essentially referring to the same process of music arrangement, although the addition of beats and electronic sounds is implied. The methods used here involve aspects of arranging and re-mixing, although the term *re-assembly* might be more appropriate given that audio segments are not modified in any way other than their temporal location. This term means nothing to musicians, so placement methods are referred to as *arrangement* methods. The reader is reminded that *arranging* techniques described in this thesis do not modify instrumentation or stylistic content of the source music as arrangements commonly do. The project takes a segmentation-reconstruction approach, in which short audio segments are re-ordered in time to create sections of music.

Music libraries are created through audio segmentation, a time-consuming and somewhat tedious task. Musical selections are divided into short musical phrases, creating audio segments lasting between 2 and 15 seconds in duration. Segments were extracted from the piece with the condition that they maintain relatively even musical features throughout the segment. If possible, segments were chosen so that they could be looped musically. The first half of a soundtrack CD used in the study yielded approximately 300 individual audio segments. The majority of the audio segments can be looped musically, allowing greater flexibility for their use by the **music-arranger model** (Section 4.3).

### 3.11 Difference-engines

In his book *The Emotion Machine*, Marvin Minsky succinctly outlines four major components (Difference-engines) of musical perception as **Feature-Detectors**, **Measure-Takers**, **Theme-Detectors** and **Section-Builders**. Inspired by these conceptual divisions, several components of the **music-arranger model** share the same name and provide analogous functionality, albeit inverted to construct rather than break down sections of music. Re-purposing music involves each difference-engine at various stages in the process (Figure 2).

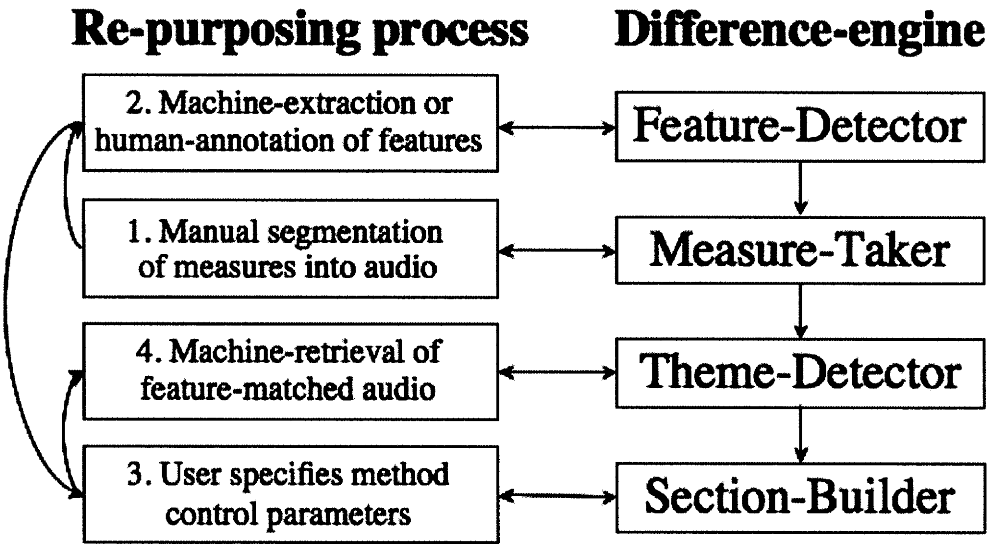

Re-purposing process
Difference-engine
2. Machine-extraction or human-annotation of features
1. Manual segmentation of measures into audio
4. Machine-retrieval of feature-matched audio
3. User specifies method control parameters
Feature-Detector
Measure-Taker
Theme-Detector
Section-Builder

Figure 2. Difference-engines used in stages of the re-purposing process

# 4. DESIGN AND IMPLEMENTATION

Each of the three components of the system is now presented in greater detail.

## 4.1 Data collection site: The Film Game

Collecting data through an online game interface has been shown to be both useful for researchers and fun for players (von Ahn, 2006). Players of The Film Game are provided three activities involving musical features and emotion. In each activity, the player provides an assertion relating musical features to emotions using either pull-down menus or their own words. Players are scored according to the popularity of their answers.

A number of classic and popular films were selected to represent a variety of genres and historical eras. From each film, a variety of scenes were selected according to a specific set of criteria. Most importantly, the music during the scene must be found on the film's soundtrack album, allowing the music to be presented alone with no dialogue or other diegetic sound. Scenes were selected to represent the emotional and musical range of each film. Any scenes that were judged to have "music running counter to the action" were expressly eliminated from the set of possible choices. Each film resulted in five to nine scenes used as rounds in the game. For each scene, video clips were extracted from DVD without the accompanying audio track. To allow for faster download over the web, clips were reduced to half their original size, maintaining the original aspect ratio. Four adjectives were selected from each of Hevner's eight emotion-groups to standardize data collection. Each time they are presented to the player, the list is randomized to eliminate order or group preference (as suggested by Huron). The order in which the musical features appear in the pull-down list is also randomized. Three versions of the game were created–two music narrative-based games, and a remix-style game.

*Music narrative games:* Two varieties (*basic* and *advanced*) of music narrative games progress through a single film to tell the story through music. The player selects one of several music clips that best matches a film clip. The composer's choice is assumed to be a wise one, although one or both of the two decoy clips may also work. The rounds of the game were constructed to be challenging, but not obscure. The player is asked to justify his/her choice in natural language. A player choosing music not actually used during the scene can still score points for accurately describing the music's emotion and features. Accurate descriptions are determined by averaging responses from other players.

*Music and film remix game:* The third game presents random selections of music and film. The player is asked to judge how well the clips work together, and justify their choice.

### 4.1.1 Basic music narrative game

*Film selection.* Players are first asked to select a film to play. To make the game more enticing, original artwork from all films is presented (Figure 3). The promotional artwork relates the general mood of the film and may remind players if they have seen the film before. Players begin their game by clicking on the artwork of the film they would like to play.

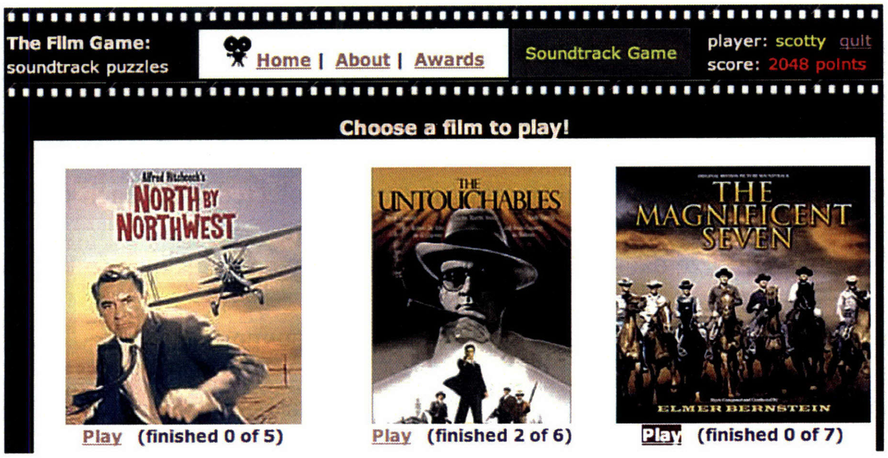
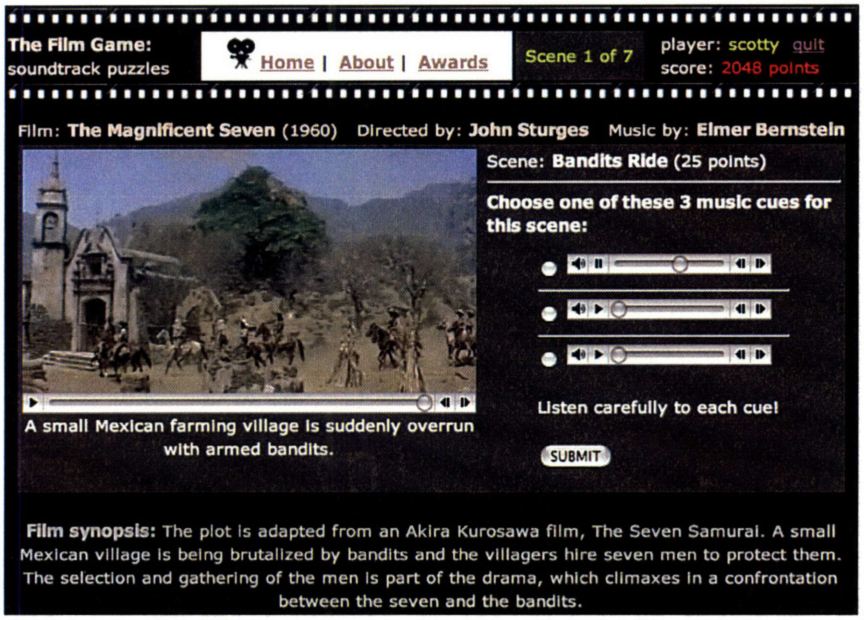


Figure 3. Film Game media selection

*Music selection.* Players are presented a film clip and three music clips (Figure 4). Players are asked to select the music most appropriate for the scene. In many cases, the emotion is dependent not just on the visuals, but the story and context. To assist players, a description of the characters and the context of the scene's action are provided. At the bottom of the page, a synopsis of the entire film is given for players who have not seen the film.



The Film Game: soundtrack puzzles

Scene 1 of 7

player: scotty

score: 2048 points

Film: **The Magnificent Seven** (1960) Directed by: **John Sturges** Music by: **Elmer Bernstein**

Scene: **Bandits Ride** (25 points)

**Choose one of these 3 music cues for this scene:**

A small Mexican farming village is suddenly overrun with armed bandits.

Listen carefully to each cue!

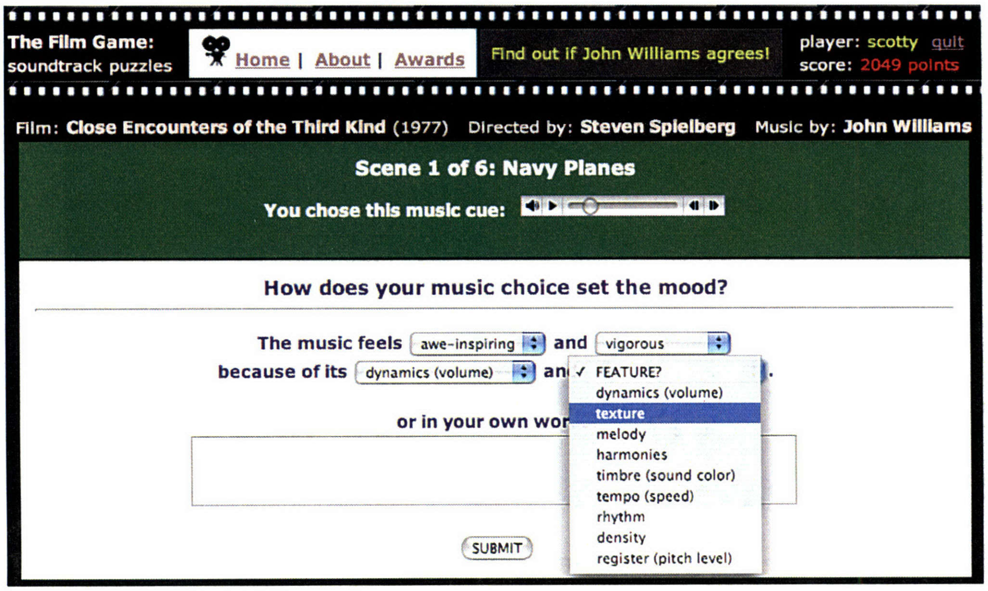
**Film synopsis:** The plot is adapted from an Akira Kurosawa film, The Seven Samurai. A small Mexican village is being brutalized by bandits and the villagers hire seven men to protect them. The selection and gathering of the men is part of the drama, which climaxes in a confrontation between the seven and the bandits.

Figure 4. Film Game music selection

*Emotion/feature selection.* The following page prompts the player to select the emotion and musical features of the music. The music clip previously selected by the player in the previous page is presented once again. To allow the player to focus on the music itself, the music is presented without the film clip. A basic assertion of musical aesthetic is provided, containing missing information for the player to complete, using the following format:

The music feels <emotion1> and <emotion2> because of its <feature1> and <feature2>.

Pull-down menus are provided to collect the player's interpretation of the music's emotions and features. The first two pull-downs contain a list of emotions, selected according to the process described above. The next two pull-downs represent the choice of available musical features, also using the above process. In addition, the player is provided a text field to describe their reaction in their own words. At the top of the page, players are reminded that they will soon discover if they have chosen the music cue that was actually used during the scene, with statements such as, "Find out if <composer> agrees!" The reminders encourage players to continue the game by sustaining their curiosity. Players may also be reminded about the director, composer and release year of each film in the reminders (Figure 5).



The Film Game: soundtrack puzzles

Find out if John Williams agrees!

player: scotty

score: 2049 points

Film: Close Encounters of the Third Kind (1977) Directed by: Steven Spielberg Music by: John Williams

Scene 1 of 6: Navy Planes

You chose this music cue:

How does your music choice set the mood?

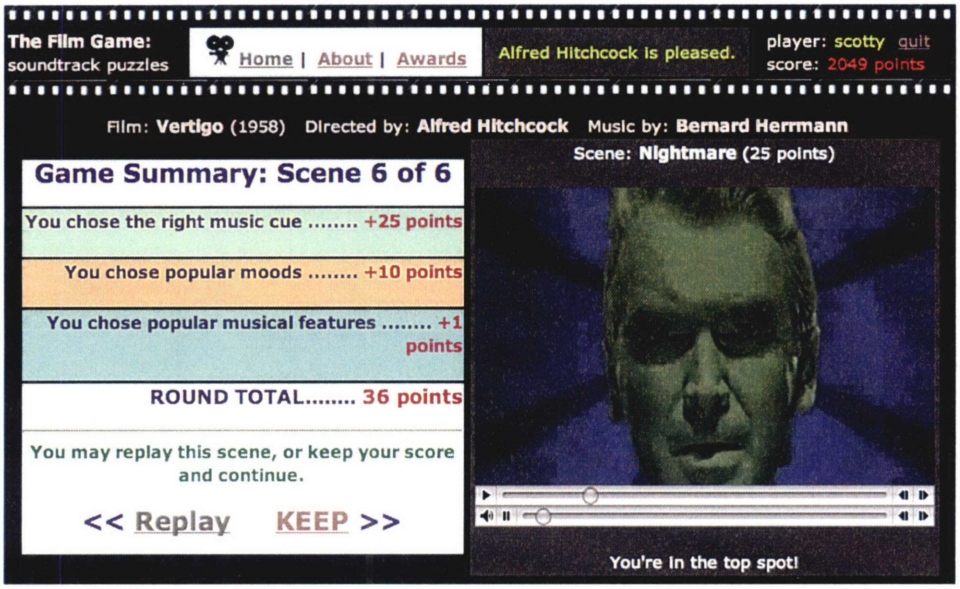
The music feels awe-inspiring and vigorous

because of its dynamics (volume) an

or in your own wor

Figure 5. Film Game emotion and musical feature selection (basic game)

*Round summary and scoring.* Players are shown their score for the current round, as well as the film clip and music choice presented together. Players are given the most points (20-30) for selecting the original music from the scene shown. Depending on their fellow player's reaction, between 5 and 10 points are awarded for selecting popular emotions. Also dependent on public opinion, 1 to 10 points are awarded for the player's choice of musical features (Figure 6).



The Film Game: soundtrack puzzles

Alfred Hitchcock is pleased.

player: scotty

score: 2049 points

Film: Vertigo (1958) Directed by: Alfred Hitchcock Music by: Bernard Herrmann

Scene: Nightmare (25 points)

Game Summary: Scene 6 of 6

You chose the right music cue ........ +25 points

You chose popular moods ........ +10 points

You chose popular musical features ........ +1 points

ROUND TOTAL........ 36 points

You may replay this scene, or keep your score and continue.

You're in the top spot!

Figure 6. Film Game round summary

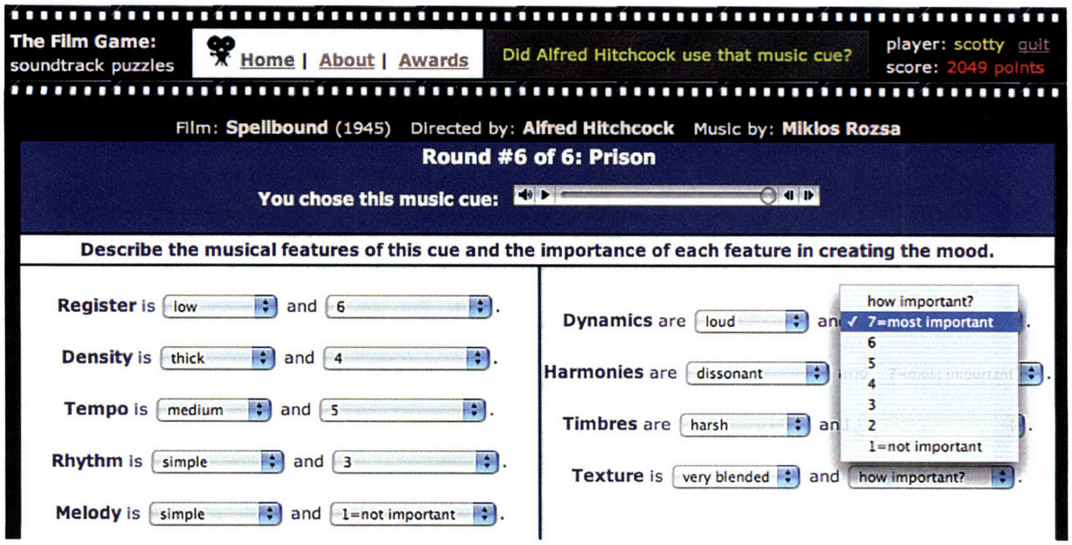
### 4.1.2 Advanced music narrative game

*Film selection/Music selection.* These are presented identically to the basic game.

*Emotion selection.* Players are presented their music selection, alone without the film clip. The emotion page allows players to describe the emotion of the music by completing an assertion using the following format:

The music feels <emotion1> and <emotion2>.

Emotion choices are presented using the same presentation and process as the basic music narrative game.



player: scotty quit
score: 2049 points

Film: Spellbound (1945) Directed by: Alfred Hitchcock Music by: Miklos Rozsa

Round #6 of 6: Prison

You chose this music cue:

Describe the musical features of this cue and the importance of each feature in creating the mood.

Register is low and 6.

Density is thick and 4.

Tempo is medium and 5.

Rhythm is simple and 3.

Melody is simple and 1=not important.

Dynamics are loud and 7=most important.

Harmonies are dissonant and 7=most important.

Timbres are harsh and

Texture is very blended and how important?.

Figure 7. Film Game musical feature selection (advanced game)

*Musical feature selection.* Players are again presented their choice of music without the film clip. The musical feature selection offers players many more options to describe the musical features of their choice (Figure 7). Players are asked to describe the musical features and the importance of each feature to generate the emotion they chose. Incomplete assertions of each of the nine musical features are presented in a random order, in the following format:

<Musical feature> is <level> and <importance>.

The level pull-down menu list includes five choices, representing a neutral choice ("medium"), two opposite poles, unmodified, and two very opposite poles. Using the musical features and positive/negative poles listed in Table 2, choices are presented in a pull-down menu list. For example, the musical feature *Tempo* provides the options *very fast, fast, medium, slow* and *very slow*. Levels of importance are presented as seven-point scales, ranging from "not important" (1) to "most important" (7).

*Round summary and scoring.* Players are shown their score for the current round, as well as the film clip and music choice presented together. The method of scoring is similar to that of the basic game.

### 4.1.3 Remix game

Players are presented a film clip and a music cue, chosen from the database at random. Because film and music clips are chosen by the system at random, starting the remix game does not permit the player to choose film or music. Most often, the film clips and the music cue originate from different films. No checking is performed in the random selection of the clips, so the film and music clips may originate from the same film. Occasionally, the film clip may be presented with the exact music used.

### 4.1.4 Exporting data

Data collected from players of the **Film Game** can be exported and viewed on-line by the researcher, reflecting data as it is collected. Data from each variation of the game can be combined or viewed separately.

## 4.2 Annotation GUI: Media Content Librarian

The **Media Content Librarian** is a content-management tool for use by expert users, such as film music editors, supervisors or directors. The application was designed to provide extensive means to annotate and represent information useful in building a feature- and emotion-based music library. A variety of representations and functions are provided to assist editors in the difficult process of building and annotating a library.

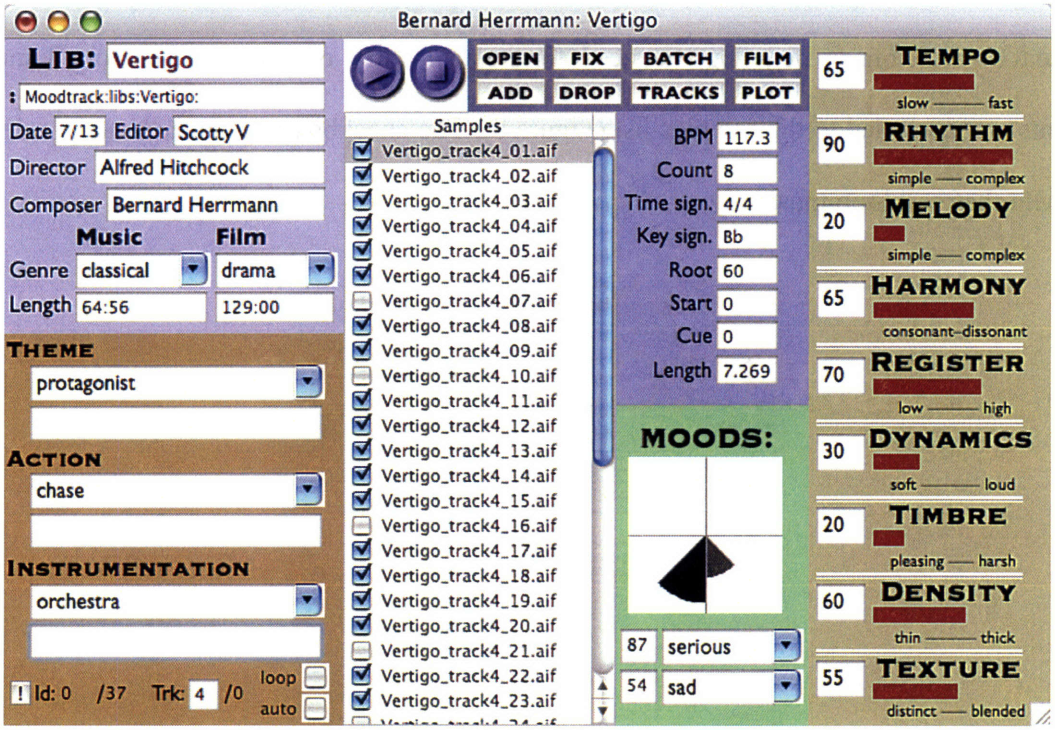
### 4.2.1 United source media

Prior to annotation, the librarian software requires media delivered in a specific format. Typical annotations involve both the soundtrack (audio CD) and film (DVD) from a single source work. Uniting two media artifacts from a single source work is an important aspect of the annotation. Methods for extraction and annotation are selected to provide portability among users via standard text files.

Soundtrack media can be extracted ("ripped") from audio CD using a number of commercial and open source sound applications. Segmentation points are determined using professional audio-editing software, and timing of each point is entered into a text file, referred to in this document as a **contiguous regions list**. An editing tool such as Audacity (2006) can take a text file containing a list of timings as input, and output a set of audio files ("samples") representing sections of audio between each timing point. Film media is extracted from DVD using a program such as MacTheRipper (2006), which is converted to Quicktime-readable format with a program such as OpenShiiva (2006).

### 4.2.2 Media Content Librarian: Virtual tour

Features of the annotation tool are presented by systematically guiding the reader through the typical steps involved in the creation of a soundtrack library.



Bernard Herrmann: Vertigo

LIB: Vertigo
: Moodtrack:libs:Vertigo:
Date 7/13 Editor ScottyV
Director Alfred Hitchcock
Composer Bernard Herrmann

| | Music | Film |
|---|---|---|
| Genre | classical | drama |
| Length | 64:56 | 129:00 |

THEME: protagonist
ACTION: chase
INSTRUMENTATION: orchestra
! Id: 0 /37 Trk: 4 /0 loop □ auto □

Samples
☑ Vertigo_track4_01.aif
☑ Vertigo_track4_02.aif
☑ Vertigo_track4_03.aif
☑ Vertigo_track4_04.aif
☑ Vertigo_track4_05.aif
☑ Vertigo_track4_06.aif
□ Vertigo_track4_07.aif
☑ Vertigo_track4_08.aif
☑ Vertigo_track4_09.aif
□ Vertigo_track4_10.aif
☑ Vertigo_track4_11.aif
☑ Vertigo_track4_12.aif
☑ Vertigo_track4_13.aif
☑ Vertigo_track4_14.aif
☑ Vertigo_track4_15.aif
□ Vertigo_track4_16.aif
☑ Vertigo_track4_17.aif
☑ Vertigo_track4_18.aif
☑ Vertigo_track4_19.aif
☑ Vertigo_track4_20.aif
□ Vertigo_track4_21.aif
☑ Vertigo_track4_22.aif
☑ Vertigo_track4_23.aif

BPM 117.3
Count 8
Time sign. 4/4
Key sign. Bb
Root 60
Start 0
Cue 0
Length 7.269

87 serious
54 sad

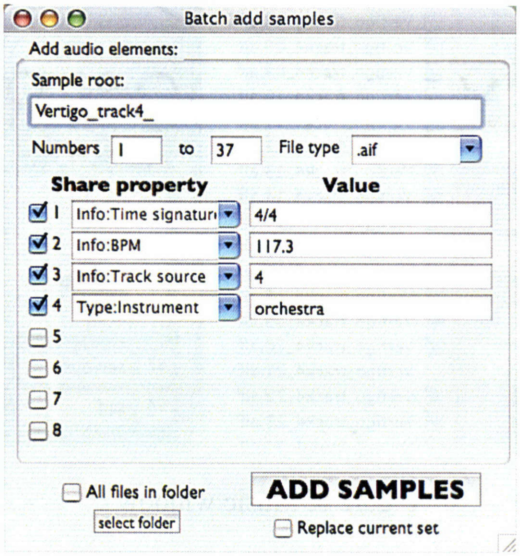
| | | |
|---|---|---|
| 65 | TEMPO | slow — fast |
| 90 | RHYTHM | simple — complex |
| 20 | MELODY | simple — complex |
| 65 | HARMONY | consonant–dissonant |
| 70 | REGISTER | low — high |
| 30 | DYNAMICS | soft — loud |
| 20 | TIMBRE | pleasing — harsh |
| 60 | DENSITY | thin — thick |
| 55 | TEXTURE | distinct — blended |

Figure 8. Music window

*Music window:* Soundtrack-related information is displayed and entered from within the music window (Figure 8). All timing information entered and displayed from the music window are in seconds, with the exception of the film and soundtrack durations, which are entered and displayed as **pseudo-timecode**.

*Library information:* In the upper-left portion of the music window (Figure 8), information about the library is entered, including the name, location, duration and genre (film and soundtrack) of the work, its composer, director and annotator. In the music window, all information displayed corresponds to audio segments with the exception of data in the library information section.

*Importing audio segments:* Because a **contiguously-segmented** soundtrack can yield hundreds of segments (audio samples), an audio file batch-import function is provided (Figure 9). The user provides the root name of the audio files, the range of numbers to import (appended to the root filename) and any properties that are shared between all samples being imported. Files are batch-imported to the library, negating the need for manual entry of each file. Imported musical segments are appended to the samples list (Figure 8). The checkbox next to each sample denotes a segment that will loop musically.



Add audio elements:

Sample root: Vertigo_track4_

Numbers 1 to 37 File type .aif

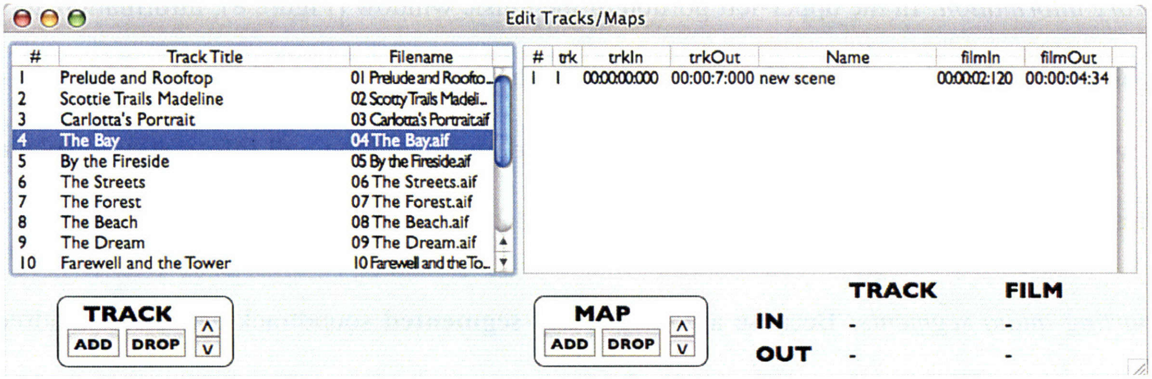
| Share property | | Value |
|---|---|---|
| ☑ 1 | Info:Time signatur | 4/4 |
| ☑ 2 | Info:BPM | 117.3 |
| ☑ 3 | Info:Track source | 4 |
| ☑ 4 | Type:Instrument | orchestra |
| □ 5 | | |
| □ 6 | | |
| □ 7 | | |
| □ 8 | | |

□ All files in folder

□ Replace current set

Figure 9. Batch-import audio window

*Adding tracks/maps:* Corresponding to the soundtrack album, the track list contains track names and file names. In addition, users can create a map between album tracks and the corresponding times the music is heard in the film (Figure 10).



| # | Track Title | Filename |
|---|---|---|
| 1 | Prelude and Rooftop | 01 Prelude and Roofto... |
| 2 | Scottie Trails Madeline | 02 Scotty Trails Madeli... |
| 3 | Carlotta's Portrait | 03 Carlotta's Portrait.aif |
| 4 | The Bay | 04 The Bay.aif |
| 5 | By the Fireside | 05 By the Fireside.aif |
| 6 | The Streets | 06 The Streets.aif |
| 7 | The Forest | 07 The Forest.aif |
| 8 | The Beach | 08 The Beach.aif |
| 9 | The Dream | 09 The Dream.aif |
| 10 | Farewell and the Tower | 10 Farewell and the To... |

| # | trk | trkIn | trkOut | Name | filmIn | filmOut |
|---|---|---|---|---|---|---|
| 1 | 1 | 00:00:00:000 | 00:00:7:000 | new scene | 00:00:02:120 | 00:00:04:34 |

| | TRACK | FILM |
|---|---|---|
| IN | - | - |
| OUT | - | - |

Figure 10. Window showing track list and music-film timing map

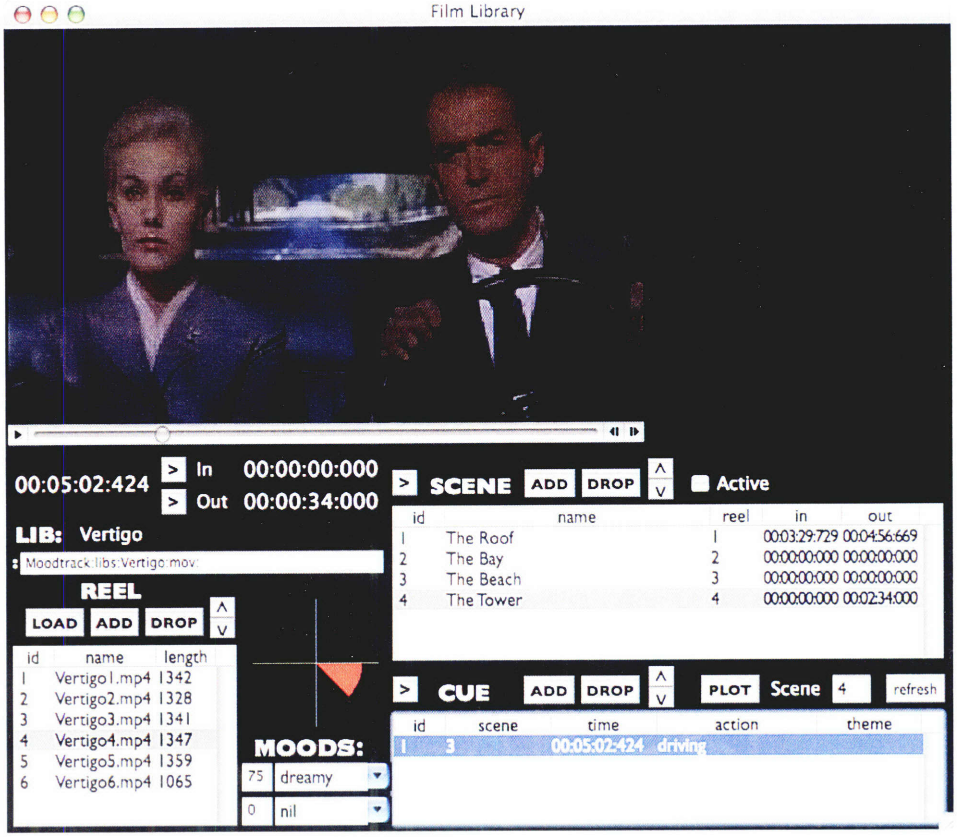
*Auditioning audio samples:* To listen to segments in the library, users click the play button or double-click the name of the segment in the samples list (Figure 8).

*Annotating theme, action and instrumentation:* The lower-left portion of the music window (Figure 8) contains edit fields for theme, action and instrumentation descriptions of each audio sample.

*Annotating audio data:* Information about each audio sample is entered into the music window, including BPM (tempo as beats-per-minute), beat count, time signature, key signature, root note, source track number, source track start time, film start time and duration (Figure 8).

*Annotating emotion:* Users enter self-reported emotional reaction to each audio sample in the "mood" section of the music window (Figure 8). As described in Section 3.4, an emotion is reported by selecting an octant and entering the intensity in the corresponding edit field. The user's adjective and intensity annotations are displayed graphically as a pie-chart, with the orientation of the slice representing the octant, and its length from the center indicating the intensity level. Two emotions can be entered, a primary and secondary emotion, respectively shown as darker and lighter shades. The octant-intensity emotion representation used in the music window (Figure 8) is also used in the film window (Figure 11).

*Annotating musical features:* As described in Section 3.4, self-reported levels of each audio sample's musical feature-dimensions are entered into the edit fields in the far right-hand portion of the music window (Figure 8). Annotated musical feature levels are graphically displayed as horizontal bars.



00:05:02:424

In 00:00:00:000

Out 00:00:34:000

LIB: Vertigo

Moodtrack:libs:Vertigo:mov:

REEL

| id | name | length |
|---|---|---|
| 1 | Vertigo1.mp4 | 1342 |
| 2 | Vertigo2.mp4 | 1328 |
| 3 | Vertigo3.mp4 | 1341 |
| 4 | Vertigo4.mp4 | 1347 |
| 5 | Vertigo5.mp4 | 1359 |
| 6 | Vertigo6.mp4 | 1065 |

MOODS:

75 dreamy

0 nil

SCENE □ Active

| id | name | reel | in | out |
|---|---|---|---|---|
| 1 | The Roof | 1 | 00:03:29:729 | 00:04:56:669 |
| 2 | The Bay | 2 | 00:00:00:000 | 00:00:00:000 |
| 3 | The Beach | 3 | 00:00:00:000 | 00:00:00:000 |
| 4 | The Tower | 4 | 00:00:00:000 | 00:02:34:000 |

CUE Scene 4

| id | scene | time | action | theme |
|---|---|---|---|---|
| 1 | 3 | 00:05:02:424 | driving | |

Figure 11. Film window

*Film window:* The main window for annotation of all film-related information is displayed and entered from within the film window (Figure 11). All times in the film window are entered and displayed in **pseudo-timecode**.

*Adding reels:* Film media is added to the library through the film window. The term reel is used to represent a portion of the film using its unique TOC-ID from the source DVD. (see *United source media* above)

*Adding scenes:* Scenes are defined by selecting the name of a reel and setting in-point and out-point in the film window. Any number of scenes can be assigned for each reel, and need not be contiguous.

*Adding cue points:* Cue points for emotion-annotation are added and assigned to the scene currently being annotated. Any number of cue points can be added for each scene.

*Annotating emotion:* Emotions of the film are annotated as described above for music-annotation. Film emotion-annotation requires the additional step of adding a cue point describing the onset timing of the emotion.

*Graphical plotting of emotion and feature contour:* Emotion and musical feature contour of the soundtrack annotation can be graphically plotted and displayed (Figure 12) Emotional contour of the film annotation can be plotted similarly. Viewing the graphical plots can assist editors to evaluate the quality of their annotation. Visually analyzing the plot can reveal aspects of compositional style by examining relationships between musical features and emotions. More extensive plotting functions are provided in the **Media Content Shell** (Section 4.3).

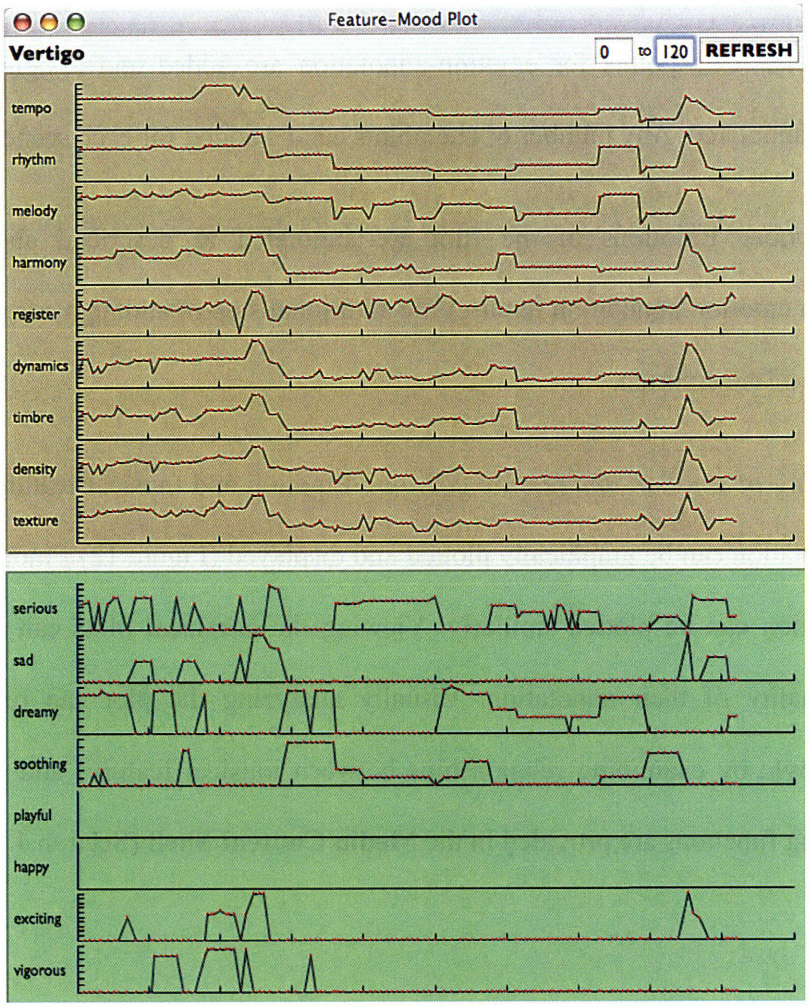


Figure 12. Musical feature and emotion plot

*Data storage:* Annotated libraries are stored as XML files using the ".sndml" (**SoundtrackML**) file extension. For a complete list of **SoundtrackML** tags and an example of an annotated library, see Appendix 2.

### 4.3 Music-arranger model: Media Content Shell

The adaptive **music-arranger model** is an exploratory work in music sensibility. The **Media Content Shell** interface provides user control of all available methods.

### 4.3.1 The "Cue"

In film, the *music cue* refers to a section of music, often corresponding to a separate track on a soundtrack album. The term **cue** is used in this document to refer to a data structure describing music, musical features, action or emotion. While some Moodtrack cue-types represent sections of music, the reader is reminded of the distinction between a *music cue* as used in the context of film, and the meaning used here. The term *cue* was chosen in particular reference to the *cue sheets* used by silent film musicians to list timing, action, time signature and appropriate music titles used throughout a film. The use of *cue sheets* is described by silent film music scholar Martin Marks:

> The cue sheets that composers use are much less widely known than scripts. In principle they are little different, being a kind of setting-down of sequences from a film in shorthand. Their function, however, is quite special: to link the music to the rest of the film. In the silent period cue sheets provided a series of suggestions for music to be used in accompaniment, "cued" to the titles and action on the screen. [...] Sound film cue sheets, normally prepared by a film's "music editor," describe the action, dialogue, and (some) sound effects of scenes for which the composer is to write music. Since the composer often works from these cue sheets after viewing the film, they become important clues to the compositional process, telling us what details were thought by the composer to deserve musical emphasis. (Marks, 1997)

The organization of film into a set of cues representing changes in action, emotion and music over time presents a set of descriptive surfaces useful for reference, display or analysis.

### 4.3.2 Media Content Shell: Virtual tour

Features of the **music-arranger model** are presented by systematically guiding the reader through the steps involved in custom-building sections of music using the shell interface. From within the shell, users can load **SoundtrackML** libraries, extract musical features from audio, manipulate/analyze/visualize data, provide control settings and build sections of music. A sequence of commands can be saved as a **Moodtrack script (.mtk) file** and launched from the command-line interface. **Moodtrack script** command syntax is identical to that of shell commands.

N.B.: **Media Content Shell** commands and parameters are denoted by single quotes, such as 'command-name parameter-value' or, indented without quotes, as typed at the shell prompt:

```
>> command-name parameter-value
```

*Importing a library:* An annotated **SoundtrackML** library created using the **Media Content Librarian** is loaded by typing 'load filename' at the command prompt. While importing a library, users may optionally execute a set of automatic audio feature-extraction routines by entering 'xload filename' at the prompt. Extracted features for each audio sample include: RMS level, peak time, peak level, and the number of zero-crossings.

*Storing a library:* Because the automatic feature-extraction routines involve intensive computation, a **SoundtrackML** library can store the library with extracted values to shorten future load times. Entering 'save' at the prompt will store the file, allowing the feature-extraction routines to be run only once per library.

*Listing elements:* At the prompt, typing 'ls' will display all elements currently loaded in the system. The use of flags allows targeted or detailed listing of certain elements. Element-listing flags are shown in Appendix 3.

*Manually adding a cue:* The Moodtrack **cue** data-structure is used to represent a variety of time-series data, including emotion-annotations, musical feature-annotations, a section of music and data summaries. **Cues** are entered using the syntax,

```
    >> cue-type: x1, y1[, z1]; x2, y2[, z2]; duration
```

where *type* represents the type of cue, duration represents the cue end point, x-values represent time, while y-values and optional z-values represent other data (Table 4). Any number of points may be entered.

*Cue-types:* Moodtrack *cues* are a general data structure used in the model. Cue-types, their represented data and formats are shown in Table 4. Individual cue-types are defined as follows:

- *scene-cue:* description of the film's action over time, such as "chase" or "Scottie and Madeleine kiss passionately and realize they are in love"
- *mood-cue:* emotional contour over time using adjectives contained in the Hevner octant-groups
- *Hevner-cue:* an emotional contour over time (octant-intensity)
- *feature-cue:* a musical feature over time
- *trax-cue:* section of music described by onset times, track-ID numbers and offset times
- *mix-cue:* section of music described by onset times and sample-ID numbers
- *remix-cue:* section of music described by sample-ID numbers and the number of times each sample is to be repeated

| **Cue-type** | **Description** | **x-data (format)** | **y-data (format)** | **z-data (format)** |
|---|---|---|---|---|
| **scene** | action | seconds (float) | action (string) | - |
| **mood** | emotion (adjectives) | seconds (float) | adjective (string) | intensity (integer) |
| **Hevner** | emotion (octants) | seconds (float) | octant (integer) | intensity (integer) |
| **feature** | musical feature | seconds (float) | level (integer) | - |
| **trax** | music section | seconds (float) | track ID (integer) | offset (float) |
| **mix** | music section | seconds (float) | sample ID (integer) | - |
| **remix** | music section | repeats (integer) | sample ID (integer) | - |

Table 4. Cue-types: Represented data and formats

| **Cue-type** | **=> mood** | **=> Hevner** | **=> feature** | **=> trax** | **=> mix** |
|---|---|---|---|---|---|
| **scene =>** | - | ConceptNet | - | - | - |
| **mood =>** | - | table | - | - | - |
| **Hevner =>** | table | - | F.-E. map | - | - |
| **feature =>** | - | - | - | - | lib query |
| **mix =>** | - | - | - | arrange | - |
| **remix =>** | - | - | - | - | arrange |

Table 5. Cue-conversion methods (F.-E. map = feature-emotion map)

N.B.: To eliminate the need to remember index numbers, wherever a cue_ID is required, users may instead use a period ('.') to represent the most recently added cue_ID.

*Converting cue-types:* Certain cue-types can be converted into other cue-types. Table 5 lists cue-conversions and the methods employed by each conversion. Cue-conversion uses the syntax,

```
        >> make-type cue_ID
```

where *type* represents the destination cue-type and *cue_ID* represents the index number of the source cue. Specific cue type-conversions are detailed below. All cue-conversions return a cue with the same number of points, except as noted.

• *scene => Hevner:* The Natural Language Processing toolkit ConceptNet (Liu/Singh, 2004) is used to infer an emotion-group based on the action-description at each point in the cue. The "guess_mood" method provided by ConceptNet is altered to allow a "guess_hevner" inference.

• *mood => Hevner:* A simple table lookup converts any adjectives used in the Hevner emotion-groups to **Hevner octants** for each point in the cue.

• *Hevner => mood:* A simple table lookup converts **Hevner octants** to adjectives for each point in the cue.

• *Hevner => feature:* Using a **feature-emotion map**, octants are converted to high or low levels of the specified musical feature for each point in the cue.

• *feature => mix:* A single feature-based query of the soundtrack library returns the highest-ranked sample for each point in the cue.

• *mix => trax:* For each sample in the cue, the track start-time is used as reference for conversion. Consecutive samples can be combined and represented as a single uninterrupted section in a *trax-cue*, thus eliminating pops that may occur between samples. If the supplied *mix-cue* contains consecutive samples, the returned cue will contain fewer points than the original.

- *remix => mix:* Repeated samples are placed end-to-end for each point in the cue. If the supplied *remix-cue* contains repeat-values of 2 or higher, the returned cue will contain more points than the original.

*Automatically adding a cue:* Users may generate a *mix-cue* by supplying a starting sample ID number and duration (in seconds), using the following syntax:

```
    >> qsamp sample_ID duration
```

Moodtrack creates the cue by appending samples until the target duration is achieved. This function is useful to create a section of music on which to base the creation of new sections.

*Building comparable sections of music:* The following steps describe building a new section of music for a scene analogous to another section supplied as reference, or **goal cue**. Comparable music section-building can employ **annotated feature-based retrieval, extracted feature-based retrieval** and **emotion-based retrieval** methods introduced in Section 3.8.

*Setting the goal:* To create a section of music comparable to a reference section, the user must first specify a soundtrack library and *mix-cue* as the referential **goal cue** by typing 'goal lib_ID cue_ID' at the prompt.

*Setting method control parameters:* After an existing *mix-cue* has been supplied as the **goal,** users may set **method control parameters** of the **Section-Builder** class, including **cut points, anchor**, and **rank**. During section-building, **cut points** are used to define sub-sections of the goal cue. The section-builder creates continuous sections within each defined sub-section. Logical **cut points** might follow on-screen cuts or other demarkations where breaks in the musical continuity may occur more safely. The **anchor** is a single point in time around which the entire section is arranged. While other points in the supplied **goal cue** may shift slightly as the section is built, the **anchor** will remain unchanged. The moment in the scene that requires the highest level of synchronization between music and film should be assigned as the anchor. The presence of a dramatic "**stinger**" creates a logical point to use as the **anchor**. The **rank** specifies the level of satisfied criteria. If using the same library, setting the **rank** to zero (0) returns the same section used as the **goal**, so the default **rank** is one (1). Setting the **anchor** and a single **cut point** at zero (0) builds a single continuous section of music. The **cut points** and **anchor** number correspond to points in the goal cue. **Cut points**, **anchor** and **rank** parameters are set, respectively, using the syntax:

```
>> cuts point1, point2, point3, etc.
>> anchor point
>> rank number
```

*Retrieval methods:* The **Section-Builder** class provides methods for music-retrieval and music-assembly based on a set of musical features, in this case the **goal cue**. To compare audio sample-candidates with the target, separate retrieval methods examine the following data-collection values as functions of time:

- *annotated features:* set of nine musical features supplied by the annotator
- *extracted features:* automatically-extracted musical features
- *annotated emotions:* self-reported emotion responses supplied by the annotator

*Hybrid emotion-feature retrieval:* The final method of retrieval involves building a section of music without reference to a **goal cue**. An emotion-feature retrieval requires a **temporal emotion cue**, which is translated into a set of musical feature cues based on the empirical data relating musical features and emotions. A **temporal emotion cue** is sent to the **Feature-Detector** class, which references the currently loaded **emotion-feature map** to lookup approximate values for each feature in the data set. Because the data only includes recommended feature dimension-polarities for each emotion, feature levels are set to mid points within the range of each pole, i.e. 25 (low feature level) and 75 (high feature level). The feature levels are then adjusted based on intensity levels of the emotions, producing the necessary set of musical feature cues.

*Rating methods:* Candidates for each sub-section are found by iteration through the entire soundtrack library, and considering each audio sample as potential starting point of the new section. Because the candidate and goal sections consist of audio samples with different durations, sample-to-sample comparison would not yield accurate comparisons. Ratings are re-examined at a default frequency of twice per second (2 Hz), which can be adjusted as needed. For each rating period, the absolute value of the difference between the goal and candidate is added to the candidate's total score. Each retrieval method provided by **Section-Builder** use a different method to return a rating, expressed in pseudo-code as:

- *annotated features:*

```
for feature in annotated_features:
     rating += abs(goal[feature] - candidate[feature])
return rating
```

- *extracted features:*

```
for feature in extracted_features:
     rating += abs(goal[feature] - candidate[feature])
return rating
```

- *annotated emotions:* (converted to *arousal-valence* emotion to allow for distance-comparisons)

```
rating += abs(goal_arousal - candidate_arousal)
                + abs(goal_valence - candidate_valence)
return rating
```

The list of candidates in each sub-section is sorted, placing the most appropriate (with the lowest returned rating value) at the top of the list.

*Assembly methods:* The **Section-Builder** class uses the same method of music-assembly, regardless of the retrieval method used. Based on the rank set by the user, consecutive audio samples are selected and concatenated until the addition of another audio sample would result in overlapping sub-sections (or, for the final **cut point**, exceeding the **goal**'s duration). The result is a sequence of audio samples for each sub-section that falls short (by less than the duration of the next consecutive audio segment) of the required time defined by the **cut points** or duration. The available retrieval and assembly methods are executed using the following syntax:

- annotated feature-based retrieval/assembly: 'fbuild'
- extracted feature-based retrieval/assembly: 'xbuild'
- annotated emotion-based retrieval/assembly: 'ebuild'

*Filling in the gaps:* To make up for differences between the sub-section lengths of the **goal** and the assembled, the user is provided optional methods to fill in the gaps. First, gap lengths are determined and sorted to find the largest gap before and after the anchor. An additional audio sample is appended to the sub-section having the widest gap. The surrounding timings are adjusted accordingly to accomodate the additional material. The anchor is preserved by shifting sub-sections away from the anchor point. In other words, pre-anchor points are shifted earlier, while post-anchor points are shifted later. The 'fold' command is used to determine the gap lengths of each sub-section and return them to Section-Builder. The 'fill' command iterates through the gap-filling process described above until the section can no longer accommodate any more material, and returns the resulting section of music. If the section was already full, no section is returned.

*Musical output:* Musical sections created by the arranger model are output as Csound files, either to disk or as a live performance. The command 'p cue_ID' writes the necessary Csound orchestra file and plays the specified cue via the machine's audio system. Entering the command 'w cue_ID' writes both the Csound orchestra and audio file to disk. The exported audio file can be 'Added to movie' in a media tool such as Quicktime to allow synchronized viewing of the film and its newly-assembled soundtrack.

*Plotting data:* To visualize and compare data contained in the library, a variety of graphical plots are offered using the Python package *matplotlib* (Hunter, 2006). The command 'compare feature1, feature2' figures a dual-bar graph to compare two sets of feature data (*feature1* and *feature2*), involving any combination of annotated and extracted musical features. Entering the command 'subftrs' iterates through all the current libraries and compiles a box-and-whisker plot of the all annotated features. For this to be useful, identical libraries annotated by different editors must be loaded into the system. A plot of musical features similar to the GUI plot is offered by entering 'linftrs', while the analogous emotion plot is figured using the command 'linemos'. Using the mapping to two-dimensional space discussed in Section 3, the command 'eplot' will figure a plot of *arousal* and *valence* components of the annotated emotion. The 'plot' command draws all the figured plots to the screen.

*Analyzing data:* Users may perform basic statistical functions through an interface with the SciPy scientific package (SciPy, 2006). Correlation coefficients for a musical feature among sets of reviewers are displayed by entering the 'coef feature' command. Basic statistical information of all currently loaded libraries is displayed via the 'describe' command.

# 5. EVALUATION

The components of the system are evaluated in the following manner:

*Data-collection site: The Film Game*

- player comments are discussed
- player data is compared with results acquired using more traditional methods

*Annotation GUI: Media Content Librarian*

- music experts are asked to use the GUI
- an open-ended questionnaire covers ease-of-use, clarity and completeness
- data entered by music experts is compared and diagrammed

*Music-Arranger Model: Media Content Shell*

- participants respond to short excerpts of film accompanied by music rendered with the system
- an open-ended questionnaire covers fit and continuity of the music for the scene

## 5.1 Evaluation of the data-collection site

The challenge of creating a music-emotion game is to make the game both fun for players and useful for data-collection. To address both issues, players' responses to the game are discussed, and data collected during game play is presented.

### 5.1.1 Player response

An early version of the game consisted of many questions for each music clip. At this preliminary point, one player observed, "Calling this a game is a little optimistic!" Another suggested each question be separated into its own page. With these suggestions, a simpler game allowed users to progress through each film more quickly. An expert player commented that people involved in film music may want to provide more details. Following this second suggestion, the game was redesigned to include variations that allow the player to choose their desired level of interaction.

While the simple and advanced games provide basic and expert players appropriate levels out of necessity, the 'remix' variation of the game adds a sense of novelty to the game. Because the music and film examples are randomly selected, the player can be continually surprised at the next outcome. If a player has seen the films excerpted for the game, the simple and advanced game variations do not have the element of surprise. One player said about the remix game, "It's fun because you never know what you're going to get." Another player agreed, saying, "That's cool...kind of like those trailer mash-ups you see on the web."

Some players found the pull-down menus lacked descriptive content, as one musically-inclined player commented, "I like the emotion choices, but the musical choices seemed a little limiting." A number of players felt the emotion list was limited, as one mentioned, "With those Hitchcock movies, you need 'suspense' in the list." Another player railed against an apparent bias towards positive emotions, saying, "You forgot the big three—dark, foreboding and evil!" A second player agreed, saying "I wanted words like 'scary' and 'creepy' and 'ominous' for the balcony and cliff scenes [from *North By Northwest*] for example." Several of the octant adjective lists that were consulted for the project contained the descriptor 'dark' in the octant adjective list, but the lists were culled down for the sake of web space.

Players often noted the difficulty of self-reporting, such as "The Film Game is fun, but it was hard to pick the right feature or mood. It took getting used to." Some players complained about the scoring system, saying the text box for a player's own words never returned a good score.

Indeed, a few points are awarded simply for entering text into the box. Future work could offer methods to evaluate open-ended descriptions of music using natural language tools such as ConceptNet (Liu/Singh, 2004).

### 5.1.2 Player data

The Film Game was open to players for a duration of two months (April-May 2006), during which time 22 players contributed data. The resulting data is compiled, revealing popular and unpopular feature-emotion relationships (Figure 13). In a side-by-side comparison with analogous data collected by Hevner and others (Gabriellsson/Lindstrom, 2001), the Film Game exhibits some peculiarities. With a few exceptions, many feature-emotion relationships observed by music-emotion researchers (and intuitive to musicians) are not revealed in the data. For example, while a fast *Tempo* is often highly correlated to *Vigorous* and *Exciting* emotions, and a slow *Tempo* often associated with *Sad* and *Dreamy* emotions, the data in Figure 13 shows conflicting results in both cases.

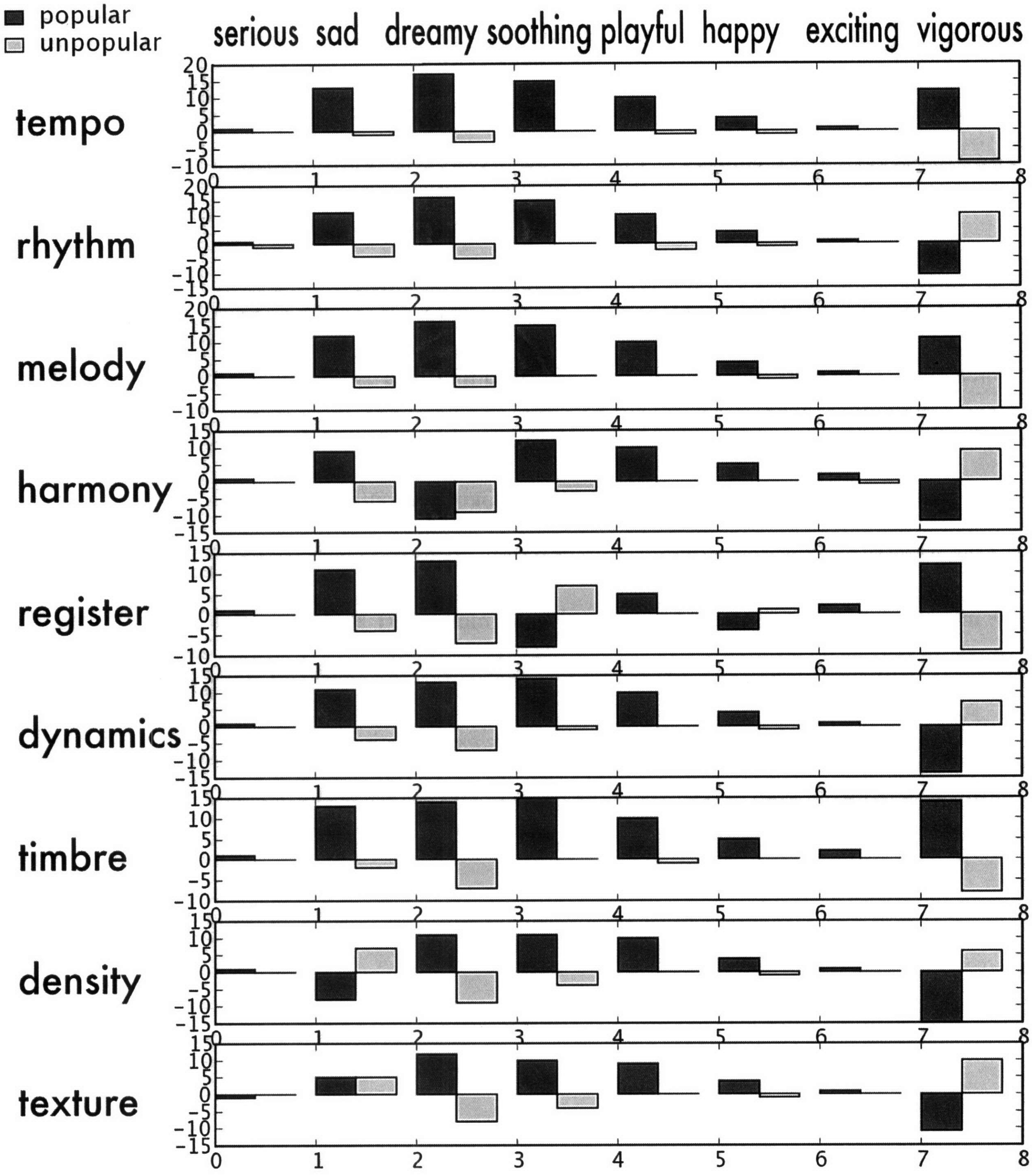


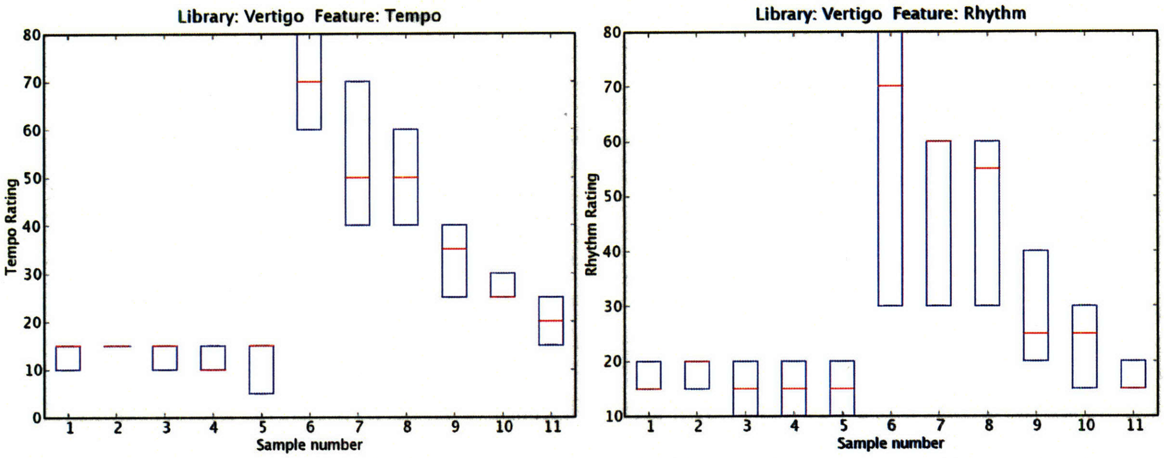
Figure 13. Film Game player data: feature-emotion map

### 5.2 Evaluation of the annotation GUI

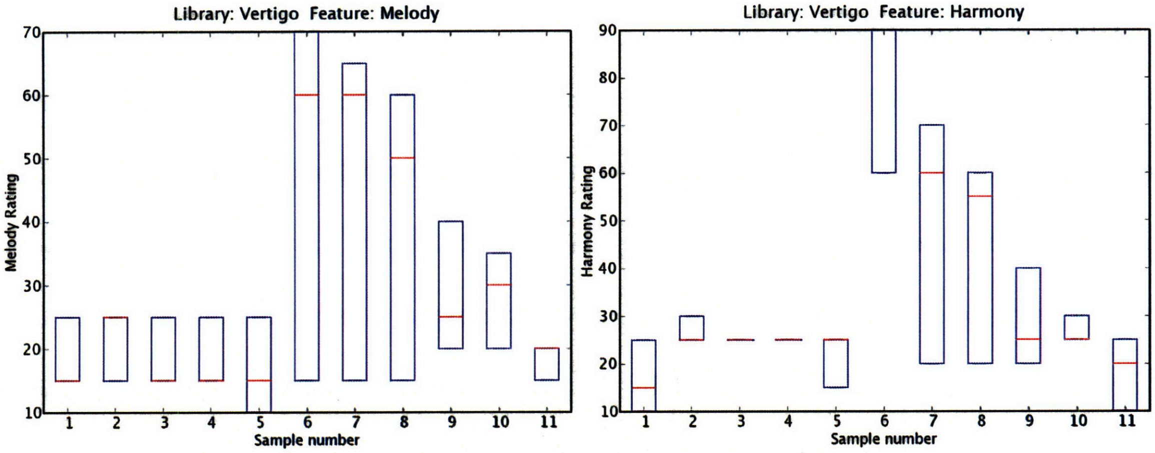
The **Media Content Librarian** was used to annotate material over a period of four months. During this time, the GUI was continually evaluated and revised for practical considerations. Though the repetitiveness and time-consuming nature of data-entry was recognized early in the development process, it was accepted as an unavoidable aspect of subjective annotation. After the revision process, the GUI was used by set of editors to annotate ten contiguous audio segments of Bernard Herrmann's "The Bay" from *Vertigo* (1958).

#### 5.2.1 User response

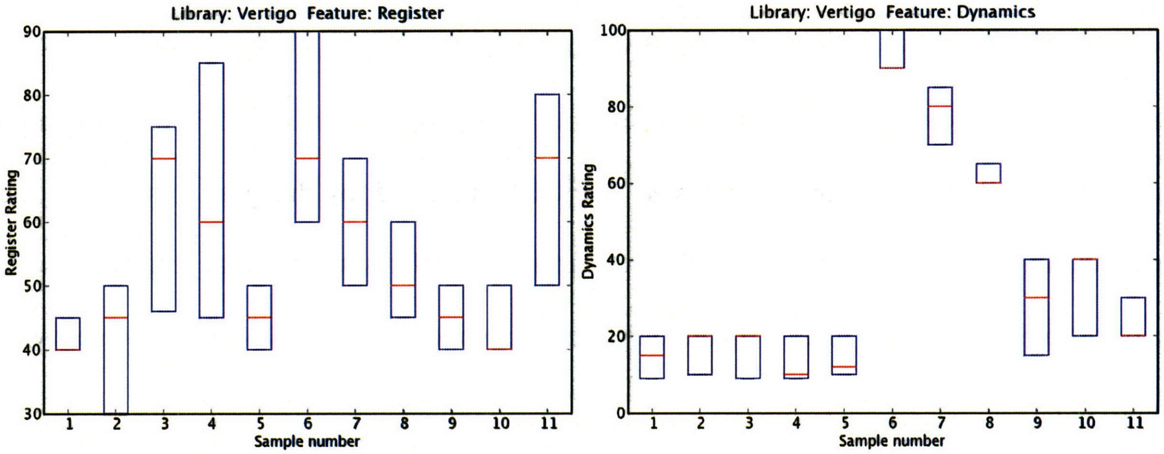
Most annotators commented that evaluation was a difficult process. In comparison to the Film Game, one user noted that the GUI "forces you to be more analytical." Editors appeared intrigued by the process, as one commented "the GUI structured my thoughts about the music [being annotated] in interesting ways." Not coincidentally perhaps, the musical feature *Texture,* cited most by editors as the most difficult to evaluate, exhibited the highest degree of subjectivity (Figure 22).



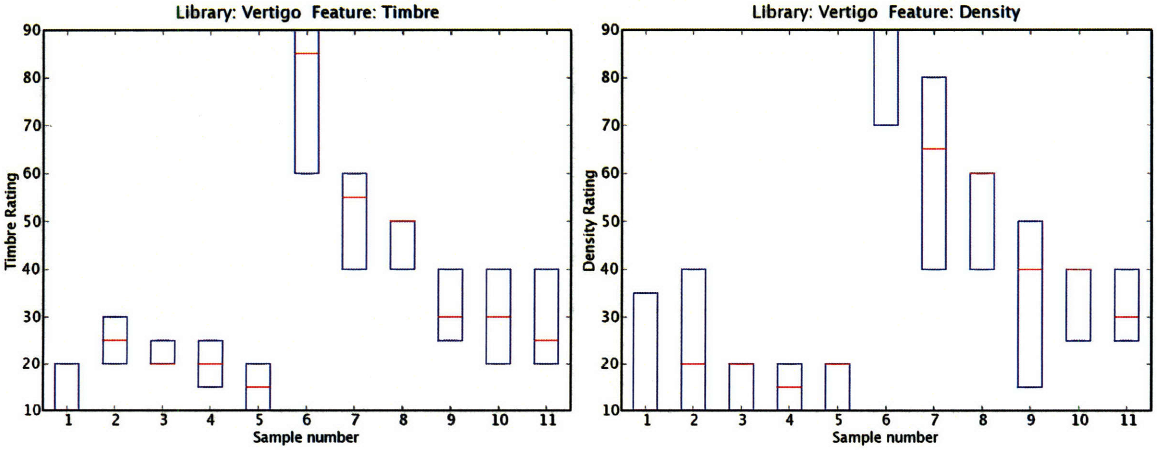
Figures 14 & 15. Annotated Tempo, Annotated Rhythm



Figures 16 & 17. Annotated Melody, Annotated Harmony



Figures 18 & 19. Annotated Register, Annotated Dynamics



Figures 20 & 21. Annotated Timbre, Annotated Density

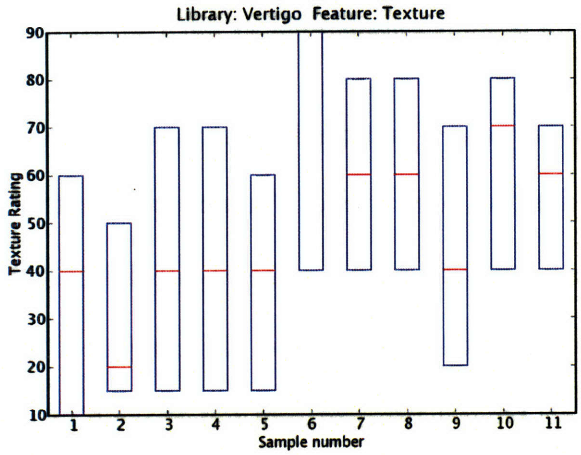


Figure 22. Annotated Texture

### 5.2.2 Subjectivity of musical features

Expert music-annotations are summarized in Figures 14 through 22. The size of the areas represents the degree of subjectivity of each musical feature given the set of editors and musical stimuli.

Several editors self-observed that, as some features moved in a certain direction, there was a bias towards making other features move in a similar direction. The *Texture* feature may reveal a "sympathetic feature-motion" bias in the resulting data. One editor noted a tendency to maintain feature level between segments. The seemingly higher subjectivity of features *Rhythm* and *Melody* in sample numbers 6 through 8 could be attributed to a "static feature-level" bias.

Abrupt musical changes appear to reveal a higher level of subjectivity in certain musical features. In the *Vertigo* excerpt, sample number 6 contains the music accompanying the **stinger** point, as Madeleine plunges into the water. Annotations of *Rhythm* and *Melody* in Samples 6 through 8 exhibit a wider spread, possibly due to users' hesitancy to annotate large leaps in these features. This "small leap" bias could be responsible for the wider spread of sample numbers 2, 3 and 6 of the musical feature *Register*. The features *Tempo* and *Dynamics* appear to be less susceptible to a preference for smaller leaps.

### 5.2.3 Comparing human-annotated and machine-extracted features

To lessen the need for annotation, some musical features can be extracted directly from the audio signal. While more extensive feature-extraction models are demonstrated elsewhere (Zils/Pachet 2004), four musical features used in this thesis are compared to three features commonly extracted by machine. Descriptive statistics of human-machine feature comparisons are shown in Tables 6 through 9, respectively comparing *Tempo/BPM*, *Dynamics/RMS*, *Density/Zero-crossings* and *Register/Zero-crossings.* Expert annotation of *Tempo* and the machine-extracted beats-per-minute (BPM) level offers a high-level of correlation (Table 7), such that editors may feel confidently use machine methods for extracting *Tempo*. Similarly, machine-extracted RMS appears a valid means of annotating *Dynamics* (Table 8). While *Tempo* and *Dynamics* may be extracted accurately by machine, other musical features (Tables 9 and 10) appear to require more sophisticated means of extraction (Whitman, 2005).

| subject | correl-coef | P | scalar |
|---|---|---|---|
| 1 | 0.962 | 2.248 | 0.533 |
| 2 | 0.973 | 5.025 | 0.711 |
| 3 | 0.868 | 0.001 | 0.622 |

Table 6. Annotated Tempo, Extracted BPM

| subject | correl-coef | P | scalar |
|---|---|---|---|
| 1 | 0.934 | 2.292 | 0.012 |
| 2 | 0.971 | 7.086 | 0.014 |
| 3 | 0.974 | 4.056 | 0.012 |

Table 7. Annotated Dynamics, Extracted RMS

| subject | correl-coef | P | scalar |
|---|---|---|---|
| 1 | -0.356 | 0.283 | 0.001 |
| 2 | -0.15 | 0.66 | 0.002 |
| 3 | 0.423 | 0.195 | 0.001 |

Table 8. Annotated Register, Extracted Zero-crossings

| subject | correl-coef | P | scalar |
|---|---|---|---|
| 1 | -0.439 | 0.177 | 0.002 |
| 2 | -0.586 | 0.058 | 0.002 |
| 3 | -0.573 | 0.065 | 0.001 |

Table 9. Annotated Density, Extracted Zero-crossings

### 5.3 Evaluation of the music-arranger model

The command-line interface itself was not considered for evaluation. Though it provides many useful functions, learning the commands and syntax was thought not to be worthy of user evaluation. Instead, music sections rendered using the system were the subject of evaluation.

Three excerpts were taken from classic and popular films. Scenes were chosen that involved a dramatic **stinger** on which to anchor the arrangement. The excerpts are as follows:

*Vertigo*: The Bay

The excerpt begins as Scottie (James Stewart) is quietly trailing Madeleine (Kim Novak). A **stinger** arrives as Madeleine plunges into the water, with Scottie rushing to dive in and save her. The original score by Bernard Herrmann continues throughout the entire excerpt.

*The Magnificent Seven*: The Duel

The excerpt depicts a knife-gun duel between Britt (James Coburn) and a townsman (uncredited). The violent end of the duel marks a clear **stinger** point. Elmer Bernstein begins his score for this scene as soon as the knife strikes the townsman.

*The Untouchables*: The Man with the Matches

The excerpt begins as an assassin (uncredited) enters Malone's (Sean Connery) apartment. The tension builds until the intruder is surprised at the sight of Malone wielding a shotgun. In his original score, Ennio Morricone provides music until the moment Malone turns around.

### 5.3.1 Selecting music to re-purpose

Soundtracks from the films *Blade Runner*, *Close Encounters of the Third Kind*, *The Adventures of Robin Hood* and *Vertigo* were chosen to segment and annotate. While the manual selection of annotated album tracks does not provide ideal testing conditions, the time required to segment and annotate libraries precluded the inclusion of entire soundtrack albums. The soundtrack segmented and annotated most extensively was Bernard Herrmann's *Vertigo*, chosen for its wide range of moods. In addition, Herrmann's compositional style lends itself well to repurposing in that he relies less on thematic associations (*leitmotifs*) than do other composers.

### 5.3.2 Building re-purposed music

To automatically re-score each excerpt, appropriate **anchor** point and **cut points** were selected. Within each section-building method, the eight best-**ranked** sections were auditioned and culled down to represent the most successful attempt for each film-soundtrack combination. Two music scores were presented for each of the three film clips described in Section 5.3.

### 5.3.3 User response

Overall, viewers reacted very favorably to the excerpts. For film excerpts scored with music from a different era or genre, many reactions were humorous at first. After watching the clips, however, many viewers would comment, "It's obvious that the music is from somewhere else, but it just kind of worked." After witnessing a familiar scene with a newly assembled music score, one viewer claimed the experience was "like seeing a film you know with fresh eyes." The viewer went on to say that the new music scores allow you to "re-see" familiar films.

Excerpts scored with music from a similar era and genre produced stronger reactions. At one point in such an excerpt, one viewer appeared confused, asking, "Is this just the original clip?" Viewers that were familiar with the films provided equally encouraging reactions, as one noted, "If I hadn't seen this film, I wouldn't know this wasn't the original score!"

# 6. CONCLUSION

Moodtrack is a multi-dimensional music-sensibility experiment in film music. Each of the three system components involves a unique media perspective: the Film Game (Section 4.1) as the *public view*, the GUI data (Section 4.2) as a *critical view*, and the arranger model as a *creative view* (Section 4.3). The components are intended to work together to assemble emotion-driven music based on the three perspectives. Though individual components may need work to achieve more usable results, automatically re-adapting musical material for film appears possible.

## 6.1 Collecting music-emotion data online

Several factors could be responsible for the fact that the Film Game did not produce data usable by the arranger model. Data from the three game variations are combined to produce the **feature-emotion map**. The volume of data collected by each separate game variation did not warrant a usable data set, resulting in the decision to combine data. By using data from the advanced game only, perhaps a more usable data set would be collected.

Offering the player two lists to describe emotions may give the appearance of allowing the user to more closely describe emotions. However, the result is overlapping data. Deciding a single adjective to describe emotional response may prove more difficult for the player, but in all likelihood the results will be more targeted.

Resulting data aside, players seemed to be responsive to the idea of a game based in film music. The number of positive reactions and comments received during testing of the Film Game should encourage other researchers to explore collecting music-emotion data in an online game format.

### 6.2 Annotating film and music

The annotation GUI provides a method for combined annotations of a film and its soundtrack. Music experts seemed particularly enthused about the concept of combined media-annotations. By using different editors, subjectivity problems may be difficult to avoid. On their own, users may provide consistent annotation data, but combined, may exhibit high level of variance. The Film Game randomized selections to help alleviate bias, but random-methods are not appropriate for an annotation tool. To alleviate subjectivity issues during music-retrieval, the model could accept weighted levels of each musical feature based on subjectivity.

### 6.3 Music-arranging methods

The feature-based music arrangement methods produced musical sections that in many cases were very usable and appropriate. Although the mismatch in genre was in some cases blatantly obvious, viewers responded very favorably. Continuity was revealed as the major issue of the arranger model. During the building process, otherwise usable sections were often effectively ruined by poor continuity.

Because of the need to map emotions to musical features, the resulting output using emotion-based arrangement was less targeted and appropriate than arrangements resulting from comparable feature-based methods. Out of the eight top-ranked arrangements, feature-based retrieval often produces several usable sections, while emotion-based retrieval rarely net more than one usable section of music. More extensive experimentation with these approaches to arrangement may reveal techniques and control settings that improve the usability of the resulting musical sections.

### 6.5 Future work

In addition to exploring the retrieval and placement methods used in this thesis, audio processing can be explored to improve continuity. Simply adding cross-fades between cuts will improve the overall continuity, particularly the feature *Dynamics*. Normal pitch-transposition of a segment can simultaneously adjust the features *Tempo* and *Register,* while a phase-vocoder can be used to adjust *Tempo* without adjusting *Register*.

Another method to improve continuity involves several attributes offered in the annotation GUI (such as *theme* and *action*), which are not considered during retrieval. The inclusion of these attributes would allow thematic elements to play a central role during the retrieval process, likely increasing continuity between segments. Continuity could also be enhanced through audio similarity-matching techniques that monitored transitions between the end of a segment and the beginning of the next segment in the arrangement. Alternately, changes in tempo, pitch register and other musical features can be applied.

The tedious tasks of audio segmentation and musical feature-analysis could be assisted or replaced by machine process. Machine-extraction of segments and analysis may still require human supervision, if only as a matter of preference. Grounding musical feature-dimensions in terms familiar to musicians may help bridge the gap between our natural-language descriptions of sound and machine-understanding of audio.

Text-annotation of commercially available music and film does not require an illegal exchange of audio files, possibly promoting the purchase of classic films and soundtracks. The use of highly-portable means of storage, such as text files, encourages long-distance collaboration over the web. Moodtrack uses a variety of text files, including the **contiguous regions list**,

**SoundtrackML**, **Moodtrack script** and Csound files. A project like the Gracenote CDDB (accessed by users of Apple's iTunes media player) relies on a large number of users to contribute annotations of audio CD's. While the Gracenote annotation is limited to artist, track and album names, a similar project could be initiated to include contiguous regions, musical features and emotions, as well as relational annotations of film and soundtrack media. Such a project could be immensely useful to re-mix artists and film music editors, as well as for music-emotion researchers.

### Soundtracks for the people!

Music researchers may find the **Film Game** an interesting approach to data-collection, while film and music professionals may find the **Media Content Librarian** a useful annotation tool. However, in a society where inexpensive video cameras and film-editing software are becoming ubiquitous, the **music-arranger model** holds the most promise for consumer interest. With a media-manipulation tool such as Moodtrack, the budding visual artist can work with a master composer, or a fan can experiment creating film music remixes. As the major contribution presented in this thesis, the approach taken with the music-arranger model is offered as a step towards emotion-driven soundtracks.

## APPENDIX 1. GLOSSARY OF TERMS

**Adapted music:** A film score is said to be adapted if the music existed prior to the film's inception. Adapted music includes pre-composed music arranged for the film as well as pre-recorded music selected for the film.

**Anchor:** Often coinciding with a **stinger**, an **anchor** is a point in time used as a reference to arrange musical sections. Cue points (including start and end points) sent to the **arranger model** may be shifted slightly for musical continuity, with the exception of the **anchor**.

**Annotation GUI:** see **Media Content Librarian**

**Arranger model:** see **Music-arranger model**

**Context-relative subjective response:** Self-reported response to an audio stimulus in **Hevner octant representation** or in a **musical feature** dimension, within the current media context.

**Contiguous regions list:** A contiguous regions list is a text file describing track timings and labels of audio regions. The list can be imported into audio editing software such as Audacity with the 'Import labels track' command (Audacity).

**Contiguously-segmented sample library:** Using a sound editing software such as Audacity, the editor must first mark (or import) a set of regions. With the 'Export multiple' command, all segments of sound between markers are saved to separate audio files, making the resulting sample library contiguously-segmented. Moodtrack uses aggregate sample durations to find matching feature points in a track.

**Cue:** In all **Moodtrack** components, a *cue* is a two- or three-dimensional data class. The Moodtrack *cue* is distinctly different from a "music cue," meaning a section of music lasting anywhere between a few seconds and several minutes in duration. In Moodtrack, a *cue* is a temporal data structure, with a representation of time on the x-axis, and various other data along the y- and z-axes, depending on cue-type. Cue-types include: *scene*, *mood*, *Hevner*, *feature*, *trax*, *mix* and *remix* (Section 4).

**Cut points: Method control parameter** of the Section-Builder class defining steps in a reference mix-cue

**Data-collection site:** see **Film Game**

**Diegetic/non-diegetic realms:**
**Diegetic:** The implied reality created by film is commonly referred to as the diegesis, which includes visual and emotional realms, as well as character dialogue, on-screen music and foley sound (sound effects). Diegetic mood cues arise only from sights and sounds in on-screen character reality. The research focuses on emotional cues that might result from one's empathy for the film's characters.

**Non-diegetic:** Presumably, the soundtrack, titles and voice-overs are perceived exclusively by the audience, so are considered parts of the non-diegetic aspect of film. For this study, emphasis is placed on non-diegetic emotion cues resulting from the soundtrack.

**Emotion-groups:** see **Hevner octant**

**Emotion-based music arrangement:** A section of music assembled according to temporal emotion cue input, using a **feature-emotion map** for reference.

**Feature-based music retrieval and arrangement:** A section of music comparable to an existing section of music, assembled by matching musical features over time.

**Feature-Detector:** Moodtrack class in the **music-arranger model** providing methods for mapping emotions to musical features using the **feature-emotion map**.

**Feature-emotion map:** Data correlating emotions and musical features provide the **music-arranger model** a means to infer features from emotions. A table extracted from Hevner's observations or similar data, such as that collected from the **Film Game**, provide music retrieval criteria. Naturally, changing **feature-emotion map** will alter the character of the **music-arranger model**.

**Film Game:** a set of web-based activities designed to collect music-emotion data

**Goal cue:** a mix-cue supplied to the Section-Builder class used as reference for comparable music section-building

**Graphical User Interface (GUI):** see **Media Content Librarian**

**Hevner octant:** eight-part emotion representation used by Kate Hevner (1936)
Emotion-groups represent *Serious, Sad, Dreamy, Soothing, Playful, Happy, Exciting* and *Vigorous* emotions (Section 3.2).

**Hevner octant-intensity vectors:** a representation of emotion described by a Hevner octant number with relative percentage (%) intensity represented by the vector length (Section 3.4)

**Media Content Librarian:** a GUI designed for experts to annotate a film and its soundtrack

**Media Content Shell:** a command-line interface for media producers
To access methods of the **music-arranger model**, a UNIX-like command-line interface allows users to enter individual commands. From the "mtk>>" shell, typing "filename" will run the **Moodtrack script file** named "filename.mtk" located in the script directory. For a full list of valid commands, see Appendix 3.

**Method control parameters:** Moodtrack takes a set of **method control parameters** that dictate the type and operation of the **music-arranger model** methods.

**Moodtrack class:** A Python class from the Moodtrack library. For a complete list of classes, see Appendix 4.

**Moodtrack script (.mtk) file:** Users can store a set of **Media Content Shell** commands as a newline-separated text file to execute together as a single script. For a complete list of commands, see Appendix 3.

**Music-arranger model:** Designed to assemble music sections, the model creates audio according to **temporal emotion cue** points. A number of methods are used, including **playlisting** of songs by mood and **algorithmic composition** of music note-lists. Using an **anchor** as a point reference, the arranger model can perform **feature-based** or **emotion-based** retrieval to build musical arrangements.

**Musical feature:** An aesthetic aspect of musical structure or perception, as used by music theorists and critics. Highly subjective features may not be possible to accurately extract automatically. Musical features used in this thesis are *Tempo, Rhythm, Melody, Harmony, Register, Dynamics, Timbre, Density* and *Texture.* (Section 3.3)

**Music supervisor:** The person on a film in charge of choosing selections from a music catalog. The supervisor often works closely with the director to select music whose genre and emotional content fit the scene. Music supervisors often select **temp tracks** for use in the film's pre-production and/or **adapted music** that remains in the final cut of the film.

**Playlist:** Lists of commercial audio tracks are typically called playlists. Emotion-based playlist-generation can be very effective because of the huge volume of annotated material, but adapting to **temporal emotion cues** is not addressed.

**Pseudo-timecode:** *SMPTE time code* in film typically uses the reel number, minute, second and frame number to describe a location in time. This project uses a slightly modified format consisting of the hour, minute, second and millisecond (i.e. 01:37:28:132 as 1 hour, 37 minutes, 28 seconds and 132 milliseconds). The **TimeCoder applet** can be used to translate **pseudo-timecode** to seconds, and vice-versa.

**Rank:** 1) the location of a sample in a **ranked-sample list**, 2) **method control parameter** in the **Section-Builder** class specifying the **rank** (meaning #1) used to select samples during section-building

**Ranked sample-list:** Based on a set of supplied criteria, several components in the music-arranger model return a ranked sample-list. A low rank corresponds to an increased amount of satisfied criteria.

**Re-purposing music:** Essentially the goal of the project, re-purposing music involves the application of music into a domain for which it was not originally intended. Specifically, this project explores taking music from a scene of film and arranging it to fit a different scene.

**Section-Builder:** Moodtrack class of the **music-arranger model** offering music section-building methods.

**SoundtrackML (.sndml) file:** proposed as an intuitive XML format for annotated soundtrack libraries; Moodtrack uses files saved in the format to exchange data between components. A single library is used to store data annotating a film (DVD) and its soundtrack (CD).

**Spotting session:** a meeting between the director, composer, producer, film editor and music editor to decide the placement of music in a film

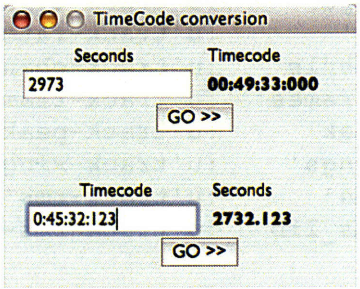
**Stinger:** Traditionally, the **stinger** is a point in time where the music and visual narrative of a film is at an emotional peak. In this project, the **anchor** often coincides with a **stinger**.

**Temporal emotion cues:** a description of emotion over time; This project uses **Hevner octants with arousal** as visual representation, which can be converted to a 2- or 3-dimension internal representation.

**Temporal musical feature cues:** a description of a musical feature over time; This project uses a concise set of nine musical features frequently used by musicologists and film music scholars (see **Musical Features**).

**Temp (temporary) track:** music selections that a director/music supervisor uses as underscoring before the composer has had a chance to score the film; Temp tracks are useful for those working on the film to get a sense of the timing and emotional content of the film. They can be dangerous if the director becomes accustomed to the temp track and requires the composer to deliver very similar-sounding music. Some composers explicitly avoid listening to the temp tracks to let their ideas grow naturally from the film's visual content.

**TimeCoder applet:** a simple utility that for converting between **pseudo-timecode** and seconds.



Seconds: 2973
Timecode: 00:49:33:000

Timecode: 0:45:32:123
Seconds: 2732.123

Figure 23. The TimeCoder applet

**Track: Moodtrack class** representing an extended section of audio, such as a CD track taken from the film's soundtrack album.

## APPENDIX 2: SOUNDTRACK MARKUP LANGUAGE

Each **SoundtrackML** tag corresponds to a unique Moodtrack class attribute. The following Python code relates a Moodtrack class attribute name (dictionary key, on left) to its unicode XML tag (dictionary value, on right):

```
# SampLib class:
libtags={  'id'       :u'lib-id',
           'title'    :u'lib-title',
           'name'     :u'lib-filename',
           'dir'      :u'lib-directory',
           'filmdir'  :u'lib-film-directory',
           'rms'      :u'lib-rms',
           'xings'    :u'lib-crossings',
           'composer' :u'lib-creator-composer',
           'director' :u'lib-creator-director',
           'editor'   :u'lib-creator-editor',
           'cues'     :u'lib-cues',
           'dateRel'  :u'lib-date-released',
           'dateEnc'  :u'lib-date-encoded',
           'samps'    :u'lib-samples',
           'tracks'   :u'lib-tracks',
           'durFilm'  :u'lib-film-length',
           'durMusic' :u'lib-music-length',
           'gnrFilm'  :u'lib-film-genre',
           'gnrMusic' :u'lib-music-genre'}

# Track class:
tracktags={'id'       :u'track-id',
           'title'    :u'track-title',
           'filename' :u'track-filename',
           'dur'      :u'track-length',
           'max'      :u'track-maximum',
           'maxtime'  :u'track-maxtime',
           'type'     :u'track-type',
           'sr'       :u'track-rate',
           'nchnls'   :u'track-channels',
           'nframes'  :u'track-frames',
           'peak'     :u'track-peak',
           'xings'    :u'track-xings',
           'rms'      :u'track-rms',
           'rms_lib'  :u'track-rms-library'}

TrackMap class:
maptags={'id'         :u'map-id',
           'name'     :u'map-name',
           'trkid'    :u'map-track-id',
           'trkin'    :u'map-track-in',
           'trkout'   :u'map-track-out',
           'filmin'   :u'map-film-in',
           'filmout'  :u'map-film-out'}
```

```
# Sample class:
samptags={'id'       :u'sample-id',
          'filename' :u'sample-filename',
          'looping'  :u'sample-looping',
          'type'     :u'sample-type',
          'sr'       :u'sample-rate',
          'nchnls'   :u'sample-channels',
          'nframes'  :u'sample-frames',
          'dur'      :u'sample-length',
          'max'      :u'sample-maximum',
          'maxtime'  :u'sample-maxtime',
          'peak'     :u'sample-peak',
          'xings'    :u'sample-crossings',
          'rms'      :u'sample-rms',
          'rms_trk'  :u'sample-rms-track',
          'rms_lib'  :u'sample-rms-library',
          'bpm'      :u'sample-bpm',
          'counts'   :u'sample-counts',
          'timeSig'  :u'sample-time-signature',
          'keySig'   :u'sample-key-signature',
          'root'     :u'sample-root',
          'trkId'    :u'source-id',
          'trkLib'   :u'source-library',
          'trkTitle' :u'source-title',
          'trkStart' :u'source-start',
          'cueStart' :u'source-cue-start',
          'cueId'    :u'source-cue-id',
          'cueName'  :u'source-cue-name',
          'ftrTmpo'  :u'feature-tempo',
          'ftrRthm'  :u'feature-rhythm',
          'ftrMldy'  :u'feature-melody',
          'ftrHarm'  :u'feature-harmony',
          'ftrRegs'  :u'feature-register',
          'ftrDyna'  :u'feature-dynamics',
          'ftrTimb'  :u'feature-timbre',
          'ftrDens'  :u'feature-density',
          'ftrTxtr'  :u'feature-texture',
          'mood1'    :u'sample-primary-hevner',
          'mood2'    :u'sample-secondary-hevner',
          'moody1'   :u'sample-primary-arousal',
          'moody2'   :u'sample-secondary-arousal',
          'typTheme' :u'sample-theme-type',
          'theme'    :u'sample-theme',
          'typAct'   :u'sample-action-type',
          'act'      :u'sample-action',
          'typInstr' :u'sample-instrument-type',
          'instr'    :u'sample-instrument'}

# Reel class:
reeltags={'id'       :u'reel-id',
          'name'     :u'reel-filename',
          'dur'      :u'reel-length'}
```

```
# Scene class:
scenetags={'id'       :u'scene-id',
           'name'     :u'scene-name',
           'reelid'   :u'source-reelid',
           'reelin'   :u'scene-reel-in',
           'reelout'  :u'scene-reel-out',
           'dur'      :u'scene-length'}

# Qpoint class:
qptags={'id'          :u'cue-id',
           'scene'    :u'cue-scene-id',
           'point'    :u'cue-point',
           'mood1'    :u'cue-primary-hevner',
           'mood2'    :u'cue-secondary-hevner',
           'moody1'   :u'cue-primary-arousal',
           'moody2'   :u'cue-secondary-arousal'}
```

The following is an example of **SoundtrackML** (.sndml) file:

```
<sample-library>
    <identification>
        <lib-id>0</lib-id>
        <lib-title>Vertigo</lib-title>
        <lib-filename>MyVertigo.sndml</lib-filename>
        <lib-directory>/usr/mtk/snd/</lib-directory>
        <lib-film-directory>/usr/mtk/mov/</lib-film-directory>
        <lib-creator-composer>Herrmann</lib-creator-composer>
        <lib-creator-director>Hitchcock</lib-creator-director>
        <lib-creator-editor>Scrappy V Bastard</lib-creator-editor>
        <lib-samples>1</lib-samples>
        <lib-tracks>1</lib-tracks>
        <lib-film-length>1342</lib-film-length>
        <lib-music-length>194.31</lib-music-length>
        <lib-film-genre>mystery</lib-film-genre>
        <lib-music-genre>classical</lib-music-genre>
    </identification>
    <track-list>
        <track>
            <track-id>1</track-id>
            <track-title>Prelude and Rooftop</track-title>
            <track-filename>01.aif</track-filename>
            <track-length>194.31</track-length>
        </track>
    </track-list>
    <reel-list>
        <reel>
            <reel-id>1</reel-id>
            <reel-filename>Vertigo1.mp4</reel-filename>
            <reel-length>1342</reel-length>
        </reel>
    </reel-list>
    <scene-list>
        <scene>
            <scene-id>1</scene-id>
            <scene-name>The Roof</scene-name>
            <source-reel-id>1</source-reel-id>
            <scene-reel-in>209.729</scene-reel-in>
            <scene-reel-out>296.669</scene-reel-out>
        </scene>
    </scene-list>
    <cue-list>
        <cue>
            <cue-id>1</cue-id>
            <cue-point>302.424</cue-point>
            <cue-scene-id>3</cue-scene-id>
            <cue-primary-hevner>3</cue-primary-hevner>
            <cue-secondary-hevner>0</cue-secondary-hevner>
            <cue-primary-arousal>75</cue-primary-arousal>
            <cue-secondary-arousal>0</cue-secondary-arousal>
            <cue-action>driving</cue-action>
        </cue>
    </cue-list>
    <sample-list>
        <sample>
            <sample-id>0</sample-id>
            <sample-filename>01-01.aif</sample-filename>
            <sample-bpm>121.3142</sample-bpm>
            <sample-counts>8</sample-counts>
            <sample-key-signature>0</sample-key-signature>
            <sample-root>60</sample-root>
            <sample-length>3.956667</sample-length>
            <source-id>1</source-id>
            <source-start>0</source-start>
            <feature-tempo>60</feature-tempo>
            <feature-rhythm>55</feature-rhythm>
            <feature-melody>52</feature-melody>
            <feature-harmony>55</feature-harmony>
            <feature-register>60</feature-register>
            <feature-dynamics>36</feature-dynamics>
            <feature-timbre>49</feature-timbre>
            <feature-density>50</feature-density>
            <feature-texture>55</feature-texture>
            <sample-primary-hevner>1</sample-primary-hevner>
            <sample-secondary-hevner>3</sample-secondary-hevner>
            <sample-primary-arousal>45</sample-primary-arousal>
            <sample-secondary-arousal>5</sample-secondary-arousal>
            <sample-looping>true</sample-looping>
        </sample>
    </sample-list>
</sample-library>
```

## APPENDIX 3: MOODTRACK COMMANDS

Single commands are typed into the **Media Content Shell**. Multiple commands can be read from a **Moodtrack script (.mtk) file** and executed from the shell.

| **Command** | **Description** |
|---|---|
| ? | show commands |
| ?all | show documentation |
| ls | list elements |
| ls -l | list library (.sndml) |
| ls -m | list feature-emotion map |
| ls -q | list cues |
| ls -s | list samples |
| ls -t | list tracks |
| load filename | load .sndml file |
| xload filename | load .sndml file/extract features |
| save | save .sndml file |
| cue-type: data | enter cue |
| make-type: # | convert cue-type |
| rmq cue_ID | remove cue at *cue_ID* |
| qsamp samp_ID dur | create cue at *samp_ID* for *dur* seconds |
| goal lib_ID cue_ID | set goal library and cue |
| cuts cut1, cut2, ... | set cut points in goal |
| anchor point | set anchor point |
| rank number | set rank |
| fbuild | feature-based section-build |
| xbuild | extracted feature-based section-build |
| ebuild | emotion-based section-build |
| fold | find gaps between goal and built-section |
| fill | fill gaps in built-section |
| pwd | print library/script/output directories |
| p | play audio |
| w cue_ID | write audio to disk |
| compare ftr1, ftr2 | compare two features |
| subftrs | compare annotations |
| linftrs | plot features |
| linemos | plot emotions |
| eplot | plot emotions (*arousal-valence*) |
| coef | calculate correlation coefficients |
| describe | show statistical data |
| filename | execute Moodtrack script filename.mtk |
| q | quit interface |

## APPENDIX 4: MOODTRACK CLASSES

The **Media Content Shell** interface consists of the following Python classes:

| **Class** | **Description** |
|---|---|
| App | command-line shell application |
| ConceptNet | interface to ConceptNet language tools |
| Cue | general data format: 2- or 3-dimension |
| EmoMap | feature-emotion map |
| FeatureDetector | set of musical features |
| Lab | plot/statistics (matplotlib/SciPy) |
| Moodscript | script (.mtk) file-reader |
| Orc | Csound file (.csd) writer |
| Qpoint | film annotation cue point |
| Reel | DVD TOC-ID movie (.mov/.mp4) |
| Sample | audio segment (.aif/.wav) |
| SampLib | media library (.sndml) |
| SectionBuilder | build methods |
| Track | CD audio track (.aif/.wav) |
| TrackMap | Track/Reel timing relationship |

## APPENDIX 5: SCRIPT EXAMPLES

Script 1 takes from *The Adventures of Robin Hood* (1938) to score an excerpt from *Vertigo* (1958) using human-annotated features as reference:

```
load Vertigo
load RobinHood
goal 0
qsamp 107 107
cuts 5
anchor 5
rank 0
fbuild
w .
```

Script 2 takes music from *Vertigo* (1958) to score an excerpt from *The Untouchables* (1987) using emotion-annnotations as reference:

```
load Untouchables
load Vertigo
goal 0
qsamp 0 70
cuts 1,2,3
anchor 3
rank 1
ebuild
w .
```

Script 3 uses music from *The Adventures of Robin Hood* (1938) to score an excerpt from *Vertigo* (1958) using machine-extracted features as reference:

```
xload Vertigo
xload RobinHood
goal 0
qsamp 107 107
cuts 5
anchor 5
rank 0
xbuild
w .
```

Script 4 loads three different annotations of the same musical stimuli and draws a box-plot to compare features (Figures 14 through 22).

```
load Vertigo_DS
load Vertigo_SV
load Vertigo_KH
subftrs
plot
```

## REFERENCES

*2001: A Space Odyssey*. Dir. Stanley Kubrick. Adapted music by Gyorgy Ligeti. MGM, 1968.

*The Adventures of Robin Hood.* Dir. Michael Curtiz & William Keighley. Music by Erich Wolfgang Korngold. Warner Bros., 1938.

Apple Computer, Inc. (2006). *Quicktime Movie Basics.* Taken from http://developer.apple.com/documentation/QuickTime/

*Audacity* (2006). Open-source audio editor, originally developed by Dominic Mazzoni and Roger Dannenberg. Downloaded from http://audacity.sourceforge.net

Barry, A., Faber, J., Law, D., Parsley, J., Perlman, G. (2005). *REALbasic Reference Manual.* REAL Software Inc.

*Blade Runner.* Dir. Ridley Scott. Music by Vangelis. The Ladd Company, 1982.

Boulanger, R. C. (2000). *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. MIT Press.

Close Encounters of the Third Kind. Dir. Steven Spielberg. Music by John Williams. Columbia Pictures, 1977.

Cohen, A. (2001). Music as a source of emotion in film. *Music and Emotion: Theory and Research*, Juslin, P.N and Sloboda, J.A. (Eds.), pp. 77-104. Oxford University Press.

Cohen, A., Bolivar, V. and Fentess, J. (1994). Semantic and formal congruency in music and motion pictures: effects on the interpretation of visual action. *Psychomusicology*, 13, 28-59.

Copland, A. (1939/1957). *What to Listen for in Music.* Revised edition. McGraw Hill.

Chung, J. & Vercoe, S. (2006). The Affective Remixer: personalized music-arranging. *Proc. of ACM CHI '06 Conference on Human factors in computing systems.*

Davis, R. (1999). *Complete Guide to Film Scoring.* Berklee Press.

Ekman, P. (1999) Basic emotions. *The Handbook of Cognition and Emotion*, Dalgleish, T. & Power, T. (Eds.), pp. 45-60, John Wiley & Sons, Ltd.

Dorai, C. and Venkatesh, S. (2003). Bridging the semantic gap with computational media aesthetics. *IEEE Multimedia Journal.*

Gabrielsson, A. and Lindstrom, E. (2001). The influence of musical structure on emotional expression. *Music and Emotion: Theory and Research*, Juslin, P.N and Sloboda, J.A. (Eds.), pp. 223-248. Oxford University Press.

*Good Morning Vietnam*. Dir. Barry Levinson. Adapted music by Louis Armstrong. Touchstone Pictures, 1988.

Harris, K. (2006). Interview with kindergarten teacher and musician Kemp Harris.

Hawley, M. (1993). "Structure out of sound," Ph.D. thesis, Media Laboratory, MIT.

Healey, J., Dabek, F. and Picard, R. (1998). A New Affect-Perceiving Interface and Its Application to Personalized Music Selection, *Proc. from the 1998 Workshop on Perceptual User Interfaces*.

Hevner, K. (1936). Experimental studies of the elements of expression in music. *American Journal of Psychology*, Vol. 48, 246-268.

Hunter, J. (2006). *The Matplotlib User's Guide.* Taken from http://matplotlib.sourceforge.net/

Huron, D. (2006). Music & emotion notes for graduate course-work at Ohio State University. Taken from http://csml.som.ohio-state.edu/Music829D/music829D.html on 8/1/06

*Jaws*. Dir. Steven Spielberg. Music by John Williams. Universal Pictures, 1975.

Jewell, M., Nixon, M. and Prugel-Bennett, A. (2003). CBS: A Concept-based sequencer for soundtrack composition. *Proc. of 3rd International Conference on Web Delivering of Music*, pp.105-108.

Juslin, P. N. (1997). Perceived emotional expression in synthesized performances of a short melody: Capturing the listener's judgement policy, *Musicae Scientiae*, 1:225-256.

Karlin, F. (1994). *Listening to Movies: The Film Lover's Guide to Film Music.* Macmillan Publishing.

*King Kong*. Dir. Merian Cooper & Ernest Schoedsack. Music by Max Steiner. RKO Radio Pictures, 1933.

Lesaffre, M., Leman, M., De Baets, B. and Martens, J. (2004). Methodological consideration concerning manual annotation of musical audio in function of algorithm development. *Proc. of the 5th International Conference on Music Retrieval*, pp. 64-71.

Liu, H. and Singh, P. (2004). ConceptNet: a practical commonsense reasoning toolkit. *BT Technology Journal*, Vol. 22. Kluwer Academic Publishers.

Lipscomb, S. and Tolchinsky, D. (2004). The role of music communication in cinema. *Music Communication*, Miell, MacDonald, Hargreaves (Eds.). Oxford Press.

Livingstone, S. R., Brown, A. R., Muhlberger, R. (2005). Influencing the Perceived Emotions of Music with Intent. *Third International Conference on Generative Systems*.

MacTheRipper (2006). DVD extractor. Downloaded from http://www.mactheripper.org

*The Magnificent Seven.* Dir. John Sturges. Music by Elmer Bernstein. MGM, 1960.

Marks, Martin Miller (1997). *Music and the Silent Film: Contexts and Case Studies, 1895-1924.* Oxford University Press.

Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology: Developmental, Learning, Personality, Social*, 14, 261-292.

Meyer, Leonard B. (1956). Emotion and Meaning in Music. Chicago: Chicago University Press.

Minsky, M. (forthcoming) *The Emotion Machine.* Taken from http://web.media.mit.edu/~minsky/E6/eb6.html on 8/1/2006

Minsky, M. (1986). *The Society of Mind.* New York: Simon and Schuster.

Minsky, M. (1981). Music, Mind and Meaning. *Computer Music Journal*, Fall 1981, Vol. 5, Number 3.

North, A. C. and Hargreaves, D. J. (1997). Liking, arousal potential and the emotions expressed by music. *Scandinavian Journal of Psychology* 38:47.

*North By Northwest*. Dir. Alfred Hitchcock. Music by Bernard Herrmann. MGM, 1959. Re-issue on Warner Home Video, 2004.

*OpenShiiva* (2006). Open-source video and audio vob to mp4 converter. Downloaded from http://openshiiva.sourceforge.net/

Picard, R. W. (1997). *Affective Computing*. MIT Press.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. Plutchik, R. & Kellerman, H. (Eds.), *Emotion: Theory, research, and experience: Vol. 1. Theories of emotion* (pp. 3-33). New York: Academic.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:6, pp. 1161-1178, American Psychological Association.

Scherer, K. R. and Zentner, M. R. (2001). Music production rules. *Music and Emotion: Theory and Research*, Juslin, P. N. and Sloboda, J. (Eds.), pp. 361-392. Oxford University Press.

Schubert, E. (1999). *Measurement and Time Series Analysis of Emotion in Music.* Doctoral dissertation, University of New South Wales.

Schubert, E. (2001). Continuous measurement of self-report emotional response to music. *Music and Emotion: Theory and Research*, Juslin, P. N. and Sloboda, J. (Eds.), pp. 77-104. Oxford University Press.

SciPy (2006). Scientific computing package for Python. Downloaded from http://www.scipy.org/

Sloboda, J. and O'Neill, S. (2001). Emotions in everyday listening to music. *Music and Emotion: Theory and Research*, Juslin, P. N. and Sloboda, J. (Eds.), pp. 415-430. Oxford University Press.

Sonnenschein, D. (2001). *Sound Design: The Expressive Power of Music, Voice, and Sound Effects in Cinema*. Studio City: Michael Wiese Productions.

Thomas, D., and Hansson, D. H. (2006). *Agile Development with Rails*. Pragmatic Bookshelf.

*The Untouchables.* Dir. Brian De Palma. Music by Ennio Morricone. Paramount Pictures, 1987.

van Rossum, G. (2006). *Python Reference Manual, v.2.4.3.* Drake, F., editor. Taken from http://docs.python.org/ref/

Vercoe, B., et al. (1992). *The Cannonical Csound Reference Manual.* Taken from http://www.csounds.com/manual/html/index.html

*Vertigo*. Dir. Alfred Hitchcock. Music by Bernard Herrmann. Universal Studios, 1958.

von Ahn, L. (2006). Games with a Purpose. *IEEE Computer*, June 2006.

Whissell, C. M., Fournier, M., Pellad, R.,Weir, D. & Makarec, K. (1986). A dictionary of affect in language: IV. Reliability, validity and application. *Perceptual and Motor Skills*, 62, 875-888.

Whitman, B. (2005) Learning the meaning of music. Doctoral Dissertation. MIT.

Yang, D. and Lee, W. (2004). Disambiguating music emotion using software agents. *Proc. of the 5th International Conference on Music Retrieval*, pp. 52-57.

Zils, A. and Pachet, F. (2004) "Automatic extraction of music descriptors from acoustic signals using EDS." *Proc. of the 116th Convention of the Audio Engineering Society.*